Transfusive Image Manipulation

Kaan Yücer[1,2] Alec Jacobson[1] Alexander Hornung[2] Olga Sorkine[1]
[1]ETH Zurich [2]Disney Research, Zurich

Figure 1: *A trained artist makes detailed edits to a single source image (left) and our method transfers the edits to the 8 target images (right).*

Abstract

We present a method for consistent automatic transfer of edits applied to one image to many other images of the same object or scene. By introducing novel, content-adaptive weight functions we enhance the non-rigid alignment framework of Lucas-Kanade to robustly handle changes of view point, illumination and non-rigid deformations of the subjects. Our weight functions are content-aware and possess high-order smoothness, enabling to define high-quality image warping with a low number of parameters using spatially-varying weighted combinations of affine deformations. Optimizing the warp parameters leads to subpixel-accurate alignment while maintaining computation efficiency. Our method allows users to perform precise, localized edits such as simultaneous painting on multiple images in real-time, relieving them from tedious and repetitive manual reapplication to each individual image.

Keywords: non-rigid alignment, Lucas-Kanade, content-aware warping, image edit transfer

Links: DL PDF WEB VIDEO

1 Introduction

The process of editing photographs is nearly as old as photography itself. Digital techniques in recent years have greatly expanded the spectrum of possibilities and improved the quality of these edits. Types of editing operations range from global tone adjustments, to color histograms (e.g., [Cohen-Or et al. 2006]) to localized pixel adjustments achieved by highly trained artists using specialized user-interfaces and software (e.g., [Photoshop 2012]). With the increasing availability of large digital photo collections, we currently witness a growing demand to process entire sets of images of similar scenes, taken from different viewpoints, exhibiting varying illumination, dynamic changes such as different facial expressions, and so on [Hays and Efros 2007; Hasinoff et al. 2010; HaCohen et al. 2011]. As pointed out by Hasinoff and colleagues [2010], the manual effort of applying the same localized edit to a multitude of photographs of the subject is often too great, causing users to simply discard some images from a collection.

Many recent works have reduced the user effort required for making image adjustments by using image content to intuitively propagate sparse user edits to the entire image (e.g., [Levin et al. 2004; Lischinski et al. 2006]). This is especially successful for the type of edits which demand less detailed or precise direction by the user, and therefore may be casually transferred to similar images or neighboring images in a video sequence [Li et al. 2010]. Such edits rely on a layer of indirection to disguise imperfect matching or correspondences. For example, in image colorization by sparse scribbles, the chrominance channels produced by [Levin et al. 2004] may contain discontinuities or may be matched incorrectly, but the final result is still acceptable when composed beneath the original (and the most perceptually salient) lightness channel. Recent works improve upon the practicality of such methods, e.g., by supporting more complex macros for photo manipulation that can be applied to larger collections of images [Berthouzoz et al. 2011], but effectively edit propagation is still supported only on a global scale rather than at the pixel level.

In contrast, edits such as local deformations or hand-painted pixel adjustments like the ones shown in Figure 1 may require tedious hours of a trained artist. Because of this cost, it would be advantageous to propagate such detailed edits to similar images of the same subjects. The nature of these edits requires accurate, semantically-meaningful sub-pixel matching between the relevant parts of the images. Simple matching based on color and/or spatial proximity, as used, e.g., in [Levin et al. 2004; Li et al. 2010], proves insufficient for this task. General purpose matching techniques such as optical

flow [Baker et al. 2011] fail due to significant variation in view point, color and shape in loose image collections. Recent works started employing advances in feature matching to propagate also local image edits [Hasinoff et al. 2010; HaCohen et al. 2011], but the applicability to pixel-level edits remains limited due to insufficient flexibility, accuracy and smoothness of the matching.

We propose an adaptation of the Lucas-Kanade (LK) framework [1981] that takes advantage of recent advancements in automatic weight computation methods for handle-based deformation techniques. The LK energy-minimizing matching framework is general enough to encompass subpixel-accurate and smooth mappings, and has been successfully employed in various application domains such as model-based tracking [Baker and Matthews 2004] or optical flow estimation for smaller, sequential displacements [Baker et al. 2011]. Yet it has been less capable of handling complicated matching between images with considerable variation. We optimize our pixel matching function $\mathcal{M} : \mathbf{p} \rightarrow \mathbf{p}'$ in the reduced subspace of maps described by the linear combination of affine transformations: $\mathbf{p}' = \sum_{k=1}^{m} w_k(\mathbf{p})\, T_k \mathbf{p}$, where $w_k(\mathbf{p})$ are carefully chosen, spatially-varying scalar weight functions, and the affine transformations T_k are optimized by our framework.

We define the necessary scalar weight functions such that they promise parametrizable smoothness and content-adaptivity. We do this by reinterpreting images as manifolds, allowing us to directly employ the automatic weighting technique of [Jacobson et al. 2011] originally designed for shape deformation.

Working in this subspace is advantageous in two ways. First, it provides a controllable balance between computational expense and the expressiveness and accuracy of the mapping. Despite optimizing only the parameters of a few affine transformations, the above content-adaptive weighted interpolation allows for mapping functions that faithfully follow the shape even of complex objects. Second, the reduced subspace acts as regularization which avoids the common pitfalls and local minima issues found in optical flow techniques. This allows us to match significant changes in illumination, viewpoint, non-rigid deformation, occlusions, and enables us to accurately transfer edits of a source image to multiple target images. Hence we name our framework "transfusive" image manipulation.

2 Related Work

Transferring pixel-level edits from one image to another requires accurate mappings between corresponding image regions. Finding such correspondences is a long-standing problem in many areas of computer vision and graphics, and there exists a variety of basic, general purpose correspondence estimation techniques ranging from dense optical flow (e.g., [Zimmer et al. 2011]) to sparse features (e.g., [Lowe 2004]). Techniques for optical flow computation and dense feature tracking allow for sophisticated edit propagation in video [Levin et al. 2004; Agarwala et al. 2004; Bhat et al. 2007; Rav-Acha et al. 2008]. However, those methods can handle only small image displacements and appearance changes [Baker et al. 2011] and hence are not suitable for the types of image collections we are aiming at. Methods for stereo correspondence estimation may cope with larger differences between views, but require camera calibration and static scenes [Scharstein and Szeliski 2002], while we would like to work also on uncalibrated images of non-rigidly deforming subjects. Model-based tracking [Baker and Matthews 2004] can handle such deformations, but is generally restricted to face tracking. Sparse feature matching and tracking [Lowe 2004; Sand and Teller 2004] does not produce the dense correspondences required for multi-view image editing applications. Using optical flow in the SIFT domain results in a dense matching [Liu et al. 2008], but the resulting warp is not smooth enough for detailed edit transfer.

For these reasons, a number of algorithms specifically designed for applying edits to multiple images have been recently proposed. For specific object classes, e.g., using face detectors or automatic labeling based on learned features, photo manipulations such as tonal adjustments or object removal have been successfully demonstrated [Bitouk et al. 2008; Brand and Pletscher 2008; Berthouzoz et al. 2011], but a general pixel-accurate mapping between images as in Figure 1 is not supported. Hasinoff et al. [2010] combine various complementary feature detectors, cluster those features whose centers define a homography between images, and then use the homographies for edit transfers on large image collections. However, as discussed in their paper, homography-based edits are limited to relatively planar regions and cannot appropriately handle the non-planar or non-rigid differences between images that we are aiming at. HaCohen et al. [2011] present an extension to the PatchMatch algorithm [Barnes et al. 2010] that enables partial, non-rigid image correspondences with impressive results for applications such as color, mask, and blur transfer between images, but it is optimized towards those types of edits that do not require pixel-accurate correspondences between large, user-defined regions. As we show in our results, our approach complements these works by addressing some of their fundamental limitations.

A fundamental framework for many techniques related to accurate alignment or matching of image regions is the Lucas-Kanade (LK) method [Lucas and Kanade 1981; Baker and Matthews 2004]. We present an extension of this framework that performs matching in a *linear blend skinning* (LBS) subspace that drastically reduces the degrees of freedom compared to techniques like optical flow, while improving the flexibility and accuracy of the matching and thereby enabling pixel-accurate edit transfer between images.

Key to our LBS extension of the LK method is a new type of content-adaptive weight functions that define the regions of influence for each degree of freedom on the image. Various different types of such weight functions have been proposed in the past for applications such as image colorization [Levin et al. 2004], edge-aware editing of tonal values [Lischinski et al. 2006], edge-aware multiscale image representation [Farbman et al. 2008], or interactions between distant pixels [An and Pellacini 2008] (see the surveys in e.g. [Li et al. 2008; Farbman et al. 2010]). However, as we demonstrate in our comparisons, the lack of higher-order smoothness renders those formulations unsuitable for computing accurate, smooth mappings for pixel-level edit transfer between images. We therefore propose extending the bounded biharmonic weights (BBW) of Jacobson et al. [2011], originally developed for LBS deformation of geometric shapes, to *content-adaptive* BBW that respect image edges while preserving the required higher-order smoothness guarantees. Similar edge-aware weights have been used to reduce degrees of freedom (e.g. [Fattal 2009; Fattal et al. 2009]), but the locality and sparsity properties of our weights are more suitable for detailed warps.

These weights in combination with our linear blend skinning formulation of the LK method are the key components for our accurate multi-view edit transfer.

3 Method

Our goal is to transfer edits from a source image I_s to target images I_t, $t = 1, \ldots, N$. To achieve this we find an *optimal* map $\mathcal{M}_t$ which takes pixel coordinates $\mathbf{p} \in I_s$ to some new pixel coordinates $\mathbf{p}' \in I_t$. In general we may consider maps over the entire source domain, but typically our edits on the source are restricted to a particular subregion $R_s \subseteq I_s$. We allow this region of interest to be *fuzzy*, defined by the mask function $r(\mathbf{p}) \in [0, 1]$, reaching 1 at pixels fully inside the region of interest and fading to 0 at the boundary of R_s. We define the optimality of our map in terms of a

robust error norm ϕ which measures the difference in color values between each source pixel $\mathbf{p}$ and the corresponding pixel $\mathcal{M}_t(\mathbf{p})$.

Ideally we would optimize over all possible maps, but this is far from tractable. Instead, our method works within the subspace of warp functions spanned by the linear combination of a small number of affine transformations:

$$\mathcal{M}(\mathbf{p}) = \sum_{k=1}^{m} w_k(\mathbf{p})\, T_k \mathbf{p}. \tag{1}$$

In character animation, this subspace of deformations is called linear blend skinning (LBS). Working in the LBS subspace makes the optimization computationally feasible, and when carefully designing the minimization process and the weight functions w_k, this subspace proves to be sufficiently expressive. It leads to accurate and intuitive warps, which have the required balance between smoothness and content-adaptiveness necessary for successful edit transfer.

It is important to note that despite its simplicity Eq. (1) describes a vast space of expressive warps. Because each weight function w_k varies over the domain, the resulting warps are in general far more interesting than constant or even piecewise affine transformations. The same applies to the blends of other per-weight function parameters like bias and gain parameters described in Section 3.4.

3.1 Warp Optimization in LBS Subspace

To achieve a consistent transfer between images, we need a highly accurate alignment of corresponding image regions. A powerful tool for computing such an alignment is the Lucas-Kanade (LK) algorithm and its extensions [Baker and Matthews 2004]. The basic LK procedure computes the parameters of a warp function $\mathcal{M}$ by an iterative minimization of the color mismatch between $I_s(\mathbf{p})$ and the corresponding warped pixels in some target image $I_t(\mathcal{M}(\mathbf{p}))$:

$$\arg\min_{\mathcal{M}} \sum_{\mathbf{p}\in I_s} r(\mathbf{p})\, \phi\left(I_t(\mathcal{M}(\mathbf{p})), I_s(\mathbf{p})\right). \tag{2}$$

For ϕ one typically employs some robust error norm for increased robustness to image inconsistencies like occlusions. Specifically, we employ adaptive thresholding with spatial coherence approximation [Baker and Matthews 2004]. Linear appearance changes can be accounted for within the same framework to compensate for changes in illumination during the matching, detailed in Section 3.4.

We reduce the search space of this optimization by only considering warps $\mathcal{M}$ defined by weighted linear combinations of a small number of affine transformations (see Eq. (1)). If the per-pixel weight functions w_k are precomputed then the only parameters of this subspace are the elements of the affine transformation matrices $T_k \in \mathbb{R}^{2\times 3}$. That is to say, if we have m pairs of weight functions and transformations, then this space is parameterized by only $6m$ degrees of freedom. Our optimization then becomes:

$$\arg\min_{T_k,\ k=1,\ldots,m} \sum_{\mathbf{p}\in I_s} r(\mathbf{p})\, \phi\left(I_t\left(\sum_{i=1}^{m} w_k(\mathbf{p}) T_k \mathbf{p}\right),\ I_s\left(\mathbf{p}\right)\right). \tag{3}$$

This new class of warp functions $\mathcal{M}$ can be readily integrated into the standard "forward additive" variant of the LK algorithm. In practice, however, this variant of LK is computationally inefficient, as it requires a re-computation of the Hessian matrix at each iteration of the algorithm. Instead we derive an alteration of the highly efficient "inverse compositional" variant of LK to search over our parameterized subspace (see, e.g., [Baker and Matthews 2004] for details about the different variants of LK).

In the "inverse compositional" LK a different incremental warp update is utilized, in which the roles of I_s and I_t are exchanged compared to Eq. (2). This enables highly efficient implementations since the Hessian and other expensive steps can be precomputed. However, the sought global warp has to be iteratively composed with the following update rule:

$$\mathcal{M}(\mathbf{p}) \leftarrow \mathcal{M}(\mathbf{p}) \circ \Delta\mathcal{M}(\mathbf{p})^{-1}, \tag{4}$$

where $\Delta\mathcal{M}$ is a warp update computed in every iteration [Baker et al. 2011]. Inserting Eq. (1) gives us:

$$\mathcal{M}(\mathbf{p}) \leftarrow \left[\sum_{k=1}^{m} w_k(\mathbf{p}) T_k \mathbf{p}\right] \left[\sum_{\ell=1}^{m} w_\ell(\mathbf{p}) \Delta T_\ell \mathbf{p}\right]^{-1} \tag{5}$$

where ΔT_k are the updates to the affine transformation matrices T_k. We may use the distributive property to rewrite this as:

$$= \sum_{k=1}^{m} \left[w_k(\mathbf{p}) T_k \left[\sum_{\ell=1}^{m} w_\ell(\mathbf{p}) \Delta T_\ell\right]^{-1}\right] \mathbf{p}. \tag{6}$$

In general, the required inversion of the inner term might lie outside the LBS subspace. However, if our weights are sufficiently content-aware, localized and smooth, we can project the inverse back into that subspace. To do so, we make the following approximation by replacing ΔT_ℓ with ΔT_k in the inner sum:

$$\approx \sum_{k=1}^{m} \left[w_k(\mathbf{p}) T_k \left[\sum_{\ell=1}^{m} w_\ell(\mathbf{p}) \Delta T_k\right]^{-1}\right] \mathbf{p}. \tag{7}$$

To understand the reasoning behind this approximation let us consider the influence of a particular transformation and weight function pair (T_k and w_k) on some pixel $\mathbf{p}$. We can distinguish the following three cases:

1. Pixel $\mathbf{p}$ is dominated by the weight function w_k, i.e., $\forall \ell \neq k : w_\ell(\mathbf{p}) \approx 0$. In this case, the influences of all other transformations ΔT_ℓ with $\ell \neq k$ on $\mathbf{p}$ are negligible and only the contribution for $\ell = k$ remains in the inner sum.
2. Pixel $\mathbf{p}$ has a low weight $w_k(\mathbf{p}) \approx 0$. Consequently, neither the associated transformation T_k nor the inner sum have a significant influence on $\mathbf{p}$ and may be neglected.
3. Pixel $\mathbf{p}$ is influenced by multiple constraints, i.e., $w_k(\mathbf{p}) > 0$ for some set $\mathcal{K}$ of k's. If we assume that our weights are content-aware, then they should fall-off quickly near edges and we may deduce that $\mathbf{p}$ is located within a smoothly varying image region in I_s. Further, it can be expected that the corresponding target region in image I_t is smoothly varying as well. By replacing ΔT_ℓ in the inner sum of Eq. (7) with some $\Delta T_k, k \in \mathcal{K}$, we make the assumption that, in such smoothly varying image regions, constraints with overlapping weight functions undergo a similar transform between images.

Thanks to this approximation, Eq. (7) can now be further simplified by recalling that the weight functions sum up to one at every pixel, resulting in its final form:

$$\mathcal{M}(\mathbf{p}) \leftarrow \sum_{k=1}^{m} \left[w_k(\mathbf{p}) T_k \left[\Delta T_k\right]^{-1}\right] \mathbf{p}. \tag{8}$$

The inner sum of weighted affine transformations has been reduced to a single transform that is easily inverted. Despite the approximation to the true inverse warp update, this solution robustly converges to a pixel-accurate matching even in challenging cases (see Figure 1) and it is very efficient to compute.

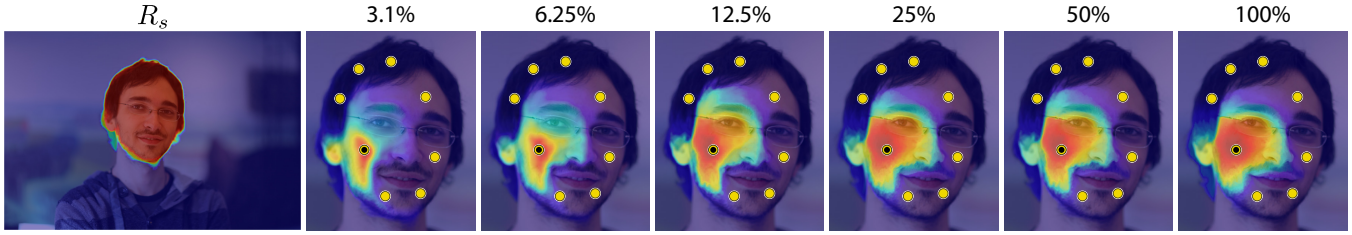


Figure 2: *Left to right: The user-input region of interest R_s on the* Young Man. *We visualize a selected content-aware bounded biharmonic weight function optimized on various downsamplings of the domain. We see diminishing returns past 12.5%, and thus use this resolution for all our remaining examples.*

3.2 Content-Aware Bounded Biharmonic Weights

Our strategy for optimizing Eq. (8) relies on strong assumptions about the weight functions w_k used in our subspace reduction. Namely we assume that our weights possess the following properties: The weights should be sensitive to image content and propagate to similar image areas, dropping at salient image edges. The weights should be local, i.e. have limited spatial interaction range (see the analysis and discussion of Farbman et al. [2010]). Globally-supported weight functions would complicate the multi-view matching due to the difficult control of global image warping. The weights should be smooth (at least C^1) everywhere to achieve smooth warping in smooth image regions [Jacobson et al. 2011], see also Figure 8. The weights should be limited to the $[0, 1]$ range. Otherwise, input constraints would have counterintuitive effects, as also demonstrated in [Jacobson et al. 2011] for deformations.

As summarized in Table 1, existing methods do not have all of these properties, in particular smoothness at fixed values. We therefore extend the smooth bounded biharmonic weights (BBW) [Jacobson et al. 2011] to incorporate image content-awareness. In the following, we first provide a brief summary of the basic BBW for completeness, and then describe our generalization.

Classic BBW. The BBW have been introduced for realtime deformation of 2D images and 3D shapes. Each weight function w_k is associated with a *handle* H_k, which is a region (or just a single point) in the domain fully controlled by that weight function. Hence w_k attains the value of 1 on H_k and 0 on all other handles. The weight functions are defined by optimizing the Laplacian energy subject to constant bound constraints:

$$\underset{w_k,\ k=1,\dots,m}{\arg\min} \sum_{k=1}^{m} \frac{1}{2} \int_{I_s} (\Delta w_k)^2 \,\mathrm{d}x\mathrm{d}y \tag{9}$$

$$\text{subject to: } w_k|_{H_\ell} = \delta_{k\ell}, \quad k, \ell = 1, \dots, m \tag{10}$$

$$\sum_{k=1}^{m} w_k(\mathbf{p}) = 1 \qquad \forall \mathbf{p} \in I_s \tag{11}$$

$$0 \le w_k(\mathbf{p}) \le 1, \ \ k = 1, \dots, m, \quad \forall \mathbf{p} \in I_s. \tag{12}$$

Assuming the 2D region of interest domain R_s is discretized with a triangle mesh Ω with n vertices, the discrete optimization problem for the weight functions $\mathbf{w}_k \in \mathbb{R}^n$ may be written as follows:

$$\underset{\mathbf{w}_k}{\arg\min} \sum_{k=1}^{m} \mathbf{w}_k^T Q \mathbf{w}_k, \tag{13}$$

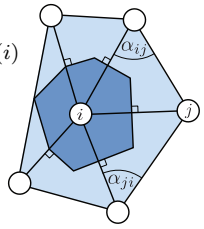
subject to the above constraints (10)-(12). The coefficients matrix of the energy is the discrete biharmonic operator $Q = LM^{-1}L$. The matrices L and M are the stiffness matrix and the diagonal mass matrix, respectively, of the discretization mesh Ω [Pinkall and Polthier 1993]:

Table 1: *Properties of different weight functions.* Harmonic *is a representative for methods such as [Levin et al. 2004; Lischinski et al. 2006],* Global *represents [Farbman et al. 2010] and related approaches that use global affinity,* BBW *is [Jacobson et al. 2011], and* Biharmonic *represents unbounded biharmonic interpolation [Finch et al. 2011].*

Property	Ours	Harmonic	Global	BBW	Biharmonic
Content-awareness	✓	✓	✓		
Local interaction	✓	✓		✓	✓
Smoothness	✓			✓	✓
Boundedness	✓	✓	✓	✓	

$$L_{ij} = \begin{cases} -\frac{1}{2}\left(\cot \alpha_{ij} + \cot \alpha_{ji}\right) & \text{if } j \in \mathcal{N}(i) \\ -\sum\limits_{l \in \mathcal{N}(i)} L_{il} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

M_{ii} = Voronoi area around vertex i,



where $\mathcal{N}(i)$ is the set of mesh neighbors of vertex i, and α_{ij} is the angle opposite of the directed edge from i to j on the incident triangle lying to the left of the edge (if it exists).

The weights resulting from this optimization enjoy smoothness guarantees of a fourth-order energy minimizer, and also locality, boundedness and interpolation properties mentioned above.

Content-Aware Metric. The BBW weights are obviously not content-aware, but notably, their definition depends solely on the handle locations H_k and the Riemannian metric of the domain R_s, imposed by the biharmonic operator. We propose to manipulate this metric in order to adapt the weights to the image content. We map the geometry of the discretization mesh Ω into a higher-dimensional feature space $\mathbb{R}^D$, such that it becomes a 2-manifold in that space. For each vertex $\mathbf{v} \in \Omega$, we concatenate some attributes to its 2D spatial coordinates as additional dimensions. For example, expressing the image colors at the vertices in CIELAB color space, we obtain a 5-dimensional embedding:

$$\mathbf{v}(x, y) \to (x, \ \ y, \ \ s_l\, l(x, y), \ \ s_a\, a(x, y), \ \ s_b\, b(x, y)). \tag{14}$$

The scaling factors s_* are necessary because color value coordinates are not readily proportional to Euclidean pixel coordinates, and furthermore allow for varying the sensitivity of the weights to image content. In all our examples we used $s_* = 16$. See Figure 2 in [Jacobson and Sorkine 2012] for a detailed comparison of choices for this parameter.

Computing optimized weights over the obtained 2-manifold in feature space will provide us with the desired properties, including content-awareness. For instance, in the case of the L*a*b* feature space above, strong image edges will become steep creases in the

5D 2-manifold, making the travel distance on the manifold across the edge long, such that the weight function will have an abrupt derivative across the edge.

To define the BBW optimization problem in feature space, we simply need to adjust the discretization of the bi-Laplacian operator $Q = LM^{-1}L$ to the Riemannian metric of our image domain embedded in the $\mathbb{R}^D$ feature space, i.e., we need to adapt the matrices L and M. The matrix entries have the same formulas of cotangents and areas, but of the mesh triangles embedded in $\mathbb{R}^D$. Angles and areas are often defined via cross products, but this is troublesome in high dimensions. Instead we use the lengths l_{ij}, l_{jk}, l_{ki} of the three sides of a triangle $\mathcal{T}_{ijk}$ to extract the necessary quantities (since lengths are easy to compute in any dimension). The triangle area $\mathcal{A}_{ijk}$ is given by Heron's formula:

$$\mathcal{A}_{ijk} = \sqrt{r(r - l_{ij})(r - l_{jk})(r - l_{ki})} \tag{15}$$

where r is the semi-perimeter $\frac{1}{2}(l_{ij} + l_{jk} + l_{ki})$. Cotangents of the given angles are revealed via trigonometric identities as:

$$\cot(\alpha_{ij}) = \frac{l_{jk}^2 + l_{ki}^2 - l_{ij}^2}{4\mathcal{A}_{ijk}}, \quad \cot(\alpha_{jk}) = \frac{l_{ki}^2 + l_{ij}^2 - l_{jk}^2}{4\mathcal{A}_{ijk}},$$
$$\cot(\alpha_{ki}) = \frac{l_{ij}^2 + l_{jk}^2 - l_{ki}^2}{4\mathcal{A}_{ijk}}. \tag{16}$$

A similar derivation is given in [Meyer et al. 2003]. With these elements we construct and solve the BBW optimization over our image surface in $\mathbb{R}^D$.

Note that we essentially discretize the weight optimization problem using finite elements, thus we enjoy a large degree of mesh independence and convergence under refinement. We typically take the mesh Ω to be the regular (triangulated) pixel grid limited to the region of interest R_s, but it is possible to adaptively mesh the image domain, for example based on an importance sampling density, and our algorithm remains the same. In contrast, previous formulations of weight optimization (e.g. [Lischinski et al. 2006]) usually use finite differences and work on regular pixels grids only. More details on this content-aware metric are available in our technical report provided in the supplemental material.

Implementation details. Solving for the weights w_k as described above amounts to sparse quadratic programming (QP) with constant inequality constraints. We utilize the MOSEK solver [Andersen and Andersen 2000]. Also, following [Jacobson et al. 2011], we solve for each weight function separately, dropping the partition of unity constraint (11) and then normalizing the weights to sum up to 1 (see supplemental material for pseudocode of the whole pipeline).

Solving on meshes with the fine resolution of our source image can be too expensive for interactive performance. Fortunately our weights are resilient to changes in discretization resolution. We may thus downsample our image before optimizing our weights and then upsample each weight function to full resolution using bicubic interpolation. Downsampling too much eventually has a smoothing effect and changes the support regions of each function, however we see diminishing change in the values and supports of our weights as the discretization approaches full resolution (see Figure 2).

3.3 Generating Seed Locations

In the original formulation of BBW, the weight functions are meant to be attached to user-specified handles H_k. These handles are placed by the user directly onto regions she intends to control. Our weights, on the other hand, are hidden from the user, and we *automatically* choose seed locations — where weights are respectively constrained to 1 and 0.

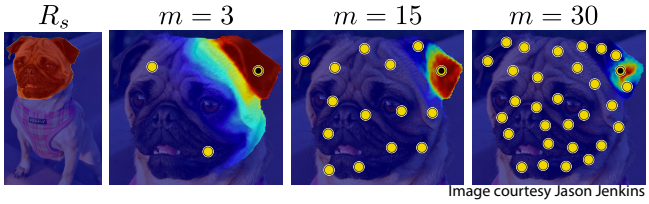


Figure 3: *Left to right: The user-input region of interest R_s on the* Skinny Pug. *We visualize seed locations (yellow dots) for various m values. The content-aware bounded biharmonic weights are shown for a selected seed (black dot).*

Our content-aware metric ensures that our weights fall off at salient image edges, so we design a method to choose seed locations in a manner that approximates uniform distribution over the region of interest R_s and places them away from strong edges. We compute the gradient magnitude of the source image and by applying heat diffusion on its values create a confidence image. We then iteratively take the m highest confidence locations, updating the confidence map after each selection by subtracting a Gaussian kernel with standard deviation $\sqrt{|R_s|/(\pi m)}$. See Figure 3 for a comparison of this distribution method for various numbers of weights m.

Many other distribution methods could be used instead, such as blue noise or dithering techniques; however, our simple method has the advantage of choosing exactly m locations, as well as its computation efficiency and simple implementation.

3.4 Per-Weight Function Bias and Gain

In addition to the robust error norm ϕ we could easily employ two additional degrees of freedom to account for *global* bias and gain [Baker and Matthews 2004]. However, in general different pixels will undergo different appearance changes, so ideally we would introduce bias and gain parameters per pixel. Just like the warp parameters, though, this becomes intractable. Fortunately, the weight functions we have just described are perfect candidates as blending functions for parameterizing a space of pixel-wise bias and gain changes. The same properties listed in Table 1 needed for high-quality spatial deformation in the warp functions make our content-aware BBW weights well-suited for blending appearance parameters. Smooth lighting variations can be handled as well as discontinuities in illumination, despite that we only introduce a pair of bias and gain parameters per weight function. In practice this simply means replacing $I_s(\mathbf{p})$ in Section 3.1 with:

$$I_s(\mathbf{p}) + \sum_{k=1}^{m} w_k(\mathbf{p}) \left(a_k I_s(\mathbf{p}) + b_k\right)) \tag{17}$$

where a_k and b_k represent the bias and gain corresponding to weight w_k. These new degrees of freedom are determined during optimization as in [Baker and Matthews 2004]. Results with varying illumination are shown in Figures 1, 4, 6, 8, 10, 11.

3.5 Initialization

We employ an iterative optimization which needs an initial warp $\mathcal{M}^0$ parameterized by initial affine transformations T_k^0. A standard approach is to use the identity transformation for each and to employ a coarse-to-fine strategy for handling large image displacements. However, the nonlinear nature of our energy causes slow convergence for such trivial initial guesses, and basic coarse-to-fine approaches may fail to converge for complex mappings with many image details. Instead we initialize using a sparse set of SIFT features [Lowe 2004] as follows.

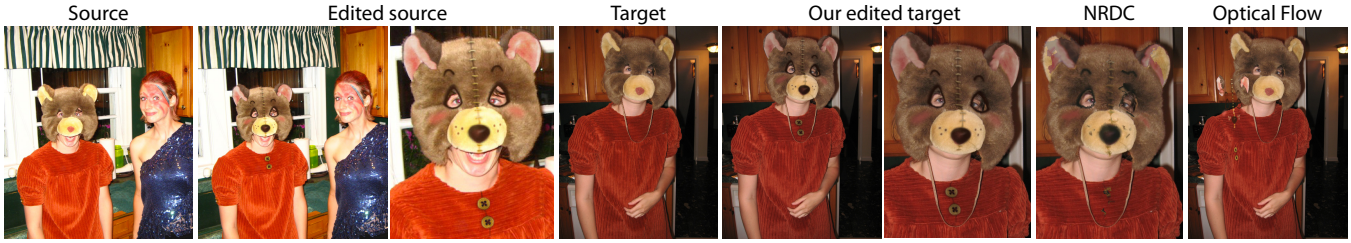


Figure 4: *Left to right: The source image of the* Bear *is edited by a professional artist. Our method finds a good match and transfers the edits faithfully despite significantly different lighting. The matching of [HaCohen et al. 2011] (NRDC) is discontinuous, fragmenting the edits. The method of [Zimmer et al. 2011] (Optical Flow) struggles with the large displacements, failing to find the alignment.*

We use the method of [Lowe 2004] to find SIFT features in the source and target images and find strong correspondences by using the distance ratio test from source to target and target to source. In the ideal case, there are many correct matches all over the region of interest R_s. We use the Delaunay triangulation $\mathcal{T}$ of the source feature locations and the mapping of this triangulation to the matched target feature locations to define a piecewise-affine map $\mathcal{M}_{\text{SIFT}}$. Before computing $\mathcal{M}_{\text{SIFT}}$, nodes causing triangle flips in the target mesh are classified as outliers, removed from consideration, and the map is recomputed. In extreme cases where there are too few matches, one may manually add corresponding points to source and target. Note, however, that neither the SIFT features nor optional manual correspondences are hard constraints (they are only used for the initial guess) and therefore they do not have to be particularly accurate.

We then project the piecewise affine map $\mathcal{M}_{\text{SIFT}}$ to the space of maps spanned by our degrees of freedom, i.e., we find T_k^0 that best reproduce $\mathcal{M}_{\text{SIFT}}$ in a least-squares sense:

$$\underset{T_k^0,\ k=1,\ldots,m}{\arg\min} \sum_{\mathbf{p}\in\mathcal{T}} r(\mathbf{p}) \|\mathcal{M}_{\text{SIFT}}(\mathbf{p}) - \sum_{i=1}^{m} w_k(\mathbf{p}) T_k^0 \mathbf{p}\|^2 \qquad (18)$$

This results in a linear system with $6m$ variables.

If SIFT features are only found in one part of the source image, the above system may be underdetermined as there might exist transformations T_k whose corresponding weight functions w_k are zero or near zero for all pixels in $\mathcal{T}$. In these cases we regularize against the best-fit global affine transformation A found using RANSAC on the SIFT features, adding the following term to the above minimization:

$$\gamma \sum_{\mathbf{p}\in I_s} r(\mathbf{p}) \|A\mathbf{p} - \sum_{i=1}^{m} w_k(\mathbf{p}) T_k^0 \mathbf{p}\|^2 \qquad (19)$$

where γ balances between fitting to $\mathcal{M}_{\text{SIFT}}$ inside $\mathcal{T}$ and matching A everywhere.

4 Experiments and Results

The workflow for our method is simple. The user marks a region of interest on a source image. She may employ any number of methods for the selection (for our examples we use the Quick Selection Tool in [Photoshop 2012]). At this point the content-aware BBW may be precomputed, taking typically on the order of a hundred milliseconds for each weight function. Then the user chooses target images of the same subject as the source. Our method now precomputes maps $\mathcal{M}_t$ for each target image. Optimization times vary depending on the difficulty of the match, but are typically on the order of a few seconds. Now the user is free to apply any range of standard image edits on the source image and in real time see the transferred edits on all targets. To summarize, weights are precomputed once per source image; warps are precomputed once per target image; edits are propagated in real time.

Model	$\lvert I_s\rvert$	$\lvert R_s\rvert$	n	m	s/weight	LK Iter	s/Iter
Lady	640K	196K	3219	11	0.19	7	0.20
Flower Pug	693K	197K	3357	8	0.22	22	0.17
Pantheon	698K	364K	5963	8	0.35	15	0.25
Crying Pug	727K	165K	2852	15	0.17	27	0.88
Corn	786K	22K	453	8	0.05	5	0.12
Bear	786K	136K	2322	15	0.27	26	0.25
Dylan	786K	168K	2858	8	0.16	8	0.18
Church	1279K	568K	10228	11	0.83	30	0.51
Young Man	1314K	204K	3353	11	0.20	12	0.34

Table 2: *Statistics and timings for the examples shown in this paper. $|I_s|$ and $|R_s|$ are the number of pixels in the source image and region of interest respectively; n is the number vertices used for the mesh when computing m weights. We report the number of seconds of precomputation for each weight function* (s/weight)*, the average number of LK iterations* (LK Iter) *and average seconds per iteration* (s/Iter) *over each precomputed target warps.*

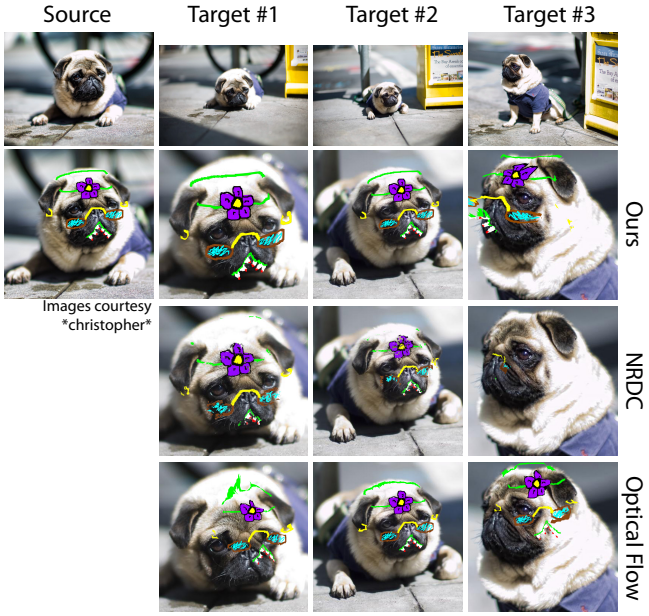


Figure 5: *Top to bottom: Edits on the source image of the* Flower Pug *are transferred to three target images using our method and those of [HaCohen et al. 2011] and [Zimmer et al. 2011]. The rightmost column shows a challenging deformation for which our method fails along with previous methods.*

We tested our implementation on an iMac Intel Core i7 3.4GHz computer with 16GB memory. We report timings and statistics for a

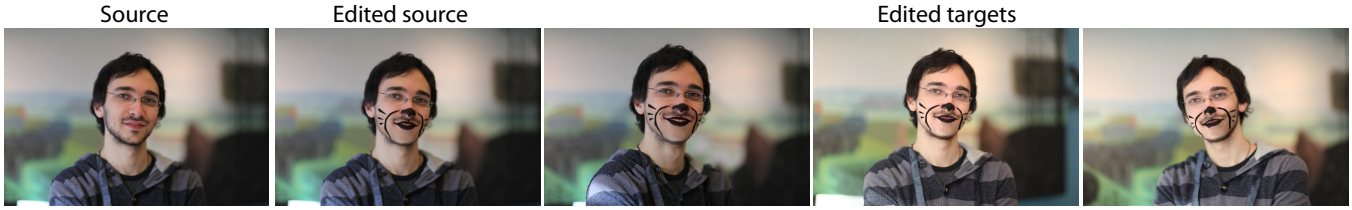


Figure 6: *Left to right: Face paint is added to the* Young Man *and transferred to three other images, which vary in pose and lighting.*

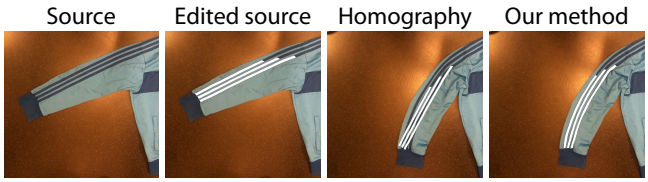


Figure 7: *Comparison of our method to homography-based methods like [Hasinoff et al. 2010]. Note that the homography is forced to keep straight lines straight.*

number of our results in Table 2. Total precomputation times are on the order of a few seconds, which is comparable to those reported by [HaCohen et al. 2011] despite their working on smaller images.

We show a comparison to the non-rigid dense correspondence (NRDC) matching of [HaCohen et al. 2011]. Their method is not designed for detailed edit transfer, and this shows in the lack of continuity and extreme non-surjectivity of the map. In Figure 4, using their method leaves edits fragmented (e.g. around the buttons) and large portions are missing (e.g. on each ear), whereas with our method the edits are seamlessly transferred. We also compare to the optical flow technique of [Zimmer et al. 2011], which struggles to capture large displacements and non-rigid deformation. Figure 5 again demonstrates the advantage of our method. Here we also demonstrate the failure of our method to match an extreme deformation, but even here our edit transfer seems to fail more gracefully than previous work. Figure 7 shows a comparison of our method to homography-based transfer [Hasinoff et al. 2010]. Note that the smoothness of the weights is a necessary condition for converging on a good match; when using the non-smooth weights of [Levin et al. 2004], LK has difficulty finding the correspondence (see Figure 8).

Unless explicitly specified, all our examples converged starting from the fully automatic, SIFT-based initialization described in Section 3.5. Figure 6 shows an example where our method succeeds in converging for two targets using the fully automatic initialization. At first the third target fails, but after the user manually selects just 10 loosely-corresponding points, our method is able to converge and produce a meaningful result.

Our method is robust to changes in lighting (see Figure 10), camera placement (see Figures 1, 9, 13) and subject pose (see Figure 11). Figure 12 shows how the subspace spanned by Eq. (1) is expressive enough to handle non-trivial changes in the *Lady*'s facial expression. The content-adaptiveness of our weights ensures that changes in certain image regions like her cheeks do not effect the edit transfer in other regions separated by salient edges. For a consistent color transfer in the painting applications, e.g., in case of varying exposure, we employed the method of Reinhard et al. [2001], utilizing our per-weight bias and gain estimation.

Occlusions are always challenging for computing image alignments, but thanks to our content-adaptive weights and our employment of a robust error norm, our optimization is able to properly ignore occluded regions such as the bushes and lamppost in Figure 1. To ensure that edits are not transferred on top of occluding objects, we first modulate them according to a threshold on the error image.

Limitations. Like all Lucas-Kanade based techniques, our method assumes some amount of smoothness between source and target and thus struggles to converge to a meaningful matching when the level of granularity in the matching is too small or the images are too fragmented with high frequency details. Defined warp degrees of freedom for each fragment would produce an ill-posed problem with likely no or at best slow convergence. This is a long-standing unsolved problem in image matching, and this limitation is not limited to our method.

In extreme cases our result may strongly depend on the initialization, which can fail to correctly align images when the subjects exhibit too strong deformations, e.g. strong perspective changes (see Figure 5). In such cases, it is possible to ask the user to specify some correspondences to guide the initialization. Areas that were not matched well can be visualized by computing the matching error (Eq. (2)) in order to guide the user to the regions where correspondences should be provided. Since we only use these manual points indirectly in the initialization and not as hard constraints, the user may choose them casually and let our method correct their final positions. Another possibility would be to use the warps of previous works such as [Hasinoff et al. 2010; HaCohen et al. 2011] as initialization for ours.

Our current method is limited to transferring edits made within the region of interest on the source image and correspondingly the warped region of interest on the target. Exploring methods to extrapolate our warp function beyond the user-selected region of interest is an interesting and tangible direction of future work.

5 Conclusions

We presented a method for image edit propagation using content-aware warping, constructed as a spatially-varying weighted combination of affine deformations. The weighting functions are computed by fourth-order energy minimization over the image manifold in feature space, with constant bound constraints, which makes them smooth everywhere except at strong image edges. This leads to successful non-rigid registration between a source image and multiple target images of the same subject using our adaptation of the Lucas-Kanade framework.

In future work we would like to improve both precomputation steps of our algorithm: weight computation and matching. For the quadratic programming we are currently using an interior point solver (MOSEK) which cannot efficiently benefit from initial guesses. However, good initial guesses can be constructed by solving on a lower resolution and/or employing the unbounded biharmonic solution. For example, it is evident that the solution is stable and consistent for different image resolutions (see Figure 2). Moreover, when many weight functions are used, the support of each individual weight function significantly decreases, implying that optimization on a much smaller area than the whole region of interest is possible.

The warps for each target are independent of each other and thus their required computation is trivially parallelized. We leave this for future work, as it would strengthen the exploration of transferable edits to see all matched images as soon as possible.

Figure 8: *Edits are added to* Dylan*'s face (left) and transferred to multiple targets with differing subject pose and camera parameters. Using the nonsmooth harmonic weights of [Levin et al. 2004] creates difficulties during matching (middle). Our weights smoothly approach seed locations and find the correct match (right).*

Figure 9: *Decorations added to the* Couch *are transferred to multiple targets with varying viewpoints.*

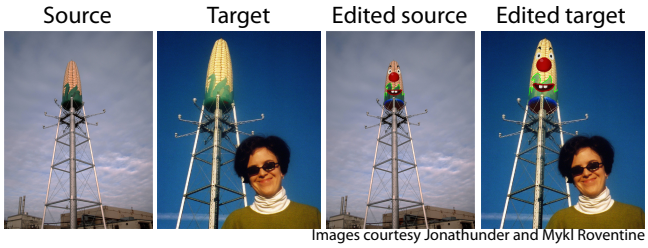


Figure 10: *Edits added to this corn water tower are faithfully transferred to a similar image taken from a different viewpoint and under different lighting conditions.*

Figure 11: *Our method transfers edits made on the* Crying Pug *to multiple targets despite varying, non-rigid subject poses and contrasting illumination and shadows.*

Acknowledgements

We are grateful to Yael Pritch and Ilya Baran for internal reviews; Alessia Marra for providing professional edits; Maurizio Nitti, Leila Gangji, Dylan Zehr, Katie Anderson, and Alex Schiebel for modeling; `Mykl Roventine`, `Jonathunder`, `ZeroOne`, `<<<TheOne>>>`, `Erik Hagreis`, `roblisameehan`, `*christopher*`, and `Jason Jenkins` for their Creative Commons images; and Sam Hasinoff, Yoav HaCohen and Henning "Hank" Zimmer for helping with comparisons.

References

AGARWALA, A., HERTZMANN, A., SALESIN, D., AND SEITZ, S. M. 2004. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph. 23*, 3, 584–591.

AN, X., AND PELLACINI, F. 2008. AppProp: all-pairs appearance-space edit propagation. *ACM Trans. Graph. 27*, 3.

ANDERSEN, E. D., AND ANDERSEN, K. D. 2000. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*. Kluwer Academic Publishers, 197–232.

BAKER, S., AND MATTHEWS, I. 2004. Lucas-Kanade 20 years on: A unifying framework. *Int. J. Comput. Vision 56*, 3, 221–255.

BAKER, S., SCHARSTEIN, D., LEWIS, J. P., ROTH, S., BLACK, M. J., AND SZELISKI, R. 2011. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision 92*, 1, 1–31.

BARNES, C., SHECHTMAN, E., GOLDMAN, D. B., AND FINKELSTEIN, A. 2010. The generalized patchmatch correspondence algorithm. In *Proc. ECCV*.

BERTHOUZOZ, F., LI, W., DONTCHEVA, M., AND AGRAWALA, M. 2011. A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations. *ACM Trans. Graph. 30*, 5, 120.

BHAT, P., ZITNICK, C. L., SNAVELY, N., AGARWALA, A., AGRAWALA, M., CURLESS, B., COHEN, M., AND KANG, S. B. 2007. Using photographs to enhance videos of a static scene. In *Proc. EGSR*, 327–338.

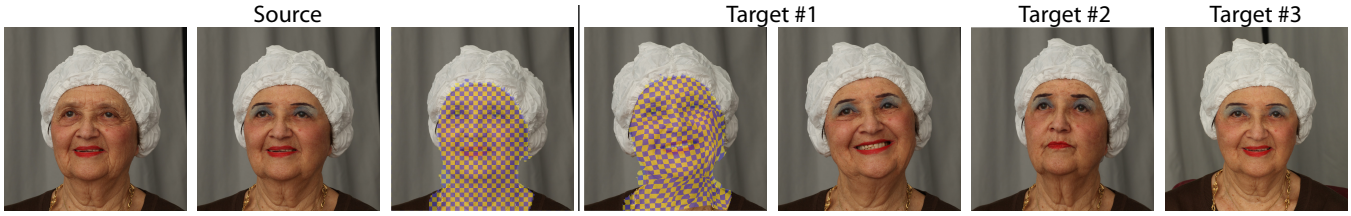


Figure 12: *Edits adding makeup to and removing wrinkles from the* Lady *are transferred to other images in which she dramatically changes her facial expression and head orientation.*

Figure 13: *Our method transfers edits on an image of the* Pantheon *to different images, varying in viewpoint and lighting. The rightmost image shows an extreme change in viewpoint, but our transfer remains faithful even to the orientation of the text edits on the pediment.*

BITOUK, D., KUMAR, N., DHILLON, S., BELHUMEUR, P. N., AND NAYAR, S. K. 2008. Face swapping: automatically replacing faces in photographs. *ACM Trans. Graph. 27*, 3.

BRAND, M., AND PLETSCHER, P. 2008. A conditional random field for automatic photo editing. In *Proc. CVPR*.

COHEN-OR, D., SORKINE, O., GAL, R., LEYVAND, T., AND XU, Y.-Q. 2006. Color harmonization. *ACM Trans. Graph. 25*, 3.

FARBMAN, Z., FATTAL, R., LISCHINSKI, D., AND SZELISKI, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph. 27*, 3.

FARBMAN, Z., FATTAL, R., AND LISCHINSKI, D. 2010. Diffusion maps for edge-aware image editing. *ACM Trans. Graph. 29*, 6.

FATTAL, R., CARROLL, R., AND AGRAWALA, M. 2009. Edge-based image coarsening. *ACM Trans. Graph. 29*, 1, 1–11.

FATTAL, R. 2009. Edge-avoiding wavelets and their applications. *ACM Trans. Graph. 28*, 3.

FINCH, M., SNYDER, J., AND HOPPE, H. 2011. Freeform vector graphics with controlled thin-plate splines. *ACM Trans. Graph. 30*, 6, 166:1–166:10.

HACOHEN, Y., SHECHTMAN, E., GOLDMAN, D. B., AND LISCHINSKI, D. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. 30*, 4.

HASINOFF, S. W., JÓŹWIAK, M., DURAND, F., AND FREEMAN, W. T. 2010. Search-and-replace editing for personal photo collections. In *Proc. ICCP*.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. *ACM Trans. Graph. 26*, 3.

JACOBSON, A., AND SORKINE, O. 2012. A cotangent laplacian for images as surfaces. Tech. Rep. 757, ETH Zurich, April.

JACOBSON, A., BARAN, I., POPOVIC, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph. 23*, 3, 689–694.

LI, Y., ADELSON, E. H., AND AGARWALA, A. 2008. Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. *Comput. Graph. Forum 27*, 4.

LI, Y., JU, T., AND HU, S.-M. 2010. Instant propagation of sparse edits on images and videos. *Comput. Graph. Forum 29*, 7.

LISCHINSKI, D., FARBMAN, Z., UYTTENDAELE, M., AND SZELISKI, R. 2006. Interactive local adjustment of tonal values. *ACM Trans. Graph. 25*, 3, 646–653.

LIU, C., YUEN, J., TORRALBA, A., SIVIC, J., AND FREEMAN, W. T. 2008. Sift flow: Dense correspondence across different scenes. In *Proc. ECCV: Part III*, 28–42.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2, 91–110.

LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. International Joint Conference on Artificial Intelligence*.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*. 35–57.

PHOTOSHOP. 2012. *Version 12.0.4*. Adobe Systems, Inc.

PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugates. *Experiment. Math. 2*, 1.

RAV-ACHA, A., KOHLI, P., ROTHER, C., AND FITZGIBBON, A. W. 2008. Unwrap mosaics: a new representation for video editing. *ACM Trans. Graph. 27*, 3.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. and Applications 21*, 5, 34–41.

SAND, P., AND TELLER, S. J. 2004. Video matching. *ACM Trans. Graph. 23*, 3, 592–599.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision 47*, 1–3, 7–42.

ZIMMER, H., BRUHN, A., AND WEICKERT, J. 2011. Optic flow in harmony. *Int. J. Comput. Vision 93*, 3, 368–388.