J. Vis. Commun. Image R. 25 (2014) 1031–1043

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Iterative transductive learning for automatic image segmentation and matting with RGB-D data

Bei He [a], Guijin Wang [b,*], Cha Zhang [c]

[a] Beijing National Railway Research & Design Institute of Signal & Communication Co., Ltd, China
[b] Department of Electronic Engineering, Tsinghua University, Beijing, China
[c] Microsoft Research, Redmond, USA

ARTICLE INFO

Article history:
Received 5 August 2013
Accepted 5 March 2014
Available online 13 March 2014

Keywords:
Image matting
Image segmentation
Transductive learning
Irregular neighboring patches
Adaptive laranger
Iterative optimization
RGB-D data
Kinect
Depth holes

ABSTRACT

In this paper, we propose a fully automatic image segmentation and matting approach with RGB-Depth (RGB-D) data based on iterative transductive learning. The algorithm consists of two key elements: robust hard segmentation for trimap generation, and iterative transductive learning based image matting. The hard segmentation step is formulated as a Maximum A Posterior (MAP) estimation problem, where we iteratively perform depth refinement and bi-layer classification to achieve optimal results. For image matting, we propose a transductive learning algorithm that iteratively adjusts the weights between the objective function and the constraints, overcoming common issues such as over-smoothness in existing methods. In addition, we present a new way to form the Laplacian matrix in transductive learning by ranking similarities of neighboring pixels, which is essential to efficient and accurate matting. Extensive experimental results are reported to demonstrate the state-of-the-art performance of our method both subjectively and quantitatively.

1. Introduction

Image matting addresses the problem of extracting foreground objects from images accurately. It has been widely used for many applications such as background replacement, augmented reality and image editing [1]. As a simple mathematical model, given pixel i, its color value $\mathbf{c}_i$ can be represented as a linear combination of the foreground component $\mathbf{f}_i$ and the background component $\mathbf{b}_i$, i.e.,

$$\mathbf{c}_i = \alpha_i \mathbf{f}_i + (1 - \alpha_i)\mathbf{b}_i, \tag{1}$$

where the alpha matte $\alpha_i \in [0, 1]$ represents the opacity of the foreground.

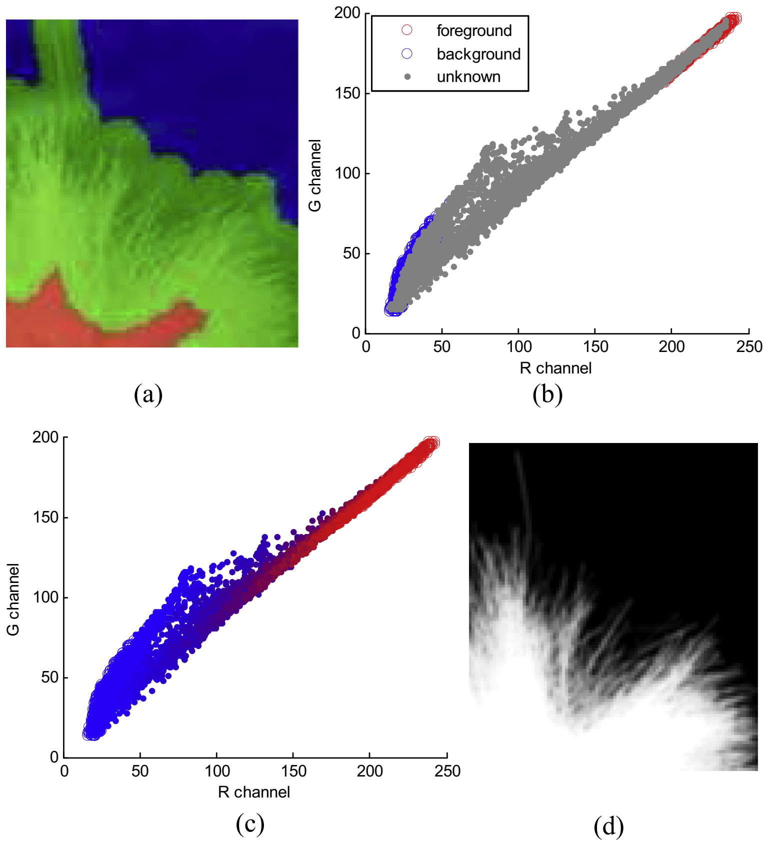
Since normally an image sensor can only observe the mixed color $\mathbf{c}_i$, image matting is in general an ill-conditioned problem. Early approaches require special hardware setup to solve the problem, for instance, blue screen matting [2] and flash matting [3]. Recently, various schemes have been proposed to use manually specified trimaps to help restrict and solve the matting problem. A trimap segments the input image into 3 regions: definite foreground $\mathcal{R}_f$, definite background $\mathcal{R}_b$ and the unknown region $\mathcal{R}_u$ (e.g., in Fig. 1(a), the foreground, background and unknown regions are labeled with red, blue and green, respectively). The goal is to extract the alpha matte of the unknown region by considering color similarities and neighboring continuities. Needless to say, the accuracy of the trimap is of paramount importance to the accuracy of image matting. While user interaction can often create satisfactory matting results, it is inconvenient and inapplicable under dynamic scenes.

In this paper, we study the problem of automatic image segmentation and matting with RGB-Depth (RGB-D) input data. This is a topic that has attracted much attention recently [4–8]. Compared with color image based segmentation and matting, the depth information can greatly help in challenging situations with illumination variations, camera shaking, foreground/background color similarity, etc. On the other hand, the problem is still non-trivial, since depth images from commodity depth sensors often suffer from various artifacts, including pixel-dependent noise, depth holes [4], edge fattening, etc.

We present a two-stage algorithm as illustrated in Fig. 2. A novel hard segmentation algorithm is first applied to iteratively refine the scene depth map and compute the foreground/background mask. After that, the trimap is generated automatically via

[*] Corresponding author.
E-mail addresses: coolhebei@163.com (B. He), wangguijin@tsinghua.edu.cn (G. Wang), chazhang@microsoft.com (C. Zhang).

http://dx.doi.org/10.1016/j.jvcir.2014.03.002
1047-3203/© 2014 Elsevier Inc. All rights reserved.



(a) (b) (c) (d)

Fig. 1. (a) Pixels in the foreground, background and unknown regions are labeled with red, blue and green, respectively. (b) Distribution of colors for the different regions. (c) After matting, the color distribution. (d) Matting result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

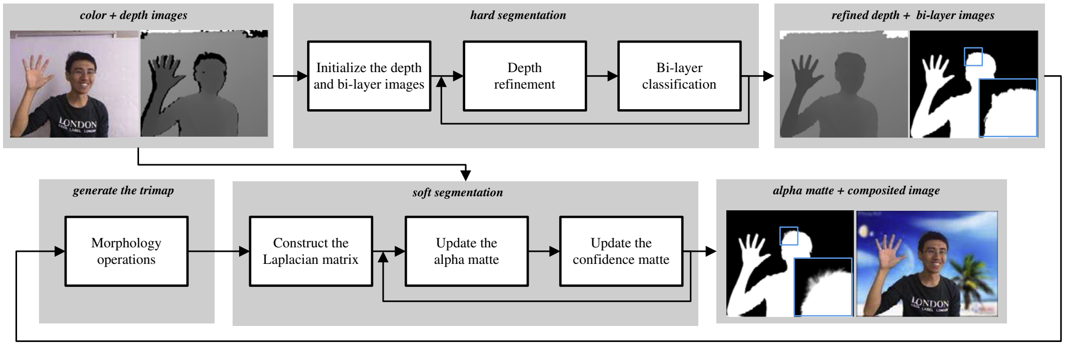

color + depth images
hard segmentation
Initialize the depth and bi-layer images
Depth refinement
Bi-layer classification
refined depth + bi-layer images
generate the trimap
Morphology operations
soft segmentation
Construct the Laplacian matrix
Update the alpha matte
Update the confidence matte
alpha matte + composited image

Fig. 2. Overview of the proposed algorithm. The hard-to-soft segmentation modules are presented to estimate the alpha matte. Enlarged regions refer to variant results of hard-segmentation and soft-segmentation.

morphological filters. Soft segmentation or matting is then used to calculate the alpha matte. The proposed matting algorithm is based on transductive learning [9], where the objective function is to minimize the affinity of the alpha matte, with labeled pixels served as constraints and unlabeled pixels closely integrated into the formulation. We demonstrate that it can achieve high accuracy through extensive experimental results.

The technical contribution of this work can be summarized as follows:

1. We formulate the hard-segmentation problem as a Maximum A Posterior (MAP) estimation problem, which iteratively performs depth refinement and bi-layer classification in a principled manner.
2. We introduce an adaptive weighting scheme between the objective function and the constraints under the transductive learning framework [9], which results in state-of-the-art accuracy for image matting.
3. We design a new Laplacian matrix that is based on extended neighboring pixels. Only pixels that have similar features are included in the Laplacian matrix, which dramatically improves accuracy and efficiency.

The rest of the paper is organized as follows. In Section 2, we review related works and motivate the proposed method. Details of our hard and soft segmentation algorithms are described in Sections 3 and 4, respectively. We give experimental results and discussions in Section 5, followed by a brief conclusion in Section 6.

2. Related work and motivation

2.1. Hard segmentation

Foreground and background segmentation has been an important research topic for many years. Here we only review recent schemes that involves RGB-D input data. The most straightforward scheme is to segment the image via depth value thresholding, as was done in Zhu et al. [5] and Cho et al. [6]. However, the performance was limited due to the difficulty in finding a good threshold for an arbitrary scene. In [4], by assuming that the background is static, Matyunin et al. constructed a background model in order to extract the foreground object. A more involved approach is to formulate the segmentation problem as a binary labeling problem in a 2D/3D random field graph, as was done in [10,7,8]. Such a problem can be solved with the well-known graph-cut optimization [11], which is a standard technique for image segmentation with color image input only.

One of the most challenging problems in RGB-D based image segmentation is the depth input which corrupts often. Due to various reasons such as occlusions and non-reflective surfaces, it is typical that we observe large regions of depth holes in the image (e.g., the hair region and the hand occluded background region in Fig. 2). In the graph-cut based scheme in [7], the color component can help bridging some of these depth artifacts. However, as shown in the experimental results, since the corrupted depth maps are directly formulated into the graph-cut framework, the method is still sensitive to depth errors. In this paper, we formulate the hard segmentation problem as a Maximum A Posterior (MAP) estimation problem, which iteratively performs depth refinement and bi-layer classification in a principled manner. Consequently, our approach is much more robust to depth corruptions.

2.2. Soft Segmentation

Hard segmentation is the first step of accurate foreground/background separation. To reduce artifacts when the foreground objects are composed with new background images, one needs to perform soft segmentation or matting to obtain the alpha matte. As mentioned earlier, one of the key techniques for image matting is to model the similarity between neighboring pixels. Levin et al. [12] presented a closed-form matting scheme that has been the foundation of many follow-up works. She designed the matting objective function as:

$$\begin{aligned} &\arg\min_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^T \cdot \mathbf{L} \cdot \boldsymbol{\alpha}, \\ &s.t.\ \mathbf{I}_k \boldsymbol{\alpha} = \boldsymbol{\alpha}_k^* \end{aligned} \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_P]^T$ represents the alpha values of all pixels, and P is the total number of pixels in the image. $\boldsymbol{\alpha}_k^*$ denotes the alpha matte marked by user interaction. The ith element in $\boldsymbol{\alpha}_k^*$ takes value 1 if $i \in \mathcal{R}_f$, and 0 if $i \in \mathcal{R}_b$. The diagonal matrix $\mathbf{I}_k$ is an indicator of all the labeled pixels. That is, its ith diagonal element takes value 1 if $i \in \mathcal{R}_f \cup \mathcal{R}_b$, and 0 otherwise.

The matrix $\mathbf{L}$ is referred to as the Laplacian matrix. Its elements describe the relationship between pixels. The above formulation has a closed form solution [12] by optimizing:

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

where λ_k is the Lagrange coefficient of labeled pixels.

Despite the simplicity of the formulation, challenges remain in determining the Laplacian matrix and the Lagrange multiplier. In Zheng and Kambhamettu [13] and Xiang et al. [14], non-linear models were used to describe the relationship between neighboring pixels in $\mathbf{L}$. However, only a small neighborhood is considered in this process, which requires strong regularization and causes the matting results to be overly smooth. As shown in Fig. 3, many pixels in hollow-out regions are wrongly judged as the foreground. More recently, Lee and Wu [15] and Chen et al. [16] alleviated the over-smoothness issue by enlarging the searching neighborhood when forming the Laplacian matrix, at the cost of high time and space complexity. In [17], He et al. presented a scheme to reduce the computational cost when solving the alpha matte; however, constructing the Laplacian matrix is still very expensive. Wang [9] introduced a transductive learning framework to improve matting accuracy for scenes with complex foreground and background textures. Since his approach considers all the neighborhood of a given pixel, low similarity neighbors would adversely affect the final matting performance. In this paper, we introduce a selective scheme for Laplacian matrix construction, which is shown to have both high accuracy and efficiency.

Another factor in closed-form matting is the choice of the Lagrange multiplier. Most existing approaches [12,13,15,9,16,17,14] use fixed Lagrange coefficients. Nevertheless, it is even impossible to find a single good value for one image. As shown in Fig. 4, the larger coefficient easily brings fuzzy boundaries (a), while the smaller one would induce the over-smoothness in hollowed-out regions (c). In this paper, we will present a novel iterative scheme to set the Lagrange coefficients adaptively.

3. Hard Segmentation

We first present our hard segmentation algorithm based on a novel MAP estimation framework, which jointly refines the depth map and performs bi-layer segmentation.

3.1. Problem formulation

Given a captured color image $\mathbf{c}^{cam}$ and its corresponding depth image $\mathbf{d}^{cam}$, we would like to estimate a refined depth map $\mathbf{d}$, and a bi-layer mask image $\mathbf{o}$, where $o_i = 1$ if pixel i belongs to the foreground and $o_i = 0$ if it belongs to the background. Mathematically, we maximize the posterior probability as follows:

$$\arg\max_{\mathbf{o},\mathbf{d}} \quad p(\mathbf{o}, \mathbf{d} | \mathbf{c}^{cam}, \mathbf{d}^{cam}). \tag{4}$$

The above joint estimation can be conducted through alternative maximization. Namely, we iteratively maximize the joint probability by fixing one unknown variable and letting the other variable adapt. According to the Bayesian rule, Eq. (4) can be re-written as:

$$\arg\max_{\mathbf{o},\mathbf{d}} \quad p(\mathbf{o} | \mathbf{c}^{cam}, \mathbf{d}^{cam}, \mathbf{d}) p(\mathbf{d} | \mathbf{c}^{cam}, \mathbf{d}^{cam}), \tag{5}$$

or

$$\arg\max_{\mathbf{o},\mathbf{d}} \quad p(\mathbf{d} | \mathbf{c}^{cam}, \mathbf{d}^{cam}, \mathbf{o}) p(\mathbf{o} | \mathbf{c}^{cam}, \mathbf{d}^{cam}). \tag{6}$$

Note $\mathbf{c}^{cam}$ and $\mathbf{d}^{cam}$ are known input variables. With Eq. (5), we may fix the hidden variable $\mathbf{d}$ to derive the optimal value of $\mathbf{o}$. Similarly, we can use Eq. (6) to derive the optimal value of $\mathbf{d}$ by fixing the value of $\mathbf{o}$.

The general procedure of our alterative maximization algorithm can be described as below. Note m is the index of iteration.

1. Fill “depth holes” to initialize the depth image $\mathbf{d}^{(0)} = \mathbf{d}^{init}$.
2. Fix the depth image $\mathbf{d}^{(m-1)}$ and estimate the bi-layer mask image by maximizing the posterior probability $p(\mathbf{o}^{(m)} | \mathbf{c}^{cam}, \mathbf{d}^{cam}, \mathbf{d}^{(m-1)})$ (named as *bi-layer classification*). Note in Eq. (5) the probability $p(\mathbf{d}^{(m-1)} | \mathbf{c}^{cam}, \mathbf{d}^{cam})$ is fixed once $\mathbf{d}^{(m-1)}$ is fixed.
3. Fix the bi-layer mask image $\mathbf{o}^{(m)}$ and refine the depth image by maximizing the posterior probability $p(\mathbf{d}^{(m)} | \mathbf{c}^{cam}, \mathbf{d}^{cam}, \mathbf{o}^{(m)})$ (named as *depth refinement*). This time $p(\mathbf{o}^{(m)} | \mathbf{c}^{cam}, \mathbf{d}^{cam})$ is fixed.

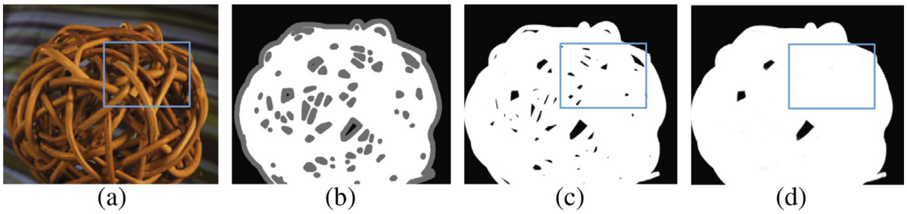


Fig. 3. (a)–(c) Refer to the color image, trimap and the ground truth, respectively. For pixels in hollowed-out regions (marked by the blue rectangle), over-smooth results (d) were provided [12–14]. That is, many pixels are wrongly judged as the foreground due to small neighborhood consideration and strong regularization.

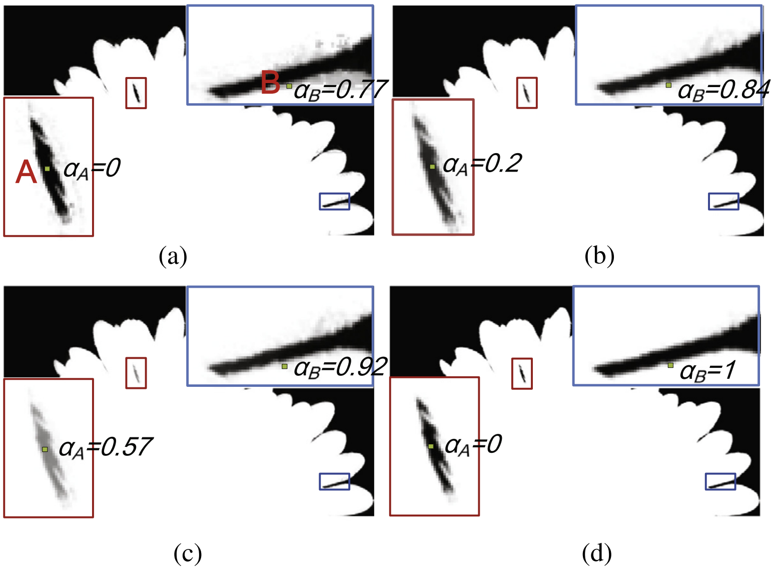


Fig. 4. Matting results using large (a), middle (b) and small (c) coefficients [9]. Compared with the ground truth in (d), the larger value would bring fuzzy boundaries (alpha values should be 1 in the blue rectangle) and the smaller value induces over-smoothness in hollowed-out regions (alpha values should be 0 in the red rectangle). We take the pixels *B* and *A* as an example. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Repeat steps (2) and (3) until the results do not change.

We explain how to compute the posterior probabilities $p(\mathbf{o}^{(m)}|\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)})$ and $p(\mathbf{d}^{(m)}|\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{o}^{(m)})$ in the following. For bi-layer classification, we have:

$$\begin{aligned}
&\arg\max_{\mathbf{o}^{(m)}} \; p(\mathbf{o}^{(m)}|\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)})\\
&\quad= \arg\max_{\mathbf{o}^{(m)}} \frac{p(\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)}|\mathbf{o}^{(m)})p(\mathbf{o}^{(m)})}{p(\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)})}\\
&\quad= \arg\max_{\mathbf{o}^{(m)}} p(\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)}|\mathbf{o}^{(m)})p(\mathbf{o}^{(m)}).
\end{aligned} \tag{7}$$

Using a Markov Random Field (MRF) model [18], the above problem can be optimized by minimizing an energy function as:

$$\arg\min_{\mathbf{o}^{(m)}} \sum_i (\lambda_o \mathcal{D}_o(o_i^{(m)}) + \sum_{j\in\mathcal{N}_i} \mathcal{V}_o(o_i^{(m)}, o_j^{(m)})), \tag{8}$$

where λ_o is a weight that balances the data term $\mathcal{D}_o(*)$ (which corresponds to the likelihood $p(\mathbf{c}^{cam},\mathbf{d}^{cam},\mathbf{d}^{(m-1)}|\mathbf{o}^{(m)})$) and smoothness term $\mathcal{V}_o(*)$ (which corresponds to the prior $p(\mathbf{o}^{(m)})$); $\mathcal{N}_i$ is a neighborhood of pixel *i*. Similarly, maximizing the posterior probability for depth refinement can be conducted by minimizing the following energy function:

$$\arg\min_{\mathbf{d}^{(m)}} \sum_i (\lambda_d \mathcal{D}_d(d_i^{(m)}) + \sum_{j\in\mathcal{N}_i} \mathcal{V}_d(d_i^{(m)}, d_j^{(m)})), \tag{9}$$

where $\mathcal{D}_d(*)$ and $\mathcal{V}_d(*)$ denote the data and smoothness terms, respectively. λ_d is the weight between the two terms. The detailed formulation of the different terms in Eqs. (8) and (9) will be described next.

As declared above, our hard segmentation not only provides precise bi-layer classification for efficient and accurate alpha mattes, but also refines the depth image to initialize alpha values and the selective neighborhood.

3.2. Bi-layer classification

We use Gaussian mixture model (GMM) to represent the foreground and background color and depth distributions. Take the color GMM as an example. The foreground and background color distributions can be represented as:

$$\begin{aligned}
p_i^{cf} &= \sum_{k=1}^{N_h} \omega_k^{cf} \cdot G_i(\boldsymbol{\mu}_k^{cf}, \boldsymbol{\Sigma}_k^{cf}),\\
p_i^{cb} &= \sum_{k=1}^{N_h} \omega_k^{cb} \cdot G_i(\boldsymbol{\mu}_k^{cb}, \boldsymbol{\Sigma}_k^{cb}),
\end{aligned} \tag{10}$$

where p_i^{cf} and p_i^{cb} are the probabilities of pixel *i* belonging to foreground and background, respectively. N_h refers to the number of the GMM components. $\boldsymbol{\mu}_k^{cf}$ and $\boldsymbol{\mu}_k^{cb}$ are the means of the *k*th components in the foreground and background GMMs, and $\boldsymbol{\Sigma}_k^{cf}$ and $\boldsymbol{\Sigma}_k^{cb}$ are the *k*th covariance matrices. The component weights ω_k^{cf} and ω_k^{cb} are normalized to sum to 1.

The GMMs can be learned through the well-known Expectation Maximization (EM) algorithm [19] with foreground/background labels. As previous works [7,5,6] did, we achieve the foreground/background labels based on the assumption that background objects are behind the foreground ones. That is, depth values of the background are higher than ones of the foreground. Hence, p_s percent of pixels with highest and lowest depth values are marked as the foreground and background labels respectively. Considering that inaccurate depth values will result in mistaken foreground/background labels, we update those labels and GMMs based on $\mathbf{d}^{(m-1)}$ in the mth iteration.

The GMMs are learned through the well-known Expectation Maximization (EM) algorithm [19] with foreground/background labels provided by the initialized depth image $\mathbf{d}^{init}$.

Define the color-based estimation of the bi-layer segmentation as $\hat{o}_i^c$ and its confidence p_i^c as:

$$\hat{o}_i^c = \begin{cases} 1, & \text{if } p_i^{cf} > p_i^{cb}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$
$$p_i^c = \max\{p_i^{cf}/(p_i^{cf}+p_i^{cb}), p_i^{cb}/(p_i^{cf}+p_i^{cb})\}.$$

Similarly, we define the estimation of $\hat{o}_i^d$ and corresponding confidence p_i^d based on the GMMs learned from the depth image $\mathbf{d}^{(m-1)}$. Therefore, the data and smoothness terms in Eq. (8) can be defined as::

$$\begin{aligned} \mathcal{D}_o(o_i^{(m)}) &= p_i^c \cdot \|o_i^{(m)} - \hat{o}_i^c\|^2 + p_i^d \cdot \|o_i^{(m)} - \hat{o}_i^d\|^2, \\ \mathcal{V}_o(o_i^{(m)}, o_j^{(m)}) &= \|o_i^{(m)} - o_j^{(m)}\|^2. \end{aligned} \tag{12}$$

And the minimization of the energy function is solved using graph cut [11].

3.3. Depth refinement

The 1st step in depth refinement, as described in the procedure in Section 3.1, is to fill "depth holes" to create the initialized depth image $\mathbf{d}^{init}$. To this end, we use the affinity of the depth image to help the initialization. First, we collect depth values of neighboring pixels along each depth hole and form a histogram. Second, some dominate peaks are chosen which have pixels more than 10% of all. Third, the dominate peak with the maximal depth value (probable to be the background) is selected, while the corresponding value is utilized to fill the current depth hole. The schematic of the initialization is drawn in Fig. 5. Though this is very crude, the depth values will be refined during the iterations as presented below.

In the following, we define the data and smoothness terms in Eq. (9) as follows:

$$\begin{aligned} \mathcal{D}_d(d_i^{(m)}) &= \|d_i^{(m)} - d_i^{cam}\|^2, \\ \mathcal{V}_d(d_i^{(m)}, d_j^{(m)}) &= \|d_i^{(m)} - d_j^{(m)}\|^2 \cdot \omega_{ij}^c \cdot \omega_{ij}^o. \end{aligned} \tag{13}$$

Here ω_{ij}^c and ω_{ij}^o evaluate the affinity between pixel i and j based on the color and bi-layer image, i.e.::

$$\begin{aligned} \omega_{ij}^c &= \exp(-\|\mathbf{c}_i - \mathbf{c}_j\|^2/\sigma_c^2), \\ \omega_{ij}^o &= \exp(-\|o_i^{(m)} - o_j^{(m)}\|^2/\sigma_o^2), \end{aligned} \tag{14}$$

where σ_c and σ_o model the variations of the difference in color and bi-layer values. Intuitively, neighboring pixels with similar color values are encouraged to share similar depth values. The energy function is optimized via the conjugate gradient descent algorithm:

$$\mathbf{d}^{(m)} = (\lambda_d\mathbf{I} + \mathbf{D}^{-1}\mathbf{W})\lambda_d\mathbf{d}^{cam}, \tag{15}$$

where $\mathbf{W}$ refers to the affinity matrix defined by $W_{ij} = \omega_{ij}^c \cdot \omega_{ij}^o$. $\mathbf{D}$ is a diagonal matrix with its ith element equal to the sum of the ith row of $\mathbf{W}$.

During depth refinement, usually only pixels along the boundaries are changed. We terminate the iteration between depth refinement and bi-layer classification when the bi-layer mask image is no longer changing. Typically, this process takes 5 to 10 iterations in our experiments.

As mentioned in the overflow of our algorithm, the trimap can be generated from the final bi-layer segmentation. We erode the foreground part as the definite foreground $\mathcal{R}_f$ and erode the background part as the definite background $\mathcal{R}_b$. The corresponding morphological sizes are S_f and S_b respectively. Residual pixels refer to the unknown region $\mathcal{R}_u$. Considering the accuracy of our bi-layer segmentation and computational cost, we fix the morphological sizes as the constants.

4. Soft segmentation

Based on the given trimap, we next present a soft-segmentation algorithm based on iterative transductive learning. The novelty of the algorithm lies in a new way to construct the Laplacian matrix, and an adaptive scheme to set the Lagrange multipliers for the objective function.

4.1. Problem formulation

In a typical trimap based image matting problem, foreground and background regions ($\mathcal{R}_f$ and $\mathcal{R}_b$) are given as training data, and the testing data (unknown $\mathcal{R}_u$) are also given. This inspires the use of transductive learning [20] for image matting. We re-define the problem as solving:

$$\arg\min_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^T\mathbf{L}\boldsymbol{\alpha} + (\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}})^T\boldsymbol{\lambda}(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}), \tag{16}$$

where $\hat{\boldsymbol{\alpha}}$ refers to an initialized alpha matte, including labeled pixels through user interaction, and estimated alpha values for unknown pixels. Details of the initialization step for $\hat{\boldsymbol{\alpha}}$ has been exploited in many recent works [21–27]. $\mathbf{L}$ is still the Laplacian matrix, and the diagonal matrix $\boldsymbol{\lambda}$ represents the Lagrange coefficients of all pixels. The alpha matte of the image can be computed as:

$$\boldsymbol{\alpha} = (\mathbf{L} + \boldsymbol{\lambda})^{-1}\boldsymbol{\lambda}\hat{\boldsymbol{\alpha}}. \tag{17}$$

The above optimization and solution are closely related to the MRF model which has also been applied for soft segmentation [1], where the objective function and the constraint would correspond to the "smoothness" and "data" terms, respectively. As a result, the improvements made in this paper can also be applicable to MRF-based image matting algorithms.

In order to allow an adaptive scheme to set the Lagrange coefficients, we compute the alpha matte iteratively. Let $\boldsymbol{\alpha}^{(0)} = \hat{\boldsymbol{\alpha}}$, m be the iteration index, we have:

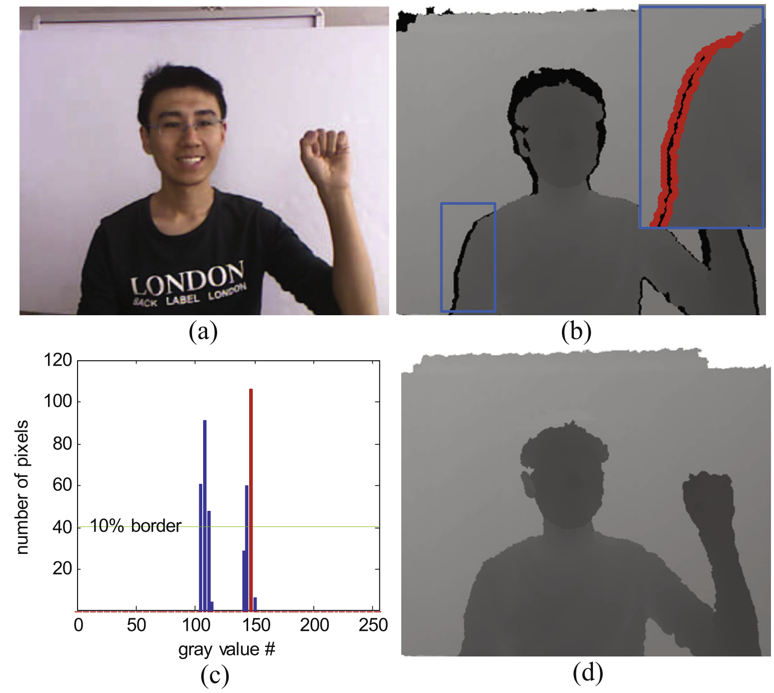
$$\boldsymbol{\alpha}^{(m)} = (\mathbf{L} + \boldsymbol{\lambda})^{-1}\boldsymbol{\lambda}\boldsymbol{\alpha}^{(m-1)}. \tag{18}$$

The iteration continues until the difference between $\boldsymbol{\alpha}^{(m)}$ and $\boldsymbol{\alpha}^{(m-1)}$ is less than a given threshold T_n. Adaptation of $\boldsymbol{\lambda}$ will be discussed in Section 4.3.

4.2. The Laplacian matrix

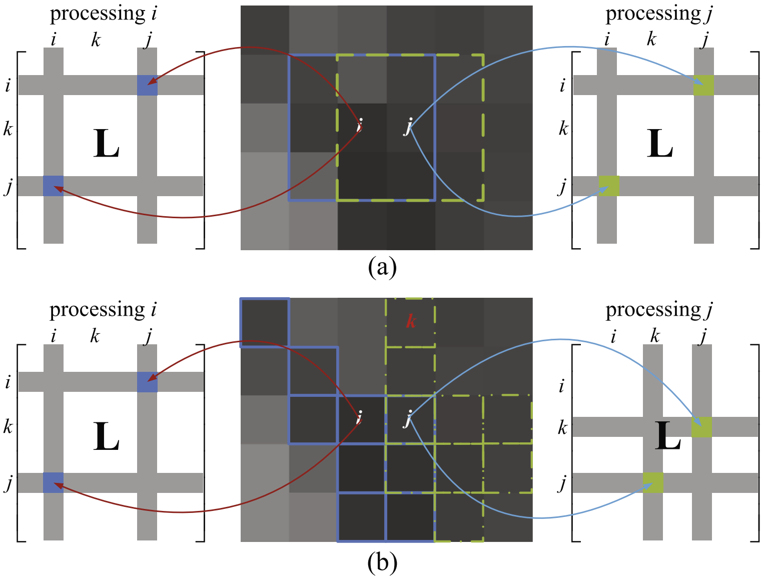
Traditionally, to define the Laplacian matrix for image matting, one often considers a 3 × 3 or 5 × 5 regular neighborhood for each pixel, as shown in Fig. 6(a). If a pixel i is within the neighborhood of pixel j, so does pixel j with respect to pixel i. Theoretically, one would like to use a relatively large neighborhood, as it will improve the reliability of the estimated neighborhood statistics. However, taking a large neighborhood indiscriminately could back-fire in images with complex textures, since neighboring pixels with low feature similarity may not be able to infer well. In addition, the computational cost of having a large neighborhood is often very high.

We propose to exclude neighboring pixels with low similarities to the center pixel while computing the Laplacian matrix. More



(a) (b) (c) (d)

Fig. 5. The color (a), depth images before (b) and after (d) our initialization. (c) refers to the histogram of the region in (b), where the bin marked by the red color corresponds to our picked dominate peak. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a)

(b)

Fig. 6. An example of the difference of the Laplacian matrix between previous works (a) [12–17,9] and ours (b). In the regular neighborhood (a), pixel i and j are neighbors to each other, where neighbors with low similarities are also preserved. In contrast, pixel i is not the one of pixel j's top N_r neighbors, so that it will be excluded by our irregular neighborhood (b). Additionally, the elements $L_{i,j}$ and $L_{j,i}$ are calculated repeatedly in (a), but not in (b).

specifically, we only preserve the top N_r neighbors with high feature similarity in an $M_r \times M_r$ neighborhood, where $N_r \ll M_r \times M_r$. The feature similarity between pixel i and j is defined as:

$$SIMI(i,j) = \frac{\mathbf{c}_i'^T \mathbf{c}_j'}{\|\mathbf{c}_i'\| \cdot \|\mathbf{c}_j'\|}, \tag{19}$$

where $\mathbf{c}_*' = [\mathbf{c}_*^T, d_*]^T$ (depth feature is introduced to avoid problems of confusing color values). For pixel i, we sort the similarity values of the pixels in its neighborhood $\mathcal{N}_i$ according to the descending order, and keep the most similar N_r neighbors as $\mathbf{C}_i = [\mathbf{c}_{i_1}'', \mathbf{c}_{i_2}'', \ldots, \mathbf{c}_{i_{N_r}}'']^T$. Here $\mathbf{c}_*'' = [\mathbf{c}_*^T, 1]^T$.

Deriving from [13], the alpha value α_i can be denoted by the combination of neighboring alpha or color values. The two representations are formulated as:

$$\boldsymbol{\alpha}_i = \mathbf{C}_i[\boldsymbol{\beta}_i, \beta_i']^T \tag{20}$$

and

$$\boldsymbol{\alpha} = \mathbf{F}^T \boldsymbol{\alpha}, \tag{21}$$

where $\boldsymbol{\beta}_i = [\beta_{i_1}, \ldots, \beta_{i_{N_r}}]^T$ and β_i' are the model coefficients of neighboring color values and $\mathbf{F}$ refers to the coefficient matrix of neighboring alpha values.

With the ridge regression technique, we can achieve $\boldsymbol{\beta}_i$ and β'_i by optimizing a quadratic problem:

$$\arg\min_{\boldsymbol{\beta}_i,\beta'_i} \|\boldsymbol{\alpha}_i - \mathbf{C}_i[\boldsymbol{\beta}_i, \beta'_i]^T\|^2 + \lambda_r[\boldsymbol{\beta}_i, \beta'_i][\boldsymbol{\beta}_i, \beta'_i]^T, \tag{22}$$

where λ_r refers to the balance coefficient. Thus, the optimal solution can be easily calculated by,

$$[\hat{\boldsymbol{\beta}}_i, \hat{\beta}'_i] = \left(\mathbf{C}_i\mathbf{C}_i^T + \lambda_r\mathbf{I}_{(m)}\right)^{-1}\mathbf{C}_i\boldsymbol{\alpha}_i, \tag{23}$$

where $I_{(m)}$ denotes the $m \times m$ identity matrix. Substituting Eq. (23) into Eq. (20), we can get $\boldsymbol{\alpha}_i = \left(\mathbf{C}_i\mathbf{C}_i^T + \lambda_r\mathbf{I}_{(m)}\right)^{-1}\mathbf{C}_i\mathbf{c}''_i\boldsymbol{\alpha}_i$. It is just the local representation of Eq. (21). Similar to Eq. (22), we can estimate the alpha matte by minimizing the following problem,

$$\min \ \|\boldsymbol{\alpha} - \mathbf{F}\boldsymbol{\alpha}\|^2 = \boldsymbol{\alpha}^T(\mathbf{I}_{(P)} - \mathbf{F})^T(\mathbf{I}_{(P)} - \mathbf{F})\boldsymbol{\alpha}. \tag{24}$$

Here the Laplacian matrix is represented by $(\mathbf{I}_{(P)} - \mathbf{F})^T(\mathbf{I}_{(P)} - \mathbf{F})$, where $\mathbf{F} = \{\xi_1, \xi_2, \ldots, \xi_P\}$. Elements of ξ_i are 0 for pixels outside $\mathcal{N}_i$ and equal to $\left(\mathbf{C}_i\mathbf{C}_i^T + \lambda_r\mathbf{I}_{(m)}\right)^{-1}\mathbf{C}_i\mathbf{c}''_i$ otherwise. Additionally, it can be seen that the Laplacian matrix is still symmetric from Eq. (24).

Fig. 6(b) explains the neighbor selection process of the our method. Note in our scenario, the pixel j is also selected among the top N_r neighboring pixels for pixel i. However, pixel i will not be selected for pixel j due to our irregular neighborhood. During this process, the elements in the Laplacian matrix: $L_{i,j}$ and $L_{j,i}$ are not calculated repeatedly. Hence, we not only preserve neighboring pixels with high similarities to the center one, but also avoid redundant computation.

4.3. The Lagrange coefficients

As mentioned earlier, setting a fixed Lagrange coefficient may not produce satisfactory results for matting. We propose to use an adaptive scheme for this purpose. The labeled and unlabeled pixels are first separated, since labeled pixels marked by user interaction or bi-layer segmentation are more reliable than estimated unknown pixels. In other words, we can split the matrix $\boldsymbol{\lambda}$ in Eq. (17) into two matrices:

$$\boldsymbol{\lambda} = \lambda_k\mathbf{I}_k + \lambda_u\mathbf{w}, \tag{25}$$

where the diagonal matrix $\mathbf{I}_k$ only take value 1 at labeled pixels. Deriving from the trimap, the labeled pixels refer to ones belonging to definite foreground $\mathcal{R}_f$ and definite background $\mathcal{R}_b$, while the unlabeled pixels correspond to the unknown region $\mathcal{R}_u$. Its coefficient λ_k is set to a large value, making sure that the final solution will always be consistent with the labeled data. The diagonal matrix $\mathbf{w}$ has non-zero entries only at pixels in unknown regions, and we have $w_i \in [0, 1]$ if $i \in \mathcal{R}_u$.

Second, since the initialized alpha matte $\hat{\boldsymbol{\alpha}}$ can be unreliable, we would like to incrementally increase the influence of $\mathbf{w}$ during the iterative optimization process. This can be realized by varying λ_u during the iterations. At the beginning, λ_u is set to a small value, such that unlabeled pixels can be quickly updated by their neighbors based on the affinity in the Laplacian matrix. The value of λ_u gradually increases, such that the computed alpha matte gradually stabilizes. In the current implementation, we let λ_u increase with the iteration index m, i.e.:

$$\lambda_u^{(m)} = \eta_u \cdot 10^m, \tag{26}$$

where η_u is a fixed base number.

Third, during the iterative matting process, the difference of the alpha values between two iterations can be an indicator of whether a pixel can be reliably derived from its neighbors. If the alpha value varies dramatically from one iteration to another, it is very likely that the pixel has discontinuity in its neighborhood, and we should reduce the Lagrange coefficient for that pixel. This paper applies the temporal variation of alpha values to update the confidence matte as follows:

$$w_i^{(m)} = \begin{cases} w_i^{(m-1)} \cdot \gamma, & \text{if } |\alpha_i^{(m)} - \alpha_i^{(m-1)}| \leqslant T_\alpha, \\ w_i^{(m-1)} \cdot \gamma^{-1}, & \text{otherwise}, \end{cases} \tag{27}$$

where T_α refers to a threshold for the temporal variation, and γ denotes the update ratio.

Overall, the iterative image matting scheme can be written as:

$$\boldsymbol{\alpha}^{(m)} = \left(\mathbf{L} + \lambda_k\mathbf{I}_k + \lambda_u^{(m)}\mathbf{w}^{(m-1)}\right)^{-1}\left(\lambda_k\mathbf{I}_k + \lambda_u^{(m)}\mathbf{w}^{(m-1)}\right)\boldsymbol{\alpha}^{(m-1)}, \tag{28}$$

where $\boldsymbol{\alpha}^{(0)} = \hat{\boldsymbol{\alpha}}$ and $\mathbf{w}^{(0)} = \mathbf{w}$. During our initialization of the alpha matte $\hat{\boldsymbol{\alpha}}$ based on RGB-D data, the depth value is added as the feature of one pixel. It especially works well for pixels with confusing color values.

5. Experiments and discussions

We have conducted extensive experiments to verify the performance of the proposed method. In this section, we will first present results for hard segmentation and matting separately, and then combine the two components to demonstrate the effectiveness of the overall algorithm.

5.1. Experimental setups

The following experiments are performed on the machine with Intel Core 2 Dual CPU at 2.2 GHz. Throughout this section, a fixed set of parameters are listed in Table 1. The parameters values for hard segmentation are referred to [7], while the ones for soft segmentation correspond to the work in [12]. Our particular parameters, such as γ and T_α, are set for iterations to balance the accuracy and efficiency. In general, the parameters in our algorithm are insensitive and robust for different images and occasions.

For hard segmentation, we compare our algorithm with Wang [7], which is the state-of-the-art for bi-layer segmentation with RGB-D data. For matting, we compare our Iterative Transductive Matting (ITM) algorithm with Learning-based Matting [13] (LM) and Transductive Matting [9] (TM). The former approach was the first to treat the matting problem as a learning-based one, while the latter adopted a transductive learning framework that is similar to ours.

Table 1
Fixed parameter values in our experiments.

Symbol	Explanation (Reference)	Value
λ_d	Constrain the data term for depth refinement (Eq. (9))	0.01
λ_o	Constrain the data term for bi-layer classification (Eq. (8))	0.01
σ_c	Balance color values (Eq. (14))	40
σ_o	Balance bi-layer values (Eq. (14))	200
p_s	Percentage of pixels for labeling (Section 3.2)	20
N^b	The number of bins for each GMM (Section 3.2)	8
N^h	The number of GMMs (Section 3.2)	3
S_f	The morphological size for the foreground (Section 3.3)	5
S_b	The morphological size for the background (Section 3.3)	5
N_r	The number of selected neighbors (Section 4.2)	9
M_r	The size of the neighborhood (Section 4.2)	11
λ_k	The coefficient of unlabeled pixels (Eq. (25))	10^8
η_u	The basic coefficient of unlabeled pixels (Eq. (26))	10^{-3}
γ	Increment of the confidence value (Eq. (27))	1.1
T_α	Threshold of alpha values' difference (Eq. (27))	0.05
T_n	Threshold of alpha mattes' difference (Section 4.1)	10^{-4}

The test data set includes:

1. *DatasetA*: This data set for hard segmentation was captured by us with a Kinect depth sensor [28–30]. There are 8 image sequences in the data set. Each is a sequence containing 200 frames of color, depth and ground truth bi-layer mask images (marked manually) at 640×480 pixels resolution. The hard segmentation is based on the assumption that the background is behind the foreground. That is, the depth values of the background is higher than ones of the foreground. p_s percent of pixels with highest and lowest depth values are marked as the foreground and background labels respectively.
2. *DatasetB*: The standard data set for soft segmentation was provided by Rhemann et al. [31]. It comprises 27 groups of color images and trimaps (definite foreground, definite background and unknown region are marked), with ground truth alpha mattes.
3. *DatasetC*: This data set for soft segmentation was also provided by Rhemann et al. [31]. It contains 24 groups of color images and trimaps (definite foreground, definite background and unknown region are marked); however, the ground truth alpha mattes were not provided.

5.2. Hard segmentation

5.2.1. Contents of the videos

The experiments are conducted on *DatasetA*. Sequence 01 and 04 contain relatively complex background, while sequence 02 and 03 have relatively simple background. In sequence 05, we evaluate the case with a moving camera. It could be noted that both the background and illumination condition vary over time during the camera's motion. Lastly, sequence 06–08 are particularly challenging, as background objects move close to the foreground object. In addition, sequence 07 and 08 contain foreground and background with similar color distributions.

5.2.2. Depth refinement and bi-layer classification

Fig. 7 shows some results of iterative hard segmentation. In the first row, depth images during the iterations are listed. The error images between our bi-layer classification results and the ground truth are shown in the second row. It can be seen that the depth values of regions like the hair are greatly improved (marked by the red rectangle), and the bi-layer segmentation of the chair region is also improved (marked by the green rectangle). Therefore, compared with other methods [7,5,6], we utilize smaller size of the morphological filter, where the chair is still judged as definite background correctly. Since corresponding unknown pixels are fewer, the accuracy of the following soft segmentation will be improved.

5.2.3. Subjective comparison

A number of frames selected from the hard segmentation results are presented in Fig. 8. We compare our algorithm (with and without iterations) with that of Wang et al. [7]. It can be seen clearly that the proposed algorithm outperforms Wang's method:

1. Depth holes are filled better by our depth refinement strategy, such as the face and the hair (e.g., in frames of sequences 01, 02, 03 and 07).
2. The boundaries between objects are estimated more accurately by our algorithm thanks to the MAP joint estimation of depth and segmentation, as demonstrated by the enlarged regions in the figures (e.g., in frames of the sequences 03, 04 and 05).
3. Our results are more robust to background with confusing color and depth distributions (e.g., in frames of the sequences 06 and 08).

Moreover, when multiple iterations are conducted for the MAP estimation, we can achieve more precise bi-layer mask images, since during the iterations the depth images are also refined.

5.2.4. Quantitative comparison

We quantitatively measure the number of mistaken pixels of the bi-layer mask image for Wang's [7] and our methods. The results are shown in Fig. 9. It can be seen that the algorithm of Wang et al. [7] induces the most mistaken pixels, whereas our algorithm with iterations performs the best. For certain sequences such as sequence 05, the reduction of mistaken pixels of our method against Wang's method is over 80%.

5.3. Soft segmentation

5.3.1. Neighbor selection for the Laplacian matrix

We use *DatasetB* to demonstrate the benefit of selecting the top similarity neighbors for the Laplacian matrix. In our method, we set the number of the top neighbors $N_r = 9$. If the size of the neighborhood $M_r = 3$, neighbors without selection are preserved. On that occasion, our algorithm degrades to the traditional methods [12–17,9]. On one hand, as shown in Fig. 10(a), when the size of the neighborhood increases, the approach which uses all neighbors ($N_r = M_r \times M_r$) performs worse than ours in matting accuracy. $M_r = 3$ corresponds to previous works [12–17,9]. On the other hand, the time complexity versus the neighborhood size is compared in Fig. 10(b). It can be seen that the proposed method has a relatively constant complexity, compared with the quadratic increase of computation for the traditional approaches ($N_r = M_r \times M_r$).

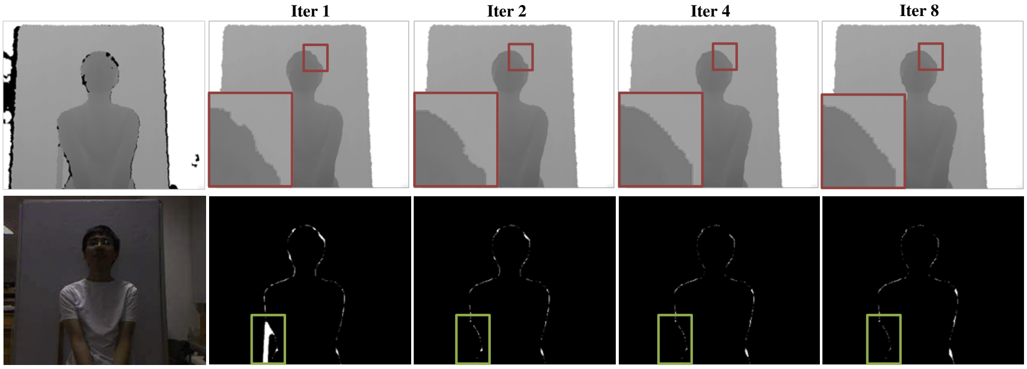


Fig. 7. Intermediate results of our hard-segmentation. Depth images and error images between the bi-layer segmentation results and ground truths are listed in the top and bottom rows, respectively. Drawn from the red rectangle, the depth image is refined. In the meantime, we reduce the errors of bi-layer segmentation as marked by the green rectangle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

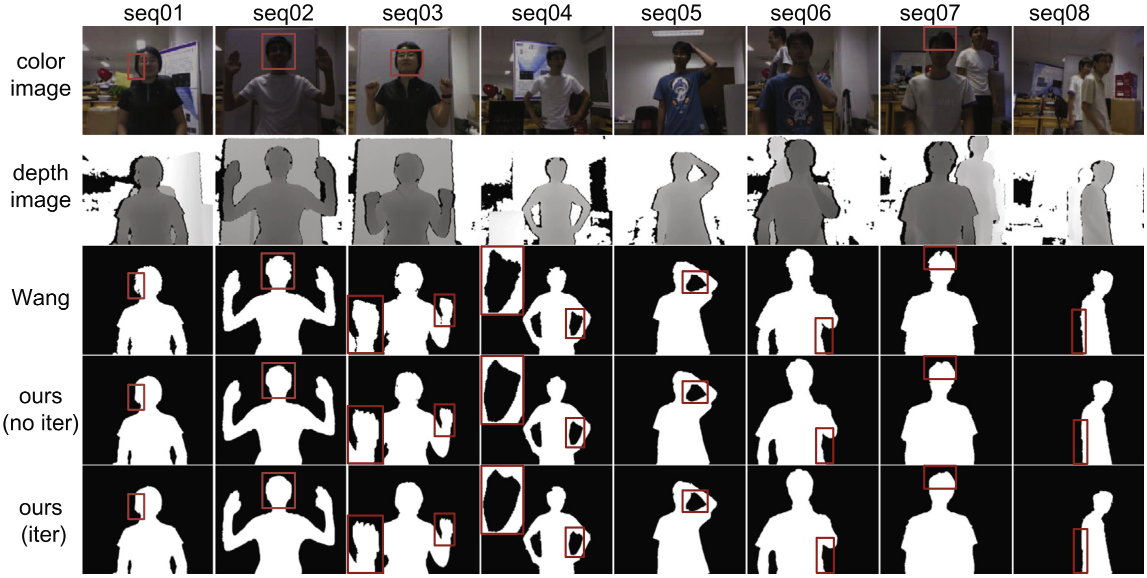


Fig. 8. Hard-segmentation results of Wang et al. [7], our algorithm (without iterations) and our algorithm (with iterations). Frames are selected from 8 image sequences (on *DatasetA*), respectively. The regions marked by the red rectangles are utilized to show our superiority of the accuracy, while comparing with Wang et al. [7]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

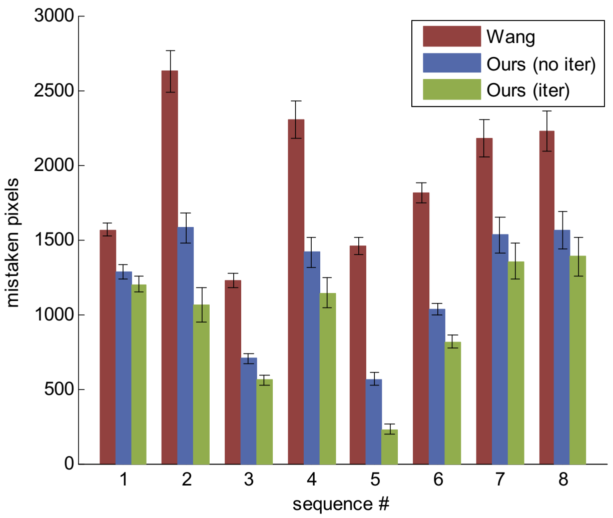


Fig. 9. Mistaken pixels of hard-segmentation results from Wang et al. [7], ours without and with iterations. Mean values as well as standard deviations of 8 image sequences (on *DatasetA*) are plotted in the figure.

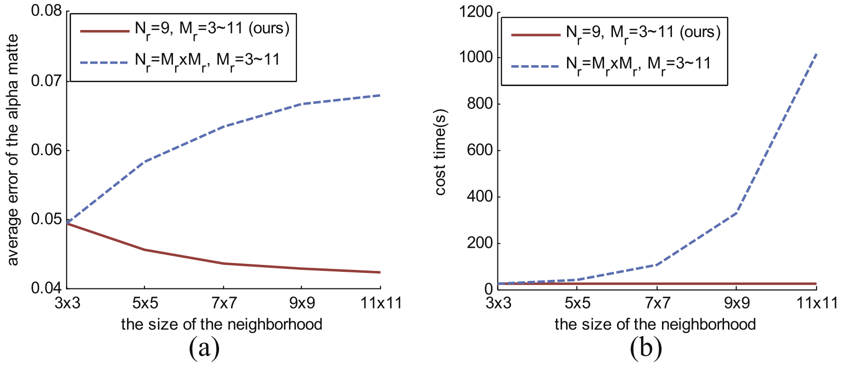
5.3.2. Convergence of the iterative optimization method

An example of our iterative optimization for soft-segmentation is illustrated in Fig. 11. When the number of iterations increases, the estimation error of alpha values and the difference of consecutive alpha mattes decreases consistently. As shown in Fig. 11(b), at the 0th iteration, the initialized alpha matte is not smooth since it is estimated pixel by pixel. During iterative optimization, high confidence pixels maintain their refined alpha values, whereas low confidence pixels are corrected by their neighbors gradually, which result in more precise alpha matte overall. The average number of iterations ranges from 10 to 20 on *DatasetB* and *DatasetC*.

5.3.3. Quantitative comparison

The estimation errors of LM [13], TM [9] and ITM on *DatasetB* are plotted in Fig. 12. Those methods perform differently on different images. ITM averagely outperforms LM [13] and TM [9], especially for images with hollow regions (such as “GT02”, “GT13”, etc.). We also list the corresponding mean and standard deviation values in the brackets for statistical comparison.

On *DatasetC*, although we do not have access to the ground truth alpha mattes, ranks of all state-of-the-art soft-segmentation algorithms in SAD (Sum of Absolute Difference) and MSE (Mean



(a) (b)

Fig. 10. The average errors of alpha mattes (a) and corresponding cost time (b) under different size of the neighborhood (on *DatasetB*). The red solid curves refer to our algorithm, while methods without selection are marked by the blue broken curves. Particularly, the point ($M_r = 3, N_r = 9$) on the blue broken curves corresponds to previous works [12–17,9]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

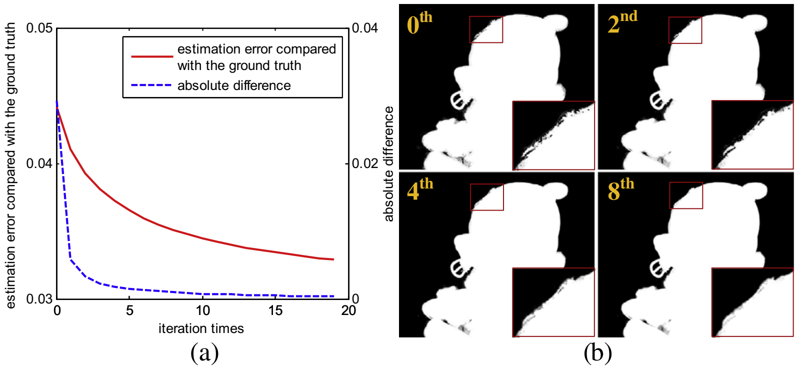


Fig. 11. Average difference between consecutive alpha mattes during the iterations (blue broken curve) and the estimation error (red solid curve) against the number of iterations (a). Intermediate matting results during the iterations are shown in (b). As shown in enlarged regions, smoother and more accurate alpha values are calculated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Squared Error) evaluation are available on website [35]. As shown in Table 2, ITM ranked 2nd and 3rd in SAD and MSE (in brackets) respectively among the top 28 matting algorithms.

5.3.4. Visual comparison

Fig. 13 shows soft-segmentation results of LM [13], TM [9] and ITM.

For the "GT04", "GT05", "GT21" and "plant" images, we not only achieve smoother results, but also avoid sticking into a local optimal solution. Taking the enlarged region in "GT05" as an example. For the pixel A, LM [13] confronts problems of over-smoothness, while the corresponding alpha value is 0.12. Whereas, the pixel B, which refers to the definite background, is wrongly judged as the combination of the foreground and background by TM. In the meanwhile, our ITM performs accurate matting results with the ground truth as a reference.

For the "GT16" and "troll" images, TM [9] directly includes unlabeled pixels in their formulation and LM [13] overemphasizes on the smoothness of the neighborhood, which both lead to imprecise alpha values (as marked by blue rectangles). Additionally, the size of neighborhood in LM [13] is 7 × 7 and introduce more neighbors with low feature similarities. As marked by the red arrow, LM [13] performs worst on pixels around the boundaries of the hair and bridge. In contrast, in our algorithm, low confidence pixels are updated by their neighbors while high confidence pixels maintain their estimations, leading to better results.

For the "net" and "donkey" images, our algorithm outperforms both LM [13] and TM [9], as shown in the enlarged regions. This is due to our enlarged neighborhood and selective mechanism of pixels during computing the Laplacian matrix, which reduces the influence of the cluttered background. We would also like to point out that since LM [13] neglects the unlabeled pixels, the results appear to be overly smooth, as marked by the arrows.

Table 2
Average rankings and total ranks (in brackets) in SAD and MSE evaluation (on *DatasetC*). Our algorithm corresponds to the bold line.

Algorithm	Sad	MSE
Learn-based matting [13]	13.7 (13th)	12.9 (12th)
Shared matting [23]	8.9 (5th)	10 (7th)
Global sampling matting [24]	10.6 (6th)	9.2 (5th)
Weighted color and texture matting [32]	8.2 (4th)	8.8 (4th)
SVR matting [33]	7.3 (3nd)	7 (2nd)
LNSP matting [34]	5.3 (1st)	4.6 (1st)
Iterative transductive matting	**6.2** (**2nd**)	**8.2** (**3rd**)

5.4. Overall matting

5.4.1. Matting results

Fig. 14 provides selected frames before and after soft-segmentation on *DatasetA*. The bi-layer mask images, alpha mattes and cores Fig. 14 provides selected frames before and after soft-segmentation on *DatasetA*. The bi-layer mask images, alpha mattes and corresponding composites with new backgrounds are given from top to bottom.

For frames in sequences 01, 02 and 05, the results after image matting are much better than the hard segmentation results. This should be attributed to the better scheme in constructing the Laplacian matrix during image matting, where background pixels

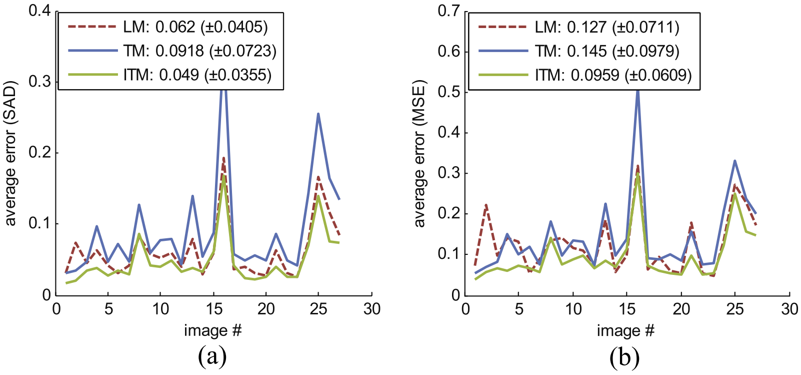


Fig. 12. The estimation errors of LM [13], TM [9] and our ITM (y-axis) versus the image index (x-axis). The SAD and MSE are evaluated in (a) and (b), respectively. (on *DatasetB*). The corresponding mean and standard deviation values are listed in the brackets, respectively.

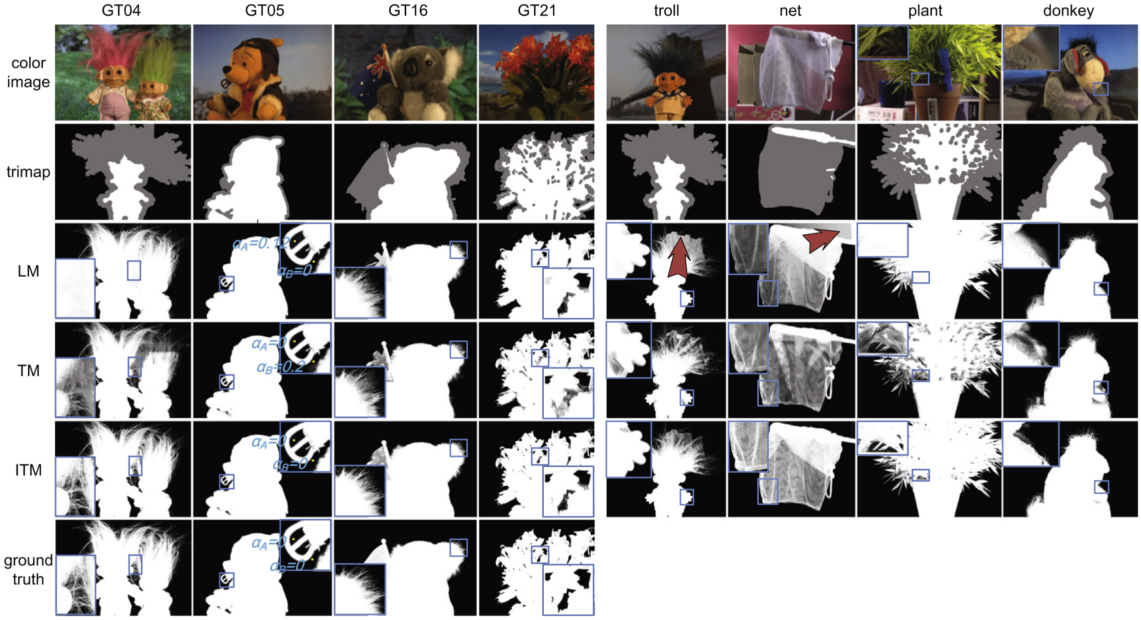


Fig. 13. Soft-segmentation results of LM [13], TM [9] and ours. The left and right 4 columns refer to evaluations on *DatasetB* (with the ground truths) and *DatasetC* (without the ground truths) respectively. Several significant differences are marked by the blue rectangles and red arrows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with similar depth values would still be excluded due to color feature dissimilarity.

For frames in sequences 03 and 07, we correct the alpha values of boundary pixels in bi-layer mask images. Note that the boundaries of the foreground in bi-layer mask images are relatively sharp, while our alpha mattes are more subtle and closer to the ground truth, as shown in the enlarged regions.

For frames in sequences 04, 06 and 08, the alpha values of the boundary pixels after matting are smoother, producing fewer jags than the results in hard-segmentation. Thanks to the adaptive setting of Lagrange coefficients, our soft-segmentation approach preserves high-confidence pixels and corrects low-confidence pixels based on their neighbors.

5.4.2. Details in fuzzy regions

To demonstrate the performance of our matting algorithm, we show two additional examples in Fig. 15. The examples contain foreground and background regions with distinct color distributions, thus the readers can clearly examine the details in fuzzy regions such as the hair. The bi-layer mask images contain sharp boundaries between objects, which is corrected in the alpha matte

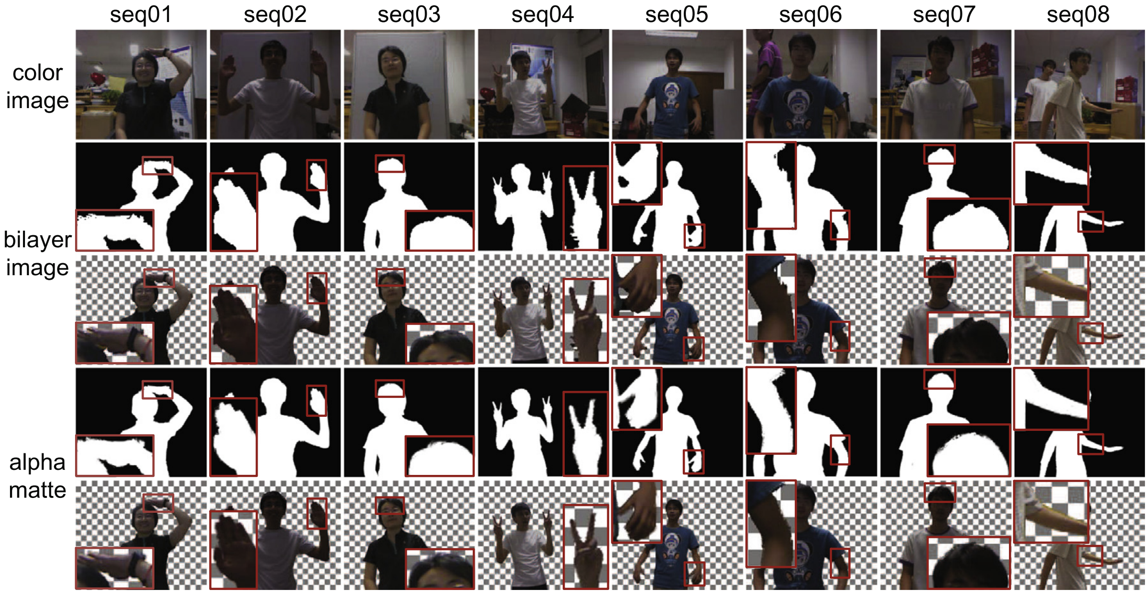


Fig. 14. Bi-layer mask images and alpha mattes as well as the composites with new backgrounds. Frames are selected from 8 image sequences (on *DatasetA*) respectively. As shown in regions marked by the red rectangles, smoother and more practical fusing results can be achieved after soft-segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

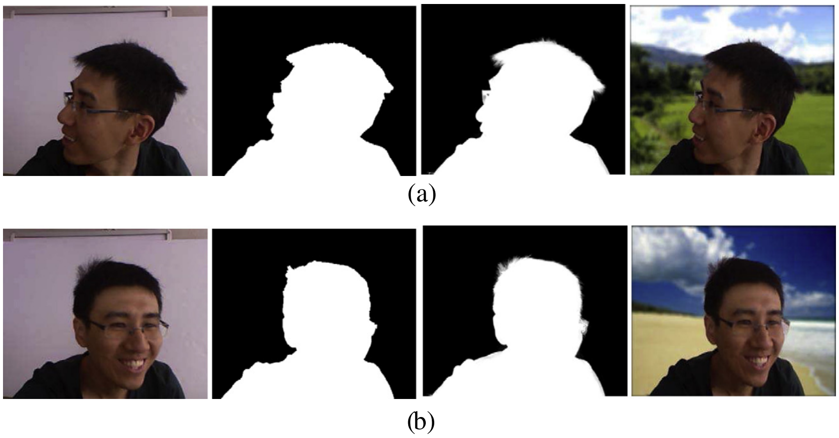


Fig. 15. Selected frames to evaluate the performance of matting in fuzzy regions (on *DatasetA*). For each frame, the color image, bi-layer mask image, trimap, initialized alpha matte, final alpha matte and composite are shown respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

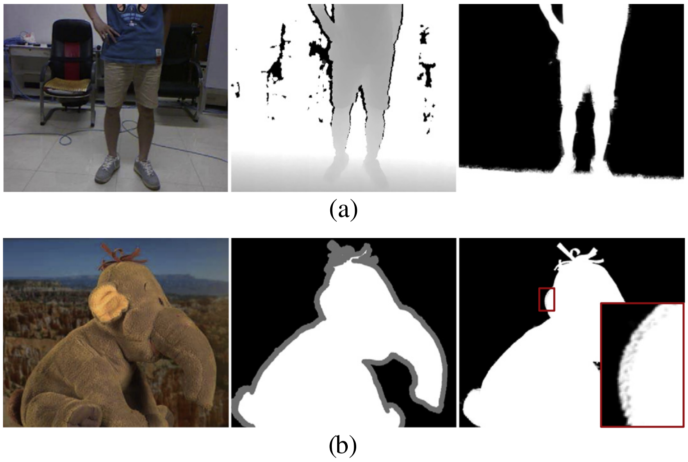


Fig. 16. Failure examples for hard segmentation and matting with the proposed method. (a) From left to right, the color, depth and bi-layer images are listed. The poor result is caused by similar depth values at the ground. (b) From left to right, the color image, trimap and alpha matte are listed. The poor result is caused by confusing foreground and background color distributions as marked by the red rectangle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

after soft segmentation. The composition with new backgrounds are visually pleasing.

5.5. Limitations

Our hard segmentation strategy is based on the assumption that the foreground and background objects are separated in depth. If the separation is not obvious, we may obtain incorrect results. For instance, in Fig. 16(a), since the floor is too close to the feet of the foreground person, the bi-layer classification is imprecise. Solving this issue requires background modeling or some semantic understanding of the scene content, which is our future work.

Regarding soft segmentation, our iterative transductive learning may perform poorly if the foreground and background color distributions are similar. An example of such a failure case is shown in Fig. 16(b). Depth input would be greatly beneficial in such scenarios to improve the matting quality.

6. Conclusion

Based on iterative transductive learning, this paper proposed a highly accurate and automatic matting algorithm with RGB-D data. The algorithm consists of two stages: hard segmentation and image matting. The hard segmentation problem was solved with a novel MAP formulation by maximizing the posterior probability, iterating between depth refinement and bi-layer classification. The soft segmentation was addressed through iterative transductive learning, where we improves the formulation of the Laplacian matrix and varies the Lagrange coefficients for each iteration adaptively. Subjective and quantitative comparisons demonstrated that our algorithm outperforms the state-of-the-art approaches for image segmentation and matting.

Acknowledgments

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

References

[1] J. Wang, M.F. Cohen, Image and video matting: a survey, Found. Trends® Comput. Graph. Vision 3 (2007) 97–175.

[2] A. Smith, J. Blinn, Blue screen matting, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1996, pp. 259–268.

[3] J. Sun, Y. Li, S. Kang, H. Shum, Flash matting, ACM Trans. Graph. (TOG) 25 (2006) 772–778.

[4] S. Matyunin, D. Vatolin, Y. Berdnikov, M. Smirnov, Temporal filtering for depth maps generated by kinect depth camera, in: 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), IEEE, 2011, pp. 1–4.

[5] J. Zhu, M. Liao, R. Yang, Z. Pan, Joint depth and alpha matte optimization via fusion of stereo and time-of-flight sensor, in: IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2009, IEEE, 2009, pp. 453–460.

[6] J. Cho, R. Ziegler, M. Gross, K. Lee, Improving alpha matte with depth information, IEICE Electron. Express 6 (2009) 1602–1607.

[7] L. Wang, M. Gong, C. Zhang, R. Yang, C. Zhang, Y. Yang, Automatic real-time video matting using time-of-flight camera and multichannel poisson equations, Int. J. Comput. Vision 97 (2012) 104–121.

[8] J. Yu, J. Zhao, Segmentation of depth image using graph cut, in: Ninth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, 2012, pp. 1934–1938.

[9] J. Wang, Image matting with transductive inference, Comput. Vision/Comput. Graph. Collab. Tech. (2011) 239–250.

[10] C. Zhang, Z. Yin, D. Florêncio, Improving depth perception with motion parallax and its application in teleconferencing, in: IEEE International Workshop on Multimedia Signal Processing, MMSP'09, IEEE, 2009, pp. 1–6.

[11] C. Rother, V. Kolmogorov, A. Blake, Grabcut: Interactive foreground extraction using iterated graph cuts, in: ACM Transactions on Graphics (TOG), vol. 23, ACM, 2004 , pp. 309–314.

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

[13] Y. Zheng, C. Kambhamettu, Learning based digital matting, in: IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 889–896.

[14] S. Xiang, F. Nie, C. Zhang, Semi-supervised classification via local spline regression, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 2039–2053.

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

[16] Q. Chen, D. Li, C. Tang, Knn matting, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 869–876.

[17] K. He, J. Sun, X. Tang, Fast matting using large kernel matting Laplacian matrices, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2165–2172.

[18] S. Li, Markov random field models in computer vision, in: Proceedings of the Third European Conference on Computer Vision, vol. II, Springer-Verlag Inc., New York 1995, pp. 361–370.

[19] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B (Methodol.) (1977) 1–38.

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

[21] J. Wang, M. Cohen, Optimized color sampling for robust matting, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07, IEEE, 2007, pp. 1–8.

[22] C. Rhemann, C. Rother, M. Gelautz, Improving color modeling for alpha matting, in: British Machine Vision Conference, vol. 2, 2008, pp. 1155–1164.

[23] E. Gastal, M. Oliveira, Shared sampling for real-time alpha matting, in: Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 575–584.

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

[25] B. He, G. Wang, Z. Ruan, X. Yin, X. Pei, X. Lin, Local matting based on sample-pair propagation and iterative refinement, in: 19th IEEE International Conference on Image Processing (ICIP), IEEE, 2012, pp. 285–288.

[26] B. He, G. Wang, C. Shi, X. Yin, B. Liu, X. Lin, High-accuracy and quick matting based on sample-pair refinement and local optimization, IEICE Trans. Inf. Syst. 96 (2012).

[27] B. He, G. Wang, C. Shi, X. Yin, B. Liu, X. Lin, Iterative transductive learning for alpha matting, in: 20th IEEE International Conference on Image Processing (ICIP), IEEE, 2013.

[28] Microsoft kinect for x-box 360, <http://www.xbox.com/en-US/kinect>, 2010.

[29] G. Wang, X. Yin, X. Pei, C. Shi, Depth estimation for speckle projection system using progressive reliable points growing matching, Appl. Opt. 52 (2013) 516–524.

[30] X. Yin, G. Wang, C. Shi, Q. Liao, Efficient active depth sensing by laser speckle projection system, Opt. Eng. 53 (2014) 013105.

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

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

[33] Z. Zhang, Q. Zhu, Y. Xie, Learning based alpha matting using support vector regression, in: 19th IEEE International Conference on Image Processing (ICIP), IEEE, 2012, pp. 2109–2112.

[34] X. Chen, D. Zou, S.Z. Zhou, Q. Zhao, P. Tan, Image matting with local and nonlocal smooth priors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013.

[35] Alpha matting benchmark, http://www.alphamatting.com, 2009.