

Data-driven Color Manifolds

CHUONG H. NGUYEN, TOBIAS RITSCHSEL
MPI Informatik and Saarland University / MMCI
HANS-PETER SEIDEL
MPI Informatik

Color selection is required in many computer graphics applications, but can be tedious, as 1D or 2D user interfaces are employed to navigate in a 3D color space. Until now the problem was considered a question of designing general color spaces with meaningful, e. g., perceptual, parameters. In this work, we show, how color selection usability improves by applying 1D or 2D color manifolds which predict the most likely change of color in a specific context. A typical use case is manipulating the color of a banana: instead of presenting a 2D+1D RGB, CIE Lab or HSV widget, our approach presents a simple 1D slider that captures the most likely change for this context. Technically, for each context we learn a lower-dimensional manifold with varying density from labeled Internet examples. We demonstrate the increase in task performance of color selection in a user study.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Color*

General Terms: Algorithms, Design

Additional Key Words and Phrases: color pickers, data-driven methods, manifolds

ACM Reference Format:

Nguyen, C. H., Ritschel, T., and Seidel, H. P. 2015. Data-driven color manifolds. *ACM Trans. Graph.* XX, X, Article XXX (2015), 9 pages.
DOI =xxx
<http://doi.acm.org/xxx>

1. INTRODUCTION

The seemingly simple task of color selection is highly important in many computer graphics applications, ranging from casual photo manipulation to professional 2D and 3D content creation. Despite being an often-used and important operation, exploring high-dimensional colors using low-dimensional user interfaces such as

linear or angular 1D slider or 2D widgets that control parameters of a certain color space [Schwarz et al. 1987; Douglas and Kirkpatrick 1999] is often disappointing. Color selection using color templates (e. g., PANTONE), or more specific, hue templates [O'Donovan et al. 2011], is an alternative but does not scale well to many colors (too large palettes) and fine-tuning of colors (too coarse palettes).

In this work, we seek to improve upon both paradigms in an example scenario as follows: A user takes a picture of a wooden chair and wants to adjust its color using a single 1D sweep or a single click on a mobile device. In any common color space, such a change is likely impossible: Going more red will require more saturation, going less red, will require less, otherwise the result is a color, but not a wood color anymore. Our approach learns this relationship from labeled exemplar data and present the user a single 1D slider to traverse the manifold of plausible wood colors. Additionally, we want the manifold to compress the color space in ranges that appear less frequently and to enlarge in frequently used areas. For the chair examples, we will enlarge the brown-beige-red areas while we shrink the green or blue areas.

The technical challenge addressed in this paper is to project a high-dimensional color space with a density acquired from Internet data, to a lower-dimensional color space such that neighborhood of important colors is preserved, and embedded area is proportional to density.

2. PREVIOUS WORK

Color Spaces. A vast choice of different physical color spaces such as sRGB, perceptual color spaces such as CIE Lab and CIE XYZ or color appearance models such as CIE CAM [Fairchild 2005] were proposed over the years. The discussion of what space or what model is best for which purpose is extensive and will not concern us here as it is mostly orthogonal to our approach to color selection, where we are only interested in traversing an existing high-dimensional color space in a meaningful low-dimensional way.

The idea of extracting statistics of colors has been used in computer vision. An example is Hsu et al. [2002], who proposed a specialized color space for human skin to be used for face detection. They use principle component analysis (PCA), which implies that the best manipulation happens along a particular linear direction in RGB with equally-sized steps. Image-dependent PCA has also been used to improve compression of color images [Clausen and Wechsler 2000]. For a general survey of dimension-reduction techniques in color science, where they are mostly applied to reducing high-dimensional spectral signals to low-dimensional spaces, see by Tzeng and Berns [2005]. We extend the aforementioned ideas of linear, PCA-based color spaces to arbitrary embeddings (i. e., 1D curves, or 2D surfaces) of varying density based on image statistics acquired from the Internet. The color science community has addressed the deformation of space to fit to certain data for problems of linearization in agreement to some measurement, e. g., hue [Lissner and Urban 2009]. Omer and Werman [2004] use a set of 1D-subsets of a color space (lines) to detect and reduce distortions

Authors' addresses: C. H. Nguyen, T. Ritschel and H-P. Seidel, Max-Planck Institut Informatik and Saarland University, Saarbrücken, Germany; email: nguyen@mpi-inf.mpg.de, ritschel@mpi-inf.mpg.de.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 0730-0301/YYYY/15-ARTXXX \$10.00

DOI 10.1145/XXXXXXX.YYYYYYY

<http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY>

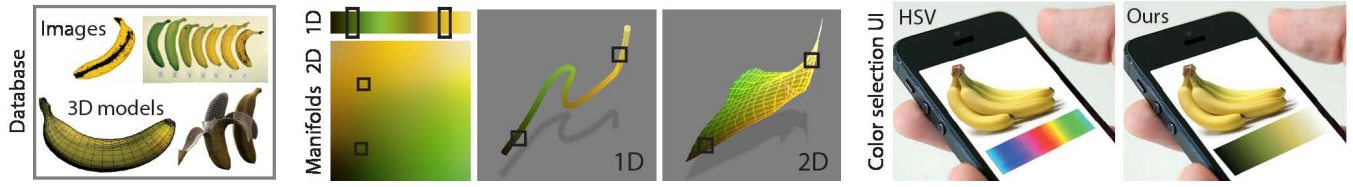


Fig. 1. Our approach uses Internet image collections (1st col., only subset shown) to learn color manifolds. The 1D manifold for the class “Banana” with a single degree of freedom (2nd col., top) and as a patch for two degrees of freedom (2nd col., bottom). The same manifold shown as a 1D line (3rd col.) or a 2D patch (4th col.) in 3D color spaces. The same two colors are marked as squares in all visualizations. A key application is user interfaces where the manifold (here 1D) is used as a slider which show only the appropriate colors (6th col.) instead of all colors as in common color pickers (5th col.).

of colors in acquisition and reproduction of images. They do not extract a single 1D, 2D or 3D manifold, but multiple disconnected 1D lines and do not account for varying (perceptual) density. While multiple disconnected lines can serve as a regularization to restrict the set of colors to plausible ones, they do not allow for an intuitive user interface as there is no obvious way how to embed a set of disconnected lines into a single slider, such as done in this work. Finally, they do not capture two-dimensional relationships.

Lalonde and Efros [2007] have analyzed color distributions of typical classes of images. A distance measure between distributions can be used to measure color compatibility of one image to a certain class or the compatibility of a foreground and a background. However, the distribution remains “high-dimensional” (3D), while we seek to find an embedding in fewer dimensions. Oskam et al. [2012] address the problem of global color balancing between images using a sparse set of desired color correspondences by deforming the color space. We also deform the color space but to the end of embedding and flattening it to a lower-dimensional space.

Color selection. As a human-computer interaction, color selection has received only little attention with the exception of work by Schwarz et al. [1987] as well as by Douglas and Kirkpatrick [1999]. They found that the choice of color space has only little impact on performance when comparing different color spaces and visual feedback is the most important usability factor. We perform a similar study, including our color manifolds. The objective of our embedding is to provide as much 3D color space feedback as a 2D screen can contain. Different from us, Shapira et al. [2009] present an exploratory interface to edit image appearance interactively. Their approach is also concerned with modeling the distribution of colors, but not by embedding them in lower dimensions. Instead, color distribution is modeled as mixture of Gaussians and used for manipulation. A user can visually navigate the high-dimensional space of possible color manipulations by transforming the Gaussian mixtures. Each pixel’s color follows its distribution accordingly. Our color manifolds could be used in their framework as additional constraints that restrict object color to remain valid (e. g., plausible human skin color).

Color templates and themes. The relation of images to colors, called “Image Themes” is extracted, transferred and enhanced in the works of Wang et al. [2010; 2011]. We share the idea of acquiring image statistics from Internet data, but for a different purpose.

Color templates are a pre-defined discrete selection of colors. Popular Internet sites such as Adobe Kuler provide a large collection of such color templates. Templates are well-suited for picking a combination of colors, but less suited to fine adjustment of colors. Hue templates were studied by Matsuda [1995] and later used for color harmonization [Cohen-Or et al. 2006]. A perceptual study of such color templates was conducted by O’Donovan et al. [2011].

Manifolds. In general, extraction of lower-dimensional embeddings is applicable to many computer graphics problems. Multi-dimensional-scaling (MDS), as used in the Design Galleries framework [Marks et al. 1997], allows for embedding higher-dimensional qualia into lower-dimensional layouts. Matusik et al. [2003] sample the even higher-dimensional space of BRDFs to create a neighborhood graph that allows to move to nearby plausible BRDFs. Our work is different in two regards. First, the comparatively low-dimensionality of colors allows us to fit a parametric model with an explicit dimensionality to the color distribution resulting in a smooth and continuous manipulation and layout of the color structure. Second, we have a density measure defined on the high-dimensional color data that is preserved as local area changes (Jacobian) of the embedding. Appearance manifolds have been widely used in computer vision and computer graphics community. Wang et al. [2006] build appearance manifolds to capture time-variant appearance of materials from data captured at a single instant in time. Xue et al. [2008] model the reflectance of weathered surfaces from a single input image as a manifold and use it for interactive editing of the weathering effects in an image.

3. OUR APPROACH

We will now describe our approach to create an n -dimensional color manifold from images of a certain class represented in a $m \geq n$ -dimensional color space. We experiment with values of $n = 1$ (line) or $n = 2$ (surface) and $m = 3$ (radiance, reflectance).

In future work, higher values of m could be used for spaces like spectral color; higher values of n in other applications or 3D color selection. Every manifold belongs to an image class like “sky”, “banana”, etc. We will not consider how to classify images and assume a state of the art-classifier to be correct in many cases [Lazebnik et al. 2006] in combination with a simple but effective background removal. The extraction of all manifolds for each class is independent and performed for all possible classes in a pre-process step. As the manifolds are smooth, they can be serialized into a file of a few hundred bytes. Exchanging those files is sufficient for standardization, e. g., in print. As our manifold construction can be performed at interactive rates, we optionally allow the user to create a set of images and extract their color manifold on-the-fly.

Overview. An overview of our approach for one class and manifold is given in Fig. 2. First, we acquire color samples from images or 3D models returned from an Internet query (Sec. 3.1). Typically, this results in many millions of colors that are put into a m -dimensional histogram in a certain high-dimensional *source* color space such as RGB or CIE Lab (Sec. 3.2). Next, this histogram is thresholded and only the α % most important colors are kept. On the remaining colors, we either use principal component analysis

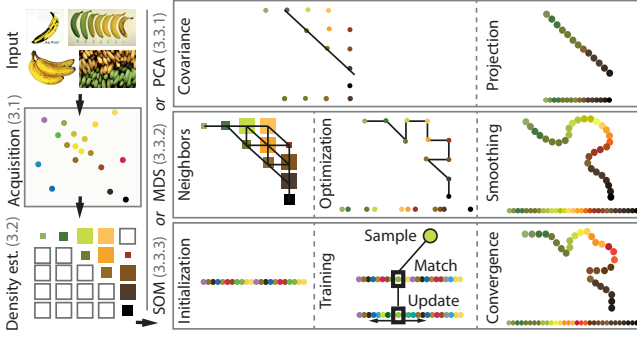


Fig. 2. Flow of our approach: The first two steps are always identical (left): Acquisition of color statistics (Sec. 3.1) and density estimation (Sec. 3.2). The dimensionality reduction is analyzed for three variants (right): PCA (Sec. 3.3.1), MDS (Sec. 3.3.2) or SOM (Sec. 3.3.3).

(PCA), multi-dimensional scaling (MDS) or self-organizing maps (SOM) to perform dimensionality reduction (Sec. 3.3).

3.1 Acquisition

Color samples are acquired using 2D Internet image search and online 3D model repositories. For images, Google Image Search is used to acquire the top 100 images for one class. Note that our input image set contains images with improper camera calibration (improper photometric calibration, incorrect white balance) and different drawing styles. We do not filter this set or compensate for any different camera calibration or image style, which would likely further improve our results. We will distinguish in our results between *weakly supervised* acquisition (where images are used as they return from an Internet search) and *strongly supervised* acquisition (invalid pixels are manually excluded by an alpha mask). Optionally, users can *interactively* remove and add images. If not mentioned otherwise, all of our results are from weakly supervised sources without user interaction.

The problem with 2D images is that they are acquired under unknown illumination and contain shading. Consequently, images can be used better to study a luminance-free, 2D color space of hue and saturation. Ideally, we would like to have a repository of true reflectance data for the purpose of studying 3D color. For this, data from online 3D model repositories are used. We assume that textures of such models do indeed have reflectance of 3D models. This is justified, as artists tend to use proper white balancing, shadow removal, etc. on their textures. The only remaining difficulty is that textures contain areas with pixel values that do not map to the surface and should be excluded as they are not part of the reflectance we seek to sample. We solve this by setting alpha to zero in all textures and then draw the UV mapping polygons with alpha set to one into the alpha channel. The result is a 2D RGBA image that can be processed like other images. Unless stated otherwise, 2D images are used to study the full 3D color space and for results in this paper, we use manifolds of radiance and not of reflectance.

For a certain class, image search results contain a foreground object belonging to the class in front of a background. We use a simple heuristic to remove this background. First, the image is blurred using Bilateral filtering. Next, two neighbor pixels are connected only if their CIEDE2000 color difference is less than a threshold. Finally, all connected components that contain pixels on the image frame boundary are considered as background. Fig. 3 shows several background-removed images. Our removal tends to be conservative and while it potentially removes pixels that belong to the class, it rarely keeps background pixels, as seen e. g., around the “Snow White” example.

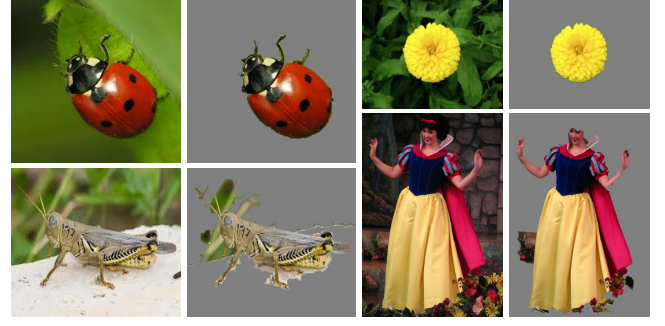


Fig. 3. Input images before (1st and 3rd col.) and after background removal where removed pixels are marked as gray (2nd and 4th col.). Input images courtesy of Dominik Stodulski (lady bug), Greg Lasley (grass hopper) and Flickr user Loren Javier (Snow White).

The acquired images are now in the RGB color space, with alpha channel used as a mask. Pixels with an alpha value of zero are skipped from further consideration. If the source color space $S = \mathbb{R}^m$ should be different, a conversion e. g., to CIE Lab is performed now. Finally, we randomly sample k (e. g., four million) pixels from the m -D color space of our query data into a vector $\mathbf{a} \in S^k$.

3.2 Density estimation

The purpose of this step is to turn a large collection of samples that implicitly describe the frequency of a certain color into a simple explicit representation of density, i. e., color frequency. To this end, the source space is conceptually covered by a lattice of 16 bins along each dimension, into which the k sampled pixels of \mathbf{a} are inserted. We call the normalized bin value the *density* of this color. We choose a regular grid instead of clustering for simplicity and to allow for an efficient (i. e., interactive-rate) construction of manifolds with a user in the loop. Next, this histogram is thresholded and only the α % of bins with the highest density are kept. An $\alpha = 15$ % is used to produce our results. This is done to eliminate outliers that corrupt the manifold structure or colors that might not belong to the class. See the supplemental material for more details on our parameter choice. The result is a set $\mathbf{b} \in S^l$ of l points in the m -D source color space with a density vector $\mathbf{d} \in \mathbb{R}^l$. We clamp the density \mathbf{d} between $0.1 |\mathbf{d}| / l$ and $2 |\mathbf{d}| / l$ to avoid under- or over-estimated color importance. While the input contained k i. e., millions of elements, the compact color point cloud \mathbf{b} typically contains $l \ll k$ i. e., several hundreds of colors only.

3.3 Dimensionality reduction

We explore several dimensionality reduction methods for our embedded manifolds: linear PCA (Sec. 3.3.1), as well as MDS (Sec. 3.3.2) and SOM (Sec. 3.3.3) which are non-linear methods.

3.3.1 Principal component analysis (PCA). Let $\mu \in \mathbb{R}^m$ be the density-weighted mean of \mathbf{b} and $\mathbf{C} \in \mathbb{R}^{m \times m}$ the density-weighted covariance matrix of the color distribution. The n eigenvectors of \mathbf{C} with the highest eigenvalues are a linear embedding into the n -dimensional space. Our method embeds a line ($n = 1$) or plane ($n = 2$) in the m -dimensional source space with higher weight to high-density colors.

3.3.2 Multi-dimensional scaling (MDS). To embed colors using MDS, we establish color neighborhoods, optimize a layout to keep those neighborhoods and finally optimize for smoothly and

completely filling the lower-dimensional space. The three steps are explained in the next paragraphs.

To extract the manifold structure, we create a graph with weighted edges. First, all colors in \mathbf{b} are interpreted as nodes and an edge is created between neighboring colors / nodes. The neighborhood is still in the m -D source space and simple to find due to the regular histogram structure. For the rare case, when the resulting set of points has several disconnected components, we discard all but the largest component. Next, we label each edge with the average density $(\mathbf{d}_i + \mathbf{d}_j)/2$ of the two nodes i and j it connects. Conceptually, edges in dense areas get longer, edges in sparse areas get shorter. Finally, we approximate the geodesic distance between all pairs of nodes (not just the neighbors) by shortest paths using the Floyd-Warshall algorithm and insert it into a pairwise distance matrix $D \in \mathbb{R}^{l \times l}$. This step is similar to the extension of Isomap [Tenenbaum et al. 2000] over MDS, but using weighted edges instead.

The distance matrix D computed in the previous step is fed into a classic MDS [Cox and Cox 2000]. The output are l new n -dimensional color points $\mathbf{c} \in \mathcal{T}^l$ ($\mathcal{T} = \mathbb{R}^n$) that preserve the desired distance matrix D in the lower-dimensional space in the least-squares sense. As an MDS solution is unique up to a rotation and a uniform scaling, we additionally normalize it to the unit hypercube and rotate it, such that the direction of largest luminance variation (found using PCA of the luminance of \mathbf{c}) aligns with the first axis.

While \mathbf{b} was a regular grid of a simple structure in the host color space \mathbb{R}^m , the embedding \mathbf{c} is an irregular point cloud in \mathbb{R}^n . Therefore, it is not clear which, how many and if at all an element in \mathbf{c} maps to any coordinate location \mathbf{x} in \mathbb{R}^n . However, we would like to use \mathbb{R}^n for smooth navigation to enumerate \mathbb{R}^m in a plausible way. To reconstruct a smooth unique mapping $f \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ defined on the entire domain \mathbb{R}^n , we employ radial basis function (RBF) reconstruction: $f(\mathbf{x}) = \sum_{j=1}^l r(\mathbf{c}_j, \mathbf{x}) \mathbf{b}_j / \sum_{j=1}^l r(\mathbf{c}_j, \mathbf{x})$ where r is the kernel $r(\mathbf{x}, \mathbf{y}) = \exp(-(s_m \|\mathbf{x} - \mathbf{y}\|)^2)$ with a constant s_m to control smoothness. Please see the supplemental material for different settings of the smoothness parameter s_m .

3.3.3 Self-organizing map (SOM). SOM produce a non-linear mapping from the m -dimensional space to an n -dimensional grid of weight nodes $\mathbf{W} \in \mathcal{S}^{h^n}$ [Kohonen 1990], where h is the grid's size.

The nodes of \mathbf{W} are initialized to random values and updated in t training steps. In a training step i ($1 \leq i \leq t$), the node with the minimum distance to a training sample is called the best matching unit (BMU). The weight of the BMU and its neighbors are adjusted toward the training sample. The magnitude of change and the neighborhood size from the BMU decrease after each training step [Kohonen 1990]. Fig. 4 shows the weight grid \mathbf{W} during the construction of the “Bananas” manifold at different training steps.

Let $\mathbf{p} \in \mathcal{S}^{s \times t}$ be the training sample set where s is the number of training samples processed in each training step. Note that s equals to 1 in the previous paragraph. The training sample set \mathbf{p} is constructed to contain colors from the compact color point cloud \mathbf{b} with distribution proportional to the density vector \mathbf{d} .

In order to fully use the advantage of modern GPUs, at training step i , we process a set of s samples $\mathbf{p}_{(i-1)s+1}, \dots, \mathbf{p}_{is}$ in parallel as proposed in Batch SOM (Fig. 2 in [Lawrence et al. 1999]). The distances between the training samples and nodes are calculated using the CIEDE2000 color difference. Nodes are then updated as described in Lawrence et al [1999]. To enforce smoothness on the weight grid \mathbf{W} , we further constrain the neighborhood size to be bigger than a predefined smoothness parameter r_0 ($r_0 \in \mathbb{N}^+, 1 \leq r_0 \leq h/2$).

For the 2D manifolds ($n = 2$), we empirically set the grid size $h = 32$, $s = 64$ training samples each step, the smoothness

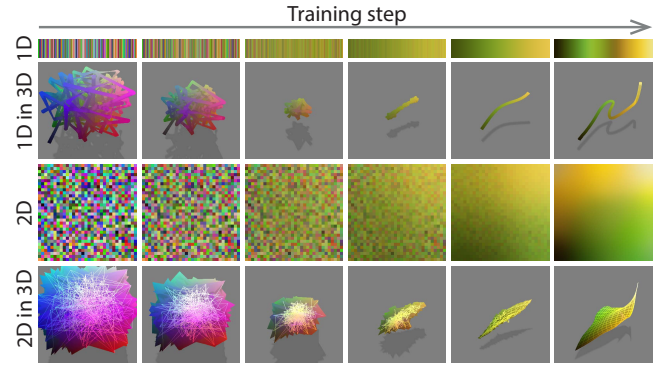


Fig. 4. Banana manifold construction (Fig. 1) using SOM: The first and third row show the colors in the evolving 1D and 2D manifold. The second and fourth row show the 1D and 2D color path resp. patch in the 3D RGB cube. From left to right, the manifold is refined as new samples are added.

parameter $r_0 = 10$ and $t = 2000$ training steps. For the 1D manifolds ($n = 1$), we set $h = 128$, $s = 64$, $r_0 = 15$ and $t = 4000$ respectively. Please see the supplemental material for more details about our parameter choices.

4. ALGORITHM EVALUATIONS

4.1 Algorithm Comparison

Fig. 5 shows several dimensionality-reduced color spaces produced using SOM, MDS and PCA respectively. PCA produces a line or plane in 3D that minimizes the variance of the color distribution \mathbf{b} . If the colors do not follow a plane or line, which is mostly the case, many colors can not be addressed. While simple to compute and easy to store, PCA-based color manifolds consistently perform worse, as also shown in our perceptual study. MDS preserves intrinsic (geodesic) distances of the color distribution \mathbf{b} and performs well if a 1D or 2D manifold exists. The bottom row of Fig. 5 shows a typical failure case of MDS where the distribution forms a cycle and geodesic distances fail to produce an embedding into a 1D or 2D disk. Such cyclic paths are rare in color distributions but do exist. SOM naturally handles cyclic distributions which can be an issue for MDS. The supplemental material provides more detailed results for different classes using different dimensionality reduction approaches.

4.2 User Study

We evaluated the usefulness of the proposed color manifolds in a color adjustment and a color exploration task.

In the color adjustment task, 13 subjects were asked to adjust the color of an image to match a reference image using different interfaces. Images used for manifold construction were excluded from the user study. In each trial, the participants were presented a color image and a second version of the same image that had a random color assigned to a soft region defined by the authors. Subjects were asked to adjust the color of the image back to the reference using a specific color picker UI. Adjustment was performed by picking a color inside the color picker and the image was modified by means of recoloring [Gastal and Oliveira 2011]. This objective is a typical color selection task in everyday photo editing. The UI was randomly either a common RGB, Lab, HSV, a “reduced” RGB, “reduced” Lab, “reduced” HSV where colors outside the color distribution of the image class are grayed out as un-selectable as well as our proposed 1D or 2D manifold interfaces, produced using either PCA, MDS or

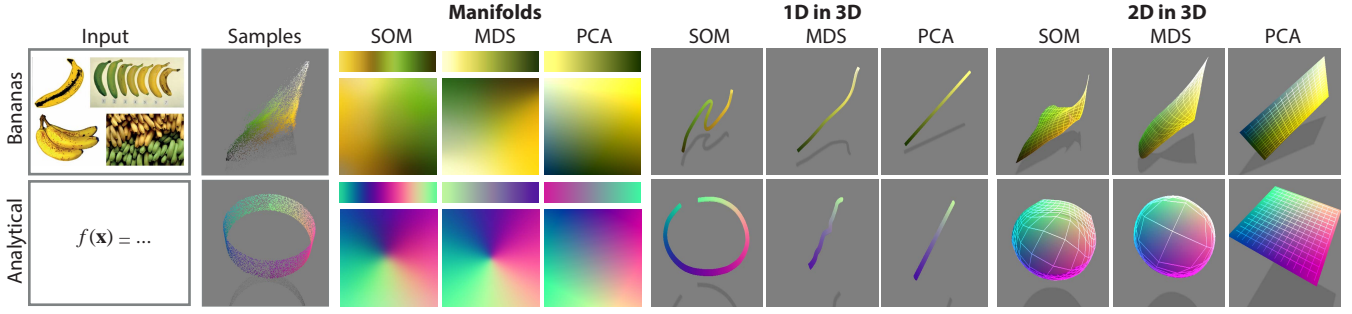


Fig. 5. Different manifolds produced using different approaches. The first column (“Input”) shows the high-dimensional input used. The second column (“Samples”) shows the color distribution in 3D RGB space. Next are the 1D and 2D manifolds (“Manifolds”) generated using SOM, MDS or PCA. Finally, we show these 1D manifolds (“1D in 3D”) and 2D manifolds (“2D in 3D”) as paths and patches in 3D RGB space. The first row shows the results of the “Bananas” class. The second row shows the results of an analytical color distribution. Here, SOM outperforms both PCA and MDS.

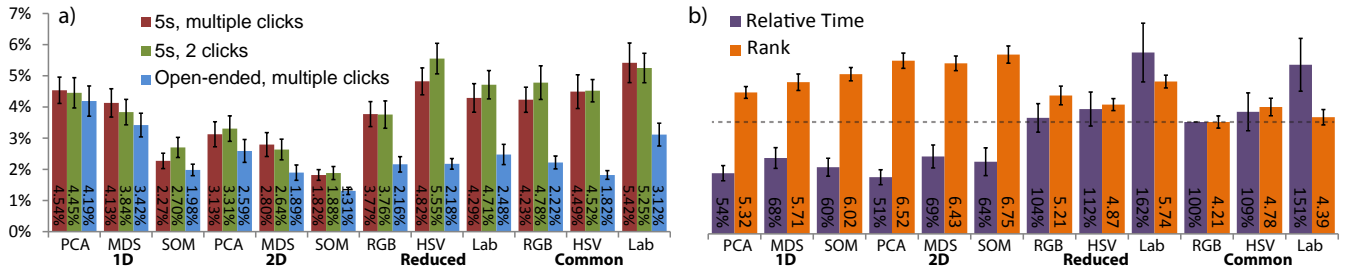


Fig. 6. User study results: (a) Adjustment task: Normalized CIEDE2000 error of several adjustment UIs and their standard error of mean. Lower numbers are better (less error). (b) Exploration task: Mean time, relative to common RGB (dotted line) and mean rank accompanied by their standard error of mean. Lower time is better (faster). A higher rank indicates a better rating from the subjects. Different experimental conditions for the same UI are encoded as different colors.

SOM. The “reduced” color spaces acted as a control group to detect whether an increase of performance of color manifolds was only due to excluding irrelevant colors (e.g., “skin” does not include blue). The task had to be finished either within a 5 second time budget with multiple mouse clicks, by using two clicks or by multiple clicks in an open-ended setting. We measured user performance as the average CIEDE2000 color difference of all pixels and all subjects denoted in percentage of the maximal error. Fig. 6a shows the performance of different interfaces. As the mean difference distribution was non-Gaussian (D’Agostino-Pearson), we used the Kruskal-Wallis non-parametric test instead of ANOVA [Siegel and Castellan 1988]. Dunn’s multiple comparisons test was used to calculate the pairwise statistical significance between different interfaces. Both our 1D and 2D SOM manifolds outperform classical and reduced color pickers with statistical significance $p < 0.05$.

The color exploration task was performed in two phases where some participants created images using different interfaces and other subjects ranked their images later on. In the *creation* phase, subjects were asked to adjust the color of an image using different interfaces until satisfied. Four expert users (computer graphics hobbyists naïve to the purpose of the study) participated in this step performing 60 trials each. In every trial, participants were presented one out of five different images where some parts needed to be colored using one out of 12 different color pickers from the first experiment. The resulting images were used in a second *ranking* phase where we randomly chose 10 different sets of images. Every set contained 12 images edited by one specific expert user on the same input image using 12 different interfaces. 10 other subjects were asked to sort the images by increasing color appearance quality. Fig. 6b shows the average finishing time of different color pickers

relative to using an RGB color picker for a given set and the average ranking for different color pickers. Similar to the adjustment study, we used a Kruskal-Wallis test and Dunn’s post-hoc. Overall, our 1D and 2D SOM outperform classical and reduced color pickers with statistical significance $p < 0.05$, in both finishing time and ranking in the exploration task. The supplementary material contains more details on statistical significance of our studies.

Conclusion. Our studies re-confirm that the choice of classical color pickers has only little impact on the performance of a color selection interface. Our studies show that encoding “class” information into classical color pickers, such as done in our “reduced” pickers, does not improve the performance in both color adjustment and color exploration task. Our data-driven 1D or 2D manifolds outperform classical color pickers and reduced color pickers in both color adjustment and exploration task. Furthermore, SOM is the best choice compared to PCA and MDS. Unless stated otherwise, weakly supervised SOM was used for further results shown in the paper. The supplementary material contains more details on our studies.

5. RESULTS

5.1 Manifolds

This section shows 1D and 2D manifolds produced automatically for a certain class, for classes with additional semantic parameters or by classes interactively composed by a user.

1D and 2D manifolds. Manifolds generated for different classes are shown in Fig. 7. As images contain shading, true reflectance data as extracted from Internet 3D model repositories can be superior (Fig. 8). If the class is not unique inside the

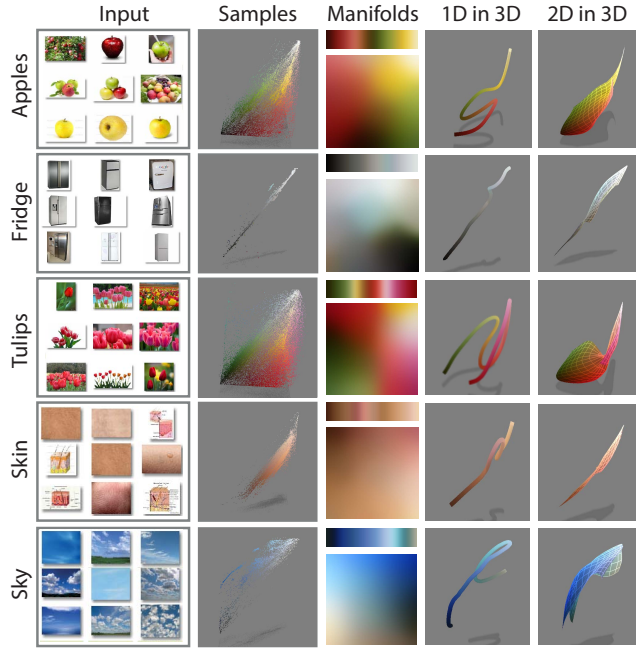


Fig. 7. Input images returned by Google Image Search (1st col., only subset shown), color distribution in 3D RGB space (2nd col.) and 1D as well as 2D manifolds (3rd col.) produced by our weakly supervised SOM for different classes (rows). Finally, we show these 1D manifolds (4th col.) and 2D manifolds (5th col.) as paths and patches in 3D RGB space.

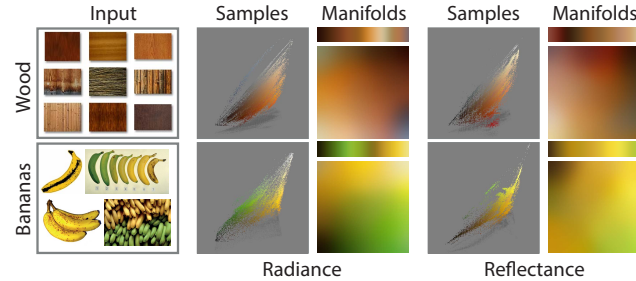


Fig. 8. Comparison of manifolds constructed from 2D images with shading returned by Google Image Search (3rd col.) and reflectance from textures of 3D meshes acquired from Google Sketchup (5th col.). The color distribution of the 2D images in 3D RGB space and the reflectance from the textures are shown in the second and forth column, respectively.

image, we allow user annotation (Fig. 9) or make use of already annotated data such as the SUN database [Xiao et al. 2010] or the OpenSurface database [Bell et al. 2013]. Fig. 10 shows different manifolds generated using different databases. Fig. 11 shows classes of our 1D and 2D manifolds compared to the classical color pickers. Manifolds are low-resolution images of only a few kilobyte that are ready to be used in any application just by using any smooth reconstruction filter. More details about our manifolds and training data are available in the supplemental material.

Parametrized families of manifolds. The range of colors found for a certain class sometimes depends on external conditions, such as the colors of a forest follow the seasons or the sky's color changes over the course of a day. Our approach naturally supports this observation by extending to families of manifolds $g_t(\mathbf{x})$,

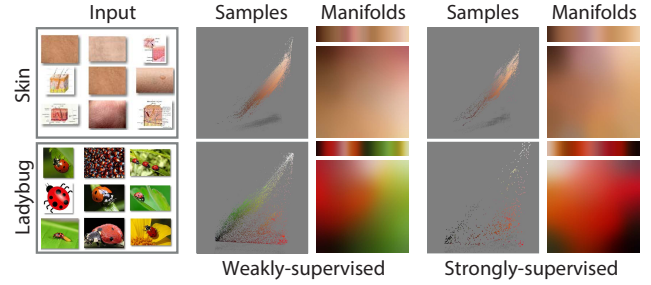


Fig. 9. Comparison of weakly and strongly supervised manifolds. Weakly supervised manifolds (3rd col.) are constructed using input images returned by Google Image Search (1st col., only subset shown). Next, image pixels that do not belong to the class are manually excluded and the refined images are then used to construct the strongly supervised manifolds (5th col.).

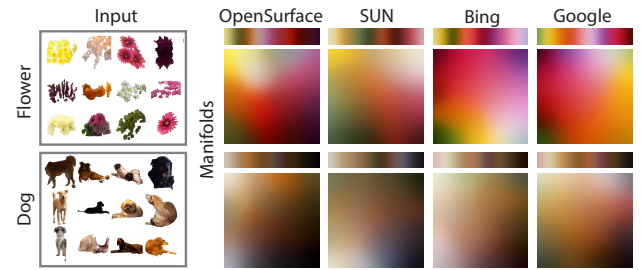


Fig. 10. Comparison of manifolds generated using different databases. The first column shows a subset of the class from the OpenSurface database [Bell et al. 2013]. Next are the manifolds generated using the input images from the annotated database OpenSurface (2nd col.), Sun [Xiao et al. 2010] (3rd col.), or weakly supervised images returned by Bing (4th col.) and Google (5th col.) Image Search using the same keywords.

parametrized by a k -dimensional vector $\mathbf{t} \in \mathbb{R}^k$. To this end, we acquire pairs $(\mathbf{a}_i, \mathbf{t}_i)$ of image colors and parameter values, such as when or where the image was taken. Then, we construct our manifold for every value of the parametric domain, e.g., one 1D MDS manifold for every time of the day as in Fig. 12.

Interactive manifold compositing. The efficient implementation of the manifold creation allows to interactively specify a set of images and observe the resulting manifold (Fig. 13). Interactive manifold creation could be used to summarize or further refine the color organization of a set of images produced on-demand, e.g., by an image search. Please see the supplemental video for several interactive manifold creation sessions.

5.2 Applications

Our manifolds allow for a range of applications, such as color editing, palettes, stylization, compression and white balancing.

User interaction. Our primary application of color manifolds is continuous 1D or 2D color slider (Fig. 1). Instead of continuous variation of color, a discrete sampling leads to meaningful recoloring suggestions (Fig. 15) or when directly used as discrete palettes (Fig. 14). Compared to simpler alternatives like clustering, the discrete elements produced by our approach have a meaningful one- or two-dimensional ordering and can be refined if required.

Color stylization. Our approach can be used to permute the hue value in an image, while staying inside the manifold of plausible



Fig. 11. Several manifolds generated using weakly supervised data from the Internet images. The final two sets on the bottom row (denoted as (S)) show the manifolds of two categories from the SUN database.

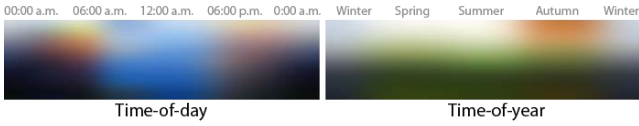


Fig. 12. Two 1D parametric families of 1D manifolds. Each vertical slice is a manifold in color space that changes along the horizontal dimension depending on a (semantic) parameter. The supplemental material shows a subset of the images used to construct the manifolds.

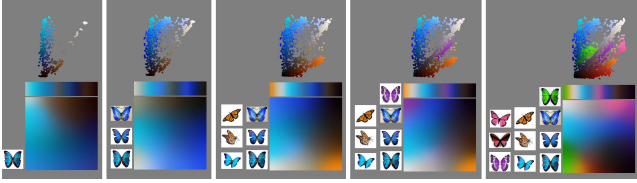


Fig. 13. Interactive manifold creation in five steps. Starting from a single input image (1st col.), a user interactively adds more images (2nd to 5th col.) to refine the “Butterfly” manifold. In every screenshot, the left column shows the set of input images, the top right shows the color point cloud in 3D and the bottom right shows the 1D and 2D manifolds.

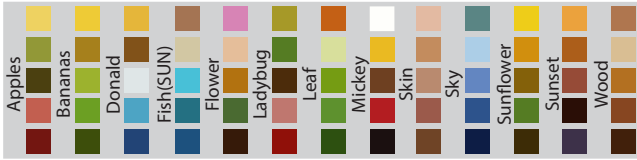


Fig. 14. Palettes produced by equally-spaced sampling of a 1D manifold. The palettes cover the space well and in a meaningful order.

images (Fig. 16). To this end, the image is segmented manually, a new random hue from the manifold is assigned to every segment.

Compression. Our approach can be used to compress color information. First, the image is converted to CIE Lab. The luminance L is not compressed and the chrominance is processed further. Second, a color manifold is chosen, and all colors are transformed into the new space and compressed. Quantization of colors to a low number of bits is used to compress colors in our example, but other operations such as DCT or wavelets would be possible. Additionally, we can quantize less important coordinates with 0 bits, skipping this dimension, i.e., reducing a 2D color to a 1D color. For decompression they are transformed back and combined with luminance (Fig. 17). Our technique is especially useful to compress a set of images from the same class.



Fig. 15. Re-coloring suggestion galleries using the “Apples” (1st row), “Skin” (2nd row) and “Sunset” (3rd row) manifold. Original image on the second row courtesy of Ross Whitehead.



Fig. 16. An original image (1st col.) and its hue permutations (2nd to 4th col.) using the “Apples” manifold. Original image courtesy of Keith Weller.

White balance. White balance seeks to disambiguate the product $\mathbf{a} = L\mathbf{r}$ of a spatially-invariant, unknown scalar RGB illuminant L and an unknown spatially-varying diffuse RGB reflectance \mathbf{r} . Humans are known to exploit knowledge of familiar i.e., plausible, reflectance to disambiguate this product [Olkkonen et al. 2008]. Our manifolds encode such knowledge and can extend the popular “grey world assumption” to an “on-manifold assumption”. This assumption defines the illuminant as the ratio of the average color in a k pixel-image and grey $L_{\text{grey}} = \sum \mathbf{a}_i/k$. In this framework, observing a slightly pink image would result in a pink illuminant. If we, however, know that parts of the image belong to the “human skin” class, pink might not be the illuminant, but the reflectance and the illuminant is white. Along those lines, our manifolds can be used to regularize plausible values for \mathbf{r} as follows. First, the set of image pixels \mathcal{M} that belong to a known class are selected. Next, we solve for the best RGB illuminant L that minimizes the cost function $\sum_{i \in \mathcal{M}} \|\alpha_f(\frac{\mathbf{a}_i}{L}) - \frac{\mathbf{a}_i}{L}\|^2$ where $\alpha_f(\mathbf{x})$ is the nearest color to $\mathbf{x} \in \mathbb{R}^m$ on the manifold f . The cost function optimizes for the manifold constraint that pixels in \mathcal{M} after white balancing should stay inside the manifold (Fig. 18). While the cost is non-linear due to the projection operation α_f , the space is small enough (all possible colors) to be enumerate exhaustively using a GPU-solver employing a 3D look-up table for α_f .

The approach is limited in that it cannot disambiguate situations, where the product of reflectance and illuminant, as well as the reflectance itself are close to the manifold: a slight blue tint in a human face which is likely not reflectance, but illuminant; a brown

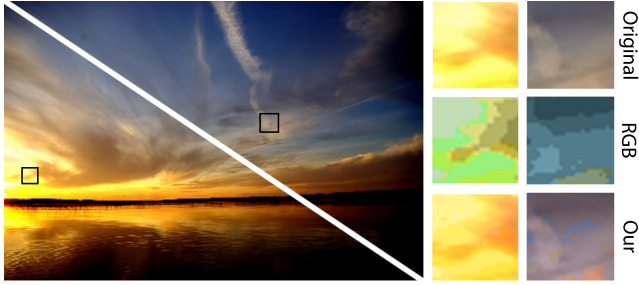


Fig. 17. Original (bottom left) and compression using our “Sunset” manifold (top right) and using the RGB space (insets).



Fig. 18. White balance (left to right): Original, grey-world assumption, Adobe Photoshop CS6 Auto Tone, our approach, finally the mask and our illuminant. The manifolds used are: “Skin”, “Donald” and “Bananas”.

shift might be the reflectance (tanned skin) or it might be a brownish illuminant. The latter cannot be disambiguated.

6. DISCUSSION AND CONCLUSION

This paper proposed content-dependent color manifolds as a replacement for established general color spaces by embedding a curve or surface into a 3D color space. This allows for improved color selection, as validated by perceptual studies, color stylization, compression and white balancing.

Discussion. Our pipeline is specifically designed for high performance and each step could be replaced to improve the quality of the constructed manifolds. During density estimation (Sec. 3.2), colors are quantized into l bins instead of using a continuous estimation over k colors such as RBF or moving least squares. This allows for higher performance and provides a unified input for different dimensionality reduction methods (Sec. 3.3), furthermore discrete histograms can be adapted easily to a simple outlier removal step by thresholding the bins. While robust statistics [Maronna et al. 2006] in a continuous estimation will likely improve the quality of outlier removal, it would further complicate our pipeline. We have shown that increasing the resolution of the histogram gives similar results in the supplementary material.

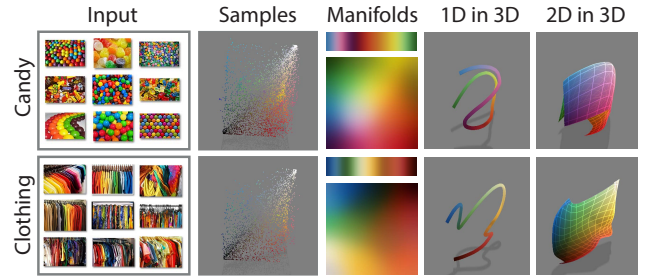


Fig. 19. Limitations: Colors of some man-made and textured objects that virtually exists in all hues and shades and consequently should be used in 3D. The dimensionality reduction step would try hard to find the best traversal in 3D, resulting in a desperate zig-zag enumeration of the entire higher-dimensional space. Detailed explanations of this figure can be found in the caption of Fig. 7.

Limitations. The main limitation of our work is the requirement of reliable classification, both to construct the manifold and to use it. For construction, we found the quality of typical search engine results to be sufficient, as indicated by our results. Typical failure modes are ambiguous queries such as “apple”, which might refer to a fruit as well as to a brand of computers. For weakly supervised manifolds, we do not perform any preprocessing on the input image set. As images returned from Google Image Search might contain improper camera calibration, color distortions or the inter-reflection between objects, they will likely produce less sharp manifolds. Our simple background removal (Sec. 3.1) might fail to remove background colors from several images in the set due to complex backgrounds or objects with holes, etc. These parts normally occupy a small part of the images which could be rectified by density thresholding (Sec. 3.2). In some cases, colors that do not belong to the class such as green in the “ladybug” (Fig. 9) are dominant. Even though these outliers (colors) might greatly reduce the usability of our manifold, they are closely related to the class semantically (leaves in the “ladybug”) and we believe that they can act as supplementary colors for a specific class. While manifold construction faces all those challenges, once the manifolds for a class are made available (such as in our supplemental material) they don’t need to be acquired again. For manifold usage, the ideal classifier would also detect the object class for a location inside the image or identify different classes if multiple classes are present in one image [Jia 2013]. Such classifiers are an active area of research in computer vision but not yet readily available to computer graphics applications. If the entire image belongs to a single class or the class is known a priori our manifolds are ready to use. For some classes, the dimensionality reduction is not always possible in a satisfying manner. We are able to find potentially non-linear lower-dimensional structures, but only if they are present. Classes that have a color distribution that roughly follows a dominant line are well-described by 1D manifolds, classes that are roughly distributed around a surface can be represented using 2D manifolds. However it is futile and counterproductive, to organize an inherently 2D distributions of colors into a 1D curve. Fig. 19 contains more results of man-made objects or objects with rich textures. Our method might fail to produce the manifolds for these objects as they could have multiple dominant colors over the surface and thus form multiple manifolds in color space. Please see the supplementary material for more results. Finally, perceptual organization of colors into lightness, saturation, or the periodicity of hue would require additional considerations.

Future work. In future work, we would like to extend our idea to very high or infinite-dimensional input spaces such as spectral colors or BDRFs while keeping smoothness, density and distance-preservation of our embedding. Another generalization could address textured variation or materials. Besides, comparison to different elaborate interfaces such as image galleries [Shapira et al. 2009] or color template [O'Donovan et al. 2011] in color adjustment and exploration tasks is an interesting future direction. Furthermore, to simplify the study, a detailed, soft selection is assumed to exist and subjects were asked to modify the images by picking a single color. This can be generalized in the spirit of “How do humans colorize images?” by allowing multiple color selections, in a manner similar to [Cole et al. 2008].

We hope that our intuitive 1D or 2D manifolds will substantially ease the navigation of color spaces (as shown in the color adjustment study) and improve color editing quality in general (as shown in the color exploration study).

ACKNOWLEDGMENTS

We would like to thank Antti Oulasvirta for help with the user study and proofreading, Glenn Lawyer and Karol Myszkowski for the voice-over, Oliver Nalbach, Daniel Scherzer, Michael Wand, Krzysztof Templin, Tania Pouli, Ondřej Karlík, Jiri Vorba, Bernhard Reinert, Oliver Klehm for proofreading and all anonymous reviewers for their constructive comments. We also thank Greg Lasley (<http://www.greglasley.net>), Ross Whitehead, Keith Weller (USDA Agricultural Research Service), Dominik Stodulski and Loren Javier for the images used in the paper.

REFERENCES

- BELL, S., UPCHURCH, P., SNAVELY, N., AND BALA, K. 2013. Open-surfaces: A richly annotated catalog of surface appearance. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4, 111:1–111:17.
- CLAUSEN, C. AND WECHSLER, H. 2000. Color image compression using PCA and backpropagation learning. *Pattern Recognition* 33, 9, 1555–60.
- COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y. 2006. Color harmonization. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 624–630.
- COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where do people draw lines? *ACM Trans. Graph. (Proc. SIGGRAPH)* 27, 3, 88:1–88:11.
- COX, T. AND COX, M. 2000. *Multidimensional scaling*. Vol. 88. Chapman & Hall/CRC.
- DOUGLAS, S. AND KIRKPATRICK, A. 1999. Model and representation: the effect of visual feedback on human performance in a color picker interface. *ACM Trans. Graph.* 18, 2, 96–127.
- FAIRCHILD, M. 2005. *Color appearance models*. John Wiley & Sons.
- GASTAL, E. S. L. AND OLIVEIRA, M. M. 2011. Domain transform for edge-aware image and video processing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4, 69:1–69:12.
- HSU, R.-L., ABDEL-MOTTALEB, M., AND JAIN, A. 2002. Face detection in color images. *IEEE PAMI* 24, 5, 696–706.
- JIA, Y. 2013. Caffe: An open source convolutional architecture for fast feature embedding. <http://caffe.berkeleyvision.org>.
- KOHONEN, T. 1990. The self-organizing map. *Proceedings of the IEEE* 78, 9, 1464–1480.
- LALONDE, J.-F. AND EFROS, A. A. 2007. Using color compatibility for assessing image realism. In *Proc. ICCV*. 1–8.
- LAWRENCE, R. D., ALMASI, G. S., AND RUSHMEIER, H. E. 1999. A scalable parallel algorithm for self-organizing maps with application to sparse data mining problems. *Data Min. Knowl. Discov.* 3, 2, 171–195.
- LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*. Vol. 2. 2169–78.
- LISSNER, I. AND URBAN, P. 2009. How perceptually uniform can a hue linear color space be? In *Proc. IST/CIC*. 97–102.
- MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUMML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*. 389–400.
- MARONNA, R. A., MARTIN, D. R., AND YOHAI, V. J. 2006. *Robust Statistics: Theory and Methods*. John Wiley and Sons.
- MATSUDA, Y. 1995. Color design. *Asakura Shoten* 2, 4, 10.
- MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. In *Proc. SIGGRAPH*. 27–31.
- O'DONOVAN, P., AGARWALA, A., AND HERTZMANN, A. 2011. Color compatibility from large datasets. *ACM Trans. Graph. (Proc. SIGGRAPH)* 3, 63:1–63:12.
- OLKKONEN, M., HANSEN, T., AND GEGENFURTNER, K. R. 2008. Color appearance of familiar objects: Effects of object shape, texture, and illumination changes. *J Vision* 8, 5, 1–16.
- OMER, I. AND WERMAN, M. 2004. Color lines: Image specific color representation. In *Proc. CVPR*. Vol. 2. II–946.
- OSKAM, T., HORNUNG, A., SUMNER, R., AND GROSS, M. 2012. Fast and stable color balancing for images and augmented reality. In *Proc. 3DIMPVT*. 49–56.
- SCHWARZ, M. W., COWAN, W. B., AND BEATTY, J. C. 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. *ACM Trans. Graph.* 6, 2, 123–158.
- SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2009. Image appearance exploration by model-based navigation. *Comp. Graph. Forum (Proc. EUROGRAPHICS)* 28, 2, 629–38.
- SIEGEL, S. AND CASTELLAN, N. J. 1988. *Nonparametric Statistics for The Behavioral Sciences*. McGraw-Hill Higher Education.
- TENENBAUM, J., DE SILVA, V., AND LANGFORD, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500, 2319–23.
- TZENG, D. AND BERNS, R. 2005. A review of principal component analysis and its applications to color technology. *Color Res. & App.* 30, 2, 84–98.
- WANG, B., YU, Y., WONG, T.-T., CHEN, C., AND XU, Y.-Q. 2010. Data-driven image color theme enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH)* 3, 146:1–146:10.
- WANG, B., YU, Y., AND XU, Y.-Q. 2011. Example-based image color and tone style enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH)* 3, 64:1–64:12.
- WANG, J., TONG, X., LIN, S., PAN, M., WANG, C., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph. (Proc. SIGGRAPH)* 3, 754–61.
- XIAO, J., HAYS, J., EHINGER, K., OLIVA, A., AND TORRALBA, A. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*. 3485–92.
- XUE, S., WANG, J., TONG, X., DAI, Q., AND GUO, B. 2008. Image-based material weathering. *Comp. Graph. Forum (Proc. EUROGRAPHICS)* 27, 2, 617–626.

Received December 2013; accepted October 2014