ITERATIVE TRANSDUCTIVE LEARNING FOR ALPHA MATTING

Bei He, Guijin Wang, Chenbo Shi, Xuanwu Yin, Bo Liu, Xinggang Lin

Department of Electronic Engineering, Tsinghua University, China

ABSTRACT

In this paper, we propose a matting algorithm based on iterative transductive learning (for short: ITM). To avoid over-smooth results of recent methods, we introduce the influence of unlabeled regions as well as the consistency of neighboring pixels to re-design the optimization for alpha matting. A novel asymmetric Laplacian matrix is also proposed to further relieve the over-smoothness. To optimize the matting problem, we adjust the constrain coefficients between the initialized alpha matte and the asymmetric Laplacian matrix iteratively to achieve accurate alpha mattes. Consequently, during the iteration, high confidence pixels maintain their refined alpha values, whereas low confidence ones are updated by their neighbors gradually. Experimental results demonstrate that our algorithm is more precise than many state-of-the-art methods in terms of the accuracy.

***Index Terms*—** image matting, transductive learning, asymmetric Laplacian matrix, iterative optimization

1. INTRODUCTION

Alpha matting techniques have attracted increasing interest recently, which are widely applied in separating layers, replacing the background and facilitating editing applications [1]. Generally, the color value $\mathbf{C}_i$ can be modeled as a linear combination of the foreground component $\mathbf{F}_i$ and the background component $\mathbf{B}_i$:

$$\mathbf{C}_i = \alpha_i \mathbf{F}_i + (1-\alpha_i)\mathbf{B}_i, \tag{1}$$

where $\alpha_i \in [0,1]$ denotes the opacity of the foreground. As the matting problem is inherently ill-posed, user interaction is usually required to constrain the problem, such as the trimap or the scribble.

Since user interaction provides labeled (definite foreground $\mathcal{R}_f$ and background $\mathcal{R}_b$) and unlabeled (the unknown $\mathcal{R}_u$) pixels, the matting problem can be treated as a learning-based technique. As a better classifier can be built by integrating unlabeled pixels into labeled ones [2], transductive learning was employed for alpha matting [3–9]. Levin [3] considered the linear combination relationship among neighboring pixels and designed the Laplacian matrix. Based on that, Zheng [4] and Xiang [5] formulated the relationship as a non-linear model and obtained more accurate results. However, the Laplacian matrix above was truncated and might lead to an over-smooth alpha matte, e.g., in the hollowed-out regions. Lee [6] and Chen [7] alleviated the problem by enlarging the neighborhood from local to global, at the expense of high time and space complexity. To deal with that, He [8] reduced the computational cost to solve the alpha matte; however, constructing the Laplacian matrix also confronted heavy time complexity. In contrast, Wang [9] introduced the Markov Random Field model and applied unlabeled pixels to relieve the over-smoothness. The disadvantage of this approach, however, is that the coefficients for balancing the initialized alpha matte and the Laplacian matrix cannot be determined easily.

Focusing on the problems illustrated above, we propose an alpha matting algorithm based on iterative transductive learning. Our contributions are twofold. First, we re-designed the optimization function for transductive matting, considering the consistency of neighboring pixels as well as the influence of labeled and unlabeled regions. Unlike existing methods, an asymmetric Laplacian matrix was then introduced. Those improvements lead to the relief of the over-smoothness, so the hollowed-out regions in the image can be calculated correctly. Second, we refined matting results iteratively, so the constrain coefficients between the initialized alpha matte and the Laplacian matrix were determined adaptively. During the iteration, precise results were achieved. Visual and quantitative experiments demonstrate that our approach is more accurate than other state-of-the-art methods.

2. PROBLEM FORMULATION

In the view of transductive learning [10], the matting problem is solved by labeling each pixel with the probability of the foreground. Thus, the alpha matte can be optimized by,

$$\begin{array}{ll} \min & \boldsymbol{\alpha}^T \cdot \mathbf{L} \cdot \boldsymbol{\alpha} \\ s.t. & \mathbf{I}_k \boldsymbol{\alpha} = \boldsymbol{\alpha}_k^* \end{array}, \tag{2}$$

where $\boldsymbol{\alpha}=[\alpha_1, \alpha_2, ..., \alpha_P]^T$ refer to alpha values of all pixels. The number of pixels in the image is recorded as P. $\boldsymbol{\alpha}_k^*$ denotes the alpha matte marked by user interaction. The i^{th} element in $\boldsymbol{\alpha}_k^*$ takes the value 1 if $i \in \mathcal{R}_f$, and the other elements are zeros. The diagonal matrix $\mathbf{I}_k$ is used to constrain

This paper is partially sponsored by NSFC (No. 61271390) and NSFC (No. 61132007).

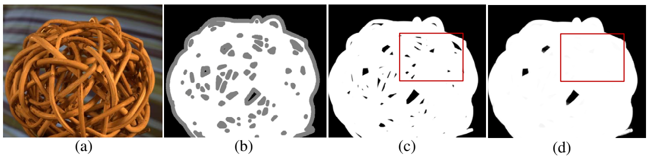

Fig. 1. The color image (a), trimap (b), ground truth (c) and matting results of [3–5] (d).

labeled pixels. The i^{th} diagonal element in $\mathbf{I}_k$ takes the value 1 if $i\in\mathcal{R}_f\cup\mathcal{R}_b$, and the other elements are zeros. $\mathbf{L}$ refers to the Laplacian matrix, where the elements of that matrix are used to evaluate the similarity between pixels. According to the Lagrange multiplier method, the objective function is optimized as,

$$\min \quad \boldsymbol{\alpha}^T\mathbf{L}\boldsymbol{\alpha} + \lambda_k(\boldsymbol{\alpha}-\boldsymbol{\alpha}_k^*)^T\mathbf{I}_k(\boldsymbol{\alpha}-\boldsymbol{\alpha}_k^*), \tag{3}$$

where λ_k refers to the Lagrange coefficients of labeled pixels. Since not every two pixels are linked [3–5], over-smooth matting results are usually induced, as illustrated in Fig.1.

3. ITERATIVE TRANSDUCTIVE MATTING

Different from the related work [3–5], we introduce the unlabeled pixels as the constraint to solve Eq.3, since good initialization of unlabeled pixels helps to estimate the alpha matte accurately. Considering not all unlabeled pixels can obtain precise estimations, we should calculate the corresponding confidence values to constrain them. Therefore, we take unlabeled pixels into account according to the following steps. First, alpha values of unlabeled pixels are estimated. Second, the corresponding confidence value is assigned to each estimation. Third, alpha and confidence values are combined to constrain Eq.3. Details of how to initialize alpha and confidence values of unlabeled pixels were described in our recent work [11, 12]. Hence, our optimization function is re-written by,

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^T\mathbf{L}\boldsymbol{\alpha} + \lambda_k(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{I}_k(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)+ \\ & \lambda_u(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{c}^*(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*), \end{aligned} \tag{4}$$

where the diagonal matrix $\mathbf{c}^*$ refers to the confidence matte. The i^{th} element takes the value $c_i^*\in[0,1]$ if $i\in\mathcal{R}_u$, and the others are zeros. $\boldsymbol{\alpha}^*$ denotes the initialized alpha matte after user interaction (for labeled pixels) and estimations (for unlabeled ones). λ_u represents the Lagrange coefficients of unlabeled pixels. It is noted that the objective function is closely related to the MRF model [9], which is also used to solve matting problems [11–14]. The Laplacian matrix and initialized alpha matte correspond to the "*smooth*" and "*data*" terms respectively. Then alpha values can be estimated as:

$$\boldsymbol{\alpha} = (\mathbf{L}+\lambda_k\mathbf{I}_k + \lambda_u\mathbf{c}^*)^{-1}(\lambda_k\mathbf{I}_k + \lambda_u\mathbf{c}^*)\boldsymbol{\alpha}^*. \tag{5}$$

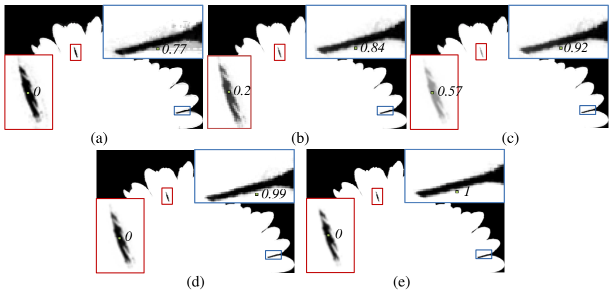

Fig. 2. Matting results using large (a), middle (b) and small (c) constrain coefficients [9] and ours (d). (e) refers to the ground truth. Alpha values of some representative pixels are marked in the figure.

To solve the Eq.5 more accurately and adaptively, we put forward two more contributions compared with the traditional work [3–5, 9]. One is the construction of the asymmetric Laplacian matrix; the other is the approach to set the constrain coefficients adaptively.

In the Laplacian matrix, neighbors of low feature similarity indicate low influence on the current pixel and should be excluded. Moreover, due to the symmetry of the neighborhood, there exist many redundant elements. To address this problem, an asymmetric Laplacian matrix is introduced, which helps to relieve the over-smoothness further.

As mentioned in [9], the constrain coefficients, i.e., λ_k and λ_u, cannot be determined easily. On one hand, fixing them with large values causes the fact that the alpha matte will solely depend on initialization. Since the initialization is obtained pixel-by-pixel, the affinity of the alpha matte cannot be guaranteed. On the other hand, fixing those values small results in over-smooth matting results. As shown in Fig.2(a)-(c), increasing and decreasing constrain coefficients result in non-smooth and over-smooth matting results respectively.

3.1. Construction of the Asymmetric Laplacian Matrix

During the construction of the Laplacian matrix, preserving non-continuous neighboring pixels leads the alpha matting to a local optimal solution [3–5, 9]. Different from the previous work, we preserve the top N neighbors of high feature similarity within the $M\times M$ window size of the neighborhood ($N<M\times M$). The similarity of two pixels i and j is calculated by,

$$SIMI(i,j) = \mathbf{C}_i^T\mathbf{C}_j/\|\mathbf{C}_i\|/\|\mathbf{C}_j\|. \tag{6}$$

Then the sparse Laplacian matrix can be constructed according to Zheng [4]. As illustrated in Fig.3, our Laplacian matrix is asymmetric, because $\mathbf{L}(i,j)$ and $\mathbf{L}(j,i)$ are not calculated repeatedly. Assuming the number of unlabeled pixels is n, nearly $n^2/2$ redundant elements of the Laplacian matrix are eliminated.

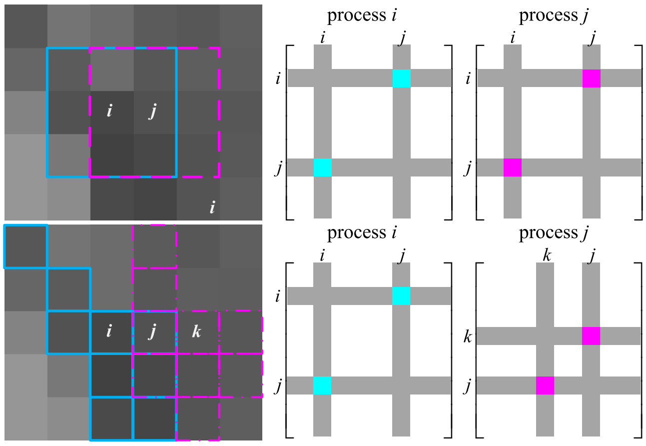

Fig. 3. Construction of the Laplacian matrix in previous work (top) [3–5, 9] and ours (bottom). In our matrix, j is a preserved pixel in i's neighborhood, but i is not preserved while processing j.

3.2. Iterative Optimization

We solve the objective function in Eq.5 by adjusting the constrain coefficients iteratively, which balances the initialized alpha matte and the Laplacian matrix adaptively. For labeled pixels, constrain coefficients are fixed to $+\infty$ since their estimation errors are unbearable. To quantize the coefficients, we set λ_k as a constant large number. For unlabeled pixels, the constrain coefficients are adjusted from small to large iteratively. In the m^{th} iteration, $\lambda_u^{(m)}$ takes the value $\eta_u{\cdot}10^m$, where η_u denotes a basic number. Consequently, the alpha matte is updated as:

$$\begin{aligned}\boldsymbol{\alpha}^{(m)} &= (\mathbf{L}+\lambda_k\mathbf{I}_k+\lambda_u^{(m)}\mathbf{c}^{(m-1)})^{-1}\\ &(\lambda_k\mathbf{I}_k+\lambda_u^{(m)}\mathbf{c}^{(m-1)})\boldsymbol{\alpha}^{(m-1)},\end{aligned} \tag{7}$$

where $\boldsymbol{\alpha}^{(0)}$ and $\mathbf{c}^{(0)}$ refer to $\boldsymbol{\alpha}^*$ and $\mathbf{c}^*$. We update the confidence matte via referring to temporal variation of alpha mattes, which is defined by,

$$c_i^{(m)} = \begin{cases} \min(c_i^{(m-1)}+\Delta c, 1), \ \mathit{if}\ |\alpha_i^{(m)}-\alpha_i^{(m-1)}|\leq T_c \\ \max(c_i^{(m-1)}-\Delta c, 0), \ \mathit{otherwise}, \end{cases} \tag{8}$$

where T_c refers to the threshold of temporal variation and Δc denotes a constant increment. The iteration continues until the norm of the difference between $\boldsymbol{\alpha}^{(m)}$ and $\boldsymbol{\alpha}^{(m-1)}$ is less than the threshold T_n. In the beginning, the coefficients of unlabeled pixels are set to small values, so the pixels can be updated by their neighbors according to the restriction of the asymmetric Laplacian matrix. During the iteration, the coefficients become larger gradually. As a result, higher confidence pixels maintain their refined estimations, while those with lower confidence continue to be corrected by their neighbors.

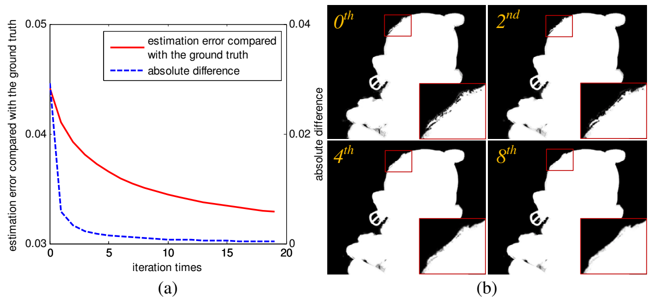

Fig. 4. The norm of the difference between temporal neighboring alpha mattes and the estimation error versus the iteration times (a). Intermediate matting results during the iteration are shown in (b).

4. EXPERIMENTS AND DISCUSSIONS

To verify the performance of our iterative transductive matting algorithm, we implemented it in Matlab and ran it on the Intel Core II CPU with 2.0GHz Dual. We determine parameter values by the cross-validation method. In our approach, they were fixed as follows: N=9, M=11, λ_k=10^6, η_u=10^{-3}, Δc=0.02, T_c=0.05 and T_n=10^{-4}. We utilized the standard test dataset provided by [15], which includes 27 groups of color images, trimaps and true alpha mattes. The results of our iterative transductive matting method (for short: ITM) were compared with those of learning-based matting [4] (for short: LM) and transductive matting [9] (for short: TM).

An example of our iterative optimization is illustrated in Fig.4. With the increasing number of iterations, the estimation error of alpha values and difference of temporal neighboring alpha mattes decrease consistently. In the 0^{th} iteration, the initialized alpha matte is non-smooth due to the pixel-by-pixel estimations [1]. During iterative optimization, high confidence pixels maintain their refined alpha values, while low confidence pixels are corrected by their neighbors gradually. Consequently, the alpha matte is estimated more precisely.

Fig.5 shows visual matting results of LM, TM and ITM. For the "GT04" and "GT05" images, we not only achieved smoother results by ITM, but also avoided plunging into a local optimal solution, e.g., in over-smooth regions. This is attributed to the adaptive constrain coefficients setting between the initialized alpha matte and the asymmetric Laplacian matrix. For the "GT16" image, imprecise alpha values were introduced in fuzzy regions in the case of TM, due to the direct application of unlabeled pixels. In contrast, comparably good results as those of LM were acheived by our method ITM. Due to construction of our Laplacian matrix and iterative optimization of the alpha matte, low confidence pixels can be corrected by their neighbors while high confidence pixels maintain their estimations. For the "GT21" image, TM and our ITM perform better than LM, e.g., in hollowed-out regions, due to the introduction of initialized alpha matte. In our

Fig. 5. The color image, ground truth, matting results of LM [4], TM [9] and ITM (from top to bottom respectively).

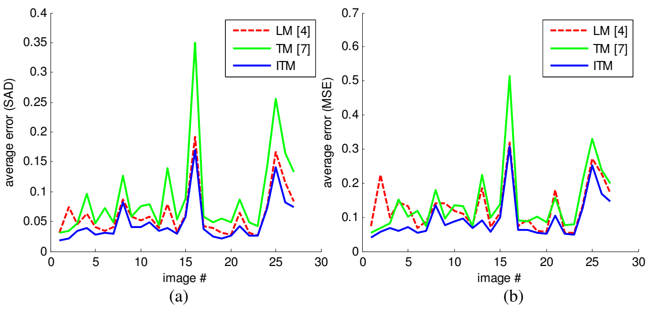

Fig. 6. The estimation errors of LM [4], TM [9] and ITM (y-axis) versus the image index (x-axis). The SAD and MSE are evaluated in (a)-(b) respectively.

approach, smoother alpha values were achieved by excluding the influence of the neighboring pixels with low similarity and balancing the constrain coefficients for pixels adaptively.

The quantitative estimation errors between LM, TM and our ITM are plotted in Fig.6. It can be figured out that ITM outperforms LM and TM, especially for images with hollowed-out regions (such as “GT02”, “GT13” and so on). The website [16] provides corresponding ranks of all state-of-the-art methods in SAD (Sum of Absolute Difference) and MSE (Mean Squared Error) evaluation. As declared in Tab.1, we ranked 1 and 4 in SAD and MSE (in brackets) respectively among top 23 matting algorithms. In experiments, our approach took $18s$ for an image (about 640×480) averagely.

Table 1. Average rankings and total ranks (in brackets) in SAD and MSE evaluation. (on the date $1/15/2013$)

ALGORITHM	SAD	MSE
Learning-based Matting [4]	10.3 (11^{th})	10.5 (10^{th})
Shared Matting [13]	6.3 (4^{rd})	7 (5^{th})
Global Sampling Matting [14]	7.5 (5^{th})	6.5 (3^{rd})
Weighted Color and Texture Matting [17]	5.8 (3^{rd})	6.3 (2^{nd})
SVR Matting [18]	5.3 (2^{nd})	5 (1^{st})
Iterative Transductive Matting	**5 ($\mathbf{1^{st}}$)**	**6.8 ($\mathbf{4^{th}}$)**

To sum up, more accurate matting results were achieved by our algorithm both visually and quantitatively.

5. CONCLUSION

This paper describes a matting approach based on iterative transductive learning. The objective function was re-designed by considering the consistency of neighboring pixels in addition to the influence of labeled and unlabeled regions. An asymmetric Laplacian matrix was then adopted, so the local optimal solution could be avoided. Those improvements relieved over-smooth results in hollowed-out regions. Additionally, the alpha matte was refined iteratively, where high confidence pixels maintain their initialized values whereas low confidence ones are corrected by their neighbors adaptively. As a result, precise and smooth alpha values were obtained during the iteration. Both visual and quantitative results have shown that our algorithm gives rise to more accurate estimation than many state-of-the-art methods.

References

[1] Jue Wang and Michael F Cohen, “Image and video matting: A survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 2, pp. 97–175, 2007.

[2] Xiaojin Zhu, “Semi-supervised learning literature survey,” Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005, http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

[3] A. Levin, D. Lischinski, and Y. Weiss, “A closed form solution to natural image matting,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. IEEE, 2006, vol. 1, pp. 61–68.

[4] Y. Zheng and C. Kambhamettu, “Learning based digital matting,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 889–896.

[5] Shiming Xiang, Feiping Nie, and Changshui Zhang, “Semi-supervised classification via local spline regression,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 11, pp. 2039–2053, 2010.

[6] P. Lee and Y. Wu, “Nonlocal matting,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2193–2200.

[7] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang, “KNN matting,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 869–876.

[8] Kaiming He, Jian Sun, and Xiaoou Tang, "Fast matting using large kernel matting laplacian matrices," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2165–2172.

[9] J. Wang, "Image matting with transductive inference," *Computer Vision/Computer Graphics Collaboration Techniques*, pp. 239–250, 2011.

[10] O. Duchenne, J.Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne, "Segmentation by transduction," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[11] Bei He, Guijin Wang, Zhiwei Ruan, Xuanwu Yin, Xiaokang Pei, and Xinggang Lin, "Local matting based on sample-pair propagation and iterative refinement," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 285–288.

[12] B He, G. Wang, C. Shi, X. Yin, B. Liu, and X. Lin, "High-accuracy and quick matting based on sample-pair refinement and local optimization," *submitted to IEICE*, 2012.

[13] E.S.L. Gastal and M.M. Oliveira, "Shared sampling for real-time alpha matting," in *Computer Graphics Forum*. Wiley Online Library, 2010, vol. 29, pp. 575–584.

[14] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A global sampling method for alpha matting," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2049–2056.

[15] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, "A perceptually motivated online benchmark for image matting," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1826–1833.

[16] "Alpha matting benchmark," `http://www.alphamatting.com`, 2009.

[17] E. Shahrian and D. Rajan, "Weighted color and texture sample selection for image matting," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 718–725.

[18] Zhanpeng Zhang, Qingsong Zhu, and Yaoqin Xie, "Learning based alpha matting using support vector regression," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2109–2112.