

Smooth Image Sequences for Data-driven Morphing

Hadar Averbuch-Elor¹ Daniel Cohen-Or¹ Johannes Kopf^{2†}

¹ Tel Aviv University ² Facebook



Figure 1: For a given pair from an image collection (left and right-most) our method automatically finds an *as-smooth-as-possible* sequence of in-between images from the same collection (top row). This can be used to create a *data driven* morph between the two outer images (bottom row).

Abstract

Smoothness is a quality that feels aesthetic and pleasing to the human eye. We present an algorithm for finding “as-smooth-as-possible” sequences in image collections. In contrast to previous work, our method does not assume that the images show a common 3D scene, but instead may depict different object instances with varying deformations, and significant variation in lighting, texture, and color appearance. Our algorithm does not rely on a notion of camera pose, view direction, or 3D representation of an underlying scene, but instead directly optimizes the smoothness of the apparent motion of local point matches among the collection images. We increase the smoothness of our sequences by performing a global similarity transform alignment, as well as localized geometric wobble reduction and appearance stabilization. Our technique gives rise to a new kind of image morphing algorithm, in which the in-between motion is derived in a data-driven manner from a smooth sequence of real images without any user intervention. This new type of morph can go far beyond the ability of traditional techniques. We also demonstrate that our smooth sequences allow exploring large image collections in a stable manner.

1. Introduction

Nowadays, every single day more than *two billion* photos are uploaded to the internet [Kle15]. As a consequence, we are currently experiencing a dramatic growth in the size of searchable digital image repositories. Thousands — sometimes millions — of images are at a user’s finger tips, for every possible query phrase.

Image search results are usually presented in sequential order, for example in the form of a list of thumbnails. In this paper we are interested in the question of what makes a “good” order. Previous work has shown that perception can be improved by arranging results by similarity [RBSW01, SA11], since this reduces the visual load when flipping from one image to the next. In this work, we take this idea one step further by making the observation that an even more pleasing order can be achieved by replacing the pairwise similarity measure with a *higher-order* measure of *smoothness*: in an *as-smooth-as-possible* order, change manifests in a more coherent manner and avoids abrupt change of direction. We present a

new algorithm that obtains *as-smooth-as-possible* sequences from unstructured image collections (see Figure 2).

The problem of computing smooth paths in image collections has been investigated before, however, in previous work, smoothness has been defined in terms of the six degrees-of-freedom motion of a camera in a coherent 3D scene [SGSS08, KSF^{*}12, ABS^{*}13, JKT^{*}15]. The images in our work, however, depict different objects that do not necessarily originate from the same physical model. Our algorithm does not have, or require, a notion of camera pose, view direction, or 3D representation of the underlying scene, but instead directly optimizes the smoothness of the apparent motion of local point matches among the images of the collection. We enrich the search space by including all similarity transformations of input images in the optimization, i.e., uniform scaling, translation, rotation, and reflection. Once having obtained a sequence, we further increase smoothness by reducing slight geometric wobble and appearance variation.

Our smoothness measure only considers point matches along the contours of objects in the images and ignores the appearance in the interior. This design works well with current-generation internet

[†] Part of this work was done while Johannes Kopf was with Microsoft Research



Figure 2: An illustration of our smooth path from Figure 1 in shoe space (embedded in 2D for illustration purposes). The path traverses among similar shoes to form a smooth transition between the source and the target. For clarity, the collection of shoes was diluted in the illustration.

image datasets, which have reached high density in shape space but are still relatively sparse in appearance space.

We demonstrate our smooth sequences in the context of a new *data-driven* image morphing algorithm. Previous techniques rely on ad-hoc motion models, such as linear or quadratic [Wol98], or as-rigid-as-possible [ACOL00] motion, which are oblivious to the real object shape. In contrast, we derive motion paths from the sequence of coherent *real* intermediate images in our smooth sequences. Data-driven morphing can often produce plausible automatic morphs even for very dissimilar input pairs (see Figure 1). As illustrated by this example, it recovers an apparent rotation between the opposing boots, a sequence that would completely fail with conventional morphing sequences without manual user assistance in carefully setting corresponding points and motion paths.

Our novel image morphing algorithm derives the in-between motion in a data-driven manner from our smooth sequences, and thus bypasses the problem of specifying correspondences. Additionally, our presented method includes the following technical contributions:

1. A new measure of smoothness for a tuple of images that considers the apparent motion of local point matches (Section 4);
2. Algorithms for finding smooth sequences in image collections with various constraints (Section 5.1);
3. Algorithms for globally aligning the sequence using similarity transformations (Section 5.2) and for reducing local geometric and appearance variation among the sequence (Section 6).

2. Related Work

Smooth image paths: The ever-growing accessibility to immense image repositories and increasing popularity of geo-tagging has opened opportunities for exploiting the content of these collections. Particularly, planning paths in image collections has received large

interest: Lu et al. [LWY*10] generate travel routes from geo-tagged photos for trip planning; Snavely et al. [SGSS08] use intermediate images to plan paths between an image pair; Kushal et al. [KSF*12] create smooth paths that visit canonical images in the collection. Their work lays the foundation to the Google Maps Photo Tours feature, which automatically creates movies of world famous landmarks from still photos. Most of these works rely on Structure-from-Motion techniques, and generate smooth paths by considering the camera location and orientation in a consistent Euclidean coordinate system representing a real-world scene.

More recently, Arpa et al. [ABS*13] addressed the problem of finding smooth paths in a small collection of images that capture the same event. Their paths are obtained without recovering the scene, but rather by computing the flow between the images. By enabling the users to mark the object of interest, the images along the path are aligned, and a plausible bullet time effect is achieved.

Similarly to the progress made on space-varying collections, techniques that address path planning in time-varying collections are also common. Recent techniques create stable hyperlapse videos that can handle significant high-frequency camera motion. Kopf et al. [KCS14] rely on 3D reconstruction, while Poleg et al. [PHAP14] and Joshi et al. [JKT*15] perform a smart sampling of the long video frame sequence.

Our approach differs from the previously mentioned works, in that we analyze images that depict different instances of inherently deformable objects. This precludes using feature matching approaches, e.g., as common in Structure-from-Motion techniques, since they assume a non-deforming rigid scene structure. Our analysis is entirely shape-based and yields paths that are smooth in shape-space, rather than in the traditional 3D Euclidean space as in previous work.

Image layouts: A commonly employed principle when arranging items is to place similar items nearby. Indeed, it has been shown that similarity-based placement can improve human image retrieval tasks by up to 20% [RBSW01, SA11]. A potential explanation for this phenomenon is that similar images induce less visual perceptual load, and, hence, allow processing the sequence more quickly. [FDH*15] explore various arrangements of image collections that go beyond the traditional grid formations that are common on Flickr, Google, and other web services. In our work, we explore the potential of replacing the ordinary pairwise distance measure with a higher order measure. Our technique is applicable for tiling images on a grid as well as more sophisticated arrangements.

Image morphing: The task of creating a smooth transitional morph between a given pair of images, is a long-standing problem in computer graphics [Wol98, LLN*14]. It is a key component in many computer graphics systems, e.g., for creating movie special effects, and has therefore been extensively researched throughout the last three decades.

One of the main challenges in image morphing lies in defining the motion paths that characterize the trajectories of corresponding points in the two images during the transition. Most traditional morphing algorithms assume simple analytic models, such as linear trajectories [Wol98]. More advanced algorithms maintain local

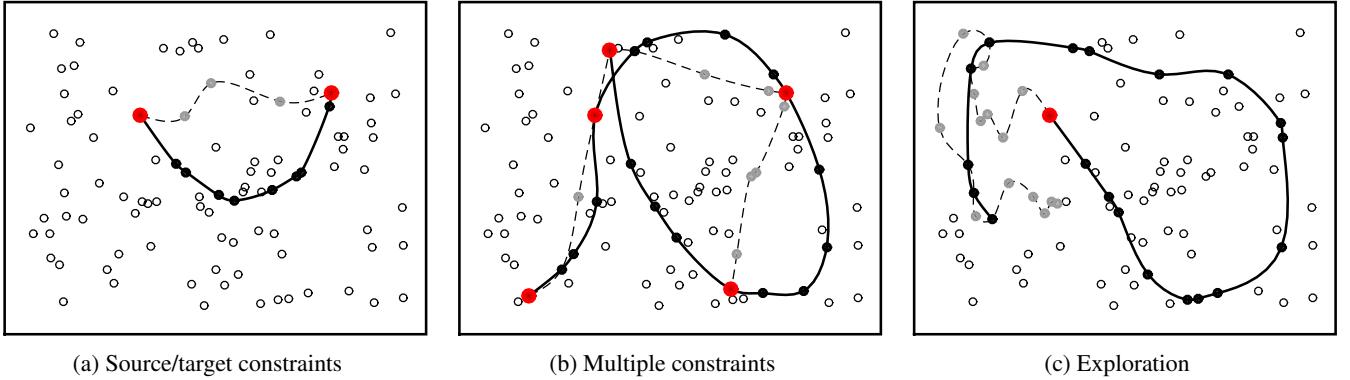


Figure 3: Various setups for constraining smooth sequence construction: (a) start/end constraints; (b) multiple anchors that must be visited, but in arbitrary order; and (c) exploration with an optional start constraint. Our smooth sequences are illustrated with a bold line, with the constrained nodes highlighted in red. The shortest paths are illustrated with a dashed line.

rigidity or area [ACOL00, IMH05]. However, these ad-hoc models for the true motion break down when the input images are too dissimilar (Figures 8, 11). In contrast to previous methods, our motion paths are derived from the real in-between images of our smooth sequences. Furthermore, note that most previous methods rely on the user to assist by marking corresponding features in the images, while in our data-driven method, the correspondence is derived fully automatically.

Gao et al. [GLHH13] employ a database of 3D shapes to facilitate 3D shape morphing. They morph between instances of the same shape under isometric deformations, which significantly simplifies the correspondence problem. In our work we deal with fuzzy, automatically computed correspondences between images. Long-range correspondence emerges transitively. Furthermore, we consider higher-order smoothness and compute as-smooth-as-possible paths, while Gao et al. use a pairwise measure and compute shortest paths.

A different approach to image morphing is to associate and understand relations between dissimilar views of objects that belong to a certain class is to make use of 3D proxies. Rematas et al. [RRFT14] create novel views of images of a known class given a set of 3D models of the class. Likewise, other techniques operate entirely on 3D shapes. For example, Yumer et al. [YCHK15] propose a shape editing technique where geometric deformations are created using a set of semantic attributes. In our work, we do not rely or assume any commonalities among the images and operate directly on the 2D shapes.

3. Overview

The input to our algorithm is a *clean* collection of foreground-segmented images of a common category of objects, e.g., “cats”, “shoes”, or “cars”. Such collections can be created using recent methods [AEWQ*15], however, an even simpler way is to ask Google or Bing image search to retrieve images with a white background by adding the words “white background” to the search query. As the reader can easily convince herself, a surprisingly large number of images for any kind of category can be found this way.

The foreground can be extracted in a trivial manner from these images.

Our first objective is to find visually smooth sequences in this collection (Sections 4–6). The user may constrain the sequences by specifying (optional) start and end images, as well as an arbitrary number of intermediate waypoints (to be visited in any order). Our algorithm then computes an as-smooth-as-possible sequence that respects these user constraints.

We first describe our measure for evaluating the smoothness of a triplet of images in Section 4, and then how to compute user constrained as-smooth-as-possible image sequences using this measure in Section 5. Since any actual image collection is “sparse” (compared to the infinite number of *possible* images), we enrich the search space by including all similarity transformations of images from the input collection, i.e., uniform scaling, translation, rotation, and reflection. We compute the optimal transformations that maximize smoothness of the sequence. As mentioned above, we only deal with correspondences along the contours of objects since current-generation internet datasets are still too sparse in appearance space to consider internal color and texture in the smoothness computation. In Section 6 we describe how to further increase smoothness by reducing remaining geometric and appearance variations through warping and color transfer techniques.

In Section 7 we describe a new data-driven morphing algorithm that derives motion paths from the in-between images in our smooth sequences. We also show other results, applications, and evaluation of our technique.

4. Shape Distance and Smoothness Measures

The first building blocks of our method are measures for determining the visual distance (Section 4.1) and smoothness (Section 4.2) of images. Recall that the input are foreground-segmented shapes; our method is oblivious to the appearance within a shape (color, texture, etc.). Distance and smoothness, therefore, are attributed in our context only to the motion of the shape contours.

4.1. Pairwise Distance Measure

A measure of shape distance is useful for quantifying the amount of change between a pair of shapes, and for computing “shortest” paths through a collection.

We use *Inner-Distance Shape Context* (IDSC) [LJ07] to compute shape distance, as well as establish correspondence between points on the contour. IDSC is an extension of *Shape Context* [BMP02] where Euclidean distances are replaced with inner-distances; this provides some robustness against articulation.

For a given pair of shapes we first sample $M = 100$ evenly distributed points on both contours. From our experiments, this was sufficient to capture most of a shape’s details. IDSC returns a subset of matched contour points (some fraction usually remains without match) as well as an overall matching cost, which provides our measure of distance. Note that in here we only extract the outer contour of the shape, but our method easily generalizes to several closed loops. In the next subsection we will extend this pairwise measure of distance to a higher-order measure of smoothness.

4.2. Triplet Smoothness Measure

In this section we describe how to evaluate the smoothness of a given shape triplet. We define smoothness in terms of the trajectories of corresponding contour points: the triplet is smooth if contour points move (1) without large displacements, and (2) without abrupt changes in direction and velocity. The first property can be evaluated on pairs; the latter is a higher-order measure of change, we can only evaluate it considering at least three shapes at a time. For simplicity we will formulate our smoothness measure in this section for triplets only, and discuss in Appendix A how to generalize it to higher orders.

We start by sampling $M = 100$ evenly distributed points on all three contours, and use IDSC to match the middle shape with the two outer ones. Since IDSC usually only matches a subset of points, we assign correspondences to unmatched points as follows. Let \mathbf{p} be an unmatched point on the first shape and a and b are the arc lengths to the clockwise and counter-clockwise nearest matched points, respectively. We assign the corresponding point \mathbf{p}' on the second shape with arc length distances a' and b' such that $\frac{a'}{b'} = \frac{a}{b}$.

The matching yields a set of point trajectories

$$\{\mathbf{p}_i\}, \quad i = 1, \dots, M. \quad (1)$$

Each trajectory p_i consists of a triplet of corresponding 2D points

$$\mathbf{p}_i = (\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \mathbf{p}_{i,3}), \quad (2)$$

one residing on each shape, respectively.

We define the smoothness of a point trajectory as an energy function, consisting of a first-order term E_{disp} and second-order term E_{smooth} ,

$$E(\mathbf{p}_i) = E_{disp}(\mathbf{p}_i) + \lambda E_{smooth}(\mathbf{p}_i), \quad (3)$$

where λ is a prescribed parameter that balances their relative contributions (empirically set to 4).

The first-order term,

$$E_{disp}(\mathbf{p}_i) = \|\mathbf{p}_{i,1} - \mathbf{p}_{i,2}\|^2 + \|\mathbf{p}_{i,2} - \mathbf{p}_{i,3}\|^2, \quad (4)$$

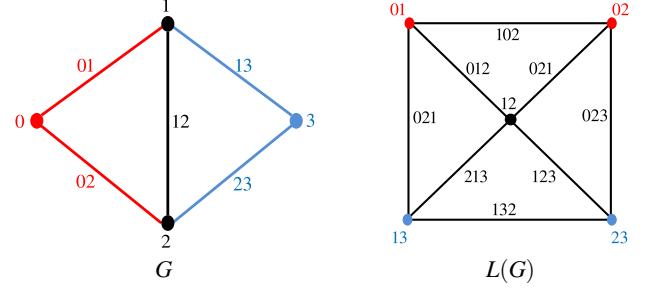


Figure 4: The graph G and its linear dual graph $L(G)$. Each vertex in $L(G)$ represents an edge in G , and each edge in $L(G)$ represents a pair of consecutive edges in G . In our implementation, the edges in G connect only similar pairs of images. See Section 5.1 for an explanation of the coloring.

penalizes large displacement between successive shapes, while the second-order term,

$$E_{smooth}(\mathbf{p}_i) = \|\mathbf{p}_{i,1} - 2\mathbf{p}_{i,2} + \mathbf{p}_{i,3}\|^2, \quad (5)$$

penalizes changes in direction and velocity.

Since the shapes are usually not well aligned with respect to each other we optimize similarity transformations to minimize the energy function. We define a translation vector \mathbf{t}_k and a scale-rotation matrix $\mathbf{R}_k = \begin{pmatrix} a_k & -b_k \\ b_k & a_k \end{pmatrix}$ for each of the three shapes. The transformed points are given by

$$\tilde{\mathbf{p}}_{i,k} = \mathbf{R}_k \mathbf{p}_{i,k} + \mathbf{t}_k. \quad (6)$$

Note, that \mathbf{t}_k and \mathbf{R}_k are *shared* for all points of the k -th shape, and, therefore, characterize a similarity transformation of the shape. To find the optimal transformation for a given triplet we optimize the following objective:

$$\mathbf{E} = \min_{\{a_k, b_k, \mathbf{t}_k \mid k \in \{1, 2, 3\}\}} \sum_{i=1}^M E(\tilde{\mathbf{p}}_i), \quad (7)$$

where $\tilde{\mathbf{p}}_i = (\tilde{\mathbf{p}}_{i,1}, \tilde{\mathbf{p}}_{i,2}, \tilde{\mathbf{p}}_{i,3})$ is the transformed point trajectory. Note, that we set $R_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\mathbf{t}_2 = \mathbf{0}$, and only optimize the outer shapes’ transformations to avoid the trivial solution that would collapse all shapes. Since all terms in this energy are quadratic, the optimal solution can be obtained by solving a sparse linear system.

5. Computing Smooth Image Sequences

With the similarity and smoothness measures defined in the previous section, we are now ready to describe our algorithm for computing as-smooth-as-possible paths through image collections. We first find a sequence of images using standard shortest-path algorithms in an appropriately constructed graph (Section 5.1), and then compute the optimal alignment using similarity transformations (Section 5.2).

5.1. Optimizing the Image Sequence

We start by constructing a graph G with a node for each image in the collection and edges representing possible transitions be-

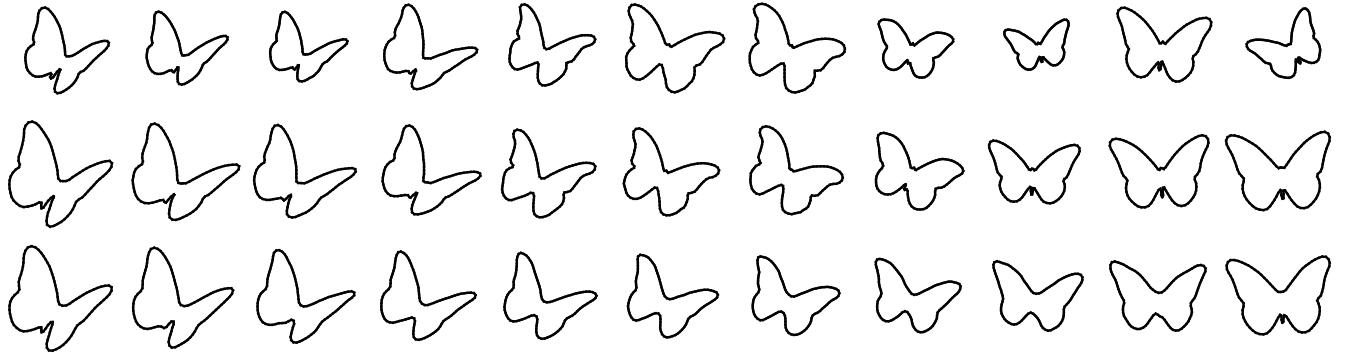


Figure 5: As-smooth-as-possible shape sequences. Equipped with the smooth sequence from Section 5.1 (top row), we optimize a similarity transformation per shape to globally align the sequence (middle row). Then, we reduce the residual geometric wobbling of the shapes through non-rigid deformations.

tween images. The edges are weighted by the pairwise shape distance (Section 4.1). To reduce complexity we consider only the 10 nearest neighbors of each image when constructing the graph. Shortest paths in this graph correspond to as-similar-as-possible image sequences.

We cannot represent the smoothness relationship in G because it involves triplets of images. Hence we consider the *linear dual graph* $L(G)$, also called the *line* or *edge graph*, which is constructed as follows:

1. every edge in G is represented by a node in $L(G)$, and
2. every pair of *adjacent* edges in G is represented by an edge in $L(G)$.

In other words, each edge in $L(G)$ stands for a triplet of images, and we weigh it using the smoothness measure defined in Eq. 7. Shortest paths in $L(G)$ correspond to as-smooth-as-possible image sequences.

Note, that linear dual graphs have been commonly used for modeling turn costs in the route planning literature [Win02, Cal61]. The “angled graph” described by Arpa et al. [ABS*13] is a similar construction.

In the following we describe computing optimal smoothest paths with different types of constraints (Figure 3):

Optimal smooth sequence between two images: The most basic type are source and target image constraints (Figure 3a). We start by identifying the corresponding nodes in G and their connected source and target edges. These edges correspond to all possible pairs at the beginning and end of the smoothest sequence. For example, the source and target nodes 0 and 3, and their connected source/target edges are colored in red/blue in Figure 4a, respectively.

The source/target edges in G correspond to nodes in $L(G)$ (colored in Figure 4b). The shortest path between any representatives from the sets of source-edge-nodes and target-edge-nodes corresponds to the optimal smoothest image sequence. It can be computed using Dijkstra’s shortest path algorithm [CSRL01].

Optimal smooth sequence with multiple constraints: We can also compute the smoothest sequences that are constrained by $k > 2$ “anchor images” that can be visited in arbitrary order (Figure 3b). This algorithm is inspired by Kushal et al. [KSF*12], who compute smooth tours through “canonical” views among 3D reconstructed image collections.

We start by constructing a fully connected graph G^A where the nodes represent the anchor images and the edges are weighted by the cost of the smoothest sequence between their nodes (computed with the technique above). The optimal anchor ordering can be computed by solving the traveling salesman problem on G^A [CSRL01]. Appending the individual smoothest sequence between the anchors yields an *almost* good solution that would be smooth everywhere except across the anchors, where it is not guaranteed to be smooth. Therefore, we instead obtain the smooth sequence from the optimal anchor ordering using the algorithm described in Section 3.3 of Kushal et al.’s paper [KSF*12]. Please refer to their paper for more details.

Exploration sequences: A smooth exploration sequence freely “roams” the collection; thus, there are no target constraints, but optionally a source constraint (Figure 3c). A greedy approach that advances the sequence towards the direction of the smoothest triplet might get trapped by a local minimum; it might venture into a part of the graph where it can only get out using a non-smooth transition. To avoid this problem we employ a beam search methodology [Bis87], and examine a small number of alternatives in parallel. At every step, the beams advance towards the smoothest solutions. In our implementation, we empirically set the beam width to 50.

5.2. Globally Aligning the Sequence

Having found the sequence of N images (Figure 5, top row), our next goal is to bring them into good alignment (Figure 5, middle row). While each of the $N-2$ overlapping triplets comes with a set of optimal transformation parameters (computed using Eq. 7), these are inconsistent because they are computed independently. We can, however, compute *globally consistent* alignment parame-

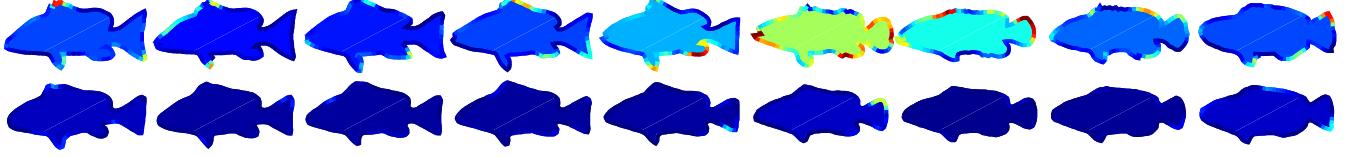


Figure 6: Removing residual geometric wobble. A sequence from our FISH dataset before (top row) and after (bottom row) the geometric wobble removal stage. The colors on the silhouettes visualize local temporal smoothness, as computed by Equation 5. The color within the shapes illustrate the average temporal smoothness. This example demonstrates how higher temporal smoothness is achieved at the expense of high-frequency details.

ters by solving a similar problem that takes all overlapping triplets into account.

Each of the $N-2$ overlapping triplets contributes M point trajectories:

$$\{\mathbf{p}_i^t\}, \quad i \in [1, M], \quad t \in [1, N-2]. \quad (8)$$

As in Section 4.2 we define rigid transformation parameters for each shape, a translation vector \mathbf{t}_t and rotation matrix $\mathbf{R}_t = \begin{pmatrix} a_t & -b_t \\ b_t & a_t \end{pmatrix}$, and obtain the transformed trajectories $\tilde{\mathbf{p}}_i^t = \mathbf{R}_t \mathbf{p}_i^t + \mathbf{t}_t$.

We solve a similar problem to Eq. 7 to obtain the optimal transformations, however, instead of fixing one of the shapes to the identity transformation we *softly* constrain every shape using a pull-back term that encourages a solution where no shape deviates too strongly from its initial configuration:

$$\underset{\{a_t, b_t, \mathbf{t}_t\}}{\operatorname{argmin}} \sum_{t=1}^{N-2} \sum_{i=1}^M \left(E(\tilde{\mathbf{p}}_i^t) + \gamma \sum_{k=1}^3 \|\tilde{\mathbf{p}}_{i,k}^t - \mathbf{p}_{i,k}^t\|^2 \right). \quad (9)$$

The coefficient $\gamma = 1$ controls the strength of the pullback term and has been empirically determined. The optimal solution to this problem is obtained by solving a sparse linear system of equations.

6. Reducing Geometric and Appearance Variation

Since the images in our collections and the computed smooth sequences usually depict different objects with different 3D shapes, under different lighting conditions, and with different colors, the global similarity transform alignment described in the previous section is usually not sufficient to avoid temporal flickering when transitioning along the sequence. Thus, to alleviate the problem, we reduce residual geometric wobbling of the shapes through non-rigid deformations (Section 6.1) and decrease the appearance variation through color transformation (Section 6.2).

6.1. Residual Geometric Wobble

We reduce residual geometric wobble of each shape by considering its neighbors along the sequence. We apply a low-pass filter to the trajectories of the contour samples. As the amount of variation can be considerable, we would ideally use a wide smoothing kernel that averages every shape with several neighbors before and after in the sequence. However, since correspondence between non-consecutive shapes has proven very unreliable, we instead approximate this by *iteratively* smoothing with a narrow kernel (10

iterations with a 3-tap 1:2:1 kernel). After each iteration we resample the M points on each contour and recompute correspondence, as described in Section 4. The bottom rows in Figures 5 and 6 illustrate the effect of reducing geometric wobble in the shape contours. As can be observed, the temporal smoothness of the shapes is increased at the expense of the high-frequency details.

After obtaining the smoothed contour we warp the interior texture using the technique of Schaefer et al. [SMW06].

6.2. Appearance stabilization

We reduce appearance variation by propagating the color space footprints of the first and last to the inner images in the sequence. To analyze and manipulate color differences, we need dense correspondence throughout the interior of the images. However, in the previous sections we have only established sparse correspondence along the shape silhouettes. Hence, we obtain dense interior correspondence from the sparse constraints using Poisson interpolation [PGB03].

Given a sequence we want to color-stabilize, we fix the appearance of the first and last images, and transfer the colors from both ends separately to each inner image (using the algorithm below). The final appearance of the inner images is obtained by blending the two color transfer results.

Suppose we want to transfer the colors from a source image (either the start or end of the sequence) to a target image (one of the inner ones). We use a technique inspired by Shih et al. [SPDF13], who compute locally affine RGB transformations to perform dramatic color changes. In our setting we found it sufficient to use simpler globally affine models, however.

We represent images in CIELAB color space, and denote by \mathbf{v}_k^s and \mathbf{v}_k^t the (a,b) chroma channels of the k -th corresponding pixel in the source and target images. $\bar{\mathbf{v}}$ denotes the homogeneous (one-augmented) version of the chroma vector \mathbf{v} . We first find the best *affine* transformation between the color space footprints by solving

$$\min_{\mathbf{A}} \sum_k \|\mathbf{v}_k^s - \mathbf{A} \bar{\mathbf{v}}_k^t\|^2, \quad (10)$$

where \mathbf{A} is a 2×3 matrix. The solution is obtained by solving a sparse linear system of equations. Colors are transferred by applying \mathbf{A} to every pixel in the target image. In addition, the lightness variations are reduced by matching the mean and variance in the source/target L channels.



Figure 7: **Appearance stabilization.** Image sequences from our CAP and CAR datasets before (1^{st} and 3^{rd} rows) and after (2^{nd} and 4^{th} rows) the appearance stabilization step.

7. Results and Applications

We evaluated our technique on various datasets containing diverse categories that span a wide range of objects. The HORSE (810 images) and CAR (1208 images) datasets were taken from the supplementary material of Rubinstein et al.’s paper [RJKL13]. The LEAF dataset (1085 images) is provided by Kumar et al. [KBB*12]. It consists of a large database of real leaves that were photographed using mobile devices. Each image is foreground-segmented and annotated with species identification estimated by their technique. We generated four new datasets, BOOT, BUTTERFLY, CAP and FISH (each about 200 images) using Google Image Search. We augmented the queries with the words “white background” and enabled retrieving images with transparent background in the search tools.

7.1. Data-driven Morphing

One application of our smooth image sequences is a novel data-driven morphing algorithm. The goal of image morphing is to create a seamless transition between two input images, by creating a sequence of gradually changing in-between images that appear natural and free of ghosting or other artifacts. The in-between images are created by warping both input images toward each other so they become aligned, and then blending the two results.

Traditional morphing techniques rely on a dense correspondence map between the input image pair, which is usually obtained with heavy manual assistance. The correspondence map defines where every point in one image should move to in the other image, however, it does not define the spatial path along which it should move. Most techniques assume simple analytical paths, such as linear motion. Recent and more involved techniques [LLN*14] generally exhibit better performance, but they still cannot automatically deal with dissimilar shapes or large changes in viewpoint. It worth noting that often even manual correspondence cannot help, as demonstrated in Figure 8.

The first step of our new algorithm makes use of the techniques described in Sections 4–6 to obtain a smooth image sequence that starts and ends with the morph input pair. We use the sequence to establish correspondence between the original input pair *transitively* by chaining up correspondence between consecutive images of the sequence. Since these images are much closer than the original pair the operation is more likely to perform successfully. Furthermore, this leads to a data-driven motion path, that is based on

real in-between images. This often yields more natural appearing motion. For example, consider the apparent 3D rotation of the boots in Figure 1; such motion cannot be achieved with analytical motion models.

The texture of the in-between images can also possibly contribute to the morphing result. We can choose to use all, or some, of the in-between textures and generate a so-called *N-step* morph. Alternatively, we can use none of the in-between textures to generate a *direct* morph. Please refer to Figures 1, 10, 11 and the accompanying video for results. In what follows, we elaborate more on the details of the morph generation.

For two consecutive images of the smooth sequence we interpolate in-between images as follows. From the contour correspondence obtained in Section 4.1, we generate a constrained Delaunay triangulation, where the shape contours serve as the constraints. Since consecutive images along the sequence are similar, a simple triangulation-based linear morph usually works well. We proceed in this manner for every consecutive image pair in the sequence and concatenate the results to obtain an N-step transition that smoothly morphs through all images of the image sequence.

To blend only the appearance of the source and target images, i.e., without showing the other images from the sequence during the transition, we propagate the textures of the original input pair throughout the sequence by chaining up the correspondence maps between consecutive sequence images. Then, we create a data-driven morph for this new sequence as described above.

In a similar manner, we can also create long transitions that morph through one or more middle images. We can either choose any middle image from the sequence, or we can specify them a priori and generate the sequence with the multi-constraint algorithm from Section 5.1.

We compared our data-driven morphing results to the state-of-the-art morphing technique of Liao et al. [LLN*14]. As we are morphing between two distant objects and the corresponding points are far from trivial to compute automatically, we also generated a standard morph with manual correspondences (see the sequence in the middle row of Figure 8). In spite of the user assistance, the results are of rather poor quality. Other morphing sequences are illustrated in Figure 11, where the correspondences along the silhouettes were computed with IDSC without any manual assistance. We also considered other morphing techniques, such as standard linear



Figure 8: Applying state-of-the-art morphing techniques [LLN*14] to the boots from Figure 1. In the top row, the morph is generated automatically, without any given correspondences. In the middle row, a few semantic correspondences are provided manually. In the bottom row, the correspondences along the silhouettes are computed automatically using the IDSC technique, yielding a semantically erroneous morph.

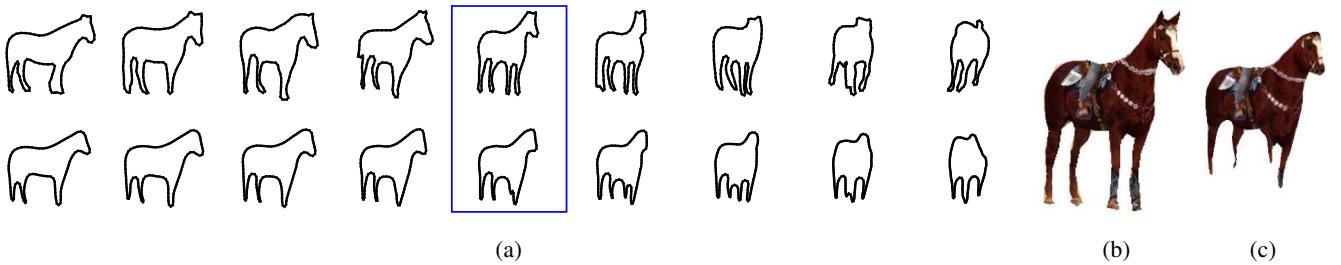


Figure 9: Failure case of the wobble reduction technique. (a) When consecutive images along the sequence are quite dissimilar and exhibit different topologies (top row), the residual wobble reduction procedure (described in Section 6.1) may lead to unnatural looking shapes due to the erroneous correspondences (bottom row). (b,c) The middle image (enclosed in blue) before/after wobble reduction.

morphs, whose performance is generally lower but comparable to Liao et al. We also examined the view-morphing technique [SD96], which use projective geometry to handle view changes of objects. However, for images that capture different objects, their method is only applicable as an interactive post-process step to the morphing technique. Since parts of the sequence are completely broken into pieces, a post process warp is not particularly beneficial and therefore, we only display the regular morphing sequences.

7.2. Collection exploration

Our ordering technique also facilitates a novel collection exploration application. Rather than browsing through the collection thumbnails, the users can automatically generate a collection flip book, where the smooth sequence can be displayed in a loop. Here, we can determine the length of the sequence and an optional starting point, and the algorithm described in Section 5 generates a smooth traversal of the specified length. See the attached video for a demo of our exploration application on the LEAF dataset.

7.3. Limitations

A notable limitation of our technique is that it is entirely shape-based and does not take the internal appearance into account. While this is in many cases sufficient and generates plausible results, it is

not always the case. In particular, this causes poor direct morphing results for the CAR dataset. The distortion of the internal structure (front-lights, car doors, etc.) overshadows the smooth change of the shape.

Another limitation lies in the premise of our method, that there is a sufficiently dense set of relevant images. As a data driven technique, the performance is bounded by the availability of the data. This is particularly noticeable when consecutive images along our sequence are not similar enough and highly non-convex. In these cases, the wobble removal stage may lead to unnatural looking shapes as demonstrated by the HORSE sequence in Figure 9.

8. Conclusions and Future Work

Smooth curves, transitions, and camera paths are aesthetic and eye-pleasing. In this paper we have introduced a new technique for extracting “as-smooth-as-possible” sequences from a collection of heterogeneous images that do not necessarily represent the same instance, object or scene. On top of this we developed a new *data-driven* morphing technique, which can create smooth transitions even between images that are significantly different.

A notable challenge in traditional image morphing techniques lies in the need to specify correspondence between the source and target images. Since the correspondence should reflect the seman-

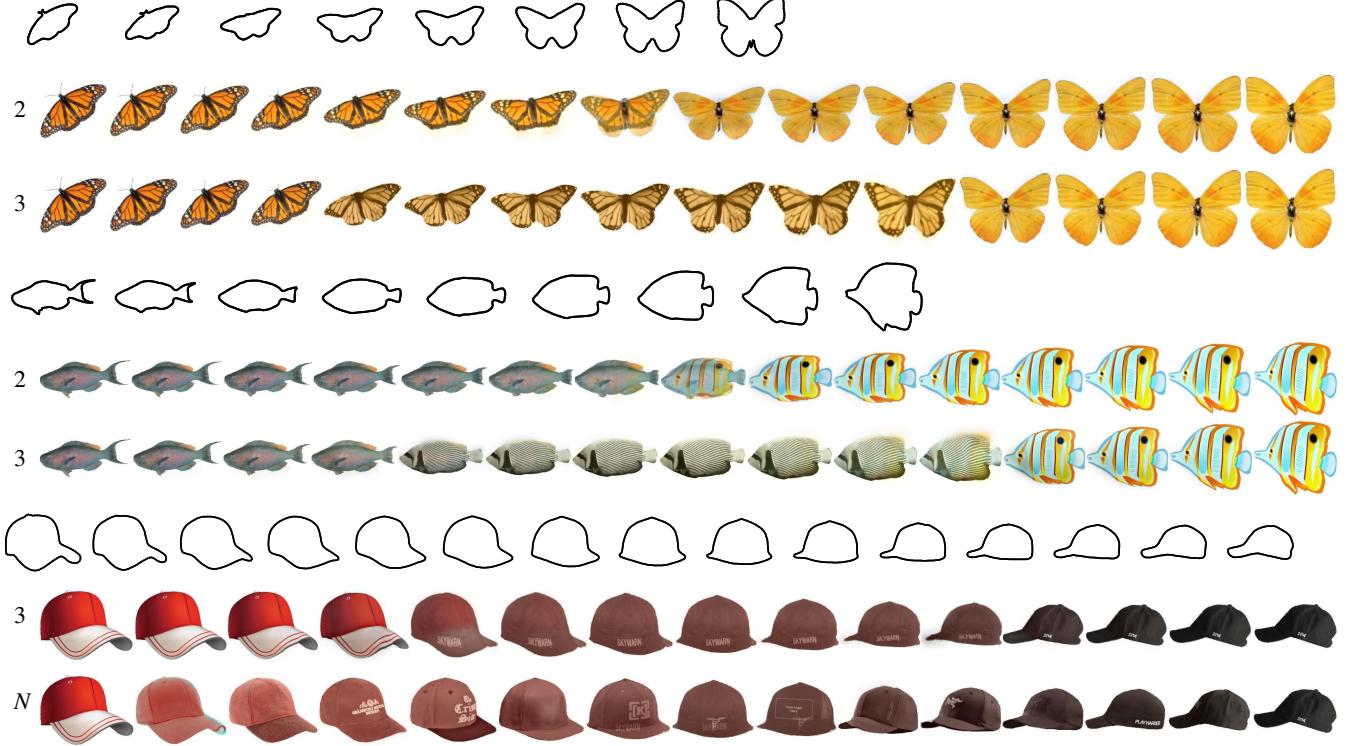


Figure 10: Given a smooth shape sequence of length N (rows 1, 4, 7), we can create a textured morph using all, some or none of the in-between images (remaining rows). The number of textures that contributed to each morph is displayed on the left.

ties of the shapes parts, it is commonly left to the artist. Our data-driven morphing technique bypasses the big hurdle of establishing correspondence between very dissimilar images by reducing the problem to finding many simpler correspondence maps among more similar images. As we have shown, our automatic method can often produce good results.

We believe that dense collections of images or shapes have more potential. The similarity measures and hence the rather simple correspondence among them can lead to more applications which are traditionally hard by avoiding semantic analysis. In the future we would like to extend the technique and lift it to higher dimensions, considering sequencing among 3D shapes. Another avenue is creating smooth sequencing based on colors or textures, rather than just the shape contours.

Appendix A: Generalizing the smoothness measure to higher orders

Smoothness has been formulated in Section 4.2 in terms of triplets. The formulation can be generalized to higher orders, where point trajectories include N shapes.

$$\tilde{\mathbf{p}}_i = (\tilde{\mathbf{p}}_{i,1}, \tilde{\mathbf{p}}_{i,2}, \tilde{\mathbf{p}}_{i,3}, \dots, \tilde{\mathbf{p}}_{i,N}). \quad (11)$$

Let $\Delta_{i,j} = \tilde{\mathbf{p}}_{i,j} - \tilde{\mathbf{p}}_{i,j+1}$ be the difference vector between two points on consecutive shapes, and $\Delta_{i,j}^k = \Delta_{i,j}^{k-1} - \Delta_{i,j+1}^{k-1}$ be the k -th order difference (with $\Delta^1 = \Delta$).

We can then define the k -th order energy term

$$E_k(\tilde{\mathbf{p}}_i) = \sum_{j=1}^{N-k} \left\| \Delta_{i,j}^k \right\|^2, \quad (12)$$

and the combined energy

$$E(\tilde{\mathbf{p}}_i) = \sum_{k=1}^{N-1} \lambda_k E_k(\tilde{\mathbf{p}}_i). \quad (13)$$

It is easy to see that for shape triplets $E_1 = E_{dist}$ and $E_2 = E_{smooth}$, and Eq. 13 is equivalent to Eq. 3.

There are two issues with using higher-order smoothness in practice. First, the search space grows exponentially with N , and it quickly becomes infeasible to perform the computations as the collection size grows. Second, the quality of point trajectories degrades rapidly beyond triplets due to error accumulation. For these reasons, all the results and evaluations were obtained with the smoothness measure on shape triplets.

Acknowledgements

We thank Pieter Peers for providing helpful discussions in an early stage of this project. We also thank Ronit Reitsstein for helping in assembling the datasets. This work supported by the Israel Science Foundation.

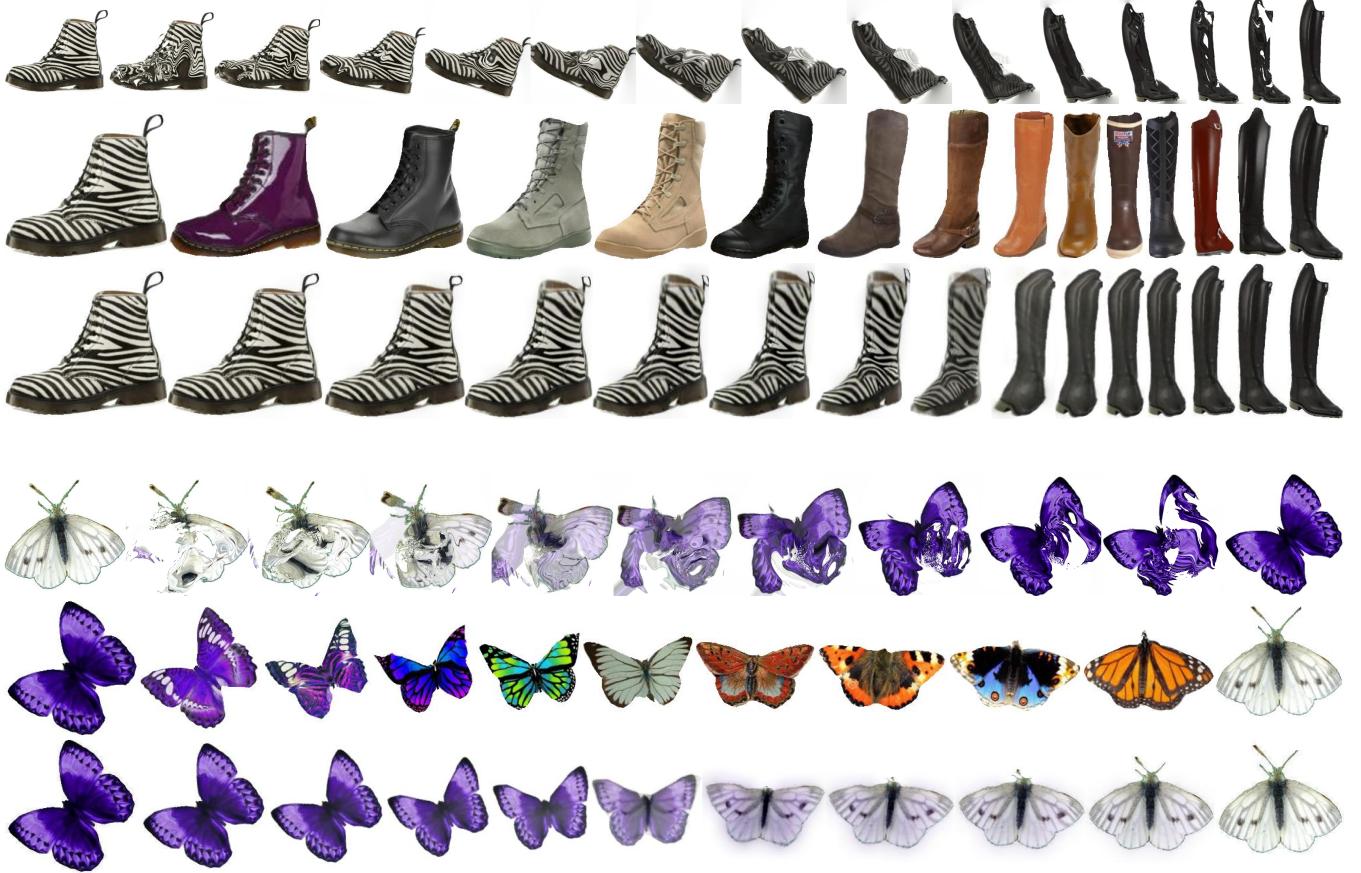


Figure 11: Data-driven morphing vs. traditional morphing techniques. Given start and end constraints, the top rows illustrate the sequence obtained by the technique of Liao et al. [LLN*14]. The middle rows depict the smoothed sequence images. The bottom rows illustrate our data-driven morphs. Note that all results were obtained without manual intervention, where the correspondences were computed with IDSC.

References

- [ABS*13] ARPA A., BALLAN L., SUKTHANKAR R., TAUBIN G., POLLEFEYS M., RASKAR R.: Crowdcam: Instantaneous navigation of crowd images using angled graph. In *3D Vision-3DV 2013, 2013 International Conference on* (2013), IEEE, pp. 422–429. [1](#), [2](#), [3](#)
- [ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 157–164. [2](#), [3](#)
- [AEWQ*15] AVERBUCH-ELOR H., WANG Y., QIAN Y., GONG M., KOPF J., ZHANG H., COHEN-OR D.: Distilled collections from textual image queries. *Computer Graphics Forum, (Proceedings Eurographics 2015) 34*, 2 (2015), to appear. [3](#)
- [Bis87] BISIANI R.: Beam search. *Encyclopedia of Artificial Intelligence* 2 (1987). [5](#)
- [BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 4 (2002), 509–522. [4](#)
- [Cal61] CALDWELL T.: On finding minimum routes in a network with turn penalties. *Commun. ACM* 4, 2 (1961), 107–108. [5](#)
- [CSRL01] CORMEN T. H., STEIN C., RIVEST R. L., LEISERSON C. E.: *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001. [5](#)
- [FDH*15] FRIED O., DIVERDI S., HALBER M., SIZIKOVA E., FINKELSTEIN A.: IsoMatch: Creating informative grid layouts. *Computer Graphics Forum (Proc. Eurographics)* 34, 2 (May 2015). [2](#)
- [GLHH13] GAO L., LAI Y.-K., HUANG Q.-X., HU S.-M.: A data-driven approach to realistic shape morphing. In *Computer graphics forum* (2013), vol. 32, Wiley Online Library, pp. 449–457. [3](#)
- [IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* 24, 3 (2005), 1134–1141. [3](#)
- [JKT*15] JOSHI N., KIENZLE W., TOELLE M., UYTENDAELE M., COHEN M. F.: Real-time hyperlapse creation via optimal frame selection. *ACM Trans. Graph.* 34, 4 (2015), article no. 63. [1](#), [2](#)
- [KBB*12] KUMAR N., BELHUMEUR P. N., BISWAS A., JACOBS D. W., KRESS W. J., LOPEZ I., SOARES J. V. B.: Leafsnap: A computer vision system for automatic plant species identification. In *The 12th European Conference on Computer Vision (ECCV)* (October 2012). [7](#)
- [KCS14] KOPF J., COHEN M. F., SZELISKI R.: First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 78. [2](#)
- [Kle15] KLEINER PERKINS CAUFIELD & BYERS: *Internet Trends 2015*. Tech. rep., 2015. [1](#)
- [KSF*12] KUSHAL A., SELF B., FURUKAWA Y., GALLUP D., HER-NANDEZ C., CURLESS B., SEITZ S. M.: Photo tours. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*,

2012 Second International Conference on (2012), IEEE, pp. 57–64. [1](#), [2](#), [5](#)

[LJ07] LING H., JACOBS D. W.: Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 2 (2007), 286–299. [4](#)

[LLN*14] LIAO J., LIMA R. S., NEHAB D., HOPPE H., SANDER P. V., YU J.: Automating image morphing using structural similarity on a halfway domain. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 168. [2](#), [7](#), [8](#), [10](#)

[LWY*10] LU X., WANG C., YANG J.-M., PANG Y., ZHANG L.: Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the international conference on Multimedia* (2010), ACM, pp. 143–152. [2](#)

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH 2003)* 22, 3 (2003), 313–318. [6](#)

[PHAP14] POLEG Y., HALPERIN T., ARORA C., PELEG S.: Egosampling: Fast-forward and stereo for egocentric videos. *arXiv preprint arXiv:1412.3596* (2014). [2](#)

[RBSW01] RODDEN K., BASALAJ W., SINCLAIR D., WOOD K.: Does organisation by similarity assist image browsing? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2001), 190–197. [1](#), [2](#)

[RJKL13] RUBINSTEIN M., JOULIN A., KOPF J., LIU C.: Unsupervised joint object discovery and segmentation in internet images. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (June 2013). [7](#)

[RRFT14] REMATAS K., RITSCHEL T., FRITZ M., TUYTELAARS T.: Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014). [3](#)

[SA11] SCHOEFFMANN K., AHLSTRÖM D.: Similarity-based visualization for image browsing revisited. *Proceedings of the Seventh IEEE International Workshop on Multimedia Information Processing and Retrieval (MIPR 2011)* (2011), 422–427. [1](#), [2](#)

[SD96] SEITZ S., DYER C.: View morphing, 1996. In *Siggraph 1996 Conference Proceedings, Annual Conference Series* (1996), pp. 21–30. [8](#)

[SGSS08] SNAVELY N., GARG R., SEITZ S. M., SZELISKI R.: Finding paths through the world’s photos. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 15. [1](#), [2](#)

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)* 25, 3 (July 2006), 533–540. [6](#)

[SPDF13] SHIH Y., PARIS S., DURAND F., FREEMAN W. T.: Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 200. [6](#)

[Win02] WINTER S.: Modeling costs of turns in route planning. *Geoinformatica* 6, 4 (2002), 345–361. [5](#)

[Wol98] WOLBERG G.: Image morphing: a survey. *The Visual Computer* 14, 8/9 (1998), 360–372. [2](#)

[YCHK15] YUMER M. E., CHAUDHURI S., HODGINS J. K., KARA L. B.: Semantic shape editing using deformation handles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015)* 34 (2015). [3](#)