

Soft Color Segmentation and Its Applications

Yu-Wing Tai, *Student Member, IEEE*, Jiaya Jia, *Member, IEEE Computer Society*, and
Chi-Keung Tang, *Senior Member, IEEE Computer Society*

Abstract—We propose an automatic approach to soft color segmentation, which produces soft color segments with an appropriate amount of overlapping and transparency essential to synthesizing natural images for a wide range of image-based applications. Although many state-of-the-art and complex techniques are excellent at partitioning an input image to facilitate deriving a semantic description of the scene, to achieve seamless image synthesis, we advocate a segmentation approach designed to maintain spatial and color coherence among soft segments while preserving discontinuities by assigning to each pixel a set of soft labels corresponding to their respective color distributions. We optimize a global objective function, which simultaneously exploits the reliability given by global color statistics and flexibility of local image compositing, leading to an image model where the global color statistics of an image is represented by a Gaussian Mixture Model (GMM), whereas the color of a pixel is explained by a local color mixture model where the weights are defined by the soft labels to the elements of the converged GMM. Transparency is naturally introduced in our probabilistic framework, which infers an optimal mixture of colors at an image pixel. To adequately consider global and local information in the same framework, an alternating optimization scheme is proposed to iteratively solve for the global and local model parameters. Our method is fully automatic and is shown to converge to a good optimal solution. We perform extensive evaluation and comparison and demonstrate that our method achieves good image synthesis results for image-based applications such as image matting, color transfer, image deblurring, and image colorization.

Index Terms—Color image segmentation and image synthesis.

1 INTRODUCTION

THIS paper presents an algorithm to perform soft color segmentation given a color image, producing overlapping and transparent segments suitable for a wide range of important image-based applications such as image matting [2], [26], [5] (Figs. 20 and 21), color transfer between images [24], [33] (Figs. 22, 23, and 24), image deblurring [16] (Fig. 25), image denoising [11], [21] (Fig. 26), and image colorization [38] (Fig. 27). Unlike traditional approaches, our segmentation approach is designed to address a large class of image-based problems, which require soft segments with an appropriate amount of overlapping and transparency. This approach is translated into an alternating optimization (AO) algorithm, which is more straightforward to implement than many state-of-the-art and complex segmentation techniques that are geared to produce a semantic segmentation of the input image for tasks such as recognition and interpretation.

We present a probabilistic framework to address soft color segmentation, where a global objective function is modeled by global and local parameters. These parameters are alternately optimized until convergence. Since our goal is to maintain natural color and texture transition across soft segments rather than assigning semantics to each segmented region, it is sufficient to model the *global* color statistics of an

image by Gaussian Mixture Model (GMM). Each pixel's color can be explained by a *local* mixture of colors derived from the optimized GMM weighted by the inferred soft labels. Our algorithm is shown to converge to a good optimal solution.

Our segmentation goal is different but related to that of traditional segmentation approaches. In this paper, we evaluate and compare our automatic method with *k*-means clustering [10], Mean Shift [6], Expectation-Maximization (EM) [8], [1], Watershed [37], J-value Segmentation (JSeg) [9], Data-Driven Markov Chain Monte Carlo (DDMCMC) [35], Information Bottleneck [13], Multiscale graph-based techniques [27], [12], statistical region merging [19], and user-assisted image matting [2], [26], [5] to show that better or comparable results are obtained in terms of region transparency, color coherence, and spatial coherence. Our method produces results comparable to the Bayesian matting [5] in terms of extracting a foreground matte from an image. In [5], a user-supplied trimap is required, whereas our method is fully automatic. Our proposed algorithm is applied to various image applications such as transferring color between images, image deblurring, image denoising, and colorizing gray-scale images.

The paper is organized as follows: Section 2 reviews the related work on color and image segmentation. Section 3 describes in detail our AO algorithm, which estimates the optimal global and local model parameters. We performed experiments to show the good optimality and convergence of our AO algorithm, whereas the theoretical aspects of these issues are addressed in [3]. In Section 4, we evaluate and analyze our AO algorithm using synthetic and real images. Results and comparisons are presented in Section 5. In Section 6, we apply our soft color segmentation to various image synthesis applications and show that significantly better results can be obtained by employing soft segments produced by our algorithm. We conclude our paper in Section 7.

• Y.-W. Tai and C.-K. Tang are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
E-mail: yuwing@gmail.com, cktang@cs.ust.hk.

• J. Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong.
E-mail: leojia@cse.cuhk.edu.hk.

Manuscript received 7 Oct. 2005; revised 9 May 2006; accepted 14 Nov. 2006; published online 3 May 2007.

Recommended for acceptance by S.-C. Zhu.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0545-1005. Digital Object Identifier no. 10.1109/TPAMI.2007.1168.

Some pertinent results in this paper have appeared in an earlier version [33], where the focus is on the application of soft color segmentation to color transfer applications. This paper goes beyond [33] by contributing in a new problem formulation, a new optimization algorithm for soft segmentation, and extensive experimental comparison.

2 RELATED WORK

We review in this section previous work most relevant to ours in image segmentation.

2.1 Hard Segmentation

The Watershed algorithm [37] is a region-based technique where “watershed” lines are used to mark the boundaries of regions. The morphological operations of closing (or opening) are then introduced to smooth ridges (or fill in valleys) of the topographical map produced. This method is sensitive to intensity changes, so a large number of small regions are usually produced. The Watershed algorithm is often used as a preprocessing step to obtain an oversegmented image to preserve as much detail as possible for further processing.

The EM algorithm, which is one form of AO, was employed in [1] to address the problem of color and texture segmentation. The joint distribution of color and texture is modeled using a mixture of Gaussians in a six-dimensional (6D) space (three dimensions for color and three dimensions for texture). Because the grouping is performed on a 6D space and no spatial coordinates are considered, small and fragmented regions are produced. A separate spatial grouping step is then applied to obtain pixel-connected components.

JSeg [9] is an unsupervised algorithm for color and texture segmentation. The first color quantization step creates a class map of color labels. The second spatial segmentation step uses the class map to create a J-image to identify color or texture regions. The two steps are sequential, where the second step is dependent upon the results produced by the first one.

The Mean Shift segmentation [6] is a clustering algorithm that performs color and texture segmentation. The algorithm takes as input a feature bandwidth, a spatial bandwidth, and a minimum region area (in pixels). Salient clusters are successively extracted by applying a kernel in the feature space, which shifts toward the significant cluster center. Because the feature space is a high-dimensional one, in order to reduce the number of shifts for achieving fast convergence, a set of random locations in the feature space is usually considered for selecting the initial location with the highest density of feature vectors.

Graph-based approaches for image segmentation and grouping have gained much attention. The Normalized Cuts [28] is one such algorithm that uses a global criterion on the total dissimilarity among (and similarity within) different pixel groups, where discrete region labels output after graph optimization.

The statistical region merging was proposed in [19], which consists of a semisupervised statistical region refinement algorithm for color image segmentation. Based on certain principles on perceptual grouping and an image generation model, a simple merging method was proposed to produce visually coherent color segments.

2.2 Soft Segmentation

The concept of soft segmentation is not a new one. For example, the traditional k -means clustering [10] can be considered as one form of soft color segmentation. In essence, each point in the feature space is associated with a label and its confidence value is calculated using some function related to the distance of each converged cluster. If spatial and color coordinates are considered simultaneously for preserving the coherence, the resulting feature space becomes sparse and high dimensional, making the method vulnerable to local optima.

The split-and-link algorithm [22] computes overlapping segments in a pyramidal framework where the levels are overlapped so that each pixel is a descendant of four others in the pyramid. The linking is done based on similarity to ameliorate some problems in initial splitting. In [13], unsupervised image clustering was proposed to cluster images, subject to minimizing the loss of mutual information between the clusters and image features. The proposed clustering can be regarded as a soft label classification, where GMMs are used to model the feature space. A graph-based approach was proposed in [27], which combines multiscale measurements of intensity contrast, texture differences, and boundary integrity. The method optimizes a global measurement over a multiscale pyramidal structure of the image and maintains a fuzzy relationship between nodes in successive levels. A follow-up of the work [12] made use of multiscale aggregation of filter responses to handle complex textures.

In [18], a clustering-based algorithm was proposed to segment color textures, where multiscale smoothing and initial clustering are first performed to determine a set of core clusters to which a subset of pixels should belong. Soft labels are then assigned and updated iteratively at all other pixels at multiple scales.

A unifying framework for image segmentation known as DDMCMC was proposed in [35], which exploits Markov Chain dynamics to explore the complex solution space and achieves a nearly global optimal solution regardless of initial segmentations. Since features occur at multiple scales, the method incorporates intrinsic ambiguities in image segmentation and utilizes data-driven techniques (such as clustering where soft assignment is made in the feature space) for computing importance proposal probabilities.

Fuzzy connectedness [36] groups image elements (pixels) by assigning a strength of connectedness to every possible path between every possible pair of image elements. The connectedness strength is related to the region where the image element belongs. An image element can be associated with more than one region with different connectedness strength. The method has been extensively experimented in segmenting delicate tissues from medical images.

In computer graphics, the class of image matting algorithms can be considered as a special case of soft color segmentation. Smith and Blinn [29] were the first to present the blue screen matting systematically. Knockout [2] is one method that gathers color samples by estimating the foreground and background with weighted averages of the pixel colors within a neighborhood. Ruzon and Tomasi [26] sampled colors by a mixture of unoriented Gaussians and proposed to use the color with the maximum probability. In Bayesian matting [5], Chuang et al. formulated the matting problem using Bayesian optimization, where the maximum a posteriori (MAP) estimation is performed to estimate the

TABLE 1
Notations Used in This Paper

Notation	Meaning
I	Image space
$I(x, y)$	RGB color at pixel (x, y)
\mathcal{G}	The 3D GMM $\mathcal{G} = \{G(i; \mu_i, \Sigma_i)\}, i = 1, \dots, N$
μ_i	Mean of Gaussian component $i, i = 1, \dots, N$
Σ_i	Covariance matrix of Gaussian component $i, i = 1, \dots, N$
$\mathcal{L}(x, y)$	Soft labels to \mathcal{G} at pixel (x, y) : $\mathcal{L}(x, y) = \{\ell_i(x, y) i = 1, \dots, N\}$
$\mathcal{C}(x, y)$	Compositing colors derived from \mathcal{G} at pixel (x, y) : $\mathcal{C}(x, y) = \{c_i(x, y) i = 1, \dots, N\}$

For local parameters, the coordinates (x, y) may be skipped when we refer to the entire set of parameters across the whole image. Note that μ_i , $I(x, y)$, and $c_i(x, y)$ are red, green, blue (RGB) tuples. N is the number of Gaussians.

optimal alpha matte for foreground extraction. This method performs pixelwise optimization without exploiting adequate spatial coherence information. Grabcut [25] and Poisson matting [30] consider matte continuity among pixels. Note that all the above matting techniques are not automatic, requiring some forms of user interaction, usually in the form of a user-supplied trimap, which specifies the “definite foreground,” “definite background,” and “uncertain” regions, in order to produce satisfactory matting results.

2.3 Comparison with Our Work

The approaches described in the previous subsection have made significant contributions in image segmentation. However, they are not suitable to be used in image-based application, which requires soft color segments with appropriate amount of overlapping and transparency due to one or more of the following reasons:

- Although the previous methods produce excellent image segmentation results for natural images, they are designed to solve the general segmentation problem, which may not be ideal for image-based applications. For instance, to obtain a good image interpretation, general image segmentation aims to cluster similar patterns or textures. However, as shown in the result section, the details inside each pattern should be preserved so that distinct colors will not get mixed up in the synthesized image. Furthermore, the resulting segments reported in the above literature are mostly hard segments, which do not preserve smooth color transition among segments.
- To maintain spatial and color coherence, many algorithms concatenate spatial and feature vectors resulting in a sparse and high-dimensional feature space. A careful initialization is therefore needed to ensure fast convergence to a reasonable solution.
- On the other hand, spatial grouping and color clustering are considered by certain approaches as independent rather than interdependent processes, so errors produced in one step are propagated to the following steps.
- All matting methods are interactive requiring somewhat careful initialization (for example, a user-supplied trimap).

In this paper, we propose an automatic color segmentation approach to address the above issues. To maintain spatial and color coherence, instead of using a high-dimensional feature space, an AO framework is adopted: Our method optimizes a global objective function that combines the advantages given by global color statistics and local image compositing. Using a global objective function, global and local information is properly integrated by using a Markov network that optimizes for the soft labels at each image pixel, subject to spatial and color coherence while preserving underlying discontinuities. The global color statistics of an image is specified by the inferred 3D GMM. A local mixture model is introduced to account for the observed color at each pixel, where soft labels are introduced to naturally encode transparent and overlapping regions in our probabilistic framework. We propose an AO algorithm to estimate an optimal set of model parameters. Readers may refer to [3] for AO’s convergence to global optimal solution. Our method also proves the convergence empirically by extensive experiments on a variety of natural and complex images. We demonstrate the efficacy of our approach in a wide variety of image-based applications and show that less human interaction or better results can be obtained using our soft color segmentation method.

3 SOFT COLOR SEGMENTATION

Our approach in soft color segmentation takes into consideration both global and local color information within the same framework. *Global color statistics* models the overall colors of the input image. *Local color compositing* models the mixture of colors contributing to the observed color at a pixel, where the colors are derived from the optimized global statistics.

In our framework, the global and local models cooperate with each other subject to the spatial and color coherency constraints in each pixel’s neighborhood, where the similarity within the same region and dissimilarity across regions are preserved. An AO scheme [7], [3] is adopted to iteratively optimize the global and local parameters. The notations are summarized in Table 1.

3.1 Problem Modeling and Formulation

Global color statistics. By observing a large number of *natural* images (Fig. 1), we find that the global color statistics can be represented by a set of overlapping regions and modeled by a

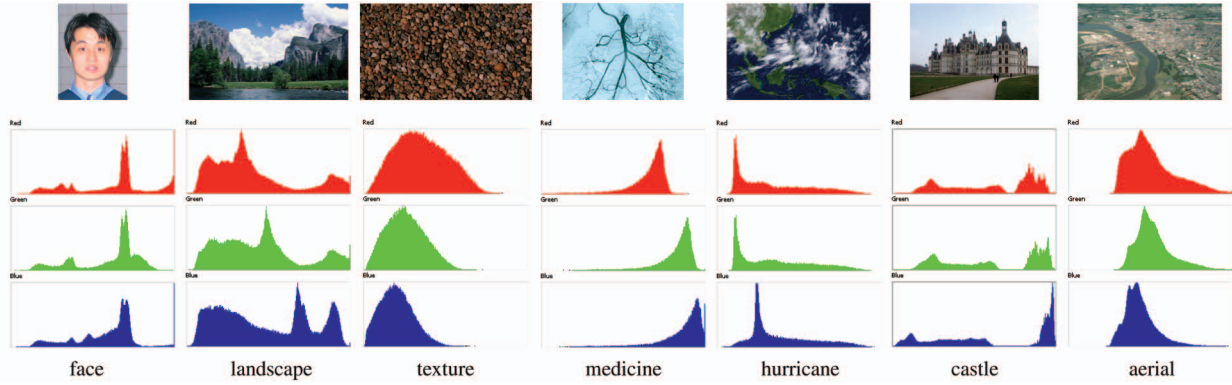


Fig. 1. The global color statistics of a natural image can be modeled by a mixture of Gaussians. Top: Typical images. Bottom: For illustration, the three rows shown are the histograms of the RGB color channels, whereas 3D Gaussians are used in our computation.

mixture of Gaussians. In other words, the color of a pixel can be predicted by a GMM, where each Gaussian is 3D for encoding the RGB channels and is parameterized by the corresponding mean and covariance matrix. Let us denote the GMM by $\mathcal{G} = \{G(i; \mu_i, \Sigma_i)\}$, where $1 \leq i \leq N$ and N is the number of Gaussians, μ_i is the mean, and Σ_i is the covariance matrix.

Note that global color statistics alone such as GMM are not sufficient for semantic segmentation or image understanding, explaining many approaches that use GMM (for example, [1], [13]) introduce additional constraints to address the semantic segmentation problem. On the other hand, given that the global color statistics of a natural image can be well modeled by a GMM and that our goal is to infer overlapping color segments that may not separate well in the color space, without any a priori information, it is a natural choice to use GMM to represent global statistics in the color space. On the other hand, our method does not purely rely on GMM. As we shall describe shortly, a local mixture model is used where Markov Random Fields (MRFs) are incorporated so that spatial and color coherences are optimized at each pixel's neighborhood during the AO process.

Local color compositing. By generalizing the image compositing equation in [23], we propose the following model for local color compositing at each pixel (x, y) , which uses a mixture of colors to encode overlapping and transparency:

$$I(x, y) = \sum_{i=1}^N \ell_i(x, y) c_i(x, y), \quad (1)$$

where $\mathcal{L}(x, y) = \{\ell_i(x, y) | i = 1, 2, \dots, N\}$ is the set of soft labels corresponding to the N color segments (or Gaussians) in \mathcal{G} and $\mathcal{C}(x, y) = \{c_i(x, y) | i = 1, 2, \dots, N\}$ is the set of compositing colors.

Problem formulation. In summary, our color model consists of a global term representing the color model for the whole image described by \mathcal{G} and a local term for each pixel described by \mathcal{L} and \mathcal{C} . The set of unknowns consists of

$$\{c_i(x, y), \ell_i(x, y), \mu_i, \Sigma_i | i = 1, 2, \dots, N \text{ and } (x, y) \in I\}.$$

We formulate the problem of soft color segmentation as follows: Given an image I and the number of Gaussians (segments) N , we seek to optimize for the global and local model parameters \mathcal{G} , \mathcal{L} , and \mathcal{C} subject to the necessary spatial and color coherency and discontinuity inherent in the image. Using the Minimum Description Length principle [15], [14]

(also used in the EM algorithms in [1], [13]), which avoids model overfitting, the number of Gaussians N can be inferred. In practice, we fix the value of N , where $N < 10$ in all cases we tested. The weights of insignificant Gaussians will be converged to zero. We describe in the following our AO algorithm to solve for the optimal \mathcal{L} , \mathcal{C} , and \mathcal{G} iteratively.

3.2 The Global Optimization Function

Given a color image I , we maximize the a posteriori probability to infer all unknowns. In a Bayesian framework, we formulate the problem as follows:

$$\begin{aligned} \arg \max_{\mathcal{G}, \mathcal{L}, \mathcal{C}} P(\mathcal{G}, \mathcal{L}, \mathcal{C} | I) &\propto \arg \max_{\mathcal{G}, \mathcal{L}, \mathcal{C}} P(I | \mathcal{G}, \mathcal{L}, \mathcal{C}) P(\mathcal{C} | \mathcal{G}, \mathcal{L}) P(\mathcal{L}, \mathcal{G}) \\ &\propto \arg \max_{\mathcal{G}, \mathcal{L}, \mathcal{C}} P(I | \mathcal{L}, \mathcal{C}) P(\mathcal{C} | \mathcal{G}, \mathcal{L}) P(\mathcal{L}), \end{aligned} \quad (2)$$

where $P(I | \mathcal{L}, \mathcal{C}) = P(I | \mathcal{G}, \mathcal{L}, \mathcal{C})$ since $I(x, y)$ is not directly related to \mathcal{G} by (1), $P(\mathcal{L}, \mathcal{G}) = P(\mathcal{L})P(\mathcal{G})$ since in our model $P(\mathcal{L})$ and $P(\mathcal{G})$ are independent. Finally, $P(\mathcal{G})$, the global color statistics, is assumed to be a uniform distribution without any prior knowledge. So, $P(\mathcal{G})$ is omitted in (2).

3.2.1 Matching Likelihood $P(I | \mathcal{L}, \mathcal{C})$

We assume the observation noise follows an independent identical distribution (i.i.d.), so we define the likelihood $P(I | \mathcal{L}, \mathcal{C})$ as a product of likelihoods at each pixel:

$$\begin{aligned} P(I | \mathcal{L}, \mathcal{C}) &\propto \prod_{(x, y)} P(I(x, y) | \mathcal{L}(x, y), \mathcal{C}(x, y)) \\ &= \prod_{(x, y)} \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(I(x, y) - \sum_{i=1}^N \ell_i(x, y) c_i(x, y))^2}{2\sigma_c^2}\right), \end{aligned} \quad (3)$$

which models the fidelity how $I(x, y)$ conforms to the local model $\sum_{i=1}^N \ell_i(x, y) c_i(x, y)$ with standard deviation σ_c . We set $\sigma_c = 0.1$ in all our experiments.

3.2.2 Matching Likelihood $P(\mathcal{C} | \mathcal{G}, \mathcal{L})$

We define the matching likelihood $P(\mathcal{C} | \mathcal{G}, \mathcal{L})$ in the same way, using a product of likelihoods at each pixel

$$P(\mathcal{C} | \mathcal{G}, \mathcal{L}) \propto \prod_{(x, y)} P(\mathcal{C}(x, y) | \mathcal{G}, \mathcal{L}(x, y)). \quad (4)$$

Based on the information theory [14], the relative entropy (Kullback-Leibler divergence) of $\mathcal{L}(x, y)$ and $P(c_i(x, y)|G_i)$ is defined as

$$\sum_{i=1}^N \ell_i(x, y) \log \frac{\ell_i(x, y)}{P(c_i(x, y)|G_i)}, \quad (5)$$

where $\ell_i(x, y)$ measures the confidence that the value c_i is generated by Gaussian element i . The relative entropy can be minimized as we maximize its negation

$$\begin{aligned} & \arg \min_c \sum_{i=1}^N \ell_i(x, y) \log \frac{\ell_i(x, y)}{P(c_i(x, y)|G_i)} \\ & \propto \arg \max_c \sum_{i=1}^N \left(\log P(c_i(x, y)|G_i)^{\ell_i(x, y)} - \log \ell_i(x, y)^{\ell_i(x, y)} \right) \quad (6) \\ & \propto \arg \max_c \sum_{i=1}^N \log P(c_i(x, y)|G_i)^{\ell_i(x, y)} - \sum_{i=1}^N \log \ell_i(x, y)^{\ell_i(x, y)}. \end{aligned}$$

In the representation of the above likelihood, \mathcal{L} is the observation. Omitting $\sum_{i=1}^N \log \ell_i(x, y)^{\ell_i(x, y)}$ will not influence the estimation result. Thus, (6) can be simplified as follows when we take the exponent:

$$\begin{aligned} & \arg \max_c \sum_{i=1}^N \log P(c_i(x, y)|G_i)^{\ell_i(x, y)} \\ & \propto \arg \max_c \prod_{i=1}^N P(c_i(x, y)|G_i)^{\ell_i(x, y)}. \quad (7) \end{aligned}$$

Given $G_i(i; \mu_i, \Sigma_i)$, the likelihood $P(c_i(x, y)|G_i)$ is modeled by the deviation of $c_i(x, y)$ from the Gaussian G_i

$$\begin{aligned} & P(c_i(x, y)|G_i) \\ & \propto \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left(-\frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2} \right). \quad (8) \end{aligned}$$

Substituting (8) into (7), we obtain the function to be maximized:

$$\begin{aligned} & \prod_{i=1}^N \frac{1}{\left((2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}} \right)^{\ell_i(x, y)}} \\ & \exp \left(-\ell_i(x, y) \frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2} \right). \quad (9) \end{aligned}$$

3.2.3 Prior $P(\mathcal{L})$

Effective modeling of the prior is very important in producing good results in a Bayesian formulation. To maintain spatial and color coherence while introducing transparency among overlapping segments, we apply the MRFs, which assert that the conditional probability of a site in the field depends only on the information of its neighboring sites.

MRFs are effective in avoiding noise or highly fragmented segments while maintaining color smoothness across and within the segmented regions. In our soft color segmentation, although the resulting segments may sometimes not correspond to those produced by manual or semantic segmentation, as will be shown in the experimental section, they are adequate for image-based applications, which require soft segments for synthesizing seamless images.

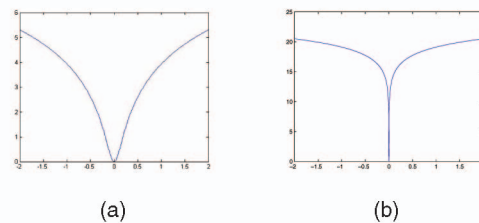


Fig. 2. The robust function for encoding the discontinuity-preserving function: plotting the Lorentzian estimator $\log(1 + \frac{1}{2}(\frac{x}{\sigma})^2)$ versus x with (a) $\sigma = 0.1$ and (b) $\sigma = 0.0005$. In all cases, the curves are bounded when $x \rightarrow \pm\infty$, which is more robust than the usual norm-squared function (that is, the unbounded x^2) in terms of encoding the error term.

In all unknowns, the prior $P(\mathcal{L})$ encodes the probability that one pixel falls in different segments. We apply the following pairwise constraint:

$$P(\mathcal{L}) \propto \prod_{(x, y)} \prod_{(x', y') \in \mathcal{N}(x, y)} \exp(-\psi(\mathcal{L}(x, y), \mathcal{L}(x', y'))), \quad (10)$$

where $\psi(\mathcal{L}(x, y), \mathcal{L}(x', y'))$ is the joint clique potential function of sites (x, y) and its first-order neighborhood sites (x', y') .

We adopt the Lorentzian estimator (Fig. 2), a robust function [4] that preserves discontinuity implicitly by penalizing any occurrence of color discontinuity between sites (x, y) and (x', y') . The joint potential function is defined as follows:

$$\psi(\mathcal{L}(x, y), \mathcal{L}(x', y')) = \log \left(1 + \frac{1}{2} \left(\frac{\|\mathcal{L}(x, y) - \mathcal{L}(x', y')\|}{\sigma_p} \right)^2 \right), \quad (11)$$

where σ_p is set to 0.1 in all our experiments. The use of a robust discontinuity-preserving function is typical, for instance, in MRF stereo [17], [31] and MRF photometric stereo [34], [40].

3.3 The Alternating Optimization

Combining the likelihoods of (3) and (9) and the prior in (11), we solve the following global optimization problem by maximizing the posterior function:

$$\begin{aligned} & \arg \max_{g, \mathcal{L}, c} \prod_{(x, y)} \frac{1}{\sqrt{2\pi}\sigma_c} \exp \left(-\frac{(I(x, y) - \sum_{i=1}^N \ell_i(x, y)c_i(x, y))^2}{2\sigma_c^2} \right) \\ & \prod_{(x, y)} \prod_{(x', y') \in \mathcal{N}(x, y)} \left(\frac{1}{1 + \frac{1}{2} \left(\frac{\|\mathcal{L}(x, y) - \mathcal{L}(x', y')\|}{\sigma_p} \right)^2} \right) \\ & \prod_{(x, y)} \prod_{i=1}^N \frac{1}{\left((2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}} \right)^{\ell_i(x, y)}} \\ & \exp \left(-\ell_i(x, y) \frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2} \right). \quad (12) \end{aligned}$$

Obviously, this optimization problem cannot be directly solved due to the large number of unknowns. To solve (12), we propose an AO algorithm [7]: By fixing the values of a subset of parameters in each iteration of the optimization process, the global objective function is maximized by optimizing each subproblem alternately.

3.3.1 Fix \mathcal{G}, \mathcal{C} , and Optimize \mathcal{L}

In the first subproblem, we use the optimized values of \mathcal{G} and \mathcal{C} from the previous iterations to optimize \mathcal{L} . In the literature of probabilistic graph models, a Markov network is an undirected graph where the nodes $\{x_s\}$ are used to encode hidden variables and the nodes $\{y_s\}$ are used to encode observed variables. Taking $X = \{x_s\}$ and $Y = \{y_s\}$, the posterior $P(X|Y)$ can be factorized as

$$P(X|Y) = \prod_s \psi_s(x_s, y_s) \prod_s \prod_{t \in \mathcal{N}(s)} \psi_{st}(x_s, x_t), \quad (13)$$

where $\psi_{st}(x_s, x_t)$ is the compatibility matrix between nodes x_s and x_t , encoding the cost between two neighboring pixels, and $\psi_s(x_s, y_s)$ is the local evidence for node x_s , representing the observation probability from the hidden variables $p(y_s|x_s)$.

In this subproblem, it can be observed that our posterior functions (2) and (13) are the same if we define

$$\psi_{st}(x_s, x_t) = \exp(-\psi(\mathcal{L}(x, y), \mathcal{L}(x', y'))), \quad (14)$$

$$\psi_s(x_s, y_s) = P(I(x, y)|\mathcal{L}(x, y), \mathcal{C}(x, y))P(\mathcal{C}(x, y)|\mathcal{G}, \mathcal{L}(x, y)). \quad (15)$$

Thus, finding the MAP solution of (12) is equal to solving the Markov network where each hidden node encodes $\mathcal{L}(x, y)$. In belief propagation terms, N is the number of labels, and \mathcal{L} is passed as one-dimensional (1D) messages among the hidden nodes. Thus, the MAP estimation can be solved (approximately) by loopy belief propagation via a message passing procedure [20]. Because $\ell_i(x, y) \in [0, 1]$ is fractional and $\sum_{i=1}^N \ell_i(x, y) = 1$, transparency is naturally encoded at (x, y) by the soft labels. The memory and computational complexity of our belief propagation algorithm is $O(ZN)$ and $O(TZN)$, respectively, where Z is the total number of pixels, N is the number of Gaussians, and T is the number of iterations.

3.3.2 Fix \mathcal{G}, \mathcal{L} , and Optimize \mathcal{C}

In the second subproblem, we use the optimized values of \mathcal{G} and \mathcal{L} obtained from the previous iterations to optimize \mathcal{C} . Since \mathcal{L} is fixed, $P(\mathcal{L})$ is a constant. Equation (12) in this subproblem is simplified to

$$\begin{aligned} & \arg \max_{\mathcal{C}} \prod_{(x,y)} \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(I(x, y) - \sum_{i=1}^N \ell_i(x, y)c_i(x, y))^2}{2\sigma_c^2}\right) \\ & \prod_{(x,y)} \prod_{i=1}^N \frac{1}{\left((2\pi)^{\frac{3}{2}}|\Sigma_i|^{\frac{1}{2}}\right)^{\ell_i(x,y)}} \\ & \exp\left(-\ell_i(x, y) \frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2}\right). \end{aligned} \quad (16)$$

By taking the logarithm of (16), we obtain a polynomial in several variables with constant coefficients. We perform pixelwise optimization by taking the partial derivative at each pixel and setting the equations equal to zero to compute the stationary points. In the following, the constant terms are ignored if they disappear after taking the partial derivative:

$$\begin{aligned} & \frac{\partial}{\partial c_i} \left(\frac{(I(x, y) - \sum_{i=1}^N \ell_i(x, y)c_i(x, y))^2}{2\sigma_c^2} \right. \\ & \left. + \sum_{i=1}^N \ell_i(x, y) \frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2} \right) = 0. \end{aligned} \quad (17)$$

Hence, we obtained the following $3N \times 3N$ linear equation system:

$$\begin{aligned} & \begin{bmatrix} \ell_1(x, y)\Sigma_1^{-1} + \mathbf{I}\ell_1(x, y)^2/\sigma_c^2 & \dots & \mathbf{I}\ell_1(x, y)\ell_N(x, y)/\sigma_c^2 \\ \vdots & \ddots & \vdots \\ \mathbf{I}\ell_1(x, y)\ell_N(x, y)/\sigma_c^2 & \dots & \ell_N(x, y)\Sigma_N^{-1} + \mathbf{I}\ell_N(x, y)^2/\sigma_c^2 \end{bmatrix} \begin{bmatrix} c_1(x, y) \\ \vdots \\ c_N(x, y) \end{bmatrix} \\ & = \begin{bmatrix} \ell_1(x, y)\Sigma_1^{-1}\mu_1 + I(x, y)\ell_1(x, y)/\sigma_c^2 \\ \vdots \\ \ell_N(x, y)\Sigma_N^{-1}\mu_N + I(x, y)\ell_N(x, y)/\sigma_c^2 \end{bmatrix}, \end{aligned} \quad (18)$$

where \mathbf{I} is the identical matrix and $\sigma_c = 0.1$. Note that when $\ell_i(x, y)$ equals to 0, we do not need to estimate $c_i(x, y)$ since the equation is already balanced, and $c_i(x, y)$ has no effect to the objective function. Therefore, we can reduce the dimension of the linear system by removing the entries with $\ell_i(x, y) = 0$. In implementation, we only estimate the color of the pixels with $|\ell_i(x, y)^t - \ell_i(x, y)^{t-1}| > \varepsilon$, where t is the iteration number, and ε is a small threshold set to be 0.01 in all our experiments. The above linear system is solved by using singular value decomposition (SVD).

3.3.3 Fix \mathcal{L}, \mathcal{C} , and Optimize \mathcal{G}

In the last subproblem, we use the optimized values of \mathcal{C} and \mathcal{L} obtained from the previous iterations to optimize the unknown \mathcal{G} . Since \mathcal{L} and \mathcal{C} are fixed, $P(I|\mathcal{L}, \mathcal{C})$ and $P(\mathcal{L})$ are constant in this subproblem. Taking the negative logarithm of our global objective function (12), we obtain

$$\begin{aligned} & \arg \max_{\mathcal{G}} \prod_{(x,y)} \prod_{i=1}^N \frac{1}{\left((2\pi)^{\frac{3}{2}}|\Sigma_i|^{\frac{1}{2}}\right)^{\ell_i(x,y)}} \\ & \exp\left(-\ell_i(x, y) \frac{(c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2}\right) \\ & = \arg \min_{\mathcal{G}} \sum_{(x,y)} \sum_{i=1}^N \ell_i(x, y) \\ & \left(\frac{3 \log(2\pi) + \log(|\Sigma_i|) + (c_i(x, y) - \mu_i)^T \Sigma_i^{-1} (c_i(x, y) - \mu_i)}{2} \right). \end{aligned} \quad (19)$$

Taking the derivative with respect to μ_i and setting it to zero, we get

$$\sum_{(x,y)} \Sigma_i^{-1} (c_i(x, y) - \mu_i) \ell_i(x, y) = 0, \quad (20)$$

which is reduced to

$$\mu_i = \frac{\sum_{(x,y)} \ell_i(x, y) c_i(x, y)}{\sum_{(x,y)} \ell_i(x, y)}. \quad (21)$$

To find Σ_i , we can write (19) as (constant terms are ignored here since they disappear after taking derivative)

$$\frac{1}{2} \sum_{(x,y)} \sum_{i=1}^N \ell_i(x,y) \left(\text{trace} \left(\Sigma_i^{-1} (c_i(x,y) - \mu_i)(c_i(x,y) - \mu_i)^T \right) - \log(|\Sigma_i^{-1}|) \right). \quad (22)$$

Taking derivative with respect to Σ_i^{-1} , we get

$$\frac{1}{2} \sum_{(x,y)} \sum_{i=1}^N \ell_i(x,y) (2\mathbf{M} - \text{diag}(\mathbf{M})) = 2S - \text{diag}(S), \quad (23)$$

where $\mathbf{M} = (c_i(x,y) - \mu_i)(c_i(x,y) - \mu_i)^T - \Sigma_i$ and $S = \frac{1}{2} \sum_{(x,y)} \sum_{i=1}^N \ell_i(x,y) \mathbf{M}$. Setting the derivative equal to 0, that is, $2S - \text{diag}(S) = 0$, implies that $S = 0$, which gives

$$\frac{1}{2} \sum_{(x,y)} \sum_{i=1}^N \ell_i(x,y) \left((c_i(x,y) - \mu_i)(c_i(x,y) - \mu_i)^T - \Sigma_i \right) = 0. \quad (24)$$

Rearranging the equation, we obtain

$$\Sigma_i = \frac{\sum_{(x,y)} \ell_i(x,y) (c_i(x,y) - \mu_i)(c_i(x,y) - \mu_i)^T}{\sum_{(x,y)} \ell_i(x,y)}. \quad (25)$$

Thus, (21) and (25) together give the optimal \mathcal{G} .

3.4 Summary

In summary, our AO algorithm is described as follows: Initialize the unknowns (\mathcal{G} , \mathcal{L} , and \mathcal{C}), and iterate the following steps until convergence or reaching a fixed number of iterations:

- Compute \mathcal{L} by loopy belief propagation with (13), (14), and (15).
- Compute \mathcal{C} with (18).
- Compute \mathcal{G} with (21) and (25).

To initialize the optimization, we use the results produced by k -means clustering, where the mean and covariance for each Gaussian component of GMM \mathcal{G} is initialized as the mean and covariance of the corresponding cluster. At each pixel, \mathcal{L} is initialized as the soft labels obtained via k -means clustering. $\mathcal{C}(x,y)$ is set to $I(x,y)$ for all pixels (x,y) .

3.5 Convergence

Although AO guarantees convergence to one type of global optimal solution [3], in this section, we experiment the convergence of our AO algorithm for soft color segmentation. In each respective step of estimating \mathcal{L} , \mathcal{G} , or \mathcal{C} , we use the values obtained for the two unknowns in the previous step to compute the maximum of the third unknown, which makes the estimated value of $P(\mathcal{G}, \mathcal{L}, \mathcal{C}|I)$ increase monotonically. Hence, the convergence of our method is guaranteed, and the maximum of $P(\mathcal{G}, \mathcal{L}, \mathcal{C}|I)$ can be reached upon convergence. In implementation, we terminate our iterations in either one of the following situations:

- $\sum_{0 \leq i < N} (\mu_i^t - \mu_i^{t-1})^2 < \eta$, where η is a predefined threshold and t is current iteration, or
- $t = 40$.

To demonstrate the optimization efficiency, we run our AO algorithm for a total of 100 iterations for each case.

Fig. 3 shows the graphs plotting the negative logarithm of the global objective function against the number of iterations. Note that maximizing the objective function (2) is equivalent to minimizing its negative logarithm. Typically, our method converges within 30 to 40 iterations. Fig. 4 shows some intermediate results during the AO iterations. We use different colors to represent each Gaussian components in \mathcal{G} and label $I(x,y)$ as color i if $\ell_i(x,y)$ is largest in $\{\ell_1(x,y), \ell_2(x,y), \dots\}$. The output shown in the figure is hard color segmentation to facilitate visual evaluation. As the number of iterations increases, the spatial connectivity and color homogeneity are progressively refined until final convergence. In Section 5, we shall show that our converged soft segmentation allows smooth and natural color transition for a wide range of image synthesis applications and produces satisfactory results.

4 EVALUATION AND ANALYSIS

This section presents examples to evaluate our AO algorithm to perform soft color segmentation. Evaluation on synthetic and real data are first presented, followed by the study of the effects of \mathcal{C} and \mathcal{G} on the soft labels \mathcal{L} estimation.

4.1 Synthetic Image

Fig. 5a shows a synthetic image used in our evaluation. This example presents a challenge because a single pixel's color can be explained by as many as six colors. The result shows that our automatic method is capable of segmenting the image into six coherent regions. Fig. 5b shows a resynthesized image $\sum_{i=1}^6 \ell_i(x,y) c_i(x,y)$ generated by compositing the estimated \mathcal{L} and \mathcal{C} at each pixel. The image difference between the input image and the synthesized image $|I(x,y) - \sum_{i=1}^6 \ell_i(x,y) c_i(x,y)|$ is shown in Fig. 5c. We achieve an average pixel error of 0.0147, given by

$$\frac{\sum_{(x,y)} |I(x,y) - \sum_{i=1}^N \ell_i(x,y) c_i(x,y)|}{Z},$$

where Z is the total number of pixels. We plot the soft labels \mathcal{L} at sample pixels in Fig. 5d to show the transparent boundaries of the resulting soft regions. Fig. 5e shows the segmentation results of each converged color region, displayed as $I_i(x,y) = \ell_i(x,y) c_i(x,y)$, $1 \leq i \leq 6$.

4.2 Real Image

In this section, we use real images to evaluate our method in the presence of rich textures and colors. We first use Poisson matting [30] to extract the graffiti c^* and the alpha matte ℓ^* from the original image. They are composited onto a set of new background images rich in textures and colors (shown in Fig. 6a). Finally, we segment the graffiti from the composited images using our soft color segmentation method. The segmented graffiti, which is shown in Fig. 6b as $\ell(x,y) c(x,y)$, is compared with $\ell^*(x,y) c^*(x,y)$ at each pixel (x,y) , and the image differences are shown in Fig. 6c. Note the small residual achieved by our method. Fig. 6d shows the ℓ image for the graffiti segment, and Fig. 6e shows the difference $|\ell(x,y) - \ell^*(x,y)|$. The segmentation accuracy, which is defined as

$$1 - \frac{\sum_{(x,y)} |\ell(x,y) - \ell^*(x,y)|}{\sum_{(x,y)} \ell^*(x,y)},$$

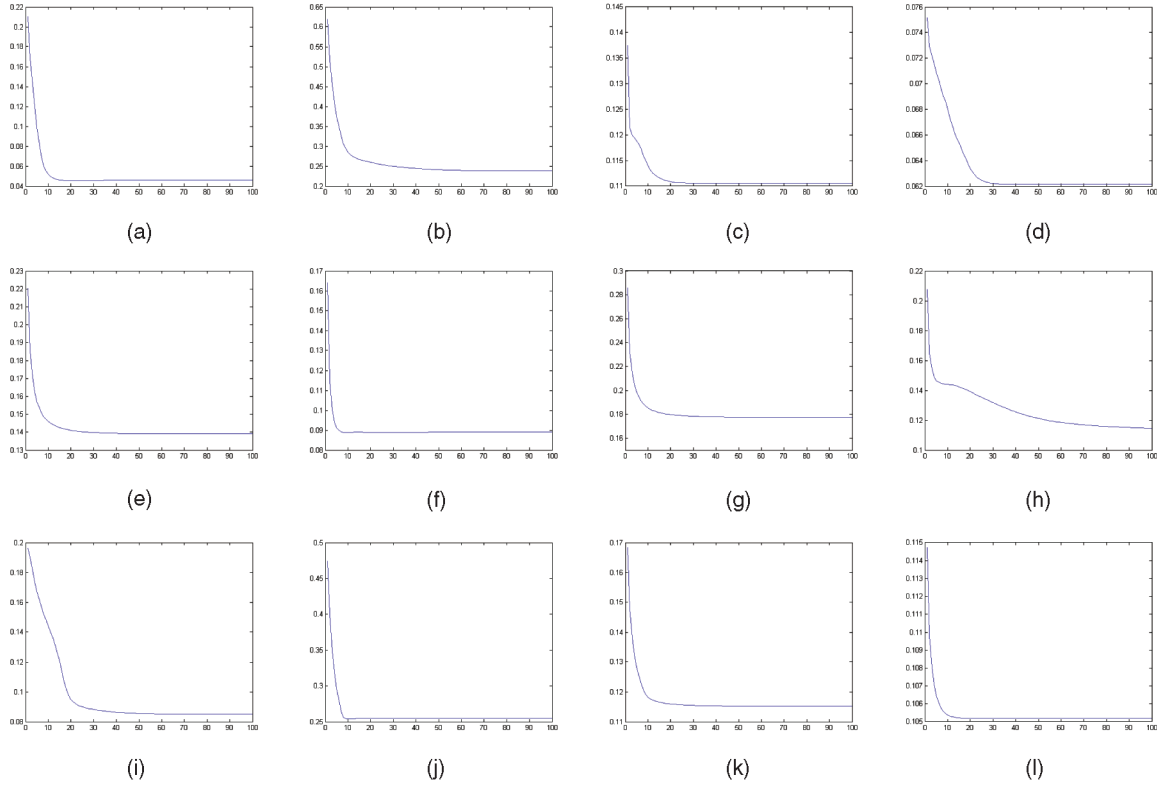


Fig. 3. We plot the negative logarithm of the global objective function (2) against the number of iterations. (a) *Synthetic image* (Fig. 5), 28 seconds. (b) *Graffiti* (Fig. 6), 114 seconds. (c) *Lighthouse* (Fig. 8), 75 seconds. (d) *Camellia* (Fig. 9), 39 seconds. (e) *River 1* (Fig. 12), 94 seconds. (f) *Blobworld* (Fig. 11), 16 seconds. (g) *Motion blur*, 106 seconds. (h) *Hurricane* (Fig. 18), 183 seconds. (i) *Nebula* (Fig. 19), 178 seconds. (j) *River 2* (Fig. 23), 86 seconds. (k) *Castle* (Fig. 24), 46 seconds. (l) *Gray-level image* (Fig. 27), 34 seconds. Our goal is to maximize (2), which is equal to minimizing the negative logarithm of (2). The values shown on the vertical axis are normalized by the number of pixels and the number of clusters in the converged \mathcal{G} . Empirically, our approach converges to a good minima within 30 to 40 iterations. Our AO approach typically runs in 180 seconds for images with resolution 256×256 on a notebook computer with a 1.40 GHz Intel Pentium M processor and 1 Gbyte RAM.

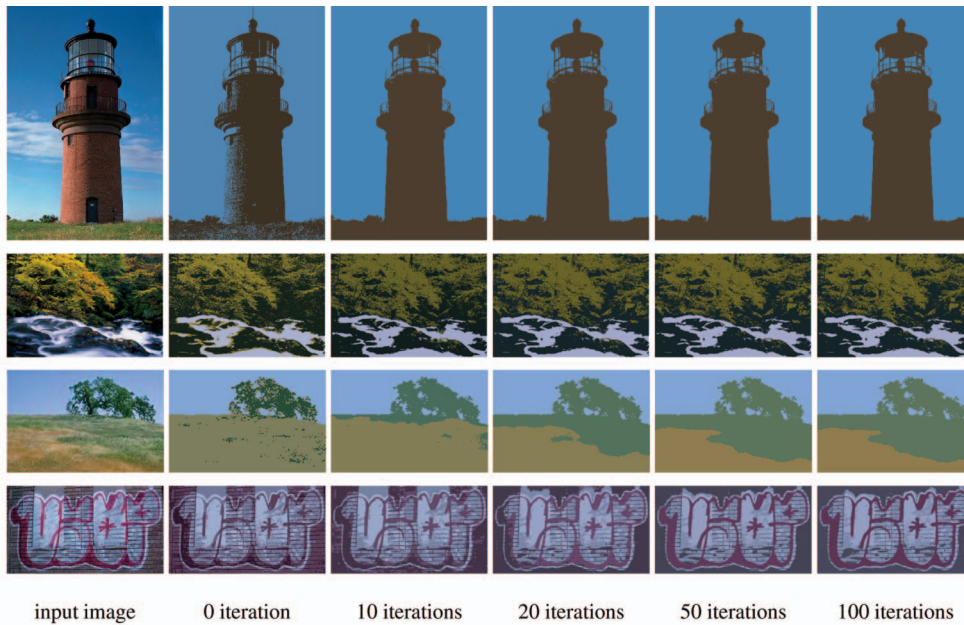


Fig. 4. We show the intermediate results of the AO algorithm. Since each pixel may belong to more than one Gaussian, we show at each pixel the Gaussian component i with the maximum ℓ_i . Initially, no spatial connectivity is considered by k -means clustering, which performs clustering in the RGB space. Spatial and color coherences are progressively preserved as the number of iterations increases.

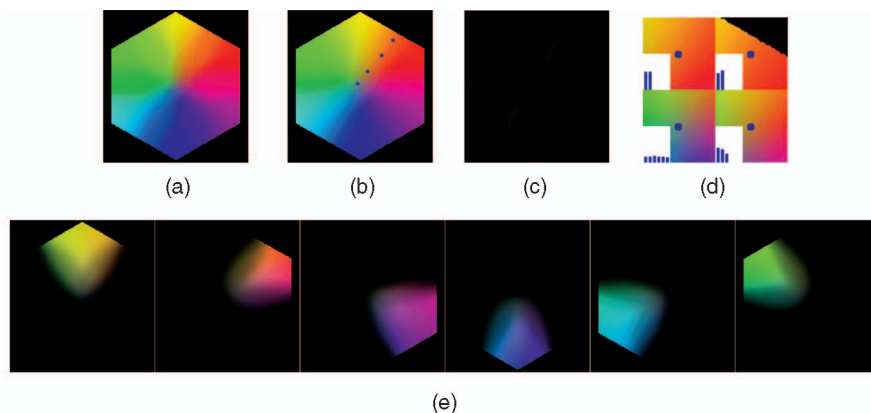


Fig. 5. Evaluation using a synthetic image. (a) Here, the observed color of a pixel may be explained by a mixture of as many as six colors. (b) The resynthesized image generated by composing \mathcal{L} and \mathcal{C} obtained upon the convergence of our AO algorithm. (c) Image difference between (a) and (b). (d) The soft labels \mathcal{L} of sample pixels. (e) The soft segments, which are displayed as $I_i(x, y) = \ell_i(x, y)c_i(x, y)$, $1 \leq i \leq 6$, depict transparent and overlapping boundaries.

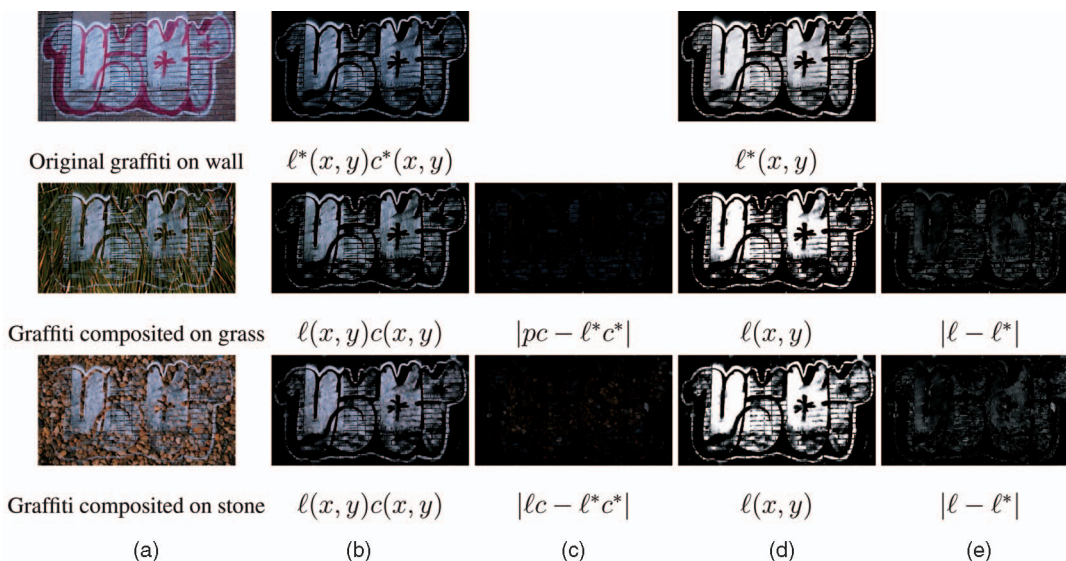


Fig. 6. Evaluation using real image. (a) Original image and image composites. (b) The segmented graffiti. (c) Image difference between the ground truth segment $\ell^*(x, y)c^*(x, y)$ and the graffiti segmented from the new image composite. (d) The $\ell(x, y)$ image for the segmented graffiti. (e) Image difference between the ground truth ℓ^* and the converged ℓ .

is over 70 percent in all the cases, whereas the segmented soft regions are visually indistinguishable from the ground truth.

4.3 Effect of Color Reestimation

Next, we show the necessity of reestimating \mathcal{C} in our AO algorithm. The color estimation step corresponds to the second subproblem in our AO approach, which improves the results at the overlapping regions. Fig. 7b shows one example where we do not perform color reestimation, and we simply set all components of \mathcal{C} to I . Compared to Fig. 7a, which was obtained using our proposed method, the segmented Gaussian components in Fig. 7b are less separable. A real example is shown in Figs. 8a and 8b.

4.4 Effect of GMM Reestimation

Finally, we show the necessity of reestimating \mathcal{G} in our AO algorithm. \mathcal{G} encodes the global color statistics for soft color segmentation. It is initialized by k -means clustering. Without \mathcal{G} reestimation, our approach is reduced to one that runs

the belief propagation on the result produced by k -means clustering, followed by the color estimation step.

Fig. 8c shows the result without \mathcal{G} reestimation, which is susceptible to errors given by the initial clustering, therefore resulting in the suboptimal estimation of \mathcal{L} and \mathcal{C} , as depicted in the figure.

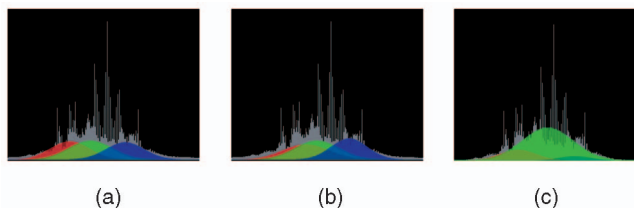


Fig. 7. The three estimated Gaussians overlaid onto the histogram of the graffiti image. Only the R channel is shown here. (a) With color estimation. (b) Without color estimation. (c) The Gaussians estimated using the original EM algorithm. The estimated Gaussian components are better separated using our AO algorithm.

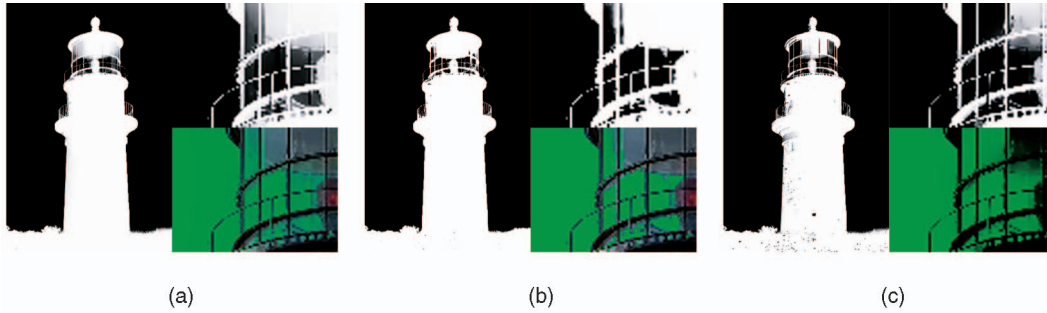


Fig. 8. The images of soft label ℓ for a lighthouse image. The subimage on the lower right side shows the compositing result using the estimated ℓ and c on a green background. (a) Segmentation result by our AO approach. (b) Segmentation result without \mathcal{C} reestimation. Comparing the result in (a), better transparent regions are obtained with color reestimation. (c) Segmentation result without \mathcal{G} reestimation. The input image is shown in Fig. 20.

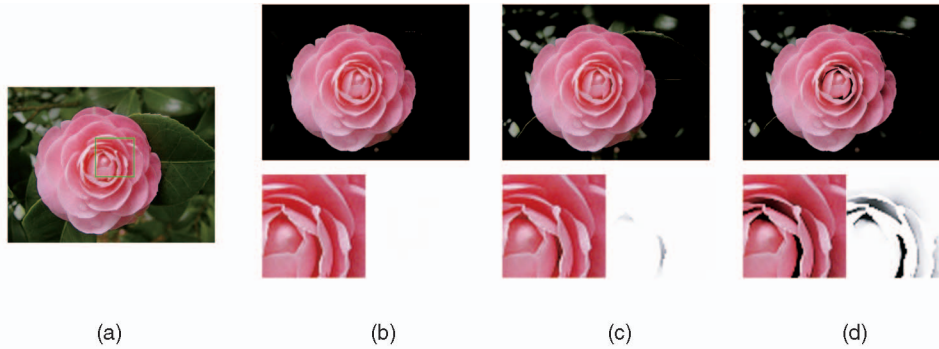


Fig. 9. (a) Input image. Soft shading and shadows are present between the petals. The segmentation result using (b) covariance matrix (oriented Gaussians), (c) standard deviation (unoriented Gaussians), and (d) k -means clustering with belief propagation and color estimation. Shading and soft shadows are better captured by an oriented Gaussian.

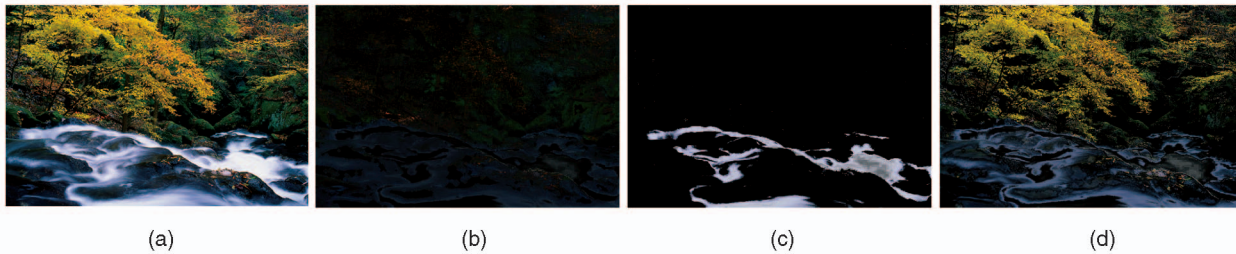


Fig. 10. Segmentation result by the original EM algorithm. Without spatial consideration, spatially dissimilar patterns are mixed up. (a) Input image. (b), (c), and (d) are the three segments output by the EM algorithm.

5 RESULTS AND COMPARISON

We have evaluated our method on real and synthetic images and studied the effects of \mathcal{C} and \mathcal{G} on \mathcal{L} in the AO algorithm. In this section, we present the results and comparisons with state-of-the-art segmentation techniques, focusing on shading and shadows, highly textured scenes, and multiscale processing.

5.1 Shading and Soft Shadows

Because our approach is designed to produce soft color segments, soft shading and shadows can be handled uniformly. We compare our method by using the following alternatives:

- standard deviation in modeling \mathcal{G} and
- k -means clustering with belief propagation and color estimation.

Fig. 9 shows that our AO method produces significantly better results. This can be explained by the use of Σ , which

encodes an oriented Gaussian and captures nonuniform color distribution due to shadow and shading.

5.2 Highly Textured Scenes

We first compare our AO algorithm with EM algorithm, which is one specific form of AO. Then, comparisons are made with other representative segmentation methods.

The **original EM** algorithm segments an image in the color domain. Fractional boundaries can be obtained if we do not use the maximum-vote filter such as the one proposed in [1]. Fig. 10 shows the results. Since no spatial information is considered, spatially different patterns such as the water and the leaves cannot be well separated.

The **revised EM** algorithm described in [1] performs GMM estimation in the 6D feature space in which color/texture segmentation is performed. Fig. 11 shows our results compared with [1]. Because no spatial information is considered during their EM iteration, the maximum-vote filter and the connected component algorithm are used to

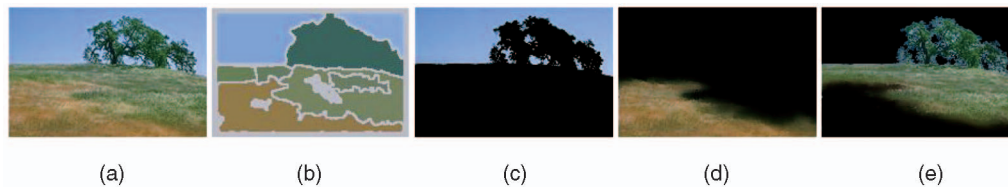


Fig. 11. (a) Input image [1]. (b) Segmentation result from [1]. (c), (d), and (e) Our soft color segmentation results $\ell_i c_i$.

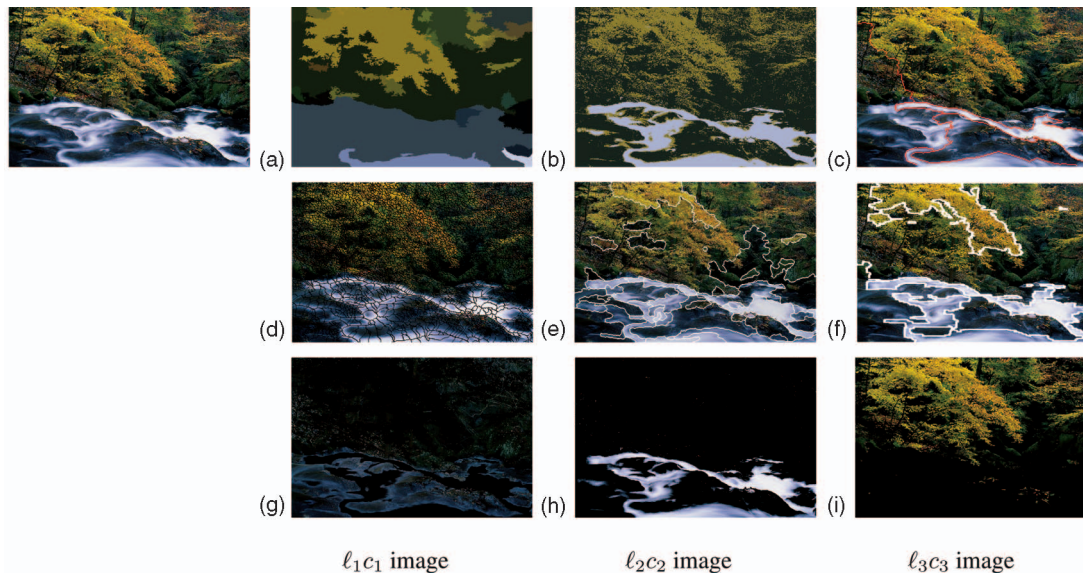


Fig. 12. Input image shown on the left. Results produced by (a) Mean Shift segmentation, (b) k -means clustering with $k = 3$, (c) Normalized Cuts, (d) Watershed algorithm, (e) JSeg method, (f) statistical region merging, (g), (h), and (i) our AO algorithm. The three segmented regions correspond to the three basic color components underlying the image.

enforce spatial connectivity. As shown in Fig. 11, although their approach can group relevant region centers, the undecided region boundaries are output as unsegmented regions. Our automatic approach produces fractional boundaries to faithfully maintain the smooth color transition among segments.

For the complex scene shown in Fig. 12, **Mean Shift** [6] cannot segment the river well. The complex region boundaries cannot be preserved, as shown in Fig. 12a. Because Mean Shift performs segmentation by first concatenating color and spatial coordinates, the resulting sparse and high-dimensional feature space makes the segmentation more challenging.

K-means segmentation [10] segments the image by clustering in the RGB space ($k = 3$ in our example). The result is good from the global statistical point of view, but since spatial information is not considered, isolated point clusters are resulted (Fig. 12b).

The image segmentation problem is formulated into **Normalized Cuts** in [28], where a graph partitioning problem is solved. For this example, the complex region boundaries are not well preserved (Fig. 12c).

The **Watershed** algorithm [37] segments the image into many small partitions (Fig. 12d), which are suitable for applications requiring an oversegmentation of the scene.

JSeg [9] is an image-based segmentation technique, which consists of regions splitting, growing, and merging. The segmentation results are subject to a user-defined color quantization threshold and a region-merging threshold,

whereas our automatic alternating maximization algorithm does not have critical threshold to set. The segmentation results produced by JSeg are not satisfactory for this complex scene as shown in Fig. 12e.

The implicit assumption used in the **statistical region merging** [19] is that the observed color variations inside the same region should be smaller than those across different regions. The use of local statistics and a single scale are not adequate in handling a complex image with multiscale features. Overmerging and undermerging are therefore possible, as shown in Fig. 12f.

In our **AO**, the global color statistics, as well as the local spatial coherence, are considered. In Figs. 12g, 12h, and 12i, the three basic colors of the image are separated. Specifically, Fig. 12g shows the dark gray transparent segment, which provides a smooth color transition between the leaves and the river. The interweaving green and yellow leaves are reasonably segmented from the complex scene.

Although our soft color segmentation does not necessarily generate a semantic segmentation of the scene, it does produce good synthesis results, which are often used to evaluate the segmentation quality. For instance, in the **DDMCMC** method [35], the input is resynthesized using $p(I|W)$, where W is the segmentation result. In our case, we resynthesize the input using $p(I|\mathcal{L}, \mathcal{C})$. Some results are shown in Fig. 13. Notice that, however, the goal of DDMCMC method is to produce coherent segmentation for image understanding, whereas our method optimizes

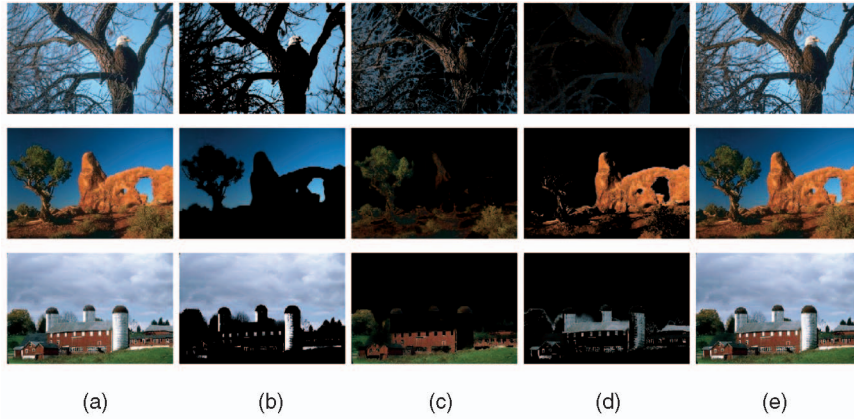


Fig. 13. Evaluation by resynthesis. (a) Input images. (b), (c), and (d) Our segmentation results displayed as $\ell_i(x, y)c_i(x, y)$, $1 \leq i \leq 3$. (e) Resynthesized results of our approach using $p(I|\mathcal{L}, \mathcal{C})$.

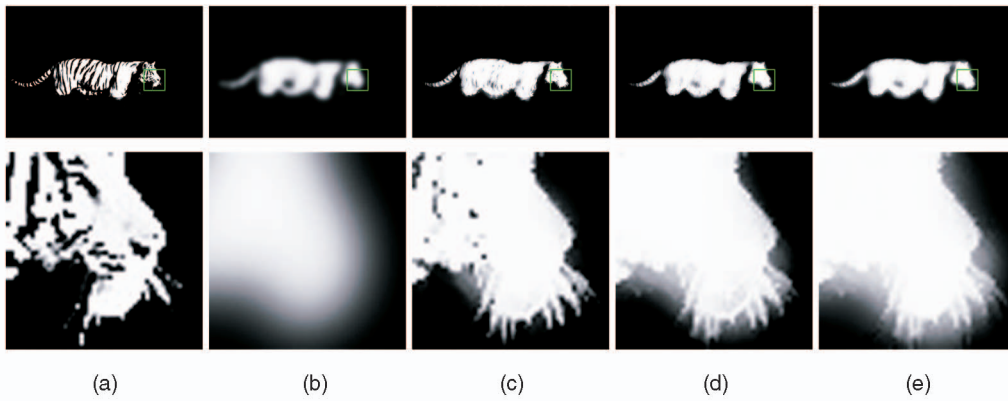


Fig. 14. (a) Image of $\ell_1(x, y)$ estimated using (10) without Gaussian pyramid construction. (b), (c), (d), and (e) The results using a five-level Gaussian pyramid. (b) $\ell_1(x, y)$ estimated in the highest level. Taking the segmentation result obtained in the immediately higher level as a soft constraint and applying (26), $\ell_1(x, y)$ is estimated in the original input scale with (c) $\sigma_{ms} = 1$, (d) $\sigma_{ms} = 0.5$, and (e) $\sigma_{ms} = 0.01$. By using smaller σ_{ms} , the output soft labels are more uniform in the textured region.

for overlapping and transparent segments to preserve natural and smooth color transition.

5.3 Multiscale Processing

Our soft color segmentation can be used to process images at multiple scales. Since we use a GMM to represent colors and a Gaussian kernel to perform prefiltering before subsampling, the scale-space theory [39] asserts that no new edge features will be produced while processing the subsampled data after Gaussian prefiltering.

We propose to use the Gaussian pyramid to reduce dissimilarity of ℓ_i among pixels in a single textured region. The result obtained in a higher level is incorporated as a soft constraint while optimizing for the result in the immediately lower level. By enforcing the same N (obtained in the lowest level) to all levels, we introduce the following prior term to replace (10) as follows:

$$P(\mathcal{L}) \propto \prod_{(x,y)} \exp\left(-\left(\frac{\mathcal{L}(x,y) - \mathcal{L}_{ms}(x,y)}{\sigma_{ms}}\right)^2\right) \prod_{(x,y)} \prod_{(x',y') \in \mathcal{N}(x,y)} \exp(-\psi(\mathcal{L}(x,y), \mathcal{L}(x',y'))), \quad (26)$$

where $\mathcal{L}_{ms}(x, y)$ is the result obtained in the immediate higher level in the pyramid, and σ_{ms} controls the similarity between $\mathcal{L}(x, y)$ and $\mathcal{L}_{ms}(x, y)$. Equation (26) indicates that a small

value of σ_{ms} gives more penalty, which, in turn, favors that $\mathcal{L}(x, y)$ and $\mathcal{L}_{ms}(x, y)$ be similar in distribution. Fig. 14 shows the effect of σ_{ms} . Note that, in image-based applications, partitioning one texture pattern into exactly one segmented region may not be always desired, so the choice of (10) or (26) in modeling the prior $P(\mathcal{L})$ depends on applications.

When (26) is used, we set $\sigma_{ms} = 0.5$. Fig. 15 shows the comparison. The top row compares the result on one example with that presented by Galun et al. [12]. Our result can successfully segment the two patterns on the wall. In the bottom row of Fig. 15, we compare our result with DDMCMC [35] in texture segmentation.

Using color transfer [24], we can recolor the segments at different scales. In Fig. 16b, we show $\ell_1(x, y)$ images produced by using (26). The top row of Fig. 16 compares the result on one example with that presented by Galun et al. [12], where the camouflage patterns are progressively segmented. In the bottom row, we compare our result using an input image from Sharon et al. [27]. Our method produces good results where the water is separated from the tiger while the stripped textures are maintained. We recolor the resulting segments using different colors (using small scale of analysis) or recolor them using the same colors (using large scale of analysis).

In Fig. 17, we show one instance that our multiple scale segments produced are consistent among each other,

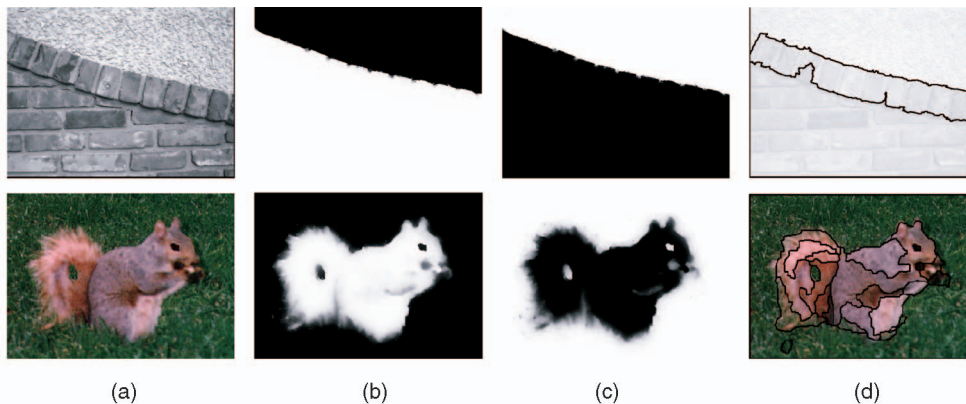


Fig. 15. We compare our multiscale results with those by Galun et al. [12] (top) and DDMCMC [35] (bottom). Using the multiscale prior to (26), our method automatically converges to two soft segments, that is, ℓ_1 and ℓ_2 . (a) Input images. (b) ℓ_1 images. (c) ℓ_2 images. Our multiscale method deals with textures and produces soft segments with appropriate boundary transparency and spatial coherence. The results in [12] and [35] are shown in (d). Note that our method works in RGB and does not work better in intensity images with one gray-scale channel.

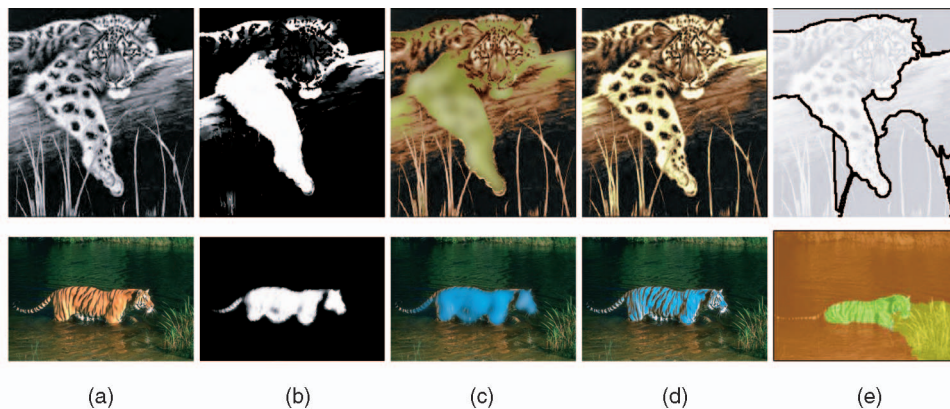


Fig. 16. To show that our approach is capable of producing scene segmentation at multiple scales, we process (by recoloring) the selected segment at different scale. These two examples are respectively presented by Galun et al. [12] (top) and Sharon et al. [27] (bottom). The selected segments are the leg of the leopard and the torso of the tiger, respectively. (a) Input image. (b) $\ell_1(x, y)$ estimated with scale prior to (26). Recolored results using (c) large scale and (d) small scale. The segmentation results in [12] and [27] are shown in (e).

allowing a user to specify and focus on regions of interest. For example, using the multiscale soft segments, we can recolor all the dining facilities along the waterfront using a single color (Fig. 17b) or recolor the individual tables and chairs (Fig. 17c) while the faraway mountain and windmills are recolored using a small scale in both cases.

6 APPLICATIONS

In this section, we present the results and applications of soft color segmentation. Our approach provides a general framework for the following applications, which were

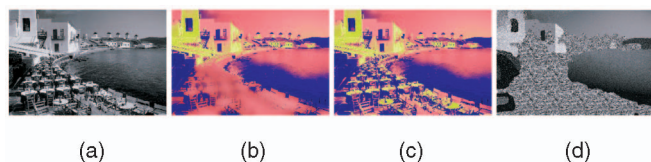


Fig. 17. Consistency of multiple scale segments. (a) Input gray-scale image. To show that our approach is capable of producing soft segmentation at multiple scales, we recolor the image using (b) large scale and (c) small scale at the chosen regions or scales of interest (the dining facilities along the waterfront). The result in [35] is shown in (d).

traditionally addressed by separate algorithms. Without any human interaction, the results produced by our method are more reasonable or comparable to previous methods, where user assistance may be required in the latter case.

6.1 Soft Color Segmentation

Hurricane images. Soft color segmentation can be applied to process satellite images on hurricane, where the specification of a tri-map for image matting is difficult in complex images such as the input shown in Fig. 18a. Hurricane segmentation from satellite images is helpful in identifying, analyzing, and predicting the formation of hurricanes that are nonsolid and partially transparent. Figs. 18b, 18c, and 18d show the results produced by our soft color segmentation. Note that the inferred ℓ values for the hurricane are proportional to the cloud density. Fig. 18e shows an unsatisfactory segmentation result produced by hard segmentation.

Nebulas. In nebula analysis, different colors of a nebula represent different components and temperatures of the nebula. Fig. 19a shows a nebula image Messier Object M20 captured by the Hubble Space Telescope. The red emission nebula with its young stars clustered near its center is surrounded by a blue reflection nebula, where different components and temperatures exist across the nebula. Similar to hurricanes, the nebula is not a solid object and

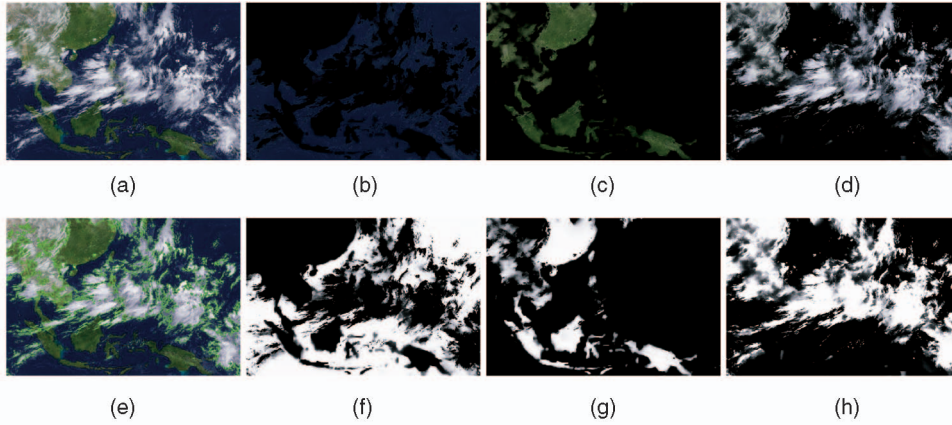


Fig. 18. Segmentation of a satellite image of a hurricane. (a) Input image. (b), (c), and (d) Our segmentation results displayed as $\ell_i(x, y)c_i(x, y)$, $1 \leq i \leq 3$. (e) Hard segmentation result. (f), (g), and (h) The corresponding ℓ_i images for (b), (c), and (d). Our approach segments the land from the hurricane. The ℓ image in (h) indicates the cloud density of the hurricane.

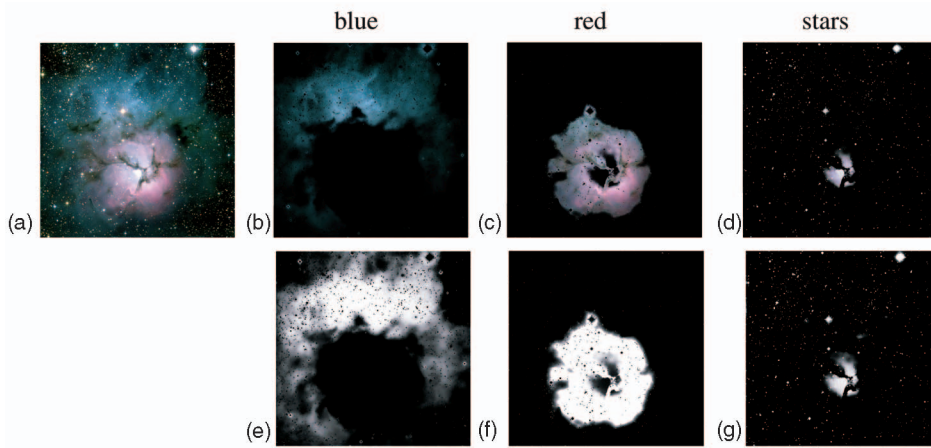


Fig. 19. Segmentation of a nebula image. (a) Input image, Messier Object M20. (e), (f), and (g) The soft labels $\ell_i(x, y)$ corresponding to (b), (c), and (d). Our algorithm segments the red and blue nebulas with fractional boundaries. Note that the bright stars are not smoothed out due to the discontinuity-preserving property of our MRF formulation.

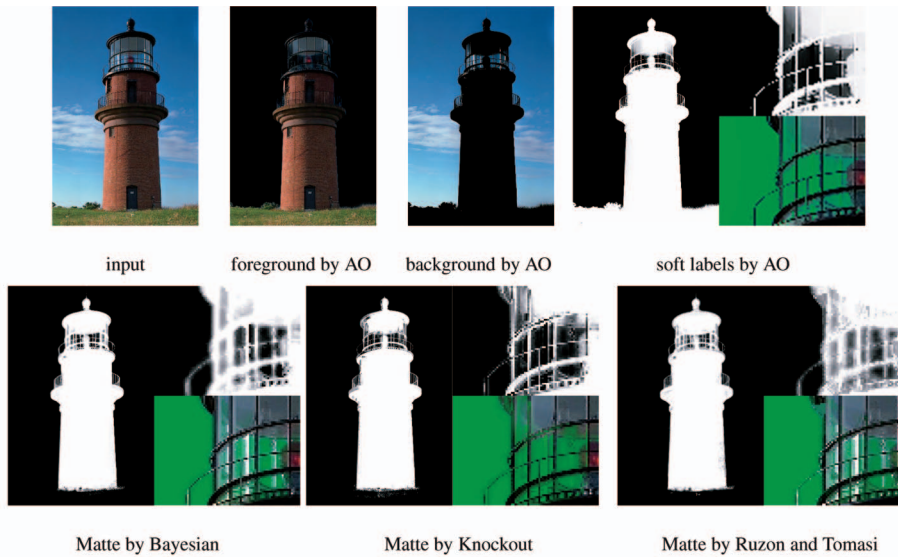


Fig. 20. Comparison with image matting. Although approaches in natural image matting use a user-specified trimap or other user-supplied hints, our method is fully automatic.

the transition boundary between the red and blue nebulas should be soft and smooth. Results obtained by hard segmentation cannot faithfully reflect this phenomenon.

Figs. 19b, 19c, and 19d show our soft color segmentation results in which a smooth and natural transition across the nebulas is achieved.

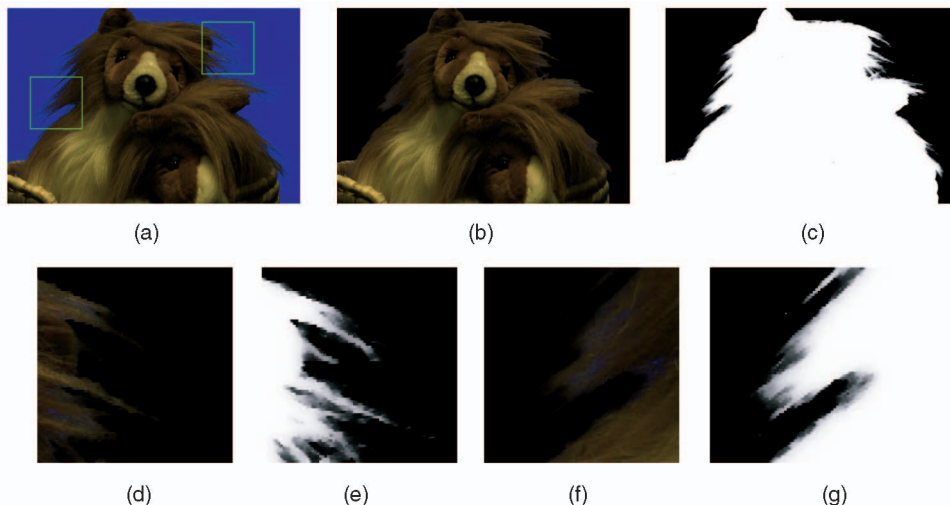


Fig. 21. Boundary smoothness and transparency for an object with long hairs. (a) Input image. (b) One automatically segmented region. (c) ℓ image of the region (or alpha matte). (d), (e), (f), and (g) Zoom-in views of the results obtained using our method. Because no trimap is used, long and thin hairs are missed, whereas short hairs are still preserved.

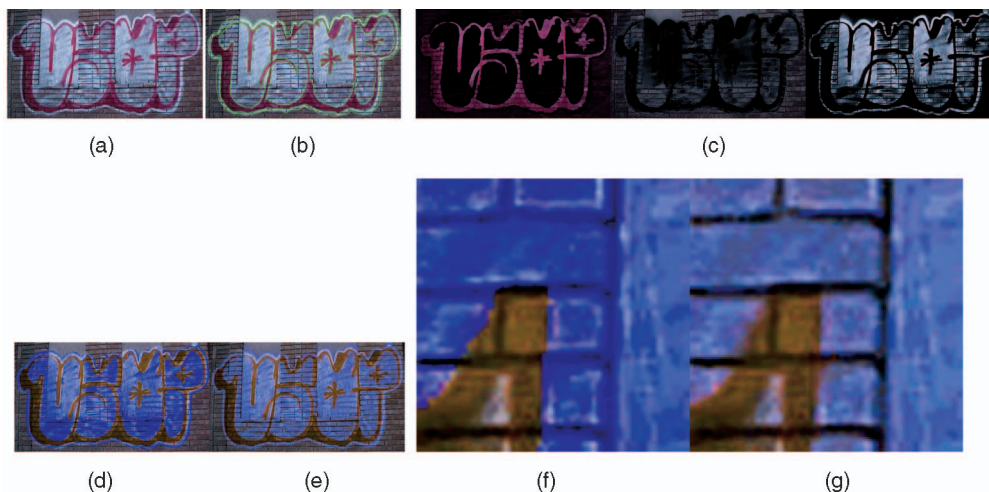


Fig. 22. (a) The graffiti image. (b) Hard segmentation manually produced, where the region boundaries are indicated by the solid green curves. (c) Soft segmentation produced by our AO algorithm. The three converged color regions are shown. (d) Color transfer result using the hard segmentation shown in (b). (e) Color transfer result using the soft segmentation shown in (c). Zoom-in views of the result for (f) hard segmentation and (g) soft segmentation. The color transfer result in (g) is smoother and more natural. The color transfer equation in [24] is used to generate (d), (e), (f), and (g).

6.2 Image Matting

Image matting, in particular, Bayesian matting [5], can be regarded as a user-assisted form of soft color segmentation for the specific case $N = 2$, if $\mathcal{L} = \{\alpha, 1 - \alpha\}$, where $0 \leq \alpha \leq 1$, then $\mathcal{C} = \{F, B\}$, where F and B are the respective optimal foreground and background colors, when \mathcal{G} is restricted in a local neighborhood governed by a user-supplied trimap.

Fig. 20 shows a result on natural image matting. Our result is comparable to the results obtained by Bayesian matting [5], although our automatic approach does not use any trimap. As depicted in our result, the fences of the lighthouse are not smoothed out. Fig. 21 shows an example in which long hairs are present. Our segmentation result is satisfactory for the example except for the long hairs because no trimap is used.

6.3 Color Transfer between Images

A single Gaussian is used in [24] to model the global color statistics of the source and target images. If diversified colors

are present in any of them, image patches must be manually specified to divide the colors into separate clusters. However, for complex images, it is difficult for a user to specify the right patches, and a small number of patches are inadequate to discern different color statistics. Here, we propose to perform soft color segmentation on both the source and target images before applying color transfer so that the transfer process is fully automatic. Recall that our soft color segmentation models the global color distribution by a GMM, that is, each color region (not necessarily connected) corresponds to a component of the converged GMM. We therefore define the final transferred color for a pixel $g(I_T(x, y))$ by

$$g(I_t(x, y)) = \sum_j \ell_{T_j}(x, y) \left(\frac{\sigma_{S_i}}{\sigma_{T_j}} (I_t(x, y) - \mu_{T_j}) + \mu_{S_i} \right), \quad (27)$$

where $\ell_{T_j}(x, y)$ is the soft label corresponding to the T_j th Gaussian component of the GMM of the target image, obtained by our AO algorithm for soft color segmentation.

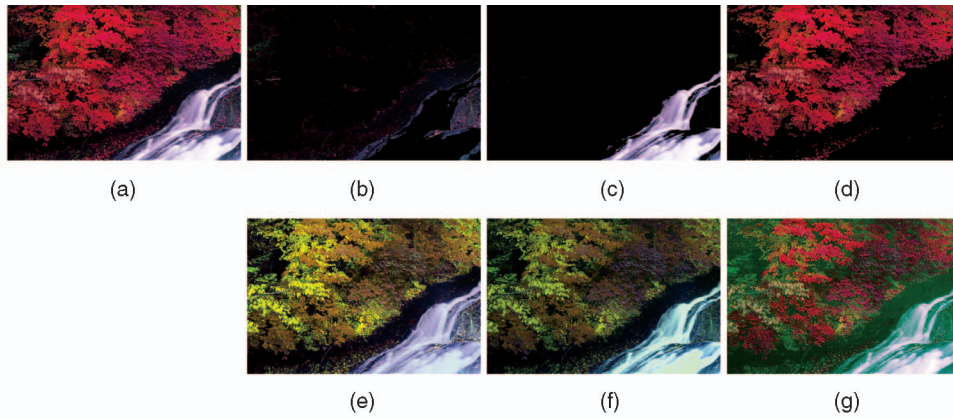


Fig. 23. Comparison of color transfer using our approach and that in [24]. (a) The target image (the source image is shown in Fig. 12). (b), (c), and (d) Soft color segmentation results. (e) The color-transfer result using our approach in which soft color segmentation is performed before transferring the colors. (f) The color-transfer result without soft color segmentation [24]. (g) The color-transfer result by histogram equalization. The results in (f) and (g) show undesirable mixture of colors of the leaves and the river.

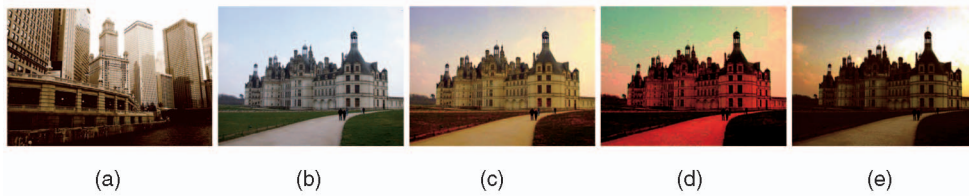


Fig. 24. Comparison of color transfer on a natural scene using our approach and that in [24]. (a) An old photograph of a downtown scene captured on an overcast day. (b) The target image captured on a sunny day. (c) Our transfer result. (d) Transfer result generated in [24] where an unacceptable mixture of colors is present. (e) Transfer result generated by histogram equalization in which undesirable saturation is observed.

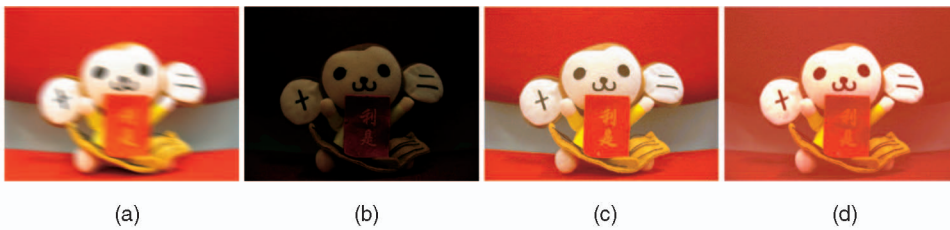


Fig. 25. Image deblurring using color transfer with/without soft color segmentation. (a) Source image. (b) Target image. (c) Result using our approach. (d) Result using a single Gaussian model. Note that the colors in result (d) are not preserved.

The following examples show that our GMM model is more suitable than a single Gaussian used in [24] in guiding the color transfer process. Our natural color transfer with soft color segmentation achieves smoother and more natural color transition, especially for regions rich in colors and textures.

Fig. 22 first demonstrates color transfer on a complex scene based on soft color segmentation, which is more preferable in comparison to the one based on hard/binary segmentation. The transferred result using soft color regions (Fig. 22g) looks smoother and more natural, as opposed to that given by hard segmentation where unnatural and abrupt changes in color are observed among adjacent color patches (Fig. 22f).

Figs. 23 and 24 compare the color transfer results using a GMM model obtained by our AO algorithm as opposed to using a single Gaussian model [24] for guiding the color transfer process. In Fig. 23, the source image and its soft segments were already shown in Figs. 12g, 12h, and 12i. The soft segments for the target segments are shown in Figs. 23b, 23c, and 23d. Our transfer result is shown in Fig. 23e. Using the GMM model and soft color segments to guide the color transfer process, the color of the rivulet (Fig. 12) is faithfully transferred to the river (Fig. 23). In comparison with that in

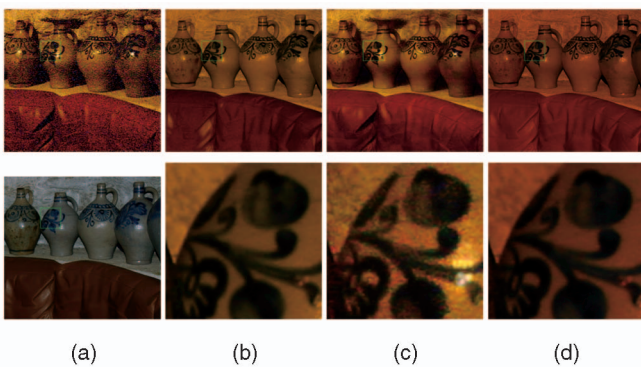


Fig. 26. Comparison on image denoising. (a) Nonflashed source and flashed target images. Results and zoom-in views obtained by (b) our approach, (c) by that in [21], and (d) by that in [11]. Our method makes use of the strong spatial relationship between the source and target images given by the soft color segmentation. Therefore, the red shade of the sofa, the bottles, and the stones are not mixed up. Such an undesirable mixture is observed in the result in (d). The result using joint bidirectional filter [21] in (c) is still very noisy.

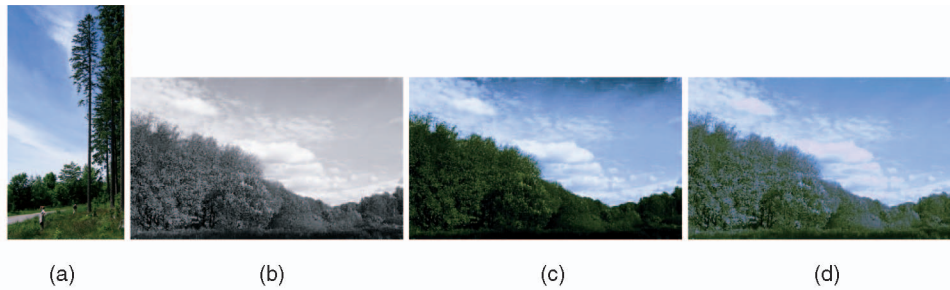


Fig. 27. Color transfer to a gray-scale image. (a) The source image. (b) The target image. (c) The result by our approach. (d) The result by that in [38].

[24], using a single Gaussian that does not adequately model the global color distribution of the image, as depicted in Figs. 23f and 23g, the transfer results are unsatisfactory as undesirable mixture of colors of the leaves and the river is easily observed.

Figs. 24a and 24b show another source/image pair. Fig. 24c is the transfer result generated using our approach. Comparing with the respective results generated by Reinhard et al. [24] in Fig. 24d and histogram equalization in Fig. 24e, our result is more natural and suffers less saturation.

6.4 Image Correction Using Image Pairs

We have extended the idea of color transfer using image pairs to correct image intensities.

Image deblurring using a normal/low exposure pair. Given two images of the same scene taken almost simultaneously and without a tripod 1) one is acquired under normal exposure and, so, motion and shape blur may be present and 2) the other is taken with a short shutter speed and, so, the image is crisp but underexposed, we want to transfer the color from image 1 to image 2 so as to generate a bright and crisp image. Since the source and the target images have strong spatial similarity, after segmenting the respective images using our AO approach, we can use, as the criterion, the largest overlapping area of the segmented regions for matching the Gaussians, so as to perform the associated color transfer from the source to target to deblur the image. Fig. 25 shows our result of image deblurring.

Image denoising using flash/no flash pair. Similarly, we can perform image denoising using two images: one is taken with camera flash (flashed image) and the other is captured using a high ISO configuration without flashing (nonflashed image) to preserve the original scene color and ambiance. Images captured with a high ISO setting contain a considerable amount of noise. In [21], the flashed image is used to denoise the nonflashed image. In our method, we first use a median filter to reduce the amount of noise. Then, both the source and target images are segmented using our AO algorithm. Finally, we map the colors from the nonflashed image to the flashed image to construct our denoised and sharp image, which faithfully preserves the original scene ambiance. Fig. 26 shows and compares the denoised result. Our current result does not transfer shadows though, which, on the other hand, can be performed as described in [11].

6.5 Colorization

For applications in image colorization, we only have the luminance channel in an input gray-scale image. To constrain the color transfer, we assume that two pixels in the same region have similar colors if they have similar luminance value. Similar to the above transfer applications, we perform

soft color segmentation in both the source and target images. Here, the only modification of our method is that we perform AO only on the luminance ℓ channel and assign the same distribution to the absent ab channels. Fig. 27 shows one result. Note in our result the smooth and natural transition between blue sky and green trees. The whole process is fully automatic.

7 CONCLUSION

We have described an algorithm based on AO to address the problem of soft color segmentation. We have demonstrated that our method produces reasonable segmentation in the form of overlapping and transparent color regions, despite the presence of rich colors, textures, shading, soft shadows, and image noises. Our soft color segmentation can be applied at multiple scales. This general approach has found useful applications such as image matting, color transfer, image correction, and image colorization. Our method combines the advantages of global color statistics and local image compositing while preserving the necessary spatial and color coherence. In the future, we shall investigate the role of soft color segmentation in fundamental computer vision:

- stereo, where traditional techniques assume opacity (except in [32]), and
- motion, where motion blur is problematic to many algorithms.

ACKNOWLEDGMENTS

The authors thank the associate editor and all reviewers for their thorough and constructive suggestions, which were instrumental in improving the quality of the final paper. This research was supported by the Research Grant Council of the Hong Kong Special Administrative Region under Grant 412206 and 620005.

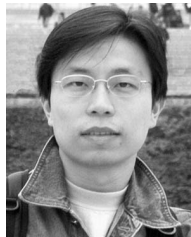
REFERENCES

- [1] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color- and Texture-Based Image Segmentation Using the Expectation-Maximization Algorithm and Its Application to Content-Based Image Retrieval," *Proc. Int'l Conf. Computer Vision*, pp. 675-682, 1998.
- [2] A. Berman, P. Vlahos, and A. Dadourian, "Comprehensive Method for Removing from an Image the Background Surrounding a Selected Object," US Patent 6,134,345, 2000.
- [3] J.C. Bezdek and R.J. Hathaway, "Convergence of Alternating Optimization," *Neural, Parallel Scientific Computing*, vol. 11, no. 4, pp. 351-368, 2003.
- [4] M.J. Black and A. Rangarajan, "The Outlier Process: Unifying Line Processes and Robust Statistics," *Proc. Computer Vision and Pattern Recognition*, pp. 15-22, 1994.

- [5] Y.-Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski, "A Bayesian Approach to Digital Matting," *Proc. Computer Vision and Pattern Recognition*, vol. II, pp. 264-271, 2001.
- [6] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.
- [7] I. Csiszar and G. Tusnady, "Information Geometry and Alternating Minimization Procedures," *Statistics and Decisions, Supplement Issue*, pp. 205-237, 1984.
- [8] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximal Likelihood from Incomplete Data via the EM Algorithm," *Royal Statistics*, vol. B, no. 39, pp. 1-38, 1977.
- [9] Y. Deng and B.S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800-810, Aug. 2001.
- [10] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.
- [11] E. Eisemann and F. Durand, "Flash Photography Enhancement via Intrinsic Relighting," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 673-678, 2004.
- [12] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture Segmentation by Multiscale Aggregation of Filter Responses and Shape Elements," *Proc. Int'l Conf. Computer Vision*, pp. 716-723, 2003.
- [13] S. Gordon, H. Greenspan, and J. Goldberger, "Applying the Information Bottleneck Principle to Unsupervised Clustering of Discrete and Continuous Image Representations," *Proc. Int'l Conf. Computer Vision*, pp. 370-377, 2003.
- [14] R.M. Gray, *Entropy and Information Theory*. Springer, Nov. 2000.
- [15] P. Grunwarld, *Minimum Description Length and Maximum Probability*. Kluwer Academic Publishers, 2002.
- [16] J. Jia, J. Sun, C.K. Tang, and H.Y. Shum, "Bayesian Correction of Image Intensity with Spatial Consideration," *Proc. European Conf. Computer Vision*, vol. III, pp. 342-354, 2004.
- [17] V. Kolmogorov and R. Zabih, "Multi-Camera Scene Reconstruction via Graph Cuts," *Proc. European Conf. Computer Vision*, p. III: 82 ff., 2002.
- [18] M. Mirmehdi and M. Petrou, "Segmentation of Color Textures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 142-159, Feb. 2000.
- [19] R. Nock and F. Nielsen, "Statistical Region Merging," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452-1458, Nov. 2004.
- [20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [21] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital Photography with Flash and No-Flash Image Pairs," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 664-672, 2004.
- [22] M. Pietikainen, A. Rosenfeld, and I. Walter, "Split-and-Link Algorithms for Image Segmentation," *Pattern Recognition*, vol. 15, no. 4, pp. 287-298, 1982.
- [23] T. Porter and T. Duff, "Compositing Digital Images," *Proc. ACM Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '84)*, pp. 253-259, 1984.
- [24] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color Transfer between Images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34-41, 2001.
- [25] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut—Interactive Foreground Extraction Using Iterated Graph Cuts," *Proc. ACM Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, 2004.
- [26] M.A. Ruzon and C. Tomasi, "Alpha Estimation in Natural Images," *Proc. Computer Vision and Pattern Recognition*, pp. 18-25, 2000.
- [27] E. Sharon, A. Brandt, and R. Basri, "Segmentation and Boundary Detection Using Multiscale Intensity Measurements," *Proc. Computer Vision and Pattern Recognition*, pp. I: 469-476, 2001.
- [28] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.
- [29] A.R. Smith and J.F. Blinn, "Blue Screen Matting," *Proc. ACM Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 259-268, 1996.
- [30] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson Matting," *Proc. ACM Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '04)*, pp. 315-321, 2004.
- [31] J. Sun, N.N. Zheng, and H.Y. Shum, "Stereo Matching Using Belief Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787-800, July 2003.
- [32] R. Szeliski and P. Golland, "Stereo Matching with Transparency and Matting," *Int'l J. Computer Vision*, vol. 32, no. 1, pp. 45-61, Aug. 1999.
- [33] Y.W. Tai, J. Jia, and C.K. Tang, "Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization," *Proc. Computer Vision and Pattern Recognition*, pp. I: 747-754, 2005.
- [34] K.L. Tang, C.K. Tang, and T.T. Wong, "Dense Photometric Stereo Using Tensorial Belief Propagation," *Proc. Computer Vision and Pattern Recognition*, vol. 1, pp. 132-139, June 2005.
- [35] Z. Tu and S.C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, May 2002.
- [36] J.K. Udupa, P.K. Saha, and R.A. Lotufo, "Relative Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1485-1500, Nov. 2002.
- [37] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583-598, June 1991.
- [38] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring Color to Greyscale Images," *Proc. Int'l Conf. Computer Graphics and Interactive Techniques (SIGGRAPH '02)*, pp. 277-280, 2002.
- [39] A.P. Witkin, "Scale-Space Filtering," *Proc. Int'l Joint Conf. Artificial Intelligence*, pp. 1019-1022, 1983.
- [40] T.P. Wu, K.L. Tang, C.K. Tang, and T.T. Wong, "Dense Photometric Stereo: A Markov Random Fields Approach," *IEEE Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1830-1846, Nov. 2006.



Yu-Wing Tai received the BEng and MPhil degrees in computer science from the Department of Computer Science at the Hong Kong University of Science and Technology. He is pursuing the PhD degree at Nanyang Technological University. His research interests include and segmentation and texture analysis. He is a student member of the IEEE and the IEEE Computer Society.



Jiaya Jia received the PhD degree from the Department of Computer Science at the Hong Kong University of Science and Technology. He joined the Department of Computer Science and Engineering at the Chinese University of Hong Kong in 2004. His research interests include computer vision and image/video processing. He is a member of the IEEE Computer Society.



Chi-Keung Tang received the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been with the Department of Computer Science at the Hong Kong University of Science and Technology (HKUST) where he is currently an associate professor. He is an adjunct researcher in the Visual Computing Group of Microsoft Research Asia. His research interests include computer vision and computer graphics. He was on the program committees of the International Conference on Computer Vision (ICCV), Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV), and the International Conference on Pattern Recognition (ICPR) and served as an area chair for Asian Conference on Computer Vision (ACCV) 2006 (Hyderabad) and ICCV 2007 (Rio de Janeiro). He is a senior member of the IEEE Computer Society.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.