

# A Palette-Driven Approach to Image Color Transfer

G. R. Greenfield<sup>1</sup> and D. H. House<sup>2</sup>

<sup>1</sup> Mathematics & Computer Science, University of Richmond, Richmond, Virginia, USA

<sup>2</sup> Visualization Lab, College of Architecture, Texas A&M University, College Station, Texas, USA

---

## Abstract

*Color analysis of images for the purpose of color balancing, color contrast, and color correction is critical in image processing applications. Color analysis of images for the purpose of palette extraction has received less attention. Motivated by the question of how best to transfer color between two non-photorealistic images in such a way that artistic intent and image aesthetics are taken into consideration, we consider a palette driven approach to the image color transfer problem. Our goal is the transfer of chromatic content from a source image to a destination image with careful consideration given to “value structure” and artistic intent. We show examples of color transfers using our methods.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation, I.4.8 [Image Processing and Computer Vision]: Scene Analysis

---

## 1. Introduction

The problem of automatically re-targeting (i.e. transferring) traits from a source image to a destination image has received increased attention following the explosion in popularity of non-photorealistic rendering [GG01]. Re-targeting problems gained further impetus from the success of Hertzmann et al [HJO\*01] in synthesizing (i.e. generating) images with certain desired traits by using a technique they called *image analogies*. Central to their technique was presenting an example of an image “relationship” by exhibiting images  $A$  to  $B$  such that when given image  $A'$  they could synthesize an image  $B'$  with the property that, by analogy,  $A'$  and  $B'$  also satisfied the relationship. Clever use of their technique provided a means of synthesizing images for which texture and other stylistic traits were re-targeted from a source image. In this paper, we consider the problem of re-targeting *color* from a source image to a destination image. Because it raises important and interesting questions about the quantification of color aesthetics, our particular emphasis is on color re-targeting of non-photorealistic images. Color re-targeting is often referred to as image color transfer or image re-coloring.

### 1.1. Grayscale re-coloring

The re-coloring of grayscale images using semi-automated techniques, where users provide cues in order to facilitate the image re-coloring, has been investigated by several research groups. In Welsh et al [WAM02], grayscale re-coloring was achieved by asking users to identify and associate small rectangles, called “swatches,” in both the source and destination images to indicate how certain key colors should be transferred. Using a technique reminiscent of image analogies, Levin et al [LLW04] produced grayscale re-colorizations of *video* by having users pick colors from a source image and draw freehand curves to cue where and how color transfer should occur for selected destination *frames*. Also of interest is an image re-coloring scheme for *gamut* replacement described in [RGW05] that uses grayscale re-colorization methods.

### 1.2. Non-interactive re-coloring

Turning to non-interactive (i.e. fully-automated) color transfer techniques, Reinhard et al [RAGS01] used statistical methods in order to *color correct* natural landscape images by transferring color “characteristics” from a source image to a destination image. Their transfer methods relied heavily on the properties obeyed by natural images when they are analyzed in Ruderman’s  $\alpha\beta$  color space [RCC01].

Chang et al [CSN02] also considered color correction of landscape images, invoking an absolute color categorization scheme and relying on geometric techniques to modify colors based on their locations within convex regions of  $L^*a^*b^*$  that were determined from their categorization. They subsequently refined the process to encompass more general color transfer problems [CSN03] [CUSN04]. Curran [Cur02] described a “computer enabled” recoloring process by Cai which featured a grayscale NASA image that was re-colored using a “coloring system from Monet.” Curran implied that this re-coloring was obtained using a fully automated re-coloring system. This re-coloring can also be found on Cai’s web page devoted to descriptions of his MONETO and SALIENT projects at <http://www.andrew.cmu.edu/user/ycai/moneto.html>. Although details have not been published, it seems reasonable to assume that the origins of this work trace back to [CS02]. Grundland and Dodgson [GD05] use sophisticated global color space transformation techniques in  $La''b''$  color space, which they refer collectively to as *histogram warping*, to achieve image color transfer based on “key color” assignments. Evidently their system can operate in either fully-automatic or semi-automatic mode. Morovic and Sun use 3D color histogram matching for similar purposes [MS03] [MS02].

### 1.3. Re-coloring for artistic effect

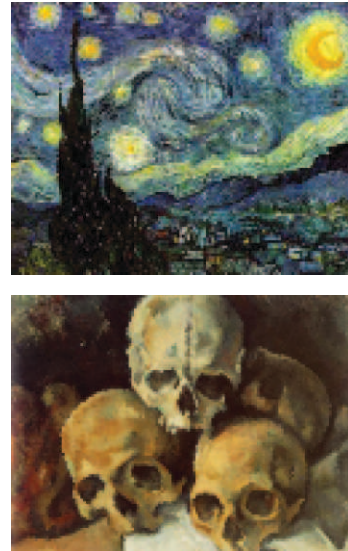
None of the systems described so far address the color transfer problem from an artist’s point of view. Meier et al [MSK04] describe a suite of interactive software tools that integrate historical, expert, and theoretical knowledge to help organize color palettes that can be used for image re-coloring. Some of their internal palettes were extracted *by hand* from digital versions of artist’s masterpieces.

Motivated by the color transfer problem for non-photorealistic images, as well as color transfer for artistic effect, we consider the color transfer problem from a palette extraction point view. That is, we consider the problem of automatically extracting a palette from a non-photorealistic image in such a way that it can be used to reconstruct the image degradation free while simultaneously capturing the essence of its color aesthetics. As an application, we use a naive palette color correspondence algorithm to examine the aesthetic results when color is transferred between some well-known fine art non-photorealistic images. The broad outline of our color transfer approach is as follows.

- Segment the image based on color.
- Form palette and identify each image segment with a palette color.
- Form color pairings between source and destination palettes.
- Transfer chromatic content from source to destination.

We begin by surveying some of the techniques first described in [GH03]. To help illustrate our methods we will consider

the problem of color transfer from Van Gogh’s *Starry Night* to Cezanne’s *Skulls*. Thumbnails of these two images are shown in Figure 1. Detailed pseudocode for all of the critical modules plus implementation details of the data structures and color spaces we use are available in our technical report available at <http://www-viz.tamu.edu/faculty/house/papers/02palette-techreport.pdf>.

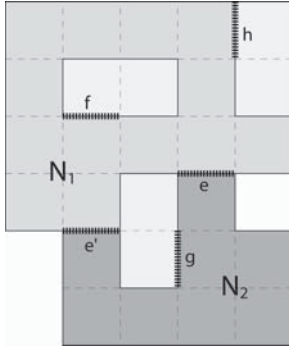


**Figure 1:** Thumbnails of test Van Gogh and Cezanne images.

## 2. Color Segmentation

Since color image segmentation is a complex subject possessing a voluminous literature, even a cursory treatment is beyond our scope. In this section we describe a slow, memory intensive hybrid color segmentation algorithm that we developed for research purposes in order to investigate the ramifications of trying to introduce value structure considerations during color segmentation while simultaneously maintaining full control over the number of segments in the segmentation. Our method is based on a bottom-up region growing algorithm. By using the pixel representation scheme of [BK91], we are able to maintain statistics on the area, boundary length, and average  $l\alpha\beta$  color of each region. Moreover, by organizing region merge events into a binary tree and using a recursive tree traversal algorithm, we are able to losslessly reconstruct the original image from its color segmentation and merge-event tree. Merge events are triggered by edge priorities that are calculated on the basis of color *differences* between adjacent regions. Edge priorities must be constantly updated during the region merging process. Figure 2 hints at the difficulty of edge priority updating by showing how a merge event combining regions  $N_1$  and  $N_2$  triggered by edge  $e$  affects the priorities of edges marked  $e^l$ ,  $f$ ,  $g$ , and  $h$ . The calculation of edge priority depends on

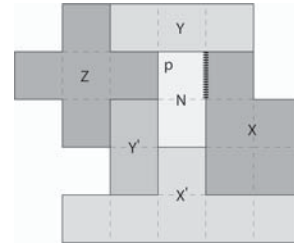
the color space used. From an artist's point of view, it would be helpful if such calculations could be done in a color space such as *HSV* or *HWB*. However, *digital* images are prone to color artifacts that render the hue channel essentially useless for certain ranges of the other channels in such spaces. Although it may not be the optimal choice, based on the results of [RAGS01], we used Euclidean distance in  $\ell\alpha\beta$  space for the transfer results shown in this paper.



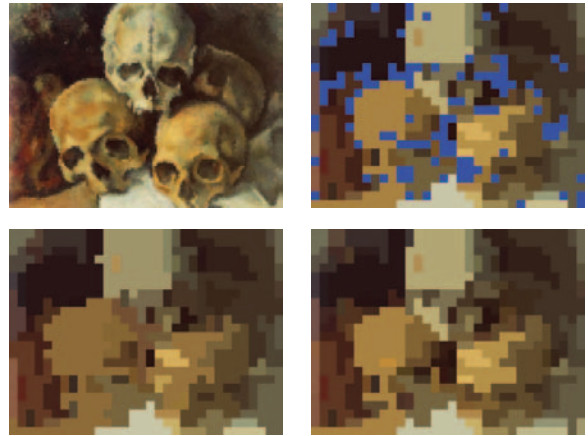
**Figure 2:** Edge priority update consequences of region merge event triggered by edge  $e$  during segmentation.

A further consequence of color artifacts in digital images is that if one tries to color segment until a pre-specified number of regions is obtained, then because of the existence of outlier or “accent” pixels, one runs the risk of winding up with one large region accounting for upwards of eighty percent of the image. In such cases the segmentation corrupts the image *composition*, something we clearly wish to avoid. This phenomenon is especially acute for the kind of non-photorealistic images we wish to consider. If we want to segment until we obtain  $n$  regions, one way to cope with this difficulty is to segment until we obtain  $n + k$  regions, then interrupt the segmentation and force the next  $k$  merge events to involve only the smallest remaining regions. As Figure 3 shows, given region  $N$  this means perhaps a decision must be made about whether to merge  $N$  with one of the two larger regions (labelled  $X$  and  $X'$ ) or one of the two smaller regions (labelled  $Y$  and  $Y'$ ). Moreover, as Figure 4 shows, regardless of how such decisions are made, there may be further consequences caused by demanding that the segmentation be reproducible by requiring, for example, that one consider the smallest regions in scanline order. Thus even though it makes artistic sense to organize color compositions by region merging and color averaging, the available digital infrastructure may not prove wholly suitable to the task.

In fact, execution time constraints and memory constraints caused by segmentations yielding too many regions with large areas lead us to segment using a truncated *image pyramid*, a multi-resolution technique that has proved successful in texture synthesis [HB95]. Segmenting the top layer of such a pyramid simplified palette extraction, but as



**Figure 3:** The quagmire of over whether to force the merge of region  $N$  with  $X$ ,  $X'$ ,  $Y$ ,  $Y'$ , or  $Z$ .



**Figure 4:** Segmentation of downsampled image (upper left) is interrupted and to reveal unmerged pixels shown in blue (upper right) that are then force merged with either largest adjacent regions (lower left) or smallest adjacent regions (lower right) to demonstrate how segmentation artifacts may occur.

will be seen below, led to difficulties in downsampling and maintaining global color continuity during color transfer.

### 3. Palette Extraction

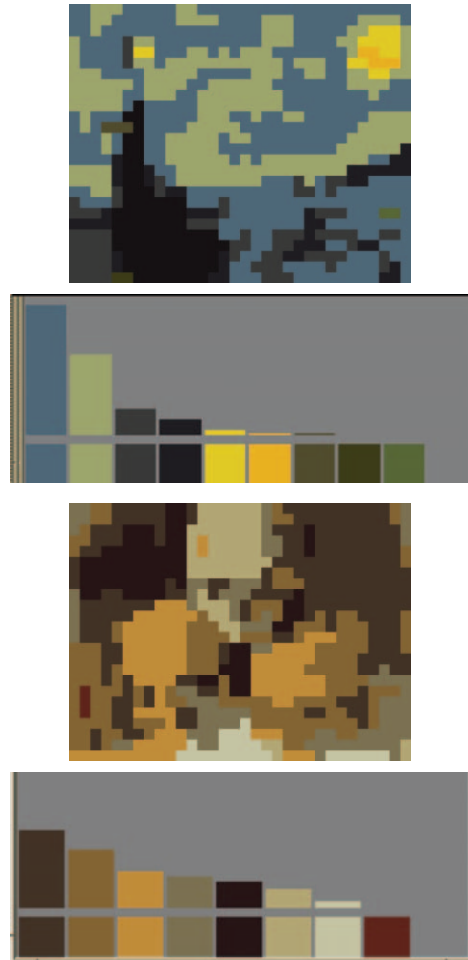
Due to the nature of its intrinsic topology, an image color segmentation typically contains many regions with virtually identical colors. Since the  $\ell\alpha\beta$  color space is logarithmic, makes too fine a distinction between dark colors, we defined colors from two regions to be identical, for our purposes, provided their Euclidean distance in RGB space was sufficiently small. By implementing a straightforward partitioning algorithm based on this idea, we were able to reduce a set of  $n$  *segment* colors to a much smaller set of  $d$  *distinct* colors by choosing as the color representative from the resulting equivalence class the averaged color of the largest region. This lowered the number of colors under consideration from  $n = 159$  to  $d = 22$  for *Starry Night*, and from  $n = 119$  to  $d = 13$  for *Skulls*. The reason these images have different values for the number of segments  $n$  is because even though

the number of forced merge events  $k$  is the same for both images, in order to take into account differences in image complexity, the value of  $n + k$  where the segmentation process is interrupted was determined by setting an edge priority threshold.

Visual inspection of the  $d$  colors still under consideration indicated that usually several *shades* of the desired colors were still present. In order to make the palette color associations needed for color transfer, it was necessary to further reduce the set of colors by identifying color shades. This proved to be a surprisingly vexing problem. Because the value structure of our images was important to us, and because no color space we tried was uniformly successful at shade clustering, we settled upon an interactive shade clustering algorithm that used HSV space to help cluster the shades of the most vivid colors and  $\ell\alpha\beta$  space to help cluster the shades of the darkest colors. More precisely, after sorting the distinct colors by saturation, we let each unused color be a representative for the shades found in a “wedge” around that color in HSV space. Next, we repeated the process in  $\ell\alpha\beta$  space defining the neighborhoods about each unused color using a thin “slab”. This provided us with palettes of size  $p = 10$  colors for *Starry night* and size  $p = 8$  colors for *Skulls* as shown in Figure 5. A “true black” chip is suppressed from the *Starry Night* palette for reasons which will be explained later. It should be pointed out that because our algorithms are non-interactive and deterministic, our extracted palettes can be thought of as *feature vectors* for our images [JFS95]. One of the reasons for using the test images we chose is the presence of dark and neutral colors that pose a challenge to algorithms that attempt to distinguish color shades that are distinct from perceptual black and white, “colors” that eventually required special considerations during our color transfer process.

#### 4. Palette Color Associations

Although one of our primary motivations was to consider palette color associations that would give rise to re-colorings that were faithful to the artistic intent and style of the source image and therefore transfer as much of its aesthetic content as possible, for testing purposes we adopted a naive, elementary approach. Note that for color transfer to make sense, every color in the destination palette must be paired with some color from the source palette, but not all colors from the source palette must be members of such pairings. Our palette color associations are based on a heuristic that maps colors used most often to each other. Details are given in [GH03]. Here it suffices to say that the two palette colors responsible for the largest areas (i.e. most prominent colors) are always paired, the two palette colors responsible for the second largest areas are *encouraged* to be paired, and the remaining pairings are determined by alignments of the palette color vectors after they have been normalized using a procedure that was inspired by the color correction meth-



**Figure 5:** Segmented test images located at the apexes of the image pyramids are shown with their segments colored by their associated palette colors and are accompanied by their palettes — palette color “chips” together with a bar chart to show the percentage of the image area that each color is assigned to.

ods presented in [RAGS01]. Figure 6 shows the pairings for transferring color from *Starry Night* to *Skulls*. Unfortunately, because the vivid yellow and gold of the *Starry Night* palette were seen as outliers by our pairing algorithm, they did not participate in this re-coloring.

#### 5. Color Transfer Implementation

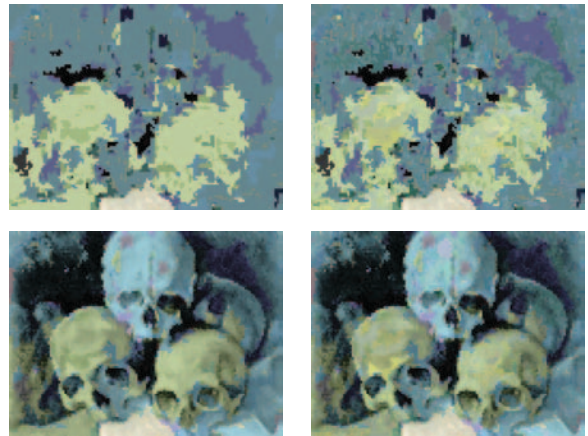
Image segmentation palette extraction depends on averaged color. In order to implement image color transfer, we must transfer chromatic content from source image to destination image on a pixel basis. Each destination pixel is the leaf of a path in the destination merge-event tree that leads from that pixel to a region in the segmentation. That region has



**Figure 6:** Re-coloring plus palette pairings for a  $128 \times 128$  pixel version of *Skulls* using the test *Starry Night* image as source image. The top row of the pairings shows the palette colors extracted from *Skulls*. The middle row shows the “corrected” palette colors. The bottom row indicates the pairings with *Starry Night* palette colors.

an associated destination palette color which, in turn, has been paired with a source palette color. Our task is to find a corresponding source pixel that has a path in the source merge-event tree that leads to a region in the segmentation that is associated with that source palette color. Space prohibits presenting the details of how we accomplish such “path matching,” but Figure 7 shows why it is important. The more source pixels that participate in the color transfer, the better the chances are that instances of visually rich color texture in the source image will be transferred to the destination image.

The fact that we work within an image pyramid means that multiple image color transfers take place, one for each layer of the pyramid. Thus if we re-color a  $128 \times 128$  image using a three layer pyramid, our system actually yields a re-coloring of the original  $128 \times 128$  image plus re-colorings of *downsampled*  $64 \times 64$  and  $32 \times 32$  versions of the image. This brings up several new issues to discuss since the re-coloring algorithms described above only govern what occurs in the very top layer of the pyramid. Lower layers of the pyramid are re-colored piecemeal, by applying our re-coloring algorithm exactly as described above each of the segmentations of the “pieces” in the layer that project onto the segments in the layer immediately above, with the understanding that forced merge-events are now prohibited by setting  $k = 0$  when the segmenting of each piece is done. While the heuristic is that this should enable us to *refine* the re-coloring of the layer above as additional color information is revealed due to segmenting the layer below, in practice we are often defeated by downsampling artifacts on one hand when new and unexpected colors are encountered, and shading problems on the other hand when extracted palette rep-



**Figure 7:** The top row shows color transfer results using our test images when all three of the  $\alpha\beta$  channels are transferred from source to destination. The bottom row shows the improved value-preserving color transfer that results when only the  $\alpha$  and  $\beta$  channels are transferred. The left column shows the results of color transfer when only one source pixel per source region participates in the color transfer. The right column shows the improved results of color transfer when path matching is invoked to allow more than one source pixels per region to participate.

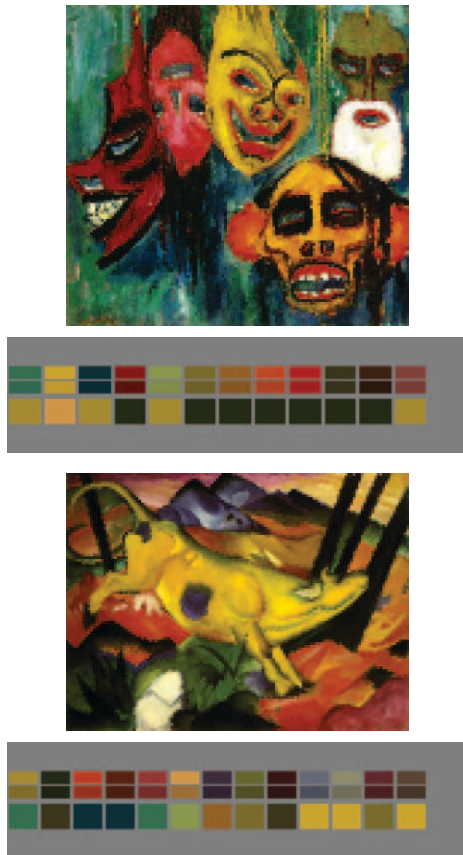
resentatives arise that wind up being treated differently than they were in preceding layers. To help address the former problem, we removed those palette colors from the *source* palette that were responsible for the least amount of area in the segmentation. In the absence of such palette pruning, a re-coloring of an image with itself would faithfully reproduce the image. With such pruning, we obtain a lossy version of the image such as the one shown in Figure 8. We have no solution to the shading problem. It explains why the two nearly white areas at the bottom of the test *Skulls* image appeared re-colored differently in the re-coloring of Figure 6. When treated as two separate pieces, one was assigned a palette color that was considered to be “true white” hence was not re-colored for reasons explained below, while the other was re-colored from source pixels.



**Figure 8:** A re-coloring of the test *Starry Night* image with itself to show the extent to which pruning palette colors in the source image leads to a lossy reconstruction of the image.

Using an image pyramid also leads to color-transfer robustness problems. To see this, compare the  $128 \times 128$  low-resolution re-coloring of the thumbnail of *Skulls* in Figure 6 that was obtained using a palette extracted from a downsampled  $32 \times 32$  version of that image with the  $512 \times 512$  high-resolution re-coloring in Figure 11 that was obtained using a palette extracted from a downsampled  $128 \times 128$  version of that image.

Another artifact of downsampling is the treatment that boundary pixels of image features receive. Figure 9 shows the palette extractions from *Yellow Cow* by Franz Marc and *Mask Still Life III* by Emil Nolde together with the palette color pairings for a bi-directional re-coloring. The recolorings are shown in Figure 12. One can observe many incorrectly re-colored pixels that arose when their projections wound up in inappropriate segments.



**Figure 9:** Extracted palettes and palette pairings for a bi-directional re-coloring between Franz Marc's *Yellow Cow* and Emil Nolde's *Mask Still Life III*.

## 6. Color Correction

For our non-photorealistic images, trying to transfer only the  $\alpha$  and  $\beta$  chroma channels from source pixels to destination

pixels led us to confront several additional problems. One particularly acute problem was trying to transfer chroma from dark source colors to light destination colors because more often than not, the resulting color was out-of-gamut. This we partially addressed by attenuating channel values during color transfer in order to help lessen the chances of oversaturation. Regardless of whether a color transfer was from light to dark, or vice versa, it was always the case that our final conversion from  $\ell\alpha\beta$  color space to RGB color space left many destination pixels out-of-gamut. This we "fixed" by first applying a global image correction factor in RGB color space based on the ninety-fifth percentiles of each the R, G, and B values and then clamping the out-of-gamut pixels in any remaining "hot spots." It was primarily because we could never settle upon a suitable color space to perform all of our computations and still remain in gamut that we adopted the measure of excluding "true whites" and "true blacks" from our palettes. This meant that non-photorealistic images with color gradients involving white, black, and other neutral colors were usually not suitable candidates for image color transfer. This fact, together with the fact that *Skulls* uses a limited range of hues in its palette while *Starry Night* has a much broader range of hues, also helps to explain why we were never able achieve our goal of obtaining a bi-directional re-coloring using both of those images.

There is one final feature to discuss that also impacts the out-of-gamut question. It is motivated by considering value-structure techniques that artists are taught. If an artist adds a new color to the palette at a later stage of a painting's completion, then much of the existing composition often must be re-worked in order for that color to be properly integrated into the final result. In the same vein, prior to color transfer, we pre-conditioned our *destination* images by invoking an algorithm to adjust the  $\ell$  channel values so that the resulting histogram closely matched the  $\ell$  channel histogram of the source image. This explains how the colors labelled "corrected" palette colors in the middle rows of our palette-pairing figures were obtained. Figure 10 shows the result of pre-conditioning Fragonard's well-known *Young Girl Reading* and an early, but obscure, painting by Kandinsky for potential bi-directional re-coloring. Since pre-conditioning renders the Fragonard virtually useless, it is easy to see why only the Kandinsky re-coloring shown in Figure 11 was reasonably successful.

## 7. Conclusions

We have shown that deciding how to extract a palette from a digital image, make palette associations between two images, and implement color re-targeting given palette associations are three very difficult problems, problems that are probably best treated separately. Further, we have shown that deciding whether it is best to extract a palette from a *non-photorealistic* image by using color histogram models such



**Figure 10:** Top Row: Fragonard's Young Girl Reading and an early painting of Kandinsky. Bottom Row: Pre-conditioned versions of these two images showing value adjustments made prior to bi-directional color transfer.

as [PNS03] or [WSM99], absolute color naming systems or, as we have done, color segmentations deserves further study.

The method we presented here for color transfer has several strengths and weaknesses with important implications for future work. Our method's principal strengths are ideological. First, it is based on a model that makes artistic and aesthetic sense — reducing an image to a palette — therefore it is well motivated. Second, it is top down resulting in color transfer proceeding from coarse to fine which again re-inforces our intuition of how color transfer process should physically take place. Third, by employing multiple color systems, it is able to more clearly identify and tackle a broader range of luminosity, hue, and perceptual color issues. Thus our work clarifies the difficulty of using existing *computational* color models to solve *perceptual* color problems that arise during color transfer. Our method's principal weaknesses are technical. First, because of the many disparate ways we chose to handle color our method requires a large number of parameters, and while all our figures were made using the same settings for these parameters, questions regarding how widely applicable our settings are or how best to tune our settings remain unanswered. Second, because of the hybrid nature of our color segmentation algorithm, our

color transfer is slow, requiring hours instead of minutes for  $512 \times 512$  images. Third, because we use an image pyramid, local-global color consistency and coherency issues arise during color transfer. Thus future work will need to re-examine the use of segmentation, consider how to streamline color analysis, and find better ways to take into consideration the global color relationships within an image.

## References

- [BK91] BIERI H., KOHLER A.: Computing the area, the circumference, and the genus of a binary digital image. In *Graphics Gems II* (1991), Arvo J., (Ed.), Academic Press, pp. 107–111.
- [CS02] CAI Y., SIEGEL M.: Texture characterization of visual appearance of paintings. In *Proceedings of IEEE Instrumentation and Measurement Technology Conference* (2002). Proc. IMIC '02.
- [CSN02] CHANG Y., SAITO S., NAKAJIMA M.: Color transformation based on the basic color categories of a painting. In *Conference Abstracts and Applications* (2002), ACM SIGGRAPH, p. 157. (SIGGRAPH '02).
- [CSN03] CHANG Y., SAITO S., NAKAJIMA M.: Color transfer between images based on basic color category. *IEICE Transaction on Information and Science* 86, 12 (2003), 2780–2785.
- [Cur02] CURRAN A.: Applying math to Monet. *Carnegie Mellon Magazine*, Fall (2002), 17.
- [CUSN04] CHANG Y., UCHIKAWA K., SAITO S., NAKAJIMA M.: Example-based color stylization based on categorical perception. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization* (2004), ACM Press, pp. 91–98.
- [GD05] GRUNDLAND M., DODGSON N.: Color histogram specification by color warping. In *Color Imaging X: Processing, Hardcopy, and Applications* (2005), vol. 5667, International Society for Optical Engineering. (Proc. SPIE '05) (to appear).
- [GG01] GOOCH B., GOOCH A.: *Non-Photorealistic Rendering*. A. K. Peters, 2001.
- [GH03] GREENFIELD G., HOUSE D.: Image recoloring induced by palette color associations. *Journal of WSCG* 11, 1 (2003), 189–196.
- [HB95] HEEGER D., BERGEN J.: Pyramid-based texture analysis/synthesis. In *ACM Computer Graphics* (1995), pp. 229–238. (Proc. SIGGRAPH '95).
- [HJO\*01] HERTZMANN A., JACOBS C., OLIVER N., CURLESS B., SALESIN D.: Image analogies. In *ACM Computer Graphics* (2001), pp. 327–340. (Proc. SIGGRAPH '01).
- [JFS95] JACOBS C., FINKELSTEIN A., SALESIN D.: Fast multiresolution image querying. In *ACM Computer Graphics* (1995), pp. 277–286. (Proc. SIGGRAPH '95).



**Figure 11:** High resolution re-colorings of a Cezanne by a Van Gogh and a Kandinsky by a Fragonard.



**Figure 12:** High resolution bi-directional re-coloring of Marc with Nolde.

[LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. In *ACM Computer Graphics* (2004), pp. 689–694. (Proc. SIGGRAPH '04) <http://www.cs.huji.ac.il/~yweiss/Colorization/>.

[MS02] MOROVIC J., SUN P.: Transforming 3D colour

histograms of images. In *First European Conference on Colour in Graphics, Imaging and Vision* (2002), pp. 104–108. (Proc. CGIV 2002).

[MS03] MOROVIC J., SUN P.: Accurate 3D image colour histogram transformation. *Pattern Recognition Letters* 24,



11 (2003), 1725–1735.

- [MSK04] MEIER B., SPALTER A., KARELITZ D.: Interactive color palette tools. *IEEE Computer Graphics and Applications* 24, 3 (2004), 64–72.
- [PNS03] PICHON E., NIETHAMMER M., SHAPIRO G.: Color histogram equalization through mesh deformation. In *Proceedings of the International Conference on Image Processing* (2003), vol. 2, pp. 117–120. (Proceedings ICIP 2003).
- [RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications* 21, 5 (2001), 34–41.
- [RCC01] RUDERMAN D., CRONIN T., CHIAO C.: Statistics of cone responses to natural images: implications for visual coding. *J. Optical Soc. of America* 15, 8 (2001), 2036–2045.
- [RGW05] RASCHE K., GEIST R., WESTALL J.: Recoloring images for gamuts of lower dimension. *Computer Graphics Forum* (2005). (Proc. Eurographics '05) (in press).
- [WAM02] WELSH T., ASHIKHMIN M., MUELLER K.: Transferring color to grayscale images. In *ACM Computer Graphics* (2002), pp. 277–280. (Proc. SIGGRAPH '02).
- [WSM99] WEEKS A., SARTOR L., MYLER H.: Histogram specification of 24-bit color images in the color difference (C-Y) color space. In *Nonlinear Image Processing X* (1999), vol. 3646, pp. 319–329. (Proceedings SPIE).