New Controls for Combining Images in Correspondence

Jing Liao, Diego Nehab, Hugues Hoppe, and Pedro V. Sander

Abstract—When interpolating images, for instance in the context of morphing, there are myriad approaches for defining correspondence maps that align structurally similar elements. However, the actual interpolation usually involves simple functions for both geometric paths and color blending. In this paper we explore new types of controls for combining two images related by a correspondence map. Our insight is to apply recent edge-aware decomposition techniques, not just to the image content but to the map itself. Our framework establishes an intuitive low-dimensional parameter space for merging the shape and color from the two source images at both low and high frequencies. A gallery-based user interface enables interactive traversal of this rich space, to either define a morph path or synthesize new hybrid images. Extrapolation of the shape parameters achieves compelling effects. Finally we demonstrate an extension of the framework to videos.

Index Terms—Image interpolation, morphing, edge-aware decomposition, design galleries

1 Introduction

Since their debut several decades ago [1], seamless animated transitions between images or videos, i.e., *morphs*, have continued to fascinate the public [2]. Such animations are typically generated in a two-step process. The first step identifies correspondences between source and target images, usually with the assistance of some artistic and semantic input. Given the correspondence map, the second step creates an animation sequence that interpolates the two images by transitioning both the position and color of every point from source to destination.

In this paper, we explore new creative controls for combining images related by a correspondence map. One application is of course to create richer morph transitions (the second step mentioned above). More broadly, these controls define a low-dimensional space of synthesized images that enables interactive melding of attributes from the source images.

Our approach draws inspiration from techniques for computing edge-aware decompositions of images e.g., [3], [4], [5]. Our key insight is to apply such decomposition not just on the color content of the two images, but also on the geometric correspondence map.

Specifically, we use a rolling guidance filter [6] to construct a two-scale representation (low- and high-frequency) of the color content. We then identify the major edges in the color images [7] to create a similar two-scale representation of the correspondence map vectors. The weights associated with these components effectively define a 4-dimensional parameter space for merging two images, as illustrated in Figures 1 and 12.

We develop an interface to quickly navigate this 4D space. One application is to adjust morph transition paths. We also demonstrate a generalization of the framework to video morphing.

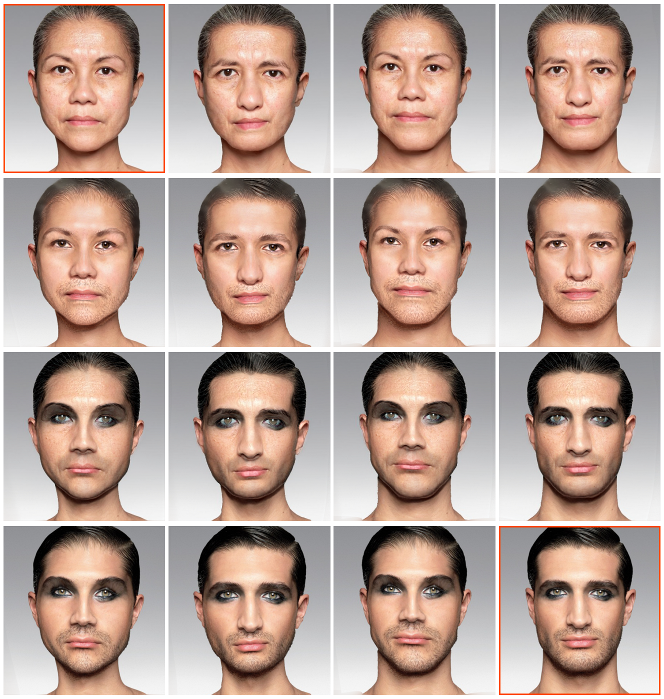


Fig. 1. Extrema images at the 16 corners of our 4D interpolation space, combining low- and high-frequency color and geometry from the two input photographs highlighted in red (at top left and bottom right).

2 Related work

Morphing There is extensive research on creating image morphs, including several surveys and books [8], [9], [10]. Early research focused on user interfaces for specifying sparse correspondences (in the form of points, segments,

- J. Liao is with the Hong Kong University of Science and Technology. Email: liaojing8871@gmail.com
- D. Nehab is with IMPA.
- H. Hoppe is with Microsoft Research.
- P. Sander is with the Hong Kong University of Science and Technology. Email: psander@cse.ust.hk

or meshes) and on techniques for interpolating these correspondences across the image domains. More recent work focuses on increasing the level of automation [11], [12], [13] and extending the problem to multiple images [14] or videos [15], [16], [17].

There is relatively less research on controlling the morph once correspondences are established. Most techniques define linear or quadratic paths on the sparse correspondences and use regularization functionals to extend this to the image domain. For instance, Nishita et al. [18] use linear interpolation across Bézier patch networks and allow spatially varying transition rates. [19] achieve spatially varying transition control using multilevel B-spline interpolation.

Image decomposition A variety of methods are able to separate image detail across scales, essentially by piecewise smooth filtering. These include anisotropic diffusion [20], bilateral filtering [3], weighted gradient-domain least squares [4], local extrema envelopes [21], guided filters [5], L^0 minimization [22], and rolling guidance filters [6]. Our approach applies this concept to obtain a multiscale decomposition of a geometric map based on edge content in the two associated color images. This also relates to joint-bilateral filtering [23], [24], which uses discontinuities in one image to modulate the smoothing filter over another image. Another noteworthy work is hybrid images [25] which superimposes two images at different scales such that the final interpretation varies with viewing distance.

Image collection subspaces Principal components are commonly used to interpolate among a collection of aligned face images [26] or 3D face models [27]. In the latter case, there is an explicit parametric description of a surface over a spherical domain. Unlike such prior work which explicitly models 2D shape, Nguyen et al. [28] apply principal components on relative differences in shape encoded implicitly by the maps between a co-aligned image collection. Our work is closely related to this, but we further decompose the subspace into different frequency bands.

Image deformation Many techniques allow controlled deformations of image domains, such as moving least squares [29] or finite element warping [30]. Our approach differs in that it operates on two images in correspondence.

Image harmonization Existing approaches can incorporate regions from a source image onto a target image while perserving key visual properties of the target image, such as texture and noise [31], [32]. However, these methods do not consider decoupling geometry frequencies, and thus cannot reproduce many of the effects of our approach.

3 Review of basic morphing framework

Let I_0 and I_1 be two input images. A morph is a procedure for generating intermediate images I_α for values $\alpha \in (0,1)$. In the process, points p_0 in I_0 are moved to their corresponding positions p_1 in I_1. At the same time, their colors are progressively changed from $I_0(p_0)$ to $I_1(p_1)$.

We build on the morphing framework of [13]. To create a correspondence map, they define a *halfway parameterization* using a vector field v on a domain Ω that is conceptually halfway between the input images. Each domain point $p \in \Omega$ associates image points $p_0(p) = p - v(p)$ in I_0 and $p_1(p) = p + v(p)$ in I_1. The vector field v is obtained using an optimization that takes into account smoothness, image structure, and user input. The construction is symmetric in the sense that swapping the two images simply negates the field v.

During the morph, each point $p_0(p)$ follows a trajectory $q_\alpha(p)$ toward $p_1(p)$ as α ranges from 0 to 1. The trajectory is a quadratic Bézier curve that interpolates a third point $q_{1/2}(p) = p + w(p)$:

$$q_\alpha(p) = p + (2\alpha - 1)\, v(p) + 4\alpha\,(1-\alpha)\, w(p). \qquad (1)$$

The vector field $w(p)$ is obtained by a second optimization that minimizes geometric distortion at the halfway frame in the animation. Note that the linear paths used in many other morphing algorithms corresponds to setting all $w(p)$ to the zero vector.

Given the trajectories encoded by the vector fields v and w, each output frame I_α is defined using linear color interpolation as

$$I_\alpha\big(q_\alpha(p)\big) = (1-\alpha)\, I_0\big(p_0(p)\big) + \alpha\, I_1\big(p_1(p)\big). \qquad (2)$$

These intermediate images are evaluated by a fast iterative algorithm.

4 Frequency-based decomposition

We begin by generalizing the morphing framework to decouple the interpolations of color and geometry, splitting the transition parameter α into separate coefficients c and g, respectively. The intermediate image $I_{c,g}$ is thus defined as

$$I_{c,g}\big(q_g(p)\big) = (1-c)\, I_0\big(p_0(p)\big) + c\, I_1\big(p_1(p)\big). \qquad (3)$$

Note that setting $c = g = \alpha$ reproduces the traditional morph.

The next two sections describe our edge-aware multiscale representations of both color and geometry, to enable further splitting the interpolation coefficients into c^L, c^H, g^L, g^H corresponding to the low- and high-frequency bands.

4.1 Color decomposition

The input images I_0 and I_1 are each decomposed into two frequency bands as follows. We apply the rolling guidance filter of Zhang et al. [6] to eliminate small details from I_0, resulting in a “low-frequency” image I_0^L that preserves sharp edges. We form the “high-frequency” image as $I_0^H = I_0 - I_0^L$. Image I_1 is similarly decomposed into I_1^L and I_1^H. Figure 2 shows two examples.

Given the decomposition, the parameters c^L and c^H can be set independently such that the resulting intermediate image is the sum of contributions from the two frequency bands. Because the detail images I_0^H and I_1^H are both residual images (with zero DC terms), it is sometimes useful to allow non-affine combinations of this detail (as discussed in Section 6). We therefore introduce separate coefficients c_0^H, c_1^H associated with the two images, to obtain:

$$\begin{aligned} I_{c^L, c_0^H, c_1^H, g}\big(q_g(p)\big) &= (1-c^L)\, I_0^L\big(p_0(p)\big) + c^L\, I_1^L\big(p_1(p)\big) \\ &\quad + c_0^H\, I_0^H\big(p_0(p)\big) + c_1^H\, I_1^H\big(p_1(p)\big). \qquad (4) \end{aligned}$$

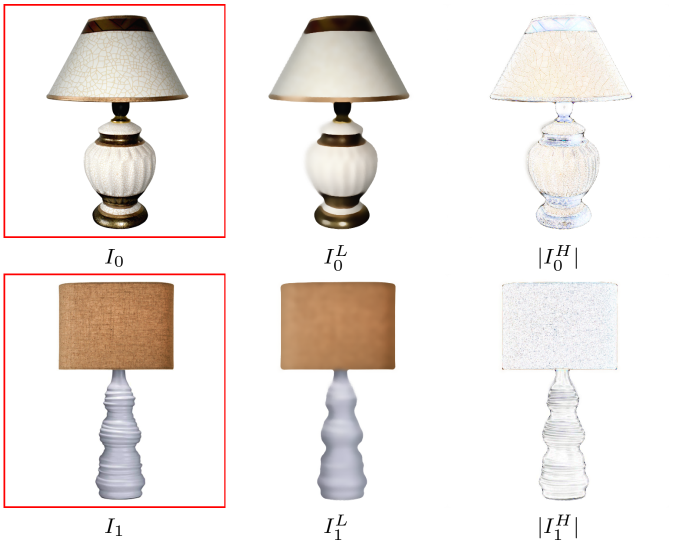


Fig. 2. Color decomposition of two input images into low- and high-frequency components.

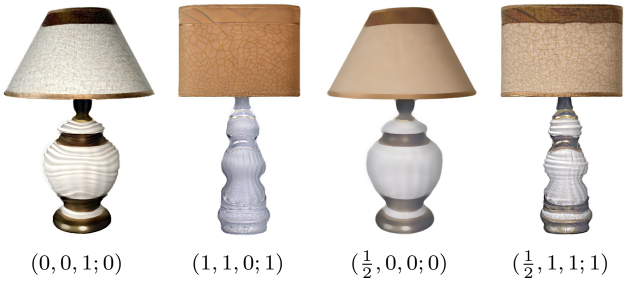


Fig. 3. Images reconstructed using different combinations $(c^L, c_0^H, c_1^H; g)$ of the low- and high-frequency color components and overall geometry from the two images in Figure 2. The first two results demonstrate transfer of color detail from one image to the other, and the last two show blending of low frequencies together with either zero detail or a superposition of detail from both images.

By default, we set $c_0^H = (1 - c^H)$ and $c_1^H = c^H$. Figure 3 shows various combinations of the color components from the lamp images in Figure 2.

Contrast normalization The magnitudes in the high-frequency image I_0^H can differ significantly from those in image I_1^H, particularly when the contrast in one input image is higher than that of the other. To produce meaningful recombinations of frequency bands, (e.g., $I_0^L + I_1^H$ or $I_1^L + I_0^H$), it is important to normalize this contrast. Otherwise, the transferred high frequencies may be too subtle to notice or may dominate the low frequencies. Figure 4 shows two examples where this happens, as well as the results of the contrast normalization procedure we describe below.

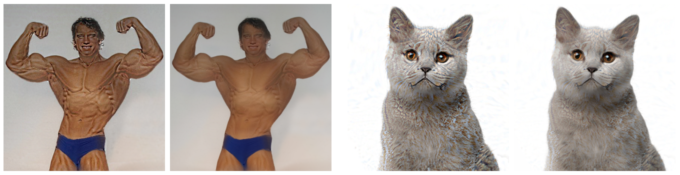


Fig. 4. In each image pair, the left image shows recombination of frequency bands without contrast normalization, whereas the right image shows the result of contrast normalization.

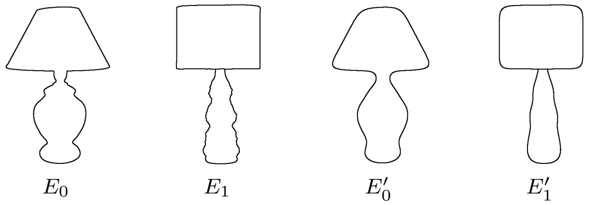


Fig. 5. Edge sets before and after Laplacian smoothing.

Let A_0 and A_1 be the RMS of I_0^H and I_1^H, respectively. We modify (4) by maintaining the low-frequency part (i.e., the first row) unchanged, but replacing the high-frequency part (i.e., the second row) with the new expression

$$\left(c_0^H \frac{I_0^H(p_0(p))}{A_0} + c_1^H \frac{I_1^H(p_1(p))}{A_1} \right) \left((1 - c^L) A_0 + c^L A_1 \right). \tag{5}$$

The first factor linearly combines the *contrast-normalized* versions of the high-frequency images, and the second factor computes the *desired contrast* as a linear combination of the contrasts in the two high-frequency images. In other words, when c_L selects the low frequencies of I_0^L we desire the contrast to match that of I_0^H, and conversely when c_L selects the low frequencies of I_1^L we desire the contrast to match that of I_1^H.

4.2 Geometry decomposition

The more challenging step is to decompose the correspondence map into low- and high-frequency components. As with color, we desire independent decompositions associated with I_0 and I_1. One initial thought was to simply apply some some edge-preserving (e.g., bilateral) filter to the vector field v. However, because the field v is symmetric, we cannot determine whether high-frequency changes in v are caused by details present in I_0 or I_1, or both. To make this distinction, we must resort to the color content of these images.

We first detect the major edges of I_0^L and I_1^L using the method of Cheng et al. [7], obtaining two edge sets E_0 and E_1. Each set is actually a network of parametric paths. We apply Laplacian smoothing to these parametric paths to obtain smooth edge sets E_0' and E_1' (Figure 5). These smoothed version can be thought of as the low-frequency components of E_0 and E_1.

We form the geometry decomposition associated with I_0 as follows. For any pixel q_0 on an edge of E_0, we compute two points: (1) the corresponding point p in the halfway domain, i.e., satisfying $q_0 = p - v(p)$, and (2) the point q_0' in the smoothed edge set E_0' at the same parametric location as q_0 in E_0. We define the low-frequency component of the map as the vector from p to q_0'. Thus,

$$v_0^L(p) = p - q_0'. \tag{6}$$

We propagate these sparse edge-pixel correspondences to the remainder of the image domain by solving a linear system with soft thin-plate spline constraints. Finally, the high-frequency component is the residual

$$v_0^H(p) = v(p) - v_0^L(p). \tag{7}$$

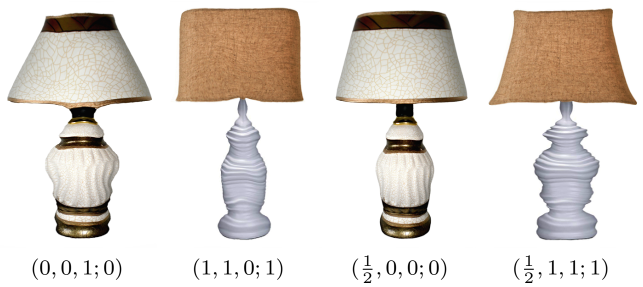


Fig. 6. Images reconstructed using various combinations $(g^L, g_0^H, g_1^H; c)$ of the low- and high-frequency geometric components from the two images in Figure 2. The first two results demonstrate transfer of geometric detail across images, and the last two show blending of low frequencies together with either zero detail or a superposition of detail from both images. (Note the silhouettes.)

The decomposition associated with I_1 is computed analogously using the edges E_1 and smoothed edges E_1' as $v_1^L(p) = q_1' - p$ and $v_1^H(p) = v(p) - v_1^L(p)$.

Because the quadratic trajectory component $w(p)$ is generally small and not associated with edges in either I_0 nor I_1, we do not decompose it and instead modulate its contribution according to the low-frequency coefficient g^L.

Given the geometry decomposition, the parameters g^L, g_0^H, and g_1^H can be set independently to define the final trajectory path as a sum of contributions from the two frequency bands:

$$\begin{aligned} q_{g^L,g_0^H,g_1^H}(p) &= p - (1-g^L)\,v_0^L(p) + g^L\,v_1^L(p) \\ &\quad - g_0^H\,v_0^H(p) + g_1^H\,v_1^H(p) + 4g^L(1-g^L)\,w(p). \end{aligned} \tag{8}$$

Figure 6 shows some examples using the same pair of lamp images.

4.3 Multiscale image combination space

Combining the color and geometry decompositions, we arrive at a 4D space parameterized by coordinates (c^L, c^H, g^L, g^H), which represent the low- and high-frequency geometric and color components. Figures 12 and 13 illustrate the extrema images of this 4D space for a few examples.

The more general version, which allows non-affine combinations of color and geometry detail from the two images, is the 6D space with coordinates $(c^L, c_0^H, c_1^H, g^L, g_0^H, g_1^H)$. We discuss our preferred way to explore this higher-dimensional space in Section 6.

5 Controlling morph transition rates

Parametric cubic splines To define a morph using our geometry and color decompositions, we express the parameters c^L, c^H, g^L, g^H as cubic spline functions of the transition parameter α. The user can control the rate, or speed, of the morph transition by adjusting a small number of spline control points, which are then smoothly interpolated as shown in Figure 7. If $c(\alpha) = g(\alpha) = \alpha$, then both color and geometry are morphed at a constant speed throughout the transition interval. Providing the freedom to manipulate these transition rate functions lets the user create more interesting effects.

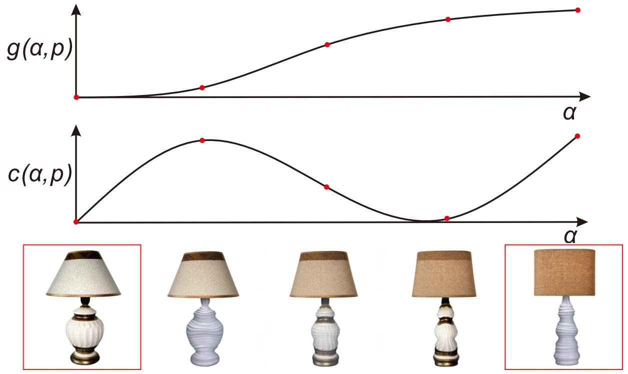


Fig. 7. A morph between two lamps (input images highlighted in red). The lamp base uses the transition functions above, while the lamp shade uses linear interpolation.

Spatial adaptivity For some image morphs, it is useful to apply different transition functions (including different endpoints) over particular spatial regions of the halfway domain. We let the user delineate these regions using either intelligent scissor [33] or polygon scissor tools. The parameters become transition rate functions of both α and position p:

$$c^L(\alpha,p), c^H(\alpha,p), g^L(\alpha,p), g^H(\alpha,p). \tag{9}$$

In the example of Figure 7, the base of the lamp has different rate functions, plotted at the top of the figure, whereas the remainder of the image (lamp shade and background) have no specified control points and thus uses default linear interpolation.

If multiple (potentially overlapping) regions are specified, they are assembled in the halfway domain in a user-configurable order. To generate each intermediate image I_α of the morph, we rasterize the 6D parameters associated with the regions into the halfway domain, then apply Gaussian filtering to obtain a smooth transition (approximately 20 pixels wide) across the region boundaries. This smoothed parameter field is then used to evaluate the intermediate image.

6 Non-affine combinations

The 6D combination space $(c^L, c_0^H, c_1^H, g^L, g_0^H, g_1^H)$ is useful for scenarios where the user wants to include details from both images, such as the rightmost results in Figures 3 and 6 (for color and geometry, respectively). Alternatively, the user may want to completely omit high frequencies, as shown in the third results of Figures 3 and 6. Thus, it is useful to relax the constraint that the pairwise coefficient combinations must form a partition of unity.

However, manipulating two additional spline functions is cumbersome. Fortunately, for the application of morphing we have found an intuitive mode of operation that achieves many of the interesting symmetric combinations without requiring the specification of additional splines.

The idea is to define the additional functions by reflecting the existing ones about the halfway point $\alpha = \frac{1}{2}$ of the morph interval. As before, we have

$$c_1^H(\alpha,p) = c^H(\alpha,p), \quad g_1^H(\alpha,p) = g^H(\alpha,p), \tag{10}$$

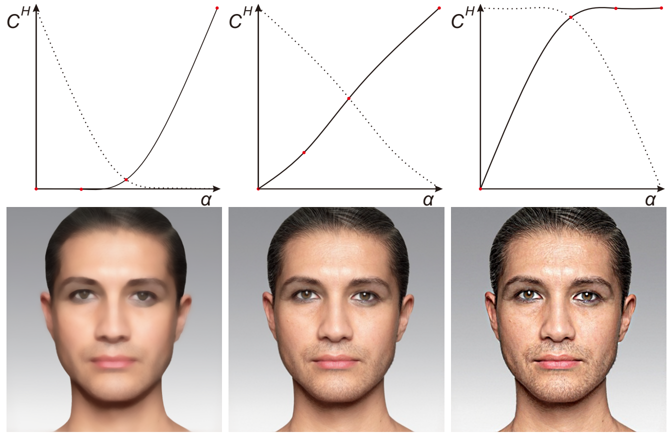


Fig. 8. Given the specified function c^H, we set $c_1^H(\alpha) = c^H(\alpha)$ as before, but generate the (dotted) function $c_0^H(\alpha) = c^H(1-\alpha)$ by reflecting about the midway point $\alpha = \frac{1}{2}$. In these three examples, color detail from both images is reduced (left), combined (center), or increased (right) in the middle of the morph.

but now we assign

$$c_0^H(\alpha, p) = c^H(1-\alpha, p), \quad g_0^H(\alpha, p) = g^H(1-\alpha, p). \quad (11)$$

Refer to Figure 8 for three useful settings for this reflection mode.

7 User interface

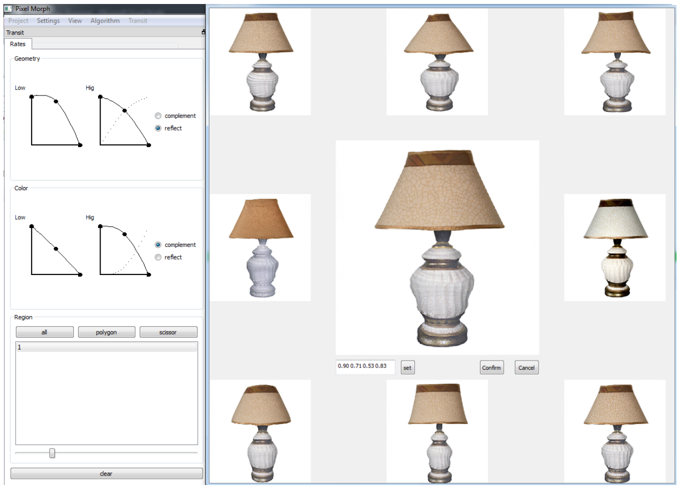
Our user interface allows manipulation of the four splines $c^L(\alpha), c^H(\alpha), g^L(\alpha), g^H(\alpha)$ associated with an image morph by directly specifying control points at any transition values α, as shown on the left side of Figure 9. The user can toggle the reflection mode described in Section 6 for color and/or geometry.

Each control point corresponds to a vector in the 4D combination space described in Section 4.3. Navigating this 4D space by adjusting coefficients using traditional slider controls can be unintuitive. Instead, when a control point is selected, we present a gallery-based interface inspired from design galleries [34] to enable visual navigation within the 4D space, as shown on the right side of Figure 9. The image that represents the currently selected 4D point is shown in the center of the gallery. The eight images in the periphery each show the potential result of advancing a fixed step in one of the eight axis directions (increasing or decreasing the value in each of the four dimensions). The user modifies the central image by holding down the mouse button towards any of these directions. The distance from the mouse cursor to the center of the gallery determines the rate of change. All 9 images update in real-time, so the user can efficiently navigate through the 4D space.

Please refer to the accompanying video for a complete demonstration of the user interface.

8 Extension to videos

Our approach extends to video morphing. Given a correspondence between two temporally synchronized videos e.g., [17], we apply our technique to each frame. The main difficulty is to track any user-delineated regions. We achieve this by following the same approach used to track control points in [17]. Essentially, we treat the user-specified polygon scissor points just like morphing control points, tracking them using optical flow and letting the user adjust their positions in different frames when necessary.



○ complement
● reflect

● complement
○ reflect

1

0.90 0.71 0.53 0.83

Fig. 9. The user interface in gallery mode.

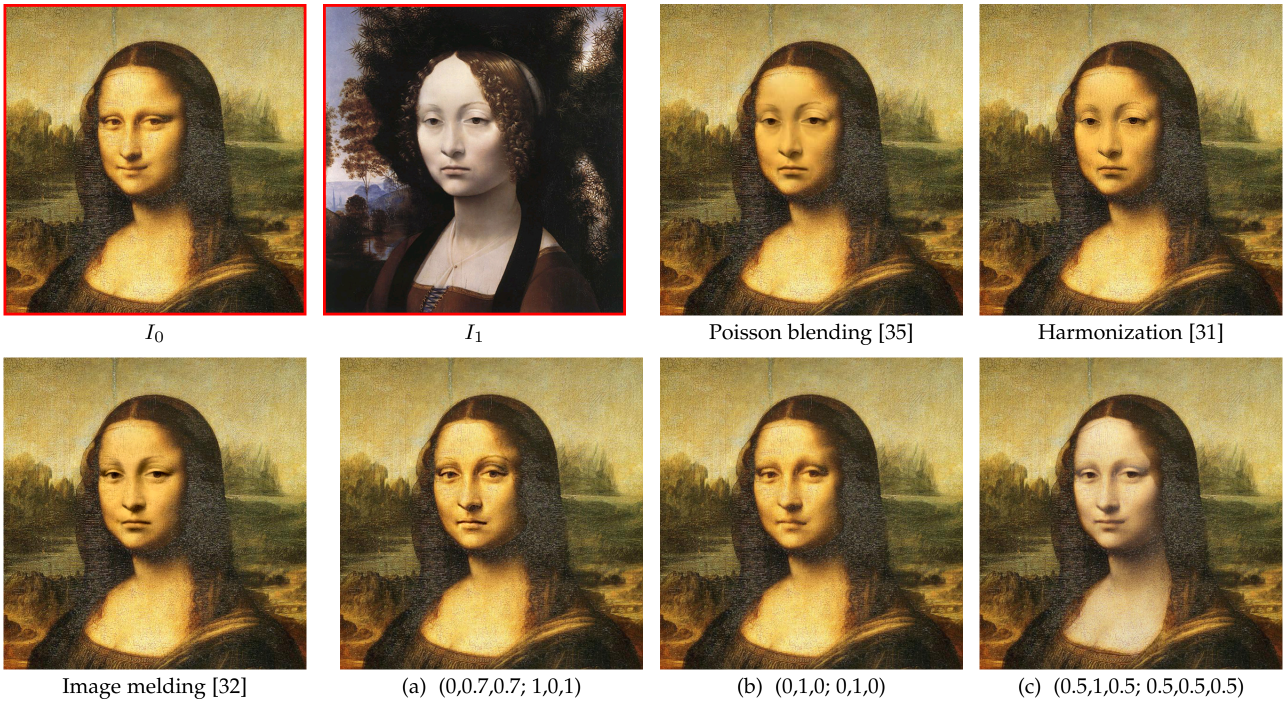
Figure 16 shows frames from two examples results. The first is that of a woman talking, in which her face's color and geometry details are replaced by those of a man. Note that the face region, which was delineated for this transition, is accurately tracked through optical flow. The second example transfers the color and high frequency geometry from a purple flower *image* to a video of a red flower blowing in the wind.

9 Results

Figures 1, 12 and 13 show the sixteen extreme affine combinations of low and high-frequency color and geometry from several pairs of input photographs. For each pair set, the top row uses low- and high-frequency colors from I_0, whereas the bottom row uses low- and high-frequency colors from I_1. The two intermediate rows use low-frequencies from one of the images and high-frequencies from the other. Conversely, each column uses different combinations of low- and high-frequency geometry from each input image. Note how the parameters span a large rich space of distinctive images, many of which are interesting and plausible.

Figure 14 shows examples of non-affine combinations. Since the weights do not form a partition of unity, an additional two high-frequency coordinates must be specified.

Figure 15 shows a selection of frames in novel morphing sequences enabled by our technique. The first two images in each set show the input. The first row shows a morph in which the transition starts earlier on the right than on the left, recreating an effect reminiscent of the seminal Exxon tiger animation [37]. In the morph between a raw steak and an apple, the steak is first warped to have the shape of the apple, then the stem and leaf appear quickly, and finally the color is transitioned to that of the apple. Finally, the bunny-origami sequence uses similar settings, but restricted to the region covered by the bunny.



I_0

I_1

Poisson blending [35]

Harmonization [31]

Image melding [32]

(a) (0,0.7,0.7; 1,0,1)

(b) (0,1,0; 0,1,0)

(c) (0.5,1,0.5; 0.5,0.5,0.5)

Fig. 10. Comparison between previous works and our results (a-c) using different combination weights $(c^L, c_0^H, c_1^H; g^L, g_0^H, g_1^H)$. Result (a) resembles the earlier work which transfers the face from I_1 while retaining the appearance from I_0; (b) preserves the facial detail of I_0 but warps its structure (e.g., smile) from I_1; result (c) shows the additional interpolation of color and geometry.

Existing 2D warping/morphing approaches do not decouple and recompose different geometry frequencies. So the effects achieved by our work are of a different nature than those of earlier techniques. Please refer to the accompanying video for a demonstration of all of the examples in the paper and the comparisons to existing morphing techniques.

In Figure 10, we compare our results with earlier approaches that also attempt to seamlessly combine different images. The Poisson-blended result [35] does not preserve the texture details of the painting. The Harmonization [31], Image melding [32], and our results (a) & (b) are able to preserve these texturing details. We achieve this by using the geometry from the source image and different combinations of colors for these two results. Furthermore, by manipulating the geometry transition functions, we can achieve a wide range of different output styles, such as the one shown in (c), where we combine both geometry and color information from the input images.

The comparisons of Figure 11 show that regenerative morphing [36] allows artistic transitions between arbitrary scenes. However, if there is a more direct correspondence between the scenes, our method can use this correspondence to obtain better results.

Limitations We have identified a range of problems that our method cannot solve, and that we believe could be ameliorated in future work. The first problem is visible in the body-builder example of Figure 12. Here, large lighting differences in the input images causes the color decompositions to be irreconcilable. As a consequence, the middle rows, which combine color frequency bands from both inputs, look unrealistic.

Another problem can be seen in the cat-lion example of Figure 12. Here, in the geometry decomposition step, the major edge sets detected on two input images are asymmetric. This causes the two central columns, which combine geometry frequency bands from both inputs, to include asymmetric results even though no such asymmetries are visible in the input images. This could potentially be addressed by manual selection of the major edge sets, or in future work using some optimized joint selection that considers both images simultaneously.

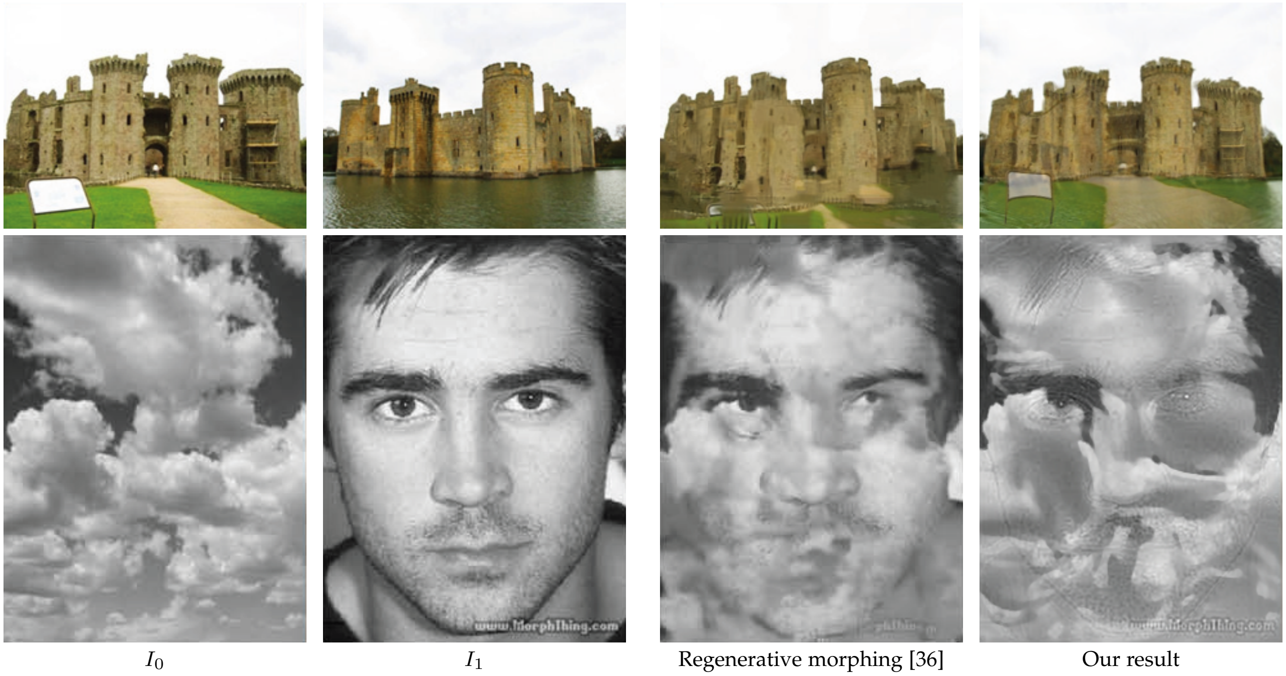
Moreover, when sharp edges in the input images are not precisely matched by the correspondence map, choices that combine color frequency bands from both inputs will include two sets of misaligned edges. This creates the ghosting artifacts visible in some of our examples, such as the hairline in human face example of Figure 13.

Finally, although the original correspondence map is guaranteed by construction to be a bijection, we cannot guarantee that the combination of low and high frequency components from different images and extrapolation of the high frequencies will still result in a bijection. The user can preview the results from the gallery-based interface and stop advancing in a direction that would cause foldovers.

10 Conclusion

We have introduced a framework for multiscale combination of shape and color attributes from a pair of images related by a correspondence map. We have shown that this space of combinations spans a rich variety of hybrid images.

An exciting area for future work is to extend this framework to multiple images, particularly for collections of images of a particular object class (e.g., faces). It may be feasible to jointly analyze a set of correspondence maps, either with respect to a single reference image or within



I_0 I_1 Regenerative morphing [36] Our result

Fig. 11. Comparison between regenerative morphing [36] and our results. In the upper example we use combination weights (0.5,0.5,0.5; 0.5,0.5,0.5) to generate the halfway transition image between I_0 and I_1; in the lower example we use combination weights (0,0,1; 0,0,1) to combine low frequencies of I_0 and high frequencies of I_1. The scenes in the top row have a more direct correspondence between objects, thus allowing our method to provide a better transition. On the other hand, the scenes on the bottom row are unrelated and therefore regenerative morphing can produce a better transition.

some hierarchical clustering, to extract useful multiscale principal components. One can also envision defining semantic paintbrushes to transform the image towards or away from particular reference points, e.g., “make this face look more like Brad Pitt”.

Acknowledgment

This work was partly supported by Hong Kong GRF grants #619509 and #618513.

References

[1] PDI, “Black or white,” 1991, video clip http://www.youtube.com/watch?v=F2AitTPI5U0#t=327.

[2] A. Cerniello, K. Sirchio, N. Meier, E. Earle, G. Cuddy, and M. Reveley, “Danielle,” 2013, video clip http://vimeo.com/74033442.

[3] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. ICCV*, 1998, pp. 839–846.

[4] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, “Edge-preserving decompositions for multi-scale tone and detail manipulation,” *ACM Trans. Graph.*, vol. 27, no. 3, p. 67, 2008.

[5] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *Proc. ECCV*, 2010, pp. 1–14.

[6] Q. Zhang, X. Shen, L. Xu, and J. Jia, “Rolling guidance filter,” in *Proc. ECCV*, 2014, pp. 815–830.

[7] M.-M. Cheng, “Curve structure extraction for cartoon images,” in *Conf. on Harmonious Human Machine Environment*, 2009, pp. 13–25.

[8] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press, 1990.

[9] J. Gomes, L. Darsa, B. Costa, and L. Velho, *Warping and Morphing of Graphical Objects*. Morgan Kaufmann, 1999.

[10] G. Wolberg, “Image morphing: a survey,” *The Visual Computer*, vol. 14, no. 8, pp. 360–372, 1998.

[11] P. Gao and T. W. Sederberg, “A work minimization approach to image morphing,” *The Visual Computer*, vol. 14, no. 8–9, 1998.

[12] E. Wu and F. Liu, “Robust image metamorphosis immune from ghost and blur,” *The Visual Computer*, vol. 29, no. 4, pp. 311–321, 2013.

[13] J. Liao, R. S. Lima, D. Nehab, H. Hoppe, P. V. Sander, and J. Yu, “Automating image morphing using structural similarity on a halfway domain,” *ACM Trans. Graph.*, vol. 33, no. 5, p. 168, 2014.

[14] S. Lee, G. Wolberg, and S. Shin, “Polymorph: Morphing among multiple images,” *IEEE Computer Graphics and Applications*, vol. 18, no. 1, pp. 58–71, 1998.

[15] R. Szewczyk, A. Ferencz, H. Andrews, and B. C. Smith, “Motion and feature-based video metamorphosis,” in *Proc. of ACM Multimedia*, 1997.

[16] Y. Yu and Q. Wu, “Video metamorphosis using dense flow fields,” *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, 2004.

[17] J. Liao, R. S. Lima, D. Nehab, H. Hoppe, and P. V. Sander, “Semi-automated video morphing,” *Computer Graphics Forum*, vol. 33, no. 4, pp. 51–60, 2014.

[18] T. Nishita, T. Fujii, and E. Nakamae, “Metamorphosis using Bézier clipping,” in *Pacific Conference on Computer Graphics and Applications*, 1993, pp. 162–173.

[19] S. Lee, K. Chwa, and S. Shin, “Image metamorphosis using snakes and free-form deformations,” in *Proc. ACM SIGGRAPH*, 1995, pp. 439–448.

[20] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE PAMI*, vol. 12, no. 7, pp. 629–639, 1990.

[21] K. Subr, C. Soler, and F. Durand, “Edge-preserving multiscale image decomposition based on local extrema,” *ACM Trans. Graph.*, vol. 28, no. 5, p. 147, 2009.

[22] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via L^0 gradient minimization,” *ACM Trans. Graph.*, vol. 30, no. 6, p. 174, 2011.

[23] E. Eisemann and F. Durand, “Flash photography enhancement via intrinsic relighting,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 673–678, 2004.

[24] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, “Digital photography with flash and no-flash image pairs,” *ACM Trans. Graph.*, vol. 23, no. 3, 2004.

[25] A. Oliva, A. Torralba, and P. G. Schyns, “Hybrid images,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 527–532, 2006.

[26] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[27] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *Proc. ACM SIGGRAPH*, 1999, pp. 187–194.

[28] C. H. Nguyen, O. Nalbach, T. Ritschel, and H.-P. Seidel, “Guiding image manipulations using shape-appearance subspaces from co-

alignment of image collections," *Computer Graphics Forum (Proc. Eurographics 2015)*, vol. 34, no. 2, 2015.

[29] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 533–540, 2006.

[30] P. Kaufmann, O. Wang, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, and M. Gross, "Finite element image warping," *Computer Graphics Forum*, vol. 32, 2013.

[31] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-scale image harmonization," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 125.

[32] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, "Image melding: combining inconsistent images using patch-based synthesis," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012)*, vol. 31, no. 4, p. 82, Jul. 2012.

[33] E. Mortensen and W. Barrett, "Intelligent scissors for image composition," in *Proc. ACM SIGGRAPH*, 1995, pp. 191–198.

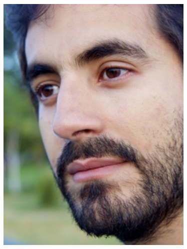
[34] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml *et al.*, "Design galleries: A general approach to setting parameters for computer graphics and animation," in *Proc. ACM SIGGRAPH*, 1997, pp. 389–400.

[35] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, vol. 22, no. 3, pp. 313–318, Jul. 2003.

[36] E. Shechtman, A. Rav-Acha, M. Irani, and S. M. Seitz, "Regenerative morphing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 615–622.

[37] PDI, "Exxon tiger," 1992, video clip http://www.youtube.com/watch?v=vZ7RI3Lt0QY#t=19.

Jing Liao is currently a post-doctoral researcher of the Department of Computer Science and Engineering, Hong Kong University of Science and Technology (HKUST). She received her dual PhD degrees from Zhejiang University and HKUST respectively in 2014 and 2015. Her research interests include image/video processing and non-photorealistic rendering.



Diego Nehab is an associate professor at the National Institute for Pure and Applied Mathematics (IMPA) in Rio de Janeiro, Brazil. He received BEng and MSc degrees in Computer Science from PUC-Rio in 2000 and 2002, respectively, and a PhD degree also in Computer Science from Princeton University in 2007. Before joining IMPA in 2010, he worked as a post-doctoral researcher at Microsoft Research in Redmond. He is interested in most topics related to Computer Graphics, but focuses on parallelism, real-time rendering, and image processing.

Hugues Hoppe is a principal researcher and manager of the Computer Graphics Group at Microsoft Research. His main interests lie in the multiresolution representation, parameterization, and synthesis of geometry, images, and video. He received the 2004 ACM SIGGRAPH Computer Graphics Achievement Award for pioneering work on surface reconstruction, progressive meshes, geometry texturing, and geometry images. He has published many papers at ACM SIGGRAPH and Transactions on Graphics. Contributions at Microsoft include mesh simplification and optimization in DirectX, texture synthesis technology, motion recognition in Kinect Star Wars, and seamless stitching of the terapixel sky in WorldWide Telescope. He is an ACM Fellow, served as editor-in-chief of ACM TOG, and was papers chair for SIGGRAPH 2011. He is a senior member of the IEEE and the IEEE Computer Society.

Pedro V. Sander is an Associate Professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests lie mostly in real-time rendering, graphics hardware, and geometry processing. He received a Bachelor of Science in Computer Science from Stony Brook University in 1998, and Master of Science and Doctor of Philosophy degrees from Harvard University in 1999 and 2003, respectively. After concluding his studies, he was a member of the Application Research Group of ATI Research, where he conducted realtime rendering and general-purpose computation research with latest generation and upcoming graphics hardware. In 2006, he moved to Hong Kong to join the Faculty of Computer Science and Engineering at The Hong Kong University of Science and Technology

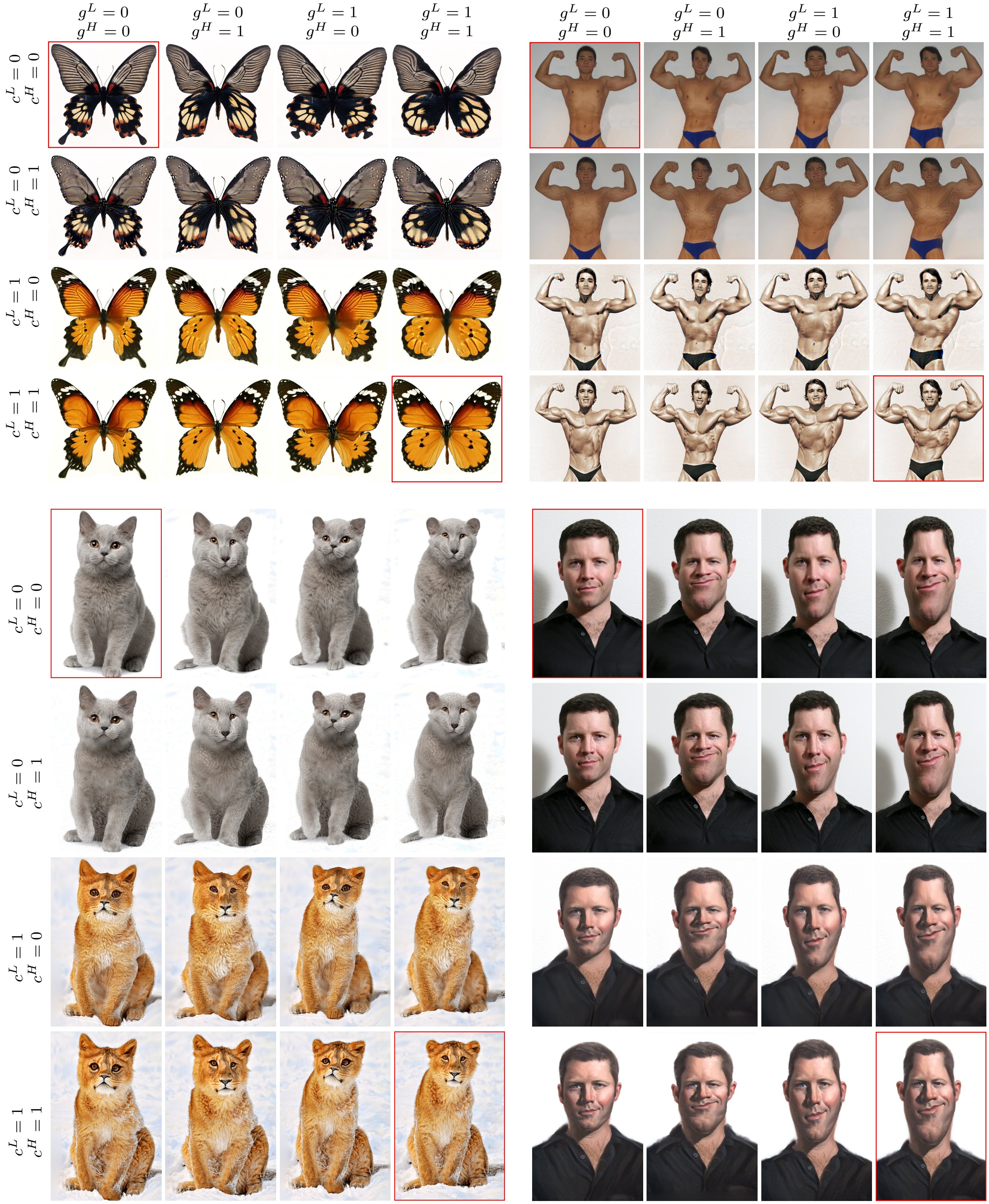


Fig. 12. Different combinations of low- and high-frequency color and geometry from the pairs of input photographs highlighted in red.

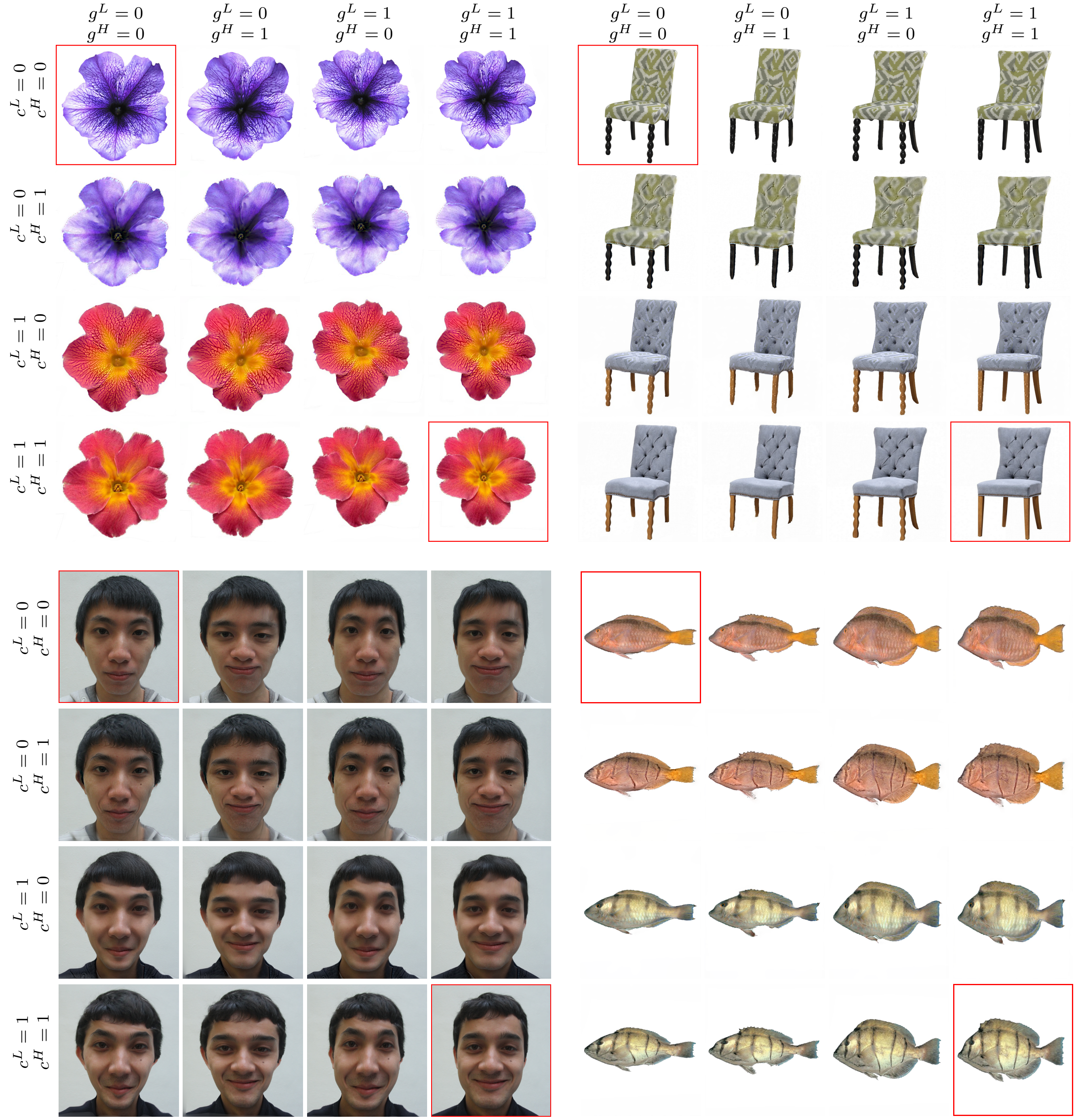
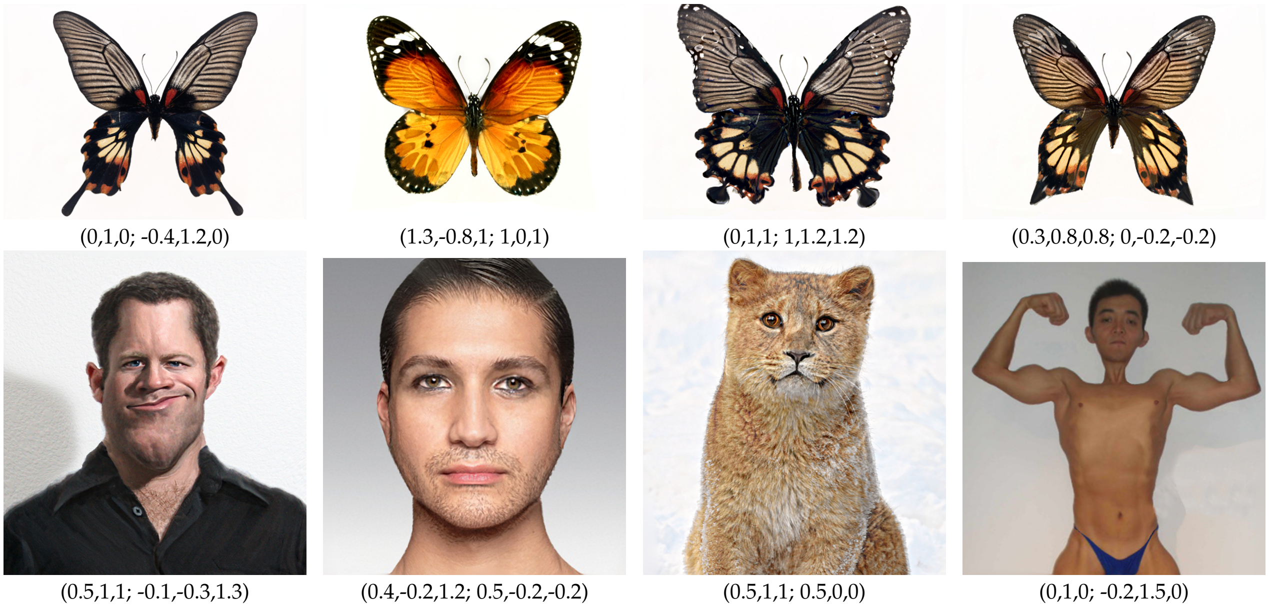


Fig. 13. Different combinations of low- and high-frequency color and geometry from the pairs of input photographs highlighted in red.



(0,1,0; -0.4,1.2,0)

(1.3,-0.8,1; 1,0,1)

(0,1,1; 1,1.2,1.2)

(0.3,0.8,0.8; 0,-0.2,-0.2)

(0.5,1,1; -0.1,-0.3,1.3)

(0.4,-0.2,1.2; 0.5,-0.2,-0.2)

(0.5,1,1; 0.5,0,0)

(0,1,0; -0.2,1.5,0)

Fig. 14. Example results using various combination weights $(c^L, c_0^H, c_1^H; g^L, g_0^H, g_1^H)$, demonstrating shape extrapolation ($g^L \notin [0,1]$) and non-affine combinations ($c_0^H + c_1^H \neq 1$ or $g_0^H + g_1^H \neq 1$) of color or geometry detail.

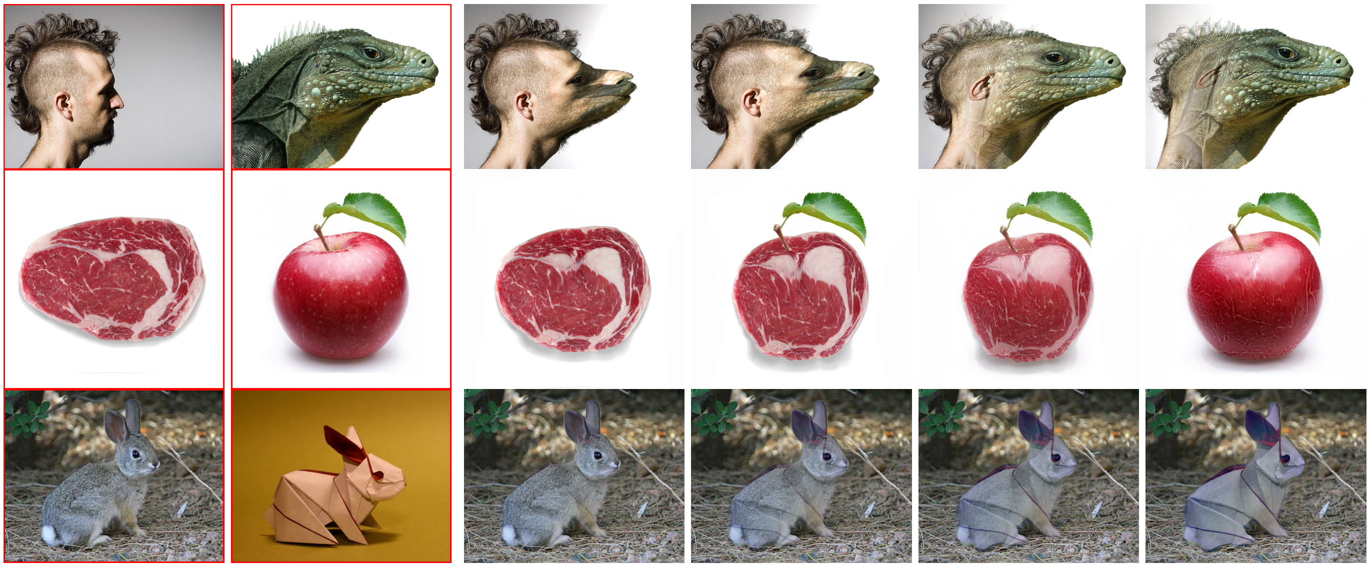


Fig. 15. Selected frames in novel morphing sequences enabled by our technique. The two leftmost images are the input pair. Examples show a range of creative effects, such as spatially varying transition rates, decoupled geometry and color transitions, and decoupled low- and high-frequency transitions.

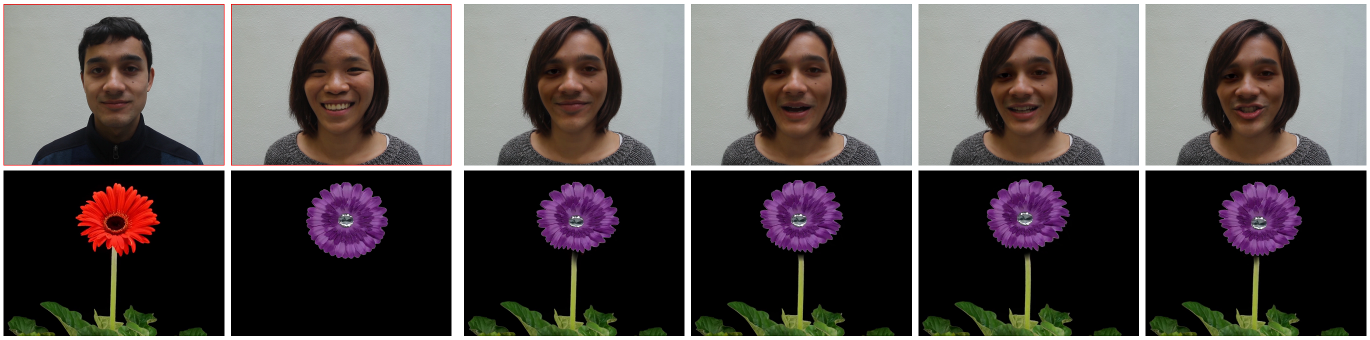


Fig. 16. Two results of our video extension. The two leftmost frames are taken from the original videos, and the remaining frames are from our combined result. For the bottom-row example, the input purple flower is a static (single-image) video.