Color Transforms for Creative Image Editing

Pavel Kisilev and Daniel Freedman
Hewlett-Packard Laboratories, Haifa, Israel
{pavel.kisilev, daniel.freedman}@hp.com

Abstract

In this paper, we present a unified approach for the problem of computing color transforms, applications of which include shadow removal, object recoloring, and scene relighting. The detection of source and target regions is performed using a Bayesian classifier. Given these regions, the computed transform alters the color properties of the target region so as to closely resemble those of the source region. The proposed probabilistic formulation leads to a linear program (similar to the classic Transportation Problem), which computes the desired transformation between the target and source distributions. This formulation allows the target region to acquire the properties of the source region, while at the same time retaining its own look and feel. Promising results are shown for a variety of applications.

Introduction

Recent years have witnessed great advances in digital photography, and with them the increasing accessibility of various image editing and manipulation tools. However, most of these tools, such as Photoshop, still require a skilled operator and are far from being automatic. In this paper, we propose a technique for computing a color transform between two objects within an image, or between two images. In our approach, source and target regions are specified, either automatically or semi-automatically, and the color properties of the target region are transformed so as to closely resemble those of the source region. We develop a unified scheme for these tasks based on the computation of a *flow* between target and source distributions. The algorithm is quite effective, and we show results on three applications: shadow removal, recoloring, and relighting.

Before delving more deeply into details, we begin with a short review of prior work.

Prior Work

Many shadow detection and removal methods are based on the intrinsic image separation approach, wherein a single image is decomposed into its reflectance and illumination components (e.g., [Wei01, TFA05]). In the algorithms employing the retinex theory [LM71], this separation is based solely on the assumption that large and sharp changes in lightness stem from the reflectance change, while small and slowly changing gradients are mainly due to the illumination changes. For example, in [ODC] the authors use the bilateral filter to separate small features, such as textures, from large features, such as strong edges. The approach requires various user inputs including the typical texture size.

In a different direction, a series of papers by Finlayson *et al.* ([FHD02, FHLD06], and references therein) exploits an approach based on a physical model, including Planckian light and Lambertian reflectance. The authors make the following simplifying assumptions: (i) the illumination is constant in the shadows region, (ii) the shadows have sharp edges, and (iii) the reflectance does not change across the shadow boundaries. These assumptions are valid only in very specific settings.

Another set of techniques is based on the *matting* approach, where an observed image is modeled as a linear combination of fully lit and fully shadowed counterparts [CGC$^+$03]. Further development of the matting approach is presented in [SL08], where the authors propose a shadow formation model in which the lit intensity at a given pixel is an affine function of the shadowed intensity. The illuminated pixel color is recovered by estimating four parameters of the affine model, based only on the mean colors of the pixels in the lit and shadowed regions, and on standard deviations of their corresponding luminances.

Recoloring methods have developed separately from those dealing with shadow removal. Many recoloring papers focus on the problem of transferring color to gray scale images, and vice versa (e.g., [WAM02]). In [XMK06], the authors propose a recoloring method that is based on alpha matting and compositing, and a color transformation algorithm. The transformation function is postulated to be single-valued and monotonically increasing in the destination pixel intensity domain. The authors estimate the cumulative distribution function (cdf) of the destination region in an image as simply a scaled version of the cdf of the source region. Finally, the compositing operation is implemented using the alpha matte equation. In [LLW04], the authors propose a natural colorization approach which is based on the assumption that pixels having similar intensities should have similar color. In this paper, the input image (or sequence) is gray scale and the output image is a colored image. In [GH03], the authors present a method for recoloring a destination image according to the color scheme from the source image. The image is first segmented into groups of pixels with similar color; then, the color palette for an image is constructed by choosing most typical colors from the above segments. Color transfer is computed by matching the segment areas between the source and destination segments, and finding the closest Euclidean distance match between pairs of colors from the source and destination segments. In [RAGS01], the authors apply a linear transformation that scales the mean and the variance of the target area according to the ones from the source area. In [TJT05], the authors use the approach in [RAGS01] along with a simple segmentation technique to perform recoloring operations.

Somewhat related work in [PGB03] describes a generic interpolation machinery based on solving Poisson equations. This approach belongs to the gradient domain methods. The authors present various editing applications such as object importation from one region (or an image) to another, seamless cloning of textures, and others.

Paper Outline

The remainder of this paper is organized as follows. In the next section, we outline the problem of computing color transforms, and divide the problem into two stages: detection and computation of the color transform. In the following section, we propose algorithms for both detection and computation of the color transform, with most of the focus on the latter. In the section on applications, we present results of applying the algorithm to the problems of shadow removal, object recoloring, and relighting.

The Problem

The problem we are interested is transfer of color properties from one object in an image to another. More specifically, given some knowledge of the source and target properties, we wish to accomplish two tasks:

1. **Detection:** Find the regions of the image which correspond to the source and target properties. These regions constitute the source and target objects, respectively.
2. **Photometric Transform:** Transform the color properties of the target object to resemble those of the source object, in perceptually meaningful way.

While we are interested in both detection and the color transform, the bulk of our exposition will focus on the latter.

Before posing the problem in a more formal fashion, we discuss two applications of this framework. The first is shadow removal. In this problem, both the source and the target come derive from the same material or object; however, the source is lit, while the target is in shadow. The second application is object recoloring. Here, the goal is to simply take a given object, and to transfer its colors to a second object. Often the two objects are of the same type (such as two cars), but this need not be the case. Both shadow removal and object recoloring are desirable operations for many computer graphics applications, as well as in the context of Photoshop-type software.

Let us now turn to a more formal statement of the problem. Let color be denoted by the vector $c \in \mathscr{C} = \mathbb{R}^d$. The color c could be represented in Lab space, RGB space, or any of many other possibilities. (In principle, the techniques presented in this paper could also be applied to learning texture transforms, where texture is represented, for example, as the output of a filter bank.) Let the image domain be $\mathscr{X} \subset \mathbb{R}^2$, so that an image is given by $c : \mathscr{X} \to \mathscr{C}$. Then our two goals may be restated as follows:

1. **Detection:** Find two subsets of the image domain $\mathscr{X}$, the source region $\mathscr{S}$ and the target region $\mathscr{T}$, with $\mathscr{S} \cap \mathscr{T} = \emptyset$.
2. **Photometric Transform:** For each pixel $x \in \mathscr{T}$, compute a mapping $c(x) \to \Phi(c(x))$ such that the collection $\{\Phi(c(x)) : x \in \mathscr{T}\}$ is in some perceptual sense similar to the collection $\{c(x) : x \in \mathscr{S}\}$.

The Algorithm

In this section, we present the algorithm for computing color transforms. Although logic would dictate that we begin our exposition with the detection of the source and target regions, we begin instead with a discussion of the color transform algorithm, which allows the detected target region to be transformed into something resembling the detected source region. This is done in order to emphasize the color transform algorithm, which is the main contribution of the paper. The basis for the algorithm is a version of the Transportation Problem [Hit41, Rac85], which can be posed as a linear program; this optimization, combined with an appropriate interpolation scheme, yields an effective technique for computing color transforms. We then move on to a discussion of the Detection Algorithm, including an examination of the form of user-input required.

The Color Transform

We begin with the main algorithmic contribution of this paper: the transformation of the color properties of the target region so that they closely resemble those of the source region. To repeat our earlier formulation of this problem: for each pixel $x \in \mathscr{T}$, we wish to compute a mapping $c(x) \to \Phi(c(x))$ such that the collection $\{\Phi(c(x)) : x \in \mathscr{T}\}$ is in some sense similar to the collection $\{c(x) : x \in \mathscr{S}\}$. This is not a straightforward problem, as the two collections may be quite different. For example, probability distributions over the source and target pixels (i.e. over their colors) may have different numbers of modes, different shapes, and so on.

Our solution to this problem is to use the classic Transportation Problem to compute the transformation between the two distributions. In the Transportation Problem [Hit41, Rac85], the goal is to match supplies of certain quantities with demands for these quantities, bearing in mind the absolute amounts of both supplies and demands. This is computed via a *flow* between supplied quantities and those demanded. In what follows, we make the analogy between supplied quantities with the target region, and demanded quantities with the source region. The flow thus computed by the solution to the Transportation Problem is exactly what we are after: it allows us to transform target pixels into source-like pixels.

Before turning to a more formal statement of the problem, it is of interest to note that the Transportation Problem has been used in computer vision applications before, most notably for the computation of the Earth Mover's Distance (EMD) [RTG00]. Indeed, a by-product of the EMD computation is the flow that will be useful to us; however, it is just that, a by-product, whose use is to compute the EMD metric, and is not employed for any other purpose. Note also that a few works have used somewhat related ideas [HZTA04], albeit in the continuous setting and for different applications, such as registration.

Let us begin by fixing notation. The labels for source and target objects will be s and t respectively, and when speaking of an object defined for both, we will use $\alpha \in \{s,t\}$. We are given source and target probability distributions, which are learned as part of the Detection Algorithm (see the next section). As usual, there are a variety of ways of representing distributions; we have chosen to use simple histograms, as more sophisticated techniques provide little or no improvement to the algorithm at the cost of increased complexity.[1] We represent the source and target distributions compactly as a list of histogram bins with non-zero probability, i.e. $\{(\bar{c}_i^\alpha, \bar{p}_i^\alpha)\}_{i=1}^{n_\alpha}$, where $\bar{c}_i^\alpha$ is a bin-center, $\bar{p}_i^\alpha$ is the corresponding probability mass for that bin, and n_α is the number of such bins. Finally, for a given color c, let $[c]^\alpha$ be the bin in

[1] This is partially due to the fact that the distributions tend to be relatively sparse. In cases in which the distributions are somewhat more dense, it may be advantageous to represent the distributions with a pared down version of the Kernel Density Estimate, for example representing each KDE by its modes.

which it resides (where again $\alpha \in \{s,t\}$).

Given the above notation, we can now turn to the problem of computing the color transform, for which we use the Transportation Problem. Recall that the Transportation Problem is formulated as follows[Hit41, Rac85]: let the *flow* between the target and source distributions be given by f_{ij}, where the indices i and j range over the (non-empty) bins of the target and source distributions, respectively. That is, f_{ij} can be thought of as the part of target bin i which is mapped to source bin j. Now, let the *color distance* between two colors[2] be given by $D(c_1, c_2)$. As we shall see in the sequel, while this distance is sometimes taken as the ordinary Euclidean distance, there are times when other choices are more appropriate. In any case, taking the color distance as given for the moment, we would like to solve the following optimization:

$$\min_{\{f_{ij}\}} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} f_{ij} D(\vec{c}_i^t, \vec{c}_j^s)$$

$$\text{subject to} \quad \sum_{j=1}^{n_s} f_{ij} = \vec{p}_i^t \quad i = 1, \ldots, n_t$$

$$\sum_{i=1}^{n_t} f_{ij} = \vec{p}_j^s \quad j = 1, \ldots, n_s$$

The goal of the objective function is to map the target colors $\vec{c}_i^t$ to corresponding source colors $\vec{c}_j^s$ in such a way that the color distance between them is as small as possible. However, we cannot reasonably expect that each bin of the target distribution maps neatly to exactly one bin of the source distribution. Thus, we allow target bins to be spread over several source bins, subject to the two constraints of the original Transportation Problem, which ensure "conservation of probability" for both the target and source distributions.

In fact, in our case requiring conservation of probability is too extreme[3] as it assumes that the source and target regions contain exactly the same amounts of "comparable colors." As a result, we modify the optimization as follows:

$$\min_{\{f_{ij}\}} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} f_{ij} D(\vec{c}_i^t, \vec{c}_j^s)$$

$$\text{subject to} \quad \vec{p}_i^t/\eta \le \sum_{j=1}^{n_s} f_{ij} \le \eta \vec{p}_i^t \quad i = 1, \ldots, n_t$$

$$\vec{p}_j^s/\eta \le \sum_{i=1}^{n_t} f_{ij} \le \eta \vec{p}_j^s \quad j = 1, \ldots, n_s$$

$$\sum_{i,j} f_{ij} = 1$$

where the final constraint, which was enforced implicitly in the original Transportation Problem, is now made explicit. $\eta \ge 1$ is a parameter which describes the slackness of the conservation constraints; typically, we choose $\eta \approx 3$.

Now, given the solution to the modified Transportation Problem, the problem of computing the color transform is effectively solved. In particular, if we were interested only in mapping the bin centers $\vec{c}_i^t$, we have the following color transform rule:

$$\vec{c}_i^t \to \frac{\sum_{j=1}^{n_s} f_{ij} \vec{c}_j^s}{\sum_{j=1}^{n_s} f_{ij}} \equiv \Phi(\vec{c}_i^t) \tag{1}$$

That is, we use the flow to average over the source bin centers in the natural way, and then normalize. Notice that $\sum_j f_{ij} \le \eta \vec{p}_i^t \ll 1$, so that the normalization is crucial.

Note that the color transform Φ is defined only for the bin-centers $\vec{c}_i^t$; however, we wish to transform not just the bin centers of the target distribution, but all of the pixels in the target region $\mathscr{T}$. To achieve this, the simplest option is to map a pixel $c(x)$ in $\mathscr{T}$ to its corresponding bin $[c(x)]^t$, and then to use the transformation rule in Equation (1) on the binned value. Predictably, however, this leads to binning artifacts; two colors which are quite close may in fact lie in different bins, and therefore be mapped to quite different values. Instead, we use the transformation rule given in Equation (1) in combination with a simple interpolation scheme.

To wit, consider a pixel in $\mathscr{T}$ with color c (we drop the argument x to simplify notation), with histogram bin $[c]^t$. Now, given a bin i in the target distribution, then let $\mathscr{N}_i$ be the union of the bin itself, as well as the neighboring (non-empty) bins within the histogram; for example, if the histogram is 2-dimensional, and we use a standard 4-neighborhood, then $\mathscr{N}_i$ would contain at most 5 elements. (In general, for a d-dimensional histogram, we use a $2d$-neighborhood, so that $\mathscr{N}_i$ contains at most $2d+1$ elements.) For each bin i in the neighborhood of c, i.e. in $\mathscr{N}_{[c]^t}$, we compute a weight based on the distance $D(c, \vec{c}_i^t)$ between c and the center of bin i, $\vec{c}_i^t$. In particular, the weights are given by

$$w_i(c) = \frac{\xi(D(c, \vec{c}_i^t))}{\sum_{j \in \mathscr{N}_{[c]^t}} \xi(D(c, \vec{c}_j^t))}$$

where ξ satisfies $\xi'(\cdot) < 0$ and $\xi(0) = \infty$; the latter property ensures that the scheme is truly interpolatory, rather than an approximation scheme. We choose $\xi(d) = d^{-1}$, though other choices are possible. In this case, the final color transform rule is given by

$$\Phi(c) = \sum_{i \in \mathscr{N}_{[c]^t}} w_i(c) \Phi(\vec{c}_i^t) \tag{2}$$

Detection

In order to detect the source and target regions, we must have some prior knowledge of both the source and target images. To begin with, let us assume that we have probability densities over c for both the source and target, i.e. $\rho_s(c) = \rho(c|s)$ and $\rho_t(c) = \rho(c|t)$. Let us further assume a uniform distribution over the regions that are neither source nor target, i.e. $\rho_n(c) = \rho(c|n) = \theta$, where θ is a constant chosen so that $\rho_n(c)$ integrates to 1. Now, the standard Bayesian classifier will classify a value of c as belonging to the source if

$$\rho(s|c) > \max\{\rho(t|c), \rho(n|c)\}$$

(Note that if there is equality, we are on the boundary of at least two classes.)

[2] In the work on the Earth Mover's Distance, D is generally called the *ground distance*.

[3] For example, suppose that the source image is 50% light red and 50% dark red, and the target image is 40% light blue and 60% dark blue; assume further that the light colors have matching L values, as do the dark values, and that our color distance is simply the absolute value of the difference in L. Then the full conservation of probability will require coloring part (10%) of the *dark* blue section of the target image in *light* red, which is obviously not desirable.

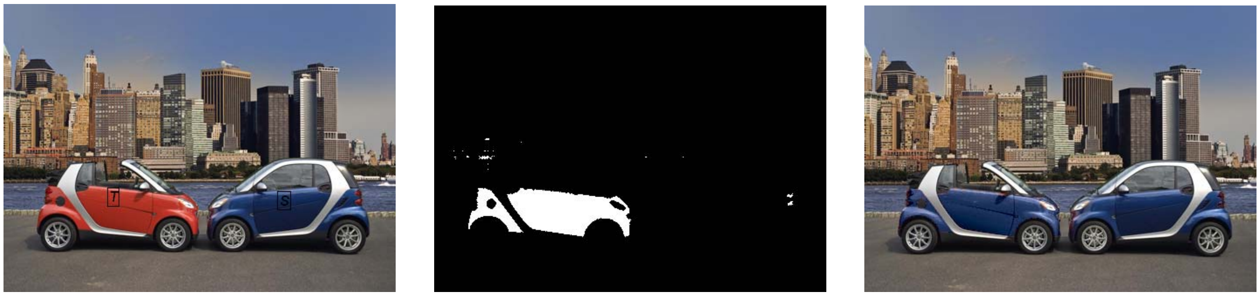

Figure 1. *Recoloring example, wherein the source and target seeds are taken from the same picture. Left: the original. Middle: the binary detection map for the target region. Right: the recolored image.*

Now, from Bayes' Rule we have that $\rho(s|c) = \rho(c|s)P(s)/\rho(c)$, where $P(s)$ is the probability that a given pixel belongs to the source. Assuming, in the absence of other knowledge, that $P(s) = P(t) = P$ and $P(n) = 1 - 2P$, then the Bayesian classifier becomes:

- Choose $x \in \mathscr{S}$ if $\rho_s(c(x)) > \max\{\rho_t(c(x)), \theta'\}$.
- Choose $x \in \mathscr{T}$ if $\rho_t(c(x)) > \max\{\rho_s(c(x)), \theta'\}$.
- Choose x as neither source nor target in all other cases.

where $\theta' = (1-2P)\theta/P$. It is clear, therefore, that given the source and target densities $\rho_s(c)$ and $\rho_t(c)$, we have a simple classifier that depends only on the choice of the single parameter θ'.

The next question, then, concerns the origin of the source and target densities. In certain applications, it is possible that these densities are known *a priori*; for example, one might use any of a number of schemes to learn the color density of blue skies, grass, or skin (e.g., [WH00]). However, assuming that this is not the case, we use two variants of a semi-automatic scheme. In the simpler variant, used for recoloring or for shadow removal, the user simply chooses two rectangles, one which surrounds source pixels and the other target pixels. Based on these rectangles, we then compute histograms for each of the source and target in the relevant space – in our case, Lab color. (For an explanation as to why histograms are used instead of more sophisticated density estimates, please see the prior section.) Note that the histogram bin size in some sense determines the extent to which we extrapolate from the information conveyed by the pixels within the user-chosen rectangles.

The second variant of the semi-automatic scheme is more complex, and may be used for shadow removal. In this case, the user chooses a single rectangle, which encompasses both lit and shadowed pixels. The pixels, which are represented in Lab space, are then divided by performing a k-means clustering, with $k = 2$, on the L-channel. Then, corresponding source and target distributions are easily computed. This heuristic works very well in practice.

Applications

As has already been noted, the framework that has been developed thus far is a general one. In what follows, we show results from the application of this framework to the specific problems of image recoloring and relighting, and shadow removal.

Recoloring and Relighting

For the problem of recoloring or relighting, we seek a mapping from target to source which aims to impose the color of the source upon the target. In order to achieve this goal, we use Lab color, and define the color distance as the regular L_2 distance in Lab space:

$$D^2((L_1,a_1,b_1),(L_2,a_2,b_2)) = (L_1-L_2)^2 + (a_1-a_2)^2 + (b_1-b_2)^2$$

As noted earlier, we use the more straightforward form of user interaction here, in which the user selects two small rectangles, one each from the source and target regions.

Figure 1 shows an example of recoloring experiment wherein both the source and target seeds are taken from the same picture. In this case our task was to paint the left car that was originally red using the blue color of the right car. Note that while the colors have been changed to match that of the right car, the shading and highlights of the left car have been preserved, as in the original image; see Figure 2. Figure 3 shows a relighting example wherein the source and target seeds are taken from two different pictures. In particular, we wanted to "embed" the blue sky from the left picture into the middle picture with gray sky. The result is depicted in the right image. Here again, we preserved the sky texture from the original lightness channel. Notice that in the above examples our method yields a natural look and feel in the reconstructed images.

Figure 2. *Detail from the recoloring example in Figure 1. Note the way in which the shading and highlights are preserved.*

Shadow Removal

For the problem of shadow removal, we seek a mapping from target (shadowed pixels) to source (lit pixels) which aims to retain the chroma characteristics of the target pixels, while at the same time inserting the lightness characteristics of the source pixels. In order to achieve this goal, we use Lab color, and define the color distance as

$$D^2((L_1,a_1,b_1),(L_2,a_2,b_2)) = (a_1-a_2)^2 + (b_1-b_2)^2$$

That is, the color distance depends only on chroma, and not on lightness. Recall that the color distance is used only in computing the flow f, i.e. in matching target pixels with source pixels;

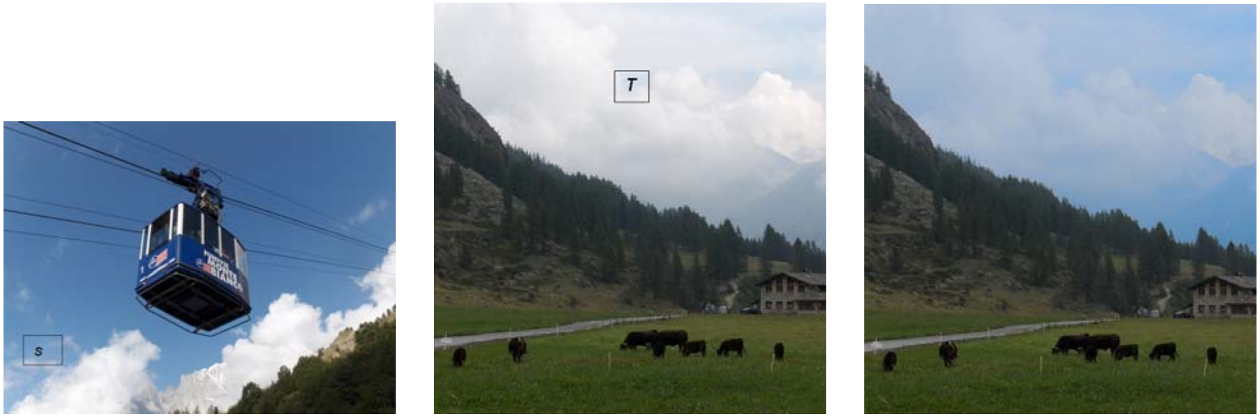

Figure 3. *Relighting example, wherein the source seed region comes from the left image, and the target seed region is taken from the original (in the middle). Right: the relit image.*

the transformation rule Φ itself (see Equation (2)) operates on the target pixels, and hence inserts the desired lightness. As noted earlier, we use the smarter form of user interaction here, in which the user selects a single small rectangle which overlaps the shadowed and lit regions; see Figure 4.

Figure 4 depicts two examples of shadow removal. The top row shows the original images. The middle row shows the detection maps. Here, white areas correspond to the lit areas detected according to the source seed, while gray areas correspond to the shadowed areas detected according to the target seed. The color code in the detection map is as follows: the closer the color is to white, the closer the pixel (from the color distance viewpoint) is to the source cluster. The black areas do not belong to either the source or the target clusters, and are therefore left unchanged in the reconstructed image. The bottom row shows reconstructed images with shadows removed using the proposed approach. Notice how the 'look and feel' from the lit areas is accurately reproduced in the shadowed areas.

Figure 5 depicts the source and target seed region clouds of points in Lab space, from the grass shadow example (the upper left in Figure 4). Notice that the location of the point clouds suggests that in order to preserve a natural look in the reconstructed image, the color transform needs to adjust both lightness and chroma values in some non-trivial way. In areas with homogeneous color the linear approach might be sufficient. However, in areas with large color variations, simple approaches fail. To further prove this point, we compare our method with the linear transformation method of [RAGS01], and with a more sophisticated pyramid based method of [SL08]. The results are presented in Figure 6. The images from left to right are: (1) the original; (2) a linear transformation that scales the mean and the variance of the shadowed area according to the ones from the lit area (as in [RAGS01]); (3) a pyramid-based approach from [SL08]; (4) the proposed approach. Note that, although there are artifacts at the boundary of the shadowed areas as a result of misdetection (these are treated in [SL08] by a separate algorithm), our method yields the most natural looking image.

Conclusions and Future Directions

We have presented a novel approach for the problem of learning color transforms. Given source and target regions, the proposed approach uses a modified Transportation Problem formulation to learn the color properties of the source region. Then, the color properties of the target region are transformed so as to closely resemble those of the source region. The usefulness of the approach is evidenced by promising results on several applications: object recoloring, relighting, and shadow removal.

Future work will concentrate on improving the detection stage, to move toward a fully-automated technique. In addition, we will investigate the use of the technique on more complex photometric feature spaces, by complementing color with additional dimensions corresponding to a texture descriptor.

References

[CGC+03] Y.Y. Chuang, D.B. Goldman, B. Curless, D.H. Salesin, and R. Szeliski. Shadow matting and compositing. *ACM Transactions on Graphics*, 22(3):494–500, 2003.

[FHD02] G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images. *Lecture Notes in Computer Science*, pages 823–836, 2002.

[FHLD06] GD Finlayson, SD Hordley, C. Lu, and MS Drew. On the removal of shadows from images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):59–68, 2006.

[GH03] G.R. Greenfield and D.H. House. Image recoloring induced by palette color associations. *Journal of WSCG*, 11(1):189–196, 2003.

[Hit41] F.L. Hitchcock. The distribution of a product from several sources to numerous localities. *J. Math. Phys. Mass. Inst. Tech*, 20:224–230, 1941.

[HZTA04] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *International Journal of Computer Vision*, 60(3):225–240, 2004.

[LLW04] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG)*, 23(3):689–694, 2004.

[LM71] E.H. Land and J.J. McCANN. Lightness and retinex theory. *Journal of the Optical society of America*, 61(1):1–11, 1971.

[ODC] B.M. Oh, F. Durand, and M. Chen. Image-based modeling and photo editing. *In Proc. SIGGRAPH 2001 pages=433–442, year=2001, publisher=ACM Press*.

[PGB03] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

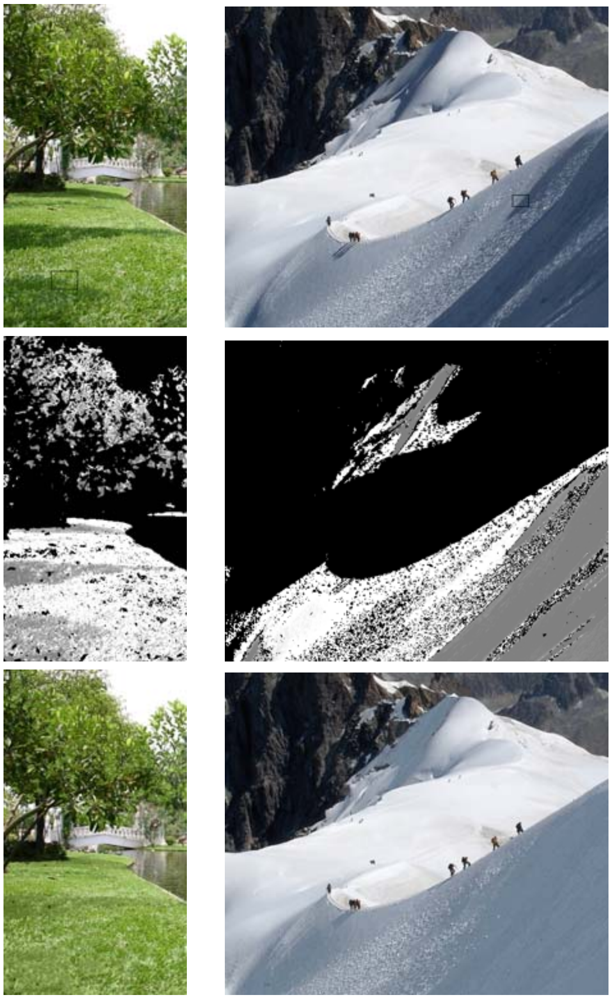

Figure 4. *Shadow removal examples. Top: the original images. Middle: the detection maps. Bottom: reconstructed images with shadows removed using the proposed approach.*

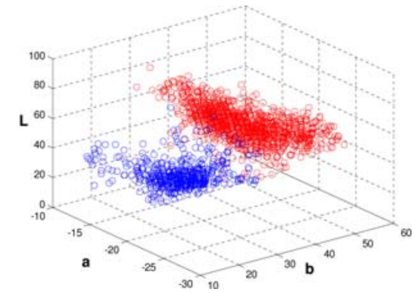

Figure 5. *The source and target seed region clouds of points in Lab space, taken from the left picture in Figure 4. Blue dots correspond to the target, red dots correspond to the source.*

Figure 6. *Comparison of the proposed method with competing approaches. From left to right: (1) the original; (2) the linear transformation from [RAGS01]; (3) the pyramid method from [SL08]; (4) the proposed approach.*

[Rac85] ST Rachev. The Monge–Kantorovich mass transference problem and its stochastic applications. *Theory of Probability and its Applications*, 29:647, 1985.

[RAGS01] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, pages 34–41, 2001.

[RTG00] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[SL08] Y. Shor and D. Lischinski. The Shadow Meets the Mask: Pyramid-Based Shadow Removal. In *Computer Graphics Forum*, volume 27, pages 577–586. Blackwell Publishing, 2008.

[TFA05] MF Tappen, WT Freeman, and EH Adelson. Recovering intrinsic images from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1459–1472, 2005.

[TJT05] Y.W. Tai, J. Jia, and C.K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 1, 2005.

[WAM02] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 277–280. ACM New York, NY, USA, 2002.

[Wei01] Y. Weiss. Deriving intrinsic images from image sequences. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, volume 2, 2001.

[WH00] Y. Wu and TS Huang. Color tracking by transductive learning. In *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings*, volume 1, 2000.

[XMK06] X. Xiao, L. Ma, and M. Kunze. Object-Based Image Recoloring Using Alpha Matte and Color Histogram Specification. *Lecture Notes in Computer Science*, 4270:194, 2006.

Author Biography

Pavel Kisilev received his BSc in Electrical Engineering and Computer Sciences from Ben Gurion University of the Negev (1994), and his MSc (1998) and PhD (2002) in Electrical Engineering from the Technion. He worked as an algorithm development engineer at the Tensor Systems (1994-1995), and as a research associate at the Vision Laboratory at the Technion (2001-2003). Since 2003 Pavel has been a Senior Research Scientist at HP Laboratories, Israel. Pavel's main interests are Image and Video analysis and processing, and biological signals and systems.

Daniel Freedman received the AB degree in physics from Princeton University in 1993 and the PhD degree in engineering sciences from Harvard University in 2000. He was at Rensselaer Polytechnic Institute (RPI) from 2000-9, first as an Assistant Professor and then as an Associate Professor in the Department of Computer Science. He is currently a Senior Research Scientist at HP Laboratories, Israel. His research interests include computer vision, image processing, and computational geometry and topology. Dr. Freedman is a member of Sigma Xi and the IEEE.