A Novel Image Matting Approach Based on Naive Bayes Classifier

Zhanpeng Zhang[1,2], Qingsong Zhu[1,*], and Yaoqin Xie[1]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China
[2] Sun Yat-Sen University, Guangzhou, 510006, China
{zp.zhang,qs.zhu,yq.xie}@siat.ac.cn

Abstract. Image matting is a fundamental technique used in many image and video applications. It aims to softly extract foreground from the image accurately. In this paper, we propose a new matting approach based on naive Bayes classifier to produce matting results with higher accuracy. Spatially-varying probabilistic models for the classifier are established. Confidence values are defined to make better use of the classification results. The results are then refined and combined with closed-form matting to obtain the final alpha matte. We conduct qualitative and quantitative evaluations. Results show that our method outperforms many recent algorithms.

Keywords: image matting, naive bayes classifier, foreground extraction, image segmentation.

1 Introduction

Image matting refers to the problem of extracting the opacity mask (typically called alpha matte) of the foreground, as well as the foreground and background color, from the target image. It is one of the fundamental techniques used in many image and video editing tasks. Specifically, for a pixel i with color I_i in the image, it can be model as a liner combination of the foreground color F_i and background color B_i as Eq.1.

$$I_i = \alpha_i F_i + (1-\alpha_i) B_i \tag{1}$$

Eq.1 is under constrained since α_i, F_i and B_i on the right-hand side are all unknown, making it a significant challenge for computer vision.

Many recent approaches require additional information from user input to build more constrains to solve the ill-posed problem. Trimaps [1] and scribbles [2] are the two most common methods, labeling some pixels which are definite foreground or background (like Fig.1b), with corresponding alpha value to be 1 or 0 respectively.

Existing matting algorithms can be classified into sampling-based or propagation-based. Sampling-based algorithms explicitly estimated the triplet (α, F, B) for every unlabeled pixel by analyzing nearby labeled pixels. These algorithms usually fit a parametric mode to the color distributions of the samples, like Bayesian Matting [1], which models the colors of the samples by oriented Gaussian distributions. However, the model assumptions may fail in some scenes. Recently, many

D.-S. Huang et al. (Eds.): ICIC 2012, LNCS 7389, pp. 433–441, 2012.

non-parametric algorithms are employed, estimating the triplet directly with the sample colors under the liner model defined by Eq.1. For example, the sampling algorithms in robust matting [3] and many later approaches [4,5] assign a confidence value to a pair of foreground and background sample and choose the samples with high confidence. The confidence values are generally measured by how well the sample explains the unlabeled pixel in the linear model (Eq.1). These algorithms perform well when the true foreground and background colors are in the sample set.

For propagation-based algorithms, some affinities assumptions are made in order to derivate a constrained objective function. Poisson matting [6] deduces that the alpha matte gradient is proportional to the image gradient under the smoothness assumption. A closed form solution proposed by Levin obtains a quadratic cost function in α analytically under the color line model [2]. This approach is called as closed-form matting and has been applied in many other approaches, drawing extensive studies [7].

However, due to the color line assumption in closed-form matting, it may fail in regions with long and thin structures or holes. Many other propagation-based approaches may also fail in these situations. This is because only neighboring pixels are used in the modeling process, lacking information from further regions. Some approaches [3,4] use color information from nearby samples to fill the gap. However, gathering appropriate samples is still a challenging problem, and additional assumptions may be introduced, causing other types of artifacts.

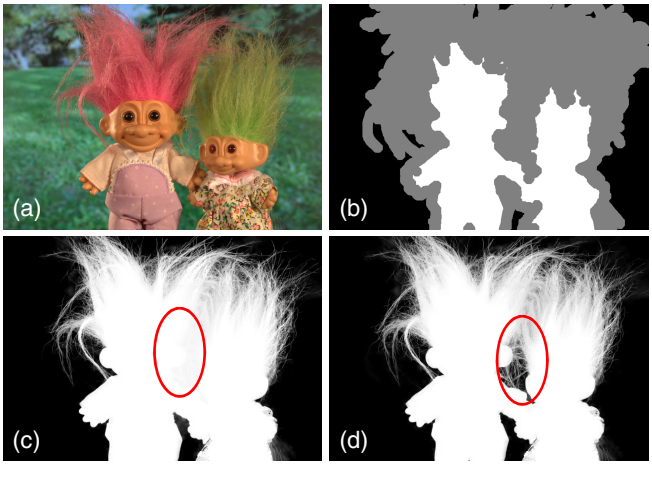

Fig. 1. (a) Original image. (b) Trimap. (c) Closed-form matting [2] result. (d) Our result (Notice the two red ellipses. The detail is missed in closed-form matting while not in the result of our method.).

In this paper, we employ naive Bayes classifier to identify the foreground and background pixels. The classification results are then refined and combined with closed-form matting. As our approach can enhance the discrimination between the foreground and the background, results show that our approach outperforms many

recent algorithms and produces better alpha mattes for the images in which the closed-form matting usually fails (like Fig.1).

2 Closed-Form Matting

Levin *et al.* proposed a closed-form solution for image matting in [2]. The key assumption in closed-form matting is color line model. It assumes that in a small window, the foreground color and background color of each pixel *i* can be formulated as linear mixtures of two constant colors, respectively. In other words, the foreground or background colors for all pixels in a small window lie on a single line in the RGB color space.

Based on the color line model, the alpha value for pixel *i* in a small window *w* can be expressed as a liner transform of the pixel color:

$$\alpha_i = \sum_c a^c I_i^c + b, \forall i \in w \tag{2}$$

where *c* denotes the color channel for RGB color space. *a* and *b* are constant in *w*.

To derivate an alpha matte obeying the color line model (or Eq.2), the algorithm aims to find the optimal a, b and α which minimize the cost function:

$$J(\alpha, a, b) = \sum_{j \in I} \{ \sum_{i \in w_j} (\alpha_i - \sum_c a_j^c I_i^c - b_j)^2 + \varepsilon \sum_c {a_j^c}^2) \tag{3}$$

Here wj is a small window centered at pixel j. ε is a regularization parameter while smoothing the alpha matte.

A quadratic function of α can be obtained by minimizing the cost function. Also, parameters a, b can be eliminated in the deducing process. The quadratic function is as follows:

$$J(\alpha, a, b) = \alpha^T L \alpha \tag{4}$$

pecifically, α is an N×1 vector, where N is the number of unlabeled pixels. L is typically called matting Laplacian matrix, as one of the most significant contributions of closed-form matting, drawing many further studies and applications. Formally, L is an N×N matrix with (i, j)-th element as:

$$\sum_{k|(i,j) \in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + (I_i - \mu_k)(\sum_k + \frac{\varepsilon}{|w_k|} I_3)^{-1}(I_j - \mu_k))) \tag{5}$$

where δ_{ij} is the Kronecker delta. $|w_k|$ is the number of pixels in this window. $\sum_k$ is a 3×3 covariance matrix, μ_k is a 3×1 mean vector of the colors in a window w_k, and I_3 is the 3×3 identity matrix.

Combined with constrains provided by the user (trimaps or scribbles), the objective function can be defined as:

$$J(\alpha) = \alpha^T L \alpha + \lambda (\alpha - \beta)^T D (\alpha - \beta) \tag{6}$$

Here, D is an N×N diagonal matrix whose elements are 1 for labeled pixels and 0 otherwise. β denotes the alpha value for the labeled pixels in the trimap (1 for

foreground and 0 for background). λ is weighted parameter with a relatively large number (like 100). In this paper, we mainly talk about the alpha value since the foreground F and background B can be obtained easier with the estimated α (like [2]).

Closed-form matting works well when the local region fits the color line model. To ensure this assumption holds and reduce computation cost, the windows size in Eq.5 is usually small (3×3 in Levin's implementation). That means the "propagation step" is relatively small implicitly. Over-smoothing may happen in regions with thin structures or small gaps as Fig.1c. Details will be lost in these situations. He et al. [7] improves it by introducing adaptively window sizes in different regions of the image. It is shown that larger window size can improve the matting result since large window may cover disconnected regions of the foreground/background. However, with larger window size, it is more like to break the color line model. Therefore, it is still hard to decide the appropriate windows size.

3 Matting with Naive Bayes Classifier

Different from the previous approaches, we employ naive Bayes classifier to decide whether a pixel belong to the foreground or background. The result is then "softened" by a sigmoid function. Confidence values are also computed for every alpha value. The results and the confidence values are then combined with closed-form matting, providing more accurate alpha matte.

3.1 Naive Bayes Classifier

Naive Bayes classifier [8] is a simple probabilistic classifier based on Bayes' theorem, assuming that features in a class are independent with each other. For class variable C with n features in the model, using Bayes' theorem, the probability that an instance is in class c is:

$$p(c \mid k_1, k_2 ... k_n) = \frac{p(c)\, p(k_1, k_2 ... k_n \mid c)}{p(k_1, k_2 ... k_n)} \tag{7}$$

where ki is the instance's value for feature i. With the assumption that every feature is independent, classification can be done by selecting the highest posterior of the classification variable with the following function:

$$\arg\max_c \; p(C = c) \prod_{i=1}^{n} p(K_i = k_i \mid C = c) \tag{8}$$

3.2 Classification Process

Features Selection. Color attribute is the most straight forward feature reflecting a pixel's characteristics. To provide more information of the region texture, colors of the 4-neighbor pixels are also selected. That means, for a pixel i, its feature vector $k = \{g_1, g_2, g_3, g_4, g_5\}$ where g_i is the color vector of a pixel. In our implementation, we use CIELAB color space (with every channel ranges from 0 to 255), so a 15-dimensional vector is selected for a pixel.

Classifier Parameters Estimation. We do not apply a uniform classifier probabilistic model for all the unlabeled pixels in the image. Instead, the probabilistic model is spatially-varying. Specifically, for an unlabeled pixel i, we collect other unlabeled pixels with spatial distance less than r (30 in our implementation). Pixels from this unlabeled region share a same probabilistic model. Pixels labeled by the users are selected as samples to estimate the parameters. We expect to obtain both local foreground and background characteristics, so samples are selected according to their spatial distance to the current unlabeled region. We expand form the border of the unlabeled region and collect foreground and background samples, until the numbers of the foreground and background samples are larger than that of unlabeled region, respectively. To make the features independent to some extent so as to satisfy the naive Bayes' assumption better, before parameters estimation, we apply PCA [9] to reduce the dimension of the features vector for the samples and unlabeled pixels.

Gaussian distribution is employed to model the probability of each feature of the two classes (foreground and background). The mean and variance can be computed with the collected samples. The class' prior ($p(c)$ in Eq.7) is calculated by assuming that the foreground and background are equiprobable. That means $p(c) = 0.5$ for both the two classes. With the computed parameters, unlabeled pixels can be classified as Eq.8.

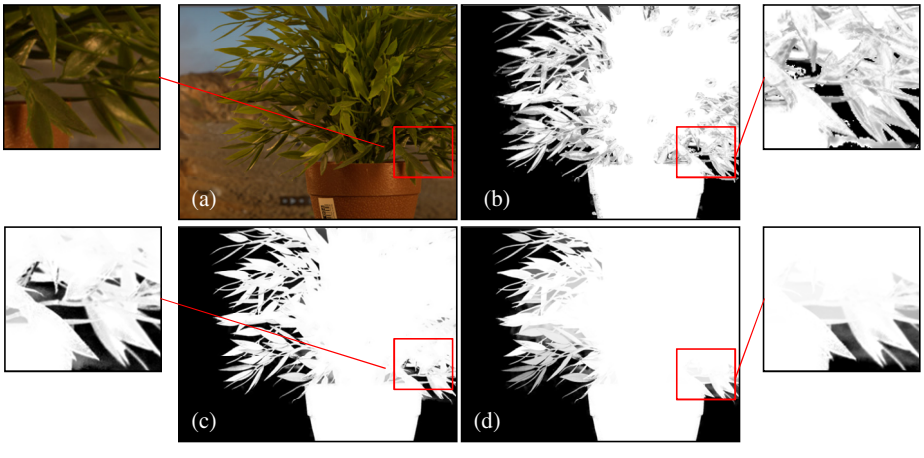

Fig. 2. (a) Original image (b) Refined result (c) Result after combined with closed-form matting (d) Closed-form matting only. Parts of the images are zoomed in for clearer distinction.

Results Refinement. With the Gaussian distribution and strong independence assumptions in the classifier, as well as the binary classification result, errors are unavoidable. To evaluate the results from the classifier, pixel-wise confidence values are computed. Also, the binary results are "softened" with a sigmoid function so as obtain values ranging from 0 to 1 indicating the opacity mask.

Instinctively, if the unlabeled pixels are more similar to the samples used to estimate the classifier, more accurate results can be obtained. To measure the similarity, we calculate D_s for every unlabeled pixel. D_s is defined as follows:

$$D_s = \min E(i, j), j \in S \tag{9}$$

where $E(\cdot)$ denotes the Euclidean distance in the original feature space. S is the collected sample set. The confidence value is:

$$G(i) = \exp\{-\rho D_s(i)\} \tag{10}$$

And the binary results are "softened" with a sigmoid function as:

$$R(i) = \frac{1}{1+\exp\{-\vartheta C(i) / D_s(i)\}} \tag{11}$$

$C(i)$ is the result of the classifier for pixel i, -1 and 1 for background and foreground, respectively. ρ and θ in the above equations are normalize parameters. In our implementation, we set $\rho = 0.5$ and $\theta = 20$. The result R is like Fig.2b. In practice, the target foreground in matting is usually connected, not several separated components. We select connected components with area less than 1% of the area of the largest one, and set the confidence values of these components to be zero.

3.3 Combined with Closed-Form Matting

We use the refined result R in Eq.11 and its corresponding confidence to define the data term as a quadratic function with the minimum at R. The data term is then combined with Eq.6. The final objective function is:

$$J(\alpha) = \alpha^T L\alpha + \lambda(\alpha-\beta)^T D(\alpha-\beta) + \gamma(\alpha-R)^T E(\alpha-R) \tag{12}$$

The first and second term is as defined in Eq.6. For the third term, R is treated as a vector. E is an $N \times N$ diagonal matrix whose elements are the confidence values for corresponding unlabeled pixels and zeros otherwise. γ is a weighted parameter (0.1 in our implementation). The combined result is like Fig. 2c.

4 Experiments

We conduct qualitative and quantitative comparisons of our method with other recent related matting algorithms, including closed-form matting [2], learning based matting [10], large kernel matting [7] and robust matting [3]. Like closed-form matting, learning based matting also used a small window, but it employs learning techniques instead of the color line assumption. Large kernel matting improves the efficiency of closed-form matting by using larger window size. Robust matting combines the sampling-based and propagation-based algorithms, similar to our approach to some extent.

Qualitative Comparisons. We compare the result of our method and closed-form matting visually in Fig. 3a. The images and ground truth are provided by [11]. We can see that closed-form matting may fail in the regions with gaps. These regions are often

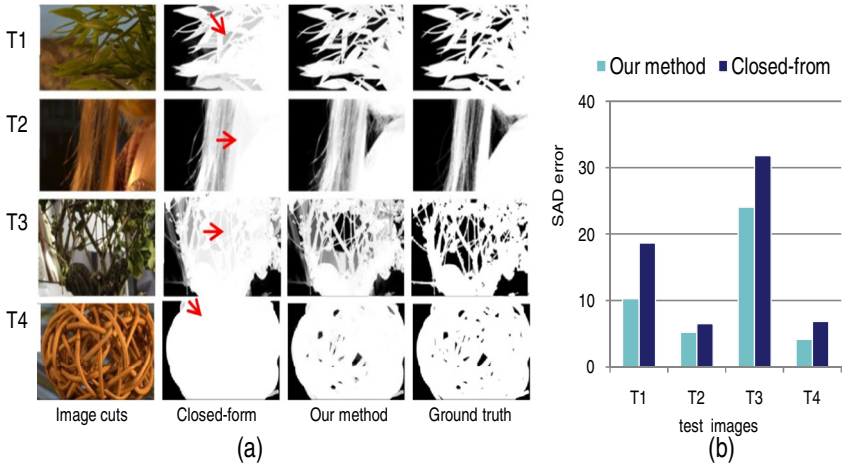

Fig. 3. (a) Qualitative comparison between closed-form matting and our method with 4 test images form [11]. Only parts of the images are shown for clearer distinction. The red arrows show the regions where closed-form matting fails while our method provides better results. (b) Quantitative comparison between our method and closed-form matting according to the SAD (sum of absolute difference) error.

recognized as definitely foreground and the details are missed. With the classification process and samples from further regions, our method can get background samples with which to indentify the background pixels and avoid this situation to some extent. However, compared to the ground truth, our method still has some artifacts. This is mainly because of the error in the classification and results refinement process. The sigmoid function simply based on the Euclidean distance in the feature space is not always effective.

Quantitative Comparisons. Fig.3b shows the quantitative comparison between our method and closed-form matting. We can see that our method can provide results with less SAD (sum of absolute difference) error. To conduct more comprehensive evaluation, we also use the matting benchmark of [11], with 8 test images and 3 different trimaps (with different sparsity level) for each of them. The average SAD errors of each method for the 3 different types of trimaps are presented in Fig.4. The comparison shows that our method is performing the best. Our method combines nearby samples and the smooth assumption for local region, providing better results. Compared to robust matting, our method does not need to find the true foreground and background color. So the accuracy can be higher in regions with color ambiguity.

Memory and Computation Cost. The memory cost in our classification process is relative small since we just establish spatially-varying probabilistic model. That means the sample size used for parameters estimation is not large (less than 3000 pixels for foreground and background respectively). We implement our algorithm in Matlab, and run it on a 3.0 GHz CPU. The classification process typically takes 20 seconds for an 800×600 image, varying with the size N of the unlabeled region in the trimap. The running time can be further reduced with parallel implementation (like GPU), since

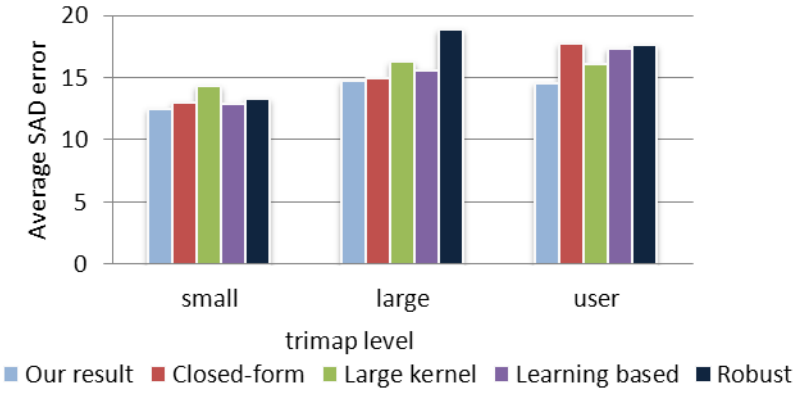

Fig. 4. Quantitative comparisons on average SAD error of alpha value in trimaps with different sparsity level (small, large, user). For the details of the sparsity level categories, refer to [11].

classification processes for different regions of pixels are independent. However, the matting Laplacian matrix needs large size of memories with the size of $N \times N$ as described in Section 2. And the time for computing the matting Laplacian matrix and solving Eq.12 in Section 3.2 is about 20 seconds for an 800 × 600 image, depending on N.

Limitation. Because the classification process is based on the color information, our method may fail in complex scenes or regions with foreground and background colors overlapping. Also, the naive Bayes classifier is based on a strong assumption and we used Gaussian distribution to model the probability. In some situations, these assumptions may not hold.

5 Conclusion

In this paper, a new matting approach based on naive Bayes classifier is proposed and evaluated. The binary classification results are "softened" with a sigmoid function and confidence values are computed to make better use of the results. The results are then combined with close-form matting to obtain the final alpha matte. Quantitative and qualitative comparisons between our method and other recent algorithms show that our method produce better results. However, color ambiguity or complex scenes are still challenging for our method. Future work may concentrate on providing better classification results and weaker model assumptions.

Acknowledgments. This study has been financed partially by the Projects of National Natural Science Foundation of China (Grant No. 50635030, 60932001, 61072031, 61002040), the National Basic Research (973) Program of China (Sub-grant 6 of Grant No. 2010CB732606) and the Knowledge Innovation Program of the Chinese Academy of Sciences, and was also supported by the China Scholarship Council (CSC) and China Postdoctoral Project.

Reference

1. Chuang, Y.Y., Curless, B., Salesin, D.H., et al.: A Bayesian Approach to Digital Matting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 264–271 (2001)
2. Anat, L., Dani, L., Yair, W.: A Closed Form Solution to Natural Image Matting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 61–68 (2006)
3. Wang, J., Cohen, M.F.: Optimized Color Sampling for Robust Matting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17–22 (2007)
4. Rhemann, C., Rother, C., Gelautz, M.: Improving Color Modeling for Alpha Matting. In: Proceedings of British Machine Vision Conference, pp. 1155–1164 (2008)
5. Gastal, E.S.L., Oliveira, M.M.: Shared Sampling for Real-time Alpha Matting. Computer Graphics Forum 29, 575–584 (2010)
6. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Transactions on Graphics 23(3), 315–321 (2004)
7. He, K.M., Sun, J., Tang, X.O.: Fast Matting Using Large Kernel Matting Laplacian Matrices. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2165–2172 (2010)
8. Pedro, D., Michael, P.: On the Optimality of the Simple Bayesian Classifier under Zero-one Loss. Machine Learning 29, 103–130 (1997)
9. Jolliffe, I.T.: Principal Component Analysis. Springer (1986)
10. Zheng, Y.J., Kambhamettu, C.: Learning Based Digital Matting. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 889–896 (2009)
11. Rhemann, C., Rother, C., Wang, J., et al.: A Perceptually Motivated Online Benchmark for Image Matting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1826–1833 (2009)