EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

A Data-Driven Approach to Realistic Shape Morphing

Lin Gao[†1], Yu-Kun Lai[2], Qixing Huang[3] and Shi-Min Hu[1]

[1] TNList, Department of Computer Science and Technology, Tsinghua University, China
[2] School of Computer Science and Informatics, Cardiff University, UK
[3] Computer Science Department, Stanford University, USA

Abstract

Morphing between 3D objects is a fundamental technique in computer graphics. Traditional methods of shape morphing focus on establishing meaningful correspondences and finding smooth interpolation between shapes. Such methods however only take geometric information as input and thus cannot in general avoid producing unnatural interpolation, in particular for large-scale deformations. This paper proposes a novel data-driven approach for shape morphing. Given a database with various models belonging to the same category, we treat them as data samples in the plausible deformation space. These models are then clustered to form local shape spaces of plausible deformations. We use a simple metric to reasonably represent the closeness between pairs of models. Given source and target models, the morphing problem is casted as a global optimization problem of finding a minimal distance path within the local shape spaces connecting these models. Under the guidance of intermediate models in the path, an extended as-rigid-as-possible interpolation is used to produce the final morphing. By exploiting the knowledge of plausible models, our approach produces realistic morphing for challenging cases as demonstrated by various examples in the paper.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—object representations

1. Introduction

Shape morphing is a fundamental technique widely used in computer graphics. Typical applications include computer animation where a smooth sequence of intermediate models are created between keyframe models, shape blending where new objects are produced with contributions of two or more objects, and deforming a template shape to fit objects [SMB12]. The given models can be the same object with different poses or even completely different objects. While it is possible to use skeleton-based interpolation for the former case, it would be most natural to use direct shape morphing techniques in the latter case, which also has the benefit of not requiring skeletons to be constructed and attached to models.

This work considers on direct shape morphing. As an active research problem, a lot of effort has been made in the computer graphics community. Previous work has mainly focused on two aspects, i.e., finding meaningful correspondences between given models and producing smooth interpolated shapes for natural morphing. Since only the source and target geometric models are available, even if the one-to-one correspondence is well established, the interpolated sequence may not look natural for challenging cases, such as models with large-scale deformations. Human body/animal movements are often difficult, partly because of the complicated potential poses and partly because they are well perceived and the expectation of naturalness is high, a psychological phenomenon called “Uncanny valley” where the comfort level of perception drops when human robots or animations look almost but not perfectly like human beings. Examples of imperfection using traditional methods include self-intersection in space, unnecessary distortions, and unnatural intermediate poses which might be difficult or even impossible to achieve in reality.

Due to the availability of large 3D model databases, it is possible to exploit the knowledge of model plausibility to improve shape morphing; this is exactly the motivation of

† Emaill: gaolinorange@gmail.com

this paper. Given a pair of source and target models and a database of similar models, we seek to compute a smooth path of intermediate shapes using the knowledge extracted from the input database. As establishing correspondence is not our focus and is already well studied, we use existing approaches to generate a common embedding between the given models. The main contributions of our work are as follows:

- We analyze the models in the database to form locally linear shape spaces which can efficiently represent plausible models in the category.
- We formulate the problem of natural morphing as finding a shortest path connecting elements in the *continuous local shape spaces*, ensuring the plausibility of in-between reference models.
- We propose a simple interpolation algorithm for efficient as-rigid-as-possible shape interpolation guided by the reference models to produce realistic morphing.

2. Related work

Shape morphing has been an active research direction for a couple of decades. Given the source and target models often represented as triangle meshes, the first step for realistic morphing is to establish one-to-one correspondence between two shapes. This also applies when multiple shapes are to be blended. This is often achieved by animators specifying a sparse set of correspondences which are then used as constraints to establish one-to-one correspondence as the source and target meshes can be very different. Various techniques have been proposed, including parameterizing over coherent base domains [LDSS99, PSS01, SAPH04], using common embeddings [LH03, LL05] or using some iterative alignment algorithm to incrementally align the surface patches [SP04, HAWG08].

Even with reliable correspondences, natural shape morphing is not trivial. If the change between the shapes is small, simple linear interpolation of the vertex position is probably sufficient. However, direct positional interpolation may cause significant artifacts when the shapes being interpolated exhibit large deformations. A well-known approach [ACOL00] generates morphing sequences that are as rigid as possible. The basic idea is to factor local affine transform matrix into a single rotation (rigid) matrix and a symmetric (elastic) matrix and linear interpolation is then applied. This method however requires *consistent* tetrahedron meshes as input for 3D mesh morphing. Huang *et al.* [HAWG08] use a similar formulation that is locally as rigid as possible and minimizes the overall movement of vertices. The algorithm takes surface meshes as input; however the number of unknowns to be solved scales linearly with the number of in-between frames and is thus expensive if the number of frames becomes moderately large.

Alternative approaches consider interpolation in some feature spaces that better preserve the geometry. Alexa [Ale03] uses local differential coordinates (Laplacian coordinates) instead of absolute positions for shape interpolation. This improves detail preservation but the differential coordinates are still rotation sensitive. To further improve the results, Hu *et al.* [HLW07] consider interpolation of mean Laplacian flow in the dual Laplacian domain [KCATLF06]. Xu *et al.* [XZWB06] interpolate the gradient fields and use a Poisson-based approach [YZX*04] to fuse the intermediate meshes. Chu and Lee [CL09] interpolate the shape in the gradient space of near-rigid components and the method is particularly suitable for pose interpolation. Lipman *et al.* [LSLCO05] propose a rotation-invariant representation of mesh models based on discrete forms and similar as-rigid-as-possible morphing can be obtained by linear interpolation of the discrete form coefficients. Killian *et al.* [KMP07] treat shapes as points on the Riemannian surfaces derived from typical rigid or isometric transformations, and convert the plausible morphing problem to finding a shortest path in the shape space which can be solved using a variational approach. To handle large models and large deformations, a multi-scale approach is also proposed based on a mesh hierarchy [WDAH10].

All the approaches described above assume that the underlying interpolation smoothly changes the shape and preserves the local geometric features as much as possible. However, these methods cannot avoid self-intersection in space, unnatural in-between shapes that violate the physical laws or are unlikely to appear in practice. Borrowing ideas from continuum mechanics, physics-based approaches have also been proposed which minimize strain energy between adjacent frames [HLZ04] or linearly interpolate the strain field and use this to recover the shape [YHM07]. While following physical laws, these methods are still restrictive because a substantially simplified model is used due to the limited knowledge of the material property and the consideration for computational efficiency. Because of these restrictions, unnatural interpolation still exists.

Previous work has used example-based approaches to improve the naturalness. Skeleton models have been used in [GMHP04, WSLG07] to achieve good results thanks to the essential low dimension of the deformation space. Further work produces results directly on meshes using inverse kinematics derived from example models [SZGP05, FB11]. These methods however are mainly suitable for pose interpolation. Our work uses a substantially larger database and is capable of handling general shape morphing. Sloan *et al.* [SRC01] produce new shapes by interpolation of example shapes. The interpolation is controlled by an abstract space specified by the designer in an offline stage. Example-based interpolation is also combined with strain field based approaches [MTGG11] to balance physical correctness with object properties learned from examples. We will demonstrate that by using a larger database, natural morphing can be achieved using a simpler and more efficient algorithm.

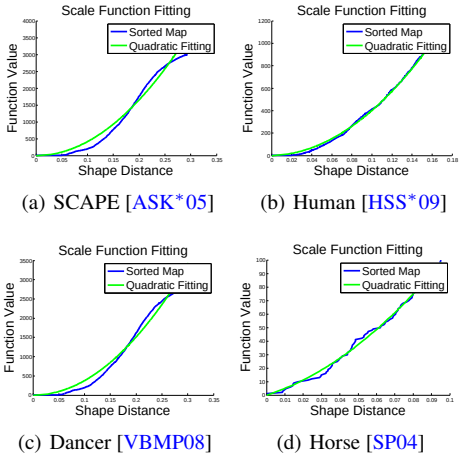
Due to improved acquisition techniques, data can now be obtained more easily and with lower cost. We propose to exploit a data-driven approach to improve shape morphing. With a database containing sufficient examples of some object, the naturalness of shape morphing can be significantly improved using constraints derived from the real captured models. Our approach uses local shape spaces to effectively represent plausible shapes. Meaningful parametric shape spaces have been extensively studied for human bodies [ACP03, SMT04, Wan05] due to its importance in practical; however, these methods use prior knowledge of human bodies and cannot be applied directly on arbitrary shapes. The work in [BVGP09] learns shape spaces from training examples to achieve semantic deformation transfer. Our approach is different in that we aim at shape morphing and use a set of local shape spaces derived automatically from a relatively large model database.

In a broader picture, data-driven shape morphing is related to motion planning [LaV06] in the robotics community. In their setting, a robot is usually characterized by its corresponding configuration space (which is specified by skeleton parameters). Given a source configuration and a target configuration, a motion planning algorithm aims at finding valid paths connecting them in the configuration space. The most successful techniques are graph-based [KScLO96, LaV06], where vertices correspond to samples in the configuration space, and edges connect neighboring vertices. Valid paths are then given by graph paths. Although our approach is conceptually similar, the fundamental difference is that we optimize the quality of the interpolated path while motion planning focuses more on the validity of the path.

3. Algorithm

We assume that a model database containing a reasonable number of models of the same category is available, capturing the variety of poses and behaviors of the object. Such databases are more widely available and can be obtained at relatively low costs. In the preprocessing stage, pairwise one-to-one correspondence can be established thanks to the recent progress of correspondence algorithms [HAWG08, LF09, KLF11]. With pairwise correspondence, we select one model as the base domain and the topology of other models can be replaced with this base domain.

As a data-driven approach, our method involves two stages. In the offline stage, we analyze the model database to form local shape spaces that better characterize the plausible distribution of models in the category. We first introduce a distance measure in Sec. 3.1 and give the details of local shape spaces in Sec. 3.2, which involves possible upsampling and clustering. This stage needs to be applied only once for a given database. In the online stage when the source and target models are given, we find reference models in the *local shape spaces* using an optimization algorithm (Sec. 3.3) and use them to guide the as-rigid-as-possible shape morphing (Sec. 3.4). Finally, we discuss a deformation transfer approach to deal with the difference between the input models and the models in the database (Sec. 3.5).



(a) SCAPE [ASK*05] (b) Human [HSS*09]

(c) Dancer [VBMP08] (d) Horse [SP04]

Figure 1: *Mapping distance $\bar{d}$ to its rank by $\bar{F}$ (in blue) and approximating the mapping with a quadratic function fitting (in green).*

3.1. Distance Measure

Assume the database contains N models each with n vertices and identical topology. We align them using rigid transforms with the known correspondences. Then a simple approach to define the distance between objects S_i and S_j is:

$$\bar{d}(S_i, S_j) = \sqrt{\frac{\sum_{k=1}^{n} \|S_i[k] - S_j[k]\|^2}{n}}, \qquad (1)$$

where $S_i[k]$ represents the k^{th} vertex of the i^{th} object. This distance metric is simple but does not show significant discernibility as a large number of distances distribute over a relatively small interval. This is particularly the case for small $\bar{d}$ where the subtle differences are more crucial. Since we have the database of models, we compute the pairwise distances $\bar{d}$ and define the function $\bar{F}(\bar{d})$ that maps the distance $\bar{d}$ to its rank (the position in the sorted list), as plotted in Figure 1 using SCAPE [ASK*05], Human [HSS*09], Dancer [VBMP08] and Horse [SP04] datasets. This nonlinear mapping increases the relative differences for the pairs with a smaller distance $\bar{d}$ as expected. To simplify the calculation, an analytical function would be preferred. In particular for the range of relatively small $\bar{d}$, $\bar{F}$ can be well approximated using a quadratic function $F(\bar{d}) = a\bar{d}^2$, where a is an unknown coefficient to be fitted. The constant of the quadratic function is zero because we expect $F(0) = 0$. The fitting results are shown in Figure 1 (green curves) which are reasonably close to the ranking functions (blue curves) for all of these datasets.

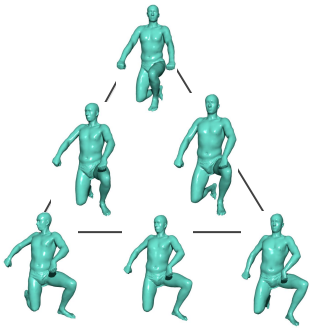


Figure 2: *An example cluster from SCAPE database and the interpolated models.*

As the exact value of a is not important for comparison purpose, we simply choose $a = 1$ and define the distance measure $d = F(\bar{d}) = \bar{d}^2$. Note that the distance measure is not a metric as it does not satisfy triangle inequality but as shown later this simple heuristic approach works well for measuring the closeness between models. Alternative distance measures could also be used but our simple distance leads to efficient computation, as detailed later in the paper.

3.2. Local Shape Spaces

Since the potential space of shape deformation is usually very large, the existing database which typically contains tens or hundreds of models may only provide a sparse sampling of the space. Two steps are used to achieve this, namely upsampling and local shape space construction.

In the first step, for datasets with a relatively small number of models (e.g. under 200), we find models which are close enough and use them to interpolate models and add them to the collection. This is achieved by first clustering the models in the database using Affinity Propagation algorithm [FD07] with the default parameter settings and within each cluster as models are close enough, we use as-rigid-as-possible interpolation [HAWG08] to sample a new model between every pair of models. An example is shown in Figure 2 with a cluster containing three models (triangle vertices) and use pairwise interpolation to obtain more plausible models. As these models are fairly close, the interpolated models are of high quality (triangle edges).

Even if the upsampled database is denser, it is not possible to directly sample all the plausible models due to the restrictions of computational costs. In the second step, we instead construct *continuous* local shape spaces to better represent plausible models. We cluster the upsampled model database into smaller groups again using Affinity Propagation algorithm with the default parameter settings and we make the assumption that simple linear interpolation is sufficient to produce reasonable in-between models (note that the first step is a non-linear interpolation). Assume there are m clusters $\mathbf{C} = \{C_k\}(k = 1,2,\ldots,m)$ in this step, and the cluster C_k contains N_k models, denoted as $S_{k,t}(t = 1,2,\ldots,N_k)$. We then build local linear shape spaces such that any model in the space w.r.t. cluster C_k is a linear combination of models in the cluster. Denote $w_{k,t}$ as weight for model t in the cluster C_k which can be chosen arbitrarily as long as $w_{k,t} \geq 0$ and $\sum_{t=1}^{N_k} w_{k,t} = 1$, the model cs_k is defined as $cs_k = \sum_{t=1}^{N_k} w_{k,t} S_{k,t}$.

3.3. Path Optimization

Given the source model S_S and the target model S_T, after establishing the correspondences, we make S_S and S_T have the same connectivity as models in the database. We cast the morphing problem as a global optimization problem of finding a shortest path from the source to the target with intermediate models from the *local shape spaces* using the distance measure. As we will show later, using the continuous shape space allows smooth paths to be found with a relatively sparse set of models. Since models are fairly close within each cluster, we assume at most one representative model in each cluster is chosen in the path. We treat the source and target models as clusters C_S and C_T with single models only. Let us denote $C_0 = C_S$ and $C_{m+1} = C_T$, we can formulate the shortest path as a 0-1 integer programming problem:

$$Minimize \sum_{i=0}^{m+1} \sum_{j=0}^{m+1} f_{ij} x_{ij}, \tag{2}$$

$$subject\ to: \sum_{j=0}^{m+1} x_{ij} - \sum_{j=0}^{m+1} x_{ji} = \begin{cases} 1 & i = 0 \\ -1 & i = m+1 \\ 0 & otherwise \end{cases}. \tag{3}$$

x_{ij} are 0-1 logical variables. $x_{ij} = 1$ means morphing from an element in C_i to an element in C_j forms part of the path. The distance between different clusters are defined as

$$f_{ij} = F\left(d_{ij}\right) = \bar{d}(cs_i, cs_j)^2, \tag{4}$$

where cs_i and cs_j are the representative element from clusters C_i and C_j respectively, which are determined by the set of weights $\{w_{i,t}\}$ and $\{w_{j,t}\}$, and d_{ij} is the distance between cs_i and cs_j.

The global optimization involves choosing the appropriate weights for cluster reprentatives and finding appropriate x_{ij} to determine the path w.r.t. clusters. This is in general an NP hard problem. We thus propose alternating two steps (an EM-like optimization) to find an approximate solution efficiently. We start with representatives chosen as the cluster centers (with equal weights). In the first step, the representatives are fixed and x_{ij} are optimized. This can be efficiently optimized using Dijkstra algorithm on the graph that connects every pair of clusters with edge weight between cluster C_i and C_j being f_{ij}. In the second step, given x_{ij}, we find the optimal combination weights $w_{i,t}$ for each cluster C_i. This is a quadratic programming problem which is solved using CVX library, a package for specifying and solving convex

Figure 3: *Reference models found without (top) and with (bottom) local spaces.*

programs [CR12]. The above steps are iterated until convergence. As the total distance decreases monotonically the method always converges and it typically takes only a few iterations in practice.

An example is shown in Figure 3 comparing different reference models obtained. The first and the last columns are source and target models. Using the SCAPE database, if we do not add interpolated models within each cluster, no reference models are found (i.e., the direct path from the source to the target is optimal). If these models are included, but without using local shape spaces, three intermediate reference models are picked (top row) which give reasonable guidance from the source to the target. However, the transition is sometimes not very smooth. For example, unnecessary head shake appears from the first column to the third. By using local shape spaces, any model in the local space can be picked, leading to a smoother set of reference models (bottom row).

3.4. Reference Guided Interpolation

After solving the above optimization problem, a sequence of intermediate reference models is obtained from the database. We denote these models as $\bar{S}_k$, $k = 1,2,\ldots,N_R$ where N_R is the number of the reference models. Given a parameter t $(0 \le t \le 1)$, let us denote $\hat{S}(t)$ as the interpolated model at time t. The corresponding time for reference models $\bar{S}_k$ can be uniformly distributed as $t_k = \frac{k-1}{N_R-1}$, or specified by the user. The energy between the unknown $\hat{S}(t)$ and $\bar{S}_k$ is defined as follows:

$$E_k = \sum_{i=1}^{n} \Big\{ \sum_{j\in N_i} w_{ij} \| (\hat{\mathbf{S}}[i]-\hat{\mathbf{S}}[j]) - \mathbf{R}_k[i](\bar{\mathbf{S}}_k[i]-\bar{\mathbf{S}}_k[j]) \|^2 + \gamma \| \hat{\mathbf{S}}[i]-\bar{\mathbf{S}}_k[i] \|^2 \Big\}, \quad (5)$$

where n is the number of vertices, N_i is the 1-ring neighbors of vertex i, w_{ij} is the cotangent weight $w_{ij} = \cot\alpha_{ij} + \cot\beta_{ij}$ [MDSB03] where α_{ij} and β_{ij} are two angles opposite to edge ij. $\mathbf{R}_k[i]$ is the optimal rotation matrix between $\hat{S}(t)$ and $\bar{S}_k$ at vertex i, which can be efficiently computed using Singular Value Decomposition [SA07]. The second term ensures that the interpolated shape is close to the reference model. γ is a constant and chosen as 0.001 in practice.

The overall energy $E(t)$ between $\hat{S}(t)$ and all the reference models are defined as:

$$E(t) = \sum_{k=1}^{N_R} \omega_k(t) E_k. \quad (6)$$

The weights ω_k are defined using Gaussian functions $\omega_k \propto \exp\{-\varepsilon|t-t_k|^2\}$ and normalized to be a partition of unity. ε controls the influence range and typically we used $\varepsilon = 6.0$ in our experiments.

To find the solution $\hat{S}(t)$ at time t, we first use linear interpolation to provide the initial value. The problem can then be solved by iterating the following two steps.

First, the optimal rigid transform $\mathbf{R}_k[i]$ for each vertex i w.r.t. the reference model $\bar{S}_k$ can be computed as follows [SA07]. Let $\mathbf{P}_k[i] = \sum_{j\in N_i} (\bar{\mathbf{S}}_k[i]-\bar{\mathbf{S}}_k[j])(\hat{\mathbf{S}}[i]-\hat{\mathbf{S}}[j])^T$. $\mathbf{P}_k[i]$ can be decomposed using singular value decomposition. $\mathbf{P}_k[i] = \mathbf{U}_k[i]\Sigma_k[i]\mathbf{V}_k[i]^T$. Then, $\mathbf{R}_k[i]$ can be explicitly obtained as $\mathbf{V}_k[i]\mathbf{U}_k[i]^T$ (choosing appropriate signs to make $\det \mathbf{R}_k[i] > 0$).

Second, assuming $\mathbf{R}_k[i]$ is known, the optimal position for $\hat{S}(t)$ can be obtained by solving the linear system $\frac{\partial E(t)}{\partial \hat{S}(t)} = 0$.

Since $w_{ij} = w_{ji}$, $\sum_{k=1}^{N_R} \omega_k(t) = 1$, this can be rewritten as

$$\begin{aligned} \frac{\partial E(t)}{\partial \hat{S}[i]} &= \sum_{k=1}^{N_R} \omega_k \{ \sum_{j\in N_i} 2w_{ij} \Big((\hat{\mathbf{S}}[i]-\hat{\mathbf{S}}[j]) - \mathbf{R}_k[i](\bar{\mathbf{S}}_k[i]-\bar{\mathbf{S}}_k[j]) \Big) \\ &+ \sum_{j\in N_i} 2w_{ji} \Big((\hat{\mathbf{S}}[i]-\hat{\mathbf{S}}[j]) - \mathbf{R}_k[j](\bar{\mathbf{S}}_k[i]-\bar{\mathbf{S}}_k[j]) \Big) \\ &+ 2\gamma(\hat{\mathbf{S}}[i]-\bar{\mathbf{S}}_k[i]) \} = 0. \end{aligned} \quad (7)$$

$$\sum_{j\in N_i} (2w_{ij}+\gamma)(\hat{\mathbf{S}}[i]-\hat{\mathbf{S}}[j]) = \sum_{k=1}^{N_R} \omega_k(t) \Big(\sum_{j\in N_i} w_{ij} \big(\mathbf{R}_k[i]+\mathbf{R}_k[j]\big)(\bar{\mathbf{S}}_k[i]-\bar{\mathbf{S}}_k[j]) \Big) + \gamma\bar{\mathbf{S}}_k[i].$$

If we rewrite the linear system as $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{x}$ is the collection of unknown vertex positions of $\hat{S}(t)$, for different t, $\mathbf{A}$ is fixed, and only $\mathbf{b}$ changes. Moreover when $\gamma > 0$, $\mathbf{A}$ is a strictly diagonally dominant symmetric matrix. $\mathbf{A}$ is positive definite and can be effectively factorized using Cholesky decomposition. For interpolation results with various t (which is often needed for morphing), $\mathbf{A}$ only needs to be decomposed once, and the linear system can be solved using back substitution.

3.5. Deformation Transfer for Morphing

The given source and target models may in general be different from the models in the database. We can still use

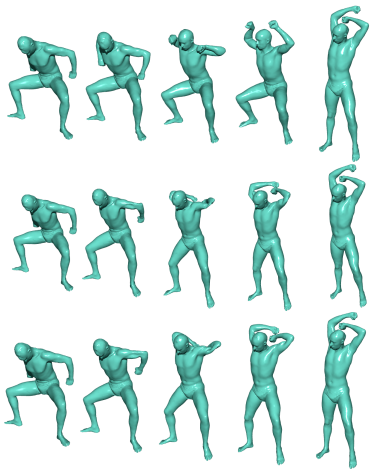


Figure 4: *Comparison of human morphing using our method (top), [XZWB06] (middle) and [HAWG08] (bottom). Unnatural deformations occur using these two methods.*

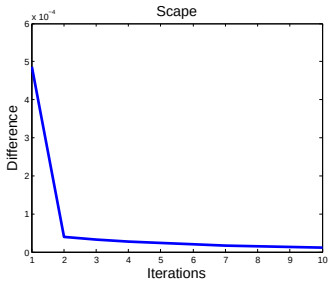


Figure 5: *The average vertex displacement after each iteration for the example in Figure 4.*

the database to guide the morphing if they are reasonably similar. Let N_S and N_T be the new source and target models different from those in the database. We make a further assumption that there exist models S_S and S_T in the local shape spaces such that they have similar pose, which is reasonable when the model database has a good coverage of the underlying shape space. We then establish one-to-one correspondences between N_S/N_T and models in the database, using blended intrinsic maps [KLF11] if automatically, or the method in [SP04] with the user specifying landmark correspondences. As models in the database have the same connectivity, establishing correspondences between N_S/N_T and an arbitrary model in the database is sufficient. To simplify the computation, we remesh N_S and N_T to have the same connectivity, leading to $\hat{N}_S$ and $\hat{N}_T$, respectively. This is achieved by taking the mesh connectivity from the database models and using the geometry of N_S and N_T. We can then follow the previous definition to find the closest models S_S and S_T. These models may not have the same pose, so we deform S_S (and S_T) in an as-rigid-as-possible manner to obtain $\hat{S}_S$ (and $\hat{S}_T$). We give the formula to be optimized for $\hat{S}_S$ here:

$$E(\hat{S}_S) = \sum_{i=1}^{n} \left\{ \sum_{j \in N_i} w_{ij} \| (\hat{\mathbf{S}}_S[i] - \hat{\mathbf{S}}_S[j]) - \mathbf{R}[i](\mathbf{S}_S[i] - \mathbf{S}_S[j]) \|^2 + \lambda \| \hat{\mathbf{S}}_S[i] - \hat{\mathbf{N}}_\mathbf{S}[i] \|^2 \right\}, \quad (8)$$

where n, N_i, w_{ij} and $\mathbf{R}[i]$ are defined as before, λ is chosen in $[0.2, 1]$ to balance the terms of local rigidity and geometric closeness. After this, we treat $\hat{S}_S$ and $\hat{S}_T$ as the source and target models of the same shape as in the database and find reference models $\bar{S}_k$ ($k = 1, 2, \ldots, N_R$) as in Sec. 3.3. We then transfer the deformation between $\hat{S}_S$ (or $\hat{S}_T$ depending on which is closer) and $\bar{S}_k$ to $\hat{N}_S$ (or $\hat{N}_T$) to obtain reference models $\hat{S}_k$ for the actual input shapes. For each face of $\hat{S}_S$ and $\bar{S}_k$, we add an auxiliary vertex as in [SP04] to ensure a unique affine transform for each face f, which is calculated and applied to the corresponding face of $\hat{N}_S$ and the Poisson equation is then used to fuse the triangles [YZX*04]. The obtained reference models $\hat{S}_k$ are used in the same way as $\bar{S}_k$ in Sec. 3.4 to obtain the in-between frames.

4. Experimental Results

We demonstrate the effectiveness of the proposed method using various examples. We carried out experiments on a computer with an Intel Core2Quad 2.4GHz CPU. The model databases we used and the running times are reported in Table 1. SCAPE and Human datasets are reconstructed from scanned models and Dancer dataset from multi-view videos. Horse and Cat datasets are obtained by articulated deformation based on results in [SP04].

Figure 4 shows an example using SCAPE [ASK*05] database with 71 human models. For all these examples, the first and last columns show the source and target models and the intermediate frames give the interpolated in-between frames. While the results with [XZWB06] and [HAWG08] are reasonable geometrically, the poses are not achievable by a normal person and thus the morphing does not look natural. Guided by the models in the database, we are able to produce natural deformation. The energy $E(t)$ in Equation 6 converges quickly. Figure 5 shows the average vertex displacement (as defined in Equation 1) after each iteration and three iterations are sufficient for examples in the paper. The running time for each iteration of the optimization is only about 1 second, as given in Table 1, so it takes a few seconds to compute each frame of morphing. Another example using the SCAPE database is shown in Figure 6. Due to the significant pose changes, previous methods [XZWB06, HAWG08] produce morphing with self-intersection and sometimes unnaturally twisted in-between frames. Although our method cannot guarantee that the obtained morphing is free from self-intersections, thanks to the guidance of the database, the deformation follows a natural path.

Dataset	# Vertices	# Models	# Models After Interp.	Finding Path (s)	Cholesky (s)	Iteration (s)
SCAPE	12500	71	225	32.22	1.27	1.13
Human	6449	500	500	43.69	1.58	0.85
Dancer	9971	149	485	89.28	4.13	0.86
Horse	8431	49	173	45.50	1.41	0.76
Cat	7207	57	162	21.18	1.38	0.71

Table 1: *Statistics of the datasets and average running times.*

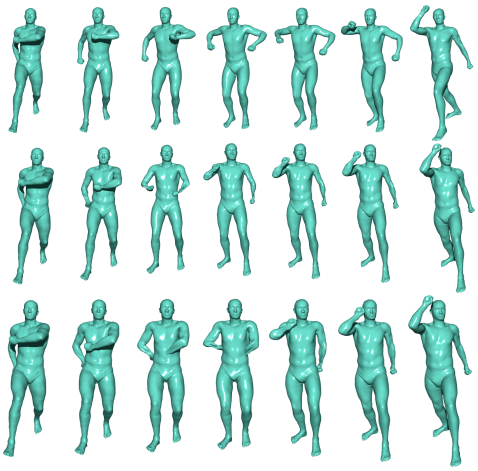


Figure 6: *Comparison of human morphing using our method (top row), [XZWB06] (middle row) and [HAWG08] (bottom row). Both unnatural deformation and self-intersections occur with [XZWB06] and [HAWG08].*

Figure 7: *Comparison of dancer morphing using our method (top), [XZWB06] (middle) and [HAWG08] (bottom). Self-intersections occur with both [XZWB06] and [HAWG08].*

Figure 7 shows an example using the Dancer database [VBMP08] with 149 models. While both results of [XZWB06] and [HAWG08] look smooth geometrically, there are self-intersections (the second column). In contrast, our method produces a longer but natural path of movement. Figure 8 is an example obtained using the Horse dataset [SP04]. The pose change is relatively small so the alternative geometric method [HAWG08] also gives a reasonable result. However, due to the physical balance, the bottom of the hooves lies flat when it hits the ground. This is shown in the third column of our output. Comparatively the deformation produced by [HAWG08] does not look as balanced. Our method also produces richer and more natural tail deformation than [HAWG08].

By using deformation transfer (Sec. 3.5), our approach is capable of handling more general morphing where the source and target models are different from those in the database. Figure 9 gives an example of morphing from a woman to a man. We use the Human dataset which do not contain these models. Figure 10 shows an example of morphing from a lion to a cat, in which case a Cat database is used. In both examples, a sequence of reference models in the database are first obtained (the first row) and the deformation transferred to the actual source and target models (first and last columns in the second row). Note that the reference models are different and sometimes of relatively poor quality, but they are still useful to guide the morphing process to make it more natural. Comparatively, purely geometry-based methods produce unnatural deformation, as for the first example the dynamics of moving the hands up from the back side are generally impossible, and for the second example the tail of the animal is not swinging smoothly.

One limitation of our method is a relatively large database covering plausible deformation is needed. If the size of the input database is small, the relevant part of the morphing will revert back to the traditional approach. Another limitation is that we require appropriate correspondences between the source/target models and database models to morph models that are different from those in the database. Thus, our method may generate low quality results if those models differ substantially as it will be non-trivial to establish reliable correspondences. Our data-driven morphing tends to produce results with the similar behavior as the models in the database and thus it would be more semantically mean-

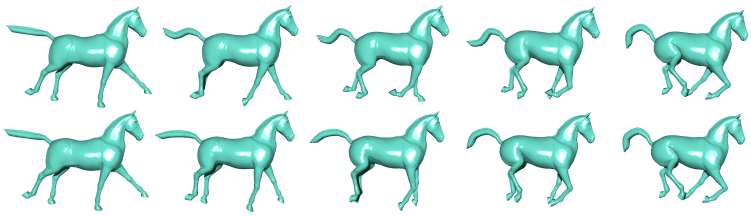


Figure 8: *Comparison of horse morphing using our method (top row) and [HAWG08] (bottom row).*

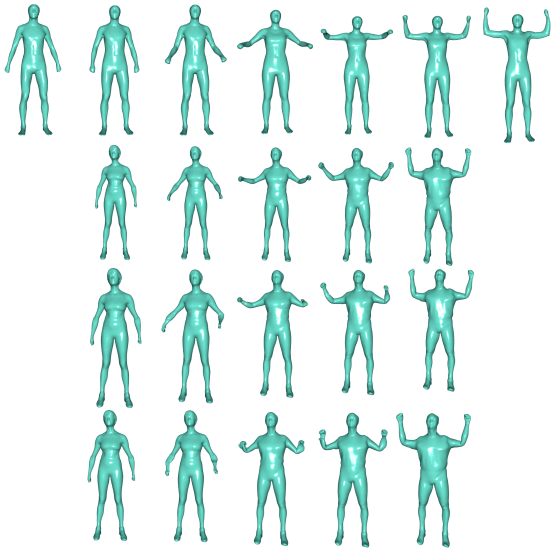


Figure 9: *Comparison of human morphing using our method (second row), [XZWB06] (third row) and [HAWG08] (fourth row). The first row shows the reference models found in the Human database.*

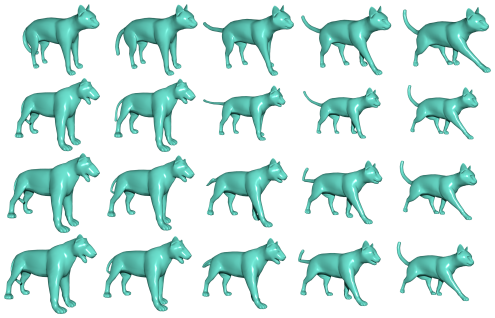


Figure 10: *Comparison of morphing from a lion to a cat using our method (second row), [XZWB06] (third row) and [HAWG08] (fourth row). The first row shows the reference models found in the Cat database.*

ingful if the database models are closer to the source/target models.

5. Conclusions

In this paper, by exploiting the information in model databases, we propose a data-driven approach to improve mesh morphing. The models in the database provide valuable information showing what a natural pose of the objects should look like which would otherwise be difficult/impossible to obtain solely from the geometry of the source and target objects alone. This helps to produce a natural deformation path as all the in-between frames are well guided. Our method produces realistic results for challenging cases. Furthermore, since reference frames are used, the artifacts are reduced as more constraints, known to be correct are given.

Our current approach only considers the energy minimizing path and as future work, it is possible to suggest multiple answers by considering several paths with low energies. Data-driven morphing can potentially be extended to other applications, such as motion sequence editing [KG08,CH12] where the database can be exploited to improve the naturalness of the edited sequences.

Acknowledgements

We would like to thank Jia-jia Sun for his help with experiments. This work was supported by the National Basic Research Project of China (Project Number 2011CB302202), the Natural Science Foundation of China (Project Number 61120106007), the National High Technology Research and Development Program of China (Project Number 2012AA011801) and National Significant Science and Technology Program (Project Number 2012ZX01039001-003).

References

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proc. ACM SIGGRAPH* (2000), pp. 157–164. 2

[ACP03] ALLEN B., CURLESS B., POPOVIC Z.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph. 22*, 3 (2003), 587–594. 3

[Ale03] ALEXA M.: Differential coordinates for local mesh morphing and deformation. *The Visual Computer 19*, 2-3 (2003), 105–114. 2

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (2005), 408–416. 3, 6

[BVGP09] BARAN I., VLASIC D., GRINSPUN E., POPOVIĆ J.: Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (2009), 36:1–6. 3

[CH12] CASHMAN T. J., HORMANN K.: A continuous, editable representation for deforming mesh sequences with separate signals for time, pose and shape. *Comput. Graph. Forum 31*, 2 (2012), 735–744. 8

[CL09] CHU H.-K., LEE T.-Y.: Multiresolution mean shift clustering algorithm for shape interpolation. *IEEE Trans. Vis. Comp. Graph. 15* (2009), 853–866. 2

[CR12] CVX RESEARCH I.: CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, 2012. 5

[FB11] FRÖHLICH S., BOTSCH M.: Example-driven deformations based on discrete shells. *Comput. Graph. Forum 30*, 8 (2011), 2246–2257. 2

[FD07] FREY B. J., DUECK D.: Clustering by passing messages between data points. *Science 315* (2007), 972–976. 4

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. *ACM Trans. Graph. 23*, 3 (2004), 522–531. 2

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. In *Proc. Symposium on Geometry Processing* (2008), pp. 1449–1457. 2, 3, 4, 6, 7, 8

[HLW07] HU J., LIU L., WANG G.: Dual Laplacian morphing for triangular meshes. *Comput. Animat. Virtual Worlds 18*, 4-5 (2007), 271–277. 2

[HLZ04] HU S.-M., LI C.-F., ZHANG H.: Actual morphing: a physics-based approach to blending. In *Proc. ACM Symposium on Solid Modeling and Applications* (2004), pp. 309–314. 2

[HSS*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. *Comput. Graph. Forum 28*, 2 (2009), 337–346. 3

[KCATLF06] KIN-CHUNG AU O., TAI C.-L., LIU L., FU H.: Dual Laplacian editing for meshes. *IEEE Trans. Vis. Comp. Graph. 12*, 3 (2006), 386–395. 2

[KG08] KIRCHER S., GARLAND M.: Free-form motion processing. *ACM Trans. Graph. 27*, 2 (2008), 12:1–13. 8

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *ACM Trans. Graph. 30*, 4 (2011), 79:1–12. 3, 6

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. *ACM Trans. Graph. 26*, 3 (2007), 64:1–8. 2

[KScLO96] KAVRAKI L., SVESTKA P., CLAUDE LATOMBE J., OVERMARS M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. In *IEEE Intl. Conf. on Robotics and Automation* (1996), pp. 566–580. 3

[LaV06] LAVALLE S. M.: *Planning Algorithms*. Cambridge University Press, 2006. 3

[LDSS99] LEE A. W. F., DOBKIN D., SWELDENS W., SCHRÖDER P.: Multiresolution mesh morphing. In *Proc. ACM SIGGRAPH* (1999), pp. 343–350. 2

[LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Trans. Graph. 28*, 3 (2009), 72:1–12. 3

[LH03] LEE T.-Y., HUANG P.-H.: Fast and intuitive metamorphosis of 3D polyhedral models using smcc mesh merging scheme. *IEEE Trans. Vis. Comp. Graph. 9* (2003), 85–98. 2

[LL05] LIN C.-H., LEE T.-Y.: Metamorphosis of 3D polyhedral models using progressive connectivity transformations. *IEEE Trans. Vis. Comp. Graph. 11*, 1 (2005), 2–12. 2

[LSLCO05] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph. 24*, 3 (2005), 479–487. 2

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Vis. and Math. III* (2003), pp. 35–57. 5

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M. H.: Example-based elastic materials. *ACM Trans. Graph. 30*, 4 (2011), 72:1–8. 2

[PSS01] PRAUN E., SWELDENS W., SCHRÖDER P.: Consistent mesh parameterizations. In *Proc. ACM SIGGRAPH* (2001), pp. 179–184. 2

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proc. Symposium on Geometry Processing* (2007), pp. 109–116. 5

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Trans. Graph. 23*, 3 (2004), 870–877. 2

[SMB12] SIEGER D., MENZEL S., BOTSCH M.: High quality mesh morphing using triharmonic radial basis functions. In *Proc. International Meshing Roundtable* (2012), pp. 1–15. 1

[SMT04] SEO H., MAGNENAT-THALMANN N.: An example-based approach to human body manipulation. *Graph. Models 66*, 1 (2004), 1–23. 3

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (2004), 399–405. 2, 3, 6, 7

[SRC01] SLOAN P.-P. J., ROSE III C. F., COHEN M. F.: Shape by example. In *Proc. Symposium on Interactive 3D Graphics* (2001), pp. 135–143. 2

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *ACM Trans. Graph. 24*, 3 (2005), 488–495. 2

[VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph. 27*, 3 (2008), 97:1–9. 3, 7

[Wan05] WANG C. C.: Parameterization and parametric design of mannequins. *Computer-Aided Design 37*, 1 (2005), 83–98. 3

[WDAH10] WINKLER T., DRIESEBERG J., ALEXA M., HORMANN K.: Multi-scale geometry interpolation. *Comput. Graph. Forum 29*, 2 (2010), 309–318. 2

[WSLG07] WEBER O., SORKINE O., LIPMAN Y., GOTSMAN C.: Context-aware skeletal shape deformation. *Comput. Graph. Forum 26*, 3 (2007), 265–273. 2

[XZWB06] XU D., ZHANG H., WANG Q., BAO H.: Poisson shape interpolation. *Graph. Models 68*, 3 (2006), 268–281. 2, 6, 7, 8

[YHM07] YAN H.-B., HU S.-M., MARTIN R. R.: 3D morphing using strain field interpolation. *J. Comput. Sci. Technol. 22*, 1 (2007), 147–155. 2

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph. 23*, 3 (2004), 644–651. 2, 6