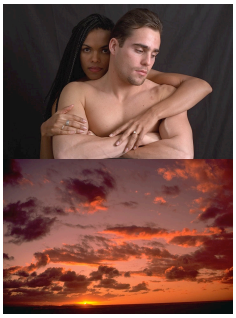
EUROGRAPHICS 2006 / E. Gröller and L. Szirmay-Kalos
(Guest Editors)

Volume 25 (*2006*), *Number 3*

Cross Dissolve Without Cross Fade: Preserving Contrast, Color and Salience in Image Compositing

Mark Grundland † Rahul Vohra Gareth P. Williams Neil A. Dodgson

Computer Laboratory, University of Cambridge, Cambridge, United Kingdom



Originals: [60%, 40%]

Linear Cross Dissolve

Salience-Preserving Cross Dissolve

Abstract

Linear interpolation is the standard image blending method used in image compositing. By averaging in the dynamic range, it reduces contrast and visibly degrades the quality of composite imagery. We demonstrate how to correct linear interpolation to resolve this longstanding problem. To provide visually meaningful, high level control over the compositing process, we introduce three novel image blending operators that are designed to preserve key visual characteristics of their inputs. Our contrast preserving method applies a linear color mapping to recover the contrast lost due to linear interpolation. Our salience preserving method retains the most informative regions of the input images by balancing their relative opacity with their relative saliency. Our color preserving method extends homomorphic image processing by establishing an isomorphism between the image colors and the real numbers, allowing any mathematical operation defined on real numbers to be applied to colors without losing its algebraic properties or mapping colors out of gamut. These approaches to image blending have artistic uses in image editing and video production as well as technical applications such as image morphing and mipmapping.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

Image compositing [Bri99] is the process of combining component images to make an integrated composite image. Compositing is a basic user task in image editing and video production. It is a full time occupation for many artists working with digital media. Composite creation involves *image matting*, selecting the component image parts to combine, and *image blending*, applying the right technique to combine them. Interactive image compositing has two primary uses: *image cloning* (cut-and-paste) places mostly opaque image components one over another whereas *image mixing* (cut-and-combine) merges mostly semitransparent image compo-

† Contact: Mark@eyemaginary.com
http://www.eyemaginary.com/compositing

nents together. Image cloning relies on very accurate image mattes whereas image mixing benefits from specialized image blending operators and can often use far simpler image mattes. Most research related to interactive image compositing has focused on image cloning, facilitating such tasks as extracting a foreground object from its background and pasting it into another image without revealing any seams. Though modern image compositing systems provide productive tools for image cloning, they do not address some of the aesthetic challenges posed by image mixing. Our work presents novel image blending operators designed for image mixing in image and video editing systems. We aim to create composite images that preserve the key visual characteristics of their component images. Our work reformulates the linear interpolation operation used in standard image compositing, showing how to better account for contrast, color and salience with only a moderate amount of extra computation. Compared to classical methods [BA83, PD84] for color image blending, our techniques exhibit superior image quality. Compared to special purpose methods [PGB03, ADA*04] for color image blending, which are primarily used for image cloning, our techniques exhibit greater flexibility by being able to simultaneously blend multiple, independent images with variable degrees of transparency. Expanding the repertoire of expressive techniques available to artists, our methods offer improved, high level control over the appearance of composite images without the need for painstaking, low level manipulation of the individual image mattes.

1.1. The Problem

The way image mixing is performed [Bri99] has changed remarkably little since the invention of image compositing. Linear interpolation, originally used for its computational efficiency, is still the standard method of blending semitransparent imagery. For all its simplicity, this conventional approach has a clear drawback. Whenever images are blended by linear interpolation, the result loses color contrast. Linear blending causes fading, vivid colors turn dull, highlights and shadows turn gray, and details are obliterated. Artists compensate by manually adjusting contrast, color, and mattes.

When the components of a composite are mainly opaque, linear interpolation suffices to soften the transitions between them. However, what is desirable for antialiasing is not necessarily appropriate for rendering partial transparency. In the post-production of film special effects, composites may be constructed from as many as over a hundred image layers. The composites often need contrast and color correction even when their components have already been corrected. A similar problem occurs when using linear interpolation to render a transition between film sequences. In cinematography, this effect is interchangeably described as a cross dissolve or a cross fade. When the transition only lasts a fraction of a second, the fading is hardly perceptible. However, when the transition is prolonged, the poor color contrast of the intermediate frames becomes apparent to the viewer. Our techniques serve to integrate automated color and contrast correction into the image compositing process.

Suppose a composite C is created by merging the foreground A with the background B according to their relative opacity w which may vary spatially or temporally. The fundamental equation of linear image compositing is:

$$C = wA + (1-w)B \quad \text{for} \quad 0 \leq w \leq 1 \tag{1}$$

Linear interpolation outputs a convex linear combination of its inputs, a weighted sum where the weights are positive and sum to one. Recall that a nondegenerate linear combination of bounded, identically distributed signals, with nonzero mean, cannot simultaneously maintain their expected intensity and their expected variation. In general, linear averaging reduces variation. Image smoothing by linear interpolation reduces sharpness, as averaging is applied to adjacent pixels of the same image. Image blending by linear interpolation reduces contrast, as averaging is applied to corresponding pixels of different images. Compared to the visual artifacts of smoothing, little attention has been paid to the visual impact of blending. Qualitatively, we see that linear interpolation compresses color histograms, enlarging the peak nearest to the mean whilst shrinking the tails. Quantitatively, we measure the global contrast of each color channel using its standard deviation σ_C as an indicator of the expected color difference $\sigma_C = \sqrt{\frac{1}{2}\mathrm{E}[(C_1 - C_2)^2]}$ between a random pair of image pixels C_1 and C_2. Though the average color of a composite is the weighted average of the colors of its components, the same relation does not hold true for its contrast, except in the trivial case of perfectly correlated components $\rho_{AB} = 1$. Assuming constant weights, as used in cross dissolve, linear image compositing inevitably acts to reduce contrast:

$$\sigma_C{}^2 = w^2\sigma_A{}^2 + (1-w)^2\sigma_B{}^2 + 2w(1-w)\sigma_A\sigma_B\rho_{AB} \tag{2}$$

$$\sigma_C \leq w\sigma_A + (1-w)\sigma_B \tag{3}$$

1.2. Our Solution

We adapt linear interpolation to the requirements of image mixing. Without forcing the artist to alter the input images, we can change three aspects of the fundamental equation (1): the result, the operators, and the weights. Our contrast preserving method applies a linear transformation to correct the composite that results from linear interpolation. In effect, the composite recovers the contrast lost due to linear interpolation. Our color preserving method replaces linear addition and multiplication by nonlinear operators defined through an isomorphism between the image colors and the real numbers. In effect, the composite maintains strong vibrant colors over subdued neutral shades. Our salience preserving method calibrates the weights to reflect the relative salience of the components. In effect, the composite retains the most salient aspects of each component image. All our techniques are readily applied to a multiresolution image representation. As shown in our experiments, effective composites can be

produced by simply using a constant factor, a smooth gradient, or a rough matte to specify the contributions of the components. To jointly control contrast, color and salience, we can composite together the results of all three methods.

2. Related Work

From the advent of photography, images have been combined with images, an artistic technique known as photomontage [Ade86]. Artistically, photomontage applies contrasting images to juxtapose ideas, overlaying images to express layers of meaning. There is a cinematic convention of superimposing images to convey thoughts, memories and dreams. Today, composites have become so ubiquitous that audiences hardly notice their presence. Digital image compositing stems from Smith and Catmull's 1977 invention of the alpha-channel [Smi95]. In 1984, Porter and Duff [PD84] enumerated the fundamental image blending operators. These ideas are based on linear interpolation and have shaped the development of computer graphics, becoming part of many image editing systems and storage formats.

Our work addresses the lack of general purpose image blending models. There are two basic mathematical models of combining information, averaging and selection, and two basic physical models of light interaction, absorption and emission. When averaging linearly interpolates between pixel values, the composite tends to lose contrast with the inclusion of each component. In effect, the composite obscures fine details. Conversely, when selection keeps the pixel value with the highest absolute magnitude, with neutral gray at the origin, the composite tends to gain contrast with the inclusion of each component. In effect, the composite exhibits spurious discontinuities. If images are treated as stacked sheets of a light absorbing material, the composite tends to darken with the inclusion of each component. Conversely, if images are treated as stacked sheets of a light emitting material, the composite tends to brighten with the inclusion of each component. Our image blending techniques instead place the degree of contrast gain or loss under user control, without bias in favor of any particular color.

Many previously proposed alternatives to linear interpolation are not suitable for compositing color images. For example, applying models of mixing color pigments, such as watercolors [CAS*97], requires specifying paint parameters that are normally unavailable in image compositing. A level-set approach [Whi00] can model shape changes in grayscale image blending, but does not readily extend to color images.

Image blending can be performed on different image representations. As images can be described with gradient fields and boundary conditions, blending can be performed by combining color gradients instead of color values [PGB03, ADA*04]. However, these approaches may exhibit color artifacts [PGB03] or require additional user input to ensure color fidelity [ADA*04]. They are much more complicated to implement and calculate than our techniques. Alternatively, multiresolution image pyramids [BA83] filter and subsample images into successive levels of detail, allowing each level to be blended independently. Our techniques are designed for use with image pyramids [BA83].

Image stitching [Sze05] adjoins image fragments without revealing the seams. These methods are generally more suited for cloning opaque images; our techniques are designed to mix semitransparent images. We use continuous image mattes to specify component image opacity while image stitching often starts with binary image masks to describe the shape of component images. However, linear interpolation is often used to blend fragments along the seams. Accordingly, several authors [NOT98, KSE*3, Sze05, ZLP*06] report blurring and contrast loss.

Image morphing [Wol98] and image based rendering [SD96] rely on image blending to mix images. In these contexts, when user specified image correspondences cannot account for image differences, such as dissimilar textures, image blending by linear interpolation causes well known fading and ghosting artifacts [SD96, Whi00]. Our salience preserving blending technique can reduce these problems (Figure 3). It could also prove applicable to rendering summarizations of video [MB96] and image collections [RKK*05].

3. Method

Our task is to create a composite image C_p from component images A_p^n with image mattes specifying their relative opacities $0 \leq w_p^n \leq 1$. Linear image compositing processes the RGB color channels of each pixel p in the same way:

$$C_p = \sum_n w_p^n A_p^n \quad \text{for} \quad \sum_n w_p^n = 1 \tag{4}$$

We modify linear interpolation in three ways. To preserve contrast, we correct the output C'_p. To preserve color, we change the meaning of addition $\oplus$ and scalar multiplication $\otimes$. To preserve salience, we alter the weights $w_p^{n\prime}$. These methods complement each other and could be combined.

3.1. Contrast Preserving Image Blending

The goal of contrast preserving blending is to produce a composite with the same color contrast as its components. Linear interpolation reduces contrast by compressing the resulting color distribution around its mean. Our method remedies this defect by linearly stretching each color channel around its mean, so that the composite reproduces both the average color and contrast of its components. Unlike the linear color mappings used in image recoloring [RAG*01], our method recovers color contrast without causing an overall color shift. Alternatively, color histogram specification [GD05] can force composites to reproduce the colors of their components by setting the composite color distribution to be a weighted combination of the component distributions. Yet,

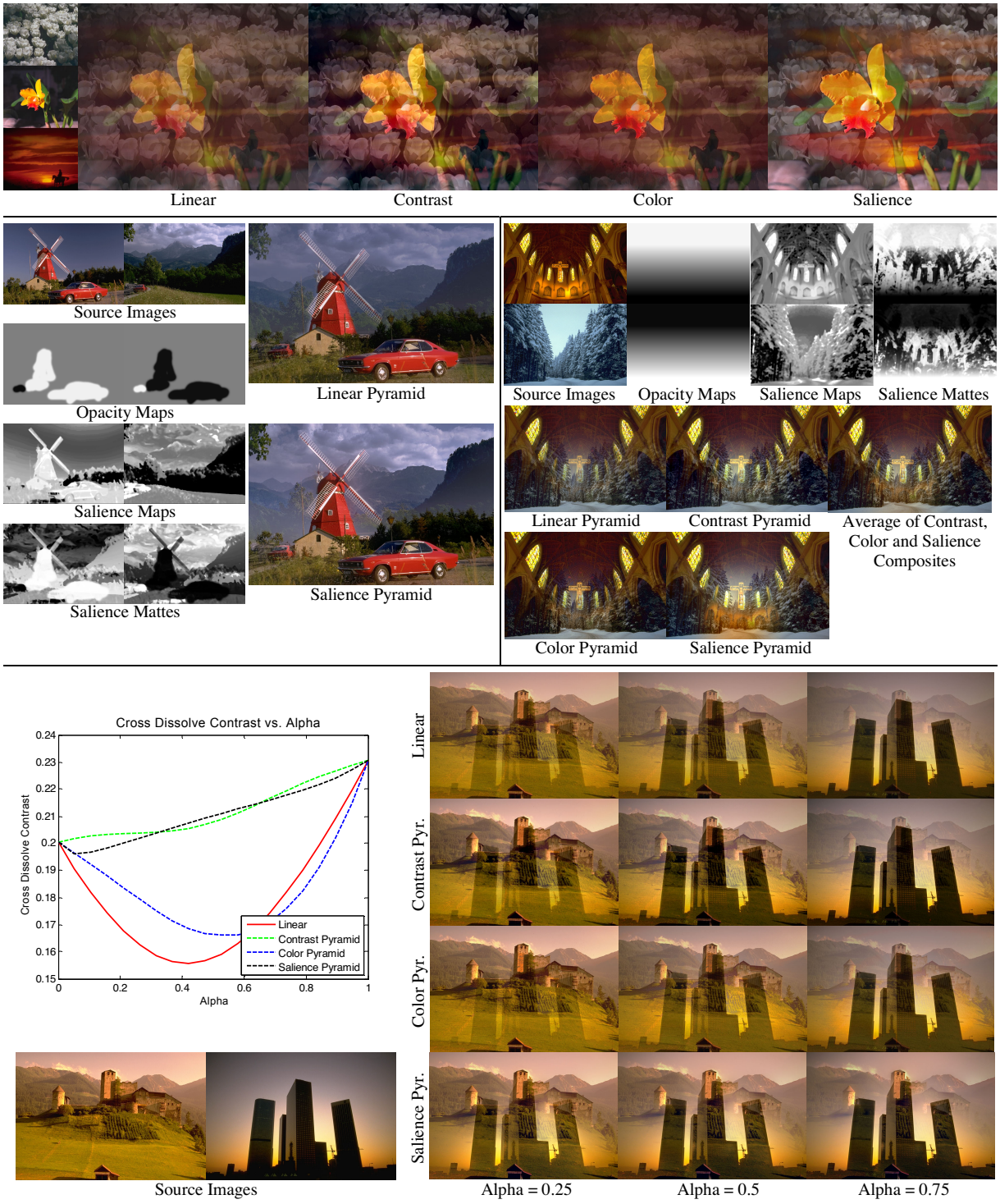


Figure 1: *Top: Linear, contrast, color, and salience preserving image blending methods are used to composite three equally weighted images. Middle, left: Salience preserving pyramid blending is used with simple, user defined opacity maps. Middle, right: Different image blending methods are applied using gradient opacity maps. Bottom: Different ways to perform a uniform cross dissolve between a pair of images are compared. The graph illustrates how the contrast and salience preserving pyramid blending methods better maintain contrast throughout the cross dissolve than linear and color preserving pyramid blending.*

such a restrictive approach can be unsatisfactory: combining a low contrast, dark image with a low contrast, bright image would result in a high contrast, noisy image. Hence, we chose to preserve color contrast instead of color distribution.

The standard deviation σ^n of a color distribution is a global measure of color contrast. The effect of each pixel A_p^n is proportional to its relative opacity w_p^n. In this way, transparent pixels $w_p^n = 0$ do not affect the outcome. We find the weighted mean μ^n and variance $(\sigma^n)^2$ of each component n:

$$\mu^n = \frac{\sum_p w_p^n A_p^n}{\sum_p w_p^n} \quad \text{and} \quad (\sigma^n)^2 = \frac{\sum_p w_p^n (A_p^n - \mu^n)^2}{\sum_p w_p^n} \tag{5}$$

We also find the weighted covariance σ^{nm} between each pair of components n and m, which is defined so that $\sigma^{nn} = (\sigma^n)^2$:

$$\sigma^{nm} = \frac{\sum_p \sqrt{w_p^n w_p^m}\,(A_p^n - \mu^n)\,(A_p^m - \mu^m)}{\sum_p \sqrt{w_p^n w_p^m}} \tag{6}$$

The relative contributions w_p^n of the component images A_p^n to the composite C_p produced by linear interpolation dictate its weighted variance $(\sigma_p)^2$ at pixel p. In this way, opaque pixels $w_p^n = 1$ keep their colors unchanged. We calculate:

$$(\sigma_p)^2 = \sum_{n,m} w_p^n w_p^m \sigma^{nm} = \sum_n (w_p^n \sigma^n)^2 + 2 \sum_{n<m} w_p^n w_p^m \sigma^{nm} \tag{7}$$

As a result of linear interpolation, the expected color μ_p at pixel p of the composite C_p is a weighted sum of the component mean colors μ^n. We want the composite contrast σ'_p at pixel p to be a weighted sum of the component contrasts σ^n:

$$\mu_p = \sum_n w_p^n \mu^n \quad \text{and} \quad \sigma'_p = \sum_n w_p^n \sigma^n \tag{8}$$

We linearly redistribute composite colors around their mean:

$$C'_p = \tau \frac{\sigma'_p}{\sigma_p} (C_p - \mu_p) + \mu_p \tag{9}$$

The contrast corrected composite C'_p preserves both the color μ_p and contrast σ'_p of its components. The desired degree of contrast enhancement is controlled by the parameter $\tau > 0$, with default $\tau = 1$. For example, $\tau = 2$ creates composites with twice the contrast of their components. However, when τ is set too high, our linear color mapping risks mapping some colors out of gamut. In practice, our procedure can often be simplified. If the weighted components $w_p^n A_p^n$ are known to be independent, then the covariance terms can be safely ignored, as $\sigma^{nm} \approx 0$ for $n \neq m$. If the components have constant opacities $w_p^n = w^n$, as in a uniform cross dissolve, the contrast correction is the same at each pixel and it can be calculated directly from the global statistics $\mu_p = \mu$ and $\sigma_p = \sigma$ of the composite C_p produced by linear interpolation.

3.2. Color Preserving Image Blending

The goal of color preserving blending is to produce composite images that capture the strong, vivid colors of their components. Viewers prefer vibrant, colorful images, but linear interpolation favors dull, neutral tones. In the standard linear

Figure 2: *Isomorphic image processing: color shift by color addition $\oplus$ and contrast change by scalar multiplication $\otimes$.*

model of color mixing, color addition and scalar multiplication are not closed operations. Interpolation is well defined but not extrapolation. General linear combinations of colors, where weights need not be positive or sum to one, can produce colors out of gamut, leading to irreversible detail loss in shadows, highlights, and saturated hues. Yet, as addition adjusts color and multiplication adjusts contrast, general linear combinations implement many useful image editing tasks. They include [HV94] compositing with pure black or white to adjust brightness, with pure gray to adjust contrast, with a grayscale image to adjust saturation, and with a smoothed image to adjust sharpness. For brighter colors and higher

contrast, artists [PR01, p.366-367] use general linear combinations to overweigh the components of a composite.

We propose a novel color algebra. We sacrifice the geometric properties of linear displacement in a color space to regain the algebraic properties of linear algebra in a vector space. Our color mixing model sensibly combines arbitrarily weighted colors without producing colors out of gamut. Up to the limits of its numerical implementation, it ensures that image compositing operations are associative and invertible. Unlike linear interpolation, it favors strong colors over neutral tones. Our operators have the following aesthetic goals:

Color Negation $\ominus$: Photographic color inversion transforms red, green, blue, and white into cyan, magenta, yellow and black. Hence, the negative of a hue is its complementary hue, the negative of a tone is its inverse tone, and the negative of neutral gray is neutral gray.

Color Addition $\oplus$: Similar colors reinforce and dissimilar colors counteract. Hence, adjacent hues form an intermediate hue, like colors combine to become more saturated, unlike colors combine to become less saturated, complementary hues and inverse tones cancel out to neutral gray, adding a light tone lightens, adding a dark tone darkens, and adding neutral gray to any color leaves it unchanged.

Scalar Multiplication $\otimes$: Negative factors invert colors, positive factors less than one reduce contrast by shifting colors toward neutral gray, positive factors greater than one raise contrast by shifting colors away from neutral gray, and a zero factor maps all colors to neutral gray. Hence, multiplying by half makes all colors more neutral while multiplying by two makes hues more saturated, light tones lighter and dark tones darker.

We wish to make operations on colors as straightforward and consistent as operations on real numbers. We show how any mathematical operation defined on real numbers can be applied to colors without losing its algebraic properties or mapping colors out of gamut. Our work extends homomorphic [OSS68, JP01] and isomorphic [SP83] grayscale processing to isomorphic color image processing. Previously, homomorphic color processing [Fau79], a forerunner of the lαβ color space [RAG*01], applied homomorphic processing to each color channel separately, which can lead to posterization artifacts when combining arbitrarily weighted colors. We address these limitations of previous methods by proposing a spherical color model with a parameterized color isomorphism. We define a continuous, bijective mapping $F:(0,1)^3\rightarrow\mathbb{R}^3$ between colors $a,b,c\in(0,1)^3$ and real values $x,y,z\in\mathbb{R}^3$. To implement a color operation $b=\textcircled{T}(a)$, we map input colors to real values $x=F(a)$, do the mathematical operation on the real values $y=T(x)$, and map them back to output colors $b=F^{-1}(y)$. Hence, color operations always result in valid colors, so that the final result of image processing is guaranteed to be a valid image. As each point in Euclidean space maps to a unique color in color space and vice versa, all stages of image processing have an unambiguous visualization. To endow colors with the algebraic structure of a normed vector space, we use our isomorphism F to define color addition and scalar multiplication with $w\in\mathbb{R}$:

$$a\oplus b=F^{-1}(F(a)+F(b)) \text{ and } w\otimes a=F^{-1}(wF(a)) \quad (10)$$

$$a\odot b=F(a)\cdot F(b) \quad \text{and} \quad \|a\|_\circ=\|F(a)\| \quad (11)$$

Similarly defined multiplication and division gives grayscale values the algebraic structure of an ordered field. We composite general linear combinations of images with $w_p^n\in\mathbb{R}$:

$$C_p=\bigoplus\nolimits_n w_p^n\otimes A_p^n=F^{-1}\left(\sum\nolimits_n w_p^n F(A_p^n)\right) \quad (12)$$

Our color model is a spherical color coordinate system. It has neutral gray at its origin, black and white at its poles, a hue circle at its equator, weak colors on the interior, and strong colors on the exterior. Spherical coordinates express perceptual properties: the azimuthal angle denotes hue and the polar angle denotes tone while the radius measures color strength. Indicating the darkness of a shadow, the lightness of a highlight, or the saturation of a hue, a color's strength is its distance from neutral gray divided by the maximal distance from neutral gray of any valid color with the same hue and tone angles. The zero element of color addition is neutral gray, $(a_R,a_G,a_B)\oplus(\frac{1}{2},\frac{1}{2},\frac{1}{2})=(a_R,a_G,a_B)$, so that color negation performs photographic inversion, $\ominus(a_R,a_G,a_B)=(1-a_R,1-a_G,1-a_B)$. We begin by linearly mapping RGB colors from $(a_R,a_G,a_B)\in[0,1]^3$ to $(a_R^*,a_G^*,a_B^*)\in(-1,1)^3$. As it is impossible to define a continuous, bijective mapping between the closed set $[0,1]^3$ and the open set $\mathbb{R}^3$, we apply a contraction by $\varepsilon=2^{-6}$ to turn the closed set $[0,1]^3$ into an approximation of the open set $(0,1)^3$. For our purposes, this has a negligible visible effect on quantized color values. For a rescaled RGB color, the maximal magnitude of its coordinates determines its color strength, $a_s=\max(|a_R^*|,|a_G^*|,|a_B^*|)$. The direction of its unit vector dictates its hue and tone angles. To map colors to real numbers, we rescale color strength using a continuous, bijective color scaling function $f_\lambda:[0,1)\rightarrow[0,\infty)$. Our mapping of colors to real numbers $x=F(a)$ and its inverse $a=F^{-1}(x)$ are specified:

$$(x_R,x_G,x_B)=(a_R^*,a_G^*,a_B^*)\frac{f_\lambda(\max(|a_R^*|,|a_G^*|,|a_B^*|))}{\|(a_R^*,a_G^*,a_B^*)\|} \quad (13)$$

$$(a_R^*,a_G^*,a_B^*)=(x_R,x_G,x_B)\frac{f_\lambda^{-1}(\|(x_R,x_G,x_B)\|)}{\max(|x_R|,|x_G|,|x_B|)} \quad (14)$$

$$a^*=(1-\varepsilon)(2a-1) \quad \text{and} \quad a=\tfrac{1}{2}+\tfrac{1}{2}(1-\varepsilon)^{-1}a^* \quad (15)$$

The color scaling function serves to determine color magnitude $\|a\|_\circ=\|F(a)\|=f_\lambda(a_s)=x_s$, the vector norm of the real valued color representation. In color preserving blending, for vivid colors to dominate neutral tones in color combinations, strong colors need to be assigned significantly greater color magnitude than weak colors. The color scaling function needs to be strictly increasing $f_\lambda'(a)>0$, convex $f_\lambda''(a)>0$, asymptotic $f_\lambda(1)=\infty$, anchored $f_\lambda(0)=0$, and normalized $f_\lambda'(0)=1$. We apply the parameterized additive generator of the Frank t-conorm [Fra79], a generalized maximum operator used in fuzzy systems theory:

$$x_s = f_\lambda(a_s) \;=\; \frac{\lambda-1}{\lambda}\log_\lambda\left(\frac{\lambda-1}{\lambda^{1-a_s}-1}\right) \qquad (16)$$

$$a_s = f_\lambda^{-1}(x_s) \;=\; 1-\log_\lambda\left(1+\frac{\lambda-1}{\lambda^{x_s}\lambda/(\lambda-1)}\right) \qquad (17)$$

For various degrees of color enhancement $\lambda > 0$, our technique encompasses previous methods of combining image values. We now analyze the effect of combining instances of the same color, with the same hue and tone angles, that vary only in their color strengths $a_s, b_s \in [0,1)$. In this situation, f_λ acts as a homomorphism on the color strength values. Our default setting $\lambda = e^2$ approximates the mapping $x_s = \tanh^{-1}(a_s)$ [SP83]. When $\lambda \to 1$, our color scaling $x_s = f_1(a_s) = -\ln(1-a_s)$ is the complement of the mapping $x_s = \ln a_s$ [OSS68, Fau79, JP01]. A positively weighted color combination $c = (v \otimes a) \oplus (w \otimes b)$, for $v \geq 0$ and $w \geq 0$, has:

$$c_s = 1-\log_\lambda\left(1+\frac{(\lambda^{1-a_s}-1)^v(\lambda^{1-b_s}-1)^w}{(\lambda-1)^{v+w-1}}\right) \qquad (18)$$

For $\lambda \to 0$, we get a selection operator $c_s = \max(a_s, b_s)$ regardless of the weights. For $\lambda \to \infty$, we get a clipped additive combination $c_s = \min(va_s + wb_s, 1)$. For $\lambda \to 1$, we get a multiplicative combination $c_s = 1-(1-a_s)^v(1-b_s)^w$. In this case, color addition corresponds to a simple, probabilistic model of composite image formation from the superposition of independent components. It assumes that pixels are partially covered by uniformly distributed, identical color particles of a given color. Color strength is taken to be proportional to the probability of encountering a color particle at a random point within the pixel, $a_s = \mathrm{P}[a]$ and $b_s = \mathrm{P}[b]$. For color addition $c = a \oplus b$, observe that $c_s = \mathrm{P}[a \bigcup b] = \mathrm{P}[a] + \mathrm{P}[b] - \mathrm{P}[a \bigcap b] = \mathrm{P}[a] + \mathrm{P}[b] - \mathrm{P}[a]\,\mathrm{P}[b]$ matches $c_s = 1-(1-a_s)(1-b_s) = a_s + b_s - a_s b_s$. Alternate image formation models exist [OSS68, Fau79, PD84, JP01].

3.3. Salience Preserving Image Blending

The goal of salience preserving blending is to produce a composite that retains the most visually informative aspects of its components. A salience map measures how much information each pixel contributes to the understanding of the image. We use it to identify which pixels must be preserved for a component image to remain recognizable in the composite. We combine user specified *opacity maps* w_p^n with computer generated *salience maps* s_p^n to produce *salience mattes* $w_p^{n\prime}$. The salience mattes then replace the opacity maps in linear image blending. Opacity prescribes the importance of each component pixel relative to the same pixel of the other components, while saliency describes the relevance of each component pixel relative to the other pixels of the same component. Typically, opacity maps are imprecise but accurate, conveying user priorities for the composite layout, while salience maps are precise but inaccurate, providing algorithmic refinement of the composite details. Without needing detailed image mattes, our users can create composites that convey the salient details of their components.

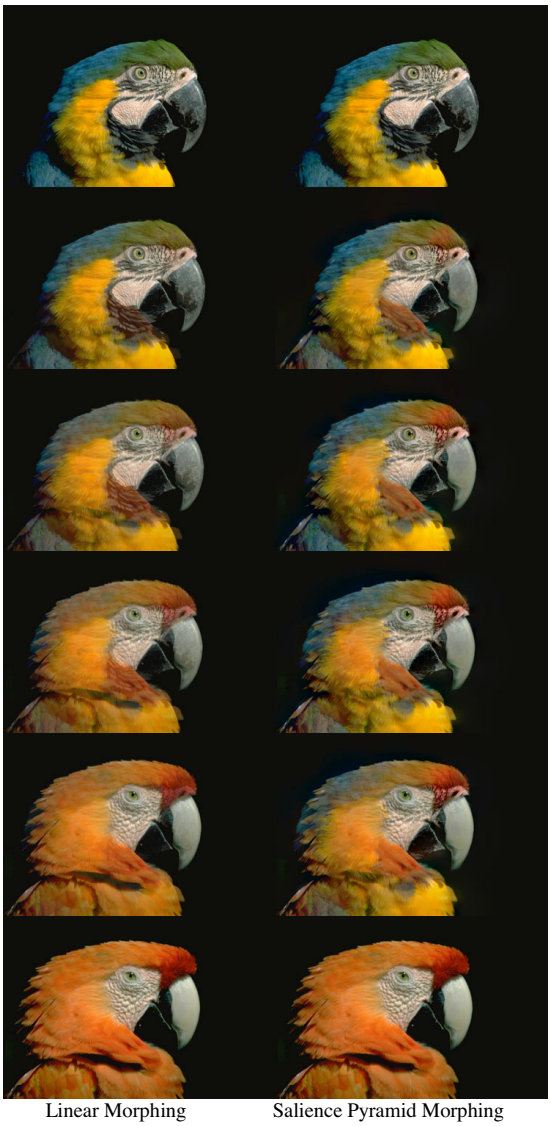


Figure 3: *Image morphing cross dissolves geometrically warped image sequences. Salience preserving pyramid blending reduces the ghosting and contrast artifacts caused by linear interpolation. At the 40% and 60% transitions, salience helps keep the parrot's blue feathers from fading.*

For each component, a pixel's relative salience $s_p^{n\prime}$ is its given salience s_p^n minus the weighted mean salience s_p at that pixel. We then examine the proportion of component pixels that are less or equally salient than that pixel. We derive each component's ranked salience $r_p^n \in [0,1]$ using the cumulative density function $\Phi_n(s_p^{n\prime})$ of its relative salience values:

$$r_p^n = \Phi_n(s_p^{n\prime}) \quad \text{for} \quad s_p^{n\prime} = s_p^n - s_p \quad \text{and} \quad s_p = \sum\nolimits_n w_p^n s_p^n \quad (19)$$

Finally, we calculate each salience matte $w_p^{n\prime}$. Its sharpness is controlled by the parameter $\gamma > 0$, with default $\gamma = 1$. We

apply a power law to the opacity weighted, ranked salience:

$$w_p^{n\prime} = \frac{(w_p^n r_p^n)^\gamma}{\sum_n (w_p^n r_p^n)^\gamma} \tag{20}$$

Our method aims to globally maintain opacity while locally preserving salience. We analyze the *dominance* and *coverage* properties of a uniform cross dissolve involving two images with constant opacities $w_p^1 = w$ and $w_p^2 = 1 - w$.

Dominance: This is the proportion of pixels where the first image contributes more to the composite than the second image, $w_p^{1\prime} > w_p^{2\prime}$. In linear cross dissolve, the first image dominates the entire composite when $w > \frac{1}{2}$ while the second image dominates the entire composite when $w < \frac{1}{2}$. In our cross dissolve, dominance is equal to the opacity.

Coverage: This is the average contribution that the first image makes to the composite. In linear cross dissolve, the average coverage is equal to the opacity. In our cross dissolve, with default $\gamma = 1$, the median coverage is equal to the opacity. For any $\gamma > 0$, equal weights give equal coverage to both images, $\mathrm{E}[w_p^{1\prime}] = \mathrm{E}[w_p^{2\prime}] = \frac{1}{2}$ when $w = \frac{1}{2}$. As the salience mattes approach binary masks with $\gamma \to \infty$, the average coverage approaches the opacity, $\mathrm{E}[w_p^{1\prime}] \to w$.

Perceptual salience is usually evaluated by models of human vision that demand complicated calculations [IKN98]. In our experiments, we estimate salience by color entropy. As a simple approximation to perceptual salience, our information theoretic salience produces surprisingly effective results. We observe that uncommon colors tend to attract attention and they typically belong to foreground objects. For each component, we calculate a 3D RGB color histogram smoothed using a 3D Gaussian filter. Its color sensitivity depends on the smoothing bandwidth. Using the 3D color histogram, we convert pixel color probabilities h_p^n to parameterized color entropies $s_p^n = \left(1 - (h_p^n)^\omega\right) / (\omega \ln 2)$. The parameter $\omega \geq 0$ provides a way to calibrate the resulting salience map, from logarithmic Shannon entropy $s_p^n = -\log_2 h_p^n$ for the default $\omega = 0$ to linear probability of color absence $s_p^n = (1 - h_p^n) / (\ln 2)$ for $\omega = 1$. Finally, to reduce noise and patch small holes, we use a median filter on the salience map.

3.4. Multiresolution Image Blending

Our methods are designed to support image blending over a multiresolution image representation. This often produces the best looking composites, especially for salience preserving blending. We represent the RGB color channels of the component images with Laplacian pyramids and their opacity maps with Gaussian pyramids [BA83]. Standard pyramid blending uses linear interpolation to combine the corresponding coefficients of the Laplacian pyramids using the weights specified by the Gaussian pyramids. For non-binary image mattes, its results often appear identical to the linear interpolation of pixel values. Contrast preserving pyramid blending applies contrast preserving image blending at each pyramid level. Color preserving pyramid blending first maps colors to real values, then performs standard pyramid blending, and finally maps real values back to colors. Salience preserving pyramid blending uses standard pyramid blending with weights specified by salience mattes calculated at each level from Gaussian pyramid representations of the component images (the salience maps' median filter size is reduced by half at each level). Hence, the most informative aspects of the component images are retained at each level of detail. The resulting composite may display high frequency, salient details and textures from one component together with low frequency, salient structures and shading from another.

4. Results

The title page figure uses constant opacity maps to show how salience preserving pyramid blending keeps both background and foreground sharp, avoiding the double exposure typical of linear blending. Notice how the man's chest remains solid while the woman's face fuses with the sky. This effect could be tedious to create with traditional tools.

Consider Figure 1. *Top*: Our composites pick up on different aspects of their components. Contrast preserving blending keeps the high contrast of the white flowers. Color preserving blending favors the red sunset and the yellow flower. Salience preserving blending highlights the yellow flower and the rider. *Middle*: We show the effect of simple, user defined, opacity maps. On the left, salience preserving pyramid blending adapts to the imprecise opacity map, reducing the fringe around the windmill seen in the linear composite. On the right, contrast preserving pyramid blending keeps noticeably more contrast. Color preserving pyramid blending keeps the yellow tone. Salience preserving pyramid blending keeps the lower arches and treetops. We then combine all these benefits into one image. *Bottom*: We use the mean standard deviation of the RGB color channels to gauge how our methods affect contrast in a uniform cross dissolve.

Consider Figure 4. *Top*: We demonstrate how our methods can simultaneously show object interiors and exteriors. *Middle left*: Trilinear filtering is a common way of producing mipmaps. It causes texture contrast to sharply dip between mipmap levels. When rendering mipmapped textures, contrast preserving blending can be used to maintain texture contrast between mipmap levels, avoiding sudden changes in contrast as the resolution varies. *Middle right*: Salience preserving pyramid blending makes superimposed objects appear solid. *Bottom*: Contrast and color preserving image blending can be used in image filtering. To apply contrast preserving filtering, we filter the image normally and then linearly stretch each color channel about its mean to regain the contrast of the input. To apply color preserving filtering, we map colors to real values, filter them, and then map real values back to colors. This example shows the results of Gaussian smoothing. The color preserving smoothing (we used $\lambda = e^{-8}$) exhibits an interesting edge preserving effect.

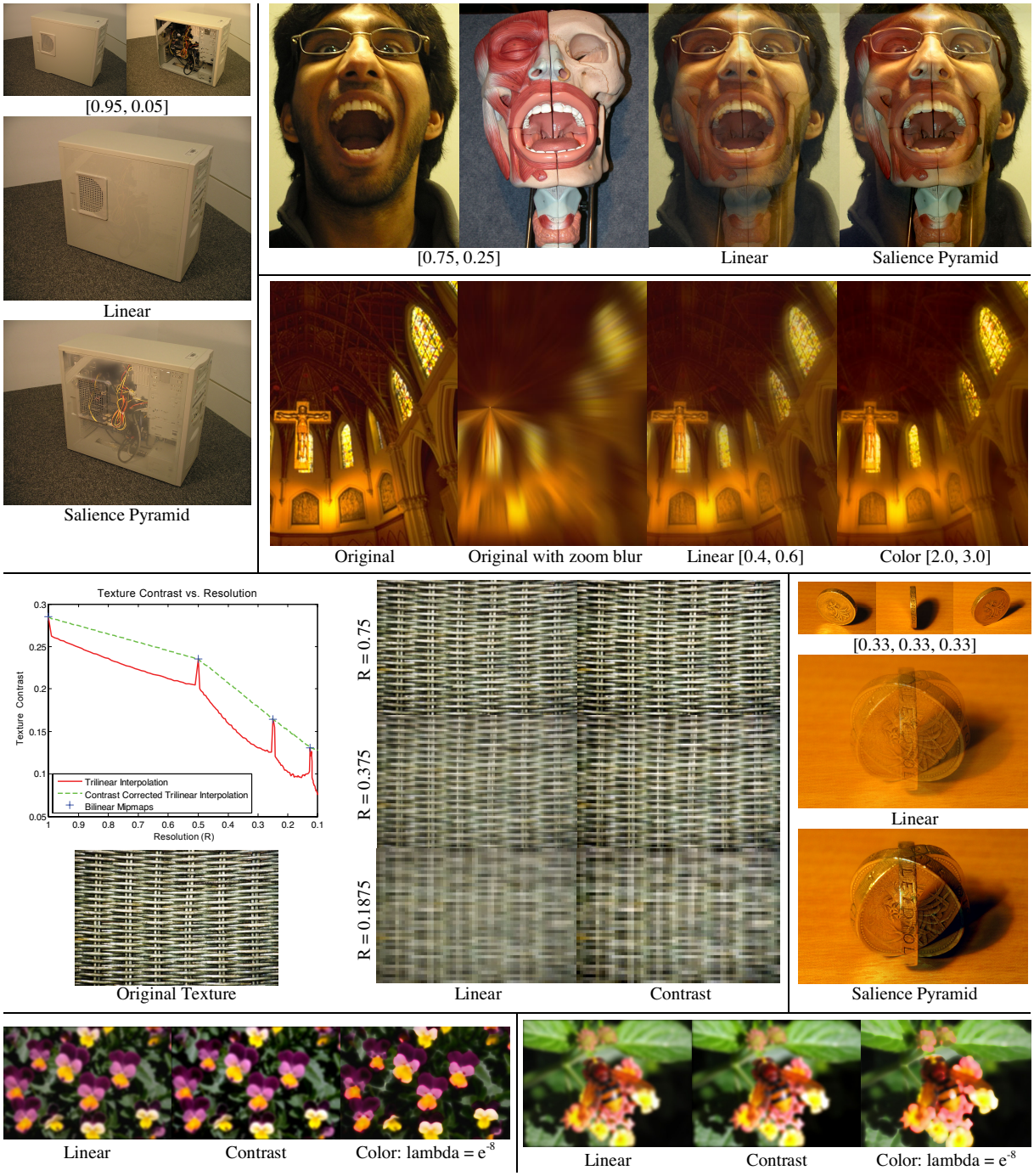


Figure 4: *Top left: Salience preserving pyramid blending is highly sensitive to image content. A blend of 95% exterior and 5% interior images is sufficient to render a translucent computer case. Top upper right: Compared to linear blending, salience preserving pyramid blending produces a much more dramatic anatomical composite. Top lower right: Color preserving blending can take advantage of non-convex image weights. As a result, the church windows appear sharp and bright. Middle left: Trilinear interpolation is used to generate mipmap textures. The graph illustrates how applying contrast preserving blending between mipmap levels maintains contrast of the intermediate textures, while normal trilinear interpolation artificially reduces texture contrast. Middle right: Salience preserving pyramid blending is applied to motion visualization. Bottom left and right: Color and contrast preserving blending can assist in image filtering. In this experiment, we use them for Gaussian smoothing.*

Naturally, all our methods work better on some images than others. The advantage of contrast preserving blending over nonlinear methods is that linear color mappings tend to cause less color distortion. Its disadvantage is that it may map a small proportion of colors out of gamut. The advantage of color preserving blending is that it upholds all the algebraic properties of linear interpolation as well as supports compositing general linear combinations of images. Its disadvantage is that it is not adaptive to image content. The advantage of salience preserving blending is that it can create effective composites using very simple opacity maps. Its disadvantage is that it is limited by the ability of the salience maps to distinguish the essential from the superfluous.

5. Conclusion

In many contexts, combining information by linear interpolation results in reduced variation. To address this longstanding problem, we reformulate linear interpolation by applying statistical analysis, vector algebra, and information theory. In effect, we show how to better preserve contrast, color, and salience when blending images. Our methods improve the artist's control over the appearance of composite images.

References

[ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., CURLESS B., SALESIN D., COHEN M.: Interactive digital photomontage. In *Proc. of SIGGRAPH*, (2004), 294-302.

[Ade86] ADES D.: *Photomontage*, Thames & Hudson, 1986.

[BA83] BURT P. J., ADELSON E. H.: A multiresolution spline with application to image mosaics. *ACM Trans. on Graphics 2*, 4 (1983), 217-236.

[Bri99] BRINKMANN R.: *The Art and Science of Digital Compositing*, Morgan Kaufmann, 1999.

[CAS*97] CURTIS C. J., ANDERSON S. E., SEIMS J. E., FLEISCHER K. W., SALESIN D. H.: Computer-generated watercolor. In *Proc. of SIGGRAPH* (1997), 421-430.

[Fau79] FAUGERAS O. D.: Digital color image processing within the framework of a human visual model. *IEEE Trans. on Acoustics, Speech, & Signal Processing 27*, 4 (1979), 380-393.

[Fra79] FRANK M. J.: On the simultaneous associativity of F(x,y) and x+y-F(x,y). *Aequationes Mathematicae 19*, 2-3 (1979), 194-226.

[GD05] GRUNDLAND M., DODGSON N. A.: Color histogram specification by histogram warping. In *Proc. of SPIE*, (2005), vol. 5667, 610-621.

[HV94] HAEBERLI P., VOORHIES D.: Image processing by linear interpolation and extrapolation. *IRIS Universe Magazine 28*, Aug (1994), 8-9.

[IKN98] ITTI L., KOCH C., NIEBUR E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis & Machine Intelligence 20*, 11 (1998), 1254-1259.

[JP01] JOURLIN M., PINOLI J. C.: Logarithmic image processing. *Advances in Imaging and Electron Physics 115* (2001), 129-195.

[KSE*3] KWATRA V., SCHODL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: Image and video synthesis using graph cuts. In *Proc. of SIGGRAPH*, (2003), 277-286.

[MB96] MASSEY M., BENDER W.: Salient stills: Process and practice. *IBM Systems J. 35*, 3-4 (1996), 557-573.

[NOT98] NAKAMURA K., OHKI M., TOTSUKA T.: Image blending by feature overwrite. In *Proc. of International Conference on Image Processing*, (1998), vol. 1, 226-230.

[OSS68] OPPENHEIM A., SCHAFER R., STOCKHAM T.: Nonlinear filtering of multiplied and convolved signals. *Proc. of the IEEE 56*, 8 (1968), 1264-1291.

[PD84] PORTER T., DUFF T.: Compositing digital images. In *Proc. of SIGGRAPH* (1984), 253-259.

[PGB03] PEREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *Proc. of SIGGRAPH* (2003), 313-318.

[PR01] POCOCK L., ROSEBUSH J.: *The Computer Animator's Technical Handbook*, Morgan Kaufmann, 2001.

[RAG*01] REINHARD E., ADHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics & Applications 21*, 5 (2001), 34-41.

[RKK*05] ROTHER C., KUMAR S., KOLMOGOROV V., BLAKE A.: Digital tapestry. In *Proc. of Computer Vision and Pattern Recognition*, (2005), vol. 1, 589-596.

[SD96] SEITZ S. M., DYER C. R.: View morphing. In *Proc. of SIGGRAPH*, (1996), 21-30.

[Smi95] SMITH A. R.: Alpha and the history of digital compositing. *Microsoft Tech Memo 7*, 1995.

[SP83] SHVAYTSER H., PELEG S.: Pictures as elements in a vector space. In *Proc. of Computer Vision and Pattern Recognition*, (1983), 442-446.

[Sze05] SZELISKI R.: Image alignment and stitching. In *Handbook of Mathematical Models in Computer Vision*, (2005), 275-294.

[Whi00] WHITAKER R. T.: A level-set approach to image blending. *IEEE Trans. on Image Processing 9*, 11 (2000), 1849-1861.

[Wol98] WOLBERG G.: Image morphing: A survey. *Visual Computer 14*, 8-9 (1998), 360-372.

[ZLP*06] ZOMET A., LEVIN A., PELEG S., WEISS Y.: Seamless image stitching by minimizing false edges. *IEEE Trans. on Image Processing 15*, 4 (2006), 969-977.