symmetry

MDPI

Article

Automatic, Illumination-Invariant and Real-Time Green-Screen Keying Using Deeply Guided Linear Models

Hanxi Li [1,2,*], Wenyu Zhu [1], Haiqiang Jin [2] and Yong Ma [1]

[1] School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China; stupidzwy@jxnu.edu.cn (W.Z.); may@jxnu.edu.cn (Y.M.)
[2] SpritAR, Shanghai 201201, China; jinhaiqiang2007@gmail.com
* Correspondence: hanxi.li@jxnu.edu.cn

Abstract: The conventional green screen keying method requires users' interaction to guide the whole process and usually assumes a well-controlled illumination environment. In the era of "we-media", millions of short videos are shared online every day, and most of them are produced by amateurs in relatively poor conditions. As a result, a fully automatic, real-time, and illumination-robust keying method would be very helpful and commercially promising in this era. In this paper, we propose a linear model guided by deep learning prediction to solve this problem. The simple, yet effective algorithm inherits the robustness of the deep-learning-based segmentation method, as well as the high matting quality of energy-minimization-based matting algorithms. Furthermore, thanks to the introduction of linear models, the proposed minimization problem is much less complex, and thus, real-time green screen keying is achieved. In the experiment, our algorithm achieved comparable keying performance to the manual keying software and deep-learning-based methods while beating other shallow matting algorithms in terms of accuracy. As for the matting speed and robustness, which are critical for a practical matting system, the proposed method significantly outperformed all the compared methods and showed superiority over all the off-the-self approaches.

Keywords: chroma key; deep learning; image matting; linear discriminant analysis; soft segmentation

Citation: Li, H.; Zhu, W.; Jin, H.; Ma, Y. Automatic, Illumination-Invariant and Real-Time Green-Screen Keying Using Deeply Guided Linear Models. *Symmetry* **2021**, *13*, 1454. https://doi.org/10.3390/sym13081454

Academic Editor: Gianluca Vinti

Received: 22 June 2021
Accepted: 3 August 2021
Published: 9 August 2021

1. Introduction

Thanks to the rapid development of computer graphics, the compositing shot has become a common choice in the film and television industry. Green/blue screen keying plays a crucial role in image/video compositing [1] and has already shown its production-level matting quality in many applications. This "well-developed" technology, however, requires professional users' guidance and other ad hoc settings such as a specially designed lighting apparatus for even illumination and a matte screen material to reduce light reflection. In recent years, with the surge of "we-media", millions of short videos are shared online every day, and most of them are produced by amateurs in relatively poor conditions. As a result, an Automatic, Illumination-invariant and Real-time (AIR) keying method could be very commercially promising in the age of the mobile Internet. In this paper, we propose a totally automatic and real-time green screen keying algorithm for unconstrained scenarios such as screens with natural light, shadows, and marks on them. Though little attention has been given by the research community, as we show later in this paper, achieving an "AIR" keying algorithm is not a trivial task. Firstly, it is hard to directly employ the existing keying methods [2,3] in AIR keying as there is no human mark or interaction in the process. Secondly, the sophisticated matting algorithms [4–7] also need initialization annotation by humans and cannot perform sufficiently well in video processing production. Most recently, the deep-learning-based matting algorithms have illustrated their high robustness in very challenging scenarios [8–11]. However, due to the high computational complexity, they cannot achieve real-time speed on low-resolution (typically below 512×512) images. This resolution cannot meet the basic requirements of today's video or image applications,

which usually require at least 1080P frames. One can of course upsample the low-resolution matting result to higher resolution, but the pixelwise matting accuracy will decrease significantly. Actually, the contradiction between the requirements of pixelwise accuracy and real-time speed is a long-standing and essential problem in the research of deep learning. In this work, we tried to address this long-standing problem by introducing deeply guided linear models and a framework for smartly combining deep models and shallow models. In the training stage, a deep network was trained to robustly classify each pixel into foreground and background, on low-resolution images. When testing, linear models were trained online under the supervision of the deep network, and then, the α value for each pixel was determined in a coarse-to-fine style. The yielded green screen keying method is totally Automatic, Illumination-invariant, and Real-time (AIR). It achieved much better matting results than the existing shallow and deep matting approaches, in terms of accuracy, speed, and robustness. When compared to the state-of-the-art commercial keying software with human interactions, our method illustrated comparable accuracy and overwhelming superiority on speed. The contribution of this work is three-fold:

- First, to the best of our knowledge, our keying algorithm is the first AIR keying method in the literature;
- Second, the combination between the coarse output of deep learning and an online-trained linear model is novel and also inspiring from the perspective of machine learning [12,13];
- Finally, to conduct a more comprehensive evaluation, we designed and generated a new green screen dataset, Green-2018. This dataset is not only larger than the existing ones [3], but also contains much more variances in terms of the foreground object category, the illumination changes, and the texture pattern of the green screens. This dataset is suitable to design better algorithm for the more challenging tasks such as outdoor green screen keying.

The rest part of this paper is organized as follows. In Section 2, the motivation of the proposed method as well as its flowchart are introduced. Section 3 proposes a small, yet effective CNN. Section 4 presents the algorithm details of the deeply guided linear model. Section 5 introduces the new green screen dataset, while the last two sections give the experiment (Section 6) and conclusions (Section 7), respectively.

2. Overview of the Proposed Method

Without controlled illumination and effective guidance by humans, one firstly needs a highly robust segmentation algorithm to distinguish background and foreground. Motivated by the success of deep learning [14,15], in this work, we also employed deep neural networks for AIR green screen keying. However, as we explain later, the robust CNN model can hardly achieve high robustness and high pixelwise accuracy simultaneously, especially when the time budget is limited.

2.1. A Dilemma Existing in Deep Learning Matting

Although deep learning has achieved great success in the field of computer vision, it still faces some fundamental difficulties. For pixelwise classification/regression problems, it is hard for a single deep network to perform prediction precisely given a limited time budget, e.g., 40 ms per image (the real-time criterion). The dilemma is two-fold: the running time of most deep networks increases quickly as the input image size grows; it is also not easy to obtain pixelwise precise prediction for a high-resolution image from a low-resolution prediction. In addition and more essential, in deep networks, each pixel of a prediction map is rendered from a large neighboring region on the input image. The neighboring region, formally termed the "receptive field" [16–18], plays a significant role in explaining the high robustness of deep learning [19–21]. However, its drawback is also obvious: as the receptive fields of two neighboring pixels are very alike, it is very hard to generate the prediction map with sharp boundaries on which the adjacent map pixels are assigned distinct values. Researchers have been making considerable effort to alleviate

the problem via more complex network topologies [22–26], while introducing even more computational complexity. We demonstrate this dilemma in Figure 1. From the figure, we can see that, although the predicted alpha matte by the deep network is globally robust, it has ambiguous boundaries, which reduces the "user experience" significantly. In contrast, narrow methods (KNN matting [6] and information flow [27]) can generate more precise alpha values in some local regions.

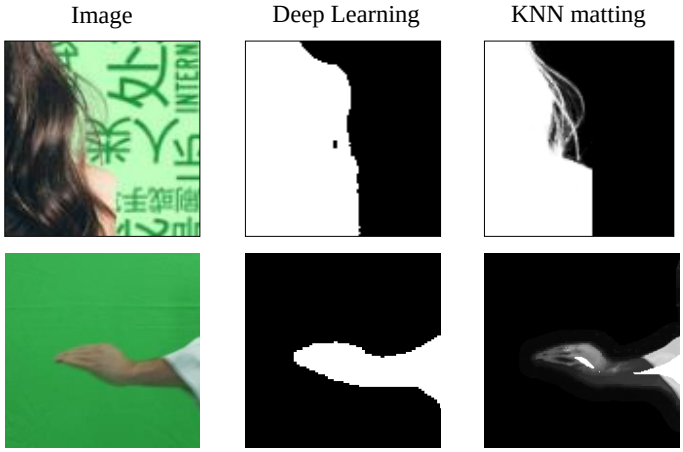

Figure 1. The dilemma of the deep-learning-based matting algorithm. From **left** to **right**: the original image, the prediction by deep learning, and the matting results by KNN matting.

2.2. Our Solution

In this work, we propose to address the above problem via smartly fusing the deep and shallow learning approaches. The flowchart of our algorithm is shown in Figure 2. From the chart, we can see that the high-resolution test image (I_h) is downsampled into one middle-resolution image (I_m) and one low-resolution image (I_l). In the first stage, an offline-trained, light-weight, and symmetrical CNN is applied to I_l to roughly classify each small region into foreground and background. The initial prediction is then upsampled to match the middle-sized I_m as learning guidance for the following shallow model. In the second step, a linear model is trained online based on the raw features (RGB values and texture features in this work) extracted only from this particular image to fine-tune the initial classification result. As we show in Section 4, the loss function employed in this stage can be considered as a Linear Discriminant Analysis (LDA) loss regularized by an affinity term, which usually yields a smoother mask while maintaining the prediction accuracy. The third step is conducted on the high-resolution image (I_h), where we focus on the "uncertain" region $\mathcal{U}$ defined by the previous linear classification. Soft matting values in this region $\alpha_i \in [0, 1], \forall i \in \mathcal{U}$ are determined by a sigmoid function, whose hyperparameter is selected via brute force searching with standard KNN matting loss, as we describe in Section 4.

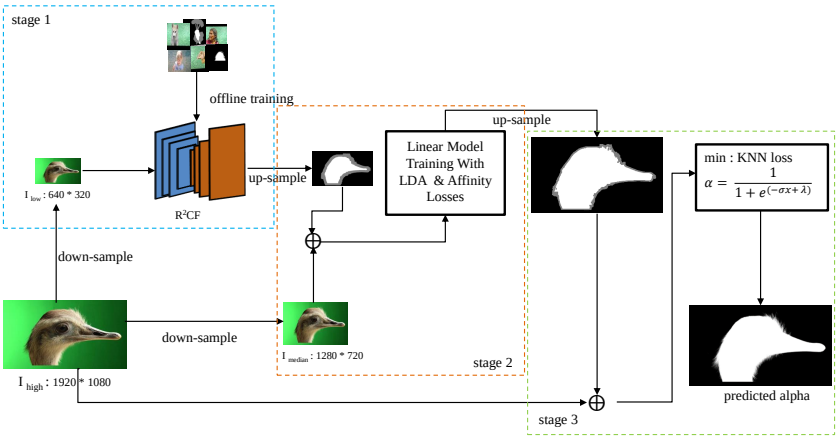

Figure 2. The flowchart of the proposed 3-stage AIR keying algorithm.

3. A Small, yet Effective CNN for Segmentation on Green Screens

In recent years, much effort has been made to handle the natural matting problem, in which the foreground and background are not predefined. Though accused of being ill-posed, deep-learning-based methods [8–10,28] still illustrate high accuracy in this task. Recent approaches have also focused on matting without any external input [29–32] and matting with a known natural background [33,34]. It seems we can easily pick one of the above "off-the-self" matting networks for our green screen matting. However, those networks are relatively large to extract more abstract semantic information, which is important for robust natural matting. On the contrary, in green screen matting, some low-level features are already informative enough, and thus, the above networks are unnecessarily complex and slow in our task.

In [35], Liu et al. proposed a small network for edge detection. Considering the similar motivation of exploiting the multiscale information, we designed our segmentation network based on its RCF model. To achieve an even higher forward speed so that the whole system is real-time, we further shrank the RCF model by reducing the channel numbers, as well as removing some redundant skip connections, as we show in Figure 3. In this work, we term this reduced RCF as R^2CF, whose structure is shown in Figure 3. We can see that the backbone of the R^2CF network is the shrunken version of the VGG-16 network [36,37] with three extra branches and their corresponding intermedia loss layers.

In practice, we trained the R^2CF model based on the training set of the proposed new green screen dataset (described in Section 5). We initialized the network's parameters via the "Xavier" strategy and employed the conventional Stochastic Gradient Descent (SGD) for optimization. The minibatch size was 32, and the base learning rate was 0.003 and dropped by 10 times every 30,000 iterations. The momentum and weight decay were set to 0.9 and 0.00004, respectively. One needs to perform SGD for 100,000 iterations to obtain good performance. The learned deep model performed sufficiently well in practice, though one can still observe some segmentation flaws (see Figure 4), which could be almost totally corrected by the following linear classifier, as we introduce in Section 4. On the other hand, the network was very efficient, with the speed below 10 ms per image on a middle-level GPU.

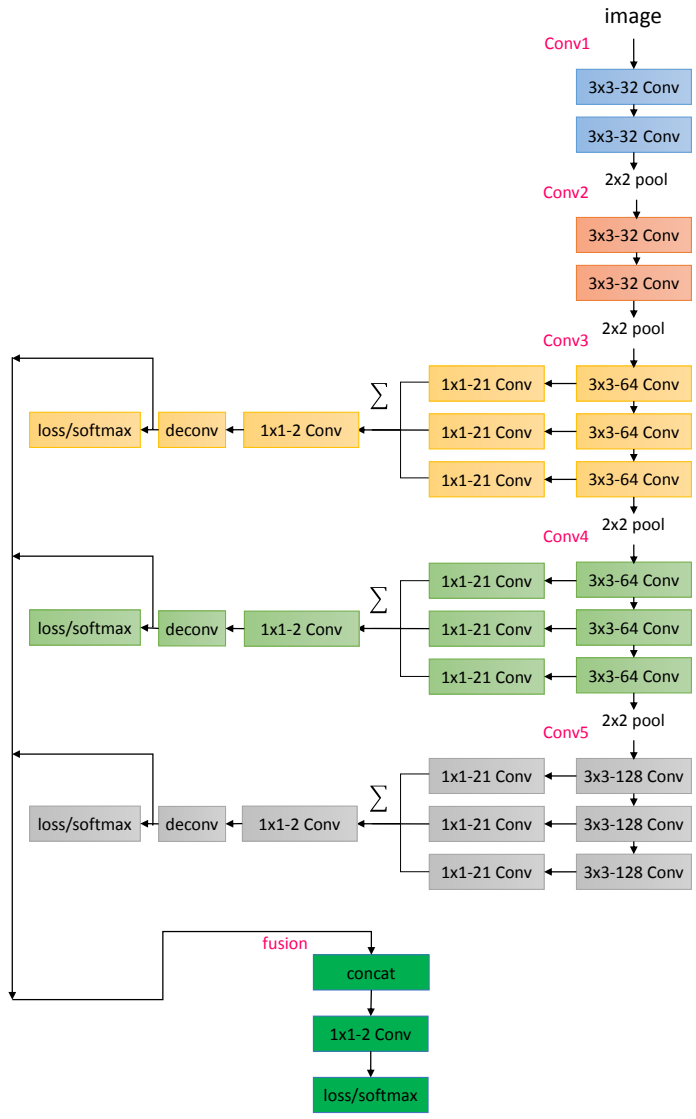

Figure 3. The R^{2}CF network composed of 13 convolution layers and 3 fully connected layers. Similar to its prototype, VGG-16 [36], all the convolutional layers are divided into 5 groups as conv1, ..., conv5. Feature maps from conv3, conv4, and conv5 are integrated together after being filtered by the 1×1 convolutional layers. The three obtained feature maps are then finally summed up elementwise, after another 1×1 convolutional layer. The upsampling process is conducted to guarantee all feature maps have the same size.

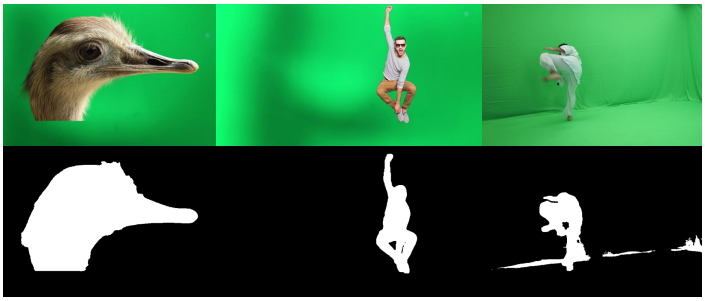

Figure 4. The illustration of the predictions of the proposed R^{2}CF.

4. The Deeply Guided Linear Models for High-Resolution Accurate Matting

4.1. Training Features

As explained in Section 2.1, one cannot expect deep learning to predict pixelwise accurate segmentations or alpha mattes, especially with a limited time budget. Given the output of R^{2}CF, we extracted the two-channel feature map just before the final softmax layer to calculate the "trimap" $\mathrm{T}_l \in \mathcal{R}^{w_l \times h_l}$ as:

$$\mathrm{T}_l^i = \begin{cases} 1 & : \eta_i > 0.75 \\ 0.5 & : 0.5 \leq \eta_i \leq 0.75 \\ 0 & : \eta_i \leq 0.5 \end{cases} \forall i \tag{1}$$

where T_l^i is the i-th pixel on the low-resolution trimap T_l^i and the value η_i is obtained via:

$$\eta_i = \frac{\exp(-\lambda \cdot f_i)}{(\exp(-\lambda \cdot f_i) + \exp(-\lambda \cdot b_i))} \tag{2}$$

where f_i and b_i are the values of the two-channel output of the R^{2}CF network, on the i-th pixel's location. They stand for the confidence of being the foreground and background on this pixel, respectively. Then, the low-resolution trimap is resized to the mid-resolution version: $\mathrm{T}_l \in \mathcal{R}^{w_l \times h_l} \rightarrow \mathrm{T}_m \in \mathcal{R}^{w_m \times h_m}$.

In the second stage, as shown in Figure 2, training samples are collected randomly on both the background region ($\mathrm{T}_m^i <= 0.01$) and the foreground region ($\mathrm{T}_m^i >= 0.99$). In this paper, the feature of each training sample contains two parts: the normalized RGB value and the texture feature extracted on a small adjacent region (3×3 in this work). In mathematical form, the feature $\boldsymbol{f}_i \in \mathcal{R}^{15}$ is written as:

$$\boldsymbol{f}_i = [R_i, G_i, B_i, \boldsymbol{\beta}_i], \forall i, \tag{3}$$

where $\boldsymbol{\beta}_i$ denotes the local texture feature of a pixel, which is defined as:

$$\boldsymbol{\beta}_i = \frac{1}{Z_i} \cdot \mathrm{Hist}_{\Delta\theta}(\boldsymbol{m}_{grad}, \boldsymbol{d}_{grad}) \tag{4}$$

where function $\mathrm{Hist}_{\Delta\theta}(\boldsymbol{m}_{grad}, \boldsymbol{d}_{grad})$ represents the histogram function based on the gradient directions weighted by the corresponding gradient magnitude; Z_i is the normalization parameter, so $1^T\boldsymbol{\beta}_i = 1$. In this work, we set $\Delta\theta = 30$; thus, the dimension of $\boldsymbol{\beta}$ and $\boldsymbol{f}_i$ is 12 and 15, respectively.

4.2. Two Types of Loss Functions

Given the training sample set $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N\}$ with the corresponding labels $\{l_1, l_2, \ldots, l_N\}, l_j \in \{0, 1\}$, which are actually the sampled pixel values on the trimap T_m, we tried to train a linear model such that:

$$\alpha_j = \boldsymbol{\omega}^T \boldsymbol{f}_j + b.$$

To obtain a good estimation of $\boldsymbol{\omega}$ and b, we firstly built the classification loss following Linear Discriminant Analysis (LDA) [38] as:

$$\min_\omega \mathrm{Loss}_c = \boldsymbol{\omega}^T(\mathrm{S}_w^+ + \mathrm{S}_w^-)\boldsymbol{\omega} \tag{5}$$

$$s.t. \frac{1}{N^+}\sum_{i\in\mathcal{P}} \boldsymbol{f}_i > 1 + \frac{1}{N^-}\sum_{j\in\mathcal{N}} \boldsymbol{f}_j \tag{6}$$

where $\mathcal{P}$ and $\mathcal{N}$ stand for the positive and negative subsets of the training samples and $\mathrm{S}_w^{\pm}$ denotes the "within scatter matrix" defined in the LDA algorithm [38].

Recall that the LDA was proposed for general classification, which is different from the matting problem, where the pixels are actually related geometrically. We thus introduced the affinity loss from the family of spectral-based matting [4,6] into the above optimization problem. Specifically, we employed the strategy of KNN matting [6] to build the affinity

matrix L_{rgb} (here, the subscript *rgb* indicates that the kernel values in this affinity matrix are calculated on the RGB values), with the hyperparameter $k = 7$. Given the affinity matrix L, our affinity loss is written as:

$$\mathrm{Loss}_a = \boldsymbol{\alpha}^T \mathrm{L}_{rgb} \boldsymbol{\alpha} = \boldsymbol{\omega}^T \mathrm{F} \cdot \mathrm{L}_{rgb} \cdot \mathrm{F}^T \boldsymbol{\omega} = \boldsymbol{\omega}^T \hat{\mathrm{L}}_{rgb} \boldsymbol{\omega} \tag{7}$$

Now, the combined loss function is defined as:

$$\min_{\omega} \boldsymbol{\omega}^T (\mathrm{S}_w^+ + \mathrm{S}_w^- + \lambda \hat{\mathrm{L}}_{rgb}) \boldsymbol{\omega} \tag{8}$$

$$s.t. \frac{1}{N^+} \sum_{i \in \mathcal{P}} \boldsymbol{f}_i > 1 + \frac{1}{N^-} \sum_{j \in \mathcal{N}} \boldsymbol{f}_j \tag{9}$$

In practice, we set $\lambda = 1000$, and the introduction of the affinity loss leads to smoother alpha output, which can benefit the following matting step.

Note that the generalization and optimization are the most time-consuming parts of the KNN matting algorithm. Each of them usually takes more than 1000 ms on a mid-resolution image. In our case, however, this problem did not exist. The reason is two-fold. First, as we assumed a linear model to represent the pixel's alpha value, one does not need to sample all the pixels on the image, whose number is usually over a million. Actually, in our experiment, we only sampled 1500 positive samples and 1500 negative samples, which were sufficient to offer good results. Secondly, and more importantly, thanks to the linear assumption, the quadratic matrix L_{rgb} collapses into the extremely small one $\hat{\mathrm{L}}_{rgb}$, which was only 15×15 in this work. As a result, the optimization problem of Equation (9) can be easily solved via off-the-shelf quadratic programming solvers, within 5 ms.

4.3. Fine-Tuning the Alpha Values via Brute Force Searching

As shown in Figure 2, in Step 3, we firstly calculate the binary version the output of last step as:

$$\tilde{\alpha}_j = \omega^T \boldsymbol{f}_j + b$$

and then, an "unknown" region on the image is obtained via a simple Gaussian filtering and thresholding process. We fix the binary value of $\tilde{\alpha}$ outside the unknown area and recalculate the inside ones as:

$$\tilde{\alpha}_j = \frac{1}{1 + \exp[-\lambda \cdot (\boldsymbol{\omega}^T \boldsymbol{f}_j + b - \mu)]} \tag{10}$$

The hyperparameters λ and μ are determined via a brute force searching procedure whose loss function is exactly the loss function defined in KNN matting [6]. Note that when performing the brute force searching, it is not necessary to take all the unknown pixels into consideration. In this work, we only randomly sampled 2000 unknown pixels to estimate the best λ and μ. The other 10,000 pixels in the known region were sampled to calculate the affinity matrix of KNN matting. All of Step-3 typically takes only 15 to 20 ms.

5. The New Green Screen Dataset

To the best of our knowledge, the only publicly available green screen dataset was that proposed in [3], which contains four videos captured in controlled environments. To test the algorithm in more challenging scenarios, in this work, we generated a bigger and more comprehensive green screen dataset, called "Green-2018" in this paper. We illustrate the dataset in Figure 5. To obtain the high-quality ground-truth alpha, all the images in the new dataset were synthetically composed from a foreground image (with a precise alpha matte) and a background image. Unlike the existing dataset, which only focuses on human objects, the Green-2018 dataset has various foreground types including animal, human, and furniture. On the other hand, the background images in the new dataset also involve more variance. As we show in Figure 5, there are two main attributes, which are textured (we only focus on the green background here; thus, the textured background is also generated by using a number (two in our case) of different green colors) or pure

green screen and natural or controlled lighting condition, respectively. We rendered our dataset through randomly locating the foreground objects with random scales. To make the synthetic images closer to the real ones, shadows were also rendered on some of the background images.

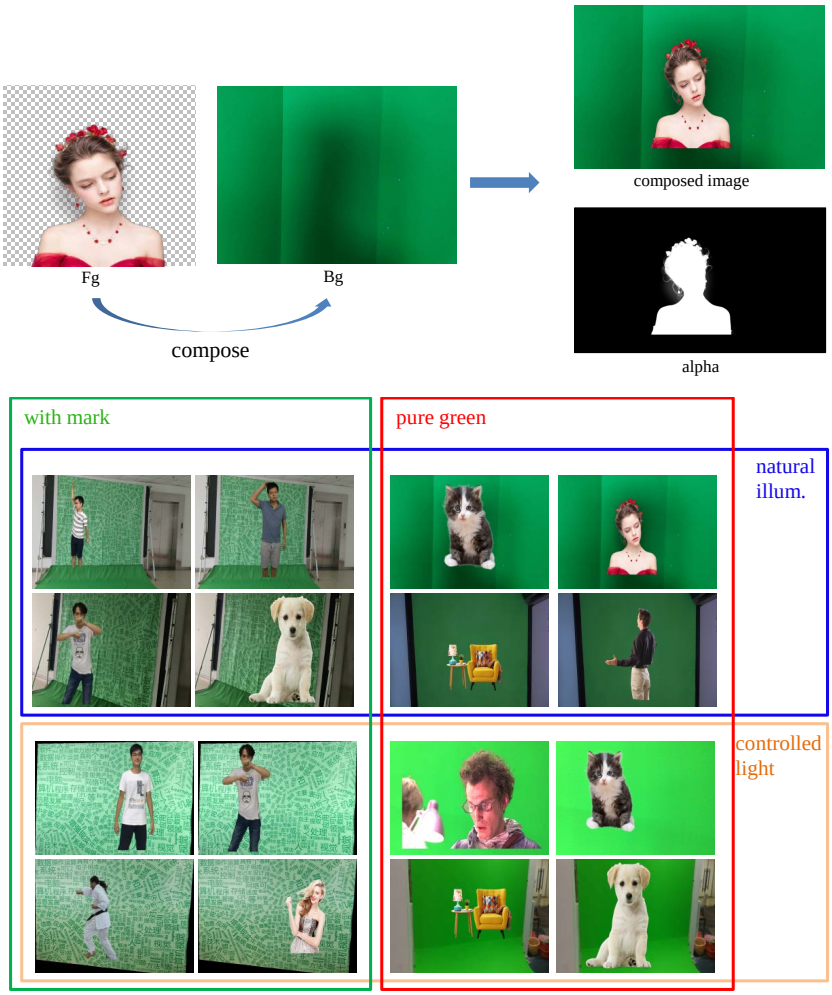

Figure 5. The illustration of the proposed new green screen dataset (portraits were permitted in October 2018). **Top**: the illustration of foreground and background fusion; **bottom**: the image samples of Green-2018 with 2 major attributes, which are textured or pure green screen and natural or controlled lighting condition, respectively. Note that the outside region of the green screen is treated as foreground in the dataset.

The whole dataset contains 657 foreground images and 2693 background images. We divided them into two subsets for training and testing, respectively. Our training subset contains 20,370 merged images, which were generated from 485 foreground and 2010 background images, while the test subset includes the last 172 foreground images and 683 background images, and 3096 composed test images were rendered.

6. Experiments and Results

In this section, we compare the proposed method with different types of approaches, which can solve the green screen matting problem. Three state-of-the-art shallow matting algorithms were compared: closed-form matting [4], KNN matting [6], and the most recently proposed information flow matting [27]. Two typical deep-learning-based matting methods, i.e., deep image matting [8] and IndexNet Matting [39], were also performed in the comparison.

Meanwhile, we also illustrate the comparison between our automatic method and the off-the-shelf manual keying software, i.e., After Effect (AE) from Adobe. Following the conventional setting in the matting literature [6,27,40], we report the performance via four evaluation metrics, which are SAD, MSE, Connectivity, and Gradient, respectively.

As mentioned in Section 5, we evaluated all the involved methods on two datasets, i.e.,

- The original dataset introduced in [3]. This is a pure green screen dataset including only four videos. We called this dataset TOG-16;
- Our Green-2018 dataset, which contains textured and pure green screen, as well as more foreground categories.

Note that there is no matting α ground-truth offered in the TOG-16 dataset; we manually labeled 100 images of this dataset and evaluated the matting performance on the shrunken version of TOG-16. The experiments was conducted on a PC with an Intel i5-8600 CPU, 32G memory, and a NVIDIA GTX-1080Ti GPU.

6.1. The Running Speed

In a practical matting system, one usually requires a real-time running speed. Consequently, we firstly compare the running speed of all the involved methods in Table 1.

Table 1. The comparison on running time (in ms) of different keying methods.

Methods	Running Time (ms/img)
closed-form [4]	3950
KNN matting [6]	20,000
information flow [27]	15,000
deep matting [8]	312
IndexNet matting [39]	6613
AE-Keylight	30,000
this work	42

From the speed comparison, we can see that only our method can be considered as real-time, the second fastest matting algorithm being deep matting [8], which only ran at around 3 fps. Note that, except the proposed method, the running time of all the other method was not taken into account in the generation time of "trimap". Our method illustrates the obvious superiority in efficiency.

6.2. The Matting Accuracy

6.2.1. The Comparison to Other Matting Algorithms

As introduced above, the proposed method is "end-to-end". However, that is not true for all the other compared methods: they all require "trimaps" for matting. For a fair comparison, the required "trimaps" were obtained by using our R^{2}CF model. The test results are shown in Tables 2 and 3. As we can see, for both the simple and complicated scenarios, our method showed comparable performance to the deep-learning-based methods and showed obvious superiority over the shallow approaches.

Table 2. Matting performance on TOG-16.

Methods	SAD ($\times 10^4$)	MSE ($\times 10^{-3}$)	Connectivity ($\times 10^{-3}$)	Gradient ($\times 10^{-3}$)
closed-form [4]	2.59	9.55	4.54	3.12
KNN matting [6]	2.24	7.43	3.87	3.86
information flow [27]	2.22	8.11	3.90	4.01
deep matting [8]	1.72	1.60	1.10	3.25
IndexNet matting [39]	2.46	3.30	1.54	4.49
this work	1.27	3.64	3.29	2.00

More comparison results are shown in Figure 6. From the images, one can say that the proposed method performed well in most scenarios and showed high robustness, as can be seen in Tables 2 and 3.

Figure 6. The green screen keying result on different video sequences. From **left** to **right**: the input image; the ground-truth alpha map; the matting result of closed-form matting; the result of information flow matting; the result of KNN matting; the result of the proposed method. Each row shows the test results on one image, with different automatic matting algorithms.

Table 3. Matting performance on Green-2018.

Methods	SAD ($\times 10^4$)	MSE ($\times 10^{-2}$)	Connectivity ($\times 10^{-3}$)	Gradient ($\times 10^{-3}$)
closed-form [4]	15.7	6.94	56.1	6.31
KNN matting [6]	10.9	4.66	39.6	6.61
information flow [27]	13.5	5.93	53.1	9.12
deep matting [8]	1.36	0.18	6.0	2.20
IndexNet matting [39]	0.87	0.15	3.0	2.12
this work	2.83	1.63	7.75	3.59

6.2.2. The Comparison with Manual Keying Software

Besides the automatic matting algorithms proposed in the literature, manual matting software dominates the current market. The software is mostly designed based on a single key color (green or blue) background. We also evaluated our method by comparing to the manual method on two randomly picked videos from TOG-16. The quantitative results are shown in Table 4 from which one can see the accuracy of our method compared to the manual commercial software. Note that the software was operated by an amateur user with one week of AE experience. When testing, the operator only performed manual keying on the first frame and used the samekeying parameter for all the following frames of the sequence.

Table 4. The keying performance comparison between our method and the manual software. Note that the software was handled by an amateur user with one week of AE experience.

Methods	SAD ($\times 10^4$)	MSE ($\times 10^{-3}$)	Connectivity ($\times 10^{-3}$)	Gradient ($\times 10^{-3}$)
AE-Keylight	14.6	52.91	34.25	14.73
this work	12.8	36.94	20.1	10.05

6.2.3. The Matting Robustness

From the comparison results shown in Section 6.2.1, one could say that the proposed method enjoys a fast running speed while usually performing worse than the deep-learning-based method, which also demonstrated state-of-the-art matting performance on some well-known matting datasets [8,39].

However, the situation changed dramatically when the same experiment was conducted on some real-life images, rather than the "synthetic" images employed in the Green-2018 dataset. We captured eight video sequences with a real human shown in front of the same background setting as in Green-2018 (see Figure 7). As can be seen, the "trimap" obtained using the R^{2}CF model became imperfect and sometimes even incorrect. In this scenario, the deep-learning-based methods deteriorated rapidly, and the proposed method still maintained a relatively high matting accuracy. Our method illustrated much higher matting robustness against the "state-of-the-art" matting approaches.

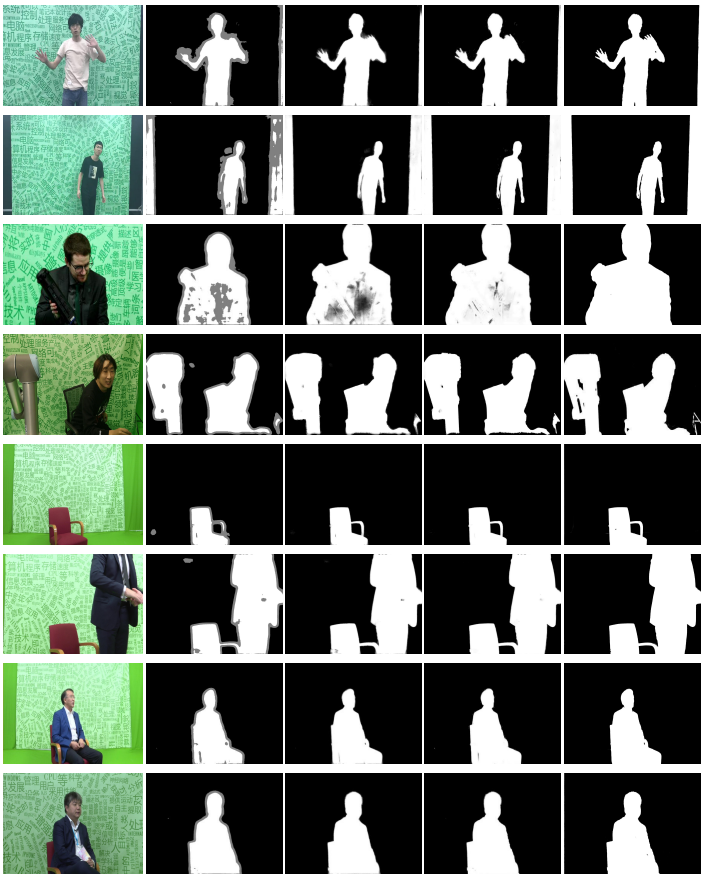

Figure 7. The real-life matting performance (portraits were permitted in October 2018). From **left** to **right**: the input image; the imperfect "trimap" obtained by using the R^{2}CF model; the matting result of deep image matting [8]; the result of IndexNet matting [39]; and the result of this work. One can see that as the "trimap" becomes incorrect, the deep-learning-based methods are influenced dramatically, while the proposed method performs much more stably.

7. Conclusions

In this paper, we proposed a novel way to achieve automatic illumination-invariant and real-time keying on green screens. Linear models and deep learning results were smartly combined to generate robust matting results, with a nearly real-time (around 42 ms per image) speed. Besides, a new green screen dataset, which contained more foreground variances and more challenging backgrounds, was built. To the best of our knowledge, this is the first algorithm that can perform AIR keying, and the proposed dataset is also the first in-the-wild green screen dataset. The superiority in the efficiency, accuracy, and robustness of the proposed method was also proven in our experiment. In the future, our work will focus on improving the quality of the coarse output of the offline-trained CNN, which is very important to us. In addition, we will apply our proposed approach to a higher image resolution and more complex scenes to verify its effectiveness.

Author Contributions: Supervision: H.L. and Y.M.; validation: W.Z. and H.J.; writing—original draft, H.L.; writing—review and editing: H.L. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 61962027.

Data Availability Statement: Our dataset (Green-2018) is available at https://github.com/wenyuzhu28/AIRMatting (accessed on 9 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Porter, T.; Duff, T. Compositing digital images. In *ACM Siggraph Computer Graphics*; ACM: New York, NY, USA, 1984; Volume 18, pp. 253–259.
2. Grundhöfer, A.; Bimber, O. VirtualStudio2Go: Digital video composition for real environments. *ACM Trans. Graph. (TOG)* **2008**, *27*, 151. [CrossRef]
3. Aksoy, Y.; Aydin, T.O.; Pollefeys, M.; Smolić, A. Interactive High-Quality Green-Screen Keying via Color Unmixing. *ACM Trans. Graph.* **2016**, *35*, 1–12. [CrossRef]
4. Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 228–242. [CrossRef] [PubMed]
5. Tang, X.; Rother, C.; Rhemann, C.; He, K.; Sun, J. A global sampling method for alpha matting. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; IEEE Computer Society: Los Alamitos, CA, USA, 2011; pp. 2049–2056. [CrossRef]
6. Tang, C.K.; Li, D.; Chen, Q. KNN matting. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; IEEE Computer Society: Los Alamitos, CA, USA, 2012; pp. 869–876. [CrossRef]
7. Liu, J.; Yao, Y.; Hou, W.; Cui, M.; Xie, X.; Zhang, C.; Hua, X.S. Boosting Semantic Human Matting With Coarse Annotations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8560–8569. [CrossRef]
8. Xu, N.; Price, B.; Cohen, S.; Huang, T. Deep Image Matting. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 311–320. [CrossRef]
9. Wang, Y.; Niu, Y.; Duan, P.; Lin, J.; Zheng, Y. Deep Propagation Based Image Matting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18), Stockholm, Sweden, 13 July 2018; pp. 999–1006.
10. Lutz, S.; Amplianitis, K.; Smolic, A. AlphaGAN: Generative adversarial networks for natural image matting. *arXiv* **2018**, arXiv:1807.10088.
11. Tang, J.; Aksoy, Y.; Oztireli, C.; Gross, M.; Aydin, T.O. Learning-Based Sampling for Natural Image Matting. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3050–3058. [CrossRef]
12. Sun, S.; Cao, Z.; Zhu, H.; Zhao, J. A Survey of Optimization Methods From a Machine Learning Perspective. *IEEE Trans. Cybern.* **2020**, *50*, 3668–3681. [CrossRef] [PubMed]
13. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
14. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]

15. Wang, S.; Wang, O.; Zhang, R.; Owens, A.; Efros, A.A. CNN-Generated Images Are Surprisingly Easy to Spot... for Now. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 8692–8701. [CrossRef]
16. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 29, Proceedings of the 30th Annual Conference on Neural Information Processing Systems 2016, Barcelona, Spain, 5–10 December 2016*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 4898–4906.
17. Shang, T.; Dai, Q.; Zhu, S.; Yang, T.; Guo, Y. Perceptual Extreme Super Resolution Network with Receptive Field Block. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 1778–1787. [CrossRef]
18. Kim, W.; Nguyen, A.D.; Lee, S.; Bovik, A.C. Dynamic Receptive Field Generation for Full-Reference Image Quality Assessment. *IEEE Trans. Image Process.* **2020**, *29*, 4219–4231. [CrossRef] [PubMed]
19. Liu, S.; Huang, D.; Wang, A. Receptive Field Block Net for Accurate and Fast Object Detection. In Proceedings of The European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 December 2018.
20. Lin, T.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Los Alamitos, CA, USA, 2017; pp. 936–944. [CrossRef]
21. Li, Y.; Chi, L.; Tian, G.; Mu, Y.; Ge, S.; Qiao, Z.; Wu, X.; Fan, W. Spectrally-Enforced Global Receptive Field For Contextual Medical Image Segmentation and Classification. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 1–6. [CrossRef]
22. Huang, Z.; Wang, X.; Wei, Y.; Huang, L.; Shi, H.; Liu, W.; Huang, T.S. CCNet: Criss-Cross Attention for Semantic Segmentation. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Seoul, Korea, 27 October–2 November 2020. [CrossRef]
23. Yuan, Y.; Chen, X.; Wang, J. Object-Contextual Representations for Semantic Segmentation. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 173–190.
24. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. DenseASPP for Semantic Segmentation in Street Scenes. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; IEEE Computer Society: Los Alamitos, CA, USA, 2018; pp. 3684–3692. [CrossRef]
25. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9196–9205. [CrossRef]
26. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-Maximization Attention Networks for Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; IEEE Computer Society: Los Alamitos, CA, USA, 2019; pp. 9166–9175. [CrossRef]
27. Aksoy, Y.; Ozan Aydin, T.; Pollefeys, M. Designing Effective Inter-Pixel Information Flow for Natural Image Matting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
28. Yu, Q.; Zhang, J.; Zhang, H.; Wang, Y.; Lin, Z.; Xu, N.; Bai, Y.; Yuille, A. Mask Guided Matting via Progressive Refinement Network. *arXiv* **2020**, arXiv:2012.06722.
29. Shen, X.; Tao, X.; Gao, H.; Zhou, C.; Jia, J. Deep Automatic Portrait Matting. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 92–107.
30. Zhu, B.; Chen, Y.; Wang, J.; Liu, S.; Zhang, B.; Tang, M. Fast Deep Matting for Portrait Animation on Mobile Phone. In Proceedings of the 25th ACM International Conference on Multimedia (MM '17), Mountain View, CA, USA, 23–27 October 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 297–305. [CrossRef]
31. Zhang, Y.; Gong, L.; Fan, L.; Ren, P.; Huang, Q.; Bao, H.; Xu, W. A Late Fusion CNN for Digital Matting. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE Computer Society: Los Alamitos, CA, USA, 2019; pp. 7461–7470. [CrossRef]
32. Qiao, Y.; Liu, Y.; Yang, X.; Zhou, D.; Xu, M.; Zhang, Q.; Wei, X. Attention-Guided Hierarchical Structure Aggregation for Image Matting. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE Computer Society: Los Alamitos, CA, USA, 2020; pp. 13673–13682. [CrossRef]
33. Lin, S.; Ryabtsev, A.; Sengupta, S.; Curless, B.; Seitz, S.; Kemelmacher-Shlizerman, I. Real-Time High-Resolution Background Matting. *arXiv* **2020**, arXiv:2012.07810.
34. Sengupta, S.; Jayaram, V.; Curless, B.; Seitz, S.; Kemelmacher-Shlizerman, I. Background Matting: The World is Your Green Screen. *arXiv* **2020**, arXiv:2004.00626.
35. Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. Richer convolutional features for edge detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5872–5881. [CrossRef]
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
37. Kim, J.; Lee, J.; Lee, K. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Los Alamitos, CA, USA, 2016; pp. 1646–1654. [CrossRef]
38. Riffenburgh, R.H. Linear Discriminant Analysis. Ph.D. Thesis, Virginia Polytechnic Institute, Blacksburg, VA, USA, 1957.

39. Lu, H.; Dai, Y.; Shen, C.; Xu, S. Indices Matter: Learning to Index for Deep Image Matting. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; IEEE Computer Society: Los Alamitos, CA, USA, 2019; pp. 3265–3274. [CrossRef]
40. Sun, J.; Jia, J.; Tang, C.K.; Shum, H.Y. Poisson Matting. In *ACM SIGGRAPH 2004 Papers*; Association for Computing Machinery: Los Angeles, CA, USA, 2004; pp. 315–321. [CrossRef]