It is the paper published as:

**Author:** J. Gao  
**Title:** Image matting via local tangent space alignment  
**Year:** 2011  
**Conference Name:** IEEE Digital Image Computing Techniques and Applications (DICTA) Conference  
**Conference Location:** USA  
**Publisher:** IEEE  
**Date:** 6-8 December 2011  

**Abstract:** Image matting refers to the problem of accurately extracting foreground objects in images and video. The most recent work [13] in natural image matting relies on the local smoothness assumptions on foreground and background colors on which a cost function is established. The closed-form solution has been derived based on certain degree of user inputs. In this paper, we present a framework of formulating new cost function from the manifold learning perspective based on the so called Local Tangential Space Alignment algorithm [25] where the local smoothness assumptions have been replaced by implicit manifold structure defined in local color spaces. We illustrate our new algorithm using the standard benchmark images and very comparable results have been obtained.

**DOI:** [http://dx.doi.org/10.1109/DICTA.2011.109](http://dx.doi.org/10.1109/DICTA.2011.109)  
**Author Address:** jbgao@csu.edu.au  
**CRO identification number:** 31098
Image Matting via Local Tangent Space Alignment

Junbin Gao

Abstract—Image matting refers to the problem of accurately extracting foreground objects in images and video. The most recent work [13] in natural image matting relies on the local smoothness assumptions on foreground and background colors on which a cost function is established. The closed-form solution has been derived based on certain degree of user inputs. In this paper, we present a framework of formulating new cost function from the manifold learning perspective based on the so-called Local Tangential Space Alignment algorithm [25] where the local smoothness assumptions have been replaced by implicit manifold structure defined in local color spaces. We illustrate our new algorithm using the standard benchmark images and very comparable results have been obtained.

I. INTRODUCTION

Image matting refers to the problem of softly extracting the foreground object from a single image. Image Matting is important in both computer vision and graphics applications and is a key technique in many image/video editing and film production applications. The approach for digital matting has been extensively studied in the literature. A 2007 survey article [23] provides a comprehensive review of existing image and video matting algorithms and systems, with an emphasis on the advanced techniques that have been recently proposed. Most of existing matting techniques depend on the so-called alpha matte. Mathematically, the alpha matte is based on the following model assumption

\[ I_i = \alpha_i F_i + (1 - \alpha_i) B_i \]  

(1)

where \( \alpha_i \), \( I_i \), \( F_i \) and \( B_i \) are the alpha matte value, image color value, the foreground image color value and the background color value, respectively, at an given pixel \( i \). The alpha matte value \( \alpha_i \) is assumed to be 0 or 1 (hard matte value). However it would be convenient to assume that \( \alpha_i \) lies between 0 and 1 as soft matte.

In a matting problem, given the image pixel information \( I \), for all the pixels, the goal is to estimate \( \alpha_i \), \( F_i \) and \( B_i \) simultaneously. Obviously this is a severely underconstrained problem. Many existing matting algorithms and systems require certain degree of user interaction to extract a good matte. For example, a so-called trimap is usually supplied to matting algorithms or systems, which indicates definite foreground, definite background and unknown regions. To infer the alpha matte value in the regions, Bayesian matting [7] uses a progressively moving window marching inward from the known regions. Bai and Sapiro [4] proposed to calculate alpha matte through the geodesic distance from a pixel to the known regions. Among the propagation-based approaches, the Poisson matting algorithm [21] assumes the foreground and background colors are smooth in a narrow band of unknown pixels, then solves a homogenous Laplacian matrix; An alike algorithm is proposed in [10] based on Random Walks. The Closed-form matting approach [13] was proposed by introducing the matting Laplacian matrix under the assumption that foreground and background colors can be fit with linear models in local windows, which leads to a quadratic cost function in alpha that can be minimized globally. Minimizing the quadratic cost function is equivalent to solving a linear system which is often time-consuming when the image size is large. In a recent work [11], He et al. proposed a fast matting scheme by using large kernel matting Laplacian matrices. By using kernel learning tricks, the authors of [26] proposed a new local learning models based on the semi-supervised learning scheme to improve upon the closed form matting. Segmentation-based matting approach by Rahman et al. [18] further improves a state-of-the art alpha matting by incorporating a new prior which is based on the image formation process. In particular, the prior probability of an alpha matte is modeled as the convolution of a high-resolution binary segmentation with the spatially varying point spread function (PSF) of the camera. To order to improve the low speed of matte computation, Gastal et al. [8] proposed the Shared matting technique for the first time aiming at real-time matting technique for natural images and videos. The technique is based on the observation that, for small neighborhoods, pixels tend to share similar attributes. Review and assessment of more matting techniques can be found at the alpha matting evaluation website http://www.alphamatting.com/.

In this paper, we try to explore the possibility of applying dimensionality reduction algorithms [22], particularly manifold learning techniques like Local Tangential Space Alignment (LTSA) algorithm [25], to the problem of image matting based on the alpha model. Our approach is closely related to the matting Laplacian matrices introduced by Levin et al. [13]. That is, our approach still uses user-specified constraints (such as scribbles or a bounding rectangle) as done in [13], [15], [19] while the assumption used in the matting Laplacian matrices that the foreground (or background) colors in a local window lie on a single line in the RGB color space has been naturally replaced with the smooth manifold assumption which is purely a mathematical assumption in learning smooth manifold. One of innovations offered by our approach is to take natural definition of pixel neighborhood on the color manifold as the neighborhood used in most standard manifold learning algorithms such as LTSA. In other words, we are
aiming to use a revised version of LTSA under a slightly different neighborhood. The idea of using manifold learning or dimensionality reduction approach for image matting offered in this paper will inspire more exploration along the direction, although the first learning based matting method has been proposed in [26].

The paper is organized as follows. Section 2 simply introduces the Local Tangential Space Alignment (LTSA) algorithm [25]. Section 3 is dedicated to formulating the new algorithms based on LTSA. In Section 4, we present several examples of using the new approach of image matting over the benchmark images, see [13]. We make our conclusions in Section 5 with suggestions of further exploring dimensionality reduction algorithm application in image matting.

II. LOCAL TANGENT SPACE ALIGNMENT IN IMAGE COLOR SPACES

To describe the proposed algorithm, throughout the paper we will make use of the following notations: For a given image $I$, denote the pixel by $i$ and its corresponding RGB color vector by $I_i$. As far as the LTSA is concerned, we are considering the dataset $I = \{I_i | i \text{ is pixel}\}$ of RGB color vectors of dimension 3. The LTSA is a method for manifold learning, which can efficiently learn a nonlinear embedding into low-dimensional coordinates from high-dimensional data, and can also reconstruct high-dimensional coordinates from embedding coordinates. The algorithm of LTSA consists of two main steps: (1) Conduct a PCA in the neighborhood of each data point; (2) Globally align the embedding under the local PCA coordinates. In the original LTSA, one of methods to define the neighborhood of a data point, i.e., the RGB color vector in our case, is to use $k$-nearest neighbors according to the distance in the data space, i.e., the color space.

In order to incorporate pixel structures, we define data neighborhood by using their spatial affinity of pixel location. For each pixel $i$, we define the neighborhood of $I_i$ as the RGB vectors $I_{ij}$ with $j$ being in the neighbor of $i$ in terms of a local window $w_i = \{i_1, ..., i_p\}$, e.g., a $3 \times 3$ window with $i$ as the center pixel. For each pixel $i$, define a subset

$$X_i = \{I_{ij} | j \in w_i, j = 1, ..., p\}. \quad (2)$$

Applying the classical PCA [6] over $X_i$, there exists a $Q_i$ of $d$ (chosen to $< 3$) orthonormal columns such that

$$I_{ij} = \bar{I}_i + Q_i \theta_j^{(i)} + \epsilon_j^{(i)} \quad (3)$$

where $\epsilon_j^{(i)} = (E - Q_i Q_i^T)(I_{ij} - \bar{I}_i)$ with the identity matrix $E$ is the reconstruction error, $\theta_j^{(i)}$ is the local coordinates over local tangent space in the color space and $\bar{I}_i$ is the mean color vector. As done in the LTSA algorithm, $Q_i$ can be obtained by computing the best $d$-dimensional affine subspace approximation for the point data in $X_i$. For example, one solution is given by the $d$ principal components of $X_i$.

The purpose of the LTSA algorithm is to find out the global coordinates of each data point $I_i$ ($i = 1, ..., N$ where $N$ is the total number of pixels on the image) in an assumed feature space of dimension $d$. For the purpose of image matting, we aim to reconstruct global matting feature $\alpha_i$ of the local coordinates $\theta_j^{(i)}$ based on the local color information on the manifold defined by local windows. Specifically, we wish for matting values $\alpha_{ij}$ to satisfy the following set of equations, according to local structures determined by the $\theta_j^{(i)}$:

$$\alpha_{ij} = \pi_i + L_i \theta_j^{(i)} + \epsilon_j^{(i)}, \quad j = 1, ..., p; \ i = 1, ..., N$$

where $\pi_i$ is the mean of $\alpha_i$ ($j = 1, ..., p$). Denote $A_i = [\alpha_{i_1}, ..., \alpha_{i_p}], \Theta_i = [\theta_1^{(i)}, ..., \theta_p^{(i)}]$ and $E_i = [\epsilon_1^{(i)}, ..., \epsilon_p^{(i)}]$, then the above linear system can be written in a matrix form,

$$A_i = \frac{1}{p} A_i e e^T + L_i \Theta_i + E_i$$

where $e = (1, ..., 1)^T$ is a $p$ dimensional vector. Similar to LTSA, we seek to find $\alpha_i$ and the local affine transformations $L_i$ to minimize the reconstruction errors $\epsilon_j^{(i)}$, i.e.,

$$\min \sum_i \|E_i\|^2 = \sum_i \|A_i (E - \frac{1}{p} e e^T) - L_i \Theta_i\|^2 \quad (4)$$

Denote $A = [\alpha_1, ..., \alpha_N]$. Note that, in the above optimal problem, $\Theta_i$ is known and $L_i$ is separated to each other. Hence for a fixed $A$, each $L_i$ can be easily solved by $L_i = A_i (E - \frac{1}{p} e e^T) \Theta_i^T$ where $\Theta_i^T$ is the Moor-Penrose pseudo inverse of $\Theta_i$. Thus $E_i = A_i (E - \frac{1}{p} e e^T) (E - \Theta_i^T \Theta_i) = A_i W_i$. Hence

$$\sum_i \|E_i\|^2 = \sum_i \|A_i W_i\|^2$$

Note that the components in $A_i$ are overlapped. Let us introduce the $0$-$1$ selection matrix $S_i$ such that $A_i = AS_i$, then the error expression (4) can be written as

$$\sum_i \|E_i\|^2 = ASWW^T S^T A^T = ABA^T$$

where $B = SWW^T S^T$, $S = [S_1, ..., S_N]$ and $W = \text{diag}(W_1, ..., W_N)$.

We call $B$ the LTSA alignment matrix whose spectral property has been investigated in [24]. To uniquely determine $A$, we may impose constraints that are suitable to our matting problem. The following are some suggestions

1) Scribble Constraint I: Let $\Omega$ be a subset of pixels and $\Gamma_\Omega$ be the operator mapping $A$ to matting given alpha matting values $A_0$ over $\Omega$, then the matting problem is to minimize the following objective function with respect to $A$ such that $\Gamma_\Omega(A) = A_0$

$$\min_{\Gamma_\Omega(A) = A_0} \sum_i \|E_i\|^2 = ABA^T \quad (5)$$

2) Scribble Constraint II: Further we can seek $A$ satisfying the matting constraints by

$$\min_{0 \leq A \leq 1} \sum_i \|E_i\|^2 = ABA^T \quad (6)$$
3) Scribble Smoothing [13]: Instead of using hard scribble constraints, we seek for \( \tilde{A} \) by smoothing matting value by
\[
\min_{\tilde{A}} \text{tr}(ABAT) + \lambda(A - A_{\Omega})D_{\Omega}(A^T - A_{\Omega}^T) \tag{7}
\]
where \( \lambda \) is some large number, \( D_{\Omega} \) is a diagonal matrix whose diagonal elements are one for constrained pixels and zero for all other pixels, and \( A_{\Omega} \) is the vector containing the specified alpha values for the constrained pixels and zero for all other pixels.

To further clarify the relationship between the approach proposed in this paper and the matting Laplacian method, we would like to make the following comments:

Remark 1. The construction of matting Laplacian matrix in [13] was proposed under a color line assumption in 3D color space while the LTSA alignment matrix is purely based on subspace assumption. Thus the new approach can be easily extended to the scenarios beyond the 3D color space. For example, the data point could be constructed from a super pixel, say a 2 x 2 pixel giving a 12 dimensional vector. Then the local PCA can be conducted in 12 dimensional space rather than simply 3 dimension color space. Or further we can augment the 3D color vector by other feature information for example texture information etc. One can not rely on the color line assumption for such extensions.

Remark 2. The matting Laplacian matrix in [13] relies on a tiny parameter \( \epsilon \) for the inner regularization. See the paper for more details. Our formulation of the LTSA alignment matrix does not depend on any tunable parameters at all.

Remark 3. The three formulations we just proposed are generalized cases in the LTSA algorithm where the constraint \( \|A\| = 1 \) is applied. Instead of single alpha matting value \( \alpha_i \) at each pixel, we can seek for a multidimensional global embedding by replacing row vector \( A = [\alpha_1, ..., \alpha_N] \) with a \( K \times N \) matrix and consider the following objective function under certain constraints
\[
\min_{\text{constraints of } A} \text{tr}(ABAT)
\]
This formulation is similar to the approach used in [14] in which matting components are constructed through learning sparsified combination of eigenvectors of the matting Laplacian matrix. We would like to directly learn multiple matting components by imposing sparse constraint conditions such as structured regularization recently introduced by Jenatton et al. [12]. For example, use the sparsity-inducing norm for each row of \( A \). Some preliminary results have been reported [3].

Remark 4. The neighbor dataset \( X_i \) in (2) is usually defined by a local window of size \( m \) at pixel \( i \). When pixel \( i \) is near to or on the boundary of the image, some care shall be taken. In this paper, the neighbor of such a pixel is defined by using symmetrical padding along the image boundary.

III. ALGORITHM FORMULATION BASED ON LTSA
A. Computing the LTSA Alignment Matrix \( B \)

All the three suggested formulations of solving image matting problem in the last section depend on the construction of the LTSA alignment matrix \( B \). Relying on the special structures of \( W_i \)'s, the authors of [25] proposed a simple method as outlined below:

Consider the local coordinates matrix \( \Theta_i = [\theta_1^{(i)}, ..., \theta_p^{(i)}] \) as defined by the PCA decomposition (3). Let \( \Theta_i^T = H_iR_i \) be the QR decomposition of \( \Theta_i^T \) [9], then it has been proved that
\[
W_i = E - \frac{1}{p}ee^T - H_iH_i^T = E - G_iG_i^T
\]
where \( G_i = [\sqrt{\rho_i}H_i] \). Then the alignment matrix \( B \) can be calculated as an iterative procedure defined by
- Start with \( B = 0 \),
- For \( i = 1, 2, ..., N \), repeatedly calculate
  \[
  B(w_i, w_i) \leftarrow B(w_i, w_i) + E - G_iG_i^T
  \]
  where \( w_i = \{i_1, ..., i_p\} \) is all the neighbors of pixel \( i \) including itself.

B. Suggested Optimization Algorithms

It is easy to solve the optimal problem (7) for the scribble smoothing as it is an unconstrained quadratic programming. To reinforce the user scribble conditions \( A_{\Omega} \) over pixels \( \Omega \), the algorithm in [13] takes a larger regularizer \( \lambda = 100 \). Thus the objective function can be optimized by solving a sparse linear system:
\[
(B + \lambda D_{\Omega})A = \lambda A_{\Omega} \tag{8}
\]

Although the optimal problem (5) is a quadratic programming with linear equality constraints, it is easy to convert it to an unconstrained problem. Without loss of generality, suppose that \( A \) has been ordered as \( A = (A_{1\Omega}, A_{2\Omega}) \) where \( A_{1\Omega} = A_0 \) is the user given scribbles. Accordingly we decompose the LTSA alignment matrix into
\[
B = \begin{pmatrix}
B_{12} & B_{12} \\
B_{21} & B_{11}
\end{pmatrix}
\]
where \( B_{12} = B_{21} \). Taking the above decomposition into the objective function results in the following new optimal problem with respect to \( A_{1\Omega} \),
\[
\min_{A_{1\Omega}} A_{1\Omega} B_{12} A_{1\Omega}^T + 2A_{1\Omega}B_{21}A_{1\Omega}^T
\]
Clearly this can be solved by
\[
A_{1\Omega} = A_0B_{21}B_{12}^{-1} \tag{9}
\]

As the alpha matte values in the solution given either by (8) or (9) may be outside the range \([0, 1]\), in the implementation of algorithm presented in [13], a post-procedure is conducted by truncating the solution to \([0, 1]\), i.e., any value less than 0 is set to 0 and any value larger than 1 set to 1. In practice, without any regularization, (9) does not always give satisfactory results. Thus in experiments, we ignore this solution.

The above post-procedure of truncation is not needed in the formulation of the scribble constraint II (6) as the constraint \( 0 \leq A \leq 1 \) is explicitly enforced. However the optimal problem is a quadratic programming with a set of linear constraints. One of approaches to solve a quadratic programming
problem is to use an interior point method that uses Newton-like iterations to find a solution of the Karush-Kuhn-Tucker conditions of the primal and dual problems. The computational complexity is polynomial time of the size of the matrix $B$. It is not possible to get accurate solution even for a medium size image.

Like handling the optimal problem (5), we can remove the equality constraints $\Gamma_0(A) = A_0$ and without loss of generality consider the following constrained quadratic problem with a linear term

$$\min_{0 \leq A \leq 1} f(A) = ABA^T + bA^T$$

where $b$ is a known vector of the same dimension as $A$.

First of all, we note that the new problem is still convex and smooth. In the sequel, we propose to apply the optimal first-order black-box method for smooth convex optimization, i.e., Nesterov’s method [16], [17], [5], to achieve a convergence rate of $O(1/k^2)$ where $k$ is the number of iterations. We first construct the following model for approximating the objective function $f(A)$ in (10) at the point $A$, $h_{C,A}(\bar{A}) = f(A) + f'(A)(\bar{A} - A)^T + \frac{C}{2}\|\bar{A} - A\|^2,$ (11) where $C > 0$ is a constant.

With model (11), the Nesterov’s method is based on two sequences $\{A_k\}$ and $\{s_k\}$ in which $\{A_k\}$ is the sequence of approximate solutions while $\{s_k\}$ is the sequence of search points. The search point $s_k$ is the convex linear combination of $A_{k-1}$ and $A_k$ as

$$s_k = A_k + \beta_k(A_k - A_{k-1})$$

where $\beta_k$ is a properly chosen coefficient. The approximate solution $A_{k+1}$ is computed as the minimizer of $h_{C_k,s_k}(\bar{A})$. It can be proved that

$$A_{k+1} = \arg \min_{0 \leq A \leq 1} \frac{C_k}{2}\left\|\bar{A} - \left(s_k - \frac{1}{C_k}f'(s_k)\right)\right\|^2$$

(12)

where $C_k$ is determined by the line search according to the Armijo-Goldstein rule so that $C_k$ should be appropriate for $s_k$, see [5]. $A_{k+1}$ defined by (12) is actually the projection of the vector $s_k - \frac{1}{C_k}f'(s_k)$ over the convex set $\{A|0 \leq A \leq 1\}$. We can easily work out the projection given by the following formula

$$A_{k+1} = \max\{0, \min\{1, s_k - \frac{1}{C_k}f'(s_k)\}\}.$$  

(13)

where both max and min operate over vectors componentwisely as the same meaning in Matlab.

The efficient algorithm for (10) is summarized on the next page.

C. Reconstruction of Foreground and Background Images

After solving for the alpha values $\alpha$, we need to reconstruct foreground $F$ and background $B$. For this purpose, we take the same strategy as [13] to reconstruct $F$ and $B$ by using the composition equation (1) with certain smoothness priors on both $F$ and $B$. $F$ and $B$ are obtained from optimizing the following objective function

$$\min_{F,B} \sum_i \|\alpha_i F_i + (1 - \alpha_i)B_i - L_i\|^2 + \|\partial \alpha_i, (\partial F_i)^2 + (\partial B_i)^2\|$$

where $\partial$ is the gradient operator over the image grid. For a fixed $\alpha$, the problem is quadratic and its minimum can be found by solving a set of linear equations.

IV. EXPERIMENT RESULTS

To assess the performance of our newly suggested LSTA alignment matrix in image matting, we first compare the results given by our Scribble Smoothing formulation with LSTA alignment matrix and the closed-form matting in [13]. Here the only difference is in the alignment matrices used. Similar to the closed-form solution, we solve (8) for matte values. All the algorithms are implemented using MATLAB on a small workstation machine with 32G memory. In all the calculation, we set $\lambda = 100$, see (8), in both algorithms. As we have pointed out in section 3.2, we ignore the experiment over Scribble Constraint I.

In the first experiment, we used two images taken from the original paper of the closed form solution for image matting [13]. Figure 1 presents the images and their stroked images used in the algorithm comparison.

Figures 2 and 3 present matting results on the two images, respectively. In Figure 2 and 3, the first row shows the results from the closed-form solution while the second row shows the results from the Scribble Smoothing, and the third row shows Scribble Constraint II with the new LSTA alignment matrix. The results given by the LSTA alignment matrix look visually comparable to the results in [13].

Further we conducted the experiments over algorithms for formulations Scribble Constraint I and II. All the algorithms

Algorithm 1 The Efficient Nesterov’s Algorithm

Input: $C_0 > 0$ and $A_0$, $K$

Output: $A_{k+1}$

1: Initialize $A_1 = A_0$, $\gamma_{-1} = 0$, $\gamma_0 = 1$ and $C = C_0.$
2: for $k = 1$ to $K$ do
3: Set $\beta_k = \frac{2k-\gamma_{k-1}}{2\gamma_{k-1}}$, $s_k = A_k + \beta_k(A_k - A_{k-1})$
4: Find the smallest $C = C_{k-1}$, $2C_{k-1}, ...$ such that
5: Set $C_k = C$ and $\gamma_{k+1} = \frac{1 + \sqrt{1 + 4\gamma_k^2}}{2}$
6: end for
are competitive with the closed-form solutions. All the results are presented in Figures 4 and 5.

The above visual results have demonstrated three proposed formulations with the LTSA alignment matrix are very comparable to the closed-form solution for the image matting.

In the second experiment, we aim to assess the new formulations and their algorithms by quantity measurements. For this purpose, we used the benchmark images from the alpha matting website http://www.alphamatting.com/ which provides ground true masks for all the images. As an example, we took the image GT04. We used the low resolution and scaled it to 320 × 226 at 40%.

Fig. 3: Mattes from strokes with the Closed-form solution (the first row), Scribble Smoothing (the second row) and Constraint II (the third row) for LTSA alignment matrix: (a) Learned Masks. (b) Reconstructed foreground images. (c) Extracted background images

Fig. 4: Results for Image Hair: Mattes from strokes. The first row gives the results of algorithms based on formulation Scribble Constraint I while the second row corresponds to the results given by the Nesterov’s algorithm based on formulation Scribble Constraint II. Column (a) shows the learnt mask image, column (b) the extracted foreground image and column (c) the remained background.

Fig. 5: Results for Image Kid: Mattes from strokes. The first row gives the results of algorithms based on formulation Scribble Constraint I while the second row corresponds to the results given by the Nesterov’s algorithm based on formulation Scribble Constraint II. Column (a) shows the learnt mask image, column (b) the extracted foreground image and column (c) the remained background.

The original image and the corresponding strokes used in this example are shown in Figure 6.

We use the mean square errors (MSEs) between learnt the masks and the ground truth mask as the criterion for quality assessment. Constrain I only extracted the foreground scribbles as the mask, as failures. The MSEs for other methods are reported in Table I. The learnt marks are shown in Figure 6 (d)-(f). Actually in many experiments that we conducted, the Scribble constraint II gave slightly better results than other algorithms.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>MSEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closed-Form Solution</td>
<td>0.0118</td>
</tr>
<tr>
<td>Scribble Constraint II</td>
<td>0.0087</td>
</tr>
<tr>
<td>Scribble Smoothing</td>
<td>0.0102</td>
</tr>
</tbody>
</table>

TABLE I: MSE Results
We have introduced a new learning based matting approach based the manifold learning technique, Local Tangential Space Alignment algorithm [25]. The similar approach can be used for the Locally Linear Embedding (LLE) which is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs, see [20]. The argument used in [2] to apply LLE algorithm for image matting is that, like the application of LTSA approach, local linear structure among the color information over a local window can be transferred to the matting value space. This observation is very intuitive for the LLE alignment matrix to be successful in image matting. Similar to the first approach of learning based matting [26], the LTSA-based matte learning introduced in this paper can be extended to the kernel PCA so that nonlinear color information may be accounted for matte learning. This is one of our future research tasks.

In this paper, through several experiments, we showed that the proposed LTSA-based manifold learning approach is comparable to the closed-form formulation. Particularly among three new algorithm formulations, the Scribble Smoothing is the best based on both visual and quantity assessments.

Our initial experience with the LTSA-based manifold learning approach opens several directions for future work. Apart from further improvement of the computation involved in the new approach, we plan to consider more dimensionality reduction/alignment learning algorithms for the purpose of image matting. For example, to further enforce the regulation favoring $0$ or $1$ alpha values, a so-called zero-one regularizer is introduced in [1].

\section*{References}


