Sparse Codes as Alpha Matte

Jubin Johnson
JUBIN001@e.ntu.edu.sg
Deepu Rajan
ASDRAJAN@ntu.edu.sg
Hisham Cholakkal
HISHAM002@ntu.edu.sg

School of Computer Engineering
Nanyang Technological University
Singapore

Abstract

In this paper, image matting is cast as a sparse coding problem wherein the sparse codes directly give the estimate of the alpha matte. Hence, there is no need to use the matting equation that restricts the estimate of α from a single pair of foreground (F) and background (B) samples. A probabilistic segmentation provides a confidence value on the pixel belonging to F or B, based on which a dictionary is formed for use in sparse coding. This allows the estimate of α from more than just one pair of (F,B) samples. Experimental results on a benchmark dataset show the proposed method performs close to state-of-the-art methods.

1 Introduction

Matting is a useful tool for image and video editing where foreground objects need to be extracted and pasted onto a different background. A pixel I_i in an image can be considered to be a composite of a foreground pixel F_i and a background pixel B_i such that

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

where α defines the opacity of the pixel and is a value in [0, 1], with 0 for background pixels and 1 for foreground pixels. Determining α for every pixel, also called *pulling an alpha matte*, is a highly ill-posed problem since it involves estimation of seven unknowns (3 color components for each of F_i and B_i and the α value) from three color components. The problem is constrained by providing additional information such as a three-level segmented image known as a trimap [4, 13, 14] or as scribbles [3, 10] specifying the definite foreground (F), definite background (B) and unknown regions.

There are three main approaches for image matting: sampling [5, 6, 8, 13, 14], alpha propagation [3, 7, 10, 15] and a combination of the two [4, 18]. In sampling-based approaches, a foreground-background sample pair is picked from few candidate samples taken from F and B regions by optimizing an objective function. This (F,B) pair is then used to estimate α at a pixel with color I by

$$\alpha_z = \frac{(I-B)(F-B)}{\|(F-B)\|^2}, \tag{2}$$

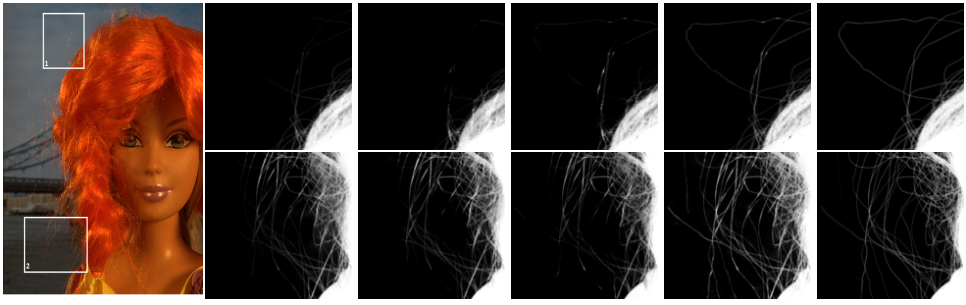
where $\|\cdot\|^2$ denotes the Euclidean distance. α-propagation based methods assume correlation between the neighboring pixels under some image statistics and use their affinities to propagate alpha values from known regions to unknown ones. The third category includes methods in which the matting problem is cast as an optimization problem in which the color sampling component forms the data term and the alpha propagation component forms the smoothness term; solving for the alpha matte becomes an energy minimization task.

The method proposed in this paper is based on sampling. However, there is one important difference between our method and other sampling-based approaches. As mentioned above, sampling-based approaches choose the best (F,B) pair from candidate samples through optimization and use it in the matting equation to determine α. This implies that a single (F,B) pair is used to determine α and the goodness of that pair depends on how well the optimization is done. In the proposed method, α is not determined based on a single (F,B) pair; instead, for a given unknown pixel, a bunch of F and B samples (which need not be in pairs) are considered together and weighted so that the sum of the weights provides the α. This allows the matting framework to determine α based on more relevant F and B samples than with only one of each. Thus, the proposed method does not use the matting equation in eq. (2) to estimate α.

The tool that enables the use of more than one F and B sample to directly estimate α as a sum of the samples' weights is sparse coding. The success of sparse coding in various computer vision tasks such as face recognition [20], super-resolution [21] and image classification [22] can be attributed to the fact that images generally lie on low-dimensional subspaces or manifolds from which representative samples could be picked, creating a sparse representation based on an appropriate basis [19]. Here, a dictionary of color values of F and B pixels is employed to determine the sparse codes for a pixel in an unknown region. The sum of the sparse codes for F pixels directly provides the α. Initially, the pixels in the trimap are classified into high-confidence and low-confidence based on probabilistic segmentation. The size of the dictionary for high-confidence pixels is smaller than that for low-confidence pixels. Pixels in the unknown region are then sparse coded with respect to the adaptive dictionary as described in section 3.2.

The only other method that uses sparse coding for matting is called compressive matting [23]. There are two important differences between their method and ours. First, they use a fixed window from which the dictionary is formed. This severely restricts the possibility of obtaining a very incoherent dictionary since images are generally smooth over a small neighborhood. Second, they map the sparse codes to the α values using a ratio of the l_2-norm of the sparse codes of F pixels to the sum of l_2-norms of sparse codes of F and B regions. In our approach, the sum of the sparse codes for F directly give α. Finally, they do not present quantitative results on test images from [1] which contains a standard benchmark dataset on which state-of-the-art matting algorithms are evaluated. We show quantitative and qualitative results on this dataset. Fig. 1(a) shows an input image with 2 windows depicting fine hairy regions. Closed-form matting [10] oversmooths the matte leading to loss of fine detail (fig. 1(b)), while weighted color and texture matting [13] in fig. 1(c) uses texture feature in addition to color . Comprehensive sampling [14] (fig. 1(d)) utilizes a large sampling set and the matting equation, but does not perform as well as the proposed method, shown in fig. 1(e), which is visually closer to the ground truth given in fig. 1(f)).

The paper is organized as follows. We review related work and its shortcomings in sec. 2 followed by description of our approach in sec. 3. Experimental results are discussed in sec. 4, and we conclude the paper in sec. 5.



(a) (b) (c) (d) (e) (f)

Figure 1: Visual comparison of alpha matte generated by our proposed method using sparse coding with other state-of-the-art methods. Top and bottom rows show zoomed in regions of windows 1 and 2 respectively. (a) Input image, (b) Closed form [10], (c) Weighted color and texture [13], (d) Comprehensive sampling [14], (e) Proposed method and (f) Ground truth.

2 Related Work

Sampling-based matting methods can be divided into parametric and non-parametric methods. Parametric methods [5, 16] describe the foreground and background samples as arising from parametric low-order statistical models and estimate alpha based on the distance of the unknown pixels to the known color distributions. They generate large fitting errors for textured regions where it is insufficient to model the higher-order statistics of color distribution. It tends to produce weak mattes when the trimap is coarse, leading to unreliable correlations between unknown pixels and F and B samples. Non-parametric methods [6, 8, 13, 14] collect a subset of known F and B samples and estimate the alpha matte from the best (F,B) pair found through an optimization process. Different sampling strategies and final pair-selection criteria distinguish these methods. Samples are collected from spatially nearest boundary pixels [18], by shooting rays from the unknown to the known pixels [6], by selecting all the pixels on the known region boundaries [8], or by selecting a comprehensive set of samples from within the known regions through Gaussian mixture model (GMM) based clustering [14]. The final (F,B) pair, found through optimization of an objective function, controls the quality of the final matte. Texture is used in addition to color as a feature for matting in [13] to address the problem of overlap in color distributions of F and B. Such approaches fail to produce a good matte when the F and B samples are nearby in the color space, or when the collected sample set fails to correlate with the actual color at the unknown pixel. Chen *et al*. [4] formulate the sampling-based alpha estimate in [18], smoothness Laplacian along with locally linear embedding as a weighted graph and obtain a closed-form solution for the matte. Zheng *et al*. [24] treat matting as a supervised learning problem and use support vector regression to learn the alpha-feature model from known samples. This method is also affected by similar F and B colors since the learned feature model could be inaccurate.

α-propagation relies on the affinity between neighboring pixels to propagate the matte. Levin *et al*. [10] used a color line model for a small neighborhood of pixels to propagate α across unknown regions. The color line model assumption does not hold in highly textured regions due to strong edges that block the propagation of alpha. KNN matting [3] considers nonlocal principle to formulate the affinities among K nearest neighbors in a nonlocal neighborhood. Similar strategies to construct the Laplacian are employed in [7, 15] to overcome

the limitation of the color line assumption. However, the smoothness assumption is insufficient to deal with complex images as high correlation among similar F and B colors wrongly propagates the matte. An extensive survey on image matting is available in [17].

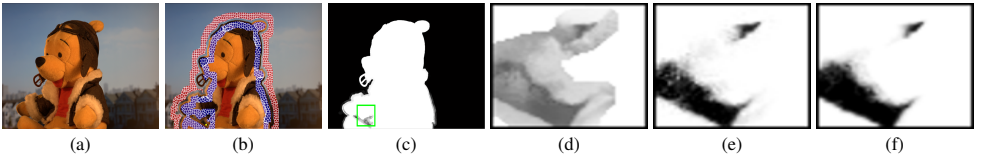
3 Proposed method

As indicated earlier, the motivation of the proposed method is to determine α from a bunch of F and B samples (as opposed to a single pair of (F,B) samples in the matting equation) in a sparse coding framework so that the sparse codes directly provide the α values. The unknown pixels are categorized into high-confidence and low-confidence to allow a smaller dictionary size for the former and a larger one for the latter. This is done through probabilistic segmentation of the image. Sparse coding of feature vector at each pixel using the appropriate dictionary yields the α value.

3.1 Confidence measure of unknown pixels

The feature vector used for coding is the 6-D vector $[R\ G\ B\ L\ a\ b]^T$ consisting of the concatenation of the two color spaces. Initially, a universal set of samples from known regions is obtained. In order to reduce the sample space for the universal set, we cluster known F and B pixels in a band of width 40 pixels around the trimap into superpixels using SLIC algorithm [2]. The mean color of each superpixel represents the F and B samples that make up the universal set. Fig. 2(a) shows an original image and fig. 2(b) shows the superpixels from the foreground and background regions in blue and red, respectively. From the universal set, samples are chosen to form separate dictionaries for high-confidence and low-confidence regions.

We desire a confidence measure that indicates how well the colors of foreground and background regions are separated in the neighborhood of the unknown pixel. If an unknown pixel has high-confidence, then it implies that the F and B colors are well separated and consequently, the size of the dictionary can be small since we can then ensure that the dictionary will be formed from highly incoherent samples. Overlapping of F and B color distributions is one of the problems that recent sampling based approaches like [14] try to address. Pixels with low-confidence come from areas that have potentially complex and overlapping color distributions requiring a larger dictionary so that the variability in the sample set can be captured.



(a) (b) (c) (d) (e) (f)

Figure 2: Illustration of sampling strategy and foreground probability map. (a) Input image, (b) superpixel samples, (c) foreground probability map, (d) zoomed low-confidence region, (e) estimated alpha and (f) ground truth.

Since the feature used for coding is color and the complexity of a region for matting is dependent on the overlap of foreground and background colors, we use probabilistic segmentation as a cue to determine the confidence of a pixel. We adopt a non-parametric sampling-based probability measure [9] to determine a probability map. The foreground probability

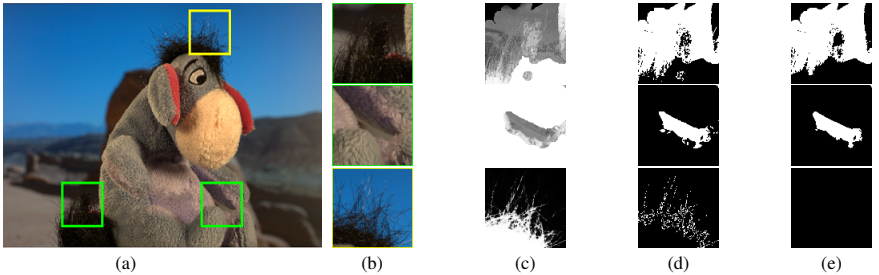
map indicates the probability of a given pixel belonging to the foreground. The probability that a pixel I_i belongs to the foreground is given by

$$p(I_i) = \frac{p_f(I_i)}{p_f(I_i) + p_b(I_i)}, \tag{3}$$

where $p_f(I_i)$ is the foreground color probability value given by

$$p_f(I_i) = exp\left(-\frac{\sum_{k=1}^{m} \|c(I_i) - c(f_k)\|^2}{m \cdot \delta}\right), \tag{4}$$

where $c(\cdot)$ is the RGB color value, m is the number of spatially close foreground samples (closest is measured in terms of Euclidean distance from unknown pixel to centroid of a superpixel) and δ is a weighting constant. In our experiments, we fix m and δ as 10 and 0.1 respectively. A similar expression is applicable to $p_b(I_i)$. Eq. (3) provides a confidence value on whether a pixel belongs to the foreground or background. In areas of complex color distributions, the spatial distance of the unknown pixel to foreground and background pixels are similar. It has been observed that in such cases, the confidence measure as given by eq. (3) ranges from about 0.3 to 0.7. However, it is also observed that in regions with thin hairy structures whose color is significantly different from the background, the confidence measure also lies between 0.3 and 0.7. In order to avoid such regions where the color distributions do not overlap, we consider a 7×7 neighborhood and classify the unknown pixel as low-confidence if the number of pixels with $p(I_i)$ in $[0.3, 0.7]$ is larger than 35 (fixed empirically). Fig. 2(c) shows the probability map with higher intensity denoting higher probability for a pixel to belong to foreground. Fig. 2(d) is a zoomed in region of a low-confidence region where there is an overlap in the foreground and background color distributions. Fig. 3(a) shows an image marked with low-confidence regions. The green windows are areas of complex color distributions. However, the yellow window at the top shows hairy region with separable F and B colors. Fig. 3(b) shows the marked windows separately. The zoomed regions of the probability map in fig. 3(c) show that even in the hairy region, we can observe probabilities in $[0.3, 0.7]$ as shown in the thresholded mask in fig. 3(d) . The additional neighborhood condition helps us to classify only the true complex color areas as the low-confidence mask, as shown in fig. 3(e) in which 1 is assigned to every low-confidence pixel and 0 to high-confidence pixels.



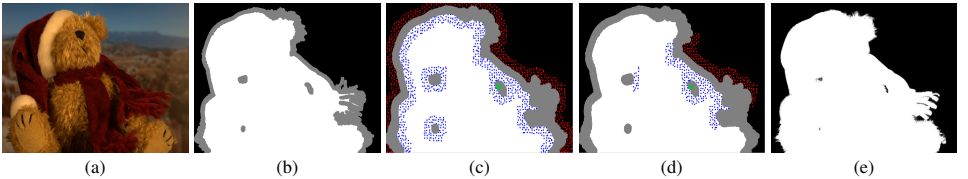
(a) (b) (c) (d) (e)

Figure 3: Defining low-confidence regions in complex areas. (a) Input image and (b) zoomed patches, corresponding (c) probability map, (d) thresholded mask for pixels in $[0.3, 0.7]$ and (e) low-confidence mask for each region.

3.2 Sparse coding to generate α

The size of the dictionary for an unknown pixel depends on whether it is classified as low-confidence or high-confidence. If the pixel is of high-confidence, i.e., the color distributions of F and B are well separated for the probabilistic segmentation to give high value for $p(I_i)$, then the dictionary is formed from the 40 (chosen empirically) spatially closest F and B samples. Note that the samples here are the mean colors of the closest superpixels. As mentioned earlier, in such regions, the samples would be sufficiently incoherent for sparse coding.

The low-confidence regions are potential areas of overlapping color distributions. In this case the dictionary size is larger. Thus, for a given unknown pixel, one-third of the spatially closest superpixels from the definite F and B regions constitute the dictionary. Fig. 4(a) and fig. 4(b) show an input image and its corresponding trimap. The universal sample set of F and B regions in blue and red respectively, is shown in fig. 4(c). For a given unknown pixel of low-confidence shown in green in fig. 4(d), the final dictionary is a larger subset of the universal sample set than that of high-confidence pixels.



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

Figure 4: Dictionary formation followed by sparse coding to generate the matte. Foreground and background regions are shown in blue and red respectively. (a) Input image, (b) trimap, (c) universal set, (d) final dictionary and (e) estimated alpha.

The 6-D color vector, which forms the feature vector for coding, is normalized to unit length. Given the dictionary $\mathbf{D}$ for an unknown pixel i, its alpha matte is determined by sparse coding as

$$\beta = argmin \|v_i - \mathbf{D}\beta_i\|_2^2 \quad s.t \quad \|\beta_i\|_1 \leq 1 \; ; \; \beta_i \geq 0, \tag{5}$$

where v_i is the feature vector at i composed of (R,G,B,L,a,b). The sparse codes β_i are generated using a modified version of the Lasso algorithm [11]. The sparse coding procedure is presented with an appropriate set of F and B samples and the sparse coefficients sum up to less than or equal to 1. In order to avoid negative sparse coefficients, the second constraint forces all coefficients to be positive. The sparse codes corresponding to atoms in the dictionary that belong to foreground regions are added to form the α for the unknown pixel i.e.

$$\alpha = \sum_{p \in F} \beta^{(p)}. \tag{6}$$

Hence, the sparse codes directly provide the value of α. Fig. 4(e) shows the alpha matte extracted from the sparse codes using our approach on the input image in fig. 4(a). A good quality matte is obtained even when the unknown region is well inside the foreground which is a challenge for propagation-based methods.

3.3 Pre and Post-processing

We use a pre-processing step to expand the known regions to unknown regions based on certain chromatic and spatial thresholds. An unknown pixel I_i is considered as foreground if, for a pixel $I_j \in F$ [14]

$$(D(I_i,I_j) < E_{thr}) \wedge (\left\|I_i - I_j\right\| \leq (C_{thr} - D(I_i,I_j)), \tag{7}$$

where $D(I_i,I_j)$ is the Euclidean distance between pixel I_i and I_j in the spatial domain and E_{thr} and C_{thr} are empirically set to 15 and 4, respectively. A similar formulation is applied to expand background regions.

The alpha matte obtained by sparse coding is further refined to obtain a smooth matte by considering the correlation between neighboring pixels' matte. We adopt the post-processing approach adopted by [14] where a cost function consisting of the data term and a confidence value together with a smoothness term consisting of the matting Laplacian [10] is minimized with respect to α. The confidence value at a pixel i is

$$O_s(i) = O_{sprec}(i) \cdot O_{colrec}(i), \tag{8}$$

where $O_{sprec}(i) = e^{-|v_i - \hat{v}_i|}$ and it measures the confidence in reconstructing the input feature vector based on the sparse coefficients, $O_{colrec}(i) = e^{-\|I_i - (\alpha_i F_i + (1-\alpha_i)B_i)\|^2}$ measures the chromatic distortion.

The cost function for smoothing is given by:

$$\alpha = argmin \;\; \alpha^T L \alpha + \lambda(\alpha - \hat{\alpha})^T D(\alpha - \hat{\alpha}) + \gamma(\alpha - \hat{\alpha})^T \Gamma(\alpha - \hat{\alpha}), \tag{9}$$

where $\hat{\alpha}$ is the estimated sparse matte from eq. (5), L is the matting Laplacian [10], λ and γ are weighting constants denoting the relative importance of the data and smoothness terms in the function. D is a diagonal matrix with values 1 for known foreground and background pixels and 0 for unknown, while the diagonal matrix Γ has 0 values for known pixels and O_s(confidence value) for unknown pixels. Fig. 2(e) shows the final alpha obtained on the low confidence region after post-processing and closely approximates the ground truth shown in fig. 2(f).

4 Experimental Results

The effectiveness of the proposed method is evaluated using the benchmark dataset [12] for image matting. It consists of 35 images covering a wide range of transparency of pixels - from opaque to fully transparent. 27 images form the training set with publicly available ground truth. The remaining 8 images form the test set whose ground truth is hidden from the public and is used for benchmark evaluation and ranking [1].

Table. 1 shows the contribution of each part of the proposed algorithm. The performance is measured by the sum of absolute difference (SAD) averaged over the test images and the three types of trimaps for each image - small, large and user. Universal set refers to the the dictionary formed from the band of superpixels along the boundary of the unknown region. The high SAD error observed indicates that simply increasing the sample set for sparse coding does not result in better estimates due to the presence of high color-correlated samples to the unknown pixel that are far away. Final dictionary refers to the refined dictionary obtained after taking into account the confidence of the pixels. There is an improvement over

Table 1: Evaluation of effect of each step in our method using sum of absolute difference (SAD) error averaged over all test images for three types of trimaps.

Term	Trimap Small	Large	User
Universal Set	23.5	19.3	19.8
Final dictionary	18.4	13.0	13.6
Laplacian refinement	**10.8**	**9.4**	**6.6**

the universal set because the spatial constraint controls the false correlated samples from entering the sample set. Laplacian refinement is the post-processing step which maintains the smoothness of the matte. This stage shows a marked improvement over the sparse coding stage because in formulating the sparse code optimization, we do not consider the smoothness constraint. The combined effect of all the terms produces the lowest error rates.

The test dataset contains the 8 most difficult subset of images from the dataset. Qualitative comparisons with the state-of-the-art methods on the elephant and plant images are shown in Fig. 5. Additional results are presented in the supplementary material. The high overlap of F and B colors make these images difficult for matting as shown in the zoomed patches in fig. 5 (b). The bottom row in fig. 5 (c) depicts a background region being interpreted as foreground due to the inaccurate learning model for SVR matting [24]. ITM matting [7] wrongly propagates the background into the leaves as well as the tail of the elephant in fig. 5 (e). Comprehensive sampling [14], which uses a larger sampling set is able to differentiate the elephant tail but still classifies the leaves as mixed pixels in the bottom row of fig. 5 (d). The smooth prior in LNSP matting [9] oversmooths the matte leading to a band between the foreground leaves and the background in fig. 5 (f). Our method is able to reduce the above said artifacts and extract out a visually superior matte in these areas as shown in fig. 5 (g).

A quantitative evaluation by the alpha matting website [1] of the proposed method is shown in Table 2. The proposed method gives promising results quantitatively as well, ranking within the top 5 in both SAD and gradient error and 7th in MSE. Table. 2 shows the relative ranking of the top 10 matting algorithms today using the SAD and MSE error mea-

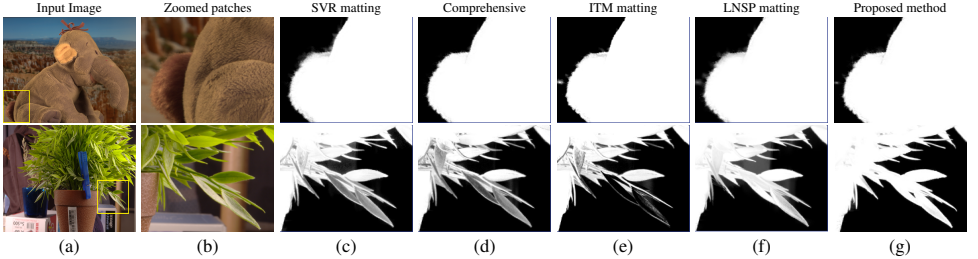


Figure 5: Qualitative comparison of proposed method on elephant and pineapple images with top 4 methods at [1]. (a) Input image, (b) zoomed windows, (c) SVR matting [24], (d) Comprehensive sampling [14], (e) Iterative transductive matting [7], (f) LNSP matting [9] and (g) Proposed method. Zoomed in regions show the effectiveness of our method.

Table 2: Ranks of different matting methods with respect to sum of absolute differences (SAD), mean squared error (MSE) and gradient error measures on benchmark dataset evaluated at [1] as on submission date (April 28, 2014).

SAD **MSE**

Method	Avg. small rank	Avg. large rank	Avg. user rank	Overall rank	Method	Avg. small rank	Avg. large rank	Avg. user rank	Overall rank
LNSP matting	**3.9**	**5.5**	9.5	6.3	LNSP matting	**4**	**4.6**	8.6	**5.8**
Iterative Transductive matting	9.1	6.8	7.3	7.7	CCM	9.4	6.6	**5.5**	7.2
Comprehensive Sampling	6.8	7.5	9.1	7.8	Comprehensive Sampling	7.1	7.6	8.6	7.8
Comprehensive Weighted C&T	8.9	8.5	7.8	8.4	SVR matting	11.3	7.4	7.6	8.8
Proposed method	10.9	9.4	**6.6**	9	Comprehensive Weighted C&T	8.8	9.3	8.5	8.8
SVR matting	11	8.3	7.6	9	Weighted Color & Texture	9.8	11.6	10.9	10.8
Weighted Color & Texture	8.1	10.9	9.8	9.6	Proposed method	12.3	11.4	8.9	10.8
CCM	11.9	9.3	8.5	9.9	Global sampling matting	7.5	13.5	12.1	11
Shared matting	10.4	12.8	9.4	10.8	Iterative Transductive matting	12.8	10.3	11.9	11.6
Global sampling matting	9.8	14	12.6	12.1	Shared matting	11.9	14	11.3	12.4

Gradient error

Method	Avg. small rank	Avg. large rank	Avg. user rank	Overall rank
LNSP matting	**5.6**	**6**	9.8	**7.1**
Comprehensive Sampling	7.6	6.8	7.9	7.4
CCM	11.1	8	7.9	9
SVR matting	10.8	9.5	**6.8**	9
Proposed method	10.9	8.3	9.4	9.5
Segmentation-based matting	12.3	8.1	8.6	9.7
Global sampling matting	10	10	9.3	9.8
Shared matting	10	10.3	9.5	9.9
Improved Color matting	11.4	10.3	8.8	10.1
Comprehensive Weighted C&T	11.6	13.5	10.9	12

sures. We achieve the best ranking among all the methods on the user trimap. The methods that are ranked higher utilize the matting equation in estimating the matte, or use features like texture in addition to color, while we show that our color based sparse coded alternative can still achieve good results on the benchmark images.

Table 3: Quantitative comparison of our method on mean squared error (MSE) measure with compressive matting [23].

Image		Compressive matting [23]	Our method
GT01	Trimap 1	5.0×10^{-4}	$\mathbf{1.46 \times 10^{-4}}$
	Trimap 2	8.1×10^{-4}	$\mathbf{2.18 \times 10^{-4}}$
GT18	Trimap 1	12.0×10^{-4}	$\mathbf{3.5 \times 10^{-4}}$
	Trimap 2	15.0×10^{-4}	$\mathbf{4.3 \times 10^{-4}}$

Finally, we compare our results with the only other method that uses sparse coding in their formulation [23]. As noted earlier, they do not provide results evaluated by [1] on the test images. Instead they provide quantitative evaluation on 2 training images (GT01,GT18), each on two trimaps, by giving the MSE that they obtained. We compared our results for the same images in Table 3 where we show that we achieve a four-fold decrease in MSE error on both the images.

5 Conclusion

In this paper, we have shown that by removing the restriction of a single $F - B$ pair in estimating the matte, we are able to approximate the matting problem towards sparse representation. The special constraints that are enforced on the sparse coding algorithm works well in pulling a high quality matte that was verified by experimental evaluations and ranks amongst the current state-of-the-art. In the future, we plan to extend our approach to videos as well.

References

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

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012.

[3] Q. Chen, D. Li, and C.-K. Tang. Knn matting. In *CVPR*, 2012.

[4] X. Chen, D. Zou, S. Z. Zhou, Q. Zhao, and P. Tan. Image matting with local and nonlocal smooth priors. In *CVPR*, 2013.

[5] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *CVPR*, 2001.

[6] E. S. Gastal and M. M. Oliveira. Shared sampling for real-time alpha matting. In *Computer Graphics Forum*, volume 29, pages 575–584, 2010.

[7] B. He, G. Wang, C. Shi, X. Yin, B. Liu, and X. Lin. Iterative transductive learning for alpha matting. In *ICIP*, 2013.

[8] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun. A global sampling method for alpha matting. In *CVPR*, 2011.

[9] J. Ju, J. Wang, Y. Liu, H. Wang, and Q. Dai. A progressive tri-level segmentation approach for topology-change-aware video matting. In *Computer Graphics Forum*, volume 32, pages 245–253, 2013.

[10] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *PAMI*, 30(2):228–242, 2008.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[12] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *CVPR*, 2009.

[13] E. Shahrian and D. Rajan. Weighted color and texture sample selection for image matting. In *CVPR*, 2012.

[14] E. Shahrian, D. Rajan, B. Price, and S. Cohen. Improving image matting using comprehensive sampling sets. In *CVPR*, 2013.

[15] Y. Shi, O. C. Au, J. Pang, K. Tang, W. Sun, H. Zhang, W. Zhu, and L. Jia. Color clustering matting. In *ICME*, 2013.

[16] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, 2005.

[17] J. Wang and M. F. Cohen. Image and video matting: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(2):97–175, 2007.

[18] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007.

[19] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[21] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[23] S. M. Yoon and G. Yoon. Alpha matting using compressive sensing. *Electronics letters*, 48(3):153–155, 2012.

[24] Z. Zhang, Q. Zhu, and Y. Xie. Learning based alpha matting using support vector regression. In *ICIP*, 2012.