Tensorial Evolutionary Optimization for Natural Image Matting

SI-CHAO LEI and YUE-JIAO GONG, South China University of Technology, Guangzhou, China
XIAO-LIN XIAO, South China Normal University, Guangzhou, China
YI-CONG ZHOU, University of Macau, Macau, China
JUN ZHANG, Hanyang University, Seoul, South Korea

Natural image matting has garnered increasing attention in various computer vision applications. The matting problem aims to find the optimal foreground/background (F/B) color pair for each unknown pixel and thus obtain an alpha matte indicating the opacity of the foreground object. This problem is typically modeled as a large-scale pixel pair combinatorial optimization (PPCO) problem. Heuristic optimization is widely employed to tackle the PPCO problem owing to its gradient-free property and promising search ability. However, traditional heuristic methods often encode F/B solutions to a one-dimensional (1D) representation and then evolve the solutions in a 1D manner. This 1D representation destroys the intrinsic two-dimensional (2D) structure of images, where the significant spatial correlations among pixels are ignored. Moreover, the 1D representation also brings operation inefficiency. To address the above issues, this article develops a spatial-aware tensorial evolutionary image matting (TEIM) method. Specifically, the matting problem is modeled as a 2D Spatial-PPCO (S-PPCO) problem, and a global tensorial evolutionary optimizer is proposed to tackle the S-PPCO problem. The entire population is represented as a whole by a third-order tensor, in which individuals are classified into two types: F and B individuals for denoting the 2D F/B solutions, respectively. The evolution process, consisting of three tensorial evolutionary operators, is implemented based on pure tensor computation for efficiently seeking F/B solutions. The local spatial smoothness of images is also integrated into the evaluation process for obtaining a high-quality alpha matte. Experimental results compared with state-of-the-art methods validate the effectiveness of TEIM.

CCS Concepts: • **Computing methodologies** → **Computer vision problems**; Image segmentation; • **Theory of computation** → *Random search heuristics*;

This work was supported in part by the National Natural Science Foundation of China under Grant 62276100, in part by the Guangdong Natural Science Funds for Distinguished Young Scholars under Grant 2022B1515020049, in part by the Guangdong Regional Joint Funds for Basic and Applied Research under Grant 2021B1515120078, and in part by the TCL Young Scholars Program.

Authors' addresses: S.-C. Lei and Y.-J. Gong (Corresponding author), School of Computer Science and Technology, South China University of Technology, University Town Campus, Guangzhou, Guangdong, 510000, China; e-mails: cssclei@outlook.com, gongyuejiao@gmail.com; X.-L. Xiao (Corresponding author), School of Computer Science, South China Normal University, Guangzhou, Guangdong, 510000, China; e-mail: shellyxiaolin@gmail.com; Y.-C. Zhou, Department of Computer and Information Science, University of Macau, Macau, 999078, China; e-mail: yicongzhou@um.edu.mo; J. Zhang, Department of Electrical and Electronic Engineering, Hanyang University Erica, Ansan, 15588, South Korea; e-mail: jun-zhang@ieee.org.

ACM 1551-6857/2024/03-ART194
https://doi.org/10.1145/3649138

Additional Key Words and Phrases: Natural image matting, tensorial evolutionary algorithm, heuristic optimization

ACM Reference Format:
Si-Chao Lei, Yue-Jiao Gong, Xiao-Lin Xiao, Yi-Cong Zhou, and Jun Zhang. 2024. Tensorial Evolutionary Optimization for Natural Image Matting. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 7, Article 194 (March 2024), 23 pages. https://doi.org/10.1145/3649138

1 INTRODUCTION

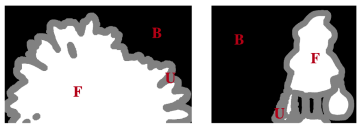
Natural image matting, aimed at precisely extracting foreground objects, is a long-standing problem in computer graphics and vision. It serves as the basis for image and video editing and is widely applied in various fields such as film production [36], object tracking [3, 38], and augmented reality [17, 26]. Typically, a four-channel image comprises Red, Green, Blue, and Alpha channels, which can be represented by a convex linear combination of the foreground and background colors as

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

where I, $I^{(F)}$, $I^{(B)}$ denote the true RGB color of the observed image, the corresponding foreground and background colors, respectively; $\alpha \in [0, 1]$ indicates the transparency of the foreground objects, which is usually stored in an image known as the alpha matte. The matting problem is then defined as to obtain the alpha matte, given only the observed image and prior information such as scribbles or a trimap [9]. Figure 1 depicts two examples of trimaps, where each image is partitioned into three distinct regions: the known foreground ($\alpha = 1$), the known background ($\alpha = 0$), and the unknown region. Trimap-based methods heavily depend on the information provided by the known **foreground (F)** and **background (B)** to compute the transparency of pixels in the **unknown (U)** region.

Normally, the matting problem is a complex and large-scale **pixel pair combinatorial optimization (PPCO)** problem. Its goal is to find the optimal F/B color pair for each pixel in the U region and then derive the alpha value. In early studies, the problem was solved by sampling to select the optimal F/B color pairs. Different handcrafted sampling strategies were designed to select sub-F and B regions, and then determine the optimal F/B pair for each unknown pixel according to the nearest neighbor [15, 16, 40, 41]. However, although the sampling methods are computationally efficient, their accuracy is rather unsatisfactory. Sampling-free methods were therefore proposed to enhance the performance, where heuristic optimization techniques are employed to seek the optimal pixel pairs under the complete search space of F/B colors. These algorithms have achieved promising performance in solving the PPCO problem, comprising methods such as **particle swarm optimization (PSO)** [20, 33], **cooperative coevolution differential evolution (CC-DE)** [4], **ant colony optimization (ACO)** [50], and **multi-objective evolutionary algorithm (MOEA)** [34].

Figure 2 illustrates the pipeline of existing heuristic matting methods. Given a trimap, the decision space is the U region, while the search space is determined based on the F/B regions. Traditional matting methods include a sampling step to narrow the search space for easing the computing pressure. However, it was later discarded, since the newly developed heuristic algorithms can greatly enhance the problem-solving capability. Then, the optimizer is employed to find the optimal F/B color pair for each unknown pixel. Finally, the alpha matte is obtained according to Equation (1). For heuristic matting, a proper evaluation is required to measure the quality of candidate F/B solutions. Nearly all the existing heuristic matting methods evaluate the F/B solutions



(a) trimap 1 (b) trimap 2

Fig. 1. Examples of the trimap. F denotes the known foreground region, B is the known background region, and U represents the unknown region.

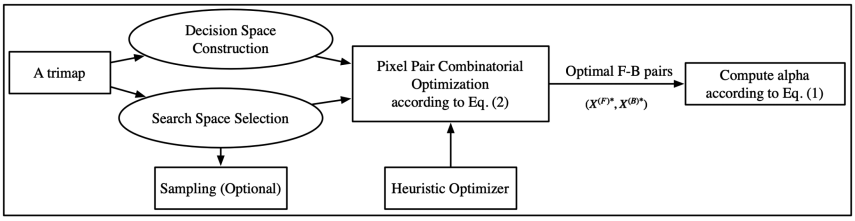


Fig. 2. Main procedures of heuristic matting methods.

based on color similarity and spatial closeness [16], which are typically formulated as

$$\begin{aligned} \min G(X) &= \sum_{k=1}^{N_U} g_c(x_k) + g_s(x_k), \\ g_c(x_k) &= \left\| p_k^{(C,U)} - \left(\hat{\alpha}_k p_{x_k^{(F)}}^{(C,F)} + (1-\hat{\alpha}_k) p_{x_k^{(B)}}^{(C,B)} \right) \right\|, \\ g_s(x_k) &= \frac{||p_k^{(S,U)} - p_{x_k^{(F)}}^{(S,F)}||}{\min_{t=1}^{N_U} ||p_k^{(S,U)} - p_t^{(S,F)}||} + \frac{||p_k^{(S,U)} - p_{x_k^{(B)}}^{(S,B)}||}{\min_{t=1}^{N_U} ||p_k^{(S,U)} - p_t^{(S,B)}||}, \end{aligned} \tag{2}$$

where $X = \{x_1, \ldots, x_k, \ldots, x_{N_U}\}$, and $x_k = (x_k^{(F)}, x_k^{(B)})$ denotes the F/B decision variables for the kth unknown pixel; $g_c(x_k)$ evaluates the color distortion of a F/B color pair where $\hat{\alpha}_k$ is the estimated alpha value, and $p^{(C,F)}$, $p^{(C,B)}$, and $p^{(C,U)}$ stand for the RGB values of F, B, and U pixels, respectively; $g_s(x_k)$ calculates the spatial costs of F and B pixel where $p^{(S,F)}$, $p^{(S,B)}$, and $p^{(S,U)}$ denote the spatial coordinates of F, B, and U pixels, respectively. The overall evaluation function $G(X)$ sums up $g_c(x_k)$ and $g_s(x_k)$ over all unknown pixels.

Despite the current advances in heuristic matting methods, there are still significant limitations in both the effectiveness and efficiency, mainly due to the high dimensionality of decision variables and the complexity of the search space. First, most of the matting methods adopt a trimap expansion procedure that expands the F/B regions toward U region based on color similarity [4, 31]. The available F/B labels are assigned to unknown pixels when the color distances are less than a pre-defined threshold. This greatly narrows the U region, improving the computation efficiency. However, the color thresholding is unstable and may fail when the F/B labels are wrongly assigned. Another critical limitation of previous studies is the failure to incorporate and utilize the spatial structure of images into the heuristic optimization process. Traditional heuristic methods operate on solutions of **one-dimensional (1D)** structure and accordingly develop 1D evolutionary operators for evolving new generations. Consequently, they follow a 1D evolving pattern, and the valuable spatial information of images is largely ignored.

To address the aforementioned limitations, in this article, we propose a spatial-aware **tensorial evolutionary image matting (TEIM)** method to improve the accuracy and efficiency of the matting methods/algorithms. First, we develop a superpixel-based trimap expansion method that effectively expands F/B regions. We then build upon a tensorial evolutionary optimizer to tackle the matting problem based on tensorial evolutionary optimization. In our previous work, we have proposed a general **tensorial evolutionary algorithm (TEA)** [25] that well models the geographical structure of solutions and demonstrates promising performance in solving 2D spatial optimization problems. Nevertheless, TEA encounters deficiencies when applied to the PPCO problem. For instance, PPCO is inherently large-scale, resulting in intensive tensor computations during the individual-based evolution of TEA, leading to computational inefficiency. Consequently, we have devised a more efficient method in TEIM, specifically tailored to address the challenges posed by the large-scale PPCO problem. Furthermore, we integrate a spatial smoothness component into the evaluation process to leverage the valuable spatial correlation of image pixels, which promotes spatial continuity of the estimated image, and ultimately generates high-quality alpha mattes.

The main contributions are summarized as follows:

(1) As far as we know, this is the first work that models the matting problem as a **spatial pixel pair combinatorial optimization (S-PPCO)** problem. The S-PPCO problem integrates pixel-level color distortion, pixel-level spatial closeness, and spatial smoothness into optimization. The last term significantly improves the performance of the S-PPCO problem as traditional 1D methods are unintuitive and inefficient to utilize the spatial information.
(2) We introduce a spatial-aware TEIM method to tackle the S-PPCO problem based on tensorial evolutionary optimization. All individuals are stacked into a third-order tensor where each tensor slice denotes an F or B individual. We devise three tensorial evolutionary operators, in which three types of evolutionary tensors are designed for the evolution based on pure tensor computations. Thereby, all dimensions of the population are evolved simultaneously, which realizes the explicit parallelism and operation efficiency.
(3) We develop a superpixel-based trimap expansion method, in which a voting strategy is carefully designed to expand the F/B regions toward U region. The superpixel-based trimap expansion significantly reduces the number of unknown pixels and brings relatively low alpha errors.
(4) We compare TEIM with state-of-the-art matting methods. Experimental results demonstrate the effectiveness and efficiency of TEIM.

The remainder of the article is outlined as follows: Section 2 reviews the related work. Section 3 formulates the S-PPCO problem, and Section 4 presents TEIM to tackle the S-PPCO problem. Experiments are conducted in Section 5. Finally, conclusions are made in Section 6.

2 RELATED WORK

Generally, natural image matting methods are classified into trimap-based and trimap-free methods. The former adopts a user-defined trimap and input, whereas the latter uses interactive tools to supply semantic prior information for solving the matting problem [11, 47]. As illustrated in Figure 3, the trimap-based methods can be further subdivided into traditional matting and deep matting [18, 28, 29]. The trimap-based methods can be more granularly classified into PPCO-based matting [32, 33, 37], propagation-based matting [2, 7, 27], or hybrids [8]. A succinct summary of the key characteristics of these aforementioned methods is provided in Table 1. Our research endeavors are focused on addressing the trimap-based PPCO problem. Under this scope, we further categorize PPCO-based models into sampling and heuristic methods. This section meticulously

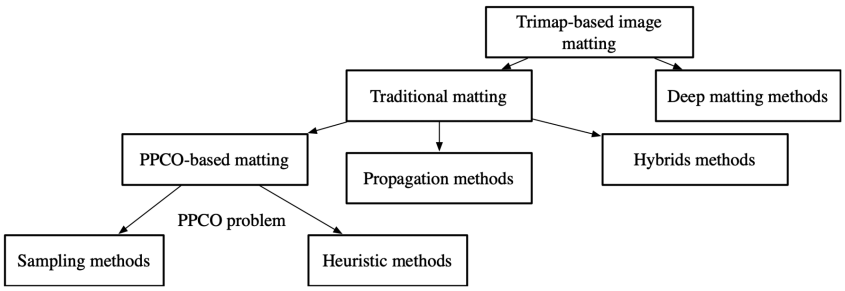


Fig. 3. Classification of trimap-based matting methods.

Table 1. Summary of Trimap-based Image Matting Methods

Category of Methods	Characteristic	Pros. & Cons.	Representatives
Propagation methods	assumption of color distribution	fast speed; only suitable for images that conform to the color distribution	[2, 9, 42, 52]
Sampling methods	different sampling regions and sampling techniques	fast speed; unstable sampling quality; fail to cover the optimal pixel pair	[13, 19, 16, 23, 41]
Heuristic methods	various heuristic algorithms	no sampling; robust; high precision; slow speed	[12, 31, 33, 34, 52]
Deep Matting methods	end-to-end network	high precision; lack of data to train the network; high cost of data generation	[18, 29]

reviews and analyzes both types of methods. Afterward, we investigate the evaluation methods for measuring the quality of F/B solutions, which is also a critical component of PPCO.

2.1 Sampling-based Matting Methods

Sampling-based matting methods aim to select the optimal F/B solution from the F/B search space using various sampling strategies [13, 16, 23, 40, 41, 45]. He et al. [16] selected the best F/B color pairs from the entire boundary of F and B regions. Gastal et al. [15] collected a subset of F/B colors for each unknown pixel from several line segments starting from the unknown pixel itself. Shahrian et al. [41] comprehensively constructed different search spaces for different unknown pixels based on the distances between the unknown pixels and the known regions. Karacan et al. [23] developed a sparse subset selection approach that performs selection from the mean colors of superpixels. Feng et al. [13] used sparse encoding to collect a set of representative samples to avoid missing the underlying F/B color pairs. Chen et al. [22] utilized edge detection to collect more potential pixel pairs from both the edges of objects and boundaries of images. Furthermore, a hybrid sampling and learning-based matting approach was proposed in Reference [43] that use deep neural networks to estimate the colors.

2.2 Heuristic Matting Methods

In early studies, heuristic optimizers were primarily used to improve the performance of sampling-based methods [4, 35]. The values of F and B decision variables are obtained from the sampled F and B regions of a trimap, respectively. The sampling strategies directly affect the quality of the alpha matte, since they construct different search spaces. For instance, one of the latest works

developed a pixel-level discrete multi-objective sampling method for image matting [19]. However, the sampling approach would inevitably result in missing the true F/B colors. Therefore, recent heuristic matting methods directly construct the search space from the entire F/B regions [33, 34].

Later, advanced optimizers appeared to solve the PPCO problem without sampling [32, 37, 49]. The approaches are mainly divided into two categories depending on whether a decomposition strategy is utilized in the optimizer. The earliest study of heuristic matting optimization was introduced in Reference [31, 48], where the PSO optimizer was applied, and a variant of PSO, namely, PSO-CSC, was then proposed in Reference [35]. Cai et al. [5] developed an optimizer based on the earthworm optimization algorithm. In Reference [33], the authors proposed a PSO-ACSC optimizer to balance the convergence speed. Other optimizers like DE, ACO, and CSO were also adopted in References [4, 50]. The aforementioned methods fall under the non-decomposition category. In contrast, for the decomposition-based methods, several **cooperative coevolution (CC)** optimizers were developed, where the PPCO problem was decomposed into subproblems by grouping the unknown pixels. Cai et al. [4] introduced a CC-DE optimizer where similar pixels in the U region were grouped using image segmentation. Besides, Liang et al. [34] developed a multi-objective optimizer based on decomposition and fuzzy multicriteria evaluation that significantly improved the matting accuracy. The unknown pixels are sampled at regular intervals, with only half of the pixels solved by the optimizer. Liang et al. [49] employed an image pyramid structure, where the CSO optimizer was employed to solve the topmost layer with a very small-size image. Feng et al. [12] utilized RGB-clustering to group unknown pixels and proposed a **grouping-based collaborative swarm optimizer (GC-CSO)** that only solves representative unknown pixels in each cluster to efficiently process high-resolution images. Liu et al. [37] proposed a group optimization algorithm for image matting that transforms the large-scale PPCO problem into multiple small-scale optimization problems. Liang et al. [32] proposed to use the optimal pixel pair of the surrogate models to approximate the optimal pixel pair of PPOP.

2.3 Pixel Pair Evaluation Methods

Pixel pair evaluation metrics are proposed to assess the quality of F/B color pairs for each unknown pixel to select the best F/B color, contributing to obtaining a high-quality alpha matte. For example, the authors in References [9, 42] used a maximum likelihood measure to estimate the F/B colors of unknown pixels, assuming that image colors follow a certain probability distribution. The color distortion objective was initially introduced in Reference [45], which computes the difference between the true color of unknown pixels and the estimated color derived from the F/B pixel pairs. Shahrian et al. [44] argued that relying solely on color features may be inadequate when images exhibit very similar color distributions. To address this limitation, they incorporated texture features into the evaluation procedure to better distinguish pixel pairs. Gastal et al. [15] also considered both color and texture features within the objective function, extending the evaluation granularity from the pixel-level to local regions. Karacan et al. [23] proposed a local contexture similarity metric based on superpixels to enhance pixel pair evaluation. Furthermore, the authors of Reference [34] proposed a fuzzy multicriteria evaluation method to quantify the uncertainty of color distortion and spatial closeness. More recently, Liang et al. [32] put forward to utilize surrogate models to approximate the quality of pixel pairs, which greatly enhance the efficiency of pixel pair evaluation.

3 THE S-PPCO PROBLEM

The existing heuristic matting methods typically model the matting problem as a PPCO problem, in which each decision variable stands for the F or B solution of an unknown pixel as shown

in Equation (2). However, this pixel-level optimization is suboptimal and inefficient. More importantly, spatial correlations among pixels are often overlooked, especially when nearby pixels share similar characteristics. The 1D representation of PPCO hinders the exploitation of spatial information, and also leads to operational inefficiency. Hence, we propose extending the PPCO problem to a Spatial-PPCO problem for leveraging the intrinsic spatial structure of images. In this section, we first present the formulation of the S-PPCO problem and then introduce the optimization objective of S-PPCO based on the spatial-aware representation of F/B solutions.

3.1 Definition of S-PPCO

The S-PPCO problem is a type of 2D spatial optimization problem that seeks optimal 2D F/B solutions of an image by leveraging the spatial property of the image. The S-PPCO problem represents all known pixels based on the inherent 2D representation of images, and optimizes them as a whole. In this view, we formulate the S-PPCO problem as

$$\begin{aligned}\min G(X) &= G(X^{(F)}, X^{(B)}) \\ &= G\left(\begin{pmatrix} x_{11}^{(F)} & x_{12}^{(F)} & \cdots & x_{1N}^{(F)} \\ x_{21}^{(F)} & x_{22}^{(F)} & \cdots & x_{2N}^{(F)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1}^{(F)} & x_{M2}^{(F)} & \cdots & x_{MN}^{(F)} \end{pmatrix}, \begin{pmatrix} x_{11}^{(B)} & x_{12}^{(B)} & \cdots & x_{1N}^{(B)} \\ x_{21}^{(B)} & x_{22}^{(B)} & \cdots & x_{2N}^{(B)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1}^{(B)} & x_{M2}^{(B)} & \cdots & x_{MN}^{(B)} \end{pmatrix}\right),\end{aligned} \tag{3}$$

where $G(\cdot)$ denotes the objective function, $X^{(F)}$ takes the form of a 2D matrix containing the foreground decision variables for all the unknown pixels, and similarly $X^{(B)}$ consists of the background decision variables. The search space for $X^{(F)}$ and $X^{(B)}$ is determined by the F and B regions in the trimap, respectively. The subscripts in Equation (3) indicate the spatial coordinates of image pixels, and $M \times N$ denotes the size of a trimap.

In the S-PPCO problem, we gather all the pixels from the F, B, and U region separately, and stored the information of the three types of pixels in the F, B, and U sets, respectively. The F, B, and U sets are defined as

$$FS = \left\{p_1^{(F)}, p_2^{(F)}, \ldots, p_{N_F}^{(F)}\right\}, \tag{4}$$

$$BS = \left\{p_1^{(B)}, p_2^{(B)}, \ldots, p_{N_B}^{(B)}\right\}, \tag{5}$$

$$US = \left\{p_1^{(U)}, p_2^{(U)}, \ldots, p_{N_U}^{(U)}\right\}, \tag{6}$$

where N_F, N_B, and N_U count the number of pixels in the F, B, and U regions, respectively. Each pixel is represented by a 5D vector containing the color feature $p^{(C)}$ and spatial feature $p^{(S)}$. For example, a pixel in the U region is denoted as $p^{(U)} = (p^{(C,U)}, p^{(S,U)})^T$. The F and B sets constitute the search space for the S-PPCO problem where the decision variables in $X^{(F)}$ and the decision variables in $X^{(B)}$ is in the range of $[1, N_F]$ and $[1, N_B]$, respectively. The U set determines the decision space where the true F/B decision variables are located.

3.2 Objective of S-PPCO

Most existing heuristic matting methods evaluate the quality of each F/B color pair based on two objectives: color distortion and spatial closeness, both of which are evaluated at the pixel level. Nevertheless, the valuable spatial correlation among pixels is barely considered, resulting in low quality of the estimated alpha mattes and the inefficiency of evaluation. Spatial smoothness is frequently considered in image field for delaminating image noises and promoting spatial continuity

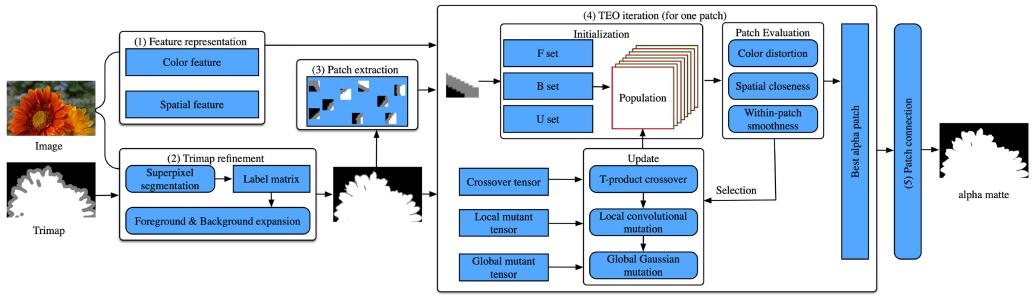


Fig. 4. Overall procedures of TEIM.

of image pixels [46]. Therefore, we propose a new evaluation metric that integrates three components: pixel-level color distortion (f_{pcd}), pixel-level spatial closeness (f_{psc}), and spatial smoothness (f_s). The last spatial term computes the total variation loss based on the spatial-aware representation of the S-PPCO problem to promote the spatial continuity of solutions. The overall objective function is defined as

$$G(X^{(F)}, X^{(B)}) = f_{pcd} + \beta f_{psc} + \gamma f_s, \tag{7}$$

where β and γ are trade-off parameters to balance different components, and the formulations of f_{pcd}, f_{psc}, and f_s will be detailed in the next section. The aim of S-PPCO is to minimize $G(\cdot)$ to find the optimal alpha matte.

4 TENSORIAL EVOLUTIONARY IMAGE MATTING

This article proposes a novel spatial-aware tensorial evolutionary image matting algorithm TEIM to solve the S-PPCO problem by **tensorial evolutionary optimization (TEO)**. The overall pipeline of TEIM is illustrated in Figure 4, which consists of five major modules:

- —**Feature Representation:** represent each pixel by a 5D feature vector consists of the color and spatial information. The 5D feature vector is defined as $p = (p^{(C)}, p^{(S)})^T$ where the $p^{(C)}$ stands for the RGB values and $p^{(S)}$ denotes the spatial coordinates.
- —**Trimap Refinement:** refine the original trimap based on superpixels, which narrows the decision space (see Section 4.1).
- —**Patch Extraction:** divide the refined trimap into equal-sized trimap patches to decompose the decision space (see Section 4.2).
- —**TEO Iteration:** optimize the trimap patches, and output the resultant alpha patches (see Section 4.3).
- —**Patch Connection:** enforce the edge connectivity among alpha patches (see Section 4.4).

4.1 Trimap Refinement

Most existing matting methods include a pre-processing step to expand the F/B regions toward U region according to the color similarity, which would directly reduce the number of unknown pixels. However, this method requires a pre-defined threshold of the color similarity that may fail when the color distinction of foreground and background is vague. This way, the expanded region generates wrong F/B labels, resulting in unexpected errors in the alpha mattes directly. Considering the characteristics of the image alpha channel, typically only the boundaries of the true F and B regions would undergo significant changes within the open interval $\alpha \in (0, 1)$, where the main difficulty of the matting problem lies. Meanwhile, the majority of the F or B regions correspond to determined alpha values of 1 or 0. Image superpixels provide an effective way to

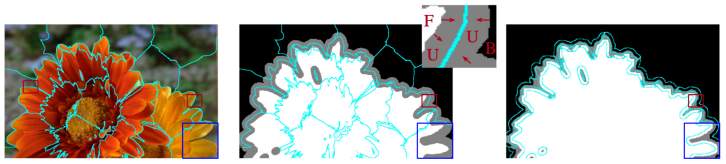


Fig. 5. Illustration of superpixel-based trimap expansion, where the green lines represent the boundaries of superpixels. (a) SLIC segmentation of an image, (b) superpixels on the trimap, (c) the refined trimap.

identify the boundaries of F/B regions. Therefore, we develop a superpixel-based trimap expansion method that effectively reduces the number of decision variables. It facilitates the subsequent matting optimizer to locate the real difficulty of the problem and reduce the computational cost.

To generate the superpixel-refined trimap, we first obtain the superpixels using an efficient SLIC segmentation method [1], and the segmentation results are stored in a label matrix L. Then, we expand the known F/B regions based on a voting strategy within each superpixel. Specifically, if an unknown pixel is not located on the boundary of the superpixel, it is directly assigned with an F or B label based on the label information of its neighbors. The voting strategy is implemented as follows:

(1) **Collect the superpixel labels and F/B labels of neighboring pixels:** The superpixel labels determine whether the unknown pixel would be assigned with a label, while the F/B labels determine whether the assigned label is F or B. Specifically, for an unknown pixel k, there are eight neighboring pixels located in a 3×3 neighborhood. The overall superpixel labels, $L1_k^{all} = \{L1_k^{(1)}, L1_k^{(2)}, \ldots, L1_k^{(8)}\}$, are collected from the label matrix L by shifting a 3×3 matrix one unit along all the eight directions. Similarly, the F/B labels of these neighboring pixels are collected from the original trimap. The overall F/B labels are defined as $L2_k^{all} = \{L2_k^{(1)}, L2_k^{(2)}, \ldots, L2_k^{(8)}\}$.

(2) **Assign F/B label to the unknown pixel:** If the neighboring labels in $L1_k^{all}$ are not consistent, then the kth pixel locates on the boundary of a superpixel, and the label remains unknown. Otherwise, the kth pixel belongs to the inner region of a superpixel, and will be assigned with a label according to $L2_k^{all}$. In this case, if all the neighboring labels in $L2_k^{all}$ are F or B, the kth pixel is directly assigned with the F (set $\alpha = 1$) or B label (set $\alpha = 0$), respectively.

Figure 5 illustrates an example of superpixel-based trimap expansion. As shown in Figure 5(b), the boundaries of superpixels go through the U region. The expansion starts from the edge of the F/U region or the B/U region, and gradually moves toward the superpixel boundaries. Note that although the boundaries of superpixels may be inaccurate in some cases (see the blue square in Figure 5(a)), the above superpixel-based trimap expansion would skip inaccurate regions where the unknown pixels remain unknown. In this way, the refined trimap directly narrows the U region to reduce the decision space and expands the F/B regions to enlarge the search space.

4.2 Patch Extraction

Then, we divide the refined trimap into multiple patches to decompose the S-PPCO problem. Only the patches containing unknown pixels need to be further considered. In addition, each trimap patch can be optimized in parallel to improve efficiency. In this view, patch extraction is a useful tool to alleviate the burden of subsequent searching and effectively speed up problem-solving. Note

that we also introduce a patch connection procedure in TEIM, detailed in Section 4.4, to mitigate the side effects resulting from the dividing-and-conquering of the patches.

After patch extraction, the S-PPCO problem is divided into several subproblems. Each subproblem is still a S-PPCO problem that handles only one trimap patch. The search spaces for each subproblem remain the same as the whole S-PPCO problem, as defined in Equations (4) and (5). The decision space is decomposed depending on the U region of a trimap patch, and each subproblem has a different number of decision variables. The U set for the pidth patch ($pid \in [1, N_{patch}]$ where N_{patch} is the total number of unknown patches) is defined as $US_{pid} = \{p_1, p_2, \ldots, p_{N_u}\}$ where N_u counts the number of unknown pixels in the current patch. In this setting, the choice of patch size is an important issue to be considered. If the patch size is too small, then the local spatiality of the image might be destroyed by patch division. Contrarily, if the patch size is too large, then it would be more easily to be trapped into local optimal within the feasible time. Further analysis on the effect of patch size would be discussed in Section 5.

4.3 TEO Iteration

The entire procedure of the spatial-aware tensorial evolutionary optimizer is illustrated in Figure 4. The input for TEO is a trimap patch. The TEO iteration contains three main components: the initialization component that generates the initial population, the evaluation component that measures the quality of pairs of F/B individuals, the update component that contains three tensorial evolutionary operators for breeding new generations. The three components are elaborated in this section.

4.3.1 Initialization. Existing 1D matting methods represent F/B solutions by a long vector with the dimensionality of $2 \times N_U$, where odd positions stand for $x_k^{(F)}$ and even positions are for $x_k^{(B)}$ with $k \in [1, N_U]$, $x_k^{(F)} \in [1, N_F]$, and $x_k^{(B)} \in [1, N_B]$. Differently in TEIM, each candidate solution for the trimap patches is represented by a pair of 2D individuals, and hence the spatial structure of images is preserved inherently. The individuals are classified into F or B individuals, since they have different search spaces. The F/B individuals are formulated as

$$X^{(F)} = \begin{pmatrix} x_{11}^{(F)} & \cdots & x_{1n}^{(F)} \\ \vdots & \ddots & \vdots \\ x_{m1}^{(F)} & \cdots & x_{mn}^{(F)} \end{pmatrix}, X^{(B)} = \begin{pmatrix} x_{11}^{(B)} & \cdots & x_{1n}^{(B)} \\ \vdots & \ddots & \vdots \\ x_{m1}^{(B)} & \cdots & x_{mn}^{(B)} \end{pmatrix}, \tag{8}$$

where $x_{ij}^{(F)} \in [1, N_F]$, $x_{ij}^{(B)} \in [1, N_B]$, and $m \times n$ is the size of the trimap patch. The sizes of F/B individuals are consistent with the patch size.

The whole population is stacked into a 3D tensor $pop \in \mathbb{R}^{m \times n \times NP}$, where NP is the population size. As shown in Figure 4 (4), the F/B individuals in the population are distinguished by red and green. Each frontal slice of the population tensor stands for an F or B individual, which is defined as

$$pop(:, :, t) = \begin{cases} X_t^{(F)}, & if\ t\ is\ odd, \\ X_t^{(B)}, & if\ t\ is\ even, \end{cases} \tag{9}$$

where $t \in [1, NP]$ denotes the tth individual in the population. However, the initialization of population is not done at this stage. The problem is that not all pixels in the patch need to be solved. If a pixel locates at position (i, j) belongs to F or B region, then the true alpha solution is already known. Therefore, we propose to utilize this information. For an F individual $X_t^{(F)}$, if (i, j) belongs to F region, $x_{ij}^{(F)}$ is determined by Equation (1), since the true color is directly determined by the

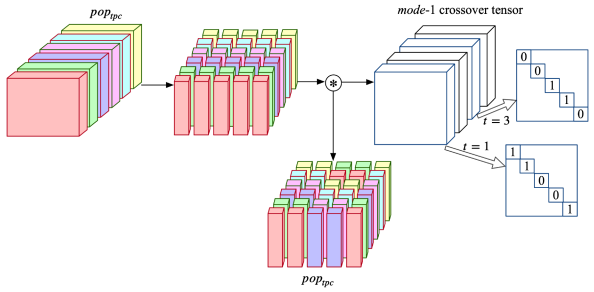


Fig. 6. *Mode*-1 crossover (The starting point $sp = 3$ and the length parameter $sl = 2$). The *mode*-1 crossover tensor determines the exchange of *mode*-1 fibers. For clarity, we use different colors to track how the fibers are swapped. The F and B individuals are marked by red and green lines.

foreground value with $\alpha = 1$. Similarly, for the B individual, if (i, j) is located in B region, $x_{ij}^{(B)}$ is also known, since the true color is uniquely determined by the background value with $\alpha = 0$.

4.3.2 Update. The update process consists of three tensorial evolutionary operators: T-product crossover, local convolutional mutation, and global Gaussian mutation.

T-product crossover: The crossover operator recombines different individuals to generate the offspring. Given the crossover probability P_{tpc}, a percentage of P_{tpc} individuals from population is collected and superposed to a crossover population tensor through an indexing operation. The crossover population is denoted as $pop_{tpc} \in \mathbb{R}^{m \times n \times NP_c}$ where $NP_c = NP \times P_{tpc}$. The crossover operator is implemented as

$$pop_{tpc} = \begin{cases} pop_{tpc} * \mathcal{T}_{tpc1}, & if\ \theta \leq 0.5, \\ \mathcal{T}_{tpc2} * pop_{tpc}, & if\ \theta > 0.5, \end{cases} \tag{10}$$

where $\mathcal{T}_{tpc1} \in \mathbb{R}^{n \times n \times NP_c}$ and $\mathcal{T}_{tpc2} \in \mathbb{R}^{m \times m \times NP_c}$ denote the *mode*-1 crossover tensor and *mode*-2 crossover tensor, respectively. The random $\theta \in [0, 1]$ controls the selection of *mode*-1 or *mode*-2 crossovers on the population. The *mode*-1 crossover tensor is defined as

$$\mathcal{T}_{tpc1}(i, i) = \begin{cases} 0, & for\ i = \langle sp \rangle_n, \ldots, \langle sp + sl \rangle_n, \\ 1, & for\ all\ other\ i \in [1, n], \end{cases} \tag{11}$$

where $t \in [1, NP_c]$ represents the tth frontal slice, $\langle \cdot \rangle_n$ is a modular function over n. Similarly, the *mode*-2 crossover tensor is defined as

$$\mathcal{T}_{tpc2}(i, i) = \begin{cases} 0, & for\ i = \langle sp \rangle_m, \ldots, \langle sp + sl \rangle_m, \\ 1, & for\ all\ other\ i \in [1, m], \end{cases} \tag{12}$$

where $\langle \cdot \rangle_m$ is a modular function over m. Figure 6 presents an example of the mode-1 crossover. The crossover operation imitates the interaction between any two of the individuals, and two slices of the crossover tensor are required to manipulate the interaction. Note that the F/B individuals are superposed in an alternating order. Since the interaction only occurs among the same type of individuals, the corresponding two slices of the crossover tensors are separated one slice unit apart, for example $t = 1$ and 3 in Figure 6. The two slices must satisfy the condition that $\mathcal{T}_{tpc1}^{(t=1)}(i, i) + \mathcal{T}_{tpc2}^{(t=3)}(i, i) = 1$ ($t = 2$ and 4 also works and generates the same result).

Local convolutional mutation: The local mutation performs the t-product-based convolution to mutate each gene based on the topology of individuals. Given a mutation probability P_{lcm}, a number of F/B individuals are selected from the population tensor. Similarly to the crossover

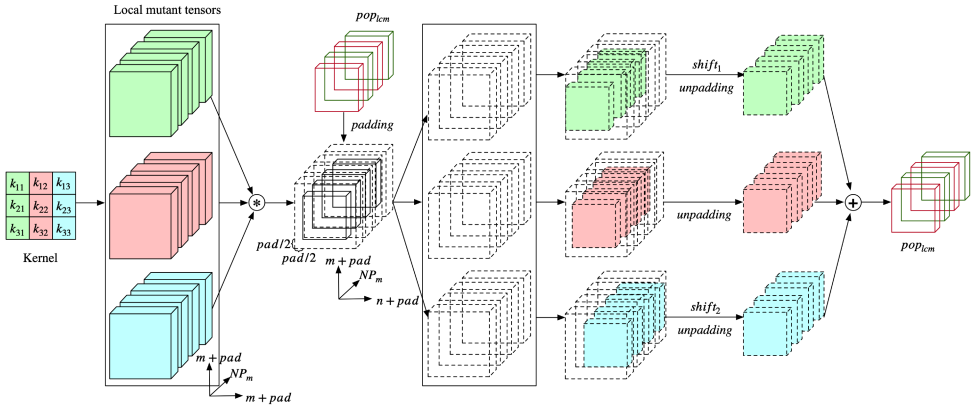


Fig. 7. Local mutation. The intermediate states of the local mutant population are marked by dashed lines. The three local mutant tensors are marked by different colors for distinction. The F and B individuals are marked by red and green lines.

operator, these individuals are stacked into a local mutant population tensor denoted as $pop_{lcm} \in \mathbb{R}^{m \times n \times NP_m}$, where $NP_m = NP \times P_{lcm}$. The local mutation operator is then implemented

$$\begin{aligned} pop_{lcm} = & \, unpadding(shift_1(\mathcal{T}_{lcm1} * pop_{lcm_1})) \\ & + unpadding(\mathcal{T}_{lcm2} * pop_{lcm_1}) \\ & + unpadding(shift_2(\mathcal{T}_{lcm3} * pop_{lcm_1})), \end{aligned} \tag{13}$$

where $pop_{lcm1} = padding(pop_{lcm})$, and $\mathcal{T}_{lcm1}, \mathcal{T}_{lcm2}, \mathcal{T}_{lcm3} \in \mathbb{R}^{(m+pad)\times(m+pad)\times NP_m}$ denote the three local mutant tensors. The $shift_1$ operation horizontally rotates all the genes one unit to the right, while the $shift_2$ operation rotates all the genes one unit to the left. Figure 7 illustrates the overall procedures of the local mutation. First, the local mutant population goes through a padding operation to facilitate the subsequent handling of genes located on the boundaries. Then, a t-product of the three local mutant tensors and the population is performed to obtain the offspring.

The local mutant tensors store the sampling locations and corresponding weights of a linear kernel. Only one slice is required to store the weights, since genes are mutated on the F or B individual itself. The local mutant tensor is defined as

$$\mathcal{T}_{lcm}^{(t)}(i, i-1:i+1) = K(:, j), \tag{14}$$

where $i = 2, \ldots, (m + pad - 1)$, $j \in [1, n_k]$, and $t \in [1, NP_m]$ represents the tth frontal slice. As shown in Figure 8, the value of j equals to 1, 2, and 3 for the local mutant tensors $\mathcal{T}_{lcm1}$, $\mathcal{T}_{lcm2}$, and $\mathcal{T}_{lcm3}$, respectively.

Global Gaussian mutation: The Gaussian distribution is performed on the population to improve the global exploration ability of TEO. It is implemented by the addition of the global mutant tensor and the population under a given probability P_{ggm}. The global mutation is defined as

$$pop = pop + \mathcal{T}_{ggm}, \tag{15}$$

where $\mathcal{T}_{ggm}$ is the global mutant tensor with the same size of the population. The global mutant tensor stores the information of locations and the degree of Gaussian distribution for the mutated genes. It is formulated as

$$\mathcal{T}_{ggm} = \mathbb{I}\left(R(0,1) < P_{ggm}\right) \times \left(G(0, 0.1) \times (ub - lb)\right), \tag{16}$$

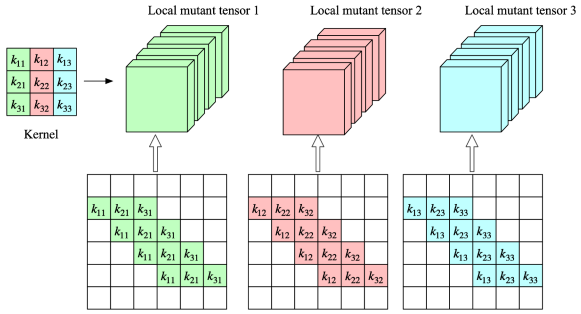


Fig. 8. Structure of local mutant tensors. The weights are distributed on the three local mutant tensors, marked by green, red, and blue, respectively.

where $\mathbb{I}$ is an indicator function, $R(0,1)$ generates a random tensor in $[0,1]$ with the same size of the population, and $G(0,0.1)$ generate a random Gaussian tensor with 0 mean and 0.1 standard deviation. The lb is set to one for both F/B individuals, while $ub = N_F$ for F individuals and $ub = N_B$ for B individuals.

4.3.3 Patch Evaluation. Each pair of F/B individuals is evaluated using Equation (7). However, the $X^{(F)}$ and $X^{(B)}$ are adjusted to the candidate F/B solutions for a trimap patch. Consequently, the first two evaluation terms of f_{pcd} and f_{psc} only consider the unknown pixels within one trimap patch, while the last spatial term f_s is also restricted to the range of a trimap patch. Here f_s is re-defined as f_{wps} that evaluates the within-patch smoothness by computing the total variation loss to promote the spatial consistency within an alpha patch. The three terms f_{pcd}, f_{psc}, and f_{wps} in patch evaluation are formulated as

$$f_{pcd}(X^{(F)}, X^{(B)}) = \sum p_{ij}^{(C,U)} - \hat{p}^{(C)}_{x_{ij}^{(F)}, x_{ij}^{(B)}}, \tag{17}$$

$$f_{psc}(X^{(F)}, X^{(B)}) = \frac{||p_{ij}^{(S,U)} - p^{(S,F)}_{x_{ij}^{(F)}}||}{\min_{(a,b)} ||p_{ij}^{(S,U)} - p_{ab}^{(S,F)}||} + \frac{||p_{ij}^{(S,U)} - p^{(S,B)}_{x_{ij}^{(B)}}||}{\min_{(a,b)} ||p_{ij}^{(S,U)} - p_{ab}^{(S,B)}||}, \tag{18}$$

$$f_{wps}(X^{(F)}, X^{(B)}) = \frac{\sum \hat{p}^{(C)}_{i(j+1)} - \hat{p}^{(C)}_{ij}}{m \times n} + \frac{\sum \hat{p}^{(C)}_{(i+1)j} - \hat{p}^{(C)}_{ij}}{m \times n}, \tag{19}$$

where $i, a \in [1, m]$, $j, b \in [1, n]$, and $\hat{p}^{(C)}$ is the estimated RGB color computed by Equation (1) using the F/B solution $(x_{ij}^{(F)}, x_{ij}^{(B)})$ and the estimated alpha value.

After evaluation, individuals are selected with probabilities proportional to their objective values, to form the new generation. During each TEO iteration, the population undergoes the aforementioned update and selection procedures iteratively until the maximum evaluation number is reached. Ultimately, the alpha matte is derived from the optimal F/B individual for the corresponding trimap patch.

4.4 Patch Connection

In this module, all alpha patches are connected to obtain the final alpha matte with the primary focus on edge smoothness. Since TEO optimizes trimap patches independently, only within-patch smoothness is considered during patch optimization. Therefore, the smoothness across patches is

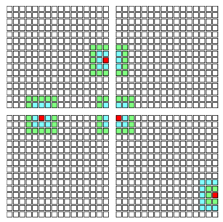


Fig. 9. Patch connection. The boundary pixels (marked by red) are refined by its eight neighbors (marked by blue) located in 3×3 regions.

taken into account to further refine boundaries between adjacent patches. Based on the connectivity among adjacent patches, the optimal F/B pair for each boundary pixel is refined by considering all F/B solutions from its neighbors, as depicted in Figure 9, and selecting the best one.

4.5 GPU Parallelism of TEIM

Traditional evolutionary algorithms typically represent an individual as a 1D vector, and a population consists of a set of 1D vectors. The offspring are generated individual by individual until the population size is reached. Moreover, the decision variables are also executed dimension by dimension, making it difficult to adjust the conventional evolutionary operators into fast multidimensional data operations for GPU parallel acceleration [14, 51].

In contrast, the TEO iteration in TEIM achieves explicit individual-level and dimensional-level parallelism. It defines the entire population as a third-order tensor rather than 2D individuals. The *mode*-1 and *mode*-2 directions of the population tensor represent the dimensions of an individual, and the *mode*-3 direction represents different F/B individuals. This structure is natural for multidimensional data operations and parallel implementation. The individuals participating in the crossover and mutation operators are packed into an evolutionary population tensor. Thereby, all individuals and all dimensions of individuals execute simultaneously in each generation owing to the parallel routine of tensor operations.

4.6 Complexity Analysis

The pseudocode of TEIM is outlined in Algorithm 1. Given the image of size T, the complexity of superpixel segmentation and trimap expansion is $O(T)$ according to Reference [30]. The complexity of fitness evaluation is $O(psize)$ where $psize$ indicates the size of patches. The construction of the crossover, local mutation, and global mutation tensors are all bit-wise operations, and hence the costs are $O(psize \times NP_c)$, $O(psize \times NP_m)$, and $O(psize \times NP)$, respectively. The overall complexity for generating these evolutionary tensors can be written as $O(psize \times NP)$, since $NP_c = NP \times P_{tpc}$ where $P_{tpc} < 1$ and $NP_m = NP \times P_{lcm}$ where $P_{lcm} < 1$. The core operations of TEO are the t-product and addition calculations. According to Reference [24], the crossover operator is at most the crossover operator is at most $O(psize \times \sqrt{psize} \times NP_c)$ complex, and the local mutation operator is at most $O(psize \times \sqrt{psize} \times NP_m)$ complex on the corresponding populations of pop_{tpc} and pop_{lcm}. The global mutation operator performs bitwise addition, and hence the complexity is $O(psize \times NP)$. The overall complexity of the TEIM is at most $O(T + psize^{3/2} \times NP)$.

5 EXPERIMENTS

In this section, we conduct experiments to validate the effectiveness of TEIM. We first introduce the experimental settings. Afterward, we compare TEIM with state-of-the-art heuristic matting methods and several representative non-heuristic matting methods separately. Finally, we also provide discussions to enhance the understanding of TEMI.

Algorithm 1: TEIM

```
/* Initialization */
SLIC segmentation: label matrix L;
Expand the original trimap: Tri;
Extract trimap patches from Tri: patch 1, 2, ..., N_patch;
Initialize the search space: FS, BS;
For pid = 1 : N_patch
    /* TEO for One Patch */
    Initialize the decision space for the pidth patch: US_pid;
    Initialize the population within the search range: pop;
    repeat /* Main Loop */
        Evaluate each pair of F/B individuals by Equation (7);
        Generate crossover tensors by Equations (11) and (12), and perform crossover by Equation (10);
        Generate local mutant tensors by Equation (14), and perform mutation by Equation (13);
        Generate global mutant tensor by Equation (16), and perform mutation by Equation (15);
    until maximum evaluations is reached;
End
Connect all the alpha patches and post-process the final alpha matte;
return The alpha matte.
```

5.1 Experimental Setup

Dataset: The experiments are conducted on a popular benchmark image matting dataset [39] including two image sets, denoted as IS_1 and IS_2. The IS_1 consists of 27 images groups where each image group constitutes the original color image and their corresponding trimaps with different U regions. The IS_1 has two types of trimaps with small and large U regions, and the ground truth alpha mattes of 27 images are available for evaluation. While the IS_2 has 8 images groups that contain three types of trimaps with small, large, and user-defined U regions. The estimated alpha mattes of images in IS_2 are evaluated online[1] to compare with other methods.

Competing methods:

- —Heuristic matting methods include PSO [31], PSO-CSC [35], PSO-ACSC [33], CSO [12], CC_1-PSO [12], CC-DE [4], GC-CSO [12], PM [49], MOEA [34], IMTO-ACO [50], and TEA [25].
- —Non-heuristic matting methods include CWCT [40], SCM [21], KLSM [23], SC-CSM [13], GSM [22], PBM [6], TSM [8], ATPM [52], GCA [29], and LA2U [10].

The parameters of all competing methods are set as recommended in the original paper. The maximum number of fitness evaluations in our TEIM method is set to 5000, and the probabilities of crossover, local mutation, and global mutation are set as the same in Reference [25]. Three widely used measures, including **sum of absolute differences (SAD)**, **mean squared error (MSE)**, and **gradient error (GE)** [39], are employed to evaluate the quality of alpha mattes produced by different matting methods. All experiments are conducted ten times independently and the average results are recorded.

5.2 Performance Comparison

We compare our TEIM with all competing methods in this section, where both numerical and visual comparisons are presented.

[1] http://www.alphamatting.com/

Table 2. Results of Alpha Mattes Obtained by Different Matting Methods Using Image Set IS_1

Method	SAD			MSE			GE		
	Overall	Small	Large	Overall	Small	Large	Overall	Small	Large
PSO	27.0526	22.3221	31.7831	0.0365	0.0297	0.0433	20.2073	15.8862	24.5284
PSO-CSC	26.1570	21.5487	30.7653	0.0348	0.0282	0.0413	19.6515	15.4307	23.8722
PSO-ACSC	26.7035	22.0137	31.3934	0.0357	0.0290	0.0424	19.9493	15.6467	24.2519
CSO	11.6948	9.5357	13.8539	0.0145	0.0112	0.0177	7.0144	5.1362	8.8926
CC_1-PSO	11.7742	9.6105	13.9379	0.0146	0.0113	0.0179	7.0213	5.1438	8.8989
CC-DE	11.2738	9.0204	13.5272	0.0116	0.0088	0.0144	6.0628	4.2296	7.8961
GC-CSO	11.6342	9.4981	13.7702	0.0143	0.0111	0.0176	6.9081	5.0508	8.7654
PM	10.4410	8.4742	12.4079	0.0087	0.0067	0.0106	5.6895	4.1052	7.2737
MOEA	8.9538	6.9293	10.9783	0.0065	0.0045	0.0084	4.3236	2.8020	5.8452
IMTO-ACO	7.9836	5.9748	9.9924	0.0060	0.0041	0.0078	3.8517	2.3338	5.3697
TEA	9.2194	7.3913	11.0475	0.0069	0.0052	0.0086	4.6937	3.0234	6.3640
TEIM (ours)	**6.9984**	**5.0929**	**8.9040**	**0.0054**	**0.0034**	**0.0074**	**3.4883**	**1.9941**	**4.9825**

5.2.1 Numerical Comparison. In the first experiment, we compare TEIM with the 11 heuristic matting methods using image set IS_1. The average SAD, MSE, and GE values over all images in IS_1 using both small and large trimaps are provided in Table 2, and the overall evaluation of the three measures is also computed. The detailed results of all the 27 images are also provided in the *Supplemental_Material*. We find that PSO, PSO-CSC, and PSO-ACSC exhibit the worst performance. Meanwhile, CC-DE, CC_1-PSO, and GC-CSO achieve similar moderate performance. Besides, we have two important observations. First, methods that have an expanded search space, such as CSO, GC-CSO, and IMTO-ACO, tend to outperform those that rely on the original trimap. More possible F/B colors enrich the search space. Second, methods that reconstruct the decision space tend to perform better than those with the original decision space. Nearby pixels in an image tend to have similar properties, and using different decision space construction methods such as grouping in PM and MOEA greatly improve the matting performance by better incorporating these properties. Overall, TEIM achieves the best results on all image cases, because it further enhances the quality of alpha mattes with the carefully designed TEO procedure. Furthermore, Figure 10 showcases box-plot diagrams that vividly depict the distribution of SAD values across all images in IS_1. The results underscore the consistent trend of TEIM outperforming other competing methods, as it consistently delivers lower SAD values across almost all cases. Most significantly, it further substantiates the superior stability of TEIM as the interquartile range is much smaller compared to other competitors.

In the second experiment, the TEIM is compared with the other ten non-heuristic matting methods using IS_2. Since the resultant alpha mattes of these ten methods have been provided online,[1] and all the competing methods have already ranked based on SAD, MSE, and GE. Therefore, the alpha mattes obtained by TEIM are uploaded for online ranking. Note that to enable a fair comparison, we equip the TEIM tested on IS_2 with a postprocessing process [15], which is employed in the competing methods and further enhances the quality of alpha mattes. The average rankings of alpha mattes over all images in IS_2 obtained by different methods are tabulated in Table 3. Moreover, an overall ranking of the three measures using three types of trimaps is also provided. The detailed results of all images in IS_2 are provided in the *Supplemental_Material*. In terms of SAD ranking, TEIM scores 2.250, 1.875, and 2.375 on the small, large, and user trimaps, respectively. Among all competing methods, TEIM ranks the second on both the small and large trimaps and ranks the first on the user-defined trimaps. For the

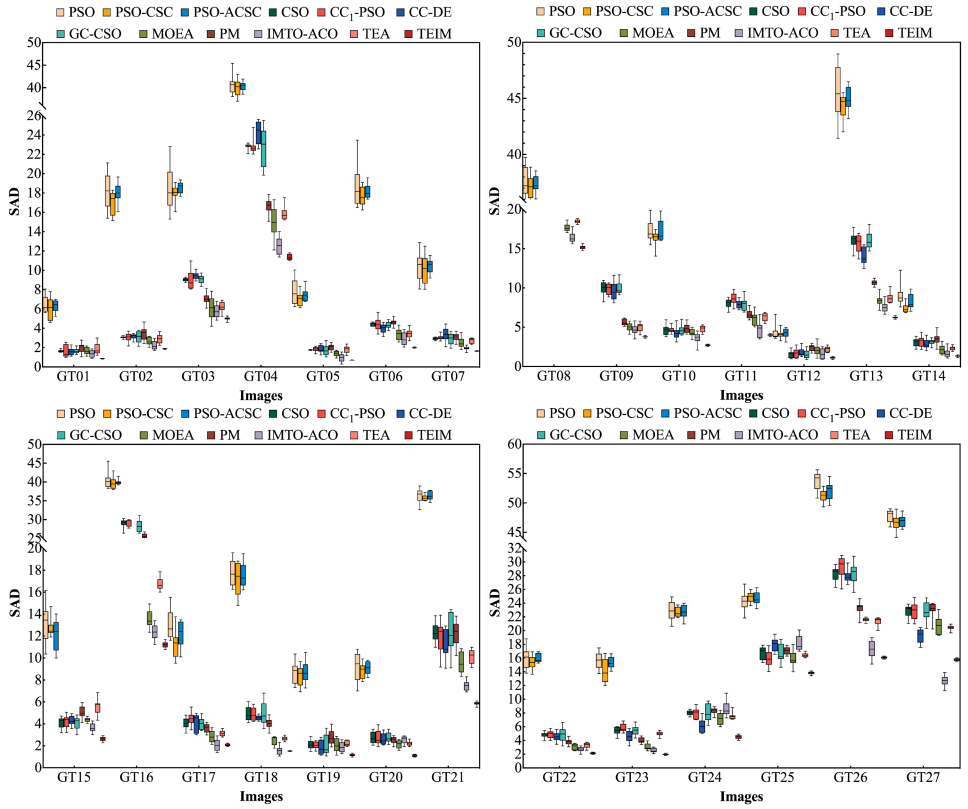


Fig. 10. Comparison of different image matting methods regarding SAD on the IS_1 dataset.

Table 3. Average Rankings of Different Matting Methods on SAD, MSE, and GE Using Image Set IS_2

Method	SAD				MSE				GE			
	Overall	Small	Large	User	Overall	Small	Large	User	Overall	Small	Large	User
CWCT	7.167	7.125	7.250	7.125	3.458	3.000	3.875	3.500	3.708	3.250	4.375	3.500
SCM	6.875	5.125	7.375	8.125	2.917	2.000	3.375	3.375	3.375	3.250	3.250	3.625
KLSM	6.667	5.750	7.125	7.125	3.167	2.500	3.250	3.750	4.083	4.000	4.000	4.250
SC-CSM	6.583	6.250	6.875	6.625	3.333	2.750	3.750	3.500	3.792	3.375	3.875	4.125
GSM	6.167	6.875	5.125	6.500	3.375	3.125	3.375	3.625	4.208	4.375	3.500	4.750
PBM	7.500	8.250	7.500	6.750	3.708	3.250	4.375	3.500	3.458	3.500	3.875	3.000
TSM	5.708	5.375	6.125	5.625	2.500	1.875	2.750	2.875	3.750	3.750	4.000	3.500
ATPM	5.542	6.250	6.000	4.375	2.750	2.750	3.375	2.125	4.000	4.250	4.500	3.250
GCA	2.542	2.875	2.250	2.500	1.542	1.250	1.500	1.875	1.667	1.750	1.625	**1.625**
LA2U	**2.167**	**1.625**	**1.750**	3.125	1.500	**1.125**	**1.250**	2.125	**1.458**	**1.250**	**1.250**	1.875
TEIM (ours)	**2.167**	2.250	1.875	**2.375**	**1.458**	1.250	1.625	**1.500**	1.500	**1.250**	1.625	**1.625**

The best results are in bold, the second-best are underlined.

MSE measure, our TEIM also ranks the first, while the performance of LA2U and GCA is also appealing. In GE evaluation, TEIM achieves the second best overall ranking among all methods. In sum, our TEIM obtains highly competitive results over the other non-heuristic matting methods.

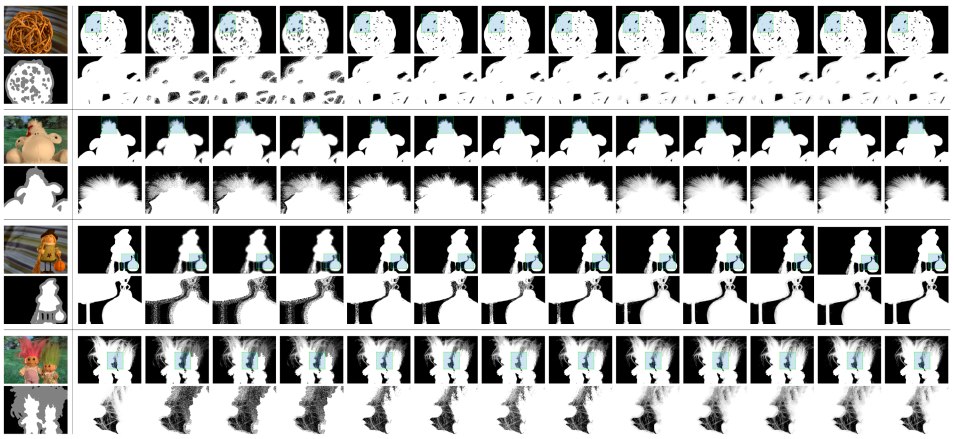


Fig. 11. Results of Alpha Mattes. For each image case, the first row is the alpha mattes obtained by different methods, and the second row is the zoomed-in region marked by green. From left to right: original image, ground-truth, PSO, PSO-CSC, PSO-ACSC, CSO, CC_1-PSO, CC-DE, GC-CSO, PM, MOEA, IMTO-ACO, TEA, and our TEIM.

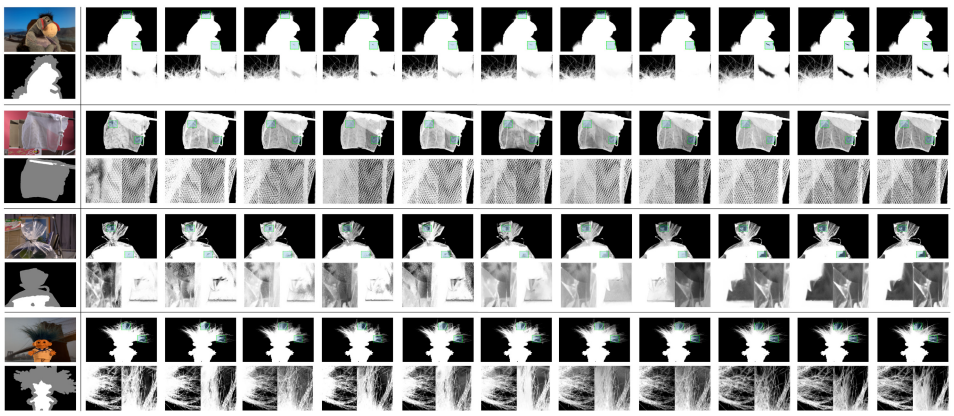


Fig. 12. Results of Alpha Mattes. For each image case, the first row is the alpha mattes obtained by different methods, and the second row is the zoomed-in region marked by green. From left to right: CWCT, SCM, KLSM, SC-CSM, GSM, PBM, TSM, ATPM, GCA, LA2U, and our TEIM.

5.2.2 Visual Comparison. Figure 11 presents a visual comparison of the 11 heuristic matting methods on five typical images in IS_1, and Figure 12 shows the alpha mattes of four representative images in IS_2 produced by the other 10 non-heuristic matting methods. From the results, we observe that the alpha mattes produced by our TEIM method are more precise and clearer than other methods. TEIM has less noise than the others, especially near the boundaries of the foreground objects.

5.3 Further Investigation and Discussion

5.3.1 Runtime Analysis. To verify the efficiency of tensorial evolutionary computations in TEIM, we compare the empirical runtime of different heuristic optimizers. We fix the population size and the maximal evaluation time for all competing algorithms to ensure a fair comparison. The results are presented in Figure 13, which depicts the average runtimes (in seconds) over all

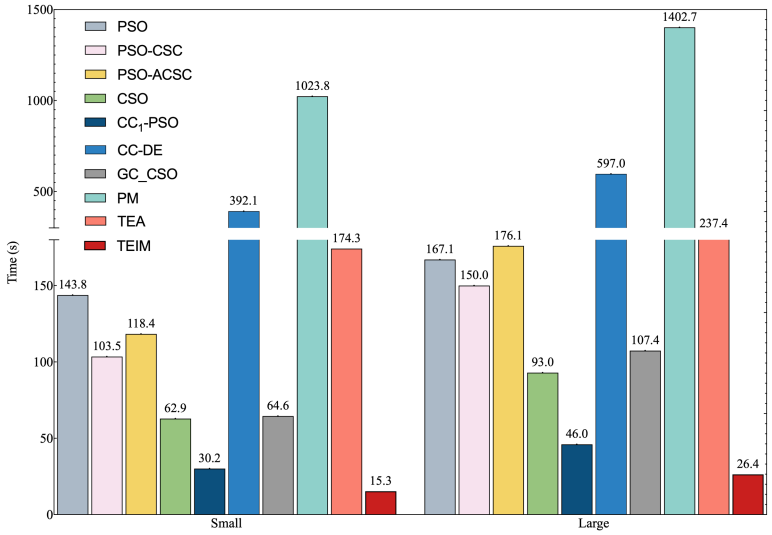


Fig. 13. Time costs (in seconds) of different heuristic matting methods.

images using two types of trimaps. Notice that the IMTO-ACO is excluded from comparison, since the heuristic optimizer is not used for pixel pair optimization. The MOEA that requires performing multiobjective optimization on each pixel is not shown, since it takes running times far more than the others (over 30,000 s) due to the huge number of unknown pixels.

Among the four methods PSO, CSO, PSO-CSC, and PSO-ACSC, CSO is the most competitive over the other three methods. The PSO-CSC and PSO-ACSC adopt different convergence controllers based on PSO optimization. The performance of CC-DE, CC_1-PSO, and GC-CSO with different grouping strategies vary significantly due to the imbalanced distribution of pixel groups. The problem is that some groups may contain only a few pixels, while others may contain thousands of pixels, leading to disparities in running times. The PM uses the image pyramid to reduce the number of unknown pixels to be optimized while the extra pyramid construction and pixel propagation times are added compared to other methods. The TEA is more time-consuming than TEIM, since its operators are burdened with intensive tensor operations. Overall, our TEIM method shows the lowest empirical runtime, benefiting from the natural tensorial representation and explicit parallel implementation based on tensor operations.

5.3.2 Ablation Study. To gain a comprehensive understanding of the various modules within TEIM, we conducted ablation studies to further validate the effectiveness of our approach. In this experiment, we incrementally add different modules of TEIM one by one: **superpixel-based trimap expansion (STE)**, **patch extraction (PE)**, within-patch smoothness loss (f_{wps}) as defined in Equation (7), and **patch connection (PC)**, to assess the individual contributions of each module to the overall performance of TEIM. The ablation results shown in Table 4 validate the effectiveness of different modules in TEIM. Specifically, the STE module facilitates the subsequent optimization by reducing the number of decision variables, and we have also noted that the U region of trimaps is narrowed by around 20% to 50%. When the PE module is added, both SAD and MSE would experience a significant growth. Moreover, the PE procedure would also accelerate TEIM. The f_{wps} component in the objective evaluation aims to improve the color smoothness within the alpha patches, While the PC module is to refine the connecting boundaries of adjacent alpha patches. Both the f_{wps} and the PC module contribute to obtaining a higher-quality alpha matte.

Table 4. Ablation Results of TEIM with Different Variants Using Small, Large, and User Trimaps in IS_2

Variation	Small		Large		User	
	SAD	MSE	SAD	MSE	SAD	MSE
STE	10.26	0.58	12.43	0.71	12.19	0.70
STE+PE	9.20	0.49	10.34	0.56	10.54	0.55
STE+PE+f_{wps}	8.39	0.40	9.51	0.48	9.74	0.58
STE+PE+f_{wps}+PC	7.62	0.39	8.95	0.45	9.53	0.49

Table 5. Results of TEIM with Different Configurations of Patch Size on Five Image Cases

		$\text{TEIM}_{8\times8}$	$\text{TEIM}_{16\times16}$	$\text{TEIM}_{32\times32}$	$\text{TEIM}_{64\times64}$	$\text{TEIM}_{128\times128}$	TEIM_{tri}
GT02	N_{patch}	1391	500	210	88	25	1
	Runtime	2.55	3.45	7.54	36.97	56.29	87.69
	MSE	54.4	54.6	56.7	57.7	56.9	64.9
GT04	N_{patch}	3659	974	290	96	42	1
	Runtime	2.73	3.96	7.71	38.85	60.61	103.94
	MSE	278.1	277.7	277.1	276.0	282.3	302.5
GT07	N_{patch}	979	317	111	54	26	1
	Runtime	2.21	3.25	7.89	28.13	51.52	96.21
	MSE	26.6	25.5	26.4	25.5	27.3	32.9
GT13	N_{patch}	2881	868	274	91	37	1
	Runtime	2.69	3.87	7.24	30.91	61.48	106.08
	MSE	187.6	171.0	177.2	180.2	186.7	201.8

5.3.3 Influence of Patch Size. The patch size is an important factor regarding both the effectiveness and efficiency of TEIM, which directly determines the number of patches to be optimized and the range of decision space. Therefore, we analyze the influence of patch size (*psize*) in this section. The TEIM is set with different configurations of *psize* for comparison: $\text{TEIM}_{8\times8}$, $\text{TEIM}_{16\times16}$, $\text{TEIM}_{32\times32}$, $\text{TEIM}_{64\times64}$, $\text{TEIM}_{128\times128}$, TEIM_{tri}, where TEIM_{tri} is a special case with only one patch. We report three major factors associated with different *psize*: the number of patches (N_{patch}) for a given trimap, the average runtime for the tensorial evolutionary optimizer to handle one patch, and the MSE value of the resulting alpha matte. The influence of *psize* is reported in Table 5. In terms of runtime, it would experience a growth as the patch size increases. A larger patch means the increasing evolution time during generations, since the time complexity of TEIM is proportional to the individual size (i.e., *psize*), as discussed in Section 4.6. Regarding MSE, it is observed that for relatively small patch sizes (less than 16×16), the MSE values tend to degrade as the patch size increases. However, when the patch size becomes much larger, the MSE values start to increase. Several reasons can be attributed to these observations. On the one hand, when the patch size is too small, the local optimality of the image would be significantly affected, since its local spatiality is destroyed by the patch division. Even with the post-processing patch connection in TEIM, it still becomes challenging to compensate for the loss of global optimal performance. On the other hand, using a larger trimap patch or even the entire trimap directly enlarges the decision space. It might increase the probability of finding a suboptimal solution within a feasible time. Taking all factors into consideration, a patch size of 16×16 is chosen in the experiment.

5.3.4 Limitation and Future Directions. The proposed TEIM has demonstrated promising performance for solving the PPCO-based image matting problem. However, it still faces several limitations. First, the inherent nature of PPCO-based methods formulates image matting as a color problem, seeking different linear combinations of foreground and background colors. This reliance on color features is highly effective for images with distinct color contrasts. However, TEIM becomes sensitive and struggles to produce accurate alpha mattes when dealing with images where the foreground and background colors are ambiguous or overlapping. Second, similar to other PPCO-based methods, TEIM heavily relies on the known regions provided by the trimap. When the known region in the trimap is small or even missing, there may not be enough color information to synthesize the color of pixels in the unknown region. These properties hinder TEIM from handling more complex images. Recently, deep matting is anticipated to offer solutions in the mentioned scenarios, and combining heuristic and deep matting techniques holds promise for further improving matting performance.

Furthermore, current research in image matting primarily concentrates on improving matting quality for various color images but often overlooks the impact of varying trimaps. At present, trimap annotation predominantly relies on manual user input and reducing the burden of user labeling is a critical research direction. However, existing automatic trimap annotation methods may occasionally lead to a reduction or loss of valuable information within the trimap. Therefore, a major future trend in natural image matting is to enhance the robustness of matting algorithms to adapt to different trimaps and ensure high-quality matte even when provided with very limited known information. This encompasses not only improving the methods themselves regarding robustness to more natural and complex environments but also developing more reliable and automated ways to generate trimaps that capture the essence of the image.

6 CONCLUSION

We designed a spatial-aware tensorial evolutionary image matting method, TEIM, to produce high-quality alpha mattes of images. To take full advantage of the spatial structure of images, we propose to formulate the image matting problem as a spatial combinatorial optimization problem and develop a novel tensorial evolutionary optimizer based on tensorial evolutionary optimization to solve the problem. Specifically, the candidate solutions for foreground and background colors are represented by tensor slices, while the entire population naturally forms a third-order tensor. Based on this spatial representation, we develop several tensorial evolutionary operators for breeding offspring under the full consideration of local spatiality, and all operators are computationally efficient with a natural parallel design and implementations. Experimental results validate the effectiveness of TEIM for natural image matting problem. Moving forward, we would investigate the application of TEIM into more complex image matting environments.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 11 (2012), 2274–2282.

[2] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. 2017. Designing effective inter-pixel information flow for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 29–37.

[3] Benish Amin, Muhammad Mohsin Riaz, and Abdul Ghafoor. 2020. Automatic aircraft extraction using video matting and frame registration. *IET Image Process.* 14, 8 (2020), 1628–1635.

[4] Zhao-Quan Cai, Liang Lv, Han Huang, Hui Hu, and Yi-Hui Liang. 2017. Improving sampling-based image matting with cooperative coevolution differential evolution algorithm. *Soft Comput.* 21 (2017), 4417–4430.

[5] Zhao-Quan Cai, Liang Lv, Han Huang, and Yi-Hui Liang. 2019. A discrete bio-inspired metaheuristic algorithm for efficient and accurate image matting. *Memet. Comput.* 11 (2019), 53–64.

[6] Guangying Cao, Jianwei Li, Zhiqiang He, and Xiaowu Chen. 2016. Divide and conquer: A self-adaptive approach for high-resolution image matting. In *Proceedings of the IEEE International Conference on Virtual Reality and Visualization (ICVRV'16)*. IEEE, 24–30.
[7] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. 2013. KNN matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 9 (2013), 2175–2188.
[8] Xiao Chen, Fazhi He, and Haiping Yu. 2017. A three-stage matting method. *IEEE Access* 5 (2017), 27732–27739.
[9] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. 2001. A Bayesian approach to digital matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, Vol. 2. IEEE, II–II.
[10] Yutong Dai, Hao Lu, and Chunhua Shen. 2021. Learning affinity-aware upsampling for deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*6841–6850.
[11] Henghui Ding, Hui Zhang, Chang Liu, and Xudong Jiang. 2022. Deep interactive image matting with feature propagation. *IEEE Trans. Image Process.* 31 (2022), 2421–2432.
[12] F. Feng, Han Huang, Q. Wu, X. Ling, Y. Liang, and Z. Chai. 2020. An alpha matting algorithm based on collaborative swarm optimization for high-resolution images. *Sci. Sin. Inform.* 50, 3 (2020), 424–437.
[13] Xiaoxue Feng, Xiaohui Liang, and Zili Zhang. 2016. A cluster sampling method for image matting via sparse coding. In *Proceedings of the European Conference on Computer Vision (ECCV'16)*. Springer, 204–219.
[14] Luis Pedro García, Javier Cuenca, and Domingo Giménez Cánovas. 2014. On optimization techniques for the matrix multiplication on hybrid CPU+ GPU platforms. *Ann. Par. GPU Prog.* 1, 1 (2014), 1–8.
[15] Eduardo S. L. Gastal and Manuel M. Oliveira. 2010. Shared sampling for real-time alpha matting. In *Proceedings of the Computer Graphics Forum*, Vol. 29. Wiley Online Library, 575–584.
[16] Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. 2011. A global sampling method for alpha matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2049–2056.
[17] Anna Katharina Hebborn, Nils Höhner, and Stefan Müller. 2017. Occlusion matting: Realistic occlusion handling for augmented reality applications. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR'17)*. IEEE, 62–71.
[18] Qiqi Hou and Feng Liu. 2019. Context-aware image matting for simultaneous foreground and alpha estimation. In *Proceedings of the IEEE International Conference on Computer Vision.* 4130–4139.
[19] Han Huang, Yihui Liang, Xiaowei Yang, and Zhifeng Hao. 2019. Pixel-level discrete multiobjective sampling for image matting. *IEEE Trans. Image Process.* 28, 8 (2019), 3739–3751.
[20] Han Huang, Liang Lv, Shujin Ye, and Zhifeng Hao. 2019. Particle swarm optimization with convergence speed controller for large-scale numerical optimization. *Soft Comput.* 23 (2019), 4421–4437.
[21] Jubin Johnson, Deepu Rajan, and Hisham Cholakkal. 2014. Sparse codes as Alpha Matte. In *Proceedings of the British Machine Vision Conference (BMVC'14)*, Vol. 1. Citeseer, 5.
[22] Jubin Johnson, Ehsan Shahrian Varnousfaderani, Hisham Cholakkal, and Deepu Rajan. 2016. Sparse coding for alpha matting. *IEEE Trans. Image Process.* 25, 7 (2016), 3032–3043.
[23] Levent Karacan, Aykut Erdem, and Erkut Erdem. 2015. Image matting with KL-divergence based sparse sampling. In *Proceedings of the IEEE International Conference on Computer Vision.* 424–432.
[24] Misha E. Kilmer and Carla D. Martin. 2011. Factorization strategies for third-order tensors. *Linear Alg. Appl.* 435, 3 (2011), 641–658.
[25] Si-Chao Lei, Xiaolin Xiao, Yue-Jiao Gong, Yun Li, and Jun Zhang. 2022. Tensorial evolutionary computation for spatial optimization problems. *IEEE Trans. Artif. Intell.* (2022).
[26] Dawa Chyophel Lepcha, Bhawna Goyal, and Ayush Dogra. 2021. Image matting: A comprehensive survey on techniques, comparative analysis, applications and future scope. *Int. J. Image Graph.* (2021), 2350011.
[27] Anat Levin, Alex Rav-Acha, and Dani Lischinski. 2008. Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 10 (2008), 1699–1712.
[28] Jizhizi Li, Jing Zhang, and Dacheng Tao. 2023. Deep image matting: A comprehensive survey. Retrieved from https:// arXiv:2304.04672
[29] Yaoyi Li and Hongtao Lu. 2020. Natural image matting via guided contextual attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11450–11457.
[30] Zhengqin Li and Jiansheng Chen. 2015. Superpixel segmentation using linear spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1356–1363.
[31] Lv Liang, Huang Han, Cai Zhaoquan, and Hu Hui. 2015. Using particle swarm large-scale optimization to improve sampling-based image matting. In *Proceedings of the Genetic and Evolutionary Computation Conference.* 957–961.
[32] Yihui Liang, Hongshan Gou, Fujian Feng, Guisong Liu, and Han Huang. 2023. Natural image matting based on surrogate model. *Appl. Soft Comput.* 143 (2023), 110407.
[33] Yihui Liang, Han Huang, and Zhaoquan Cai. 2020. PSO-ACSC: A large-scale evolutionary algorithm for image matting. *Front. Comput. Sci.* 14 (2020), 1–12.

[34] Yihui Liang, Han Huang, Zhaoquan Cai, and Zhifeng Hao. 2019. Multiobjective evolutionary optimization based on fuzzy multicriteria evaluation and decomposition for image matting. *IEEE Trans. Fuzzy Syst.* 27, 5 (2019), 1100–1111.
[35] Yihui Liang, Han Huang, Zhaoquan Cai, and Liang Lv. 2018. Particle swarm optimization with convergence speed controller for sampling-based image matting. In *Intelligent Computing Theories and Application*. Springer, 656–668.
[36] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. 2022. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 238–247.
[37] Shuang Liu, Fujian Feng, Hongshan Gou, Zhulian Zhou, Man Tan, and Lin Wang. 2021. Grouping optimization algorithm for natural image matting. In *Proceedings of the International Conference on Computer Science, Electronic Information Engineering, and Intelligent Control Technology (CEI'21)*. 421–426.
[38] Xiaofei Qin and Zepei Fan. 2019. Initial matting-guided visual tracking with siamese network. *IEEE Access* 7 (2019), 41669–41677.
[39] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. 2009. A perceptually motivated online benchmark for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1826–1833.
[40] Ehsan Shahrian and Deepu Rajan. 2012. Weighted color and texture sample selection for image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 718–725.
[41] Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. 2013. Improving image matting using comprehensive sampling sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 636–643.
[42] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. 2004. Poisson matting. In *Proceedings of the ACM SIGGRAPH*. 315–321.
[43] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. 2019. Learning-based sampling for natural image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3055–3063.
[44] Ehsan Shahrian Varnousfaderani and Deepu Rajan. 2013. Weighted color and texture sample selection for image matting. *IEEE Trans. Image Process.* 22, 11 (2013), 4260–4270.
[45] Jue Wang and Michael F. Cohen. 2007. Optimized color sampling for robust matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.
[46] Rui Wang, Jun Xie, Jiacheng Han, and Dezhen Qi. 2022. Improving deep image matting via local smoothness assumption. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 1–6.
[47] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Hanqing Zhao, Weiming Zhang, and Nenghai Yu. 2021. Improved image matting via real-time user clicks and uncertainty estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 15374–15383.
[48] Xueming Yan, Zhifeng Hao, and Han Huang. 2018. Alpha matting with image pixel correlation. *Int. J. Mach. Learn. Cyb.* 9 (2018), 621–627.
[49] Liang Yihui, Feng Fujian, and Cai Zhaoquan. 2020. Pyramid matting: A resource-adaptive multi-scale pixel pair optimization framework for image matting. *IEEE Access* 8 (2020), 93487–93498.
[50] Genji Yuan, Jinjiang Li, and Zhen Hua. 2021. Image matting trimap optimization by ant colony algorithm. *Multimed. Tools Appl.* 80, 4 (2021), 6143–6169.
[51] Zhi-Hui Zhan, Jun Zhang, Ying Lin, Jian-Yu Li, Ting Huang, Xiao-Qi Guo, Feng-Feng Wei, Sam Kwong, Xin-Yi Zhang, and Rui You. 2021. Matrix-based evolutionary computation. *IEEE Trans. Em. Top. Comp. I.* 6, 2 (2021), 315–328.
[52] Xiangyu Zhu, Ping Wang, and Zhenghai Huang. 2018. Adaptive propagation matting based on transparency of image. *Multimed. Tools Appl.* 77 (2018), 19089–19112.

Received 8 June 2023; revised 4 October 2023; accepted 5 February 2024