A Novel Nonlinear Regression Approach for Efficient and Accurate Image Matting

Qingsong Zhu, Zhanpeng Zhang, Zhan Song, *Member, IEEE*, Yaoqin Xie, and Lei Wang, *Member, IEEE*

***Abstract*—Current image matting approaches are often implemented based upon color samples under various local assumptions. In this letter, a novel image matting algorithm is investigated by treating the alpha matting as a regression problem. Specifically, we learn spatially-varying relations between pixel features and alpha values using support vector regression. Via the learning-based approach, limitations caused by local image assumptions can be greatly relieved. In addition, the computed confidence terms in learning phase can be conveniently integrated with other matting approaches for the matting accuracy improvement. Qualitative and quantitative evaluations are implemented with a public matting benchmark. And the results are compared with some recent matting algorithms to show its advantages in both efficiency and accuracy.**

***Index Terms*—Foreground extraction, image matting, image segmentation, support vector machine.**

I. Introduction

IMAGE matting and compositing are essential operations in a variety of multimedia editing systems [1]. It refers to the problem of extracting foreground objects from still images or video sequences. Mathematically, in image matting, the input image I can be modeled as a linear combination of the foreground image F and the background image B:

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

where α is the opacity mask (alpha matte) of the foreground. The opacity (alpha value) of every pixel ranges from 0 to 1. Equation (1) is under constrained since α, F and B are all unknown. A practical means is the trimap method which demands for some user-input labels as shown in Fig. 1.

Generally, current image matting techniques can be classified as sampling-based or propagation-based approaches. Sampling-based techniques select color samples from the labeled regions to estimate the alpha values [2]–[4]. Propagation-based techniques interpolate the alpha matte by propagating constrains from the labeled regions to the unlabeled regions with local assumptions such as local smoothness etc. [5]. However, foreground and background color samples are sometimes difficult to determine or even not retreievable (e.g., for the translucent objects). Meanwhile, some local assumptions may not always hold, especially to the scenarios with thin structures or gaps. And that degrades the performance of propagation-based matting methods. In [6], a learning-based approach is presented based on the matting Laplacian matrix. But the errors occurred in the learning process may accumulate quickly. In [7], a support vector machine (SVM) classifier is used for the image matting so as to provide global discriminative information of foreground and background.

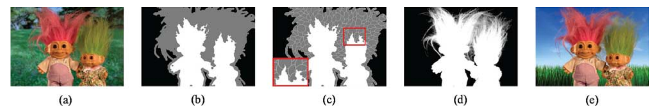

Fig. 1. (a) Original image (b) Trimap (c) Unlabeled region segmentation (d) Alpha matte produced by our approach (e) New composite.

In this letter, a learning-based image matting algorithm is introduced for the estimation of alpha matte. Specifically, we treat alpha matting as a nonlinear regression problem and learn spatially-varying relations between pixel features (like raw intensities, derivative filter response) and alpha values (i.e., features-α models) using support vector regression (SVR) [8]. A training samples selection algorithm is designed and the adaptive parameters are adopted in the SVR to improve its learning accuracy. Compared with the SVM-based matting algorithm [7], the the spatially-varying features-α relations can be effective learned via ε-SVR [9] method.

II. SVR Matting

A. Overview

Our approach requires a trimap as the user input. Firstly, the algorithm segments the unlabeled region into pieces and then learns a features-α model for each piece. To deal with the discontinuities which may appear in the alpha matte, the learning results are smoothed by a matting Laplacian matrix [5] in the post-processing phase. Algorithm 1 shows the flow chart of the proposed approach.

B. Segmentation of Unlabeled Region

In an image, adjacent pixels are usually with similar characteristics. These pixels can share a features-α model to reduce computation. The segmentation of unlabeled region aims to divide the unlabeled region into small pieces and decide the order of training models for them. Because our approach uses previously estimated pixels as training samples, accumulative errors are unavoidable. An appropriate order should make good use

Manuscript received May 03, 2013; revised July 08, 2013; accepted July 19, 2013. Date of publication August 15, 2013; date of current version September 16, 2013. This work was supported in part by the National Natural Science Foundation of China under Grants 61002040, 50635030, and 60932001), the Shenzhen Key Lab for Computer Vision and Pattern Recognition under Grant CXB201104220032A), by grants from the Introduced Innovative R&D Team of Guangdong Province "Image-Guided therapy technology," and by the National Basic Research 973 Program of China under Grant 2010CB732606). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Henry Leung.

The authors are with the Shenzhen Institutes of Advanced Technology, CAS/CUHK, Shenzhen 518055, China (e-mail: qs.zhu@siat.ac.cn; zhan.song@siat.ac.cn; yq.xie@siat.ac.cn; wang.lei@siat.ac.cn) (*Corresponding author*: wang.lei@siat.ac.cn).

Digital Object Identifier 10.1109/LSP.2013.2274874

of labeled pixels in the trimap. The accumulative errors can be reduced by starting from the slim regions in the trimap. The segmentation algorithm is as shown in Algorithm 2.

Segmentation results $P_m (m > 0)$ are circular or fan-shaped pieces as illustrated by Fig. 1(c), where m indicates the order to train models for these pieces. At the beginning of this algorithm, we find the largest connected component of F and ignore the rest. This is because in the trimap, users may add some small scribbles besides the largest one, but the foreground samples in these regions are usually not enough. Besides, $O(\cdot)$ is computed for later use.

Algorithm 1 SVR Matting Algorithm

Input: Image & Trimap

Begin

1. Segmentation of Unlabeled Region;
2. For each segmented unlabeled piece:
3. step1: Training samples selection;
4. step2: Learning a features-α model with SVR;
5. Result smoothing with a matting Laplacian matrix;

End

Algorithm 2 Segmentation of Unlabeled Region

Begin

1. Find the largest connected component F_1 of F;
2. For every pixel $p \in U$, compute $D_{F_l}(p) = \min_{q \in F_l} S(p,q)$, $D_B(p) = \min_{q \in B} S(p,q)$ where $S(\cdot)$ denotes spatial distance;
3. Let $U = \{p_1, p_2, \cdots, p_n\}$. All pixels p in U are sorted by $D(p) = D(F_l)(p) + D_B(p)$ in ascending order;
4. Set $p \in P_0$ for all $p \in U$ and $m = 0$;
5. **for** $t = 1$ to N
6. **if** $p_t \in P_0$;
7. **then** $m = m + 1$, Find all pixels $p \in P_0$ with
8. $S(p_t, p) < r$. Set $p \notin P_0, p \in P_m, O(P_m) = p_t, t++$;
9. **else** $t++$;
10. **end if**
11. **end for**

End

C. Model Training for an Unlabeled Piece

1) Support Vector Regression: SVR is an extension of SVM for regression problems. Like SVM, SVR is also based on the maximum margin separation and kernel method [9]. Given a training set $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \subset \chi \times \Re$, where χ denotes the space of the input patterns (e. g. $\chi = \Re^n$), x_i is the input vector and y_i is the output scalar. According to the principle of ε-SVR [10], it aims to find a function $f(x)$ which has ε-deviation at most from the actually obtained targets y_i for all the training data. ε-SVR can be formulated as the following convex optimization problem:

$$\min_{w,b,\xi,\xi^*}(w,\xi,\xi^*) = \min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi + \xi^*)$$

$$\text{s.t.} \begin{cases} y_i - \langle w, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \quad (2)$$

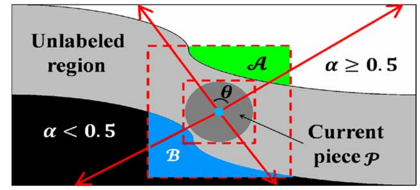

Fig. 2. An illustration of candidate samples collection.

The regression function can be described as: $f(x) = \langle w, x \rangle + b$, where $w \in \chi, b \in \Re$, and $\langle \cdot \rangle$ denotes the inner product in χ. The second term of the objective function in (2) represents the soft margin with C as the penalty parameter, ξ and ξ^* are the slack variables. w is the normal vector of the objective hyperplane that maximize the margin between patterns, and ε is the largest deviation without penalty.

2) Training Samples Selection: For an unlabeled piece, our goal is to obtain training samples which are similar to the unlabeled pixels and represent both local foreground and background characteristics. We first collect candidate samples from the trimap and previously estimated pixels, and then select the best candidate samples.

Candidate samples collection. Given the current unlabeled piece P, two empty sample sets A and B (i.e., $|A| = 0, |B| = 0$), the collection method can be described as following steps (also illustrated by Fig. 2):

• ***Step 1****: Define the smallest rectangle R that can enclose P (like the small red rectangle in Fig. 2).*

• ***Step 2****: For every labeled pixel i newly enclosed by R, if $\alpha_i \ge 0.5$ and $|A| < \lambda|P|$, then $i \in A$; else if $\alpha_i < 0.5$ and $|B| < \lambda|P|$, then $i \in B$.*

• ***Step 3****: If $|A| \ge \lambda|P|$ and $|B| \ge \lambda|P|$, turn to step 4, else increase the width and height of R (by one pixel) and go back to step 2.*

• ***Step 4****: Shot $2n$ rays from $O(P)$ (like the blue pixel in* Fig. 2*) with the length of η. The rays are separated by the equal angle $\theta = \pi/n$. Add labeled pixels on the rays to A if their alpha values ≥ 0.5, to B if their alpha values <0.5.*

In the experiments, the parameters of λ, n, η are set to 1.3, 6, 300 respectively. An expanding rectangle is used to collect nearby samples, so as to avoid the calculation of spatial distances between pixels. The samples on the rays are also collected, because similar pixels may not fall into adjacent labeled regions.

Samples selection. We define $H(p,q)$ to measure the similarity between an unlabeled pixel p in P and a candidate sample q:

$$H(p,q) = E(x_p, x_q) + \rho E((p_x, p_y), (q_x, q_y)) \quad (3)$$

where $E(\cdot)$ denotes the Euclidean distance between two vectors, x_p and x_q are the feature vectors of the pixel p and q, ρ is a weighting factor which is set to 0.8 in the experiments, and the subscripts x and y denote horizontal and vertical coordinates respectively. For every pixel in P, we select l most similar pixels and l most dissimilar pixels as the training samples. The value of l is usually small, and is set to 3 empirically in the experiments. This is because we want to obtain samples similar to P while preserve the comprehensiveness. All the selected pixels for P are treated as the training samples after removal of redundant samples. In addition, a value $V(\cdot)$ is assigned to every pixel in P, which measures the confidence of alpha values estimated with these training samples. The confidence values are computed for later use and can be formulated as:

$$V(p) = \exp\{-w_1 H_{\min}(p) - w_2 D(O(P))\} \tag{4}$$

where $H_{\min}(p) = \min_{q \in A \cup B} H(p,q), w_1$ and w_2 are the parameters for normalization (e.g., $w_1 = 5, w_2 = 0.005$).

D. ε-SVR Training

The feature-α model is trained via ε-SVR with the radial basis function (RBF) kernel as:

$$k(x_i, x_j) = \exp\{-\|x_i - x_j\|^2 / 2\sigma^2\} \tag{5}$$

where σ is the kernel width, which affects the similarity measurement between two feature vectors. According to optimization condition (2), we can get the dual problem of SVR:

$$\begin{aligned} \max_{a, a^* \subset \Re^n} Q(a, a^*) = & \sum_{i=1}^{n} y_i(a_i - a_i^*) - \varepsilon \sum_{i=1}^{n} y_i(a_i - a_i^*) \\ & - \frac{1}{2} \sum_{i,j=1}^{n} (a_i - a_i^*)(a_j - a_j^*) k(x_i, x_j) \\ \text{s.t.} & \begin{cases} \sum_{i=1}^{n}(a_i - a_i^*) = 0 \\ 0 \le a_i, a_i^* \le C, i = 1, 2, \cdots, n \end{cases} \end{aligned} \tag{6}$$

Therefore, we can obtain the regression estimation expression of SVR as:

$$f(x) = \sum_{i=1}^{n} (a_i - a_i^*) k(x_i, x_j) + b \tag{7}$$

Inappropriate selection of parameter σ, and C and ε in ε-SVR, may cause underfitting or overfitting problems [11]. In the following, we will show how these parameters can be selected adaptively.

Let the training sample set be T, and M is a set contains the Euclidean distance between every two features vectors in $P \cup T$, and the variance of M is σ^2. According to [11], the penalty parameter C can be computed as:

$$C = \max(|\bar{\alpha} + 3\tau|, |\bar{\alpha} - 3\tau|) \tag{8}$$

where $\bar{\alpha}$ and τ are the mean and standard deviation of the alpha values of T. As the alpha values range from 0 to 1 and ε affects the precision of the model, we set

$$\varepsilon = 0.05 + \min(0.05, \tau\sqrt{\ln|T| / |T|}) \tag{9}$$

which is a combination of the method in [11] and our specific problem. With these calculated parameters, the feature-α model can be trained. After that, the alpha values for P can be estimated by the regression function (7).

E. Result Smoothing

Although nearby pixels share a feature-α model, roughness may still arise in smooth regions. This is partly because spatial information is not involved in the training process. So after learning the alpha values for all unlabeled pixels, a post-processing method is implemented as did in [2]–[4]. By combining the learning result $\tilde{\alpha}$ with the matting Laplacian matrix L [5] that can be treated as a smoothness term, the final alpha matte can be obtained by minimizing the following function:

$$\mathcal{J}(\alpha) = \alpha^T L \alpha + \varphi(\alpha - \tilde{\alpha})^T \gamma (\alpha - \tilde{\alpha}) + \omega(\alpha - \tilde{\alpha})^T \Gamma (\alpha - \tilde{\alpha}) \tag{10}$$

where φ and ω are the parameters for weighting and normalization. φ is relative large (e.g., 100) while ω is small (e.g., 0.1). α and $\tilde{\alpha}$ are treated as $N \times 1$ vectors, where N is the number of unlabeled pixels in the trimap. γ and Γ are $N \times N$ diagonal matrices. Diagonal elements in γ are one for labeled pixels in the trimap and zero for the others. Diagonal elements in Γ are the confidence values $V(\cdot)$ for unlabeled pixels in the trimap and zero for the others. Finally, with the estimated alpha matte, we can further use a closed-form method in [5] to estimate the foreground color.

III. Experiments and Evaluations

A. Implementation

The proposed approach is implemented in Matlab with LIBSVM (implemented in C++) [12] for ε-SVR. For the pixel features, we use the gradient and raw intensity of every color channel in the image. The algorithm runs on a 3.0 GHz CPU. It takes 75 seconds for an image (with the resolution of 800×600 pixels) in [13] on average. The time spent in LIBSVM is about 5 seconds per image.

B. Quantitative Evaluation

We compare the alpha matte produced by the proposed approach with that of the state-of-the-art matting algorithms on the matting benchmark provided by Rhemann *et al.* [13]. The online benchmark has 8 natural test images and 3 different trimaps for each test image. Results of many recent image matting algorithms (including the proposed one) are available on website of the benchmark (*www.alphamatting.com*).

For every approach, the benchmark uses ground truth alpha mattes to evaluate the 24 results (8 test images and 3 trimaps or each test image in the benchmark). Fig. 3 lists the average mean squared errors (MSE) and sum of absolute differences (SAD) of the 24 results by different approaches. Specifically, closed-form matting uses pure propagation-based technique. Segmentation-based [14] and shared matting [3] methods are combined by the sampling-based and propagation-based techniques. The average MSE by our approach is smaller than other approaches, while the average SAD is just larger than that of share matting method. However, share matting uses trimap expansion [3] as a preprocessing to reduce the unknown pixels. So, we also integrate the trimap expansion into our approach and the results are indicated by 'our*' in Fig. 3. In comparison, our methods can outperform in both metrics.

The benchmark can rank the approaches with respect to four different error metrics: SAD, MSE, gradient and connectivity. We list the overall ranks in Table I. The results show that our approach can outperform in three metrics. Generally, approaches combining sampling and propagation-based techniques can produce better results than the pure propagation-based [5], [15] methods. However, pure propagation-based approaches perform better on the fourth metric. This is mainly because the Laplacian matrices of the propagation-based approaches concentrate on the neighboring relations and thus can preserve

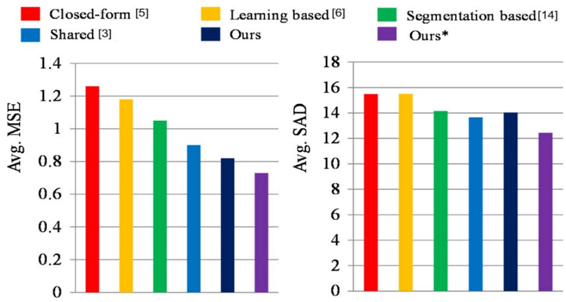

Fig. 3. An illustration of candidate samples collection.

TABLE I
OVERALL RANKS OF THE TOP PERFORMING APPROACHES ON THE BENCHMARK OF [13] WITH RESPECT TO FOUR METRICS

Algorithm	SAD	MSE	Gradient	Connectivity
Our approach	**4.3**	**4**	**4.4**	7.1
Shared [3]	5	5.5	4.8	10.5
Global sampling [4]	5.8	4.9	5	9.2
Segmentation-based[14]	6.3	6.3	4.8	9.1
Improved color [2]	6.7	6.5	5.1	7.3
Learning-based [6]	8.3	8.5	9.4	8.3
Closed-form [5]	8.5	8.7	9.4	**4.5**
Shared (Real Time)[3]	8.4	9.3	11.2	13
Large kernel [15]	10.9	10	10.7	6.6
Robust [1]	11.8	11.2	9.8	12.9

TABLE II
AVERAGE MSE AND SAD ERROR IN COMPARISON WITH THE METHOD IN [6]

Metric	Method in [6]	Our approach
Avg. MSE	1.18	**0.96**
Avg. SAD	15.51	**14.15**

the connectivity of the foreground objects. Table II shows the comparison with another learning-based approach in [6]. From the results, we can see that better matting accuracy can be achieved by the proposed method.

C. Quantitative Evaluation

Fig. 4 shows some cropped images in the benchmark [13] and alpha mattes produced by different approaches. In the first image of Fig. 4(a), we can see that the closed-from and learning-based matting methods fail in the gaps of the foreground because of the pure propagation-based style, while our approach can preserve the details. For the second image in Fig. 4(a), the foreground is a plastic bag. The closed-form matting fails due to the rope (indicated by the red arrow in the image) violating the color line model [5]. The learning-based and our matting approach can reveal the transparency. For the two images in Fig. 4(b), the background is complex and colors of the dolls hair and leaves are close to that of the background. The color ambiguity poses a great challenge for all of the three algorithms. But the proposed approach can still resolve the ambiguity to some extent and achieve better matting quality, which demonstrates the advantages of the learning technique and effective sampling method.

IV. CONCLUSION

This letter presents a novel image matting method which treats the alpha matting as a regression problem. In the algorithm, the ε-SVR method is introduced to learn the spatially-varying relations between pixel features and alpha values. We also use effective training samples selection method and

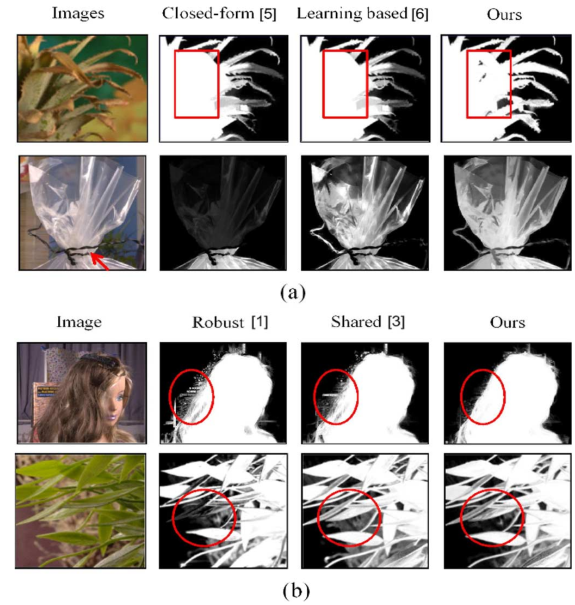

Fig. 4. Visual comparison of various matting algorithms on benchmark [13].

adaptive parameters selection mechanism for SVR to improve the learning accuracy. Various experiments on a public benchmark and comparisons with recent algorithms are used to demonstrate its advantages in matting accuracy. Future work can address how to improve its robustness under complex scenarios, especially to the image regions with color ambiguities. Another interesting work is to investigate some other regression techniques instead of SVR algorithms.

REFERENCES

[1] J. Wang and M. F. Cohen, "Optimized color sampling for robust matting," in *CVPR*, 2007, pp. 17–22.
[2] C. Rhemann, C. Rother, and M. Gelautz, "Improving color modeling for alpha matting," in *BMVC*, 2008, pp. 1155–1164.
[3] E. S. L. Gastal and M. M. Oliveira, "Shared sampling for real-time alpha matting," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 575–584, 2010.
[4] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A global sampling method for alpha matting," in *CVPR*, 2011, pp. 2049–2056.
[5] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, no. 2, pp. 61–68, 2008.
[6] Y. Zheng and C. Kambhamettu, "Learning based digital matting," in *ICCV*, 2009, pp. 889–896.
[7] T. Hosaka, T. Kobayashi, and N. Otsu, "Image matting based on local color discrimination by SVM," *Pattern Recognit. Lett.*, vol. 30, pp. 1253–1263, 2009.
[8] Z. Zhang, Q. Zhu, and Y. Xie, "Learning based alpha matting using support vector regression," in *ICIP*, 2012, pp. 2109–2112.
[9] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.
[10] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.
[11] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Netw.*, vol. 17, pp. 113–126, 2004.
[12] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[13] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, "A perceptually motivated online benchmark for image matting," in *CVPR*, 2009, pp. 1826–1833.
[14] C. Rhemann, C. Rother, P. Kohli, and M. Gelautz, "A spatially varying PSF-based prior for alpha matting," in *CVPR*, 2010, pp. 2149–2156.
[15] K. He, J. Sun, and X. Tang, "Fast matting using large kernel matting Laplacian matrices," in *CVPR*, 2010, pp. 2165–2172.