Vision, Modeling, and Visualization (2016)
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

Matting with Sequential Pair Selection Using Graph Transduction

A. Al-Kabbany and E. Dubois

School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Abstract

We are concerned with the natural image matting problem, where the goal is to estimate the partial opacity of a foreground object so that it can be softly segmented from a background. In sampling-based matting techniques, user interactions are first acquired to provide prior information about foreground and background regions. Samples are then chosen from those interactions to calculate the alpha (opacity) value of every pixel in an image. In this research, we propose a new sampling approach that brings relevant samples to every pixel with an unknown alpha value; this yields accurate alpha maps. We also present two new formulations for objective functions used to assess the suitability of the chosen samples. The evaluation of the proposed method, on the alpha matting online benchmark, shows that its performance is close to the state-of-the-art techniques.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification

1. Introduction

Natural image matting is a cornerstone for image compositing, which is one of the fundamental image editing operations. For visually plausible composites, we want to be able to estimate the partial coverage of every pixel in an image so that the foreground objects, even those with thin fuzzy structures and transparent surfaces, can be overlaid seamlessly on a variety of backgrounds. To calculate that opacity map (or alpha map), a linear convex model is used to represent every pixel, so that the color of each pixel is expressed as:

$$I_i = \alpha_i \times F_i + (1-\alpha_i) \times B_i \ , \tag{1}$$

where i is the pixel index, I_i is the pixel value (or feature vector), F_i and B_i are the feature vectors of the foreground (Fg) and the background (Bg) pixels contributing to the color of I_i respectively, and $0 \leq \alpha_i \leq 1$ is the opacity value of I_i, with 1 for Fg pixels and 0 for Bg pixels. Equation 1 will be referred to as the compositing equation throughout the rest of this document. Since we want to estimate α_i, and we don't know F_i and B_i, the compositing equation represents an under-determined problem. Thus, the pool of solutions is downsized by providing additional information in the form of sparse scribbles or a dense three-level segmented image named 'trimap' specifying definite Fg ($\alpha = 1$), definite Bg ($\alpha = 0$) and unknown regions. The pixels in the Fg and the Bg regions of the trimap will be referred to as 'the known pixels' throughout the rest of this document, while the term 'unknown pixels' will be used to refer to the pixels with alpha values to be computed.

One approach for computing alpha maps is to propagate the alpha values of the known pixels to the unknown ones [LRAL08, LLW08, CLT12, HWS*13, SAP*13]. This is achieved by defining a similarity measure between the image pixels, based on which an affinity matrix can be constructed. Different members of the propagation-based matting family adopt various affinity measures which determine the accuracy of propagation. The limitations of the members of this family are attributed to the underlying smoothness assumptions [LRAL08], which may not hold, and/or the high correlation between the Fg and Bg samples, which leads to wrongly propagated alpha values.

The second family of techniques [KEE15, JSRC16, JRC14, SRPC13, CZZ*13, VR13, HRR*11, GO10, WC07b] adopt a hybrid approach that is comprised of the three following stages. First, the user interactions (trimap or scribbles) are sampled to bring a subset of Fg/Bg pairs to every unknown pixel; this step is called 'sample gathering'. Second, an objective function is optimized (often by brute-force) to single out the pair that best describes the color of the pixel under consideration; we will call that pair 'a good pair' or 'a suitable pair' throughout this document. A classical example of such an objective function, which has been followed by more robust alternatives, is known as the chromatic distortion, and is given by

$$\xi_{color} = \| I_i - (\hat{\alpha} F_u + (1-\hat{\alpha}) B_v) \| \ , \tag{2a}$$

$$\hat{\alpha} = \frac{(I_i - B_v) \cdot (F_u - B_v)}{\|F_u - B_v\|^2} \ , \tag{2b}$$

where (F_u, B_v) is a particular Fg/Bg pair, $\hat{\alpha}$ is the estimated α, $\cdot$ is the inner product and $\|\cdot\|$ is the Euclidean norm. The dot product and the norm are calculated over the RGB color coordinate system. The best pair is thus the pair that minimizes the color distance between the original pixel value and the value we get from a par-

DOI: 10.2312/vmv.20161349

ticular pair with a particular estimated alpha. This can be visualized as the perpendicular distance between the unknown pixel and the line joining the Fg/Bg pair being assessed. The second stage, that is the pair assessment stage, is often done on a per-pixel basis; thus, the computed alpha maps undergo a post-processing smoothing step in the final stage of the pipeline. The smoothing step involves solving a system of linear equations to minimize a quadratic cost with the two following terms. The first term is the data term which represents the sampling-based computed alpha maps. The second term is the smoothness term which is meant to propagate alpha values among pixels based on their affinity. The definition of that affinity is an aspect of variation among the proposed methods. References [WC07a] and [ZSLW15] present more comprehensive, detailed and up-to-date surveys for the natural image matting literature.

The most recent members of the second family (hybrid approaches) include the methods proposed in [JRC14, SRPC13, VR13, CZZ*13, KEE15]. Instead of adopting the linear convex composition model in eqn. 1, the authors of [JSRC16, JRC14] used sparse codes to jointly describe an unknown pixel with multiple samples, rather than a single Fg/Bg pair. Rather than considering only the spatially-close samples to an unknown pixel, the method in [SRPC13] determined the number of samples gathered for every unknown pixel based on its spatial distance from the unknown region's boundaries in the trimap. Its objective function included a term for favouring spatially-close samples and a Cohen's d-based term to favour Fg/Bg samples that are well-separated in the color space. Building on the comprehensive sampling method of [SRPC13], the authors of [VR13] introduced a texture descriptor to better discriminate between Fg and Bg samples with overlapping color distributions. The method in [CZZ*13] gathered spatially-local samples to compute initial alpha maps, and then post-smoothed them by minimizing a quadratic cost function in α. To construct the Laplacian matrix of the smoothness term, affinities between neighbouring pixels in the spatial domain (local neighbours) and the feature space (non-local neighbours) were considered. The method in [KEE15] formulated the sample gathering step as a sparse subset selection problem. During pair assessment, the compatibility of a certain Fg/Bg pair with an unknown pixel was determined based on the classical chromatic distortion, the spatial closeness and the statistical feature similarities of the corresponding super-pixels.

We propose a hybrid, sampling-based approach for matting. Our contributions are a new sample gathering method and two new formulations for the objective function used for pair assessment. The sample gathering method aims at bringing good Fg/Bg samples to every unknown pixel, leading to more accurate alpha maps, while the presented objective functions are meant to augment the discriminative power of the classical objective function given by eqn. 2a. Subjective and objective results on standard matting datasets [mat] show that the performance of our method is close to the state-of-the-art (SoA) techniques.

2. Motivation Behind Sequential Pair Selection

The matting equation models the color of an unknown pixel as a mixture of a Fg/Bg pair of samples. However, alpha maps are assumed to be sparse [LRAL08, WC07b]; hence, the pair that best describes an unknown pixel is comprised of one similar half-pair (in feature) to the unknown pixel, while the other is dissimilar. This is emphasized by methods which encourage the nomination of Fg/Bg samples that come from well-separated color distributions [SRPC13, WC07b, RRG08]. Good samples that well-describe the color of an unknown pixel do not necessarily exist spatially-nearby to that unknown pixel [HRR*11, SRPC13, AKD14], but at least the dissimilar half-pair should. Figure 1 illustrates the two cases encountered during trimap sampling. In the first case, both half-pairs lie nearby in space to the unknown pixel, while in the second case, the similar half-pair lies far away from the unknown pixel. The figure features a diagram in addition to two examples for each case. In the first example of the second case, a pixel from the nearby leaf would constitute a *good half-pair* for a pixel in the blue unknown region because it is quite distinctive from it. The same example shows that favouring nearby samples would bring a bad Bg half-pair, because the blue unknown region is spatially close to a yellow (known) background region as depicted in the trimap. With textured backgrounds, relying on the spatial distance, even to decide the size of the gathered pool of pairs [SRPC13], could disqualify good samples from reaching the pair assessment stage.

Whenever the gathered pool of pairs gets large in size, the color ambiguity problem arises [HRR*11]. It refers to the case where a *wrong pair* minimizes the objective function (chromatic distortion) during the pair assessment stage, leading to a wrong alpha value. Figure 2 illustrates that problem. The case depicted in the figure shows that the wrong pair of pixels Fg_2/Bg_2 would be nominated, for the unknown pixel U, instead of the pair Fg_1/Bg_1 because E_2 (representing the chromatic distortion) is less than E_1. Methods with objective functions that encourage sparsity in alpha maps [WC07b, RRG08] could efficiently deal with this problem, but the sample gathering in these methods is limited to spatially-nearby regions in the trimap. If U would have been initially paired with Fg_1 (or Bg_1) and a complement half-pair would then be sought, the ambiguity problem would be avoided.

3. Learning by Transduction

Assuming that some training data (with known class/label) is available, the classical inductive model for inference uses the labelled data points to construct a predictive model or a mapping function with which new (testing) points can be labelled. For data lying on complex manifolds, even powerful discriminative model construction approaches, Adaboost and SVM for example, may fail to crystallize a generic model that works equally well with the labelled and the out-of-sample data points. In specific cases or problems, the necessity of learning a general rule can be avoided [GVV13], and both the labelled and the unlabelled data can be used to classify the unlabelled points; this is transduction.

Figure 3 depicts an instance of the two half-moons configuration. As shown, all the observed data points, labelled and unlabelled, are available beforehand. In Fig. 3(a), the green and orange points are the labelled data, and the goal is to label the rest of the grey points with a binary label (green or orange). Inferring a model using the labelled points only may result in fitting a hyperplane for example, as shown in Fig. 3(a), which results in wrong labels. However,

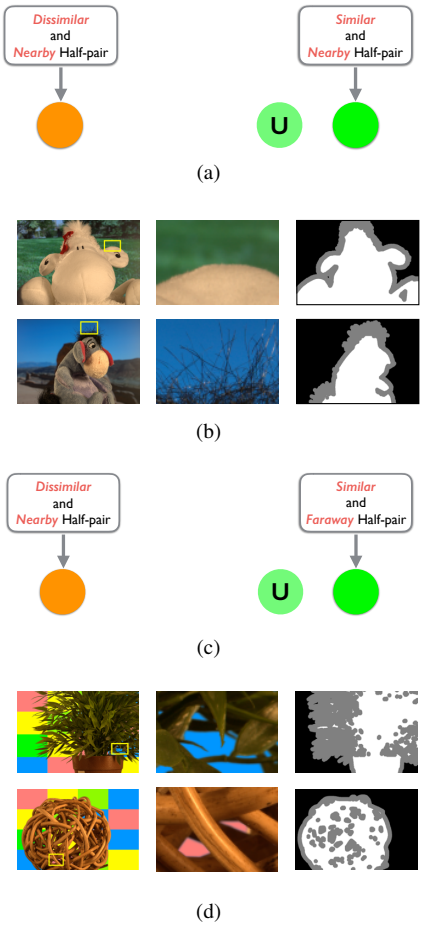


Figure 1: Cases encountered during trimap sampling. (a) Case 1: The unknown pixel U has a similar half-pair and a dissimilar half-pair, both nearby in space. (c) Case 2: The unknown pixel U has a dissimilar half-pair nearby in space, but the similar half-pair exists far away in space. (b) and (d) Two examples (patches) that depict each case are shown. The three columns show an image, an enlarged patch from it, and one possible trimap for it.

if there is a high confidence that the points are well-separated in the feature space, a function might be learnt from *all the observed data points* such that it passes through the low-density regions in the feature space; this is shown as the black curve in Fig. 3(b). Although it looks appealing, transduction cannot be used in the case of streaming data, and the high-margin feature space should exist to guarantee the availability of low-density regions.

We are particularly concerned with the graph Laplacian-based transductive inference that was discussed in [BN04] and developed in [DAK*08]. Figure 3 shows that the goal of transduction is to find a smooth mapping f that varies only in regions of low density in the input space, and simultaneously maps every training point to its associated (or a very close) label, i.e., $f(X_i) = Y_i$, where Y_i is the label of the training point X_i. The previous requirements can be formulated as an optimization problem, for which a discrete alternative was presented in [HAvL05]. That discretization approach adopts

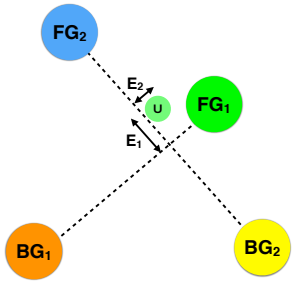


Figure 2: An illustration of the color ambiguity problem.

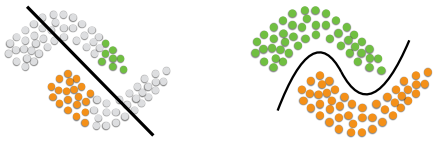


Figure 3: The iconic illustration of transduction on two half-moons configuration. Please see text for more details.

graph Laplacian methods that are based on a discrete approximation of the s-weighted Laplacian operator [DAK*08]. These methods construct a graph with nodes representing the data points (the X_is), and the weights of that graph are induced using a kernel (often an exponential kernel) that quantifies the affinities between the X_is in their feature space. The proposed discrete approximation for the original optimization problem is given by

$$\min_{F \in \mathbb{R}^n} (F-Y)^T C (F-Y) + F^T L\, F, \tag{3}$$

where n is the total number of labelled and unlabelled data points, C is the diagonal $n \times n$ matrix in which the i^{th} diagonal element is c_i for a labelled point, and 0 for a test point, Y is the n-dimensional vector in which the i^{th} element is Y_i for a labelled point, and 0 for a test point and L is the graph Laplacian. The n-dimensional vector F can then be obtained by solving the linear system given by

$$(L+C)\, F = C\, Y. \tag{4}$$

For the binary labelling problem depicted in Fig. 3, F should be thresholded. The labels of the testing points are the elements in F whose indices are the corresponding indices of the nodes of the testing points in the constructed graph. Transductive inference has been introduced to many problems in computer vision including segmentation [DAK*08] and matting [Wan11]. Transduction is adopted in this research as a part of the proposed trimap sampling strategy.

4. Proposed Method

The proposed method is comprised of the following stages: Segmenting the image into SPs and nominating delegate pixels for each of them, computing a suitable half-pair SP for every unknown SP, allowing neighbouring SPs to share their half-pairs, finding a good complement half-pair for every unknown SP, given its computed half-pair, and finally assessing the Fg/Bg pairs gathered for every unknown SP. The following sub-sections explain each of these

stages. Setting the values of the various parameters and algorithmic decisions is discussed in section 5.

4.1. Choosing Delegates for Super-pixels

The algorithm starts by computing the SLIC super-pixels [ASS*12, vlf] (region size=20 and regularizer=1) of the input image. A subset of each super-pixel's members are then chosen, according to the following procedure, to represent it. For every super-pixel (SP), we calculate the mean RGB color vector; members are then sorted according to their deviation from the mean. The whole range of deviation-from-mean is then divided into N subgroups ($N = 10$ in our experiments) of equal length and S_{SG} samples are picked evenly from every subgroup as a function of a budget that is given by

$$S_{SG} = \left\lceil B \times \frac{N_{SG}}{N_T} \times \frac{(MAD)_{SG}}{(MAD)_T} \right\rceil, \tag{5}$$

where B is the budget ($B = 40$ in our experiments), N_{SG} and N_T are the number of members in the subgroup and the whole SP respectively, $(MAD)_{SG}$ is the mean absolute deviation from the mean in the subgroup and $(MAD)_T$ is the mean absolute deviation from the mean in the whole SP. Once the delegates are determined for every SP, we calculate a weighting matrix which indicates how the rest of a SP's members can be obtained from its delegates. This matrix is calculated using the same procedure of [RS00]. It can be expressed as:

$$W := \underset{w_{ij}}{\operatorname{argmin}} \sum_{i=1}^{N_T} \left| \vec{X}_i - \sum_{j=1}^{K} w_{ij} \vec{X}_j \right|^2 \quad s.t. \quad \sum_{j=1}^{K} w_{ij} = 1, \tag{6}$$

where N_T is the total number of pixels in a super-pixel, K is the number of delegates and $\vec{X}_i$ (and $\vec{X}_j$) is a pixel's feature vector. This procedure simply applies the local linearity principle within every SP. During pair assessment, the alpha values of an unknown SP's members can be reconstructed using the weighting matrix and the alpha values of the delegates only. Hence, the purpose of using the local linearity principle is different from that of the method in [CZZ*13].

4.2. Good Half-pair Computation and Sharing

We compute the cartoon-texture decomposition [BLMV11] of the input image. With delegates represented by their cartoon-texture feature vector, we solve a binary graph transduction problem, akin to [DAK*08], to find the best half-pair (a Fg or a Bg super-pixel) for every unknown SP. Cartoon-texture decomposition is an additive decomposition model which aims at analyzing the signal into a piece-wise smooth (cartoon) component and an oscillatory (textural) component. The feature of every delegate is a 6×1 vector, comprised of the cartoon component and the range-filtered texture component. The cartoon-texture decomposition is used in lieu of the color to avoid the ambiguity that may arise if the Fg and Bg color distributions overlap.

We start by, and loop over, the unknown SPs that are not farther than 50 pixels from known regions in the trimap; unknown SPs are those that contain unknown pixels. The known SPs that are 50 pixels (or less) away from an unknown SP represent its proposals. Every unknown SP is offered one of its proposals at a time. To decide whether the unknown SP under consideration accepts a proposal or not, we build a graph using the delegates of the two SPs. The entries of the Laplacian matrix of this graph are calculated using the kernel function given by

$$k(X_i, X_j) = \frac{\widetilde{k}(X_i, X_j)}{[\widetilde{d}(X_i)\ \widetilde{d}(X_j)]^{\lambda}} \quad \text{where} \tag{7a}$$

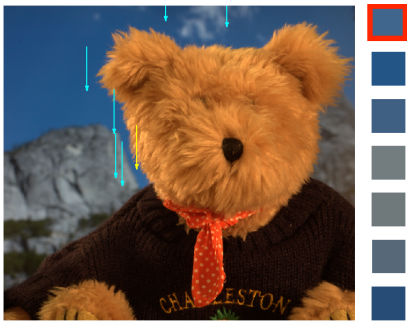
$$\widetilde{k}(X_i, X_j) = \mathrm{e}^{-\frac{\|X_i - X_j\|^2}{2\sigma^2}} \quad \text{and} \tag{7b}$$

$$\widetilde{d}(X_i) = \sum_{j=1}^{n} \widetilde{k}(X_i, X_j). \tag{7c}$$

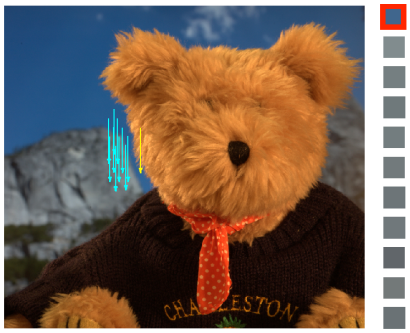
In the above equations, n is the dimension of the Laplacian (square) matrix and X_i (and similarly X_j) is defined as the cartoon-texture feature vector. The graph is then transduced by minimizing an objective function and solving a corresponding linear system given by eqn. 3 and eqn. 4. After obtaining and thresholding F, if at least 30% of the delegates in the unknown SP accept the proposal, the latter will be assigned as its best half pair, i.e. the number of ones among the entries in F that correspond to unknown delegates should be at least 30% of its length. The loop over the proposals is interrupted once the unknown SP under consideration accepts a proposal. We continue the best half-pair computation in propagation fashion. The unknown SPs that have been already paired (assigned a half-pair) will represent the proposals for the unpaired unknown SPs that are 50 pixels (or less) away from them. This propagation stops once all the unknown SPs are paired. Due to the small sizes of the matrices, building graphs with the delegates of a super-pixel, rather than all of its constituent pixels, has contributed to the computational efficiency of this stage in the pipeline.

The unknown SPs are allowed to share their half-pairs as follows: We compute the mean color-cartoon-texture feature for all the unknown and the Bg super-pixels in the image. Every unknown SP is then allowed to share its half-pair with the five spatially-nearest SPs, the five most similar SPs according to the mean color-cartoon-texture feature and the five most similar Bg super-pixels to account for isolated backgrounds. We also determine which of the unknown SPs has 25% or more of its constituent pixels already known in the trimap; we give them the symbol U_{SN} – the unknown SPs with significant number of known pixels. Every unknown SP is then assigned the most similar member to it among U_{SN}; for this stage only, the similarity is measured using the joint color-cartoon-texture-xy feature. During pair assessment, an unknown SP gets access to the known pixels in the U_{SN} member paired with it.

Figure 4 demonstrates the merit of the proposed technique with regards to gathering good half-pairs, as compared to methods that consider spatially nearby samples [JRC14] and the methods that determine the number of gathered samples based on the spatial distance between the unknown pixel and the known regions in the trimap [SRPC13]. In Fig. 4(a), the unknown SP is pointed to by a yellow arrow, and it is very close in space to a wrong Bg. The unknown SP's mean color value is shown on the right, surrounded by a red square. Its half-pairs computed using the proposed method are pointed to by cyan arrows and their mean color values are shown on the right. In (b), only the spatially-closest Bg super-pixels are considered, and their mean color values are shown on the right. We computed the same number of half-pairs for (a) and (b); however,



(a) Half-pairs With Graph Transduction



(b) Spatially-nearby Half-pairs

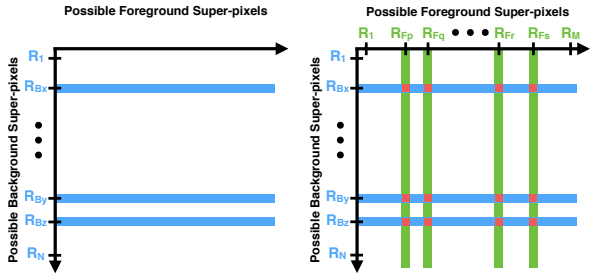
Figure 4: A demonstration of the benefit of using the proposed algorithm to determine a suitable half-pair for every unknown super-pixel. In (a) and (b), the unknown SP under consideration is pointed to by a yellow arrow, while its gathered half-pairs from Bg are pointed to by cyan arrows. Please see text for more details.

in (a) some of the shared SPs are duplicated. Gathered samples in (a) are clearly more similar (and thus suitable) to the unknown SP than those in (b).

The proposed method for half-pair computation also demonstrates more flexibility with regards to matching (or establishing correspondence between) known and unknown SPs, as compared to methods that match super-pixels by calculating the Euclidean distance in the color coordinate system between their mean color values [JRC14]. Matching two super-pixels with their mean color values would fail in highly-textured regions. For example, it would be difficult to match a mostly-green super-pixel with another SP containing green + red pixels. One way to overcome this problem is to use very small super-pixel sizes. However, this contradicts the main purpose of segmentation in the first place, that is the alleviation of the computational burden in the sample gathering and the pair assessment stages. In the proposed technique, if one delegate in a super-pixel SP_x has similar cartoon-texture feature to enough delegates in SP_y, the latter accepts the former as a proposal.

4.3. Punching the Pair Space

Figure 5(a) depicts the result of the best-half-pair computation step for a particular unknown SP (U_{sp}); this is shown on a 2D space that represents all the Fg and Bg super-pixels in an image. The same logic holds for the rest of the unknown super-pixels. As an example,



(a) Best Half-pair Computation (b) Pair space Punching

Figure 5: An illustration of the best half-pair computation and the punching steps for a single unknown super-pixel.

in this figure, U_{sp} is assigned to three (out of the N available) Bg super-pixels: R_{B_x} which is its best half-pair, in addition to R_{B_y} and R_{B_z} that it shares with its neighbours. It is worth mentioning that if the unknown SP under consideration would have preferred Fg super-pixels as its best half-pairs, we would have vertical streaks in Fig. 5(a).

Instead of the streaks in Fig. 5(a), we need to further narrow-down the search space to a few *patches* (or parts of streaks) in this space. Otherwise, the pair assessment step would be computationally inefficient as in [AKD14]. If we can determine the Fg super-pixels that best suit U_{sp}, given its best half-pairs, then those sought *patches* are the intersections between the previously-found Bg super-pixels (best half-pairs) and the most-suitable Fg super-pixels. Examples for those patches are shown as red squares in Fig. 5(b). Our final pair-space (*shortlisted pair space*) for that particular unknown SP will thus be the set of pairs comprised of the delegates of the SPs in these red squares. We name this step: *punching* the pair-space.

In order to determine the most-suitable complement half-pair among the Fg super-pixels for U_{sp}, we calculate the mean color feature for all the Fg super-pixels in the image; these will represent the F**s** in eqn. 2a. We also have the delegates of the best half-pairs for U_{sp} (all of them are Bg super-pixels in our example); these will represent the B**s** in eqn. 2a. For every delegate in U_{sp}, we retain the K foreground super-pixels that result in the least K values for the chromatic distortion (eqn. 2a); if U_{sp} has 3 delegates, we will have a bag of $3 \times K$ potential Fg super-pixels. Finally, we compute the L mode foreground super-pixels in the bag of potential Fg super-pixels; these will represent the complement half-pairs of U_{sp}. $K = 10$ and $L = 5$ in our experiments.

Before proceeding to the pair assessment stage, we wanted to check the goodness of the Fg/Bg pairs we gathered for every unknown pixel. Towards this goal, we used the training dataset of the online matting benchmark [mat] for which the ground-truth alpha maps are provided. For every delegate in every unknown SP, we checked all the possible alpha values that can be generated from the pairs it has access to. Then we calculated $MADG_\alpha$ – the minimum absolute deviation between the ground-truth alpha value of the delegate and the possible alpha values. The mean minimum absolute deviation over the delegates of each super-pixel is then computed, followed by the mean over all the unknown super-pixels in the im-

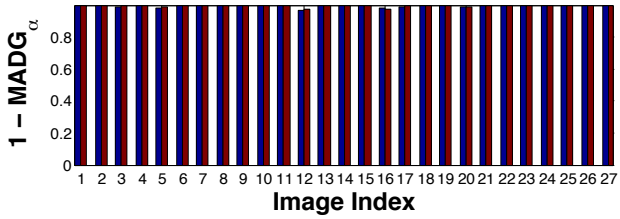


Figure 6: Assessing the gathered pairs for the unknown super-pixels over the whole training dataset of the benchmark (27 images). The blue bar represents the results for trimap 1, while the red bar represents the results for trimap 2. The closer the bars to 1 the better.

age. Ultimately, this computed value should be close to zero, and we calculated this value for the 27 images in the training dataset using the two provided trimaps. Figure 6 depicts a bar chart for the mean minimum absolute deviation of the 27 images; the blue bars represent trimap 1, and the red bars represent trimap 2. For the sake of clarity of presentation of the bar chart, we subtracted the computed values from 1 before plotting them, so the closer the bars to 1 in the figure the better.

4.4. Pair Assessment

In this stage, we assess the Fg/Bg pairs gathered for every unknown SP. Pair assessment is carried out for the delegates of the unknown SPs only, and the alpha values of the non-delegates were reconstructed by the weighting matrix computed using eqn. 6. For every unknown SP, the set of pairs is comprised of the delegates of the super-pixels depicted as red patches in Fig. 5(b). It is required to find the pair that best describes every delegate in the unknown SP under consideration. We propose two new formulations for the objective functions used to assess the pairs. The first formulation consists of two terms and is given by

$$\xi_{rs} = \frac{\xi_{color}}{\|F_u - B_v\|} \times \min\left\{\frac{W_1}{W_1+W_2}, \frac{W_2}{W_1+W_2}\right\} \quad \text{where} \qquad (8a)$$

$$W_1 = \exp(-\|I-F_u\|), W_2 = \exp(-\|I-B_v\|). \qquad (8b)$$

The numerator of the first term is the chromatic distortion of eqn. 2a. The denominator is inspired by the methods in [WC07b] and [RRG08], and is meant to favour Fg and Bg samples that are widely separated in the color coordinate system. The second term is inspired by the proposed sequential pair selection, and encourages the sparsity of alpha maps [LRAL08] using a different approach from that of the methods in [WC07b] and [RRG08]. Basically, we encourage samples (Fg *or* Bg) that are *distant*, in the color space, from the unknown pixel. Given that:

1. The unknown pixel, the Fg sample and the Bg sample form a triangle in the color coordinate system.
2. The Fg/Bg pair is required to be robust, i.e., $\|F_u - B_v\|$ should be large, to minimize ξ_{rs}.
3. One of the half-pairs (either the Fg sample *or* the Bg sample) is required to be distant from the unknown pixel to minimize ξ_{rs}.

Therefore, the other half-pair has to be close (in the color space) to the unknown pixel, satisfying the sparsity property of alpha maps. It is worth mentioning that if both half-pairs are distant from the unknown pixel, the pair would be ruled out by the chromatic distortion term. According to the premise of sequential pair selection, if the suitable half-pair, that is nearby in space, could be identified, the complement half-pair is not necessarily close in space; this is the reason our cost function did not include the spatial-closeness term as in the methods of [SRPC13, KEE15].

Inspired by the literature of the image completion problem [HS12, AKD15], we propose another formulation for the objective function. Image completion and matting have several aspects in common, one of which is the notion of information propagation from the known regions to unknown regions. Unknown regions are the hole regions in image completion, and the trimap's grey regions in matting. In [HS12], key patch offsets were obtained by computing statistics on patch correspondences. In this research, during the pair assessment stage, instead of picking the Fg/Bg pair that minimizes eqn. 8a, we considered the best H pairs instead ($H = 15$ in our experiments). Afterwards, we calculate the alpha values that correspond to these pairs, threshold them around 0.5 and take a vote. If the mode is $\alpha_m = 1$, the alpha of the unknown pixel under consideration will be the maximum alpha value among the H alpha values. Otherwise, the alpha of the unknown pixel under consideration will be the minimum alpha value among the H alpha values. This aligns with the sparsity property of alpha maps.

4.5. Trimap Expansion and Post-smoothing of Alpha Maps

Following other [KEE15, JRC14, SRPC13] recently proposed matting techniques, our pipeline started with a trimap expansion step and ended by smoothing the alpha maps. The condition for expansion is given by

$$(D(p,F_i) < E_{threshold}) \wedge (\|I_p - I_{F_i}\| \leq (C_{threshold} - D(p,F_i))), \quad (9)$$

which means that an unknown pixel p in the initial trimap will be considered as a definite Fg if the Euclidean distance in the spatial domain $D(p,F_i)$ between it and a foreground pixel F_i is less than $E_{threshold}$ *and* if the Euclidean distance between them in color space is less than $C_{threshold} - D(p,F_i)$. $E_{threshold}$ and $C_{threshold}$ are both constants in the spatial domain and color space respectively, and they were empirically set to 9. The same condition is applied for comparing the unknown pixels with the background pixels.

For smoothening the maps, the smoothing module of the publicly available code of [SRPC13] is used, in which a quadratic cost function in α is minimized. This function is the right-hand side of the equation given by

$$E = \alpha^T L\alpha + \gamma(\alpha-\hat{\alpha})^T \Gamma(\alpha-\hat{\alpha}) + \lambda(\alpha-\hat{\alpha})^T \Sigma(\alpha-\hat{\alpha}), \quad (10)$$

where α is a vector containing the values in the alpha map, $\alpha^T L\alpha$ is a smoothness term that encodes the smoothness constraints of [LLW08] in the Laplacian matrix L. The other two terms in the function serve as data terms. The vector $\hat{\alpha}$ is the values of the alpha map to be smoothed, $\gamma = 10^{-1}$ encodes the relative importance of the data and smoothness terms, Γ is a diagonal matrix whose zero entries for the known foreground and background pixels, and a confidence f for the unknown pixels. The presented results were obtained with $f = \beta \times \exp(-E_{min})$, where $\beta = 10$ and E_{min} is the minimum value attained by the right-hand side of eqn. 8a. Since

the pair assessment is carried out for the delegates of the unknown SPs only, E_{min} for the non-delegates was reconstructed from the weighting matrix computed using eqn. 6. The last term involves Σ which is a diagonal matrix with zero entries for unknown pixels and a value of 1 for the known pixels, while λ is a weighting parameter ($\lambda = 100$ in our experiments).

5. Results and Discussion

All the experiments were implemented using Matlab®, and were run on a PC with Intel Core2Quad 2.66GHz processor and 4GB of RAM. The proposed method was evaluated on the online matting benchmark [mat], and the results were uploaded on the 18th of June 2016. In this section, results obtained with eqn. 8a and with the voting formulation will be referred to as *TSPS-R* and *TSPS-RV* respectively. The complete ranking tables for each of them are included in the supplementary material. Nevertheless, in the publicly visible ranking on the benchmark website, the proposed technique should be represented by only one method (either *TSPS-R* or *TSPS-RV*); we chose the results obtained by the voting procedure because they attained a better average rank. Our method appears on the website under the name 'TSPS-RV Matting'.

For the algorithmic decisions and thresholds in section 4, inspired by the approach of [BRK*11], we aimed at setting our parameters so that we minimize the average MSE on the training dataset. We experimented with just a few parameters in the algorithm, over a discrete set of values, namely, the model parameter (σ) in the transduction step $\{20, 30, 40\}$, the SP size $\{10, 20, 30\}$, and the maximum distance for half-pair propagation $\{30, 50, 70\}$. The impact on the average MSE was found to be small – the percentage change is $< 5\%$ between the least and the maximum average MSE. The average computation time on the training dataset, using trimap 2, of our method is 330 seconds, while for [SRPC13], it is 313 seconds. The maximum computation time for our method is 514 seconds on image 21, for which [SRPC13] took 376 seconds, while the maximum computation time for [SRPC13] was 1036 seconds on image 25, for which our method took 231 seconds.

We start by comparing the ranking of the proposed method with the recently proposed hybrid approaches in the literature, namely, Comprehensive Sampling matting (CS) [SRPC13], Comprehensive Weighted Color and Texture matting (CWCT) [VR13], KL-Divergence Sparse Sampling matting (KL-Div), [KEE15], Sparse coded matting (SpCM) [JRC14] and Graph-based Sparse matting (GbSM) [JSRC16]. Table 1 indicates the position of each of the aforementioned methods in the benchmark tables, according to the four adopted metrics, namely, SAD, MSE, Gradient metric and connectivity metric [RRW*09]. The right-most column indicates the average position (or rank) of each method. Some SoA methods, such as the technique in [SRPC13] perform well according to the SAD, MSE and Gradient metrics, then their performance deteriorates remarkably according to the connectivity metric; this is the reason for calculating the average rank of every method across the four metrics to demonstrate the efficiency of the proposed method. The results summarized in the table shows that the performance of our method is comparable to the SoA, with an average rank equal to that of [KEE15] and a better average rank than the methods in [SRPC13], [VR13] and [JRC14].

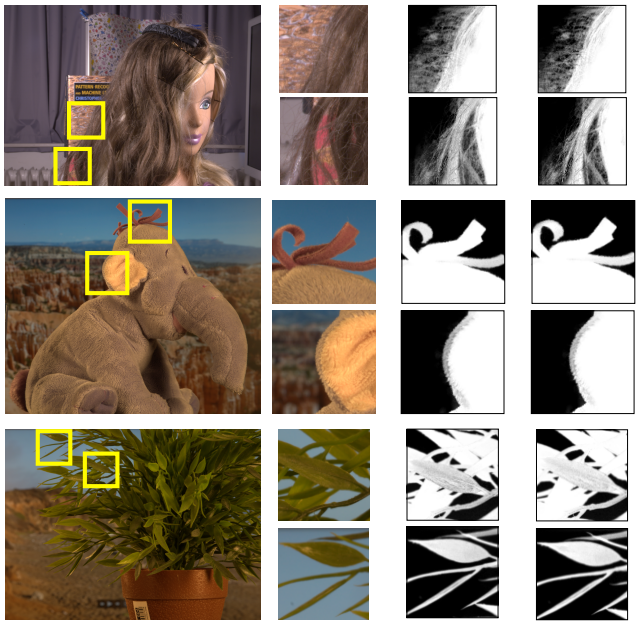


Figure 7: Three cases of subjective comparison between *TSPS-R* and *TSPS-RV* from the testing and training datasets. The 2nd column depicts enlarged patches from the 1st column. The 3rd and 4th columns show the result of *TSPS-R* and *TSPS-RV* respectively.

Results also show that the voting scheme (*TSPS-RV*) yields better performance than that of *TSPS-R* according to the first three metrics; the average rank of the former is also better. In Fig. 7, we show three cases of subjective comparison between the performance of *TSPS-R* and *TSPS-RV*. The chosen cases feature crisp boundaries and hairy boundaries, and the merits of the voting procedure is apparent in both of them.

To demonstrate the significance of the delegate-nomination-and-alpha-reconstruction step, we computed the alpha value of every unknown pixel in the training dataset, without nominating delegates for the unknown SPs. Under trimap 2, delegate nomination resulted in 71% average time reduction in the pair assessment stage at a cost of $< 1\%$ average increase in the MSE.

Table 1: Comparing the rankings of the proposed method with some of the SoA hybrid techniques, on the testing dataset, according to the four metrics of the matting online benchmark. The less the average rank the better. Please see text for more details.

Method	SAD	MSE	Grad.	Conn.	Average Rank
TSPS-R	13	10	17	6	11.5
TSPS-RV	7	7	16	13	10.75
CS	9	8	6	32	13.75
CWCT	10	11	15	25	15.25
KL-Div	6	4	3	30	10.75
SpCM	12	15	9	27	15.75
GbSM	4	6	2	29	10.25

We also used the publicly available code of [SRPC13] and [VR13] on [mat] and compared their performance with the proposed method objectively. We computed the alpha maps of the

training dataset, which is comprised of 27 images, using the two trimaps provided for that dataset. Since the ground-truth alpha maps are available, we computed the SAD attained by *TSPS-R*, *TSPS-RV*, [SRPC13] and [VR13]. Table 2 summarizes the results; for each trimap, we record the number of images for which each method attained the least SAD.

Table 2: Objective comparison of the proposed method with the methods in [SRPC13] (CS) and [VR13] (CWCT) over the whole training dataset of the benchmark (27 images). Each column shows the results obtained using one of the two provided trimaps. The table shows the number of images in which each corresponding method attained the least SAD.

Method	Trimap 1	Trimap 2
TSPS-R	4	0
TSPS-RV	8	14
CS	7	8
CWCT	8	5

6. Conclusion

We proposed a sampling-based method for image matting with performance close to the SoA techniques. Given that at least one good half-pair lies nearby in space to every unknown pixel, we used graph transduction to find that half-pair. A complement half-pair can then be computed by punching the Fg/Bg pair space. We showed the efficiency of our sample gathering method as compared to relying solely on spatial distance for sample gathering. We also proposed two new formulations for the objective functions that encourage sparse maps, favour robust pairs, and uses statistics over the best pairs to assign an alpha value for every pixel. Future directions include further exploration of optimal setting of the parameters values, incorporating other statistical measures in the cost function, and extending the research to video datasets.

References

[AKD14] AL-KABBANY A., DUBOIS E.: Improved global-sampling matting using sequential pair-selection strategy. In *Visual Information Processing and Communication V* (2014). 2, 5

[AKD15] AL-KABBANY A., DUBOIS E.: Image completion using image skimming. In *Visual Information Processing and Communication VI* (2015). 6

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SU S.: Slic superpixels compared to state-of-the-art superpixel methods. *PAMI 34*, 11 (2012), 2274 – 2282. 4

[BLMV11] BUADES A., LE T., MOREL J.-M., VESE L.: Cartoon+Texture Image Decomposition. *Image Processing On Line 1* (2011). 4

[BN04] BELKIN M., NIYOGI P.: Semi-supervised learning on Riemannian manifolds. *Machine Learning 56*, 1-3 (2004), 209 – 239. 3

[BRK*11] BLEYER M., ROTHER C., KOHLI P., SCHARSTEIN D., SINHA S.: Object stereo – joint stereo matching and object segmentation. In *CVPR* (2011). 7

[CLT12] CHEN Q., LI D., TANG C.-K.: KNN matting. In *CVPR* (2012). 1

[CZZ*13] CHEN X., ZOU D., ZHOU S., ZHAO Q., TAN P.: Image matting with local and nonlocal smooth priors. In *CVPR* (2013). 1, 2, 4

[DAK*08] DUCHENNE O., AUDIBERT J.-Y., KERIVEN R., PONCE J., SEGONNE F.: Segmentation by transduction. In *CVPR* (2008). 3, 4

[GO10] GASTAL E. S. L., OLIVEIRA M. M.: Shared sampling for real-time alpha matting. *Computer Graphics Forum 29*, 2 (2010), 575–584. 1

[GVV13] GAMMERMAN A., VOVK V., VAPNIK V.: Learning by transduction. *CoRR abs/1301.7375* (2013). 2

[HAvL05] HEIN M., AUDIBERT J.-Y., VON LUXBURG U.: From graphs to manifolds-weak and strong pointwise consistency of graph laplacians. In *the 18th Annual Conference on Learning Theory* (2005). 3

[HRR*11] HE K., RHEMANN C., ROTHER C., TANG X., SUN J.: A global sampling method for alpha matting. In *CVPR* (2011). 1, 2

[HS12] HE K., SUN J.: Statistics of patch offsets for image completion. In *ECCV* (2012). 6

[HWS*13] HE B., WANG G., SHI C., YIN X., LIU B., LIN X.: Iterative transductive learning for alpha matting. In *ICIP* (2013). 1

[JRC14] JOHNSON J., RAJAN D., CHOLAKKAL H.: Sparse codes as alpha matte. In *BMVC* (2014), Valstar M., French A., Pridmore T., (Eds.), BMVA Press. 1, 2, 4, 5, 6, 7

[JSRC16] JOHNSON J., SHAHRIAN E., RAJAN D., CHOLAKKAL H.: Sparse coding for alpha matting. *IEEE Transactions on Image Processing 99* (2016). 1, 2, 7

[KEE15] KARACAN L., ERDEM A., ERDEM E.: Image matting with KL-divergence based sparse sampling. In *ICCV* (2015), pp. 424–432. 1, 2, 6, 7

[LLW08] LEVIN A., LISCHINSKI D., WEISS Y.: A closed-form solution to natural image matting. *PAMI 30*, 2 (2008), 228–242. 1, 6

[LRAL08] LEVIN A., RAV-ACHA A., LISCHINSKI D.: Spectral matting. *PAMI 30*, 10 (2008), 1699 – 1712. 1, 2, 6

[mat] Alpha matting online benchmark. URL: http://alphamatting.com. 2, 5, 7

[RRG08] RHEMANN C., ROTHER C., GELAUTZ M.: Improving color modeling for alpha matting. In *BMVC* (2008). 2, 6

[RRW*09] RHEMANN C., ROTHER C., WANG J., GELAUTZ M., KOHLI P., ROTT P.: A perceptually motivated online benchmark for image matting. In *CVPR* (2009). 7

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE 290* (2000), 2323–2326. 4

[SAP*13] SHI Y., AU O., PANG J., TANG K., SUN W., ZHANG H., ZHU W., JIA L.: Color clustering matting. In *ICME* (2013). 1

[SRPC13] SHAHRIAN E., RAJAN D., PRICE B., COHEN S.: Improving image matting using comprehensive sampling sets. In *CVPR* (2013). 1, 2, 4, 6, 7, 8

[vlf] VLFeat library. URL: http://www.vlfeat.org/index.html. 4

[VR13] VARNOUSFADERANI E., RAJAN D.: Weighted color and texture sample selection for image matting. *IEEE Transactions on Image Processing 22*, 11 (2013), 4260 – 4270. 1, 2, 7, 8

[Wan11] WANG J.: Image matting with transductive inference. In *Computer Vision/Computer Graphics Collaboration Techniques*, vol. 6930. 2011, pp. 239–250. 3

[WC07a] WANG J., COHEN M.: Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision 3*, 2 (2007), 97 – 175. 2

[WC07b] WANG J., COHEN M.: Optimized color sampling for robust matting. In *CVPR* (2007). 1, 2, 6

[ZSLW15] ZHU Q., SHAO L., LI X., WANG L.: Targeting accurate object extraction from an image: A comprehensive study of natural image matting. *IEEE Transactions on Neural Networks and Learning Systems 26*, 2 (2015), 185–207. 2