The Image Matting Method with Zero-One Regularization

Junbin Gao and Manoranjan Paul
School of Computing and Mathematics
Charles Sturt University
Bathurst, NSW 2795, Australia
{jbgao,rpaul}@csu.edu.au

Jun Liu
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA
j.liu@asu.edu

Abstract

Image matting refers to the problem of accurately extracting foreground objects in images and video. The most recent work [11] and [7] in natural image matting relies on the local and manifold 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 regularization for robust solutions under the so-called zero-one regularisers. We also illustrate our new algorithm using the standard benchmark images and very comparable results have been obtained.

Keywords: Image Matting; Local Tangent Space Alignment; Manifold Learning; Locally Linear Embedding; Laplacian Matrix

1. 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 [20] 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

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

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 which is desired in the matting process. For most of digital matting approaches, an appropriate assumption that \( \alpha_i \) lies between 0 and 1 is widely used, i.e., the so-called soft matte.

In a matting problem, given the image pixel information \( I_i \) 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 [6] uses a progressively moving window marching inward from the known regions. [2] 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 [18] 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 [8] based on Random Walks. The Closed-form matting approach [11] 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 [9], He et al. proposed a fast matting scheme by using large kernel matting Laplacian matrices. Review and assessment of more matting techniques can be found at the alpha matting evaluation website http://www.alphamatting.com/.

Majority of matting algorithms rely on certain amount of user’s scribble information to make the algorithm themselves successful. Too few information may result in matting failure as demonstrated by examples in [9]. Basically many of the current matting approaches like the Laplacian Matting [11, 12], LTSA matting [7], and LLE mat-
ting [1] formulate the matting problem as an unconstrained quadratic optimization problem (except for user-specified scribble constraints which can be removed by variable substitution) although one formulation proposed in [7] involves the constrain $0 \leq \alpha_i \leq 1$. In this paper, we try to introduce a different robust quasi-hard matte regularization to reduce the dependency of user-specified constraints (such as scribbles or a bounding rectangle) as done in [11, 12, 16].

The paper is organized as follows. Section 2 simply reviews three main image matting algorithms, Laplacian Matrix Matting [11], LTSA alignment matrix matting [21] and LLE alignment matrix matting [1]. Section 3 is dedicated to formulating the new regularization and propose a fast optimization algorithm for the new formulation. In Section 4, we present several examples of using the new approach over the benchmark images, see [11]. We make our conclusions in Section 5 with suggestions of further exploration in image matting.

2. Reviews of Matrix-Based Matting Approaches

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 $I_i$.

2.1. Matting Laplacian Matrix Approach

The matting Laplacian matrix approach is proposed in [11]. The matrix defines certain affinity of color information specially designed for image matting. The success of the proposed approach relies on the so-called color line model: the foreground colors in a local window lie on a single line in the RGB color space, which is described as a local piecewise linear transformation of the image $I$,

$$\alpha_{ij} = a_i^T I_j + b_i, \forall ij \in w_i = \{i_1, ..., i_K\}$$

where $w_i = \{i_1, ..., i_K\}$ containing $K$ neighbor pixels of $i$ denotes a local window of certain size at pixel $i$ and $a_i$ and $b_i$ are assumed to be constant in the local window. Under the above model assumption, accordingly, an objective function $E(\alpha, A, b)$ can be defined to reinforce the alpha values fitting this model:

$$E(\alpha, A, b) = \sum_{i} \sum_{j=1}^{K} \left[ (\alpha_{ij} - a_i^T I_j - b_i)^2 + \epsilon a_i^T a_i \right]$$

(2)

where $\epsilon > 0$ is a given parameter controlling numerical stability, $\alpha = (\alpha_1, ..., \alpha_N)^T$ is the vector of all the alpha values and $A = [a_1, ..., a_N]$ and $b = [b_1, ..., b_N]^T$ with $N$ the number of pixels on the image $I$.

By minimizing the cost function with respect to $(A, b)$, a quadratic function of $\alpha$ can be obtained:

$$E(\alpha) = \alpha^T L \alpha$$

Please refer to [11] for the detail of calculating the elements of the matting Laplacian matrix $L$.

Combining this objective function with the user-specified constraints like a trimap $\Omega$, the whole objective function to be optimized is defined as

$$F(\alpha) = \alpha^T L \alpha + \lambda (\alpha - \alpha_D)^T D_\Omega (\alpha - \alpha_D)$$

(3)

where $\alpha_D$ is the trimap or user’s scribble, and $D_\Omega$ is a diagonal matrix whose elements are one for trimap scribble constrained pixels $\Omega$ and zero otherwise and $\lambda$ is a large number enforcing larger penalty over the alpha components corresponding to constrained pixels $\Omega$.

Eqn (3) defines an unconstrained optimization problem which can be easily solved and the user-specified information is enforced by choosing a larger regularizer $\lambda$.

2.2. LTSA Alignment Matrix Approach

The Local Tangent Subspace Alignment (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 order to incorporate pixel structures, [7] defines data neighborhood by using their spatial affinity of pixel location. That is, for each pixel $i$, the neighborhood of $I_i$ is all the RGB vectors $I_{ij}$ with $ij$ being in the neighbor of $i$ in terms of a local window $w_i = \{i_1, ..., i_K\}$. For each pixel $i$, define a subset

$$X_i = \{I_{ij} | ij \in w_i, j = 1, ..., K\}.$$  (4)

Applying the classical PCA [4] over $X_i$, there exist a $Q_i$ of $d \leq 2$ orthonormal columns such that

$$I_{ij} = \bar{I}_i + Q_i \theta_{ij}^{(i)} + \xi_{ij}^{(i)}$$  \hspace{1cm} (5)

where $\xi_{ij}^{(i)} = (E - Q_i Q_i^T) (I_{ij} - \bar{T}_i)$ with the identity matrix $E$ is the reconstruction error, $\theta_{ij}^{(i)}$ is the local coordinates over local tangent space in the color space and $\bar{T}_i$ is the mean color vector. As done in the LTSA algorithm, $Q_i$ can be obtained by computing the SVD decomposition of $X_i$. For more details refer to [21, 7].

For the purpose of image matting, the authors of [7] reconstruct global matting feature $\alpha_i$ of the local coordinates $\theta_{ij}^{(i)}$ based on the local color information on the manifold.
defined by local windows. Specifically, the prospective matting values $\alpha_i$, satisfies the following set of equations, according to local structures determined by the $\theta_j^{(i)}$,

$$\alpha_i = \overline{\alpha}_i + L_i \theta_j^{(i)} + \epsilon_j^{(i)}, \quad j = 1, ..., K; i = 1, ..., N$$

where $\overline{\alpha}_i$ is the mean of $\alpha_{ij}$ ($j = 1, ..., K$). Denote $\alpha_i = [\alpha_{i1}, ..., \alpha_{ik}]$, $\Theta_i = [\theta_1^{(i)}, ..., \theta_K^{(i)}]$ and $E_i = [\epsilon_1^{(i)}, ..., \epsilon_K^{(i)}]$, then the above linear system can be written in a matrix form,

$$\alpha_i = \frac{1}{K} \alpha_i e e^T + L_i \Theta_i + E_i$$

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

$$\min_{L_i, \alpha} \sum_i \|E_i\|^2 = \sum_i \|\alpha_i (E - \frac{1}{K} ee^T) - L_i \Theta_i\|^2 \quad (6)$$

Denote $\alpha = (\alpha_1, ..., \alpha_N)^T$. Similar to the matting Laplacian matrix formulation, the above optimal problem can be solved by minimizing it with respect to each $L_i$, then the minimizer $\alpha$ should be the solution of the following revised optimal problem after some notation manipulation

$$\min_{\alpha} \sum_i \|E_i\|^2 = \alpha^T B \alpha$$

where $B$ is called LTSA alignment matrix which can be calculated by a simple iterative procedure. See [21, 7]. Like the matting Laplacian matrix $L$, the matrix $B$ is also semi-definite positive.

In [7], three different formulations are proposed by combining the above objective function with different constraint. The below is one of examples. The matting values $\alpha$ is sought by

$$\min_{\Gamma(\alpha) = \alpha_\Omega} F(\alpha) = \sum_i \|E_i\|^2 = \alpha^T B \alpha \quad (7)$$

where $\Gamma_\Omega$ is the operator mapping $\alpha$ to the alpha value $\alpha_\Omega$ over user-specified pixels $\Omega$.

2.3. LLE Alignment Matrix Approach

Locally linear embedding (LLE) is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs, see [17]. LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by images of faces or documents of text.

The argument discussed in [1] to use 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.

In the sequel, we still use $X_i = \{I_{ij} | w_{ij} \in w_i, j = 1, ..., K\}$ to denote the subset of color vectors over a local window pixels of pixel $i$. Note that the pixel $I_i$ is contained in $X_i$. Under the LLE assumption, the color vector $I_i$ at pixel $i$ can be approximated by a linear combination $w_{ij}$ (the so-called reconstruction weights) of its $K - 1$ nearest neighbors $X_i \setminus I_i$. Hence, LLE fits a hyperplane through $I_i$ and its nearest neighbors in the color manifold defined over the image pixels. The fitting is achieved by solving the following optimal problem

$$\min_{W} \sum_i \|I_i - \sum_{j=1}^{K} w_{ij} I_{ij}\|^2$$

under the condition $\sum_{j=1}^{K} w_{ij} = 1$. For the sake of simplicity, we assume that $w_{ij} = 0$ when $I_{ij} = I_i$. Assuming that the manifold is locally linear, the local linearity may be preserved in the space of matting values. Once the weights $W$ have been determined, the matting values $\alpha$ can be determined by minimizing the following objective function

$$\min_{\alpha} F(\alpha) = \sum_i \|\alpha_i - \sum_{j=1}^{K} w_{ij} \alpha_{ij}\|^2 = \alpha^T R \alpha \quad (8)$$

where $R = (E - W)^T (E - W)$ is called LLE alignment matrix. In the standard LLE algorithm, (8) is usually solved with additional constrained conditions like $\|\alpha\|^2 = 1$ which results in an eigen vector problem. Instead of such standard constraints, in image matting we can formulate a matting solution by including similar smoothing constraint as (3) or matting constraint as (7).

3. Zero-One Regularized Matting and Its Efficient Algorithm

3.1. Zero-One Regularization

Each of the three methods introduced in the last section involves in a quadratic objective function with some additional constraints. As we have already pointed out, in practices, we favor a constraint which results in a hard matting value, meaning $\alpha_i = 0$ or $\alpha_i = 1$. Inspired by the recent research in sparse modeling approaches [13, 10], the hard matting values may be enforced by applying a sparsity inducing penalty. Obviously the LASSO [19] is one of widely used sparsity inducing penalties. A lot of practical algorithms have been proposed since then. In this paper, we
propose to use the $\ell_1$ penalty as well to enforce hard matting values. The $\ell_1$ penalty is combined with the common objective functions in the three approaches reviewed in the last section. Thus, the matting problem is formulated as finding a solution to the following optimization problem

$$
\min_{\alpha} \quad F(\alpha) = \alpha^T \Phi \alpha + \lambda_1 |\alpha|_1 + \lambda_2 |e - \alpha|_1
$$

where $e$ is a vector of all components 1s and $\Phi$ is a semi-positive definite matrix such as the matting Laplacian matrix $L$, both $\lambda_1 > 0$ and $\lambda_2 > 0$ are two user-specified regularizers, and $|\alpha|_1 = \sum |\alpha_i|$ and $|e - \alpha|_1 = \sum |1 - \alpha_i|$. Given the fact that the constraints $\Gamma_\Omega(\alpha) = \alpha_\Omega$ is the equality for the variables, this constraint can be removed by variable substitution. Without loss of generality, suppose that $\alpha$ has been ordered as $\alpha = (\alpha_{\Omega_1}, \alpha_{\Omega_2})$ where $\alpha_{\Omega_2}$ is the user given scribbles or trimap. Accordingly we decompose the $\Phi$ into

$$
\Phi = \begin{pmatrix} \Phi_{\Omega_1} & \Phi_{\Omega_2} \\ \Phi_{\Omega_2} & \Phi_{\Omega_2} \end{pmatrix}
$$

where $\Phi_{\Omega_2} = \Phi_{21}$. Taking the above decomposition into $F(\alpha)$ results in the following unconstrained problem with respect to $\alpha_{\Omega_2}$,

$$
\min_{\alpha_{\Omega_2}} \alpha_{\Omega_2}^T \Phi_{\Omega_1} \alpha_{\Omega_2} + 2\lambda_2 \Phi_{21} \alpha_{\Omega_2} + \lambda_1 |\alpha_{\Omega_2}|_1 + \lambda_2 |e - \alpha_{\Omega_2}|_1
$$

In the sequel, without loss of generality, we consider a general unconstrained formulation with a linear term

$$
\min_{\alpha} F(\alpha) = \alpha^T \Phi \alpha + b^T \alpha + \lambda_1 |\alpha|_1 + \lambda_2 |e - \alpha|_1
$$

(9)

where $b$ is a known vector of dimension $N$. We call this type of penalty the zero-one LASSO penalty.

3.2. Constrained Zero-One Regularization

The above zero-one regularization formulation can be further strengthened by imposing the constraints $0 \preceq \alpha \preceq e$. Thus we have the following constrained zero-one regularization,

$$
\min_{0 \preceq \alpha \preceq e} F(\alpha) = \alpha^T \Phi \alpha + b^T \alpha + \lambda_1 |\alpha|_1 + \lambda_2 |e - \alpha|_1
$$

(10)

3.3. Smoothed Zero-One Regularization

Two formulations presented above use the strict trimap/scrribble constraints $\Gamma_\Omega(\alpha) = \alpha_\Omega$ which is removed by using variable substitution. However the smoothing trimap/scrribble method as done in (3) has been proven to be robust in matting [11]. This inspires us to define a combined version of smoothes zero-one regularization as

$$
F(\alpha) = \alpha^T \Phi \alpha + \lambda(\alpha - \alpha_\Omega)^T D_\Omega(\alpha - \alpha_\Omega) + \lambda_1 |\alpha|_1 + \lambda_2 |e - \alpha|_1
$$

(11)

where $\Phi$ as usual is the Laplacian matrix $L$, LTSA alignment matrix $B$ or LLE alignment matrix $R$.

3.4. Suggested Optimization Algorithms

In (9) and (10), denote $f(\alpha) = \alpha^T \Phi \alpha + b^T \alpha$ or $f(\alpha) = \alpha^T (\Phi + \lambda D_\Omega) \alpha - 2\lambda (D_\Omega^T \alpha_\Omega)^T \alpha + \lambda \alpha_\Omega^T D_\Omega \alpha_\Omega$ and $\phi(\alpha) = \lambda_1 |\alpha|_1 + \lambda_2 |e - \alpha|_1$, then we can see that $f(\alpha)$ is a convex smooth function while $\phi(\alpha)$ is a convex nonsmooth function. Both problems in (9) and (10) are challenging to solve, as the zero-one LASSO is non-smooth and there is a constraint. One of algorithms to solve (9) is to reformulate it as an equivalent constrained smooth optimization problem by introducing additional variables and constraints, and then apply the standard solver for optimization.

Note that the only difference between (9) and (10) is the extra constraint $0 \preceq \alpha \preceq e$. First we focus on (9). In this paper, we develop an Efficient Zero-One Lasso Algorithm (EZOLA) based on the fact that the objective function of (9) is a composite function with the smooth part $f(\alpha)$ and a non-smooth part $\phi(\alpha)$. One appealing feature of EZOLA is that it makes use of the special structure of (9) for achieving a convergence rate of $O(1/k^2)$ for $k$ iterations, which is optimal for the first-order black-box methods. Note that, when directly applying the black-box first-order method for solving the non-smooth problem (9), one can only achieve a convergence rate of $O(1/\sqrt{k})$, much slower than $O(1/k^2)$.

In the sequel, we propose to apply the optimal first-order black-box method for composite non-smooth convex optimization, i.e., one type of Nesterov’s methods [14, 15, 3], to achieve a convergence rate of $O(1/k^2)$. We first construct the following model for approximating the objective function $F(\alpha)$ in (9) at the point $\alpha$,

$$
h_{C,\alpha}(\tilde{\alpha}) = f(\alpha) + f'(\alpha)(\tilde{\alpha} - \alpha)^T + \frac{C}{2} \|\tilde{\alpha} - \alpha\|^2 + \phi(\alpha),
$$

(12)

where $C > 0$ is a constant.

With model (12), the Nesterov’s method is based on two sequences $\{\alpha_k\}$ and $\{s_k\}$ in which $\{\alpha_k\}$ 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 $\alpha_{k-1}$ and $\alpha_k$ as

$$
s_k = \alpha_k + \beta_k (\alpha_k - \alpha_{k-1})
$$

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

$$
\alpha_{k+1} = \arg\min_{\alpha} \frac{C_k}{2} \|\tilde{\alpha} - \left( s_k - \frac{1}{C_k} f'(s_k) \right) \|^2 + \phi(s_k)
$$

(13)

where $C_k$ is determined by the line search according to the Armijo-Goldstein rule so that $C_k$ should be appropriate for
\[ s_k, \text{ see } [3]. \] Optimization problem defined in (13) can be decoupled into the following single-variable optimization problem

\[ \min_{\alpha} h(\alpha) = \frac{1}{2} ||\alpha - v||^2 + \lambda_1 |\alpha| + \lambda_2 |1 - \alpha| \]  (14)

where \( v \) is known and determined by \( s_k - \frac{1}{\epsilon_2} f'(s_k) \).

(14) can be easily solved by the following lemma:

**Lemma 1.** The solution to the problem (14) can be analytically computed as:

\[ \alpha^* = \begin{cases} v - \lambda_1 - \lambda_2, & v - \lambda_1 - \lambda_2 > 1 \\ 1, & |1 - v - \lambda_1| \leq \lambda_2 \\ v - \lambda_1 + \lambda_2, & 0 < v - \lambda_1 + \lambda_2 < 1 \\ 0, & |v + \lambda_2| \leq \lambda_1 \\ v + \lambda_1 + \lambda_2, & v + \lambda_1 + \lambda_2 < 0 \end{cases} \]

**Proof:** The necessary and sufficient condition for \( \alpha^* \) being the optimal solution to (14) is that \( 0 \notin \partial h(\alpha^*) \). We note that, the solution is unique. We can consider the following five cases.

- Case 1: \( \alpha^* > 1 \). Since \( 0 \notin h(\alpha^*) = {\alpha^* - v + \lambda_1 + \lambda_2} \), we have \( \alpha^* = v - \lambda_1 - \lambda_2 \) and \( v - \lambda_1 - \lambda_2 > 1 \).
- Case 2: \( \alpha^* = 1 \). Since \( 0 \notin h(\alpha^*) = {1 - v + \lambda_1 + \lambda_2 SGN(0)} \), we have \( |1 - v + \lambda_1| \leq \lambda_2 \).
- Case 3: \( 0 < \alpha^* < 1 \). Since \( 0 \notin h(\alpha^*) = {\alpha^* - v + \lambda_1 - \lambda - 2} \), we have \( \alpha^* = v - \lambda_1 + \lambda_2 \) and \( 0 < v - \lambda_1 + \lambda_2 < 1 \).
- Case 4: \( \alpha^* = 0 \). Since \( 0 \notin h(\alpha^*) = {-v + \lambda_1 SGN(0) - \lambda_2} \), we have \( |v + \lambda_2| \leq \lambda_1 \).
- Case 5: \( \alpha^* < 0 \). Since \( 0 \notin h(\alpha^*) = {\alpha^* - v - \lambda_1 - \lambda_2} \), we have \( \alpha^* = v + \lambda_1 + \lambda_2 \) and \( v + \lambda_1 + \lambda_2 < 0 \).

This completes the proof.

Now let us turn to the constrained version (10). Nesterov’s method can be easily applied to the constrained version [3]. In the constrained case, the building block similar to (14) is formulated according to constrain conditions. In its single component, this is given by

\[ \min_{0 \leq \alpha \leq 1} h(\alpha) = \frac{1}{2} ||\alpha - v||^2 + \lambda_1 |\alpha| + \lambda_2 |1 - \alpha| \]  (15)

Then we have

**Lemma 2.** The solution to the problem (15) can be analytically computed as:

\[ \alpha^* = \begin{cases} 1, & v \geq 1 + \lambda_1 - \lambda_2 \\ v - \lambda_1 + \lambda_2, & \lambda_1 - \lambda_2 < v < 1 + \lambda_1 - \lambda_2 \\ 0, & v \leq \lambda_1 - \lambda_2 \end{cases} \]

**Proof:** Due to constraint condition, the absolute operation in the objective function can be removed, so we are considering the following optimal problem

\[ \min_{0 \leq \alpha \leq 1} h(\alpha) = \frac{1}{2} (\alpha - v)^2 + \lambda_1 \alpha + \lambda_2 (1 - \alpha) \]

The corresponding Lagrangian is

\[ L(\alpha, \mu_1, \mu_2) = \frac{1}{2} (\alpha - v)^2 + \lambda_1 \alpha + \lambda_2 (1 - \alpha) + \mu_1 \alpha + \mu_2 (1 - \alpha) \]

where \( \mu_1 \leq 0 \) and \( \mu_2 \leq 0 \). For the optimal value \( \alpha^* \), \( \mu_1^* \) and \( \mu_2^* \), TTK conditions [5] are

\[ \alpha^* - v + \lambda_1 - \lambda_2 + \mu_1^* - \mu_2^* = 0 \]
\[ 0 \leq \alpha^* \leq 1 \]
\[ \mu_1^* \alpha^* = 0 \] and \( \mu_1^* \leq 0 \)
\[ \mu_2^*(1 - \alpha^*) = 0 \] and \( \mu_2^* \leq 0 \)

Now we consider three cases:

- Case 1: \( \alpha^* = 0 \). When this happens, we must have \( \mu_2^* = 0 \). Hence \( v = 1 + \lambda_1 - \lambda_2 + \mu_2^* \leq 1 + \lambda_1 - \lambda_2 \).
- Case 2: \( \alpha^* = 1 \). When this happens, we must have \( \mu_1^* = 0 \). Hence \( v = 1 + \lambda_1 - \lambda_2 - \mu_2^* \leq 1 + \lambda_1 - \lambda_2 \).
- Case 3: \( 0 < \alpha^* < 1 \). In this case, we must have both \( \mu_1^* = \mu_2^* = 0 \). Hence \( \alpha^* = v - \lambda_1 + \lambda_2 \) and \( \lambda_1 - \lambda_2 < v < 1 + \lambda_1 - \lambda_2 \).

This completes the proof of Lemma 2.

In the sequel, as an example, we propose the efficient algorithm for (9). To do so, taking as \( v \) in Lemma 1 each component of \( s_k - \frac{1}{\epsilon_2} f'(s_k) \) in (13) in turn, we can work out each component of the minimizer \( \alpha_{k+1} \) of (13).

The whole algorithm for (9) or (10) is summarized in Algorithm 1 on the next page.

Finally the smoothed zero-one regularization problem (11) can be easily solved by using the above Nesterov’s method.

### 3.5. 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,
Algorithm 1 The Efficient Nesterov’s Algorithm

Input: $C_0 > 0$ and $\alpha_0$, $K$
Output: $\alpha_{k+1}$
1: Initialize $\alpha_1 = \alpha_0$, $\gamma_{-1} = 0$, $\gamma_0 = 1$ and $C = C_0$.
2: for $k = 1$ to $K$ do
3: Set $\beta_k = \frac{\gamma_{k-1}}{\gamma_k}$, $s_k = \alpha_k + \beta_k(\alpha_k - \alpha_{k-1})$
4: Find the smallest $C = C_{k-1}, 2C_{k-1}, \ldots$ such that $F(\alpha_{k+1}) \leq h_{C,s_k}(\alpha_{k+1})$, where $\alpha_{k+1}$ is defined by (13) and solved by Lemma 1 or by Lemma 2 for the constrained case.
5: Set $C_k = C$ and $\gamma_{k+1} = \frac{1 + \sqrt{1 + 4\gamma_k^2}}{2}$
6: end for

we take the same strategy as [11] 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

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

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.

4. Experiment Results

To assess the performance of three newly suggested formulations for image matting, we will take the LTSA alignment matrix as an example. LTSA alignment matrix has been proven to be very comparable to the original Laplacian matrix [7, 11] in image matting.

All the algorithms are implemented using MATLAB on a small workstation machine with 32G memory. We use benchmark images taken from the matting website http://www.alphamatting.com. Four images (GT02, GT04, GT13 and GT26) are chosen for testing the proposed Nesterov’s algorithms for Zero-One Regularization (ZOR), Constrained Zero-One Regularization (CZOR) and Smoothed Zero-One Regularization (SZOR). Figure 1 presents those original images (column one), their trimap images (column two) and ground truth matting masks (column three). The trimap images are used in the algorithms as users guidance. In order to run the algorithms on our machine, we downsize both images and trimaps by 40%, respectively.

We note that when $\lambda_1 = \lambda_2$ the algorithm of ZOR is equivalent to that of CZOR. To distinguish them in the experiment, we set $\lambda_1 = 100, \lambda_2 = 100.01$ for ZOR and $\lambda_1 = 100, \lambda_2 = 100.01$ for CZOR formulations. That is, we slightly prefer to foreground objects. We set $\lambda = 100, \lambda_1 = 100, \lambda_2 = 100.01$ for SZOR formulation. In Nesterov’s method we set $C_0 = 1$ for all the experiments. The LTSA alignment matrix and the Laplacian matrix are constructed with window size 2.

In Figure 2, the first column shows the result from the
Nesterov’s method for the ZOR formulation, the second column for the CZOR formulation and the third column for the SZOR formulation. Visually it is easy to identify that the results in column three of Figure 2 are quite close to the ground truth masks shown in the third column of Figure 1. The results of SZOR are superior to the results given by both ZOR and CZOR formulations and the ZOR is quite comparable to CZOR. For images GT02 and GT04, most of background in holes are recovered by the SZOR formulation. However, for image GT26, it is very hard to distinguish the results given by three formulations.

As a comparison, we also applied the standard close-form solution of Laplacian matrix [11] to the same images and their trimaps. The learned masks are displayed in Figure 3. In general the results are slightly inferior to the results given by both ZOR and CZOR formulation.

![Figure 3](https://via.placeholder.com/150)

**Figure 3: Foreground Masks Learned by the Standard Close-form Solution based on the Laplacian Matrix**

To better assess the results quantitatively, we also measure the error between the ground truth masks and learnt masks. Table 1 shows the mean square errors (MSEs) for each case. The MSE rank of our algorithms shows the SZOR formulation is the winner for most case while the SZOR performs worst for image GT26.

![Table 1](https://via.placeholder.com/150)

**Table 1: MSE Results**

We also tested the impact of different window sizes in the smoothed zero-one formulation using image GT04. The LTSA alignment matrices are constructed at four window size $w = 2, 3, 4, 5$. The learnt masks at four window sizes are shown in Figure 4.

![Figure 4](https://via.placeholder.com/150)

**Figure 4: Foreground Masks Learned by the SZOR formulation with four different window sizes when constructing LTSA alignment matrix**

Table 2 reports the MSEs given by SZOR formulation from which we can see that the best result is obtained in the case of window size $w = 3$. We have to stress that we failed to run the Matlab program of the closed-form solution provided in [11] for large window sizes due to the fact that the matting Laplacian is getting less sparse in larger window size. In the later case, both the memory and the time needed to solve the linear system will increase tremendously.

![Table 2](https://via.placeholder.com/150)

**Table 2: MSE Results for GT04 at Different Window Sizes**

In the last experiment, we aim to assess the quality of three new formulations for image matting with users scribbles. Due to the space limitation, here we only report the testing result for image GT02 in which there are many small holes. The original image and the corresponding strokes used in this experiment are shown in Figure 5.

![Figure 5](https://via.placeholder.com/150)

**Figure 5: Benchmark Image GT02: The original image (left) and The image with strokes (right).**

Table 3 reports the MSEs for all the tests. The measures show the SZOR is the best among three formulations.

![Table 3](https://via.placeholder.com/150)

**Table 3: MSE Results for GT02 at Different Window Sizes**

For each of formulations ZOR, CZOR and SZOR, we tested the LTSA matrices with four different window sizes. Figure 6 shows the learnt masks (the first row), the extracted foreground images (the second row) and the background images (the third row) for the best window size in each formulation. It is clear to see that the mask given by SZOR formulation reveals more small holes in the image.

Furthermore, Table 3 reports the MSEs for all the tests. The measures show the SZOR is the best among three formulations.
Figure 6: Best Results for Each Formulation: Column one for ZOR with $w = 2$, Column two for CZOR with $w = 5$, and SZOR with $w = 4$.

### Table 3: MSE Results for GT02 for ZOR, CZOR and SZOR at Different Window Sizes

<table>
<thead>
<tr>
<th>Formulations</th>
<th>$w = 2$</th>
<th>$w = 3$</th>
<th>$w = 4$</th>
<th>$w = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZOR</td>
<td>0.00651</td>
<td>0.00699</td>
<td>0.00713</td>
<td>0.00706</td>
</tr>
<tr>
<td>CZOR</td>
<td>0.02839</td>
<td>0.00715</td>
<td>0.00724</td>
<td>0.00712</td>
</tr>
<tr>
<td>SZOR</td>
<td>0.00912</td>
<td>0.00659</td>
<td>0.00638</td>
<td>0.00662</td>
</tr>
</tbody>
</table>

### 5. Discussion

We proposed three optimization formulations for the image matting based on the LTSA alignment matrix. The similar approach can be applied to the Laplacian matrix introduced in the closed-form solution of image matting in [11] and the Locally Linear Embedding (LLE) alignment matrix [17]. The experiment result shows that the smoothed zero-one regularization gives the best visual and quantitative result in image matting assisted with both the trimap and user’s scribbles.

### Acknowledgements

This work is partially supported by [University Name Removed] Competitive Research Grant OPA 4818.

### References


