

Palette-based image decomposition, harmonization, and color transfer

JIANCHAO TAN, George Mason University

JOSE ECHEVARRIA, Adobe Research

YOTAM GINGOLD, George Mason University

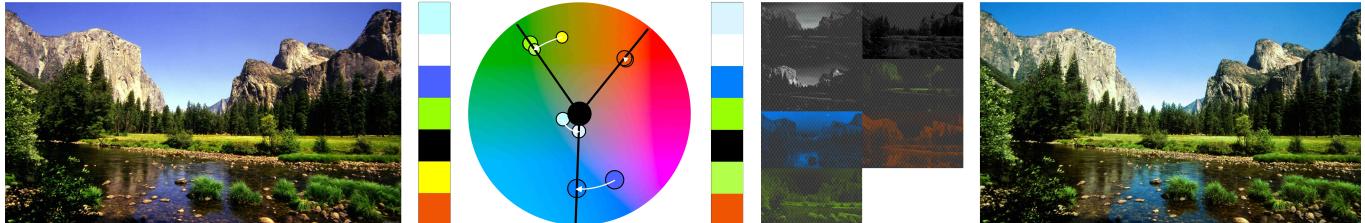


Fig. 1. Our palette-based color harmony template is able to express harmonization and various other color operations in a concise manner. Our edits make use of a new, extremely efficient image decomposition technique based on the 5D geometry of RGBXY-space.

We present a palette-based framework for color composition for visual applications. Color composition is a critical aspect of visual applications in art, design, and visualization. The color wheel is often used to explain pleasing color combinations in geometric terms, and, in digital design, to provide a user interface to visualize and manipulate colors.

We abstract relationships between palette colors as a compact set of axes describing *harmonic templates* over perceptually uniform color wheels. Our framework provides a basis for a variety of color-aware image operations, such as color harmonization and color transfer.

To enable our approach, we introduce an extremely scalable and efficient yet simple palette-based image decomposition algorithm. Our approach is based on the geometry of images in RGBXY-space. This new geometric approach is orders of magnitude more efficient than previous work and requires no numerical optimization.

CCS Concepts: • Computing methodologies → Image manipulation; Image processing;

Additional Key Words and Phrases: images, layers, painting, palette, generalized barycentric coordinates, harmonization, contrast, convex hull, RGB, color space, recoloring, compositing, mixing

ACM Reference Format:

Jianchao Tan, Jose Echevarria, and Yotam Gingold. 2019. Palette-based image decomposition, harmonization, and color transfer. 1, 1 (December 2019), 11 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Color composition is critical in visual applications in art, design, and visualization. Over the centuries, different theories about how colors interact with each other have been proposed [Westland et al.

Authors' addresses: Jianchao Tan, George Mason University, tanjianchao@stc@gmail.com; Jose Echevarria, Adobe Research, echevarr@adobe.com; Yotam Gingold, George Mason University, ygingold@gmu.edu.

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 bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2019 Copyright held by the owner/author(s).

XXXX-XXXX/2019/12-ART

https://doi.org/10.475/123_4

2007]. While it is arguable whether a universal and comprehensive color theory will ever exist, most previous proposals share in common the use of a color wheel (with hue parameterized by angle) to explain pleasing color combinations in geometric terms. In the digital world, the color wheel often serves as a user interface to visualize and manipulate colors. This has been explored in the literature for specific applications in design [Adobe 2018] and image editing [Cohen-Or et al. 2006].

In this paper, we embrace color wheels to present a new framework where color composition concepts are easy and intuitive to formulate, solve for, visualize, and interact with; for applications in art, design, or visualization. Our approach is based on palettes and relies on palette-based image decompositions. To fully realize it as a powerful image editing tool, we introduce an extremely efficient yet simple new image decomposition algorithm.

We define our color relationships in the CIE LCh color space (the cylindrical projection of CIE Lab). Contrary to previous work using HSV color wheels, the LCh color space ensures that perceptual effects are accounted for with no additional processing. For example, a simple global rotation of hue in LCh-space (but not HSV-space) preserves the perceived lightness or gradients in color themes and images.

To represent color information, we adopt the powerful palette-oriented point of view [Mellado et al. 2017] and propose to work with color palettes of arbitrary numbers of swatches. Unlike hue histograms, color palettes or swatches can come from a larger variety of sources (extracted from images, directly from user input, or from generative algorithms) and capture the 3D nature of LCh in a compact way. They provide intuitive interfaces and visualizations as well.

Color palettes also simplify the modelling and formulation of relationships between colors. This last point enables the simplification of harmonic templates and other relationships into a set of a few 3D axes that capture color structure in a meaningful and compact way. This is useful for various color-aware tasks. We demonstrate applications to color harmonization and color transfer. Instead of using the sector-based templates from Matsuda [Tokumaru et al.

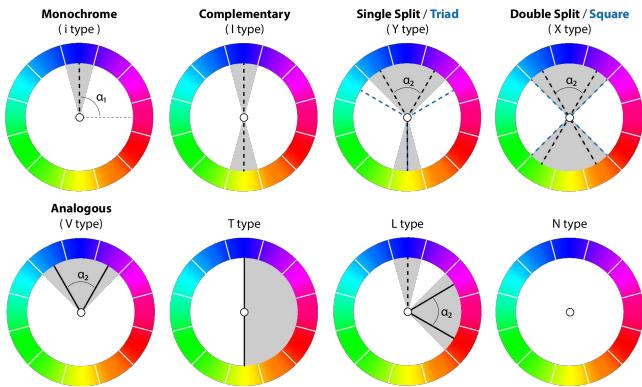


Fig. 2. Comparison between the sector-based harmonic templates from [Tokumaru et al. 2002] (shaded in grey), and our new axis-based ones. We use two different types of axes: the dashed ones attract colors towards them; the solid ones define sectors between them, so colors inside remain in the same position, but colors outside are attracted towards them. We found this distinction helps handling templates like *analogous* properly. Note that our templates derived from [Itten 1970] separate *Y type* into *single split* and *triad*, and the same for *X type*. These templates are popular among creatives, but they are also in agreement with the definitions of similarity and ambiguity by Moon and Spencer [1944]. Although we don't use it in our results, our approach can also describe hybrid templates like *L type*. Each template can be modeled by a single global rotation α_1 , although some of them have a secondary degree of freedom α_2 that enforces symmetry. In this paper we focus on *monochrome*, *complementary*, *single split*, *triad*, *double split*, *square* and *analogous*.

2002] (appropriate for hue histograms) we derive our harmonic templates from Itten's color theory [Itten 1970] (see Figure 2). We also propose new color operations using this axes-based representation. Our proposed framework can be used by other palette-based systems and workflows, either for palette improvement or image editing.

At the core of our and other recent approaches [Aksoy et al. 2017; Chang et al. 2015; Tan et al. 2016; Zhang et al. 2017] to image editing, images are decomposed into a palette and associated per-pixel compositing or mixing parameters. We propose a new, extremely efficient yet simple and robust algorithm to do so. Our approach is inspired by the geometric palette extraction technique of Tan et al. [2016]. We consider the geometry of 5D RGBXY-space, which captures color as well as spatial relationships and eliminates numerical optimization. After an initial palette is extracted (given an RMSE reconstruction threshold), the user can edit the palette and obtain new decompositions instantaneously. Our algorithm's performance is extremely efficient even for very high resolution images (≥ 100 megapixels)—20x faster than the state-of-the-art [Aksoy et al. 2017]. Working code is provided in Section 3. Our algorithm is a key contribution which enables our approach and many other applications proposed in the literature.

In summary, this paper makes the following contributions:

- A new palette-based color harmonization framework, general enough to model classical harmonic relationships, new color

composition operations, and a compact structure for other color-aware applications.

- An extremely efficient, geometric approach for decomposing an image into spatially coherent additive mixing layers by analyzing the geometry of an image in RGBXY-space. Its performance is virtually independent from the size of the image or palette. Our decomposition can be *re-computed* instantaneously for a new RGB palette, allowing designers to edit the decomposition in real-time.

We demonstrate other applications like color transfer, greatly simplified by our framework.

2 RELATED WORK

There are many works related with our contributions and their applications. In the following we cover the most relevant ones.

2.1 Color Harmonization

Many existing works have applied different concepts from traditional color theory for artists to improve the color composition of digital images. In their seminal paper, Cohen-Or et al. [2006] use hue histograms and harmonic templates defined as sectors of hue-saturation in HSV color space [Tokumaru et al. 2002], to model and manipulate color relationships. They fit a template (optimal or arbitrary) over the image histogram, so they can shift hues accordingly to harmonize colors or composites from several sources. Additional processing is needed to ensure spatial smoothness. Several people have built on top of this work, extending or improving parts of their proposed framework. Sawant and Mitra [2008] extended it to video, focusing on temporal coherence between successive frames. Improvements to the original fitting have been proposed based on the number of pixels for each HSV value [Huo and Tan 2009], the visual saliency [Baveye et al. 2013], the extension and visual weight of each color. [Baveye et al. 2013], or geodesic distances [Li et al. 2015]. Tang et al. [2010] improves some artifacts during the recoloring of [Cohen-Or et al. 2006]. Chamaret et al. [2014] defines and visualizes a per-pixel harmony measure to guide interactive user edits.

Instead of using hue histograms from images, our framework is built on top of color palettes, independently of their source. Given the higher level of abstraction of palettes, we simplify harmonic templates to arrangements of axes in chroma-hue space (from LCh), interpreted and derived directly from Itten's theories [1970]. This more general and simpler representation makes for more intuitive metrics, easier to solve, that enable a wider range of applications. When working with images, this approach fits perfectly with our proposed palette extraction and image decomposition for very efficient and robust image recoloring. Related to our approach, Mellado et al. [2017] is also able to pose harmonization as a set of constraints within their general constrained optimization framework. Our new templates, posed in LCh space, could be added as additional constraints.

Finally, there is a different definition for harmony in composited images, in terms of contrast, texture, noise or blur. Works dealing with it [Sunkavalli et al. 2010; Tsai et al. 2017] focus on a completely different set of goals and challenges than the work discussed above.

2.2 Palette Extraction and Image Decomposition

Palette Extraction A straightforward approach consists of using a k-means method to cluster the existing colors in an image, in RGB space [Chang et al. 2015; Phan et al. 2017; Zhang et al. 2017]. A different approach consists of computing and simplifying the convex hull enclosing all the color samples [Tan et al. 2016], which provides more general palettes that better represent the existing color gamut of the image. A similar observation was made in the domain of hyperspectral image unmixing [Craig 1994]. (With hyperspectral images, palette sizes are smaller than the number of channels, so the problem is one of fitting a minimum-volume simplex around the colors. The vertices of a high-dimensional simplex become a convex hull when the data is projected to lower dimensions.) Morse et al. [2007] work in HSL space, using a histogram to find the dominant hues, then to find shades and tints within them. Human perception has also been taken into account in other works, training regression models on crowd-sourced datasets. [Lin and Hanrahan 2013; O’Donovan et al. 2011]. Some physically-based approaches try to extract wavelength-dependent parameters to model the original pigments used paintings. [Aharoni-Mack et al. 2017; Tan et al. 2017]. Our work builds on top of Tan et al. [2016], adding a fixed reconstruction error threshold for automatic extraction of palettes of optimal size, as described in Section 3.1.

Image Decomposition For recoloring applications, it is also critical to find a mapping between the extracted color palette and the image pixels. Recent work is able to decompose the input image into separate layers according to a palette. Tan et al. [2016] extract a set of ordered translucent RGBA layers, based on a optimization over the standard alpha blending model. Order-independent decompositions can be achieved using additive color mixing models [Aksøy et al. 2017; Lin et al. 2017a; Zhang et al. 2017]. For the physically-based palette extraction methods mentioned previously [Aharoni-Mack et al. 2017; Tan et al. 2017], layers correspond to the extracted multispectral pigments. We prefer a full decomposition to a (palette-based) edit transfer approach like Chang et al. [2015]’s. With a full decomposition, edits are trivial to apply and spatial edits become possible (though we do not explore spatial edits in this work). We present a new, efficient method for layer decomposition, based on the additive color mixing model (Section 3.2). Our approach leverages 5D RGBXY-space geometry to enforce spatial smoothness on the layers. This geometric approach is significantly more efficient than previous approaches in the literature, easily handling images up to 100 megapixels in size.

2.3 Color Transfer

We also explore color transfer as an *application* of our work. Color transfer is a vast field with contributions from the vision and graphics communities. As such, we describe only the most closely related work to our approach. Hou et al. [2007] conceptualize and apply color themes as hue histograms in HSV space. Wang et al. [2010] solve an optimization that simultaneously considers a desired color theme, texture-color relationships as well as automatic or user-specified color constraints. Phan et al. [2017] explored the order of colors within palettes to establish correspondences and enable

interpolation. Nguyen et al. [2017] find a group color theme from multiple palettes from multiple images using a modified k-means clustering method, and use it to recolor all the images in a consistent way. Han et al. [2017; 2013] compute a distance metric between palettes in the *color mood* space, and then sort and match colors from palettes according to their brightness. Munshi et al. [2015] match colors between palettes according to their distance in Lab space. Based on our harmonic templates, palettes, and the LCh color space; we propose several intuitive metrics for color transfer that take into account human perception for goals like colorfulness, preservation of original colors, or harmonic composition. The final image recoloring is performed using our layer decomposition.

3 PALETTE EXTRACTION AND IMAGE DECOMPOSITION

A good palette for image editing is one that closely captures the underlying colors the image was made with (or could have been made with), even if those colors do not appear in their purest form in the image itself. Tan et al. [2016] observed that the color distributions from paintings and natural images take on a convex shape in RGB space. As a result, they proposed to compute the convex hull of the pixel colors. The convex hull tightly wraps the observed colors. Its vertex colors can be blended with convex weights (positive and summing to one) to obtain any color in the image. The convex hull may be overly complex, so they propose an iterative simplification scheme to a user-desired palette size. After simplification, the vertices become a palette that represents the colors in the image.

We extend Tan et al. [2016]’s work in two ways. First, we propose a simple, geometric layer decomposition method that is orders of magnitude more efficient than the state-of-the-art. Working code for our entire decomposition algorithm can be written in under 30 lines (Figure 5). Second, we propose a simple scheme for automatic palette size selection.

3.1 Palette Extraction

In Tan et al. [2016], the convex hull of all pixel colors is computed and then simplified to a user-chosen palette size. To summarize their approach, the convex hull is simplified greedily as a sequence of constrained edge collapses [Garland and Heckbert 1997]. An edge is collapsed to a point constrained to strictly add volume [Sander et al. 2000] while minimizing the distance to its incident faces. The edge whose collapse adds the least overall volume is chosen next, greedily. After each edge is collapsed, the convex hull is recomputed, since the new vertex could indirectly cause other vertices to become concave (and therefore redundant). Finally, simplification may result in out-of-gamut colors, or points that lie outside the RGB cube. As a final step, Tan et al. [2016] project all such points to the closest point on the RGB cube. This is the source of reconstruction error in their approach; some pixels now lie outside the simplified convex hull and cannot be reconstructed.

We improve upon this procedure with the observation that the reconstruction error can be measured geometrically, even before layer decomposition, as the RMSE of every pixel’s distance to the simplified convex hull. (Inside pixels naturally have distance 0.) Therefore, we propose a simple automatic palette size selection

based on a user-provided RMSE reconstruction error tolerance ($\frac{2}{255}$ in our experiments). For efficiency, we divide RGB-space into $32 \times 32 \times 32$ bins (a total of 2^{15} bins). We measure the distance from each non-empty bin to the simplified convex hull, weighted by the bin count. We start measuring the reconstruction error once the number of vertices has been simplified to 10. By doing this, we are able to obtain palettes with an optimal number of colors automatically. This removes the need for the user to choose the palette size manually, leading to better layer decompositions.

(If non-constant palette colors were acceptable, instead of clipping one could cast a ray from each pixel towards the out-of-gamut vertex; the intersection of the ray with the RGB cube would be the palette color for that pixel. There would be zero reconstruction error. The stopping criteria could be the non-uniformity of a palette color, measured by the area of the RGB cube surface intersected with the simplified convex hull itself.)

3.2 Image decomposition via RGBXY convex hull

From their extracted palettes, Tan et al. [2016] solved a non-linear optimization problem to decompose an image into a set of ordered, translucent RGBA layers suitable for the standard “over” compositing operation. While this decomposition is widely applicable (owing to the ubiquity of “over” compositing), the optimization is quite lengthy due to the recursive nature of the compositing operation, which manifests as a polynomial whose degree is the palette size. Others have instead opted for additive mixing layers [Aksoy et al. 2017; Lin et al. 2017a; Zhang et al. 2017] due to their simplicity. A pixel’s color is a weighted sum of the palette colors.

In this work, we adopt linear mixing layers as well. We provide a fast and simple, yet spatially coherent, geometric construction.

Any point p inside a simplex (a triangle in 3D, a tetrahedron in 3D, etc.) has a unique set of barycentric coordinates, or convex additive mixing weights such that $p = \sum_i w_i c_i$, where the mixing weights w_i are positive and sum to one, and c_i are the vertices of the simplex. In our setting, the simplified convex hull is typically not a simplex, because the palette has more than 4 colors. There still exist convex weights w_i for arbitrary polyhedron, known as generalized barycentric coordinates [Floater 2015], but they are typically non-unique. A straightforward technique to find generalized barycentric coordinates is to first compute a Delaunay tessellation of the polyhedron (in our case, the simplified convex hull), which tessellates it into a collection of non-overlapping simplices (tetrahedra in 3D). The *Delaunay generalized barycentric coordinates* for a point can then be computed as the barycentric coordinates of whichever simplex the point falls inside of. For a 3D point in general position in the interior, the mixing weights will have at most 4 non-zero weights, which corresponds to the number of vertices of a tetrahedron.

This is the approach taken by Tan et al. [2016] for their *as-sparse-as-possible* (ASAP) technique to extract layers. Because the weights are assigned purely based on the pixel’s colors, however, this approach predictably suffers from spatial coherence artifacts (Figure 4). The colors of spatially neighboring pixels may belong to different tetrahedra. As a result, ASAP layers produce speckling artifacts during operations like recoloring.

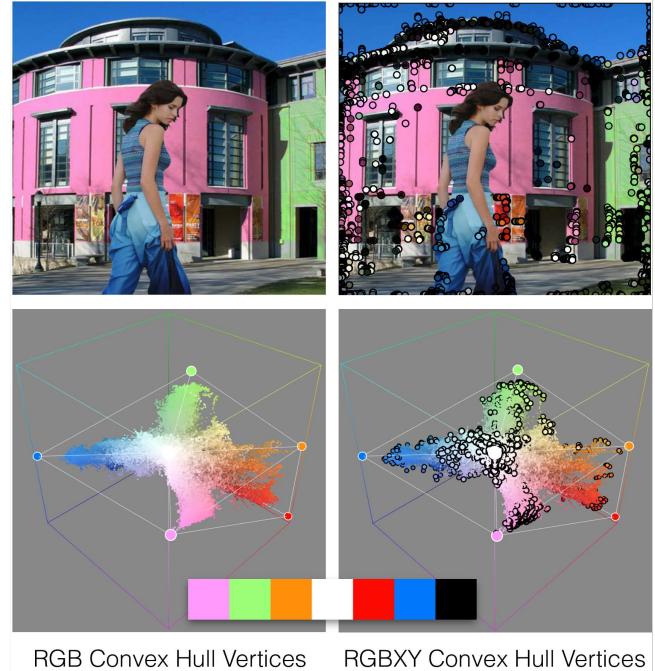


Fig. 3. Visualization of the two convex hulls. *Left:* the simplified RGB convex hull is the basis for the methods in Tan et al. [2016], capturing the colors of an image but not their spatial relationships. *Right:* Our 5D RGBXY convex hull captures color and spatial relationship at the same time. We visualize its vertices as small circles; its 5D simplices are difficult to visualize. Our approach splits image decomposition into a two-level geometric problem. The first level are the RGBXY convex hull vertices that mix to produce any pixel in the image. The second level are the simplified RGB convex hull vertices, which serve as the palette RGB colors. Since the RGBXY convex hull vertices lie inside the RGB convex hull, we find mixing weights that control the color of the RGBXY vertices. The two levels combined allow instant recoloring of the whole image. The top right image shows the locations of the RGBXY vertices in image space. The bottom row shows the geometric relationships between the 3D and 5D convex hull vertices, and how the simplified RGB convex hull captures the same color palette for both algorithms.

Our key insight is to extend this approach to 5D RGBXY-space, where XY are the coordinates of a pixel in image space, so that spatial relationship are considered along with color in a unified way (Figure 3). We first compute convex hull of the image in RGBXY-space. We then compute Delaunay generalized barycentric coordinates (weights) for every pixel in the image in terms of the 5D convex hull. Pixels that have similar colors *or* are spatially adjacent will end up with similar weights, meaning that our layers will be smooth both in RGB and XY-space. These mixing weights form an $Q \times N$ matrix W_{RGBXY} , where N is the number of image pixels and Q is the number of RGBXY convex hull vertices. We also compute W_{RGB} , Delaunay barycentric coordinates (weights) for the RGBXY convex hull vertices in the 3D simplified convex hull. We use the RGB portion of each RGBXY convex hull vertex, which always lies inside the RGB convex hull. Due to the aforementioned out-of-gamut projection step when computing the simplified RGB convex

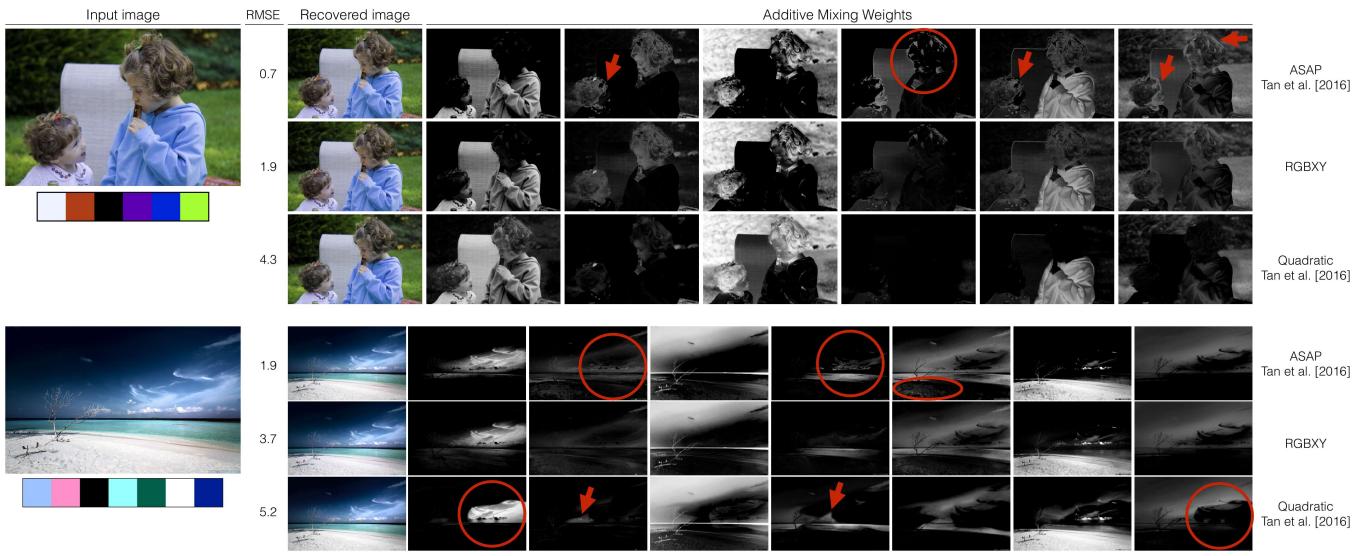


Fig. 4. Mixing weights comparison between our proposed RGBXY method and Tan et al. [2016] (ASAP and the quadratic, additive mixing adaptation of their optimization). The quadratic technique occasionally makes undesirable tradeoffs between its sparsity and smoothness terms, which manifest as awkward layer boundaries. The ASAP algorithm does not maintain spatial smoothness. While somewhat less sparse, our proposed 5D RGBXY convex hull-based approach naturally considers spatial and color proximity.

hull, however, an RGBXY convex hull vertex may occasionally fall outside it. We set its weights to those of the closest point on the 3D simplified convex hull. W_{RGB} is a $P \times Q$ matrix, where P is the number of vertices of the simplified RGB convex hull (the palette colors).

The final weights for the image are obtained via matrix multiplication: $W = W_{\text{RGB}} W_{\text{RGBXY}}$, which is a $P \times N$ matrix that assigns each pixel weights solely in terms of the simplified RGB convex hull. These weights are smooth both in color and image space. To decompose an image with a different RGB-palette, one only needs to recompute W_{RGB} and then perform matrix multiplication. Computing W_{RGB} is extremely efficient, since it depends only on the palette size and the number of RGBXY convex hull vertices. It is independent of the image size and allows users to experiment with image decompositions based on different palettes in real-time.

3.3 Comparison between the proposed methods

We evaluated our proposed layer decomposition method in terms of quality and speed.

A comparison of additive mixing weights can be seen in Figure 4 and 6. Tan et al. [2016]'s 3D ASAP method exhibits speckling and unnatural boundaries. We also compare our approach to a variant of Tan et al. [2016]'s higher quality optimization. We modified their reconstruction term to the simpler, quadratic one that matches our additive mixing layer decomposition scenario. With that modification, all energy terms become quadratic. However, because the sparsity term is not positive definite, it is not a straightforward Quadratic Programming problem; we minimize it with L-BFGS-B and increased the solver's default termination thresholds since RGB colors have low precision (gradient and function tolerance 10^{-4}). Optimization-based approaches typically have parameters to tune.

The competition between the sparsity and smoothness terms occasionally produce unnatural boundaries, as seen in the example shown. For the 100 images, our RGBXY method's average RMSE is 2.7. The Tan et al. [2016] variant's optimization method produces images with an average RMSE of 4.1. The additional error is due to a smoothness/reconstruction tradeoff made by the objective function. Aksoy et al. [2017]'s algorithm can reconstruct image with very small errors, since they greedily compute new layers until they can achieve a very small reconstruction errors, and because their palettes are color distributions rather than fixed colors. However, it

```

from numpy import *
from scipy.spatial import ConvexHull, Delaunay
from scipy.sparse import coo_matrix

def RGBXY_weights( RGB_palette, RGBXY_data ):
    RGBXY_hull_vertices = RGBXY_data[ ConvexHull( RGBXY_data ).vertices ]
    W_RGBXY = Delaunay_coordinates( RGBXY_hull_vertices, RGBXY_data )
    # Optional: Project outside RGBXY_hull_vertices[:,3] onto RGB_palette convex hull.
    W_RGB = Delaunay_coordinates( RGB_palette, RGBXY_hull_vertices[:,3] )
    return W_RGBXY.dot( W_RGB )

def Delaunay_coordinates( vertices, data ): # Adapted from Gareth Rees
    # Compute Delaunay tessellation.
    tri = Delaunay( vertices )
    # Find the tetrahedron containing each target (or -1 if not found).
    simplices = tri.find_simplex(data, tol=1e-6)
    assert (simplices != -1).all() # data contains outside vertices.
    # Affine transformation for simplex containing each datum.
    X = tri.transform[simplices, :data.shape[1]]
    # Offset of each datum from the origin of its simplex.
    Y = data - tri.transform[simplices, data.shape[1]:]
    # Compute the barycentric coordinates of each datum in its simplex.
    b = einsum( '...jk,...k->...j', X, Y )
    barycoords = c_[b, 1-b.sum(axis=1)]
    # Return the weights as a sparse matrix.
    rows = repeat(arange(len(data)).reshape((-1,1)), len(tri.simplices[0]), 1).ravel()
    cols = tri.simplices[simplices].ravel()
    vals = barycoords.ravel()
    return coo_matrix( (vals,(rows,cols)), shape=(len(data),len(vertices)) ).tocsr()

```

Fig. 5. Python code for our RGBXY additive mixing layer decomposition.

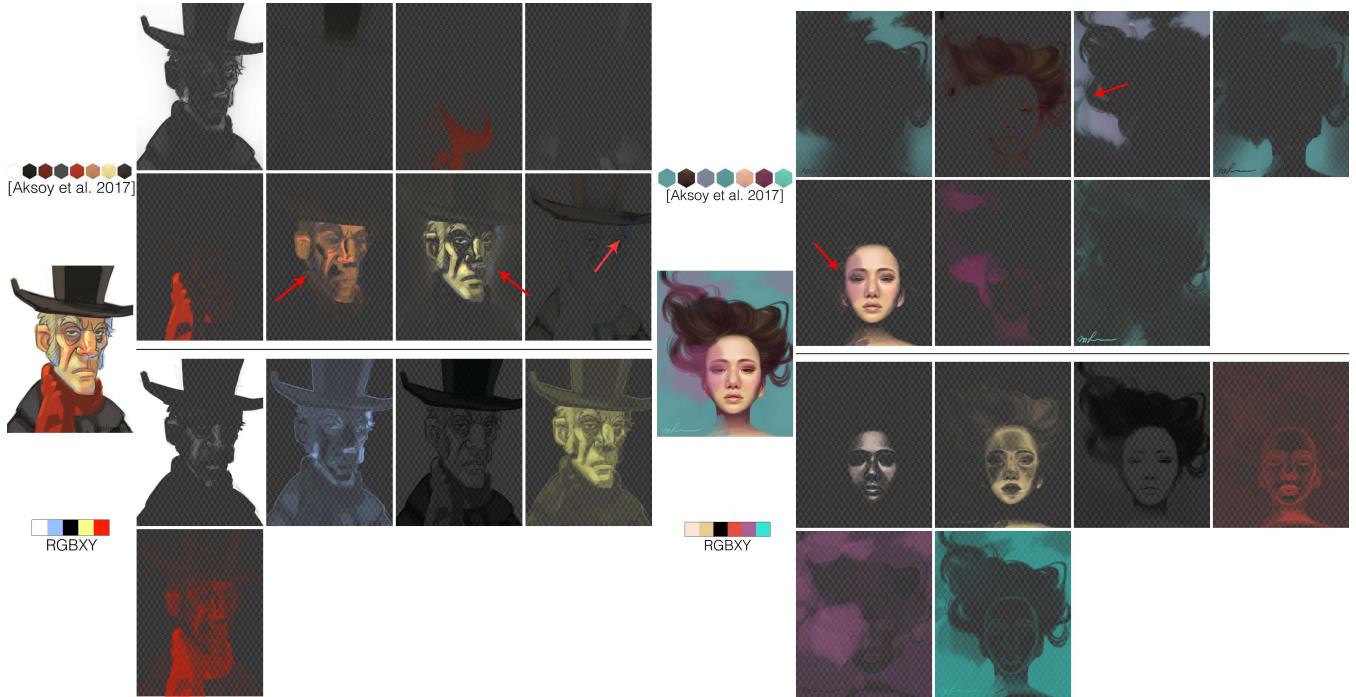


Fig. 6. A comparison between our proposed RGBXY image decomposition and that of Aksoy et al. [2017]. Aksoy et al. [2017] creates an overabundance of layers (two red layers above) and does not extract the blueish tint, which appears primarily in mixture. Our RGBXY technique identifies mixed colors is able to separate the translucent purple haze in front of the girl’s face.

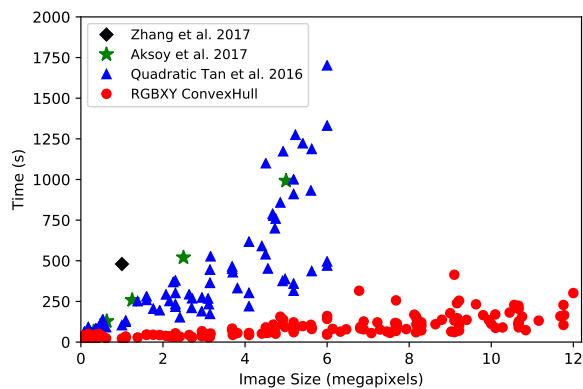


Fig. 7. Time comparison between four additive mixing image decomposition algorithms. We evaluated our RGBXY algorithm on 170 images up to 12 megapixels and an additional 6 100 megapixel images (not shown; average running time 12.6 minutes). Our algorithm’s performance scales well with image size. Occasional outliers (three red dots somewhat above the others) illustrate that the number of RGBXY convex hull vertices has a greater effect on performance than image size.

sometimes over-decomposes the image and fail to create layers for colors that appear primarily in mixture.

We have no explicit guarantees about the sparsity of our weights. W_{RGB} is as sparse as possible to reconstruct 3D colors (4 non-zeros). W_{RGBXY} has 6 non-zeros among the (typically) 2000–5000 RGBXY convex hull vertices, which is also as sparse as possible to recover a point in RGBXY-space. The sparsity of the product of the two matrices depends on which 3D tetrahedra the 6 RGBXY convex hull vertices fall into.

In Figure 7, we compare the running time of additive mixing layer decomposition techniques. We ran our proposed RGBXY approach and additive mixing variant of Tan et al. [2016] on 100 images under 6 megapixels with an average palette size of 6.95 and median palette size of 7. Computation time for our approaches includes palette selection (RGB convex hull simplification). The performance of the modified Tan et al. [2016] is somewhat unpredictable, perhaps owing to the varying palette sizes. Aksoy et al. [2017]’s performance is the fastest previous work known to us. The performance data for Aksoy et al.’s algorithm is as reported in their paper and appears to scale linearly in the pixel size. Their algorithm was implemented in parallelized C++. Zhang et al. [2017]’s sole performance data point is also as reported in their paper. Our approach and our implementation of Tan et al. [2016] were written in non-parallelized Python using NumPy/SciPy and their wrapper for the QHull convex hull and Delaunay tessellation library [Barber et al. 1996]. (Our approach could be parallelized by dividing the image into tiles, since the convex hull of a set of convex hulls is the same as the convex hull of the underlying data.) A working implementation of the RGBXY decomposition method can be found in Figure 5.

Because of its scalability, we also ran our proposed RGBXY approach on an additional 70 large images between 6 and 12 megapixels, and an additional 6 extremely large images containing 100 megapixels (not shown in the plot). The 100 megapixel images took on average 12.6 minutes to compute. Peak memory usage was 15 GB. Aksoy et al. [2017] reported that their approach took 4 hours and 25 GB of memory to decompose a 100 megapixel image. Our running times were generated on a 2015 13" MacBook Pro with a 2.9 GHz Intel Core i5-5257U CPU and 16 GB of RAM.

The fast performance of our approach is due to the fact that the number of RGBXY convex hull vertices Q is virtually independent of the image size and entirely independent of the palette size. Finding the simplex that contains a point is extremely efficient (a matrix multiply followed by a sign check) and scales well. Our algorithm's performance is more correlated with the number of RGBXY convex hull vertices and tessellated simplices. The three outliers for our algorithm (the red dots somewhat above the others in the performance plot) are a testament to the occasional unpredictable nature of this.

4 COLOR HARMONIZATION

In the following we describe our palette-based approach to color harmonization and color composition. Our work is inspired by the same concepts and goals as related previous work [Cohen-Or et al. 2006]. However, we also aim for a simpler and more compact representation that can express additional operations and be applied directly to palettes. First, we explain how we fit and enforce classical harmonic templates. Next, we describe how our framework can be used for other color composition operations.

4.1 Template fitting

Figure 2 shows our new axis-based templates compared to the sector-based ones from Tokumaru et al. [2002]. For our results in this paper we use seven templates T_m , $m = 1 \dots 7$. A template is defined by $T_m^j(\alpha)$, where j is the index of each axis (the total number of axes varies between templates), and α is an angle of rotation in hue. While our templates are valid in any circular (or cylindrical) color space (e.g. HSV), we apply them in LCh-space (Lightness, Chroma, and hue) to match human perception.

Given an image I and its extracted color palette P , we seek to find the $T_m(\alpha)$ that is closest to the colors in P in the Ch plane. For that, we find the closest axis to each color, and solve for the global rotation and additional angles that define the template. We define the distance D between a palette P and a template $T_m(\alpha)$ as:

$$D(P, T_m(\alpha)) = \sum_{i=1}^{|P|} W(P_i) \cdot L(P_i) \cdot C(P_i) \cdot |H(P_i) - T_m^{j^*}(\alpha)| \quad (1)$$

$$j^* = \arg \min_{j=1 \dots \# \text{axes}} |H(P_i) - T_m^j(\alpha)|$$

where j^* is the axis of template $T_m(\alpha)$ that is closest to palette color P_i . $|\cdot|$ measures the difference in Hue angle. Note that for the *analogous* template, any palette color inside that arc area will be zero distance to the template. $W(P_i)$ is the contribution of color P_i to all the pixels in image, computed as the sum of all the weights for layer i and normalized by the total number of pixels in the image.

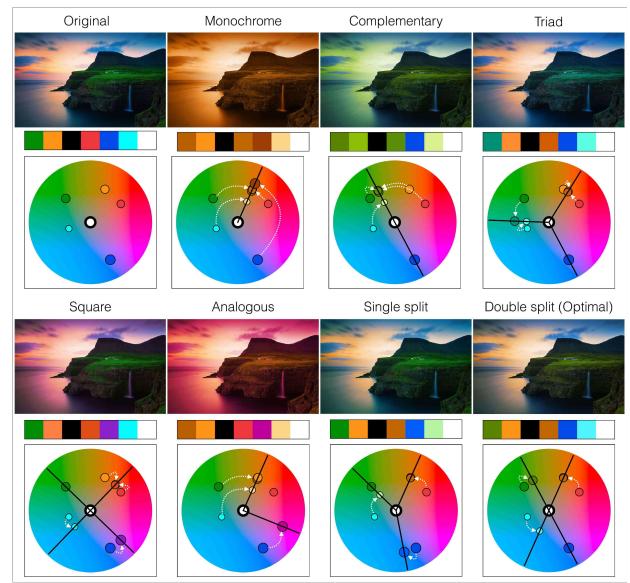


Fig. 8. Results of our harmonic templates fit to the an input image. We can see how each of them is able to provide a balanced and pleasing look when the harmonization is fully enforced. In an interactive application, the user can control the strength of harmonization, which interpolates the hue rotation of each palette color.

$W(P_i)$ promotes the template to be better aligned with the relevant colors of the image. When using color palettes that do not come from images, $W(P_i)$ is the same for each color and can be discarded. The lightness $L(P_i)$ and chroma $C(P_i)$ of the color are also used as weights so that we measure the arc distance around the color wheel (the angular change scaled by radius). The darker the color or the less saturated, the smaller the perceived change per hue degree.

We use a brute-force search to find the optimal global rotation angle α_m^* fitting a template $T_m(\alpha)$ to a palette P :

$$\alpha_m^* = \arg \min_{\alpha} D(P, T_m(\alpha)) \quad (2)$$

Monochrome, *complementary*, *triad* and *square* templates have only one degree of freedom, so we search the global rotation every 1 degree in $[0, 360]$. For *analogous*, *single split* and *double split* we allow an additional degree of freedom (angle between axes), which we allow $[-15, 15]$ degrees. In this case, $\alpha_m^* = [\alpha_{m,1}^*, \alpha_{m,2}^*]$. With $\alpha_{m,1}^*$ being the optimal global rotation, and $\alpha_{m,2}^*$ the optimal angle between axes. Given that palettes are typically small (less than 10 colors), our brute force search is very fast (less than a second).

Once a template is fit, we harmonize the input image by using $T_m(\alpha_m^*)$ to move the colors in P closer to the axis assignment that minimizes equation 1. We leverage the image decomposition to recolor the image. Because we use a spatially coherent image decomposition, no additional work is needed to prevent discontinuous recoloring as in Cohen-Or et al. [2006]. Figure 8 shows different harmonic templates enforced over the same input image. Additional examples can be found in the supplementary material. Users can

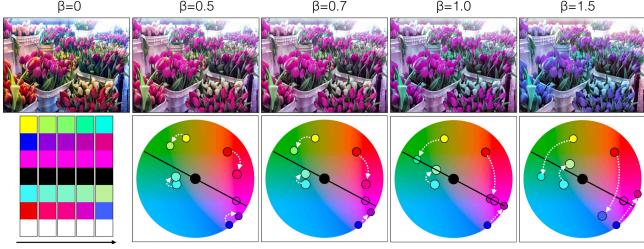


Fig. 9. Results from enforcing a given template $T_m(\alpha^*)$ with varying degrees of strength β . Bottom left shows the consistent palette interpolation across $\beta = [0, 1.5]$. Even beyond $\beta = 1$ (full harmonization), results remain predictable.

control the strength of harmonization via an interpolation parameter, where $\beta = 0$ leaves the palette unchanged and $\beta = 1$ fully rotates each palette color to lie on its matched axis (Figure 9). In the LCh color space, this affects hue alone.

Depending on the colors in P , some templates are a better fit than others as measured by Equation 1. We can determine the optimal template T_m^* automatically:

$$T_m^* = \arg \min_{T_m} D(P, T_m(\alpha^*)) \quad (3)$$

Depending on the palette size or its distribution, some axes may end up without any color assigned to them. We deem those cases not compliant with the intended balance of the harmonic template and remove them from this automatic selection.

Figure 10 shows the best fitting template for a set of images, and the fully harmonized result. More examples can be found in the supplementary material. We compare our results with harmonizations from previous works in Figure 11. While our result is clearly different, it arguably produces a more balanced result. Cohen-Or et al. [2006] demonstrated harmonization between different parts of an image using masks or harmonization of image composites. We provide comparisons for this scenario in Figure 12.

4.2 Beyond hue

Compared with sector-based templates, our axes are straightforward to use for other constraints in LCh.

Color-based contrast As part of his seminal work on color composition for design, Itten [1970] described additional pleasing color arrangements to create contrast. In contrast with sector-based templates, it is straightforward to model them with our axis-based representation. Here is the exhaustive list of Itten’s additional contrasting color arrangements and how they fit into our framework:

- Hue: *Triad* template aligned with the RGB primaries. No need to solve for α^* .
- Light-dark: *analogous* or *monochrome* template. Potential additional constrains for L.
- Complementary: same as *complementary* template.
- Simultaneous: *complementary* template, plus the axis with the smaller overall W scales down its chroma.
- Saturation: *analogous* or *monochrome* template. Potential additional constrains for C.

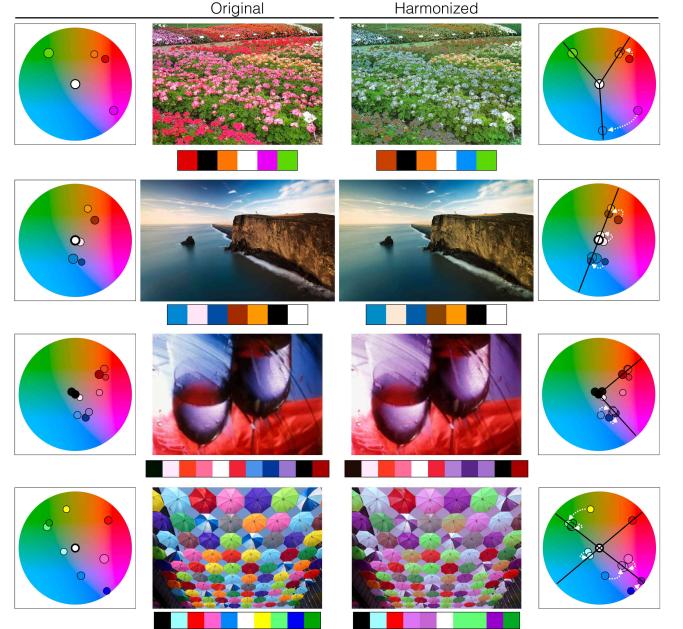


Fig. 10. Examples of optimal templates for different images, and the fully harmonized results they produce.

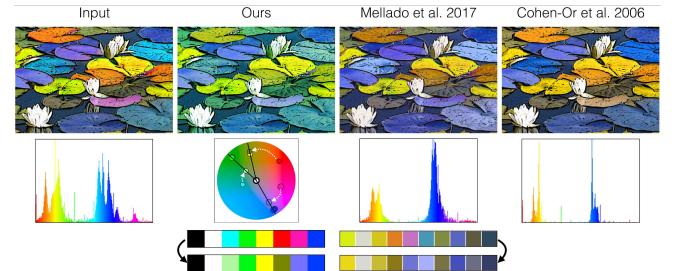


Fig. 11. Comparison of harmonizations using the best fitting template from different methods. Cohen-Or et al. [2006] fit a *complementary* (*I type*) template in HSV space, producing unexpected changes in lightness for some colors. Mellado et al. [2017] formulated the same harmonic template using their framework, again in HSV but including additional constraints to preserve lightness. Our optimal template is a *single split*, and our rotations in LCh directly preserve lightness. Our result looks different but arguably more balanced. This image harmonized via our other templates can be found in the supplementary material. None of them look similar to these, but all of them look more *harmonic*.

- Extension: solve for L so the total sum of $L(P_i)W(P_i)$ for each axis j in $T_m^j(\alpha^*)$ is the same.
- Cold-warm: a *complementary* template whose axis is aligned perpendicular to the cold-warm divide (). The cold-warm divide is the complementary axis shown in Figure 2.

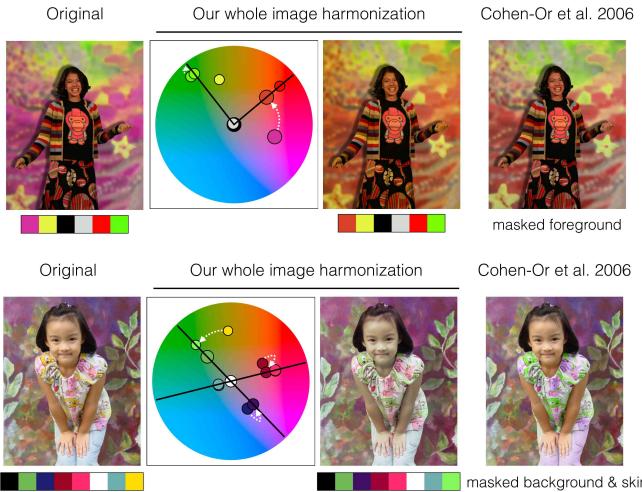


Fig. 12. Comparison with the masked results from Cohen-Or et al. [2006]. In the top row, Cohen-Or et al. harmonized the background to match the colors of the masked foreground person. We achieve comparable results without masking, and better preserve the background’s luminance. In the bottom row, Cohen-Or et al. masked the background and skin, harmonizing the girl’s clothing to match the background. The clothing, however, ends up overly bright. Our unmasked harmonization produces a closer match of the clothing to the background, though the unmasked skin takes on a slight greenish cast. This effect is most noticeable when viewed side-by-side. When viewed independently, color constancy may come into play, and our result looks natural and harmonic without the need to mask.

5 COLOR TRANSFER

We also use our palette extraction, image decomposition and harmonic templates to explore new color transfer methods with different goals.

5.1 Template-based color transfer

From our previous section, we can see how harmonic templates carry important information about the color distribution in a palette or an image. In the following we explore how to transfer that information and the results obtained.

Template alignment Given an input image I and a reference image R , we already know how to extract their palettes P^I and P^R , and estimate their optimal templates, $T_I(\alpha_I^*)$ and $T_R(\alpha_R^*)$. After fitting, we can use the sum of weights of each color $W(P_i)$ matched to an axis as the influence of that axis within its template. With this, we have an estimate of the main axis for each template—the one with the greatest influence over the image. This simple procedure helps to establish a straightforward match between palettes, something we can leverage to find the global rotation γ that aligns T_I with T_R . Next, we can apply γ to globally rotate the colors of the input image palette P^I and then harmonize them to the target’s template $T_R(\alpha_R^*)$ fully ($\beta = 1$). This method achieves results where I is recolored so it is harmonic with respect to R , taking into account the overall relevance of each color of the palette. Figure 13 shows results of this approach. We found that this method is good for matching

dominant colors, which works well for images whose contents do not have real reference colors (e.g. graphics design or man-made objects).

Template fitting When the final recoloring should better preserve the input image colors, a more conservative method can be formulated. In this case, we harmonize the input image colors P^I directly to the reference image’s best-fitting template $T_R(\alpha_R^*)$, without adjusting the template’s angle. We match palette colors to template axes according to equation 1. Figure 13 shows results from this method.

After rotating the hues of P^I with any of the proposed methods, we attempt to match lightness and chroma between palettes by scaling the lightness and chroma of each palette color to that the average L/C of the input and reference palette colors match.

5.2 Non-templed color transfer

We also build on top of our palettes and their perceptual distribution in LCh to explore other color transfer methods. We found it easy to formulate color transfer goals that can accommodate other uses beyond color balance.

The first option is a conservative mapping, where the colors in P^I map to the nearest ones from P^R , based on CIEDE 2000 distance in LCh color space. From this mapping, we only change the hues in P^I , adjusting lightness and chroma globally as described above. Figure 14 shows some results from this method. This very conservative transfer may leave out some of the colors in P^R , but it can be useful when a minimal change is desired.

To ensure that all colors are transferred from R , our second proposal performs a repeated bipartite matching from P^I to P^R (eliminating input colors matched in the previous round until all reference colors are matched), again using the CIEDE 2000 distance in LCh color space as bipartite edge weights. This ensures that all reference colors will be mapped to by at least one input color. After the mapping is found, lightness, chroma, and hue are directly transferred between pairs, producing very colorful results that do not preserve gradients or other structures of I .

Our third proposal lies between the two previous ones: the bipartite edge weight between two colors is scaled by the weight of the input palette color, thus taking the influence of colors into account. The same repeated bipartite matching ensures all colors are mapped. Similar to the first option, we only transfer hue directly between pairs, and adjust lightness and chroma globally between P^I and P^R .

Figure 14 shows comparisons between these three methods. We also compared them with previous work in Figure 15.

6 CONCLUSION

We have presented a very efficient, intuitive and capable framework for color composition. It allows us to formulate previous and novel approaches to color harmonization and color transfer with very robust results. Our palette manipulations can be plugged into any palette-based system. Our image decomposition can be used generally by artists for manual editing or in other algorithms.

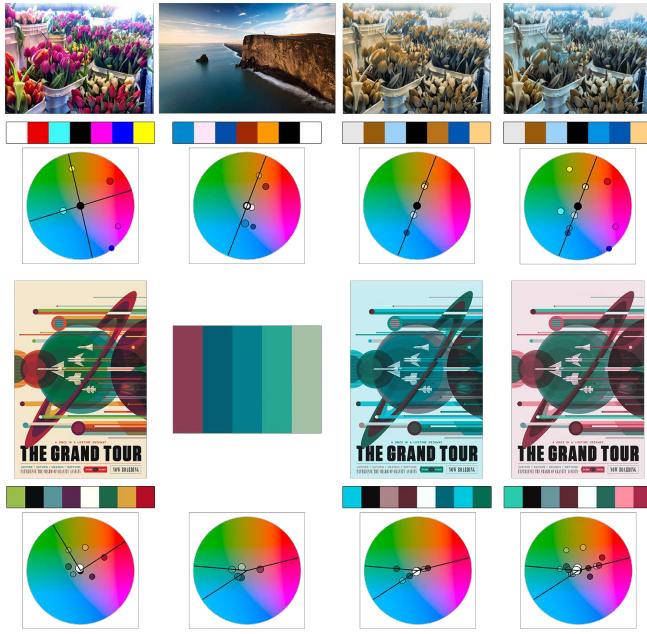


Fig. 13. Comparison between our two template-based color transfer methods. Third row shows how aligning primary axes (method 2) transfers better the overall color proportions, something that tends to work better for content without critical color semantics. On the other hand, the more conservative method 3 (fourth row) produces more subtle changes that still match the colors from the reference. From left to right: input image, reference, primary axis alignment and harmonization, and straight harmonization.

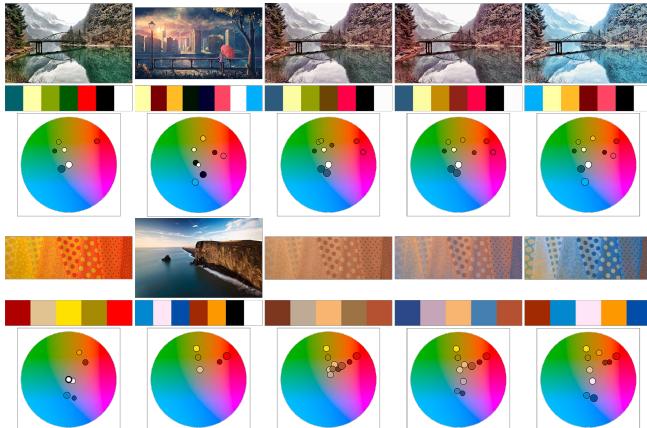


Fig. 14. Results from our direct palette-based color transfer. From left to right: input image, reference, first method from Section 5.2, third method, and second. From left to right, it can be seen the increasing amount of colors transferred, and the colorfulness achieved when not preserving lightness.

6.1 Limitations

During our performance tests for the image decomposition, we found isolated cases where the computation of the 5D convex hull takes somewhat longer than usual. We believe it is due to very

specific color distributions (3 out of 170 tested images), but we would like to study the phenomenon in more depth.

There are also cases for the templated color transfer where a the input palette tries to match a reference palette with a higher number of axes. This is usually a case of colorization (adding more colors than the existing ones) that we currently handle with varying success depending on the input color palette. These cases may need more elaborate formulations for the transfer. Our non-templated transfers help alleviate the problem.

Because there is not a universal color theory, the concepts we leverage for our methods may not work for everybody. In fact, we already saw clear differences in our results with respect to previous work, even building on top of comparable foundations. This may expose the need for perceptual studies evaluating the perceived quality of results from different algorithms, but also the need for intuitive frameworks like ours, enabling users to use and interact with color harmony despite only passing familiarity.

6.2 Future work

Image decomposition Inspired by Lin et al. [2017b], we wish to explore the use of superpixels to see if we are able to achieve greater speed ups. We also plan to explore replacing the current Delaunay barycentric coordinates with Mean Value Coordinates, as they do not require a Delaunay tessellation and may have additional smoothness benefits. We also wish to explore parallel and approximate convex hull algorithms.

Harmonization Apart from our current chrome and hue (Ch) harmonic templates, we want to explore full 3D LCh templates for more complex palettes and compositions [Moon and Spencer 1944; Tokumaru et al. 2002]. While altering lightness and chroma can have undesired effects on images (gradients, contrast), we believe this can lead to more varied color palettes and themes.

Other color-aware applications We believe that our templates may carry semantic structure that we would like to keep exploring in the future. Among others, we believe this can enable higher level and more intuitive image search algorithms, where images or palettes can be used transparently to retrieve other images and color themes for design. We also also plan to extend our framework to video, exploring the spatial-temporal coherence of our templates, to potentially provide more robust color grading methods.

REFERENCES

- Adobe. 2018. Adobe Color CC. (2018). <http://color.adobe.com>
- Elad Aharoni-Mack, Yakov Shambik, and Dani Lischinski. 2017. Pigment-Based Recoloring of Watercolor Paintings. In *NPAR*.
- Yağız Aksøy, Tunç Ozan Aydin, Aljoša Smolić, and Marc Pollefeys. 2017. Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 19.
- B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot. 2016. Automatic Texture Guided Color Transfer and Colorization. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering (Expressive '16)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 21–32. <http://dl.acm.org/citation.cfm?id=2981324.2981328>
- C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* 22, 4 (Dec. 1996), 469–483.
- Yoann Baveye, Fabrice Urban, Christel Chamaret, Vincent Demoulin, and Pierre Hellier. 2013. Saliency-Guided Consistent Color Harmonization.. In *CCIW*. 105–118.



Fig. 15. Comparison with some previous works, Arbelot et al. [2016] (first and second rows) and Pitié et al. [2007] (bottom row). From left to right: input image, reference, templated method performing alignment of primary axes, straight harmonization, our methods from Section 5.2(1, 3, 2), and the results from the others. Our methods provide some results closer to [Arbelot et al. 2016], especially straight harmonization (column 4) and the third direct transfer method (column 6), our most conservative ones. Compared to [Arbelot et al. 2016], our transfers do not capture the overall tone that well, but produce usable results nevertheless.

- Christel Chamaret, Fabrice Urban, and Lionel Oisel. 2014. Harmony-guided image editing. In *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2171–2173.
- Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based Photo Recoloring. *ACM Trans. Graph.* 34, 4 (July 2015).
- Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. 2006. Color Harmonization. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH '06)*. ACM, New York, NY, USA, 624–630. <https://doi.org/10.1145/1179352.1141933>
- Maurice D Craig. 1994. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing* 32, 3 (May 1994), 542–552. <https://doi.org/10.1109/36.297973>
- Michael S Floater. 2015. Generalized barycentric coordinates and applications. *Acta Numerica* 24 (2015), 161–214.
- Michael Garland and Paul S. Heckbert. 1997. Surface Simplification Using Quadric Error Metrics. 209–216.
- Yu Han, Chen Xu, George Baciu, Min Li, and Md Robiul Islam. 2017. Cartoon and texture decomposition-based color transfer for fabric images. *IEEE Transactions on Multimedia* 19, 1 (2017), 80–92.
- Yu Han, Dejun Zheng, George Baciu, Xiangchu Feng, and Min Li. 2013. Fuzzy region competition-based auto-color-theme design for textile images. *Textile Research Journal* 83, 6 (2013), 638–650. <https://doi.org/10.1177/0040517512452953>
- Xiaodi Hou and Liqing Zhang. 2007. Color conceptualization. In *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 265–268.
- Xing Huo and Jieqing Tan. 2009. An improved method for color harmonization. In *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*. IEEE, 1–4.
- Johannes Itten. 1970. *The elements of color*. John Wiley & Sons.
- Xujie Li, Hanli Zhao, Guizhi Nie, and Hui Huang. 2015. Image recoloring using geodesic distance based color harmonization. *Computational Visual Media* 1, 2 (2015), 143–155.
- Sharon Lin, Matthew Fisher, Angela Dai, and Pat Hanrahan. 2017a. LayerBuilder: Layer Decomposition for Interactive Image and Video Color Editing. *arXiv preprint arXiv:1701.03754* (2017).
- Sharon Lin, Matthew Fisher, Angela Dai, and Pat Hanrahan. 2017b. LayerBuilder: Layer Decomposition for Interactive Image and Video Color Editing. *CoRR abs/1701.03754* (2017). <http://arxiv.org/abs/1701.03754>
- Sharon Lin and Pat Hanrahan. 2013. Modeling how people extract color themes from images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3101–3110.
- Nicolas Mellado, David Vanderhaeghe, Charlotte Hoarau, Sidonie Christophe, Mathieu Brédif, and Loïc Barthe. 2017. Constrained palette-space exploration. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 60.
- Parry Moon and Domina Eberle Spencer. 1944. Geometric Formulation of Classical Color Harmony*. *J. Opt. Soc. Am.* 34, 1 (Jan 1944), 46–59. <https://doi.org/10.1364/JOSA.34.000046>
- Bryan S Morse, Daniel Thornton, Qing Xia, and John Uibel. 2007. Image-based color schemes. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, Vol. 3. IEEE, III–497.
- Bharat Munshi and Navjyoti Singh. 2015. Palette based colour transfer using image segmentation. In *Image and Vision Computing New Zealand (IVCNZ), 2015 International Conference on*. IEEE, 1–6.
- RMH Nguyen, B Price, S Cohen, and MS Brown. 2017. Group-Theme Recoloring for Multi-Image Color Consistency. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 83–92.
- Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2011. Color compatibility from large datasets. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 63.
- Huy Phan, Hongbo Fu, and Antoni Chan. 2017. Color Orchestra: Ordering Color Palettes for Interpolation and Prediction. *IEEE Transactions on Visualization and Computer Graphics* (2017).
- François Pitié, Anil C. Kokaram, and Rozenn Dahyot. 2007. Automated Colour Grading Using Colour Distribution Transfer. *Comput. Vis. Image Underst.* 107, 1–2 (July 2007), 123–137. <https://doi.org/10.1016/j.cviu.2006.11.011>
- Pedro V. Sander, Xianfeng Gu, Steven J. Gortler, Hugues Hoppe, and John Snyder. 2000. Silhouette Clipping. In *Proceedings of ACM SIGGRAPH*, 327–334.
- N. Sawant and N. J. Mitra. 2008. Color Harmonization for Videos. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 576–582.
- Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. 2010. Multi-scale image harmonization. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 125.
- Jianchao Tan, Stephen DiVerdi, Jingwan Lu, and Yotam Gingold. 2017. Pigmento: Pigment-Based Image Analysis and Editing. *arXiv preprint arXiv:1707.08323* (2017).
- Jianchao Tan, Jyh-Ming Lien, and Yotam Gingold. 2016. Decomposing Images into Layers via RGB-space Geometry. *ACM Trans. Graph.* 36, 1, Article 7 (Nov. 2016), 14 pages. <https://doi.org/10.1145/2988229>
- Zhen Tang, Zhenjiang Miao, and Yanli Wan. 2010. Image composition with color harmonization. In *25th International Conference of Image and Vision Computing New Zealand*.
- Masataka Tokumaru, Noriaki Muranaka, and Shigeru Imanishi. 2002. Color design support system considering color harmony. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, Vol. 1. IEEE, 378–383.
- Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. 2017. Deep Image Harmonization. *ArXiv e-prints* (Feb. 2017). [arXiv:cs.CV/1703.00069](https://arxiv.org/abs/1703.00069)
- Baoyuan Wang, Yizhou Yu, Tien-Tsin Wong, Chun Chen, and Ying-Qing Xu. 2010. Data-driven image color theme enhancement. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 146.
- Stephen Westland, Kevin Laycock, Vien Cheung, Phil Henry, and Forough Mahyar. 2007. Colour harmony. *JAI-C-Journal of the International Colour Association* 1 (2007).
- Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. 2017. Palette-Based Image Recoloring Using Color Decomposition Optimization. *IEEE Transactions on Image Processing* 26, 4 (2017), 1952–1964.