

# Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization\*

Yu-Wing Tai<sup>†</sup>    Jiaya Jia\*    Chi-Keung Tang<sup>†</sup>

<sup>†</sup>The Hong Kong University of Science and Technology

\*The Chinese University of Hong Kong

## Abstract

We address the problem of regional color transfer between two natural images by probabilistic segmentation. We use a new Expectation-Maximization (EM) scheme to impose both spatial and color smoothness to infer natural connectivity among pixels. Unlike previous work, our method takes local color information into consideration, and segment image with soft region boundaries for seamless color transfer and compositing.

Our modified EM method has two advantages in color manipulation: First, subject to different levels of color smoothness in image space, our algorithm produces an optimal number of regions upon convergence, where the color statistics in each region can be adequately characterized by a component of a Gaussian Mixture Model (GMM). Second, we allow a pixel to fall in several regions according to our estimated probability distribution in the EM step, resulting in a transparency-like ratio for compositing different regions seamlessly. Hence, natural color transition across regions can be achieved, where the necessary intra-region and inter-region smoothness are enforced without losing original details. We demonstrate results on a variety of applications including image deblurring, enhanced color transfer, and colorizing gray scale images. Comparisons with previous methods are also presented.

## 1 Introduction

Recent advances in digital image processing and enhancement techniques have made new and useful applications possible. One involves color manipulation between images, which can be applied to perform color correction, noise reduction, and production of high-quality composite images [14, 8, 4, 5, 6].

In [9], Reinhard et al. reported a simple but very successful technique that transfers color characteristics from a source to a target image. In the target image  $I_t$ , the transferred color at pixel  $C_t$  in the  $l\alpha\beta$  color space is:

$$g(C_t) = \mu_s + \frac{\sigma_s}{\sigma_t}(C_t - \mu_t) \quad (1)$$

---

\*This research is supported by the Research Grant Council of Hong Kong Special Administration Region, China: AOE/E-01/99, HKUST6175/04E, and the Chinese University of Hong Kong (CUHK) Direct Grant: 2050332.

where  $\mu_s, \mu_t$  are the means of the underlying Gaussian distribution in the  $l\alpha\beta$  color space of the respective source and target images.  $\sigma_s, \sigma_t$  are the respective standard deviations. We call this approach *global color transfer* because the color statistics are calculated by taking into account all pixels in the respective images.

Global color transfer does not have adequate spatial consideration, so it cannot avoid the following two problems. One is that if the source or target image contains different color regions, the global transfer cannot distinguish the different statistics and will mix regions up. The other problem is that if the color of the two images are very different, in the  $l\alpha\beta$  color space, the chromaticity channels are easily exaggerated which will cause unnatural and saturated result. Swatches [9] can alleviate the errors in some situations, but they cannot solve all problems because they do not provide without sufficient clustering information.

In this paper, we address the above issues, and propose a method to establish spatial connectivity so that local color statistics can be inferred simultaneously with the optimal partitioning of the image space. We call the transfer in the presence of color correspondences and spatial relations as *local color transfer*. It has following three properties:

**Probabilistic segmentation.** Pixel clustering is achieved via *probabilistic segmentation*, where we estimate the probability  $iP_{xy} \in [0, 1]$  that pixel  $I(x, y)$  belongs to region  $i$  in image  $I$ . Hence,  $I(x, y)$  may concurrently exist in several regions according to its probability distribution  $\mathbf{P}_{xy}$ . Previous binary segmentation where  $iP_{xy} \in \{0, 1\}$  is just a special case in our definition.

The advantage of using probabilistic segmentation over binary segmentation is that inter-region smoothness can be enforced by encoding natural connectivity among pixels. In our method, pixel  $I(x, y)$  tends to have a large probability falling into a single region  $i$  when it is in the center of the region, while it has a more uniform probability distribution among several regions if it lies on boundary. This guarantees smooth color transition across regions without mixing up colors in the interior of the regions.

**Expectation-Maximization.** We estimate these probabilistic distributions through a novel propagation step by the Expectation-Maximization (EM) algorithm, and appropriately model all segments as Gaussian Mixtures where the

color statistics of each segment is modelled by one component. This guarantees that if two regions respectively satisfy this model, both of them should have roughly one peak in the corresponding color statistics. Thus, in the transfer process, dominant colors will not be mixed up and natural color transition across regions can be maintained.

**Unified approach.** Because of the generality of GMM, our unified approach can be applied to a variety of applications, including image deblurring, image restoration and colorization of gray scale images.

The rest of this paper is organized as follows. Section 2 reviews previous work in related areas. In Section 3, we describe our method which includes the modified EM method for probabilistic segmentation, mapping function construction and result compositing. We present our results in section 4, with detailed comparison with previous work. The summary and conclusion are given in section 5.

## 2 Previous work

We review previous work proposed for color manipulation between two images, together with several image segmentation techniques related to our probabilistic segmentation and EM techniques.

### 2.1 Color transfer and related applications

Reinhard et al. [9] first performed global color transfer according to Eqn. 1 in [10], after converting the RGB space to the decorrelated color space  $l\alpha\beta$ . In their approach, if the image contains different color regions, swatches are specified by users and are used to divide the colors into clusters. However, for complex scenes, either swatches are inadequate to discern different color statistics, or a lot of swatches need to be specified by user.

The method was extended to gray scale image colorization in [14], where chromaticity transfer is performed after equalizing the luminance channels of both input images. In [7], Levin et al. colorized a gray scale image or movie by assuming that neighboring pixels in space-time with similar intensities should have similar colors. A quadratic optimization was formulated which can be solved efficiently by standard techniques. These colorization techniques require a lot of human interactions if complex textures or patterns are present. Without human interaction, undesirable color mixture will be observed.

As for other related applications in color transfer, in [5], a Bayesian approach was proposed to correct image intensity. Without performing motion deblurring, a low/normal exposure image pair is used. Global color statistics are transferred from the normal exposure (blurred) image to the low exposure (sharp) image. Explicit spatial correspondence was used.

Later, two image restoration or denoising techniques using flash/no-flash pairs were proposed independently in [4, 8]. In comparison, [4] used a compositing approach to in-

	Meaning
$G_i$	Gaussian distribution $G_i(i; \mu_i, \sigma_i)$ for region $i$
$\mu_i^t$	Mean of $G_i$ at iteration $t$
$\sigma_i^t$	Standard deviation of $G_i$ at iteration $t$
${}_iP_{xy}^t$	Probability of pixel $I(x, y)$ in $G_i$ at iteration $t$
$\mathbf{P}_{xy}^t$	Probability distribution of pixel $I(x, y)$ estimated at $t$ . $\mathbf{P}_{xy}^t = \{{}_iP_{xy}^t   i = 1, 2, \dots\}$

Table 1: Notations used in this paper.

tegrate colors from flash/no flash pairs, while in [8], a joint bidirectional filter is used to perform de-noising. The latter approach cannot avoid texture smoothing. Otherwise, noise will be remained even with the guidance of details from the flashed image, due to the use of smoothing filter.

### 2.2 Color segmentation

The watershed algorithm [13] performs color segmentation which easily produces a large number of small regions with hard boundaries. Mean shift segmentation [3] maintains spatial consistency by incorporating the spatial coordinates into their feature space representation. In the filtering step, the kernel size is determined with the help of the spatial domain parameter.

An earlier work in [1] performs color and texture segmentation by EM, which models the joint distribution of color and texture with a mixture of Gaussians in 6-dimensional space (three dimensions for color and three for texture). Since no spatial coordinates are incorporated, after the model has been inferred, it needs a spatial grouping step by applying a maximum-vote filter and connected component algorithm.

In our new EM algorithm, we perform our model estimation in the 3D color space, which makes the EM estimation more stable and run faster. To maintain spatial consistencies and introduce natural probabilistic region boundaries, we introduce an additional propagation step in the loop. For probabilistic segmentation, note that natural matting techniques (such as [2] and [11]) are not applicable in our work, since they do not guarantee smoothness for in certain situations.

## 3 Our approach

Given an input image pair  $I_s$  and  $I_t$ , we first construct a probabilistic segmentation in each of them so that the colors in every segmented region  $r_i$  can be fitted by a Gaussian component  $G_i$  of a GMM appropriately. It guarantees that any two regions in the two input images have similar statistical model, and a natural mapping can be achieved between them. Hence, region  $r_i$  and Gaussian component  $G_i$  are closely related. Table 1 shows the notation to be used in the rest of the paper.

In this section, we first review in section 3.1 the original EM algorithm for estimating general 3D GMM. Afterward, a detailed description on our new model construction and

mapping estimation will be presented. The outline of our approach is as follows:

- Segment the two input images into a set of regions with soft boundaries. The optimal number of regions is also determined automatically in this step (sections 3.2-3.3).
- Construct the mapping function from the source image to the target with or without spatial correspondence. (section 3.4).
- Composite the final image where no visual artifact should be observed among regions (section 3.5).

We will describe these steps in separate sections.

### 3.1 The original EM algorithm for 3D GMM estimation

Initially, we assume that the image can be partitioned into  $N$  regions, namely, the underlying joint color distribution in  $l\alpha\beta$  channels can be approximated by a set of  $N$  Gaussians. It is equivalent to 3D GMM estimation:

**E-step:** The probability that a pixel color  $I(x, y)$  belongs to the  $i$ th Gaussian  $G_i(i; \mu_i, \sigma_i)$  is calculated as:

$${}_iP_{xy} = \frac{\exp(-\frac{(I(x,y)-\mu_i)^2}{2\sigma_i^2})}{\sum_{j=1}^N \exp(-\frac{(I(x,y)-\mu_j)^2}{2\sigma_j^2})} \quad (2)$$

**M-step:** the mean  $\mu_i$  and standard deviation  $\sigma_i$  of each Gaussian  $G_i$  are re-estimated in region  $i$  as:

$$\mu_i = \frac{1}{Z} \sum_{x,y} {}_iP_{xy} I(x, y) \quad (3)$$

$$\sigma_i = \sqrt{\frac{\sum_{x,y} {}_iP_{xy} (I(x, y) - \mu_i)^2}{Z}} \quad (4)$$

where  $Z$  is the normalization factor, which is equal to  $\sum_{x,y} {}_iP_{xy}$ .

The E-step and M-step are iterated until convergence. Usually, the initial values of the above EM algorithm is calculated by performing K-means. The EM algorithm works well on estimating 3D GMM from only the probability or the statistical point of view. However, for an image with 3 color channels, it does not take spatial correlation into account. Note that it is also not practical to upgrade this method to 5D (including 2 image spatial dimensions), which may easily lead to local minima. Also, the number of Gaussians converged is not guaranteed to be optimal.

### 3.2 Modified EM for estimating probabilistic segmentation

In our modified EM, we encode an additional spatial smoothness step within the loop, in order to simultaneously constrain spatial connectivity among pixels and refine the

number of Gaussians iteratively until no two Gaussian components are largely overlapping. Our output is optimized in the number of Gaussian  $N$ .

Given an input image  $I$ , we iterate the following EM process.

- 1. Expectation** Estimate probability distribution  $\mathbf{P}_{xy}^t$  in iteration  $t$  according to the result from iteration  $t-1$ . Instead of using Eqn. 2, we introduce a new representation, subjected to  $\sum_{i=1}^N {}_iP_{xy}^t = 1$ :

$${}_iP_{xy}^t = {}_iP_{xy}^{t-1} + \frac{\exp(-\frac{(I(x,y)-\mu_i^{t-1})^2}{2(\sigma_i^{t-1})^2})}{\sum_{j=1}^N \exp(-\frac{(I(x,y)-\mu_j^{t-1})^2}{2(\sigma_j^{t-1})^2})} \quad (5)$$

where  ${}_iP_{xy}^{t-1}$  is the result from spatial smoothness propagation in step 2 of the last iteration. We initialize  ${}_iP_{xy}^0$  using Eqn. 2. The explanation of Eqn. 5 will be given in section 3.2.1.

- 2. Spatial smoothing** In this step we perform spatial smoothness propagation and update the expectation values as  ${}_iP_{xy}^t$ . Details of this step will be described in section 3.2.2.

- 3. Maximization** Re-estimate  $\mu_i^t$  and  $\sigma_i^t$  using Eqn. 3 and 4 with new probability distribution  ${}_iP_{xy}^t$  for each Gaussian component  $i$ .

- 4. Refine Gaussians** Given the new estimation of Gaussian parameters from the previous step, for every pair of Gaussian components in the same image, e.g.,  $G_i(i; \mu_i^t, \sigma_i^t)$  and  $G_j(j; \mu_j^t, \sigma_j^t)$ , if  $|\mu_i^t - \mu_j^t| < \delta$  for some small  $\delta$ , we merge them and re-estimate their parameters. Subject to the tolerance  $\delta$ , this step produces an optimal number of Gaussians in each image. In practice, we iterate the algorithm for a fixed number of Gaussians until it converges before applying region merging. This produces better convergence result.

- 5. Output** If the difference of means and variances between all corresponding pairs of Gaussian components between iterations  $t$  and  $t-1$  is small enough, and there is no Gaussian merging in current iteration  $t$ , stop. Otherwise, go back to step 1 and begin iteration  $t+1$ .

Steps 3–5 are straightforward. We give further descriptions on steps 1 and 2 in the following subsections.

#### 3.2.1 Expectation

We explain step 1 of our modified EM algorithm. Suppose the region  $i$  to be inferred is shaped like a disc. If a pixel is in the center of region  $i$ , its probability of belonging to this region, or equivalently the corresponding Gaussian component  $G_i$ , should be very high. On the other hand, if the

region is elongated, or the pixel lies on boundary of region  $i$ , we want the pixel to have non-zero probability to fall in other regions. This will introduce a transparency-like ratio among regions to ensure boundary smoothness.

According to the above analysis, we explain how Eqn. 5 operates in different situations. Let us denote  ${}_iP''^t_{xy} =$

$$\frac{\exp(-\frac{(I(x,y)-\mu_i^{t-1})^2}{2(\sigma_i^{t-1})^2})}{\sum_{j=1}^N \exp(-\frac{(I(x,y)-\mu_j^{t-1})^2}{2(\sigma_j^{t-1})^2})}, \text{ then:}$$

- If both  ${}_iP'^{t-1}_{xy}$  and  ${}_iP''^t_{xy}$  have similar distributions, it is highly probable that the neighboring pixels have similar color values, which indicates that this pixel is not lying on any region boundary. Hence, the addition operation increases the probability that the pixel belongs to some dominant Gaussian component.
- If  ${}_iP'^{t-1}_{xy}$  and  ${}_iP''^t_{xy}$  have disparate probability distributions, the pixel receives different propagated value from its neighbors. Hence, it indicates that the pixel is lying close to the boundary of some regions. The addition operation averages these distributions, so that this pixel will have similar probability of fitting into several Gaussian components concurrently.

### 3.2.2 Spatial smoothing

Step 2 in the previous section defines the spatial propagation operation. Denote  $\mathcal{N}(x, y)$  as the neighborhood of  $(x, y)$ , and also suppose  $(x', y') \in \mathcal{N}(x, y)$ . If  $I(x, y)$  and  $I(x', y')$  have similar colors, they should also be close in Gaussian distributions considering spatial connectivity after smoothing, namely,  ${}_iP_{xy} \approx {}_iP_{x'y'}$ . Otherwise, their probability distribution should not be affected so as to generate a spatial partition. According to this analysis, we define the smoothing operation as:

$${}_iP'^t_{xy} = \frac{1}{Z_i} \sum_{(x', y') \in \mathcal{N}(x, y)} D(I(x, y), I(x', y')) {}_iP^t_{x'y'} \quad (6)$$

where  $Z_i = \sum_i \sum_{(x', y') \in \mathcal{N}(x, y)} D(I(x, y), I(x', y')) {}_iP^t_{x'y'}$  is the normalization factor. For simultaneous color and spatial smoothness, we adopt the bilateral filter [12]:

$$D(I(x, y), I(x', y')) = \exp(-\frac{(x-x')^2 + (y-y')^2}{\sigma_d}) \exp(-\frac{|I(x, y) - I(x', y')|^2}{\sigma_g}) \quad (7)$$

where  $\sigma_d$  and  $\sigma_g$  are parameters to control the strength of smoothing in the spatial and color spaces respectively.



Figure 1: The top is the input image. The second row shows one of our automatically segmented regions where we transfer the color from red to blue. The last two rows show the transferred result and the zoom-in view. Note that the spatial smoothness is not uniform along the boundary. Different configurations of neighborhood size  $N$  are shown in 3 columns. See online version for color images.

### 3.3 Evaluation of probabilistic segmentation

After EM estimation,  $I_s$  and  $I_t$  will respectively be associated with a GMM, denoted as  $\{G_{s_i} | 0 \leq i \leq N_s\}$  and  $\{G_{t_j} | 0 \leq j \leq N_t\}$ , where  $N_s$  and  $N_t$  are the total number of components in  $I_s$  and  $I_t$  respectively.

Let us evaluate our probabilistic segmentation by 1) boundary smoothness, and 2) quality of segmentation in comparison with some representative techniques. Since each pixel may belong to several segments according to  ${}_iP_{xy}$ , in the following results (Figs. 1 and 2), we display each pixel color as  $I(x, y) \cdot {}_iP_{xy}$  in each segmented region  $i$ .  ${}_iP_{xy}$  thus acts as a transparency ratio in display.

#### 1. Boundary smoothness of probabilistic segmentation.

In Fig. 1, we show the result for different configurations of neighborhood size  $\mathcal{N}$  where  $|\mathcal{N}^*|$  is the optimal value. The top of figure 1 is the input image. The second row shows one of the regions we segment by our modified EM method. Note that the left side of the image consists of a smooth background, while the right consists of complex patterns. For the shirt of the Teddy bear, the left boundary is smoother than the right boundary. This example depicts different amount of smoothness, which is automatically computed by our

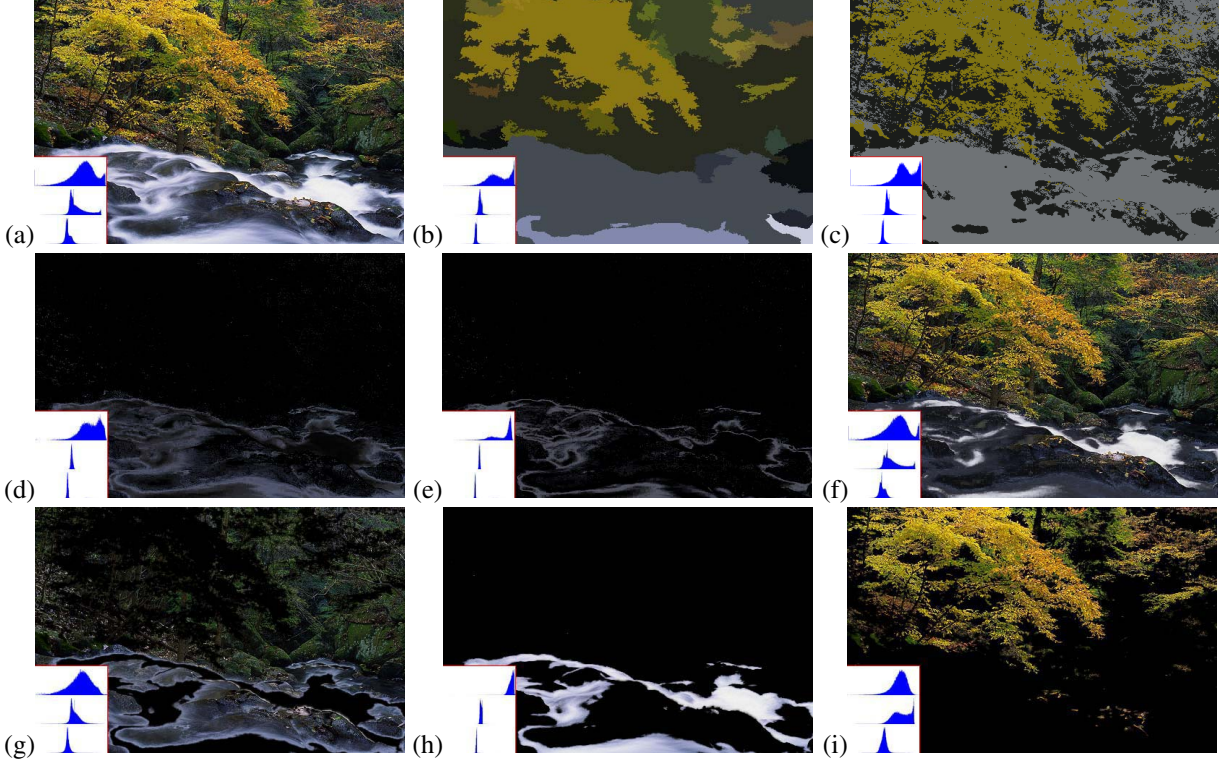


Figure 2: (a) The original image. (b) Image segmented by mean shift segmentation [3]. (c) Image segmented by K-means clustering with  $k = 3$ . (d)–(f): The image segmented by the original EM algorithm, using three Gaussians for estimation. Without spatial consideration, spatially dissimilar patterns are mixed up. (g)–(i): Image segmented by our modified EM algorithm. Spatial information is taken into account, making the estimation less vulnerable to local minima. In (d)–(i), the probability distribution  $\mathbf{P}_{xy}$  is displayed as the alpha value. The histograms in the  $l\alpha\beta$  channels are respectively shown for comparison. Note that our method successfully separates several textures with smooth boundary connection. See online version for color display.

method. The transferred results and their zoom-in are shown in the last two rows for comparison. We observe that if  $|\mathcal{N}|$  is too large, the region boundaries are over-smoothed, leading to low probability that the original red color are completely substituted by the blue color on the boundary of the shirt. On the contrary, if  $\mathcal{N}$  is too small, unnatural results are generated due to the binary boundary. For a wide range of  $\mathcal{N}$ , we can achieve the result for  $|\mathcal{N}| \approx |\mathcal{N}^*|$ . The typical size of  $\mathcal{N}$  is  $7 \times 7$ .

2. *Quality of probabilistic segmentation.* In Fig. 2, we compare our result with those from other methods. Note that mean shift and K-means segmentation cannot preserve complex region shapes, e.g., white river, which violates the single Gaussian assumption and mixes different patterns up. As a result, simple boundary smoothing or feathering cannot generate comparable result to ours. Similarly, the original EM algorithm, cannot separate spatially different patterns easily, e.g., water and leaves. In our method, spatial relations are effectively encoded in our EM algorithm, resulting in clear segmentation of several patterns with well-

preserved complex boundary. Even the interweaved green and yellow leaves are roughly segmented.

### 3.4 Gaussian mapping for color transfer

After performing probabilistic segmentation on the two input images  $I_s$  and  $I_t$  respectively, we obtain a set of segmented regions. In this section, we establish the mapping function  $f(\cdot)$  which maps each Gaussian component  $G_{t_j}$  (related to region  $r_{t_j}$ ) in image  $I_t$  to some Gaussian in image  $I_s$ . We discuss the function construction in two situations:

#### 3.4.1 Without spatial correspondence

In most general cases where two input images are not similar, we estimate  $f(\cdot)$  directly from color statistics. In practice, humans are more sensitive to the luminance channel, where the monotonic constraint should be maintained. Specifically, suppose we have a pair of Gaussian components  $G_{t_i}(t_i; \mu_{t_i}, \sigma_{t_i})$  and  $G_{t_j}(t_j; \mu_{t_j}, \sigma_{t_j})$  where  $\mu_{t_i} \geq \mu_{t_j}$  in the luminance channel. Then, the transferred Gaussian means  $\mu'$  should also satisfy  $\mu'_{t_i} \geq \mu'_{t_j}$ . In our



method,  $G_{t_j}$  is mapped to  $G_{s_i}$  where 1) their means in luminance channel are closest and 2) monotonic constraint is enforced simultaneously.

### 3.4.2 With spatial correspondence

In some applications, such as deblurring [5] and denoising [8], images  $I_s$  and  $I_t$  are similar in content. We define our mapping function  $f(\cdot)$  as the region correspondence in  $I_s$  and  $I_t$ , using the largest amount of overlapping in the respective regions as the mapping criterion. In other words,  $f(G_{t_j}) = G_{s_i}$  if corresponding regions  $r_{t_j}$  and  $r_{s_i}$  are overlapping significantly.

### 3.5 Result compositing

After obtaining the mapping function  $f(\cdot)$  which maps Gaussian component from  $G_{t_j}(t_j; \mu_{t_j}, \sigma_{t_j})$  to  $G_{s_i}(s_i; \mu_{s_i}, \sigma_{s_i})$ , we compute the final transferred color in pixel  $I_t(x, y)$  as:

$$g(I_t(x, y)) = \sum_j t_j P_{xy} \left( \frac{\sigma_{s_i}}{\sigma_{t_j}} (I_t(x, y) - \mu_{t_j}) + \mu_{s_i} \right) \quad (8)$$

where  $t_j P_{xy}$  is the probability that  $I(x, y)$  belongs to region  $t_j$  of the target image. It guarantees the smoothness after the transfer, especially in regions rich in colors.

## 4 Results and comparison

We implement our method and apply it to a variety of examples on a Pentium-M 1.4GHz laptop computer. Our new EM algorithm converges within 4 minutes for an image of size  $512 \times 512$ , starting with 20 Gaussians initially.

We show results on local color transfer as well as other applications, including motion deblurring, denoising using flash/no flash image pairs, and colorizing gray scale images. Comparisons with previous methods are also given.

### 4.1 Local color transfer

Fig. 3 and Fig. 4 show two results of local color transfer. In Fig. 3, the source and the target images are shown in Fig. 2(a) and Fig. 3(a) respectively. Fig. 2(g)-(i) show the segmented regions in the source image while the segmented regions in the target image are illustrated in Fig. 3(b)-(d). We use the mapping function defined in section 3.4.1. The whole process is automatic. Our result is shown in Fig. 3(g).

In Fig. 4, (a) and (b) are two input images. (c) is the result generated by our automatic method. Comparing with results generated by global color transfer in Fig. 4(d) and histogram equalization in Fig. 4(e), our result is natural and less saturated. Our method not only determines the necessary color regions, but also maintains smooth color transition in the entire image.

### 4.2 Motion deblurring from normal/low exposure pairs

Given two images: 1) one is acquired under normal exposure without a tripod and thus motion blurring is inevitable, 2) the other is taken with short shutter speed where the image is under-exposed but has crisp boundary, we transfer the color from image 1 to image 2 to generate a bright and crisp image composite. In this situation, because the source and the target images have strong spatial coherence, we estimate the mapping function using the following relation (section 3.4.2): for each region in the target image, we use the largest overlapping area as the criterion to select matched region in the source image. Fig. 5 shows the result of deblurring via local color transfer.

### 4.3 Denoising using flash/no flash pairs

Similarly, we perform image restoration also using two images: one is taken under flashing and the other is captured using high ISO configuration, which causes a large amount of noise. In [8], the flash image was used to guide denoising in the image with high ISO setting for preserving appropriate colors. However, if the noisy image is severely contaminated, smoothing and noise are inevitable in their result. As an alternative, in our method, we first use a median filter to alleviate the effect of noise. Then, both the source and target images are segmented probabilistically by our EM estimation. Finally, we map colors from the noisy image to the flashed image to construct our noise-free and crisp result. Fig. 6 shows the restoration result. Note that our current method does not consider transferring shadow.

### 4.4 Colorizing gray scale images

In image colorization, we only have luminance channel in the gray scale image. To constrain color transfer, we assume that two pixels in the same region should have similar colors if they have similar luminance value, as described in [7]. Accordingly, in the source and target images, we also perform the probabilistic segmentation. The only modification of our method for this application is that we perform the EM method on the only  $l$  channel in the gray scale image and assign the same distribution to the absent  $ab$  channels. Our method is automatic with the mapping function constructed as in section 3.4.1. Fig. 7 shows one result. By probabilistic segmentation and mapping function construction, we optimally generate 2 Gaussian components, and appropriately propagate blue and green colors in the target image. Note the smooth transition in our result between blue sky and green trees. The whole process is fully automatic.

## 5 Summary and conclusion

In this paper, we propose to perform local color transfer to automatically and faithfully maintain color and spatial coherence. We propose probabilistic segmentation, and

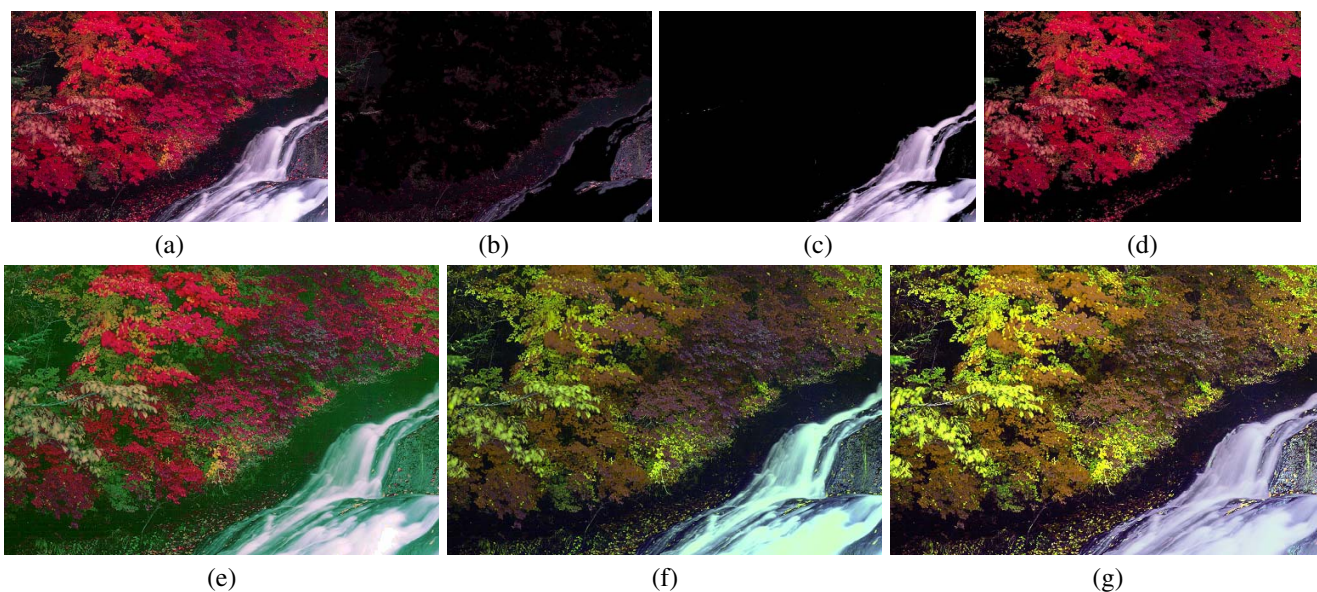


Figure 3: (a) The target image, with the source image shown in Fig. 2(a). (b)-(d) Probabilistic segmentation results. (e) Color transfer result by histogram equalization. (f) Result by global color transfer. Note that (e) and (f) mix up the color of leaves and the river, causing undesirable and inconsistent transferred color to the water. (g) Result from our local color transfer, where the color of water is transferred faithfully from the corresponding color of the rivulet in Fig. 2(a). See color display in online version.



Figure 4: Local color transfer. (a) An image scan of an old photograph of a downtown scene on an overcast day. (b) The original target image captured on a sunny day. (c) Transfer result generated by our local color transfer where Gaussians with closest means are mapped. The scene is automatically modeled by 3 Gaussian components. (d) Transfer result generated by global color transfer. It contains large areas with unacceptable mixture of colors. (e) Transfer result generated by histogram equalization. Undesirable saturation are observed. See online version for color display.

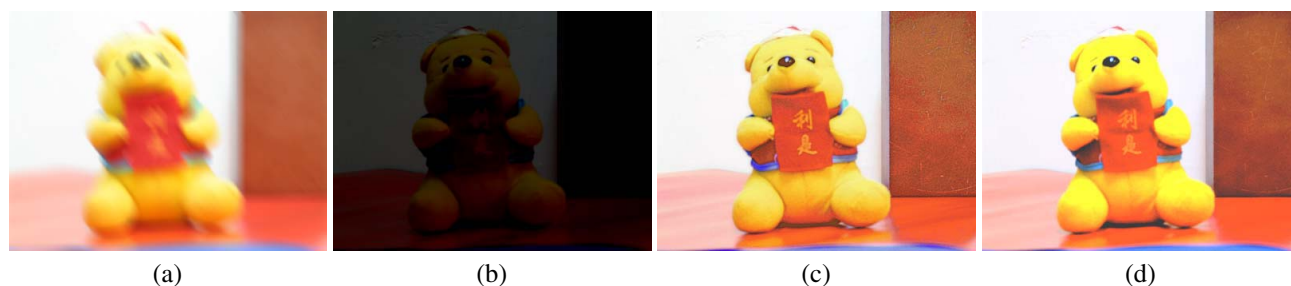


Figure 5: Image deblurring. (a) Source image. (b) Target image. (c) Our result. (d) Result by global transfer. Note that we achieve a smoother transition in our result. Neither the textures on the red packet nor the furs on the doll are smoothed out because of color saturation. See online version for color display.

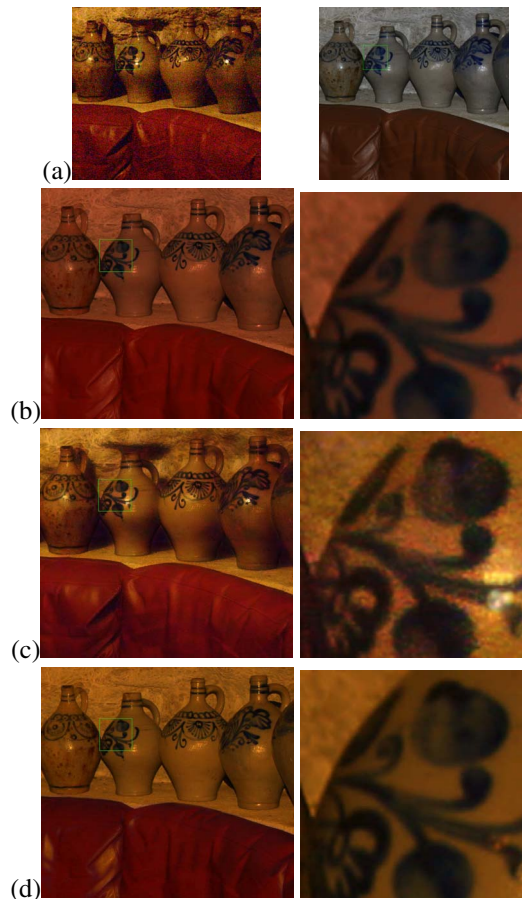


Figure 6: Image denoising comparison. (a) Source/non-flashed and target/flashed images from by using global color transfer, (c) from [8], (d) by using our automatic local color transfer. Our method makes use of the strong spatial coherence between source and target images, so it does not mix up the red shade of the sofa with the bottles and stones, as suffered by global transfer in (b). The result using joint bidirectional filter in [8] still cannot eliminate all noise. See online version for color display.

model the set of regions as Gaussian Mixtures. A modified EM algorithm is also introduced, by augmenting a smoothness propagation step to enforce spatial and color consistency among regions or Gaussian components. Our unified approach is general, and can be applied to a range of applications, including deblurring, image restoration and colorization of gray scale images. In our future work, we will investigate spatially coherent texture transfer and video transfer from examples.

## 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. In *ICCV98*, pages 675–682, 1998.
- [2] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. A bayesian approach to digital matting. *Proceedings of*

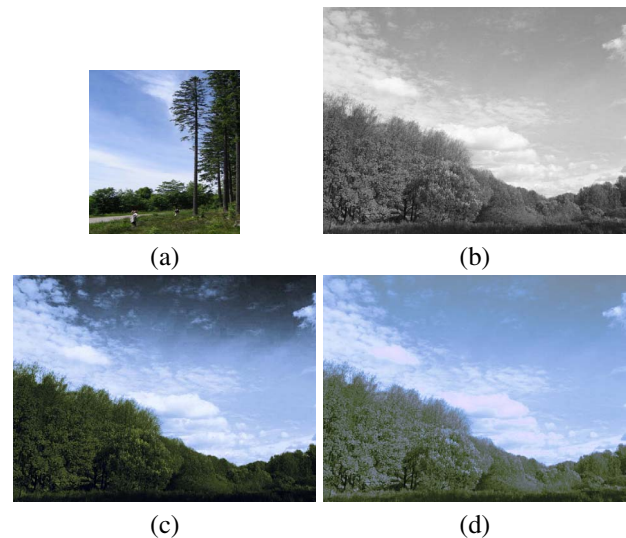


Figure 7: Transfer color to gray scale image (a) The source image. (b) The target image. (c) Result by local color transfer. (d) Result by global color transfer. See online version for color display.

*CVPR'01, Vol. II*, 264–271, 2001.

- [3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, May 2002.
- [4] Elmar Eisemann and Fredo Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, 2004.
- [5] J. Jia, J. Sun, C.K. Tang, and H.Y. Shum. Bayesian correction of image intensity with spatial consideration. In *ECCV04*, pages Vol III: 342–354, 2004.
- [6] J. Jia and C.K. Tang. Image registration with global and local luminance alignment. In *ICCV03*, pages 156–163, 2003.
- [7] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, 2004.
- [8] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004.
- [9] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE CG&A*, 21:34–41, 2001.
- [10] D.L. Ruderman, T.W. Cronin, and C.C. Chiao. Statistics of cone responses to natural images: implications for visual coding. *J. Optical Soc. of America*, 15(8):2036–2045, 1998.
- [11] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *Proceedings of ACM SIGGRAPH'04*, pages 315–321. ACM SIGGRAPH, 2004.
- [12] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV98*, pages 839–846, 1998.
- [13] Lee Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE PAMI*, 1991, 13(6):583–598, 1991.
- [14] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *SIGGRAPH*, pages 277–280, 2002.