Smart Scribbles for Image Matting

XIN YANG*, Dalian University of Technology and Beijing Technology and Business University, China
YU QIAO* and SHAOZHE CHEN, Dalian University of Technology, China
SHENGFENG HE, South China University of Technology, China
BAOCAI YIN, Dalian University of Technology and Peng Cheng Laboratory, China
QIANG ZHANG and XIAOPENG WEI, Dalian University of Technology, China
RYNSON W. H. LAU, City University of Hong Kong, China

Image matting is an ill-posed problem that usually requires additional user input, such as trimaps or scribbles. Drawing a fine trimap requires a large amount of user effort, while using scribbles can hardly obtain satisfactory alpha mattes for non-professional users. Some recent deep learning–based matting networks rely on large-scale composite datasets for training to improve performance, resulting in the occasional appearance of obvious artifacts when processing natural images. In this article, we explore the intrinsic relationship between user input and alpha mattes and strike a balance between user effort and the quality of alpha mattes. In particular, we propose an interactive framework, referred to as smart scribbles, to guide users to draw few scribbles on the input images to produce high-quality alpha mattes. It first infers the most informative regions of an image for drawing scribbles to indicate different categories (foreground, background, or unknown) and then spreads these scribbles (i.e., the category labels) to the rest of the image via our well-designed two-phase propagation. Both neighboring low-level affinities and high-level semantic features are considered during the propagation process. Our method can be optimized without large-scale matting datasets and exhibits more universality in real situations. Extensive experiments demonstrate that smart scribbles can produce more accurate alpha mattes with reduced additional input, compared to the state-of-the-art matting methods.

CCS Concepts: • **Computing methodologies** → **Image segmentation**; *Artificial intelligence*; *Computer vision*; *Computer vision problems*;

Additional Key Words and Phrases: Image matting, alpha matte, markov chain, deep learning, label propagation

*Both authors contributed equally to this research.

This work was supported in part by the National Natural Science Foundation of China under Grant 91748104, Grant 61972067, Grant 61632006, Grant U1811463, Grant U1908214, Grant 61751203, in part by the National Key Research and Development Program of China under Grant 2018AAA0102003, Grant 2018YFC0910506, in part by the Open Research Fund of Beijing Key Laboratory of Big Data Technology for Food Safety (Project No. BTBD-2018KF).

Authors' addresses: X. Yang, Y. Qiao, S. Chen, Q. Zhang, and X. Wei (corresponding author), Dalian University of Technology, 2 linggong road, Dalian, Liaoning, China 116024; emails: {xinyang, zhangq, weixp}@dlut.edu.cn, {qiaoyu2017, csz}@mail.dlut.edu.cn; S. He (corresponding author), South China University of Technology, Guangzhou University Town, Guangzhou, Guangdong, China 510006; email: hesfe@scut.edu.cn; B. Yin, Dalian University of Technology, 2 linggong road, Dalian, Liaoning, China 116024 and Peng Cheng Laboratory, 2 Xingke First Street, Shenzhen, Guangdong, China 518055; email: ybc@dlut.edu.cn; R. W. H. Lau, City University of Hong Kong, 83 Tat Chee Road, Kowloon Tong, Hong Kong, China 999077; email: rynson.lau@cityu.edu.hk.

1551-6857/2020/12-ART121 $15.00
https://doi.org/10.1145/3408323

ACM Reference format:
Xin Yang, Yu Qiao, Shaozhe Chen, Shengfeng He, Baocai Yin, Qiang Zhang, Xiaopeng Wei, and Rynson W. H. Lau. 2020. Smart Scribbles for Image Matting. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 4, Article 121 (December 2020), 21 pages.
https://doi.org/10.1145/3408323

1 INTRODUCTION

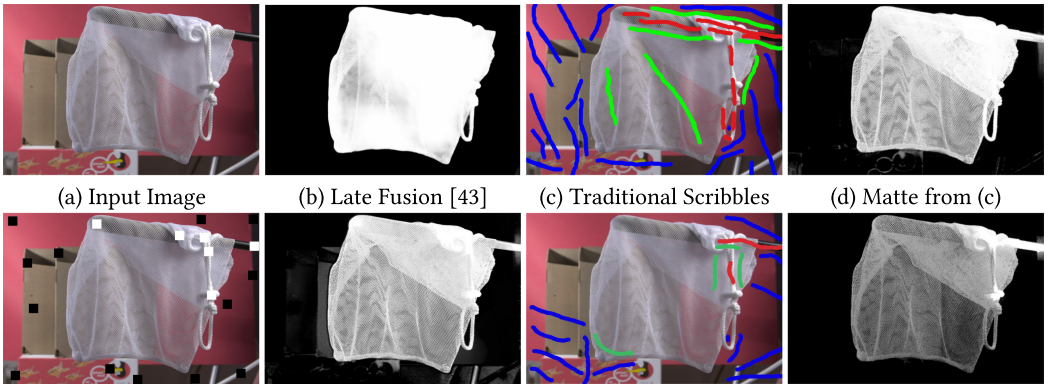
There are different object layers in the image, and we always define the layers of interest as the foreground and the rest as the background. Separating the foreground and background from an image precisely, defined as image matting, is a long-standing problem in both academia and industry. Different from common image segmentation [3, 29, 36], image matting is required to precisely pull out the foreground object, specific to sophisticated internal texture details and clear boundary contour. It has high value in applications such as movie making and image editing and is modeled by solving the following under-constrained equation:

$$I_p = \alpha_p F_p + (1 - \alpha_p) B_p, \tag{1}$$

where I represents the observed image, p refers to a pixel location, and F and B refer to the foreground and background layers, respectively. α represents the alpha matte, where α_p varies in the range of [0,1] to indicate the foreground proportion. Equation (1) is a highly ill-posed problem, as the only known variable is the input I. Therefore, additional information from users is essential to solve this problem, which can confine the scope of foreground and background. There are mainly two types of additional inputs: trimaps [10, 16, 25, 28, 35, 38, 41] and scribbles [14, 18, 19, 34].

A well-defined trimap is a densely-annotated auxiliary input, and each pixel in it has one of the following three category labels: foreground, background, or unknown. The trimap can effectively constrain the input image according to Equation (1): foreground and background indicate that $\alpha_p = 1$ and $\alpha_p = 0$, respectively, and unknown represents the critical areas between foreground and background ($\alpha_p \in (0, 1)$). As densely annotated additional input, trimaps can provide rich information for matting problem, and hence most state-of-the-art matting methods [2, 38] typically use input images and trimaps as input, and they can produce comparable alpha mattes. Nevertheless, generating a perfect trimap means dense annotations for each pixel, which is tedious and time-consuming for common users. The complexity of trimap generation limits their versatility in real-world scenarios. Compared to trimaps, scribbles are more user-friendly additional input. In traditional scribbles-based methods, users are nominally free to draw a few scribbles on the input image to suggest the foreground, background, and unknown. The certain regions with scribbles can provide reference for the others in matting algorithms. Actually, the quality of resulting alpha mattes depends heavily on the number of scribbles and where they are drawn. In addition, drawing useful scribbles requires matting knowledge and experience to meet some prior assumptions of the matting algorithms, making it little impractical to non-professional participants.

In addition, although recent deep learning–based matting methods [4, 15, 22, 33, 43] can achieve impressive alpha mattes, they sometimes fail when handling real-world images, because their prediction models are usually trained on the composite datasets. The composition rules in Reference [38] can sometimes make the foreground and background disharmonious, resulting in obvious artifacts on training images. Factors such as illumination and shadow in natural images can amplify the influence of artifacts, and discount the final results. Our motivation is to automatically select relevant regions that meet the prior requirements, and all the user has to do is draw scribbles on these suggested regions. We refer to these regions that determine the quality of alpha mattes as *informative regions*. The *informative regions* can acquire accurate labels through limited user



(a) Input Image (b) Late Fusion [43] (c) Traditional Scribbles (d) Matte from (c)

(e) Active Matting [40] (f) Matte from (e) (g) Smart Scribbles (h) Matte from (g)

Fig. 1. The comparison of *smart scribbles* with Late Fusion [43], traditional scribbles, and Active Matting [40]. The proposed method is able to produce more accurate alpha mattes with refined texture details and clear foreground contour, requiring relatively little additional input (both scribbles and interactive boxes are appropriately enlarged for better distinction).

interaction, which ensures that we have correct labels as a reference when processing different kinds of input images. Besides, we observe that overmuch scribbles are unnecessary for matte generation, because we can infer their category labels from existing scribbles in the *informative regions*.

The main challenge of our motivation is to identify informative regions and effectively reduce user interactions. To strike a balance between the user effort and matting accuracy, we propose a unified framework, referred to as *smart scribbles*, to guide users to draw scribbles on the suggested regions and then propagate category labels to the remaining regions. *Smart scribbles* can achieve alpha mattes with relatively little user effort: They only need to draw fewer scribbles on *informative regions* to separate the foreground, background, and unknown. Specifically, our framework first automatically select *informative regions* based on the similarity, diversity, label entropy, and edge maps [45] of different regions, and these four terms are summarized as *information content*. Users are then asked to draw scribbles on these *informative regions* to label different categories (i.e., foreground, background, and unknown). After that, these labels are propagated to adjacent regions via Markov propagation. The *informative regions* selection, drawing scribbles, and Markov propagation are iterated several times to provide essential category labels, and then we employ a deep network to capture semantic relevance in the image and adopt high-level semantics to further spread scribbles. In contrast to Markov propagation that is limited to local image features, Convolutional Neural Network (CNN) propagation can refine the alpha mattes in a global manner. Extensive experiments show that our *smart scribbles* can generate high-quality alpha mattes with little user effort. In addition, the proposed two-phase propagation can spread scribbles to the whole image to improve alpha mattes, outperforming the state-of-the-art methods. Figure 1 compares smart scribbles with Late Fusion [43], traditional scribbles, and Active Matting [40], and our method illustrates more sophisticated texture details and clear boundary contour.

The main contributions of this article are as follows:

- We propose a novel interactive framework (*smart scribbles*) to generate alpha mattes from limited scribbles, which is more flexible and robust for natural image matting and can dramatically reduce user effort.
- We design an efficient method for computing *information content*. It can identify the most *informative regions* that are significant for the quality of alpha mattes.

- We present a two-phase propagation approach that can aggregate local and global information to spread limited scribbles to the whole image to generate an alpha matte effectively.
- Our method can achieve high-quality alpha mattes independently of large-scale matting datasets, and demonstrate more adaptability in real-world applications.

2 RELATED WORK

In this section, we briefly review image matting from scribbles-based and trimap-based methods, and then introduce some deep learning–based methods that have developed rapidly in recent years.

Scribbles-based Matting. In earlier works [14, 18, 19, 34], scribbles are widely used for natural image matting. Users only need to draw several scribbles on the input image with distinguished color to label the foreground, background, and unknown. Scribbles-based methods are very convenient for common users and can achieve comparable alpha mattes in a user-friendly way. However, such scribbles only cover a small portion of the image, which means the alpha mattes will discount for slightly more complicated images. Besides, scribbles must fit the initial assumptions or prior distribution of matting algorithms, hence users are required to possess professional knowledge about the matting algorithms and rich experience in where to place their scribbles. For example, some methods [18, 19] assume that all colors within a small window around an unknown pixel lie in a line of the color space. This color-line assumption cannot provide global information, and drawing scribbles that fit these algorithms would be a great challenge for novice users.

Trimap-based Matting. Compared with scribbles, trimaps can supply annotations for each pixel in the image, thus can provide sufficient category information (foreground, background, or unknown) for image matting. Due to this advantages, currently, most state-of-the-art image matting algorithms take trimaps as assistant input [2, 8, 10, 38]. Their core ideas are spreading certain labels (the foreground and background) to unknown areas. According to the different way of utilizing foreground-background information, these algorithms can be divided into two categories: sampling-based and propagation-based. Sampling-based methods [10, 16, 25, 28, 35, 41] assume that each unknown pixel can be represented by a pair of certain foreground/background pixels. Propagation-based methods [2, 7, 13, 17–19, 31] use affinity of neighboring pixels to propagate the alpha values from certain areas to unknown ones.

Compared with scribbles, trimap-based methods do not need experience or professional skills but consume a lot of user labors. A well-defined trimap requires pixel-level interpretation provided by users, which is very time-consuming and fussy. Therefore, traditional matting methods usually utilize the online benchmark [26] dataset to evaluate their algorithms, which only contains 27 training examples and eight test images. Although trimap-based methods can achieve competent alpha mattes, they have poor performance in practical application, because delicate trimaps are inconvenient to label for common users.

Deep Learning–based Matting. Deep learning has contribute a lot to computer vision [37, 39, 42] and also has been applied in the alpha matting problem. Cho et al. [8] fused the results of KNN [7] and ClosedForm [18] with input images and then fed them into a well-designed CNN to generate alpha mattes. Xu et al. [38] concatenated the input image and trimap as four-channel input and applied an encoder-decoder network to reconstruct the alpha mattes from high-level semantic representation. The subsequent matting methods [4, 15, 22, 23, 33] mostly follow this design philosophy: extracting high-level features from the input image and the corresponding trimap and then predicting the alpha matte through complicated network architecture. However, they all require trimaps as auxiliary input, which limits their promotion in practical applications.

Some matting networks employ semantic segmentation network [6, 43] or attention mechanism [24] to accomplish alpha mattes without trimaps, and failure case will probably occur when semantic segmentation is not applicable [43]. Xin et al. [40] proposed an active framework to

guide matte generation. Although the above methods can produce alpha mattes without trimaps, they all are trained on the composite images, which can sometimes result in obvious artifacts or poor performance on real-world images. Our *smart scribbles* accept user scribbles and then execute Markov and CNN propagation in superpixels to spread category labels. The proposed model is independent of large-scale composite data and thus exhibits greater robustness and generalization on real-world images.

3 METHODOLOGY

According to our previous analysis, a limited number of scribbles must meet some prior conditions to achieve comparable alpha mattes, while trimaps can help produce accurate matting results at the expense of much time and user labor in practical applications. Our goal is to achieve a balance between user effort and the quality of alpha mattes. For this purpose, we first use automatic region selection to replace the professional knowledge requirements for drawing scribbles, then we employ well-designed two-phase propagation to maximize the diffusion of limited category scribbles across the whole image, which can effectively reduce the number of scribbles drawn by users. Based on the observation that different image regions contribute unequally for matting algorithms and scribbles on critical ones can determine a desired alpha matte, we define *informative regions* to represent the whole image and users only need to draw a few random scribbles on them. Since the informative regions receive accurate labels (foreground, background, or unknown) from users, they can actively contribute to the rest part of the input image (i.e., effectively propagating user labels to uncertain regions).

3.1 Overview

To interpret our *smart scribbles*, we divide our framework shown in Figure 2 into four main components:

Over Segmentation: Our framework is performed on superpixels level for a faster and more efficient generation. We intend to spread limited scribbles across all image areas, while pixel-level nodes do not reflect color and texture correlations between regions. In addition, compared to pixels, propagation on superpixels level can effectively reduce the number of nodes to decrease the total amount of computation. The input image is segmented into superpixels via simple linear iterative clustering (SLIC) algorithm [1] (as shown in Figure 2). The number of superpixels can automatically adapt to the input size, thus we can obtain decent segmentation results for arbitrary-size images. The following steps are all based on superpixels, irrelevant to the attributes of image itself (size or classes, etc.).

Regions Division and Selection: After over-segmentation, as shown in Figure 2, informative regions are chosen for expressing the whole image. We evenly divide the input image into $M \times M$ regular regions, then select the regions crucial for alpha mattes automatically, which can replace the professional knowledge requirements of users. After the regions division, we define the *information content* of each region according to the superpixels inside it, which is explained in Section 3.2. For each iteration, the most critical region (the one with largest *information content*) is selected and users only need to simply draw few scribbles on it. In practice, M can be adjusted according to the size of the input image and is empirically set as 4 considering the convenience of users in most cases.

Drawing Scribbles: Users are required to draw scribbles on the selected region with distinguished color to label different areas (Foreground, red; Background, blue; and Unknown, green), as shown in Figure 2.

Information Propagation: *Smart scribbles* will record category labels and propagate these information to unlabeled regions. To achieve this, we construct the probability matrix (PrM) to represent the transition possibility between different superpixels, and the label probabilities of different

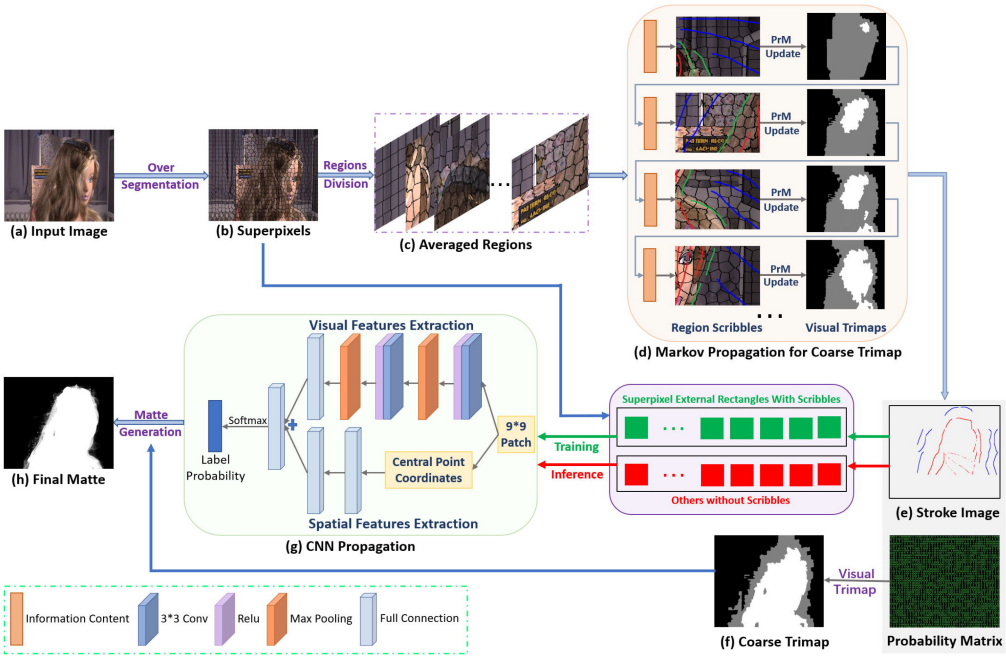


Fig. 2. The pipeline of the proposed method. The input image is first over-segmented into superpixels and then is divided into regular rectangle regions of the same size. We calculate the information content of each region and the most informative region is automatically selected for users to draw scribbles, specifying the foreground (in red), background (in blue), and unknown areas (in green). These labels are then propagated to unlabeled regions to update the probability matrix (PrM) via two-phase propagation. During CNN propagation, we gather all superpixel external rectangles as input, and the superpixels with scribbles are used for training, while for the others we predict category labels for them using the trained model. After CNN propagation, we update PrM to generate a refined trimap and the final matte can be produced by an embedded existing matting algorithm.

categories (*pb*-background, *pu*-unknown, and *pf*-foreground) of each superpixel can be calculated according to the probability matrix. The proposed two-phase propagation is essentially a continuous update of the probability matrix, including two stages: (1) The informative labels propagate to their adjacent regions using Markov chain [11] by taking spatial and appearance similarity into account (as illustrated in Figure 2), dubbed Markov propagation. (2) The output is further refined using a convolutional neural network by exploiting high-level features of superpixels in a global manner, dubbed CNN propagation. The detailed propagation process is developed in Section 3.3.

Overall, we iterate three operations, regions selection, drawing scribbles, and Markov propagation for N times, to accomplish the full spread of scribbles in the neighborhood, and CNN propagation is then applied to generate the fine trimaps. Final alpha mattes are produced with embedded existing matting algorithms. In our experiments, parameter N is set as 6 to balance the tradeoff between efficiency and quality. More iterations can improve mattes slightly with user effort increasing apparently, and the detailed analysis about iteration number can refer to baseline 2 in Section 4.4.

3.2 Information Content Formulation

Scribbles often need to meet some prior assumptions, hence we propose *informative regions* to replace such professional requirements of users. We hold that *informative regions* should contain

representative image semantics or decisive local features, and placing limited scribbles on these regions can achieve a competent alpha matte. In contrast, image areas with single color or smooth texture are insignificant for drawing scribbles (we can assign their labels through later propagation). It is impractical for a novice user to distinguish where are *informative regions*, and the definition of *information content* can select the most informative region iteratively guiding users to draw scribbles.

Specifically, we conclude that the informative ones should: (1) have a high similarity with the neighborhood, which means representative; (2) have a high color and texture diversity inside, which makes the region contain as much scenarios as possible; (3) involve both foreground and background for having diverse and balanced labels; and (4) locate on the boundary of some object, which may fully exploit context correlation. According to these prerequisites, we present a formulation of *information content*, termed as $Info$, of a region based on the superpixels inside it. During each iteration, the region with maximum $Info$ is selected as the most informative one for drawing scribbles. $Info$ is expressed as Equation (2),

$$Info = \Upsilon + \Gamma + \Lambda + \Delta, \tag{2}$$

where Υ denotes the similarity with its neighbors, Γ is the diversity inside the region, Λ represents the region entropy (foreground, background and unknown distribution), and Δ is the edge score of the region, corresponding the above prerequisites respectively. In practice, all these four entries will be balanced according to the number of superpixels in the current region. Among those, similarity (Υ) is calculated as:

$$\begin{aligned}\Upsilon = \lambda_1 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{out}} exp\left(-\frac{||cm_i - cm_j||^2}{2\sigma^2}\right) \\ + \lambda_2 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{out}} \frac{2(ch_i - ch_j)^2}{ch_i + ch_j + \theta} + \lambda_3 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{out}} \frac{2(th_i - th_j)^2}{th_i + th_j + \theta},\end{aligned} \tag{3}$$

meanwhile, diversity (Γ) is expressed as:

$$\begin{aligned}\Gamma = -\left(\lambda_1 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{in}} exp\left(-\frac{||cm_i - cm_j||^2}{2\sigma^2}\right)\right. \\ \left. + \lambda_2 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{in}} \frac{2(ch_i - ch_j)^2}{ch_i + ch_j + \theta} + \lambda_3 \sum_i^{\Omega_{in}} \sum_j^{\Omega_{in}} \frac{2(th_i - th_j)^2}{th_i + th_j + \theta}\right),\end{aligned} \tag{4}$$

where Ω_{in} is the set of superpixels inside the region, while Ω_{out} is the set of superpixels outside the region. The variables cm_i, ch_i, and th_i are middle-level features (color mean, color histogram, and texture histogram) at superpixel i. θ is a bias to prevent the denominator from being 0. The coefficients λ_1, λ_2, and λ_3 are set to 0.4, 0.35, and 0.25 in practice. The diversity is obtained by taking the overall similarity negative, because the higher the similarity, the smaller the difference and vice versa.

We define region entropy (Λ) as:

$$\begin{aligned}\Lambda = -\sum_i^{\Omega_{in}} [(pb_i)log(pb_i) + (pu_i)log(pu_i) \\ + (pf_i)log(pf_i)],\end{aligned} \tag{5}$$

where pb, pu, and pf refer to the Foreground, Background, and Unknown probabilities of superpixel i, respectively.

Input Image Edgemap Image

Fig. 3. Edge score (S) is calculated according to the corresponding edgemap image.

According to the object edges in Reference [45], we define edge score (Δ) as:

$$\Delta = \sum_{i}^{\Omega_{in}} e_i = \sum_{i}^{\Omega_{in}} \frac{\sum_{k}^{\psi_i} exp(em_k \delta)}{\varepsilon}, \tag{6}$$

where e_i is the edge value of superpixel i, calculated via edge value of pixels inside it. ψ_i is the set of pixels inside the superpixel i, em_k is the edge value of pixel k, and δ, ε are coefficients. An example edgemap image is shown in Figure 3, where white lines denotes potential object edges. Obviously, a high edge score means the region has a high probability of being at the edge of the object, corresponding to potential unknown in trimaps. The edge score can suggest transition content between different types, providing context correlation of separate regions.

3.3 Two-phase Information Propagation

The region with maximum *information content* is selected for drawing scribbles with distinguished colors, and some superpixels inside it are assigned with explicit category labels (foreground, red; background, blue; unknown, green) correspondingly. To minimize user effort and exploit the limited feedback from users (i.e., the labeled superpixels), we present a two-phase information propagation strategy to spread scribbles to the whole image, which not only takes advantage of local/low-level features with a re-modeled Markov process, but also the global/high-level semantics with a CNN model. We construct a probability matrix (PrM) to solve the label possibility of foreground, background, and unknown areas. The element PrM_{ij} in PrM indicates the transfer probability from superpixel i to j, and the two-phase propagation can update PrM continuously. Finally, a refined trimap can be inferred from PrM according to Algorithm 1.

Markov Propagation: We expect that user scribbles can propagate to the adjacent unlabeled superpixels massively, so employ the excellent spatial transfer capacity of Markov Chain [11, 32] to achieve this purpose (see Figure 2). In our framework, each superpixel in the image is modeled as a "node" in the Markov Chain (i.e., node i denotes superpixel i). We consider the superpixels with certain labels as absorbing nodes, while the unlabeled ones are transition nodes. Therefore, the labels propagation can be treated as the state transfer in Markov Chain, and solving unknown superpixel category labels is a process of probability transfer from transition nodes to absorbing ones. For the concrete implementation, suppose we divide n superpixels from over-segmentation; first, we can construct a connection matrix (CoM) to describe the position relevance of superpixels in the whole image. Then we can establish an affinity matrix (AfM) to measure the color and texture similarity between the superpixels. Both CoM and AfM are obviously $n \times n$ symmetric matrix. The probability matrix (PrM) is calculated by the CoM, AfM, and labels information, which is a $n \times m$ matrix (suppose there are m superpixels with user scribbles until current iteration). The element

ALGORITHM 1: Markov Propagation

Input: Connection Matrix (CoM), Affinity Matrix (AfM), Labeled Superpixels Set (L), Unlabeled Superpixels Set(U).

Output: Probability of Unlabeled Superpixels $\{pf_i\}$, $\{pb_i\}$, $\{pu_i\}$

for iterations **do**
  receives labels information from user drawing scribbles;
  update L and U;
  (CoM, AfM,L,U) $\rightarrow$ Probability Matrix (PrM);
  for Superpixel $i \in U$ **do**
    $SUM_f = 0, SUM_b = 0, SUM_u = 0$;
    for Superpixel $j \in L$ **do**
      Case $j \in Foreground : SUM_f = SUM_f + PrM_{ij}$
      Case $j \in Background : SUM_b = SUM_b + PrM_{ij}$
      Case $j \in Unknown : SUM_u = SUM_u + PrM_{ij}$
    end for
    $pf_i = \frac{SUM_f}{SUM_f+SUM_b+SUM_u}$
    $pb_i = \frac{SUM_b}{SUM_f+SUM_b+SUM_u}$
    $pu_i = \frac{SUM_u}{SUM_f+SUM_b+SUM_u}$
  end for
end for

return $\{pf_i\}$, $\{pb_i\}$, $\{pu_i\}$

PrM_{ij} in PrM indicates the probability of transition from node j (denotes labeled superpixel j) to node i (denotes unlabeled superpixel i), which is named as transition probability.

Transition probability is defined by the low-level appearance (e.g., color, texture) and spatial similarity between superpixels, and a higher transition probability PrM_{ij} indicates that i and j more likely share the same labels (foreground, background, or unknown). With the labeled superpixels increasing by drawing scribbles, the PrM is extended through the Markov Propagation in each iteration. For unlabeled superpixel i, its labels probabilities pf_i, pb_i, and pu_i in Equation (5) will be revised according to Algorithm 1. The updated labels probabilities will in turn renovate region entropy (E) in Equation (2) for next selection. Markov Propagation iterates along with informative regions selection, and label probabilities are revised continuously.

CNN Propagation: After N times of Markov Propagation, a coarse trimap is generated as shown in Figure 2. Although most superpixels can acquire proper labels, the propagation flow is localized and lack of consideration for high-level semantics of the superpixels. CNN has gained success in extracting high-level features of the input images [5, 21] and is also proved to be capable of spreading information and labels regardless of spatial limit [9]. Thereby, we develop an efficient CNN to extract features from the superpixels for global information propagation. We gather all the superpixels and extract their external rectangles to feed in the CNN Propagation and then acquire corresponding labels information from the preserved stroke image (Figure 2), the course trimap, and the input image. The labeled superpixels (with users scribbles or sky-high label probabilities) are considered a training set. The unknown typically lie in the transition between foreground and background, and their characteristics vary widely. Therefore, in context irrelevant superpixels-level CNN propagation, we only consider foreground and background (Figure 2) labels. After training, we can predict the label probabilities pf_i, pb_i for unlabeled superpixels via the trained model.

Figure 2 illustrates the CNN Propagation. The architecture has two branches: One is to extract high-level semantics from the centering patch of each superpixel via three convolutional layers. The visual features extraction takes a mini-batch for input and output a vector with 256 elements suggesting semantic relevance. The other branch is to encode the spatial information to a vector with 256 elements via a fully connected (FC) layer. We feed the center coordinates of the mini-batch into FC layer to provide some context correlation considering the global propagation. The semantic and spatial vectors are integrated by adding element-wisely. The integrated output is converted to a $3 * 1$ label probability map using a softmax function at last. The whole network is trained 20 epoches with descending learning rate and cross-entropy loss.

The proposed two-phase propagation can accomplish the overall update of PrM, and we can predict a desired trimap via Algorithm 1. Finally, some superpixels in the whole image obtain their labels (foreground, background, or unknown) from scribbles, while others from the two-phase information propagation by thresholding label probabilities. Here, we regard the superpixels whose pf_i is bigger than 0.65 as foreground empirically. Similarly, our framework can infer more background superpixels from the unlabeled ones. Remainder unlabeled superpixels are considered as Unknown. Afterwards, *smart scribbles* transform the foreground superpixels to white, background ones to black, and unknown ones to gray. Thus, we can obtain a refined trimap and the corresponding alpha matte is generated by embedding existing matting algorithms.

4 EXPERIMENTS AND RESULTS

In this section, we present our experiments and results analysis. The visual results on the real-world images are shown in Section 4.1, which can demonstrate the robustness and generalization of *smart scribbles*. In addition, we compare with traditional scribble-based methods on the matting benchmark [26], the portrait dataset [30], and the deep image matting [38] dataset (denoted as DIM) to exhibit the superiority of *smart scribbles*. We compare the alpha mattes generated by *smart scribbles* with several kinds of artificial trimaps based on the matting benchmark [26] to demonstrate the applicability with different matting algorithms. Then we evaluate our approach with three baselines and perform ablation study for *smart scribbles*, both on the matting benchmark [26]. The following experiments are all implemented using MATLAB, on a PC with an NVIDIA GTX 1080Ti GPU. We invite 20 users to participate in our experiments, and all of them are inexperienced to matting as well as our framework. All these users draw scribbles through "mspaint" program under windows system. In all following illustrations, red scribbles, blue scribbles and green scribbles represent the foreground, background, and unknown, respectively (the scribbles are appropriately bolded to make them more visible).

4.1 Comparison with Active Matting on the Real-world Images

Active Matting (AM) [40] can achieve competent alpha mattes with simple user interactions. The combination of CNNs and Reinforcement Learning in AM can effectively spread limited user information to the whole image to produce an accumulated trimap and the corresponding alpha matte can be generated from an existing embedded matting algorithm. However, like other deep learning–based matting networks, AM is also trained with artificial images, which significantly restricts the robustness and generalization of the network. We follow the matting model in AM to generate accumulated trimaps, then produce alpha mattes with existing matting methods. In this experiments, we feed the trimaps from *smart scribbles* into existing matting methods to compare with AM on the real-world images. The visual comparisons are illustrated in Figure 4 and Figure 5. The alpha mattes generated by *smart scribbles* and AM are both from four identical existing matting algorithms: ClosedForm [18], KNN [7], Deep Convolutional Neural Networks (DCNN) [8], and Information Flow Matting (IFM) [2]. The results of AM are coarse in diverse methods, even

Fig. 4. The visual comparisons with Active Matting [40] and Late Fusion [43] on the real-world images.

with more user interactions or relatively simple background. The reason for this is obvious: The overall model is fine-tuned on the rendered images. Illumination, blurring, and bokeh in real-world pictures can greatly reduce the network performance obtained from training on artificial images.

Compared to AM, *smart scribbles* can achieve alpha mattes in a more general and valid fashion. The proposed Markov and CNN propagation can spread limited scribbles to the whole image, and all operations are executed in superpixels-level of the input image, which means *smart scribbles* is independent of large-scale dataset. Although the synthetic images supply new possibility for deep learning matting models training, the clear artifacts on them significantly reduce the robustness and versatility of matting network. *Smart scribbles* can automatically adjust and adapt to the specific situation of the input image: (1) our informative region selection can effectively choose some

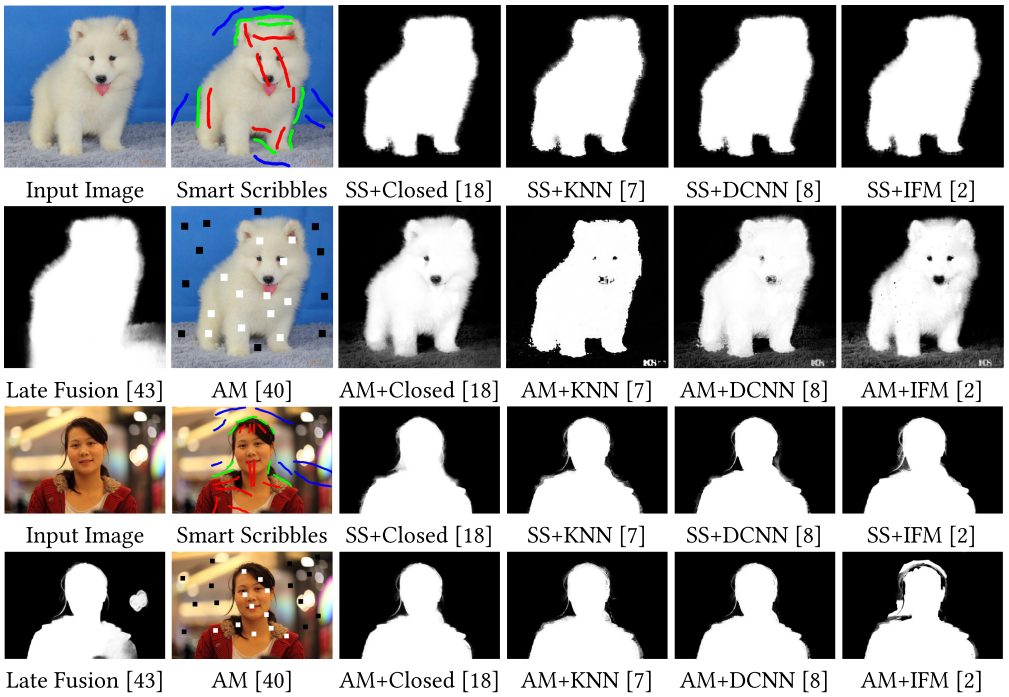


Fig. 5. The visual comparisons with Active Matting [40] and Late Fusion [43] on the real-world images.

representative regions (the ones with noise, blurring, etc.); (2) users can draw few scribbles on suggested regions to separate the foreground, background, and unknown; (3) the proposed two-phase propagation can effectually spread category labels to the whole image. Therefore, *smart scribbles* can assign category labels for each superpixel precisely, though the input image is taken in a poor or specific situation. The first row of Figure 4 demonstrates the robustness of our method, the bokeh and blurring in the input image did not degrade the performance of *smart scribbles*. The hair of the woman is clearly visible and the complicated background is perfectly eliminated.

4.2 Scribbles Evaluations

Traditional scribbles usually require professional knowledge to draw essential labels. The common users may get poor results even if they draw many scribbles on the whole image, because some prior conditions of the matting algorithms are left out of consideration. Compared with traditional scribbles, *smart scribbles* can generate better mattes with less scribbles. To demonstrate the superiority of *smart scribbles*, we conduct this experiment on the matting benchmark [26], the portraits testing dataset [30] with 300 images and the DIM dataset. Here we divide 20 inexperienced participators into two groups on average. One group draws traditional scribbles, and the other performs *smart scribbles*. Both groups are only told the basic knowledge of separating foreground, background, and unknown areas. The alpha mattes are produced with diverse matting algorithms (ClosedForm [18], SharedMatting [12], KNN [7], DCNN [8], and IFM [2]. The results are shown in Table 1 with different datasets. The evaluation metric is Root Mean Square Errors (RMSEs) compared with ground truths and *smart scribbles* achieves more than 40% improvement over conventional scribbles methods. *Smart scribbles* can obtain better alpha mattes with different matting algorithms. We attempt to generate alpha mattes with state-of-the-art matting algorithms

Table 1. The RMSE Comparisons with Traditional Scribbles on the Portrait Dataset [8] and Deep Image Matting (DIM) Dataset [38]

Methods & datasets	Smart scribbles	Traditional scribbles
ClosedForm [18]+Portraits	0.1024	0.1806
SharedMatting [12]+Portraits	0.0998	0.1778
KNN [7]+Portraits	**0.0726**	**0.1599**
DCNN [8]+Portraits	0.0870	0.1626
IFM [2]+Portraits	0.0733	0.5197
ClosedForm [18]+DIM	0.1104	0.1616
KNN [7]+DIM	0.0859	**0.1355**
DCNN [8]+DIM	0.0974	0.1390
IFM [2]+DIM	**0.0856**	0.4257

Fig. 6. The visual comparisons with traditional scribbles. (a) The input images from the portrait testing dataset or the DIM [38] dataset. (b) Ground truths. (c) Traditional scribbles. (d) Traditional scribbles + DCNN [8]. (e) *Smart scribbles*. (f) *Smart scribbles* + KNN [7]. (g) *Smart scribbles* + DCNN [8]. (h) *Smart scribbles* + IFM [2]. The alpha mattes produced by *smart scribbles* have more complete outlines and abundant texture details.

(IFM [2], but the traditional scribbles achieve poor results. The qualitative results are shown in Figure 6 and the preponderance of *smart scribbles* is more obvious on visual outcomes. Compared to traditional scribbles, the alpha mattes produced by *smart scribbles* have no extra background, and the boundary between foreground and background is clearly demarcated. Besides, some furry details can be detected more clearly by *smart scribbles*.

Figure 7 reflects the quantitative comparison of additional user inputs in this experiment. The entry "coverage percentage" indicates the percentage of the images covered by scribbles, which suggests how much labels information is provided through the user interactions. *Smart scribbles*

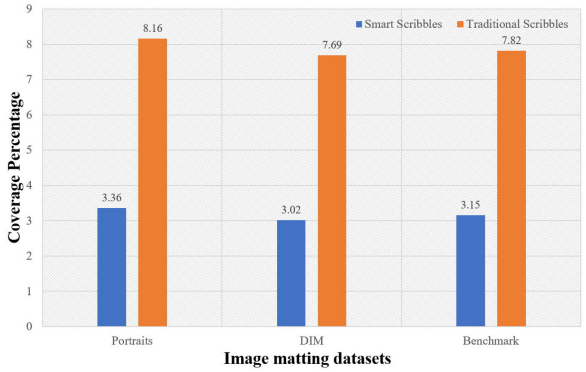


Fig. 7. The summarized coverage percentage on diverse datasets, which is the percentage of the input image covered by scribbles. The smaller percentage of *smart scribbles* indicates that it can produce better alpha mattes with less user interactions, as compared with traditional scribbles.

has lower "coverage percentage," approximately 60% reduction in the number of user scribbles on diverse datasets, indicating less user inputs getting above results. With informative regions selection, we can assure scribbles are drawn on crucial image areas. Limited user information can effectively spread to the whole image spatially and globally via Markov and CNN propagation. The above two points ensure that *smart* scribbles can produce refined mattes with minor scribbles.

4.3 Comparison with Trimaps

Both trimaps and scribbles are classical assistant matte inputs, and scribbles can also be regarded as sparse trimaps. To demonstrate the generality and availability of *smart scribbles*, here we show the comparisons with different forms of trimaps. Trimaps ground truths are accurate pixelwise annotations and difficult to implement (generally generated by dilation or erosion from alpha mattes, impractical for user interactions based on RGB images), hence we utilize full scratched and Grabcut trimaps instead. Full scratched means drawing scribbles on the whole images according to the trimaps ground truths. Grabcut trimaps are achieved by executing Grabcut [27] iteratively. Specifically, users are first asked to draw scribbles and a bounding box for distinguishing the foreground and background, and then we utilize Grabcut [27] to generate the corresponding trimaps. Such trimaps are employed to produce alpha mattes, and users continue to draw scribbles on the original images to improve these alpha mattes. Drawing scribbles and mattes generation are executing alternately, terminated when users are satisfied with the final alpha mattes. Both full scratched and Grabcut trimaps are approximately pixelwise annotations, close to trimaps ground truths (Figure 9). We evaluate our experiment on the matting benchmark [26]. The matte RMSEs compared to full scratched and Grabcut trimaps are shown in Figure 8 (there is no numerical value on the line chart indicating that the input is not suitable for this method.

As Figure 8 shows, the mattes RMSEs of full scratched and Grabcut trimaps are apparently lower than *smart scribbles*, suggesting that abundant user inputs can significantly improve the quality of final mattes. Nevertheless, the matte results generated by *smart scribbles*, using certain algorithms (e.g., KNN [7] and IFM [2]), can get close to some poor results in full scratched or Grabcut trimaps. Compared to full scratched and Grabcut trimaps, *smart scribbles* are more time-saving and user-friendly. We achieve the alpha mattes in Table 2 with average interaction time 41s, while full scratched and Grabcut trimaps take 8 min and 198 s, respectively. Besides, *smart scribbles* only requires several scribbles from users to label different categories, in contrast, more relaxed for novice users.

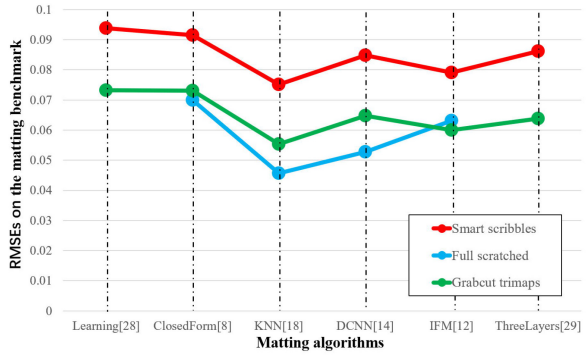
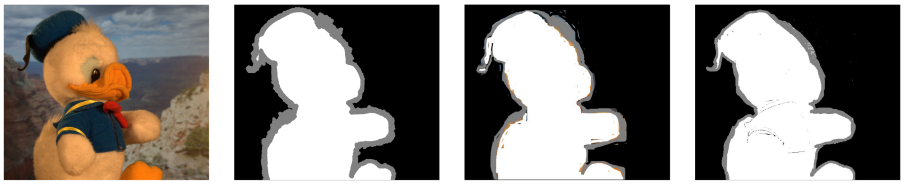


Fig. 8. The histogram comparisons with artificial trimaps on the matting benchmark [26]. Obviously, *smart scribbles* can get close to their poorer results. Full scratched is unavailable to Learning [44] and ThreeLayers [20] matting due to the presence of unspecified pixels.



(a) Input Image (b) Trimap GT (c) Full scratched (d) Grabcut trimap

Fig. 9. Diverse trimaps in our experiments. Full scratched and grabcut trimaps are both hand-crafted pixel-level trimaps.

Table 2. The Comparisons with Artificial Trimaps on the Matting Benchmark [26]

Methods	Smart scribbles	Full scratched	Grabcut trimaps
Learning [44]	0.0938	None	0.0732
ClosedForm [18]	0.0915	0.0700	0.0731
KNN [7]	0.0751	0.0456	0.0554
DCNN [8]	0.0848	0.0528	0.0647
IFM [2]	0.0791	0.0631	0.0600
ThreeLayers [20]	0.0862	None	0.0638
Average time	41s	8min	198s

The last row reflects the running time and the others are RMSEs. Full scratched means hand-crafted trimaps, and grabcut trimaps are produced via executing Grabcut [27] iteratively. *Smart scribbles* can approximate their poorer results (red marks), while taking much less time.

4.4 Comparison to Region Selection Baselines

We have constructed three baselines in our experiments:

B1: How to select informative regions? We propose *information content* to select *informative regions*, summarizing color, texture, label information, and object boundary. Here we compare the proposed region selection method with the other two. The first randomly select one region from the top 6 of information content per iteration, and the second way is for the users to specify regions based on their visual observation, which is to imitate the professional requirements for users in traditional scribbles. We term these two methods as random and user-specified regions, respectively, and replace the *information content* regions selection in our pipeline for comparisons.

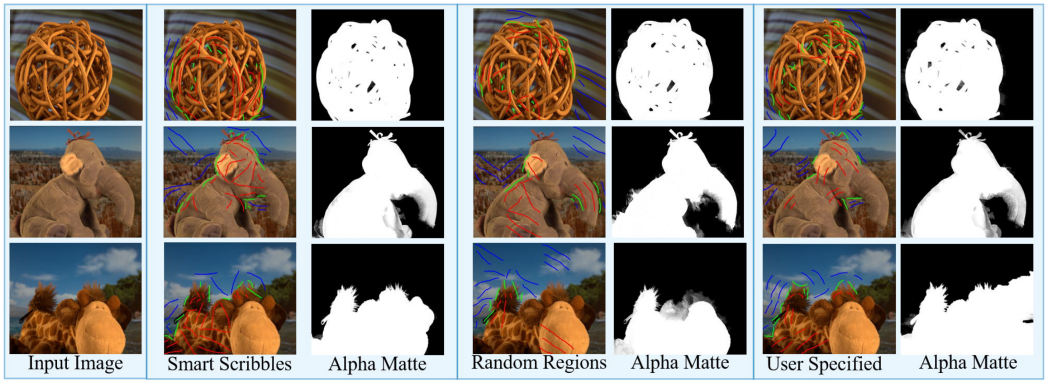


Fig. 10. *Smart scribbles* compared with randomly selected regions and user-specified regions. Both of them fail to take full advantage of limited scribbles, causing some details to be lost or even incomplete silhouettes in the final mattes.

Table 3. The Quantitative Results of Baseline1 and Two Ablation Study (RMSEs)

Methods	ClosedForm	KNN	DCNN
Random regions	0.2010	0.1434	0.1608
User-specified regions	0.1326	0.0942	0.1020
Without Markov propagation	0.2198	0.1717	0.1886
Without CNN propagation	0.1934	0.0963	0.1234
Without neighboring similarity	0.1452	0.0946	0.1068
Without inner diversity	0.1319	0.0940	0.0969
Without regions entropy	0.1624	0.1065	0.1182
Without edge score	0.2447	0.1766	0.1899
Smart scribbles	**0.0915**	**0.0751**	**0.0848**

The first two rows compare different region selection strategies and the next two lines are propagation ablation results. The last four rows besides *smart scribbles* reveal the significance of different information content entries.

Figure 10 shows the performance of different selection manners. We evaluate Root Mean Square Errors (RMSEs) of the alpha mattes produced by smart scribbles and ground truths on the matting benchmark training dataset, and the quantitative results are shown in Table 3. Random regions manner (Figure 10) specially fails in the thin structures (the furs of the elephant and monkey). User-specified manner give an inaccurate shape estimation (the last row of Figure 10). In contrast, as proposed *smart scribbles* consider the local affinity structure and the correlations across regions synthetically by a novel information content formulation, the global representative regions can be selected successfully. The *smart scribbles* demonstrates enormous superiority in three aspects: the structure details (the third row in Figure 10, the hair on the monkey), the shape completeness (the second row in Figure 10, the abdomen of the elephant), and the lower RMSE in Table 3.

B2: How many regions is essential? This experiment is to verify the impact of iterations in the phase of Markov propagation. In Markov propagation, we produce a coarse trimap after six iterations. A large number of iterations can memorably improve the quality of immediate trimaps

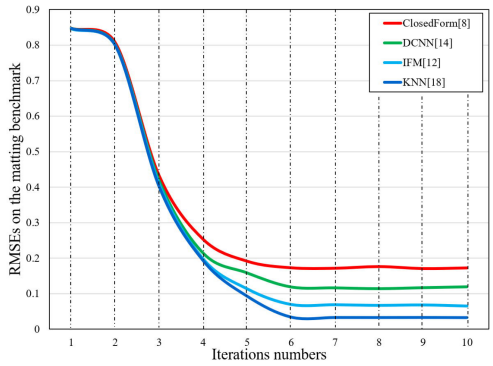
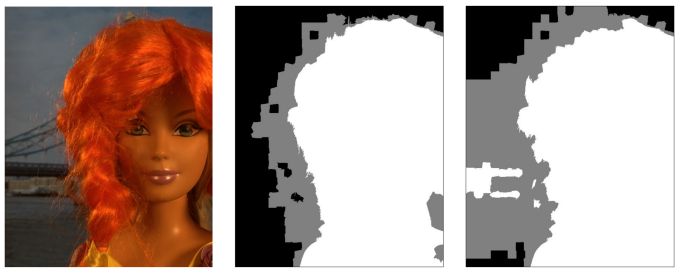


Fig. 11. Compare the number of iterations. The RMSEs gradually descend as the number of iterations increases and become stable after 6 times.



(a) Input Image (b) With Iteration (c) Without Iteration

Fig. 12. Trimap comparison with/without iterative region selection.

and final mattes. Nevertheless, the cumbersomeness of user interaction increases significantly as the number of iterations increases. We conduct our experiments with different iteration numbers to reflect the rationality of setting the number of iterations in *smart scribbles*.

As illustrated in Figure 11, here we employ diverse matting algorithms (ClosedForm [18], KNN [7], DCNN [8], and IFM [2]) to calculate alpha mattes, and the number of iterations is calculated from 1 to 10. In the initial two iterations, the two-phase propagation has insufficient labels information for reference, and therefore the results are unsatisfactory and RMSEs are higher. RMSEs decline as the number of iterations increases, and the trend of descending levels off when the iterations times up comes to 6. More iterations can improve mattes slightly, but more users labors are involved, which goes against our intention of reducing user labors. To balance the tradeoff between the quality and efficiency, we set the number of iterations as 6 in practice.

B3: Why select regions iteratively? For each region selection, we record users scribbles and perform a Markov propagation to update labels probabilities. For next selection, the entropy in the information content is recalculated according to new labels probabilities, while similarity, diversity, and the edge score remain. Here, we exclude the entropy from information content and select N regions at a time for users to drawing scribbles. Then Markov propagation is executed once, followed by the CNN propagation. Here we display the trimaps comparisons for more intuitive (Figure 12). Selecting regions at once weakens the impact of Markov propagation on adjacent affinity, leading to local discontinuity, as shown in the left side of Figure 12.

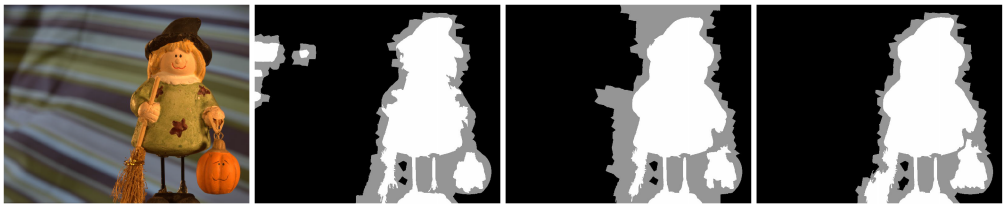


Fig. 13. The qualitative results of the propagation ablation study. The proposed two propagation phases are complementary and the absence of either can possibly lead to prediction errors in the refined trimaps.

4.5 Ablation Study and Analysis

Propagation ablation. Here we verify the effectiveness of Markov propagation and CNN propagation, where the two components are evaluated by removing the other one from *smart scribbles*. We conduct ablation experiment on the matting benchmark training images to evaluate three kinds of framework: *smart scribbles*, the framework without Markov propagation and the framework without CNN propagation. Here we adopt ClosedForm [18], KNN [7], and DCNN [8] for mattes generation. The quantitative results are summarized in Table 3, and the visual trimaps are shown in Figure 13. The framework without Markov propagation and the framework without CNN propagation both produced less competent results, inferior to our full framework. The absence of Markov propagation affects the continuity of spatial label distribution, while the CNN propagation resorts to the high-level features of the superpixels themselves, lacking the constrain of image context. The ablation experiment demonstrates that both propagation are essential in *smart scribbles*.

Information content entry ablation. We summarize four entries for information content formulation according to Equation (2), and all them are essential for informative regions selection. Here we conduct an ablation experiment removing four entries in turn, to illuminate the significance of them. We calculate information content without neighboring similarity, without inner diversity, without entropy, and without the edge score, respectively, and produce corresponding mattes with DCNN [8]. The quantitative RMSEs results are shown in Table 3 and the combination of these four terms leads to a minimum RMSE (0.0848). The absence of either entry can discount the final mattes: the removal of similarity or diversity lack of consideration for image color and texture, the entropy elimination has no regard for existing labels information, and the edge score has the greatest impact due to its accurate identification of object edges.

5 CONCLUSION AND FUTURE WORK

In this article, we propose a new interactive framework for image matting, called *smart scribbles*. We explore the principles of informative regions for matting, and an informative measurement strategy is presented for proposing regions for users labeling. It suggest informative regions for users to draw scribbles for labeling the foreground, background, and unknown. A fine trimap can then be obtained by the proposed two-phase information propagation. Extensive experiments have proved the validity and universality of our framework and *smart scribbles* can be applied to various matting algorithms.

Figure 14 shows a failure case of the proposed method. All alpha mattes in this example are generated using IFM [2]. *Smart scribbles* cannot handle regions where the foreground and background intersect severely, causing many details to be missing (Figure 14). For future research, we aim to extend the proposed framework to a real-time editing system.

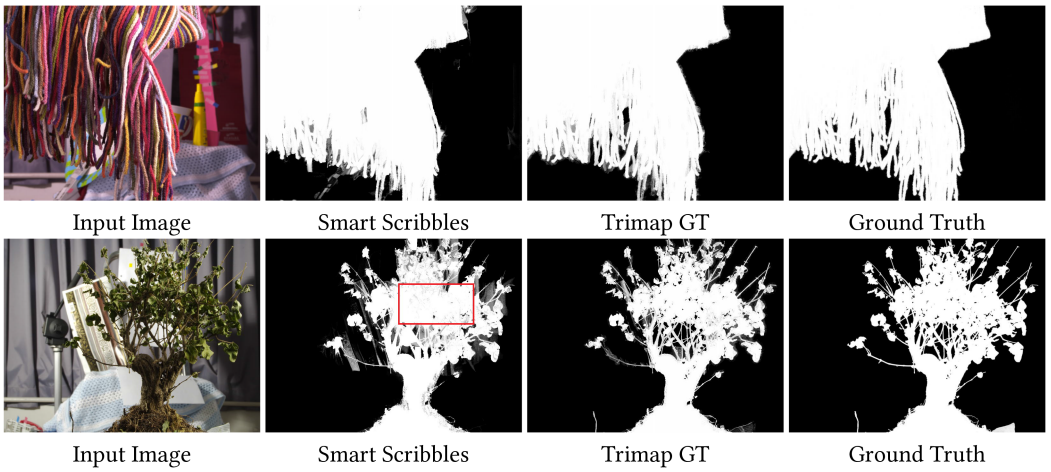


Fig. 14. The comparisons with trimap ground truths. Although *smart scribbles* demonstrate complete contours and sophisticated texture details, compared to trimap GTs, there are still some local estimation errors.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 11 (2012), 2274–2282.

[2] Y. Aksoy, T. O. Aydin, and M. Pollefeys. 2017. Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 228–236.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 12 (2017), 2481–2495.

[4] S. Cai, X. Zhang, H. Fan, H. Huang, J. Liu, J. Liu, J. Liu, J. Wang, and J. Sun. 2019. Disentangled image matting. In *Proceedings of the International Conference on Computer Vision (ICCV'19)*. 8818–8827.

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 834–848.

[6] Quan Chen, Tiezheng Ge, Yanyu Xu, Zhiqiang Zhang, Xinxin Yang, and Kun Gai. 2018. Semantic human matting. In *Proceedings of the ACM International Conference on Multimedia (MM'18)*. 618–626.

[7] Q. Chen, D. Li, and C. Tang. 2013. KNN matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 9 (2013), 2175–2188.

[8] Donghyeon Cho, Yu-Wing Tai, and Inso Kweon. 2016. Natural image matting using deep convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. 626–643.

[9] Yuki Endo, Satoshi Iizuka, Yoshihiro Kanamori, and Jun Mitani. 2016. DeepProp: Extracting deep features from a single image for edit propagation. In *Proceedings of the Annual Conference of the European Association for Computer Graphics (EG'16)*. 189–201.

[10] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. 2016. A cluster sampling method for image matting via sparse coding. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. 204–219.

[11] Mauro Gasparini. 1997. Markov chain Monte Carlo in practice. *Technometrics* 39, 3 (1997), 338–338.

[12] Eduardo S. L. Gastal and Manuel M. Oliveira. 2010. Shared sampling for real-time alpha matting. *Comput. Graph. Forum* 29, 2 (2010), 575–584.

[13] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. 2005. Random walks for interactive alpha-matting. In *Proceedings of the Visualization, Imaging, and Image Processing (VIIP'05)*. 423–429.

[14] Yu Guan, Wei Chen, Xiao Liang, Zi'ang Ding, and Qunsheng Peng. 2006. Easy matting—A stroke based approach for continuous image matting. *Comput. Graph. Forum* 25, 3 (2006), 567–576.

[15] Q. Hou and F. Liu. 2019. Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the International Conference on Computer Vision (ICCV'19)*. 4129–4138.

[16] L. Karacan, A. Erdem, and E. Erdem. 2015. Image matting with KL-divergence based sparse sampling. In *Proceedings of the International Conference on Computer Vision (ICCV'15)*. 424–432.

[17] P. Lee and Ying Wu. 2011. Nonlocal matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 2193–2200.

[18] Anat Levin, Dani Lischinski, and Yair Weiss. 2007. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 2 (2007), 228–242.

[19] Anat Levin, Alex Rav-Acha, and Dani Lischinski. 2008. Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 10 (2008), 1699–1712.

[20] Chao Li, Ping Wang, Xiangyu Zhu, and Huali Pi. 2017. Three-layer graph framework with the sumD feature for alpha matting. *Comput. Vision Image Understand.* 162 (2017), 34–45.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*. 3431–3440.

[22] H. Lu, Y. Dai, C. Shen, and S. Xu. 2019. Indices matter: Learning to index for deep image matting. In *Proceedings of the International Conference on Computer Vision (ICCV'19)*. 3265–3274.

[23] Sebastian Lutz, Konstantinos Amplianitis, and Aljoscha Smolic. 2018. AlphaGAN: Generative adversarial networks for natural image matting. In *Proceedings of the British Machine Vision Conference (BMVC'18)*. 259.

[24] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20)*.

[25] C. Rhemann and C. Rother. 2011. A global sampling method for alpha matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 2049–2056.

[26] C. Rhemann, C. Rother, Jue Wang, M. Gelautz, P. Kohli, and P. Rott. 2009. A perceptually motivated online benchmark for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. 1826–1833.

[27] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23, 3 (2004), 309–314.

[28] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. 2013. Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*. 636–643.

[29] E. Shelhamer, J. Long, and T. Darrell. 2017. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017), 640–651.

[30] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. 2016. Deep automatic portrait matting. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. 92–107.

[31] Jian Sun, Jiaya Jia, Chi Keung Tang, and Heung Yeung Shum. 2004. Poisson matting. *ACM Trans. Graph.* 23, 3 (2004), 315–321.

[32] J. Sun, H. Lu, and X. Liu. 2015. Saliency region detection based on Markov absorption probabilities. *IEEE Trans. Image Process.* 24, 5 (2015), 1639–1649.

[33] J. Tang, Y. Aksoy, C. Oztireli, M. Gross, and T. O. Aydin. 2019. Learning-based sampling for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 3050–3058.

[34] Jue Wang and Michael F. Cohen. 2005. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of the International Conference on Computer Vision (ICCV'05)*. 936–943.

[35] Jue Wang and Michael F. Cohen. 2007. Optimized color sampling for robust matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. 1–8.

[36] Yuhang Wang, Jing Liu, Yong Li, Junjie Yan, and Hanqing Lu. 2016. Objectness-aware semantic segmentation. In *Proceedings of the ACM International Conference on Multimedia (MM'16)*. 307–311.

[37] Ke Xu, Xin Wang, Xin Yang, Shengfeng He, Qiang Zhang, Baocai Yin, Xiaopeng Wei, and Rynson W. H. Lau. 2018. Efficient image super-resolution integration. *Visual Comput.* 34, 6–8 (2018), 1065–1076.

[38] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. 2017. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 311–320.

[39] Xin Yang, Haiyang Mei, Jiqing Zhang, Ke Xu, Baocai Yin, Qiang Zhang, and Xiaopeng Wei. 2019. DRFN: Deep recurrent fusion network for single-image super-resolution with large factors. *IEEE Trans. Multimedia* 21, 2 (2019), 328–337.

[40] Xin Yang, Ke Xu, Shaozhe Chen, Shengfeng He, Baocai Yin Yin, and Rynson Lau. 2018. Active matting. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS'18)*. 4590–4600.

[41] Yung-Yu Chuang, B. Curless, D. H. Salesin, and R. Szeliski. 2001. A Bayesian approach to digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*. II–II.

[42] Jiqing Zhang, Chengjiang Long, Yuxin Wang, Xin Yang, Haiyang Mei, and Baocai Yin. 2020. Multi-context and enhanced reconstruction network for single image super resolution. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'20)*. 1–6.

[43] Y. Zhang, L. Gong, L. Fan, P. Ren, Q. Huang, H. Bao, and W. Xu. 2019. A late fusion CNN for digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 7461–7470.

[44] Yuanjie Zheng and Chandra Kambhamettu. 2009. Learning based digital matting. In *Proceedings of the International Conference on Computer Vision (ICCV'09)*. 889–896.

[45] C. Lawrence Zitnick and Piotr Dollar. 2014. Edge boxes: Locating object proposals from edges. In *Proceedings of the European Conference on Computer Vision (ECCV'14)*. 391–405.

Received January 2020; revised April 2020; accepted May 2020