

Guiding Image Manipulations using Shape-appearance Subspaces from Co-alignment of Image Collections

Chuong H. Nguyen^{1,2} Oliver Nalbach¹ Tobias Ritschel^{1,2} Hans-Peter Seidel^{1,2}

¹MPI Informatik ²Saarland University, MMCI

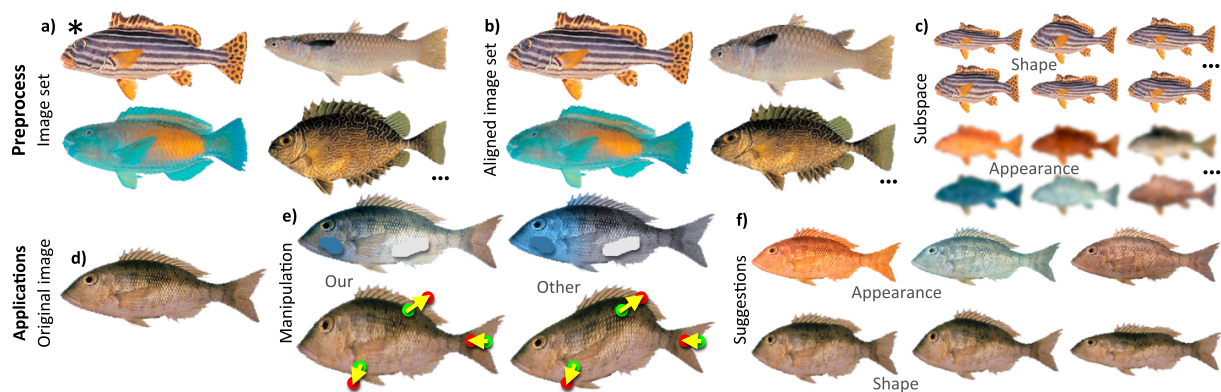


Figure 1: Our approach automatically aligns a set of images (a) showing instances of one object class to a “reference image” (b) and constructs a subspace of shape and appearance (c). This subspace is used to guide manipulations of a different image (d). When painting a colored stroke, our subspace is used to propagate plausible colors to plausible locations (e, top) or to restrict shape deformations to plausible shapes (e, bottom). Additionally, the space is used to suggest relevant shape and appearance alternatives (f).

Abstract

We propose a system to restrict the manipulation of shape and appearance in an image to a valid subspace which we learn from a collection of exemplar images. To this end, we automatically co-align a collection of images and learn a subspace model of shape and appearance using principal components. As finding perfect image correspondences for general images is not feasible, we build an approximate partial alignment and improve bad alignments leveraging other, more successful alignments. Our system allows the user to change appearance and shape in real-time and the result is “projected” onto the subspace of meaningful changes. The change in appearance and shape can either be locked or performed independently. Additional applications include suggestion of alternative shapes or appearance.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Application packages

1. Introduction

The ability to manipulate digital images is a fascinating opportunity taken both by professional artists producing digital content as well as by casual users editing their home photo collection. While the option to change shape and appearance of an image into any possible other shape or appearance sounds like a good idea at first, in practice too many possible options actually

decrease the human ability to make the right decision [TK81]. Therefore, the right balance between generality and reduction of choices has to be found. We devise a computational way to automatically suggest such choices for a class of images.

One option to restrict manipulations in a meaningful way is the construction of subspaces within the space of all possible images. Here, images are understood as points in a

high-dimensional space. Images from a certain class, such as faces, do not cover the entire space but a lower-dimensional manifold which is likely related to the human mental representation of this class [SL00]. This idea was first proposed for human faces, both in 2D [TP91, CET01] and 3D [BV99], 3D human bodies [ACP03] or other specialized 3D shapes [CF13], where the manifold is approximated using principal component analysis (PCA).

However, to construct subspaces, training data needs to be aligned manually by means of careful selection of a template and intervention [CET01], such as clicking correspondences [BV99]. This excludes casual users, such as a hypothetical biologist seeking to create a subspace of leaf images for a class of plants captured in a collection of 1000 non-calibrated images or even non-professional users striving to create a subspace of a special breed of dogs from a dozen of images. No simple and efficient way exists to align a large collection of images allowing for the creation of subspaces.

In this work, we propose a simple alignment strategy applicable to casual image collections such as images of butterflies. As finding perfect or even sufficiently correct correspondences for all images in a large set is infeasible, we first build an incomplete partial alignment that is later completed to align all images.

Our key application is appearance and shape manipulation in images as seen in Fig. 1. Selecting the right color for a fish and assigning it to the right spot is difficult as the selection is from a high-dimensional shape-appearance space. While selection of color alone is already challenging, selecting shape and respecting the combination of shape and appearance are even harder. Our system allows the user to change appearance or shape in real-time and the result is restricted to meaningful changes by finding a close image in the subspace corresponding to the image class. If desired, changes in appearance and shape can be locked to result in correlated changes of both. As the images forming a subspace lack detail, our approach only captures the change in shape and appearance and transfers this change to the image being edited.

Our particular contributions are:

- Efficient partial alignment of images in common image collections with varying appearance.
- Completion of partial alignment to a global alignment for all images using an alignment graph.
- A novel interactive user interface for appearance and shape subspace manipulation that preserves detail.

2. Background

In this work we are concerned with intuitive manipulation of shape and appearance in images. The two main existing approaches either propagate spatially localized manipulations of some control primitives to the rest of the image, or suggest entire images as global alternatives.

The first line of work builds on propagation of sparse manipulations of appearance or shape to the full image. To manipulate appearance, sparse strokes defining a certain appearance for some locations in the image are propagated to the rest of the image, either using edge-aware (local) [LLW04, LFUS06, GO11] or all-pairs propagation (global) approaches [PL07, AP08] or a unification of both global and local methods [XYJ13]. For example, a red ball is covered by a green stroke, and other red and spatially close red balls are changed to become green. The goal of the propagation often is to fill smooth regions and stop propagation at edges. For manipulating shape, instead of making strokes, sparse control points are moved in the image and a plausible deformation is propagated to the rest of the image [Boo89, ACOL00, IMH05, SMW06]. As an example, a user clicks two control points such as the feet of a character in the image to remain fixed and places a third one on the head. Moving the control point on the head, the latter follows the displacement, but the feet remain in place. A key challenge is to produce an intuitive deformation response and avoid shape distortion and overlap. Deformations that locally preserve distances are of particular interest here [ACOL00, IMH05, SMW06]. As a second strategy, plausible suggestions for alternatives can be made if a parametric space is available [MAB*97, SSC09]. Such alternatives are usually applied to appearance where they are not spatially localized but change the entire image.

For both approaches, the user either knows where and how to change the image, or changes everything at once. Also, combinations of appearance and shape are not addressed. We want to exploit data available on the Internet to restrict manipulation and suggestion and capture the conjoint variation of shape and appearance in a subspace. Using exemplar images directly without creating a subspace allows to present sample suggestions e. g., in response to sketches [LZC11], but does not explain how to apply these changes to new instances. This is complementary to the chosen method from above by adding the restriction to a subspace of appearance and shape changes. The difficulty is to acquire such a space.

Subspaces have a long history of use in computer graphics where they come in form of Eigenfaces [TP91], morphable models [BV99] or active appearance models [CET01]. Their applications range from faces [BV99] over human body poses [ACP03] to other objects such as demonstrated for sea animals by Cashman and Fitzgibbon [CF13]. Recently, the idea of finding subspaces has been extended to 3D objects when either given a collection of 3D objects [OLGM11] or even just a collection of images [PFZG10].

The underlying difficulty in constructing subspaces is the requirement of correspondences between training exemplars. Early work used either controlled conditions to acquire the exemplars [BV99] or manual alignment. Modern semi-automatic approaches to align image pairs e. g., for morphing [LLN*14] often combine a data and a smoothness term with user-defined constraints. Our approach targets automatic alignment, such as done for 3D shapes [SZGP05, FKY08, OLG11] and

deformations limited to rigid parts [KY07]. Optical flow algorithms [LK81, TBKP12, LWA*12] are designed to align video frames with their temporally adjacent frames. However, they are not suitable for image pairs containing large deformations, significant structural differences or changes in appearance, such as found when aligning two butterflies. Recently, several image alignment techniques such as SIFT flow [LYT11] or Patch Match [BSFG09] were proposed that are able to deal with drastically different scales and would allow to align training images. While they are successful in producing plausible images by shuffling image patches, they do not yield meaningful deformation fields. Non-rigid dense correspondence (NRDC) [HSGL11] is based on Patch Match and works best for pairs of images depicting similar regions acquired by different cameras or under different lighting conditions undergoing non-rigid deformations. The idea of co-aligning an image collection is demonstrated in ImageWeb [HGO*10] which assumes a partial, per-region affine transformation. That is great for browsing image collections with repeating objects but does not capture smoothly varying deformations such as between two faces. Furthermore, no considerations are made for the spatial placement of the regions themselves.

Cheng et al. [CZM*10] use a boundary-band map to find repetitions of similar objects in single images. Goldberg et al. [GCZ*12] present a semi-automatic system that leverages a collection of aligned images to improve image manipulations related to some of our applications. Different from our approach, manual intervention is used and the alignment is limited to align the outer boundary using shape contexts [BMP00]. Furthermore, their approach does not create a space and cannot be used to understand the variation of e. g., horses, as a whole, such as required for our suggestion applications. Our approach is strictly automatic and uses all structures that can be matched reliably, including internal structures, such as the eyes of animals. Aligning these properly is essential for creating an expressive space.

Outside of computer graphics, the most successful alignment methods proposed in computer vision use involved learning and graph matching machinery for alignments [CMC*09]. We will show how producing a collection of approximate alignments is good enough to construct a useful subspace.

3. Our Approach

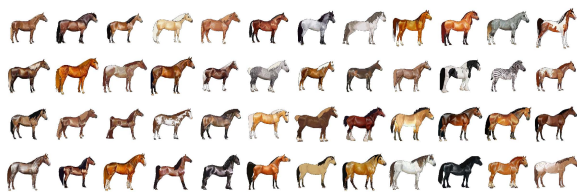


Figure 2: Some instances from the example class “Horses”.

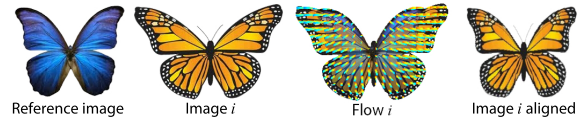


Figure 3: All exemplars are aligned to a reference image, here shown for the case of a butterfly class. In the 3rd column, the gradient-colored lines show the flow direction (from blue to orange) that align (4th col.) the exemplar (2nd col.) to the reference (1st col.).

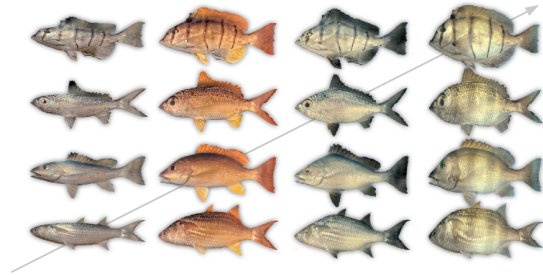


Figure 4: We separate shape and appearance of a collection of images (here shown on the diagonal) to create a space that contains arbitrary, continuous re-combinations (off-diagonal elements).

Overview After acquiring a set of images, our approach proceeds in three steps: alignment of example data (Sec. 3.1), construction of a subspace from the aligned images (Sec. 3.2) and application to novel user interfaces (Sec. 4). The first two steps are performed offline while our user interfaces always provide real-time feedback.

Input of our system is a set of unaligned RGB images from one class (Fig. 2). Those images have to show instances in roughly the same pose on a constant background. If the background is not constant, it has to be removed manually. The instances have different appearance, slightly different pose and perspective and are centered. Our example classes were collected from Internet image queries, manually removing outliers in terms of the above requirements. No other manual intervention was performed.

The core of our approach automatically aligns every image to a reference image (Fig. 3). An alignment is a deformation field, that, when applied to the respective instance from the set, produces an image with the same appearance but in the shape of the reference, i. e., shape and appearance are factored out (Fig. 4). From the aligned images and their deformation fields, a subspace of appearance and shape is created which allows for novel image manipulation interfaces. Such user interfaces constrain the result to be part of the subspace in terms of appearance, shape, or both.

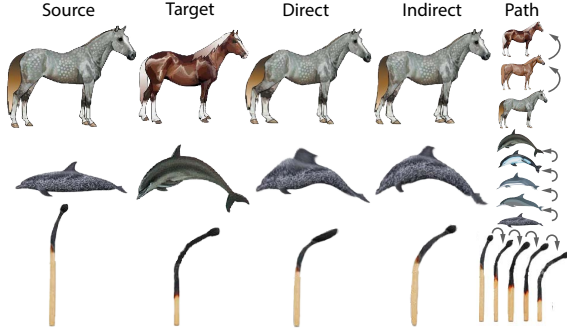


Figure 5: Comparison between the alignment of source and target exemplars using either direct or indirect of alignment. The alignment paths between exemplars are shown on the right.

3.1. Alignment

Alignment is difficult as our images are not taken under controlled conditions and vary drastically in their appearance. These challenges are addressed as follows: First, we predict how well each pair of images might align. Next, from this information, we identify a reference image to which all other images might align well. The actual alignment to this reference is performed as a concatenation of simple alignments along the shortest path in a graph over the images. For each pair that needs to be aligned, we first find the best per-pixel correspondences and a measure of confidence, before we blur areas with low confidence and regularize the resulting flow to locally rigid transformations. Each step will be detailed in the following paragraphs.

Alignment Graph Directly aligning a large number of images with a reference image is likely to fail as appearance and shape of our input images show substantial variation. Establishing an alignment between similar images however is routinely done. Regrettably, this does not suffice to align every image with the reference image as required for our needs. As a solution, we perform indirect alignment: We create an alignment graph, where similar images are aligned directly and the alignment of dissimilar image pairs is found as a sequence of edge hops (an *alignment path*) in this graph. This idea has been successfully applied to the alignment of multi-view stereo images [Hub02] and 3D shapes [HZG*12].

To create the alignment graph, we first define a distance metric d between images. The Gram matrix A of this metric, with $A_{a,b} = d(a,b)$ holds in each entry the distance between each pair a, b .

As image distance d , a robust measure is required since common metrics such as L_2 or perceptual ones such as SSIM [WBSS04] are not resilient to the, at this point unknown, deformations. However, we would like to use a metric that reports a small difference, even if the images differ by a small deformation, as long as their appearance is similar. Conversely,

a high value should be returned if the appearance is very different or the deformation is large. Let, $a, b \in [0..1]^2 \rightarrow \mathbb{R}^3$ be images encoded in the LAB-color space. We define d by

$$d(a,b) = \int_{[0..1]^2} \min_{\mathbf{y}, \mathbf{z} \in [-s..s]^2} \|a(\mathbf{x} + \mathbf{y}) - b(\mathbf{x} + \mathbf{z})\|_2 d\mathbf{x},$$

i. e., as the sum of the min-pooling LAB-color-difference in a neighborhood of size $s = 0.01$. Similar spatial pooling is believed to be used by the human visual system too and has shown to provide robustness in recognition [SWB*07].

When viewed as an adjacency matrix, A implicitly defines a fully-connected graph with the image distances as edge weights. We further add a small constant $\epsilon = 0.1$ to each entry of A to penalize longer paths in the graph. The reference image is chosen to be the one with minimum total length of the shortest paths to all other images. To align each image a with the reference, first, the shortest path to the reference according to A is found. Then, the pairwise alignments along this path are performed as described in the next subsection and finally concatenated. Fig. 5 shows several examples where a concatenation of alignments (indirect alignment) improves over a single direct alignment. In addition to these “backward” alignments, we analogously compute the “forward” deformation fields, aligning the reference to each respective image. The latter are later used to build the subspace of shape variations (Sec. 3.2). Note that the shortest paths from the reference to other images form a shortest-path tree (see supplementary material for a visualization).

To improve the quality of the resulting subspace, we discard images that have a low alignment-quality, which is judged by the SSIM image difference between the reference and the backward-deformed image. Fig. 11 shows the forward and backward alignments for several classes.

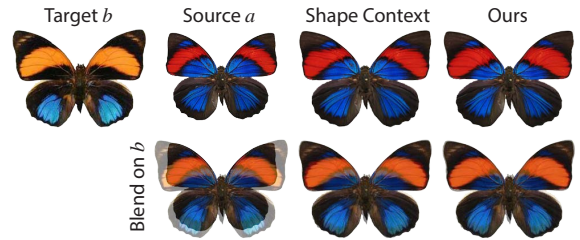


Figure 7: Alignment of a source (1st row, 2nd col.) to a target image (1st row, 1st col.) using Shape Context [BMP00] (3rd col.) and our approach (4th col.). The second row shows blends of the respective image in the first row and the target image. Note how our method improves the alignment of interior structures, e. g., of the red stripe on the wings of the butterfly to the orange one, while Shape Context only aligns the outer boundary.

Alignment We seek to find a flow field that is smooth and aligns the images well but cannot assume it is produced by a simple camera motion, even if it was 3D such as in

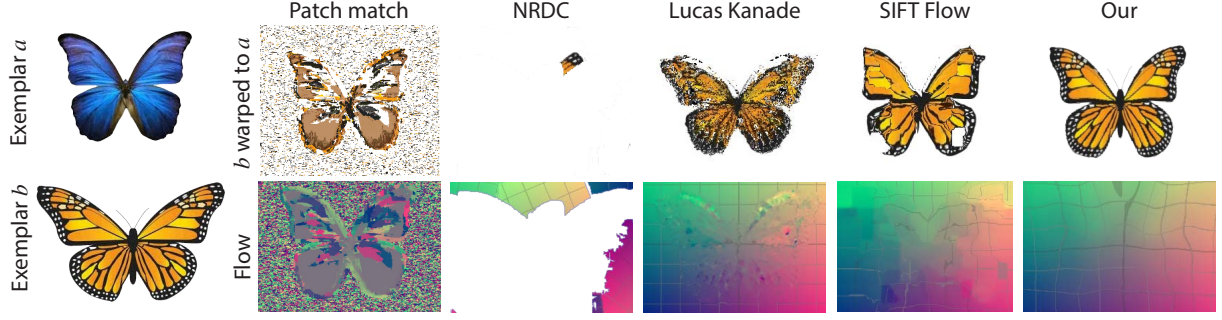


Figure 6: Alignment of exemplar b to exemplar a (1st col.) using different methods. As our exemplars differ drastically, both, PatchMatch [BSFG09] (2nd col.) and NRDC [HSGL11] (3rd col.), failed to produce a reasonable flow. Optical flow methods based on the assumptions of small disparity and image similarity such as Lucas Kanade [LK81] (4th col.) or Simple Flow [TBKP12] are not designed to work for our problem. SIFT Flow [LYT11] (5th col.) allows robust matching between objects of different appearance at the cost of only piecewise smooth flow. Our method (6th col.) works best as it is specifically designed for images containing a single object.

alignment for structure-from-motion. Possible techniques to deal with such problems are SIFT Flow [LYT11], Patch Match [BSFG09] or NRDC [HSGL11]. We found however that, while those techniques are good at aligning images that are different in a plausible way, this comes at the cost of creating flow fields that are often close to meaningless. E. g., in the case of our butterflies they are more successful than our approach to transfer appearance from one exemplar to another, but at the price of a puzzle-like flow field that reassembles image a using image b in a piecewise smooth manner (Fig. 6). Silhouette-based matching [BMP00, CZM*10] might succeed to align boundaries (Fig. 7) but fails to align internal structures (Fig. 8). Finding a plausible flow is a key requirement to capture the shape variation underlying our data, though.

To this end, we devise the following alignment (Fig. 8) between two images a and b which is computed for each pixel independently and in parallel: Each $\mathbf{x} \in [0..1]^2$ in b searches over a two-dimensional neighborhood of size $\mathbf{r} \in [0..1]^2$ in a to find the best corresponding pixel $f_{b,a}(\mathbf{x})$, i. e., the one with the lowest cost $E_{b,a}$:

$$f_{b,a}(\mathbf{x}) = \arg \min_{\mathbf{y} \in [\mathbf{x}-\mathbf{r}, \mathbf{x}+\mathbf{r}]} E_{b,a}(\mathbf{x}, \mathbf{y}), \quad \text{with}$$

$$E_{b,a}(\mathbf{x}, \mathbf{y}) = E_{b,a}^{\text{dat}}(\mathbf{x}, \mathbf{y}) + w_{\text{mag}} E^{\text{mag}}(\mathbf{x}, \mathbf{y}),$$

$$E_{b,a}^{\text{dat}}(\mathbf{x}, \mathbf{y}) = \frac{\int_{[-l..l]^2} w_s(\mathbf{z}) \|b(\mathbf{x} + \mathbf{z}) - a(\mathbf{y} + \mathbf{z})\|_2^2 d\mathbf{z}}{\int_{[-l..l]^2} w_s(\mathbf{z}) d\mathbf{z}},$$

$$E^{\text{mag}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 / \|\mathbf{r}\|_2$$

where $E_{b,a}^{\text{dat}}$ is the data term defined as the sum of the weighted LAB-color difference of the corresponding patches of size l in a and b , $w_s(\mathbf{z}) = \exp(-\sigma_s \|\mathbf{z}\|_2^2 / (2l^2))$ is a spatial smoothness weighting function, controlled by σ_s [TBKP12]. The flow magnitude term $E^{\text{mag}}(\mathbf{x}, \mathbf{y})$, weighted by w_{mag} , constrains the flow vectors to be as small as possible.

Besides the corresponding pixel's location $f_{b,a}(\mathbf{x})$, we also

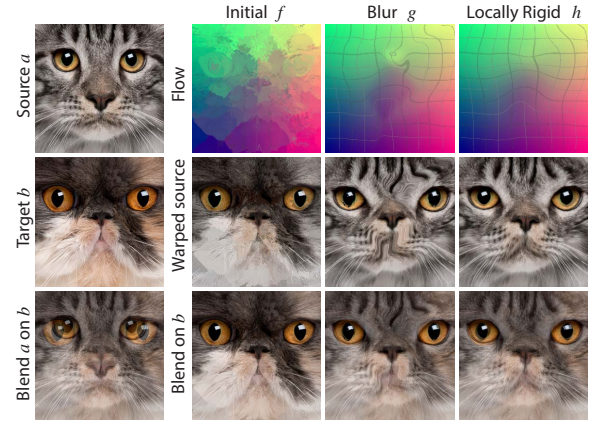


Figure 8: Given source and target images with different appearance (1st col.), we compute an initial flow and its confidence (cf. Fig. 9, 3rd col., bottom). This initial flow field is blurred using the confidence as guidance and regularized to enforce locally rigid transformations (1st row, 2nd to 4th col.). The second row shows the warped source using the respective flow from the first row. The third row shows the blends between the warped source and the target.

compute a measure of confidence $c_{b,a}(\mathbf{x})$ as the product of two factors: The first is the curvature κ of the energy function $E_{b,a}(\mathbf{x}, f_{b,a}(\mathbf{x}))$, a common indicator of reliability in stereo correspondence problems [TMDSA08]. The second factor is an agreement check, to test if the flow from a to b is the same as the flow from b to a : $(1 - \|\Delta_{b,a}(\mathbf{x}) + \Delta_{a,b}(f_{b,a}(\mathbf{x}))\|_2 / (2\|\mathbf{r}\|_2))^\gamma$, where γ is a parameter to control the function and $\Delta_{b,a}(\mathbf{x}) = f_{b,a}(\mathbf{x}) - \mathbf{x}$. We then remove all areas with confidence below a certain threshold. Fig. 9 illustrates the effect of the two factors and the thresholding on our alignment. For brevity, we denote $\Delta_{b,a}(\mathbf{x})$ by $\Delta(\mathbf{x})$ and $c_{b,a}(\mathbf{x})$ by $c(\mathbf{x})$ in the following.

The confidence $c(\mathbf{x})$ is used to replace unreliable ar-

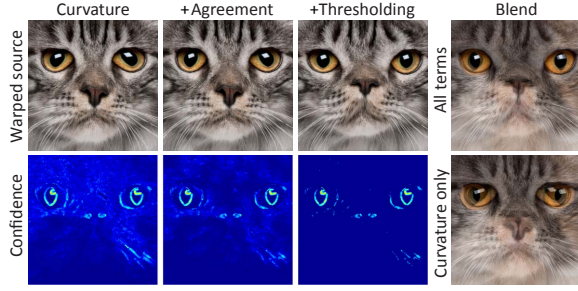


Figure 9: Our alignment computes the confidence from the curvature of the matching quality (1st col.), by an agreement check term (2nd col.) and by removing low-confidence areas (3rd col.). An alignment using all three terms (4th col., top) improves over using only the first term (4th col., bottom).

eas of $\Delta(\mathbf{x})$ by extrapolating from more confident areas [TBKP12, LWA*12]. This is achieved by the following modified blur filter:

$$g(\mathbf{x}) = \mathbf{x} + \frac{\int_{[0..1]^2} w(\mathbf{x}, \mathbf{y}) \Delta(\mathbf{y}) d\mathbf{y}}{\int_{[0..1]^2} w(\mathbf{x}, \mathbf{y}) d\mathbf{y}},$$

$$w(\mathbf{x}, \mathbf{y}) = \exp(-\sigma_d \|\mathbf{x} - \mathbf{y}\|_2^2) \exp(-\sigma_c c(\mathbf{x}) \|\mathbf{x} - \mathbf{y}\|_2^2) c(\mathbf{y}).$$

The first term of the blur kernel w is the spatial smoothness term, controlled by the parameter σ_d . Furthermore, the blur kernel has a two-fold dependency on the confidence c : First, pixels with a high confidence are less affected by other pixels (second term), σ_c controls the steepness of the function. Second, pixels with a high confidence contribute more to the value of other pixels (third term).

The resulting blurred flow field g is now smooth but might distort the shape. An improved flow h is found by mapping the flow in a neighborhood of each pixel to the closest flow in this neighborhood being a rigid transformation. This is done using the flow as point correspondences and orthogonalization [Hor87, SMW06]. In discrete settings, we use a neighborhood of size 32×32 pixels for our results. The final output is a flow field that extrapolates from high-confidence areas to fill unreliable areas by a smooth and locally-rigid deformation. The orthogonalization of the flow field is similar to our proposed locally-rigid color transfer (Sec. 4.1), but deals with 2D deformation fields instead of 3D colors. Fig. 10 shows how our orthogonalization can be used to improve the flow field generated using other approaches, nevertheless, our alignment achieves better quality.

Discussion Our method is simple but efficient for the difficult problem of aligning substantially different images that undergo non-rigid deformations without manual intervention. At each step (Fig. 8), the result for every pixel is computed independently and in parallel. In a discrete setting, for a pair of images with size 256×256 pixels, we set $\mathbf{r} = 80 \times 80$ pixels, $l = 8$ pixels, $\sigma_s = 15$, $w_{\text{mag}} = 0.1$, $\gamma = 5$, $\sigma_d = \sigma_c = 100$ and the computation of the deformation field takes 2 seconds. For

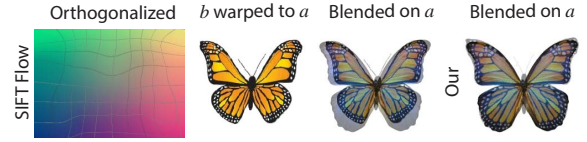


Figure 10: Orthogonalization of the flow field generated using SIFT Flow (Fig. 6, 2nd row, 5th col.). The two exemplars are shown in the first column of Fig. 6. The first column shows the orthogonalized flow. Next are the warped source (2nd col.) generated using this flow and its blend with the target (3rd col.). Finally, we show the blend (4th col.) of our result (Fig. 6, 1st row, 6th col.) and the target.

a reef fish collection of $N = 50$ exemplars, it takes less than 3 minutes to construct the alignment graph and to align the fish along the shortest paths. The complexity of our algorithm is $\mathcal{O}(N^2)$ for both adjacency matrix and single-source shortest path construction, which are fast for realistic values of N . The bottleneck is the alignment to the reference image along the shortest paths. Albeit the time complexity being only $\mathcal{O}(N)$ (since the shortest paths form a tree with $N - 1$ edges, i. e., alignment steps), the individual alignments are costly.

Similar to Lang et al. [LWA*12], we use a set of confident areas to construct a flow field. However, different from their method, our initial flow and confidence are dense [TBKP12]. Furthermore, as their framework aims for scene flow, an edge stopping-based confidence propagation was exploited to improve the quality; in our case, as images come from drastically different sources, we propagate the flow solely based on the confidences. High-confidence areas, including, but not limited to, the global boundary as for shape contexts [BMP00] are propagated to low-confidence areas. Yet, compared to boundary-based approaches, we are also able to match more general internal features (Fig. 8) without manual intervention [GCZ*12].

Still, our method has limitations. First, by using an alignment graph, we assume that the input contains enough image pairs with similar appearance. Second, we use rotationally variant features based on patch difference and flow magnitude. As a result, our method fails to align drastically rotated objects. Those unsuccessful alignments however can be detected using SSIM and removed from the construction of the subspace. Third, as for the construction of our subspace we do not need pixel-accurate alignment (Sec. 3.2), we focus on a simple approach aiming scalability. Further processing might be necessary for applications requiring high quality alignment. Our confidence contains many false positives; although we try to filter out those areas by an agreement check and thresholding, they strongly affect the quality of the reconstructed flow. Last, finding the initial flow is the bottleneck of our method's performance. As our images are diverse in appearance, we need to search over large neighborhoods ($\mathbf{r} = 80 \times 80$ pixels for all of our results) in a to search for $f_{b,a}(\mathbf{x})$, the best corresponding pixel in a for \mathbf{x} in b . Optimizing this step would greatly improve scalability.



Figure 11: Alignment of different exemplars (columns) from several classes (rows). For each class, the first column shows the reference image, the first row shows the input, the second row shows the forward alignment of the reference to the input and the third row shows the backward alignment of the input to the reference. The color below each exemplar indicates the match quality determined by SSIM, cf. the legend on the right. Smaller values indicate better alignment quality. Instances marked with a tick improved by using indirect alignment while instances marked with a cross were excluded because their image difference after alignment remained high.

3.2. Subspace Construction

Given the appearance, in form of the aligned images (Fig. 11, the 3rd row of every class) and shape variations, in form of the forward deformation fields, we compute subspaces using PCA [BV99]. An image with n pixels is considered a point (*image vector*) in a $5n$ -dimensional space: 3 dimensions for RGB color and 2 for the forward deformation field, per pixel. For the color, if the input images come with an alpha channel, the RGB values of transparent pixels are set to the average of the RGB values of all other images where alpha is non-zero.

The PCA is performed on the set of all image vectors, resulting in a set of basis vectors which, in combination with an average, captures the variation in shape and appearance best with respect to the L_2 -norm. Each such “basis image” represents one direction of conjoint variation of appearance and shape.

To control the relative importance of appearance and shape, the respective components in the image vectors are multiplied by weights before the PCA ($0 \leq \alpha \leq 1$ for color and $1 - \alpha$ for shape). Alternatively, using the extreme cases where $\alpha \in \{0, 1\}$, we can also create shape and appearance subspaces independently. In the following, we assume that this weighting is performed before “projecting” any data to the subspace and, consequently, undone after reconstruction from the subspace by dividing by the same respective weights.

For all classes shown, we observed that the first $m = 10/20/30$ basis vectors $\{\mathbf{b}_j\}_{j=1..m}$ with the largest corresponding eigenvalues λ_j contain more than 87/97/99% of the deformation and 72/85/93% of the appearance variation. We keep $m = 20$ basis vectors to represent the space. A spatial resolution of $n = 32 \times 32$ was found to be sufficient to capture the amount of detail present in the data used. As the precision is limited by the alignment and the number of exemplars per class, we cannot expect fine details to be reproduced in practice.

4. Applications and Results

Our approach allows for a range of novel user interactions when manipulating a *query image* unknown at training time. The query image is to be aligned with the reference image, either using our approach or manually. If the image comes with a segmentation in form of an alpha matte, this matte is also used. The basic idea of all applications is to use the subspace to restrict the options a user has to interact with an image to a few plausible ones.

4.1. Shape and Appearance Manipulation

Given a user-manipulated query image u in RGB space from a known class, with a known deformation and appearance change, we can restrict the manipulation to become a plausible

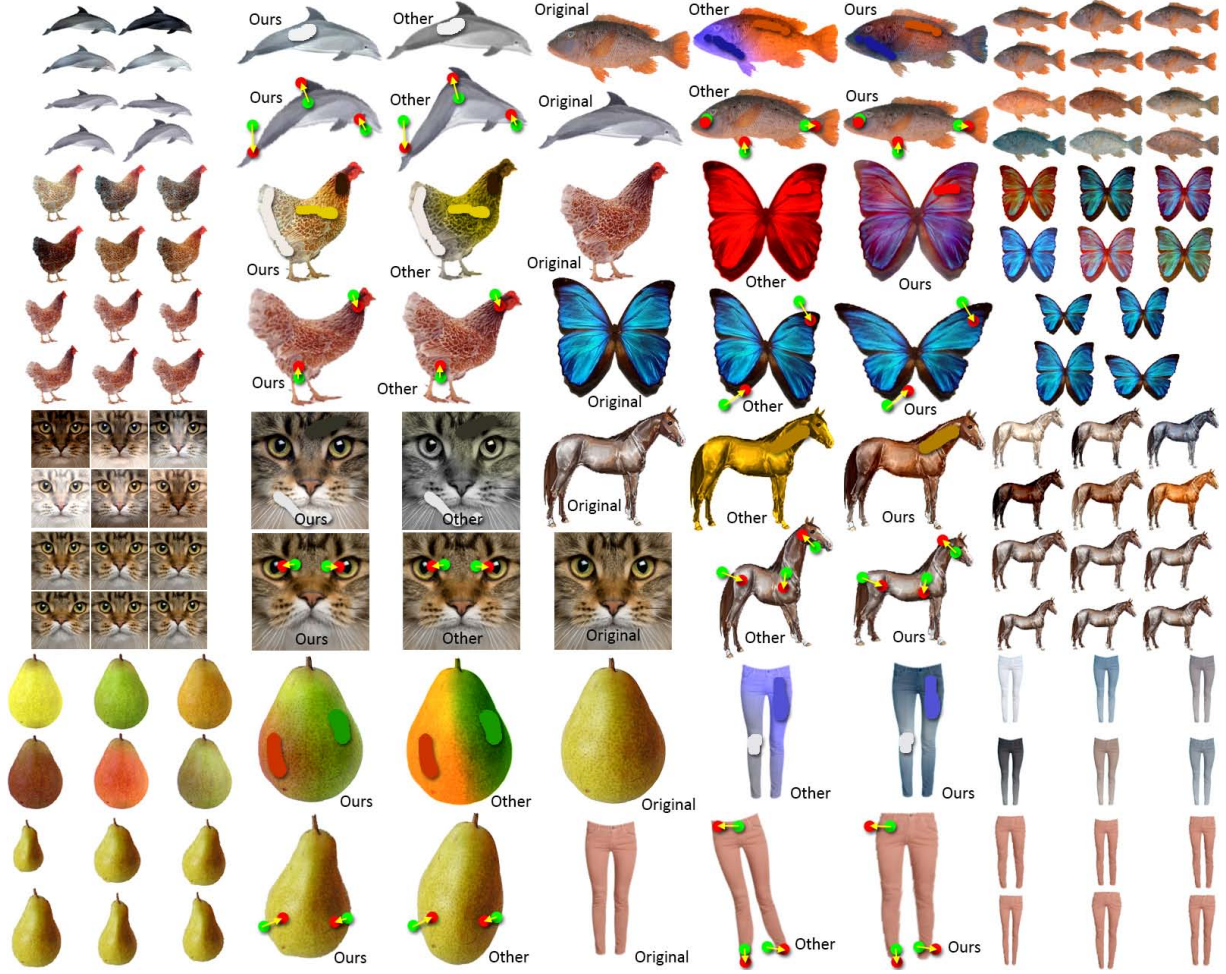


Figure 12: Results of our method (“Ours”) and other methods (“Other”) applied to images from several classes (“Original”). Shape manipulation is compared to Schaefer et al.’s rigid MLS approach [SMW06], appearance manipulation to Gastal and Oliveira’s colorization system [GO11]. For every class, we show suggestions for appearance (top) and shape (bottom). Please see Sec. 4.1 and Sec. 4.2 for a discussion.

one by constructing its image vector $\mathbf{v} \in \mathbb{R}^{5n}$ (cf. Sec. 3.2) and finding the closest point $\bar{\mathbf{v}}$ which can be represented in the subspace using numerical minimization.

All processing happens with respect to the reference shape, so u is first warped to the reference and then down-sampled to obtain the color components of \mathbf{v} . The deformation components correspond to the (down-sampled) deformation field of u w. r. t. the reference image.

We adopt an idea from geometric modeling using blend shapes [SLS*12]. The image vector $\bar{\mathbf{v}}$ which we regard as the closest to \mathbf{v} inside the subspace is the vector with the coordinates \mathbf{x} that minimizes

$$\sum_{i=1}^{5n} \mathbf{k}_i \left\| \mathbf{v}_i - \sum_{j=1}^m \mathbf{x}_j \mathbf{b}_{j,i} \right\|^2 + \mu \sum_{j=1}^m \frac{\mathbf{x}_j^2}{\lambda_j}.$$

The first term is the data term, forcing the reconstruction of $\bar{\mathbf{v}}$ to be close to \mathbf{v} , while the second term is a prior term that gives more weight to coordinates \mathbf{x} which are plausible in the subspace [SLS*12]. The vector \mathbf{k} is used for spatial weighting. For the color components, we use a 20 times higher weight at the positions of the painted strokes than for the rest of the image. This also allows us to apply our method to the colorization of gray images by setting the weight of non-stroke pixels to 0, as demonstrated in the supplementary material. We also exclude pixels from optimization that are not part of the query image, i. e., have an alpha value of zero, by assigning them zero weight. For the deformation components, we use a smooth fall-off of the weights around the source controls of the user’s constraints. The weights for both, color and deformation components, are normalized separately. Finally, μ is a scalar regulating the prior’s contribution.

The resulting guidance $\bar{\mathbf{v}}$ is then transferred back to the original, high-resolution query image u as follows. First, we up-sample the color and deformation components of $\bar{\mathbf{v}}$ separately using bicubic up-sampling. The deformation, which is w. r. t. the reference image, is transformed to be relative to the instance's original shape before we apply it to the query image. As the deformations are expected to be low-frequency, no special measures are taken during up-sampling.

Appearance Transfer We preserve high-frequency appearance details of the query image u which are not captured in the subspace itself. To this end, at each pixel i of u , we find the moving least squares (MLS) solution for the rigid transformation [SMW06] in color space that aligns the query image's colors around i to best fit those of the guidance $\bar{\mathbf{v}}$, similar to affine transformations used to transfer patch appearance [SPDF13].

These transformations can be computed independently and in parallel for each pixel i as follows. Let $\mathbf{p}_j, \mathbf{q}_j \in \mathbb{R}^3$ be the query and guidance pixel's colors, respectively, at pixels j in a square patch around i . Let \mathbf{w}_j be Gaussian weights depending on the spatial distance of j to i , which have been multiplied by the respective alpha values at j in the query and guidance image and normalized such that $\sum_j \mathbf{w}_j = 1$. We find the rotational matrix \mathbf{R}_i that minimizes $\sum_j \mathbf{w}_j \|\mathbf{R}_i(\mathbf{p}_j - \bar{\mathbf{p}}) - (\mathbf{q}_j - \bar{\mathbf{q}})\|_2^2$, where $\bar{\mathbf{p}} = \sum_j \mathbf{w}_j \mathbf{p}_j$ and $\bar{\mathbf{q}} = \sum_j \mathbf{w}_j \mathbf{q}_j$ are the weighted centroids of \mathbf{p}_j and \mathbf{q}_j , respectively. We use a polar decomposition $\mathbf{R}_i \mathbf{S}_i = \mathbf{A}_{\mathbf{pq}} + \delta \mathbf{I}$ where $\mathbf{A}_{\mathbf{pq}} = \sum_j \mathbf{w}_j (\mathbf{p}_j - \bar{\mathbf{p}})(\mathbf{q}_j - \bar{\mathbf{q}})^\top$, similar to [MHTG05]. The parameter $\delta > 0$ controls the regularization which is necessary to avoid singular matrices. Combining \mathbf{R}_i with the translation vector $\mathbf{t}_i = \bar{\mathbf{q}} - \bar{\mathbf{p}}$ yields the complete rigid transformation. Fig. 13 compares our appearance transfer method to alternative approaches. We use a patch size of 32×32 . The influence of the patch size is analyzed in the supplemental material.

Most notable, manipulations in shape lead to changes in appearance and vice versa. If this is not desired or required, manipulation of shape and appearance can also be performed independently by using a subspace created from shape or appearance alone.

Discussion Typical results for manipulation of shape and appearance are shown in Fig. 12. In the following, we will discuss a selection. The dolphin space is created from 29 exemplars. Deforming the straight dolphin into a curved version using the common approach leads to unnatural bending that is not found in our approach. The change of color in the reef fish (a space created from 50 exemplars) propagates in a plausible way to all body parts while the reference struggles with the subtle texture of the fish. Changing the posture of the chicken (40 exemplars) is easy to accomplish using only two constraints which only lead to rotation for the common approach. For the case of the butterfly, the preservation of symmetry is particularly striking when using our manipulation tools. When moving the eyes of the cat further apart, the nose will move up to keep cat-like

proportions, while the MLS deformed cat appears unnatural. Assigning an unlikely color to a horse (72 exemplars), can still produce a reasonable result with our approach because of the subspace restriction and regularization we use. Using our approach, we can easily create a realistic green pear with a red spot (47 exemplars) while the common method fails to position the spot properly. From the space of jeans, we can easily produce classic stonewashed jeans whereas the reference neither knows about the correct extent of the bright spots, nor about the symmetry. Combined manipulation of shape and appearance is demonstrated in Fig. 14.

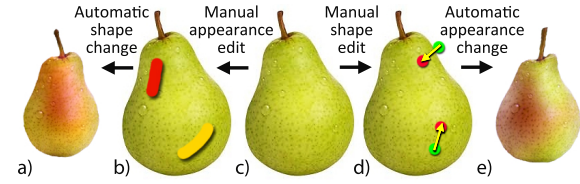


Figure 14: Co-manipulation of appearance and shape: The subspace has captured the fact that smaller pears tend to be more red than larger ones. When making a pear (c) smaller (d), its color changes to reddish at the same time (e). Conversely, painting the pear with red and yellow (b) will also result in a smaller pear (a).

All PCA-related computations are carried out on a standard CPU while still providing real-time performance due to our modest subspace resolution. The image processing components are computed on the GPU at interactive rates.

Study We compared subspace-aware against common shape and appearance manipulation interfaces in a user study. Common manipulations were provided by means of Gastal and Oliveira's interactive colorization [GO11] for appearance and a rigid MLS image deformation [SMW06] for shape. Appearance and shape manipulation tasks were performed separately. For each task, 7 subjects were asked to adjust appearance (by painting color strokes) or shape (by setting position constraints) of a source image to become visually similar to a target image; once using our and once using the common interfaces. Pairs of source and target images, both not from the training set, from three classes (butterfly, pear, fish) were presented to the subjects in combination with one of the interfaces in random order. The task had to be finished either within a 20 second time budget or in an open-ended setting.

In the open-ended setting, for appearance manipulation, the average time-to-finish was 42.5s using our and 67.0s using a common interface (significant at $p < .01$, paired t -test; large effect size: Cohen's $d = 0.9$). For shape manipulation, the average time-to-finish was 40.0s (our) and 58.0s (common interface), the average number of constraints used was 2.95 (ours) and 4.57 (common interface), cf. Fig. 15 (all significant at $p < .01$, paired t -test; large effect sizes: Cohen's

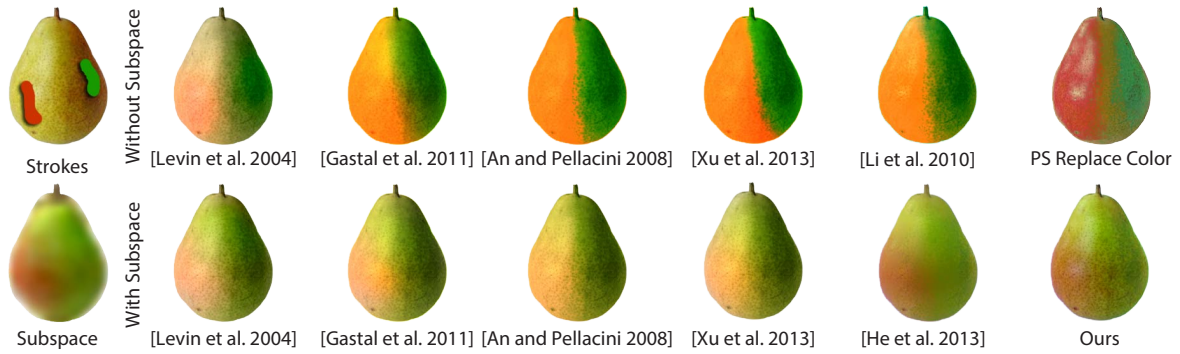


Figure 13: Comparison of color transfer methods. The first column shows the original with the user's strokes (top) and the image reconstructed from the subspace (bottom). Starting in the second column, the first row shows stroke-based colorization using the methods of Levin et al. [LLW04], Gastal and Oliveira [GO11], An and Pellacini [AP08], Xu et al. [XYJ13], Li et al. [LJH10] and Adobe Photoshop's "replace color" function, respectively. On the second row, the second to the fifth column shows colorization given the subspace guidance using the methods used on the first row, the sixth column shows guided image filtering [HST13] using subspace as guidance and finally our locally-rigid appearance transfer on the final column.

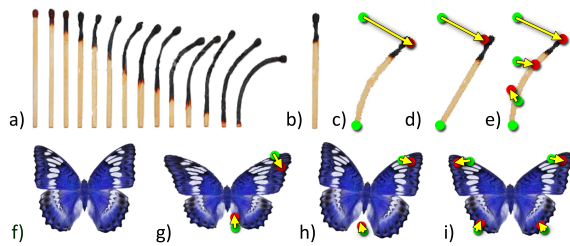


Figure 15: Our approach achieves meaningful deformations using a lower number of constraints. Top row: A subspace is created from frames of a match animation (a). As we learned the most important deformation, two constraints (c) on the match (b) will be enough to curve it (c). Using MLS (d), more constraints (e) are necessary to achieve a smooth bending. Bottom row: Using only two constraints on a butterfly (f), we achieve an interesting new shape (g). For MLS (h), more constraints (i) are required to preserve symmetry.

$d = 0.8$ and 0.9). This indicates less effort with our approach compared to the common one.

After verifying that our interface can indeed simplify manipulation, we were interested to see if it could also improve the quality of the results. In a rating task, for each image pair produced in the manipulation study, we asked a second group of 12 subjects to choose the one they found visually more similar compared to the target image. The subjects preferred our method in 84.6% (limited-time appearance), 83.0% (open-ended appearance), 77.9% (limited-time shape) and 75.8% (open-ended shape) of the manipulation tasks. This indicates with statistical significance ($p < .01$, binomial test) that our shape and appearance interfaces outperform common approaches in terms of quality, too.

In a final study, we tested whether the common colorization

approach by Gastal and Oliveira [GO11] could be improved by using our subspace, independent of our suggested locally-rigid appearance transfer. The average time-to-finish was 54.0s for this approach. The difference to color manipulation without using a subspace is statistically significant ($p < .01$, ANOVA), indicating that our subspace indeed improves task performance for appearance manipulation, independent of the transfer method used. Preference ratings by subjects lead to a similar significant overall picture. The supplementary material contains further details on all studies.

4.2. Shape and Appearance Suggestions

The principal directions of variation of shape and appearance can also serve as suggestions for manipulations. This allows for an interface similar to "Design Galleries" [MAB*97], just that the parameter space is found automatically. We support suggestions for both shape and appearance in combination or in isolation.

Suggestions are presented by showing potential positive and negative steps along the n_d most relevant eigen-directions. The number of interesting directions n_d can be determined by using heuristics based on the respective eigen values. In practice, we use the smallest number of eigen vectors such that the sum of their eigen values is larger than a given percentage, e.g., 70%, of the sum of all eigen values, adding them in decreasing order of their eigen values.

For determining a number of sufficiently diverse and specific suggestions, we found it best to use the intercept points of lines, going through the initial point along the eigen vectors, with an n_d -dimensional sphere of a certain radius, which is scaled with respect to the distance of the most-extreme image in the training set from the origin. We assume a previous normalization of the eigen vectors and values. These guidance

samples are then transferred as done for manipulation and shown as altered copies of the original.

Discussion Typical results for suggestion of shape and appearance are shown in Fig. 12. For the dolphin, typical shapes like jumping or bent exemplars show up as shape suggestions. The appearance suggestions for the given reef fish come up in a range between blue, orange and gray. The most important shape changes for butterflies turn out to affect the size and shape of the pair of wings, respectively, and are symmetrical like the butterflies themselves. For horses, their posture is an important component, as well as their height and length. The appearance space of pears nicely brings up all variants of more or less ripe pears while the shape suggestions produce small, large or differently tilted fruits.

4.3. Manipulation of Complex Images

To apply our method to complex images, a user has to provide a mask for an instance of an image class, e. g., for a dolphin in an underwater image (Fig. 16, top). The segmented object can then be aligned to the reference from the respective class using our automatic alignment (by constructing the alignment graph and accumulating multiple alignments along the shortest path as described in Sec. 3.1) or manual alignment and both, manual edits and suggestions, work the same as previously described. The altered instance is then pasted in, either in place of the original or as an additional element. Fig. 16 demonstrates both use cases.

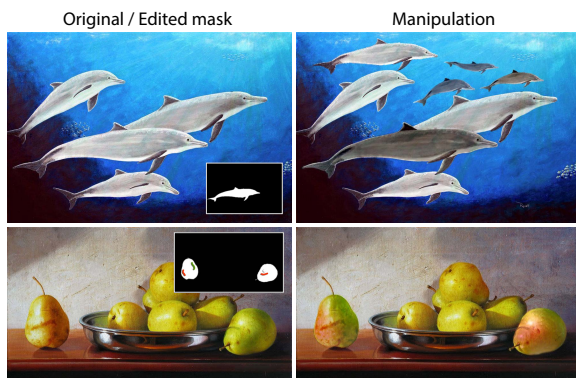


Figure 16: Subspace-supported manipulation in complex images. After a user has marked an instance of the class (inset), it can be used for cloning with new shape, new appearance or both from the space (dolphins) or to constrain color manipulation (still life).

5. Discussion and Conclusion

This paper proposes a system to restrict the manipulation of shape and appearance in an image to the subspace of valid changes which we learn from a collection of exemplar images.

We use an automatic alignment of exemplars to create the subspace from sample images.

Our system is subject to several limitations. First, the use of PCA assumes that the deformation and appearance changes are linear. This however is only true for simple classes and invalidated in the presence of strong perspective and occlusion. Future work could try to compensate for camera motion, perspective and occlusion in a dedicated way. We are not able to deal with multiple objects that appear in one image and always assume the image either shows a single instance or that the latter has been manually selected beforehand. To be applicable to an instance from one class in a general image, an alignment has to be available. Finally, our method is not yet ready to change arbitrary objects appearing, for example, in a home photo collection if the subspace of the object class is not available. Expanding the collection of objects would greatly increase the practicability of our method.

In future work, we would like to generalize the approach to other domains such as 3D meshes and video to cover the mutual relation of artistic style, texture, reflectance, or other properties. While we preserve details of the target instance, we would also like to use our alignment to construct non-linear subspaces to capture more detailed variations.

Acknowledgements We would like to thank Stefanie Wuhler and James Tompkin for discussion, Mathieu Flinders and Oskar Elek for the voice-over, Bernhard Reinert, Petr Kellnhofer, Oliver Klehm, Oskar Elek and Stefanie Wuhler for proofreading and all anonymous reviewers for constructive comments. We also thank Sebastian Schuon and Sergii Khomenko (StyLight GmbH) for providing us with the Boot and Jeans datasets.

References

- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH* (2000), pp. 157–164. 2
- [ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.* 22, 3 (2003), 587–594. 2
- [AP08] AN X., PELLACINI F.: Approp: All-pairs appearance-space edit propagation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27, 3 (2008), 40:1–40:9. 2, 10
- [BMP00] BELONGIE S., MALIK J., PUZICHA J.: Shape context: A new descriptor for shape matching and object recognition. In *NIPS* (2000), vol. 2, p. 3. 3, 4, 5, 6
- [Boo89] BOOKSTEIN F. L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE PAMI* 11, 6 (1989), 567–585. 2
- [BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 28, 3 (2009), 24:1–24:11. 3, 5
- [BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH* (1999), pp. 187–194. 2, 7
- [CET01] COOTES T. F., EDWARDS G. J., TAYLOR C. J.: Active appearance models. *IEEE PAMI* 23, 6 (2001), 681–85. 2

- [CF13] CASHMAN T. J., FITZGIBBON A. W.: What shape are dolphins? Building 3D morphable models from 2D images. *PAMI* 35, 1 (2013), 232–244. 2
- [CMC*09] CAETANO T., MCAULEY J., CHENG L., LE Q. V., SMOLA A.: Learning graph matching. *IEEE PAMI* 31, 6 (2009), 1048–1058. 3
- [CZM*10] CHENG M.-M., ZHANG F.-L., MITRA N. J., HUANG X., HU S.-M.: Repfinder: Finding approximately repeated scene elements for image editing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 4 (2010), 83:1–83:8. 3, 5
- [FKY08] FENG W.-W., KIM B.-U., YU Y.: Real-time data driven deformation using kernel canonical correlation analysis. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27, 3 (2008), 91. 2
- [GCZ*12] GOLDBERG C., CHEN T., ZHANG F.-L., SHAMIR A., HU S.-M.: Data-driven object manipulation in images. *Comp. Graph. Forum (Proc. EUROGRAPHICS)* 31, 2pt1 (2012), 265–274. 3, 6
- [GO11] GASTAL E. S. L., OLIVEIRA M. M.: Domain transform for edge-aware image and video processing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011), 69:1–69:12. 2, 8, 9, 10
- [HGO*10] HEATH K., GELFAND N., OVSJANIKOV M., AAN-JANEYA M., GUIBAS L.: Image webs: Computing and exploiting connectivity in image collections. In *Proc. CVPR* (June 2010), pp. 3432–3439. 3
- [Hor87] HORN B. K.: Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4, 4 (1987), 629–642. 6
- [HSGL11] HACHOEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011), 70:1–70:10. 3, 5
- [HST13] HE K., SUN J., TANG X.: Guided image filtering. *IEEE PAMI* 35, 6 (2013), 1397–1409. 10
- [Hub02] HUBER D. F.: *Automatic three-dimensional modeling from reality*. PhD thesis, Carnegie Mellon U, Pittsburgh, 2002. 4
- [HZG*12] HUANG Q.-X., ZHANG G.-X., GAO L., HU S.-M., BUTSCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 31, 6 (2012), 167. 4
- [IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 1134–41. 2
- [KY07] KOKKINOS I., YUILLE A.: Unsupervised learning of object deformation models. In *Proc. CVPR* (2007), pp. 1–8. 3
- [LFUS06] LISCHINSKI D., FARBMAN Z., UYTENDAELE M., SZELISKI R.: Interactive local adjustment of tonal values. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3 (2006), 646–653. 2
- [LJH10] LI Y., JU T., HU S.-M.: Instant propagation of sparse edits on images and videos. *Comp. Graph. Forum (Proc. Pacific Graphics)* 29, 7 (2010), 2049–2054. 10
- [LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Proc. IJCAI* (1981), pp. 674–679. 3, 5
- [LLN*14] LIAO J., LIMA R., NEHAB D., HOPPE H., SANDER P., YU J.: Automating image morphing using structural similarity on a halfway domain. *ACM Trans. Graphics* 33, 3 (2014). 2
- [LLW04] LEVIN A., LISCHINSKI D., WEISS Y.: Colorization using optimization. *ACM Trans. Graph.* 23, 3 (2004), 689–94. 2, 10
- [LWA*12] LANG M., WANG O., AYDIN T., SMOLIC A., GROSS M.: Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 4 (2012), 34:1–34:8. 3, 6
- [LYT11] LIU C., YUEN J., TORRALBA A.: SIFT flow: Dense correspondence across scenes and its applications. *IEEE PAMI* 33, 5 (2011), 978–994. 3, 5
- [LZC11] LEE Y. J., ZITNICK C. L., COHEN M. F.: Shadowdraw: real-time user guidance for freehand drawing. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011), 27. 2
- [MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUMMLER W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH* (1997), pp. 389–400. 2, 10
- [MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 471–478. 9
- [OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph.* 30, 4 (2011), 33. 2
- [PFZG10] PRASAD M., FITZGIBBON A., ZISSERMAN A., GOOL L. V.: Finding Nemo: Deformable object class modelling using curve matching. In *Proc. CVPR* (2010), pp. 1720–27. 2
- [PL07] PELLACINI F., LAWRENCE J.: Appwand: editing measured materials using appearance-driven optimization. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26, 3 (2007), 54. 2
- [SL00] SEUNG H. S., LEE D. D.: The manifold ways of perception. *Science* 290, 5500 (2000), 2268–69. 2
- [SLS*12] SEOL Y., LEWIS J. P., SEO J., CHOI B., ANJO K., NOH J.: Spacetime expression cloning for blendshapes. *ACM Trans. Graph.* 31, 2 (2012), 14. 8
- [SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3 (2006), 533–40. 2, 6, 8, 9
- [SPDF13] SHIH Y., PARIS S., DURAND F., FREEMAN W. T.: Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 32 (2013), 200:1–200:11. 9
- [SSCO09] SHAPIRA L., SHAMIR A., COHEN-OR D.: Image appearance exploration by model-based navigation. *Comp. Graph. Forum (Proc. EUROGRAPHICS)* 28, 2 (2009), 629–38. 2
- [SWB*07] SERRE T., WOLF L., BILESCHI S., RIESENHUBER M., POGGIO T.: Robust object recognition with cortex-like mechanisms. *IEEE PAMI* 29, 3 (2007), 411–426. 4
- [SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 488–495. 2
- [TBKP12] TAO M., BAI J., KOHLI P., PARIS S.: Simpleflow: A non-iterative, sublinear optical flow algorithm. *Comp. Graph. Forum (Proc. EUROGRAPHICS)* 31, 2pt1 (2012), 345–353. 3, 5, 6
- [TK81] TVERSKY A., KAHNEMAN D.: The framing of decisions. *Science* 211 (1981), 453–458. 1
- [TMDSA08] TOMBARI F., MATTOCCIA S., DI STEFANO L., ADDIMANDA E.: Classification and evaluation of cost aggregation methods for stereo correspondence. In *Proc. CVPR* (2008), pp. 1–8. 5
- [TP91] TURK M., PENTLAND A.: Eigenfaces for recognition. *J Cog. Neuroscience* 3, 1 (1991), 71–86. 2
- [WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE TIP* 13, 4 (2004), 600–612. 4
- [XYJ13] XU L., YAN Q., JIA J.: A sparse control model for image and video editing. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 32, 6 (2013), 197:1–197:10. 2, 10