Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2009)
E. Grinspun and J. Hodgins (Editors)

Spatial Deformation Transfer

Mirela Ben-Chen   Ofir Weber   Craig Gotsman

Technion – Israel Institute of Technology

Abstract

Much effort is invested in generating natural deformations of three-dimensional shapes. Deformation transfer simplifies this process by allowing to infer deformations of a new shape from existing deformations of a similar shape. Current deformation transfer methods can be applied only to shapes which are represented as a single component manifold mesh, hence their applicability to real-life 3D models is somewhat limited. We propose a novel deformation transfer method, which can be applied to a variety of shape representations – tet-meshes, polygon soups and multiple-component meshes. Our key technique is deformation of the space *in which the shape is embedded. We approximate the given source deformation by a harmonic map using a set of harmonic basis functions. Then, given a sparse set of user-selected correspondence points between the source and target shapes, we generate a deformation of the target shape which has differential properties similar to those of the source deformation. Our method requires only the solution of linear systems of equations, and hence is very robust and efficient. We demonstrate its applicability on a wide range of deformations, for different shape representations.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Creating a natural deformation of a 3D shape is a difficult and time-consuming task. The body of research devoted to 3D shape deformation is huge, and yet the problem is still not considered solved. This problem could be significantly alleviated by reusing existing deformations of similar shapes to generate new deformations. In their influential paper, Sumner and Popovic [SP04] described how to reuse deformations using an approach called "deformation transfer" (DT). The setup is as follows: Given a source reference shape in some pose, a set of deformed source poses, and a target reference shape in a pose similar to the source reference pose, generate a set of deformed target poses, which are "analogous" in some way to the source deformations. For example, if we are given a horse as the source reference pose, keyframes of a gallop animation as deformed source poses, and a dog as the target reference shape, the output of the deformation transfer are keyframes of a galloping dog. The DT method successfully transfers deformations, however, it is only applicable to single component manifold triangle meshes. As real-life models more than often do not fall into this category – for example: multiple-component meshes, polygon soups, and tetrahedral volumetric meshes, the DT method is somewhat limited.

Recently, new methods for shape deformation (not deformation *transfer*) have been proposed [JSW05, LLCO08, WBCG09, BCWG09, SSP07, BPWG07]. These deform the ambient space the shape “lives” in, instead of the shape itself. Such methods are very versatile, since they can be applied to any 3D shape representation, and are not limited to manifold triangular meshes. In this paper we propose to combine the two ideas to facilitate *spatial deformation transfer*. This allows us to transfer a deformation between various shape representations. For example, we can use a skeleton-driven animation as the input deformation, and transfer it to a multiple-component model. Hence, our method is much more versatile than the original DT approach.

The deformation transfer process consists of two basic components: 1) Analysis of the source deformation, to extract the "essence" of the deformation, and 2) Synthesis of a deformation of the target shape using the same "essence", thus mimicking the source deformation. In order to transfer the information from the source to the target shapes, some sparse correspondence between the two reference shapes is needed. For example, a foot corresponds to a foot, a nose to a nose and so on.

Given such a correspondence, we propose novel methods for the analysis and synthesis components of the deformation transfer. First, we enclose the source and target reference shapes with two polyhedral domains - *cages*. Then, to analyze the source deformation, we project the deformation onto a linear space of harmonic maps on the source cage. We use the harmonic basis functions proposed in the recent "Variational Harmonic Maps" (VHM) deformation method [BCWG09], and show that many deformations can be approximated this way with a relatively small error. The main advantage of such a representation, is that the differential properties of the deformation, such as the Jacobians, can be computed analytically and easily transferred to the target deformation. Hence, we consider the Jacobians of the mapping at the correspondence points to be the "essence" of the deformation, and transfer this to the target shape. For the synthesis of the target deformation, we use a method similar to VHM deformation (yet linear), which, in essence, is just a reversal of the analysis procedure.

We demonstrate the applicability of our method by transferring deformations from different input representations to various output representations. We show additionally, that for manifold meshes, our method performs comparably to the original DT method, while avoiding the need to compute a dense correspondence between the meshes. Furthermore, our method is efficiently implemented on the GPU, hence the entire deformation transfer process may be done in real-time: the user can manipulate the correspondence points to interactively control the result of the deformation transfer. Moreover, it is possible to simultaneously deform two shapes at interactive rates, by deforming one shape and transferring the deformation to the other.

1.1. Previous work

Since quite a few shapes are involved in the deformation transfer process, some terminology is in order. A deformation transfer method receives as input a *source reference pose*, a *deformed source pose* and a *target reference pose.* The output of the deformation transfer is a *deformed target pose*. Typically many deformed source poses are given, usually the keyframes of an animation sequence. Deformation transfer can be applied to each of them independently, resulting in keyframes of the target animation sequence. See Figure 1 for an illustration.

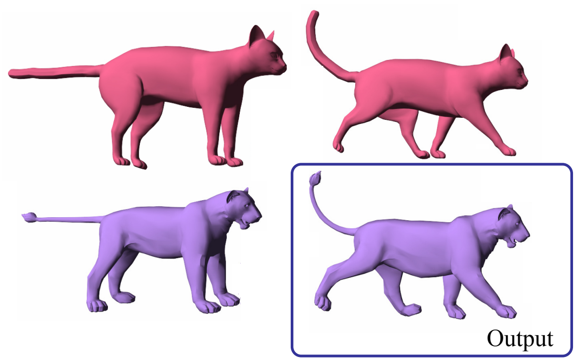

Figure 1: *Terminology of deformation transfer. (top left) Source reference pose. (top right) source deformed pose. (bottom left) target reference pose. (bottom right) target deformed pose – the output of the deformation transfer process.*

The deformation transfer problem can be decomposed into two independent sub-problems – analyzing the source deformation, and synthesizing a new deformation for the target shape. The output of the analysis step is a descriptor of the deformation, having some desirable invariance properties. For example, it should be invariant to global translations and/or rotations of both the reference and deformed poses. Once such a descriptor is generated, it is applied to the target reference pose to create a new deformation.

One of the most popular deformation descriptors is the so-called *deformation gradient*. When the deformation is given by two triangular meshes with the same (“compatible”) triangle structure, the deformation gradient of a triangle is the unique affine transform which maps the tetrahedron spanned by the source reference triangle and its normal vector to the tetrahedron spanned by the corresponding deformed triangle and its scaled normal. Since, for triangle meshes, the deformation function (on the surface) is a piecewise linear deformation, mapping triangles to triangles, the deformation gradient is just the (piecewise-constant) Jacobian of the deformation function. This deformation descriptor is invariant to global translations, and is widely used for deformation transfer [SP04, ZRKS05, XZY*07], shape editing [YZX*04, SA07, BSPG06] and shape blending [SZGP05]. However, such a deformation descriptor can only be extracted when the source reference and deformed poses are given as compatible triangle meshes. Our method can be considered a generalization of this approach. We approximate the given source deformation by a harmonic mapping of 3D space, whose Jacobian has a closed form expression. Hence, we can compute the Jacobian of the deformation for any point inside the source cage, and transfer it to the target deformation.

Once the deformation gradient information has been computed, the deformation of the target shape can be generated by applying any shape deformation method which takes advantage of these gradients. The most straightforward way to do this on a triangular mesh, which was employed in [SP04, ZRKS05, SZGP05, YZX*04], is to solve a Poisson problem, i.e. to find a mesh whose gradients (relative to the target reference mesh) are as similar as possible, in the least-squares sense, to the deformation gradients of the source deformation. To solve this problem, the gradients of *all* the target triangles are required. Since usually the correspondence given by the user is a sparse correspondence, only a sparse set of gradients can be transferred from the source to the target mesh. Hence, in order to perform deformation transfer in practice, one of the following two methods must be used. Either [SP04] generate a dense correspondence from the source to the target reference poses, and transfer *all* the gradients from the source deformation, or [ZRKS05] transfer the gradients only at the correspondence points, and then propagate these (i.e. interpolate them) to the rest of the target mesh.

The first method is somewhat problematic, as generating a dense correspondence between unrelated meshes is a difficult problem in itself. On the other hand, the second method cannot be applied directly to spatial deformation transfer, as it will require the computation of harmonic coordinates on the *interior* of the shape, which is computationally expensive. Hence, we opt for a different approach altogether. We pose the deformation transfer problem as a regular shape deformation problem, where a sparse set of orientation constraints are "learned" from the source deformation, as opposed to the traditional deformation scenario, where a sparse set of positional constraints are obtained interactively from a user. In fact, from this point of view, deformation transfer is even *simpler* than shape deformation, as one of the most difficult challenges in shape deformation is to infer orientations from positional constraints (which are much more user-friendly than orientation constraints). In practice, we generate the target deformation by modifying slightly the recently proposed space deformation method "Variational Harmonic Maps" (VHM) of Ben-Chen *et al.* [BCWG09].

For completeness' sake, we give a brief overview of the VHM method. To apply the VHM deformation to a shape, one should supply also a polyhedral cage enclosing the shape, and a set of positional and/or Jacobian constraints that the deformed shape should satisfy. By solving a non-linear optimization problem on the cage, the VHM method finds a harmonic map which satisfies the constraints, expressed as a linear combination of harmonic basis functions. Both the Jacobian and Hessian of the deformation map can be computed using closed-form expressions. The harmonic map is then used to deform the given shape.

A similar application to deformation transfer is *motion retargeting* - adapting motion capture data to characters with different proportions [Gle98]. Although this application is somewhat similar to ours, motion retargeting algorithms cannot be applied directly in our setup, where the animation is not represented by skeletal movement, rather by the raw geometry of the keyframe shapes.

1.2. Method overview

The spatial deformation transfer method receives as input a source reference pose, a deformed source pose, and a target reference pose. The output of the process is a deformed target pose, such that the target deformation mimics the source deformation. See for example, Figure 1. The first step in the process is to generate cages for the source and target reference poses. Once the cages are computed, the source poses are analyzed, to compute the best approximating harmonic maps of the deformation. Given a sparse set of corresponding *landmarks* between the source and target reference poses, we compute the Jacobian of the source deformation at the source landmarks, and transfer them to the corresponding target landmarks. Finally, we seek a harmonic map of the target cage, which best approximates these constraints at the target landmarks, using a variant of VHM. The whole process may be user-guided, in the sense that the user may interactively modify the corresponding landmarks on the source reference shape, until the desired result is obtained.

The rest of the paper is organized as follows. In the next Section, we explain how the source deformation is approximated using a harmonic map, in order to extract the deformation gradients. In Section 3, we show how to use these gradients to synthesize the target deformation. Section 4 presents various deformation transfer examples, and comparisons with the original DT method. We conclude with a discussion and some future research directions in Section 5.

2. Deformation analysis by harmonic projection

The first step in the deformation transfer process is to analyze the source deformation. To be as general as possible, we assume the input poses are given as a collection of points. Hence, $S = \{p_1, p_2, ..., p_m \mid p_i \in R^3\}$ is the set of m points of the reference source pose, and $\tilde{S} = \{\tilde{p}_1, \tilde{p}_2, ..., \tilde{p}_m \mid \tilde{p}_i \in R^3\}$ is the corresponding set of points of some deformed source pose. Our goal is to find a smooth function f, which maps the reference source pose to the deformed pose: $f(p_i) = \tilde{p}_i$. If an analytic description of such a function is available, we can compute its Jacobian matrices, and use them as our deformation descriptor.

This analysis can be considered an interpolation problem, and, as is common in these cases [Kyt95], we try to approximate f as a linear combination of a set of basis functions. Inspired by the recent work indicating that harmonic functions generate pleasing deformations [JMD*07, LLCO08, WBCG09, ZRKS05, WSLG07], we choose as our basis functions the harmonic functions successfully used for shape deformation in the VHM method [BCWG09]. To compute these functions, the shape should be contained in a polyhedral mesh – a *cage,* which largely determines the character of the basis functions. Denote by $C_S = \{V_S, F_S\}$ a cage enclosing the reference source shape S, where V_S are the vertices, and F_S the faces of C_S, and by Ω_s the interior of C_s.

Let $h_j^S : \Omega_S \to R, j = 1..a_S$, be the VHM harmonic basis functions defined in [BCWG09], where $a_S = |V_S| + |F_S|$. The deformation f is defined by its coefficients w_j^S on this basis:

$$f(x) = \sum_{j=1}^{a_S} w_j^S h_j^S(x)$$

To project our deformation onto the basis, we must solve the following optimization problem:

$$\min_{w_1^S,\ldots,w_{a_S}^S \in R^3} E_{Approx} = \sum_{i=1}^{m} \left\| \sum_{j=1}^{a_S} w_j^S h_j^S(p_i) - \tilde{p}_i \right\|_2^2 \qquad (1)$$

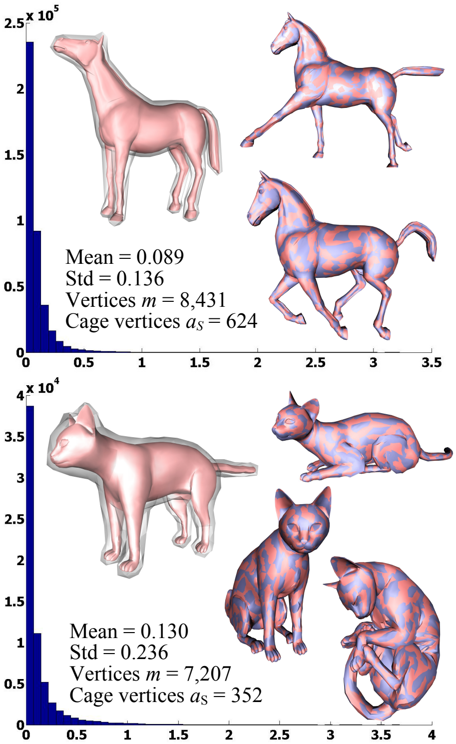

Figure 2: *Reconstruction error of harmonic projection, per vertex, as % of the bounding box diagonal, for two sets of poses, including 9 cats, and 48 horses. Also shown - the reference pose within its cage, and a few representative reconstructions (purple), overlayed on the original shape (pink).*

This is an over-determined linear least-squares problem, having a closed-form solution. However, usually $a_S \ll m$, so it is not immediately clear why such an approximation would be good, in the sense that the approximation error E_{Approx} will be sufficiently small. To empirically justify the use of these basis functions, we have computed the approximation error (1) for a few sets of reference and deformed poses. For each source pose we computed a histogram of the approximation error per vertex from all the poses, given by $E(p_v) = \| f(p_v) - \tilde{p}_v \|_2$. Figure 2 shows the resulting histograms for two sets of deformations. In addition, the figure shows a few representative deformed poses, overlayed with their "reconstruction" using the basis functions. As is evident from the figure, the error per vertex is quite small, with a mean value of 0.1% of the size of the bounding box diagonal. In addition, the video accompanying this paper shows a live interaction session, where the projected harmonic map mimics a skinning animation, demonstrating that such an animation can be accurately represented using a harmonic map.

Once we have successfully approximated the source deformation f as a linear combination of harmonic basis functions, the Jacobian of the deformation for any point inside the domain Ω_S can be computed using the gradients of the VHM basis functions (whose expressions are also given in [BCWG09]). The Jacobian of the deformation at a point $p \in \Omega_S$ is:

$$J^S(p) = \sum_{j=1}^{a_S} w_j^S \nabla h_j^S(p)$$

where w_j^S is a column vector, and ∇h_j^S a row vector, both of length 3.

Equipped with the Jacobian of the deformation, we can now define our deformation descriptor, which we will later transfer to the target pose. Since we can compute the Jacobian at any point inside the domain, we have the freedom to choose which Jacobians to transfer. An obvious choice would be to densely sample the source domain, and transfer as many Jacobians as possible. However, this would require a dense correspondence between the source and target volumes, which is, in itself, a difficult problem.

Fortunately, transferring Jacobians from the *entire* volume of the source shape is not necessarily the best approach for deformation transfer, as some of this information may be misleading. Consider the case where the deformation of a bend of a thick bar is transformed onto a thin bar. To accommodate the bend, the Jacobians on the boundary of the thick bar must include a large scaling component, whereas the thin bar can bend using less scaling. This phenomenon was investigated in [LCOG*07]. The medial axis of the bars however, will have similar Jacobians. Hence, it is reasonable to transfer *only* the Jacobians on the medial axis of the source pose to the target pose. Motivated by the same reasons, the VHM shape deformation method requires only the Jacobians on the medial axis of the shape to be rotations, in order to generate an As-Rigid-As-Possible deformation.

To summarize, the deformation analysis stage contains the following steps. First, we generate a polyhedral cage enclosing the source reference pose C_S (we elaborate on the cage generation process in Section 4). Then we project the deformation on the linear subspace of harmonic maps, spanned by the VHM basis functions, to obtain the coefficients w_j^S. Finally, we compute the medial axis of the target cage, using the skeleton extraction algorithm of Au et al. [ATC*08], and sample it. We offer the user this set of samples, as potential landmarks to generate the correspondence to the source reference shape. The user selects the landmarks he is interested in, $\{r_i^T \in \Omega_T \mid i = 1..k\}$, and maps them to points on the source reference shape $\{ r_i^S \in \Omega_S \mid i = 1..k \}$. Our deformation descriptor is the Jacobian matrices of f at these landmarks on the source shape: $J^S(r_i^S)$.

3. Deformation synthesis by harmonic reconstruction

Once the deformation descriptor has been extracted from the source deformation, we can apply it to the target reference pose. We would like to create a deformation of the target reference pose, which interpolates a sparse set of Jacobian constraints. Thus, we reverse the analysis process: first, we project our deformation descriptor on the gradients of the VHM harmonic basis functions on the *target* cage, to find a set of coefficients w_j^T. Then, we compute the deformed pose as a linear combination of the VHM basis functions using these coefficients.

Let the target reference shape be given as a set of n points: $T = \{q_1, q_2, ..., q_n \mid q_i \in R^3\}$, and let $C_T = \{V_T, F_T\}$ be a cage enclosing T with $a_T = |V_T| + |F_T|$. Given the Jacobians $J^S(r_i^S)$ at the landmarks, we would like to solve the following optimization problem:

$$\min_{w_1^T,\dots,w_{a_T}^T \in R^3} \sum_{i=1}^{k} \left\| \sum_{j=1}^{a_T} w_j^T \nabla h_j^T(r_i^T) - J^S(r_i^S) \right\|_F^2$$

where $h_j^T : \Omega_T \to R, j = 1..a_T$ are the VHM harmonic basis functions of the target cage. Unlike in the analysis step, here $a_T \gg k$, hence the problem is under-determined. To regularize it, we use the same method as in VHM, requiring, in addition, that the magnitude of the Hessian of the resulting deformation on a set of sampled points on the boundary of the cage be minimized. In addition, since the gradients determine the deformation only up to a translation, we add one of the landmarks as a single positional constraint. The optimization problem is now:

$$\min_{w_1^T,\dots,w_{a_T}^T \in R^3} \sum_{i=1}^{k} \left\| \sum_{j=1}^{a_T} w_j^T \nabla h_j^T(r_i^T) - J^S(r_i^S) \right\|_F^2 + \lambda \oint_{z \in C_T} H\Big(\sum_{j=1}^{a_T} w_j^T h_j^T(z)\Big) \quad (2)$$

As the gradients and Hessians of the basis functions h_j^T have closed-form expressions, given in [BCWG09], solving this optimization problem for the coefficients w_j^T boils down to solving a linear system of equations. Once we have the coefficients, the deformed target pose is given as a linear combination of the harmonic basis functions, using these same coefficients:

$$\tilde{q}_i = \sum_{j=1}^{a_T} w_j^T h_j^T(q_i)$$

The synthesis step is thus a variant of the VHM deformation method: since we have only gradient constraints, we can avoid the nonlinear step in VHM whose goal is to learn the rotations on the medial axis of the shape, and use a simpler (and more efficient) linear solver.

Figure 3 shows a horse transferred to the model of a dog, and the enclosing cages, and corresponding landmarks. The accompanying video shows the transfer of a time-periodic animation of the galloping horse to the dog.

The dog model has multiple components, but our space deformation method is indifferent to that, and seamlessly transforms the deformation.

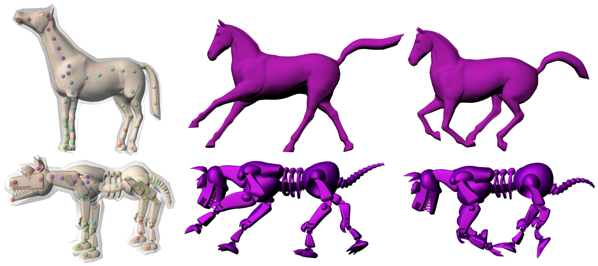

Figure 3: *Transferring the poses of a galloping horse to a multiple component robot dog, using 40 corresponding landmarks. The reference poses inside their cages, and the corresponding landmarks are shown in the leftmost column*

4. Experimental Results

All the steps of our spatial deformation transfer boil down to solving two sets of linear equations – for analyzing the source deformation, and for synthesizing the target deformation. We have implemented all the required linear algebraic computations, using the Intel MKL parallel library, where the multiplication of the large matrices were done on the GPU using the off-the-shelf CUDA BLAS library by NVIDIA. The user interaction was implemented as a Maya™ plug-in.

To evaluate the performance of our deformation transfer method, we have compared it to the original DT method [SP04] which is applicable only to manifold meshes. In this section, and in the accompanying video, we show the results of this comparison. In addition, we show the application of our method to interactive simultaneous deformation of two shapes, and the usefulness of the ability to transfer deformations from and to different shape representations.

First we address some implementation details, which are necessary for using our spatial deformation transfer algorithm.

4.1. Implementation details - Caging

The input to a surface-based deformation transfer application is given by the source and target reference poses, and a deformed source pose. In our method, to apply the space deformation analysis and synthesis we need to envelope the source and target reference shapes with polyhedral cages C_S and C_T respectively.

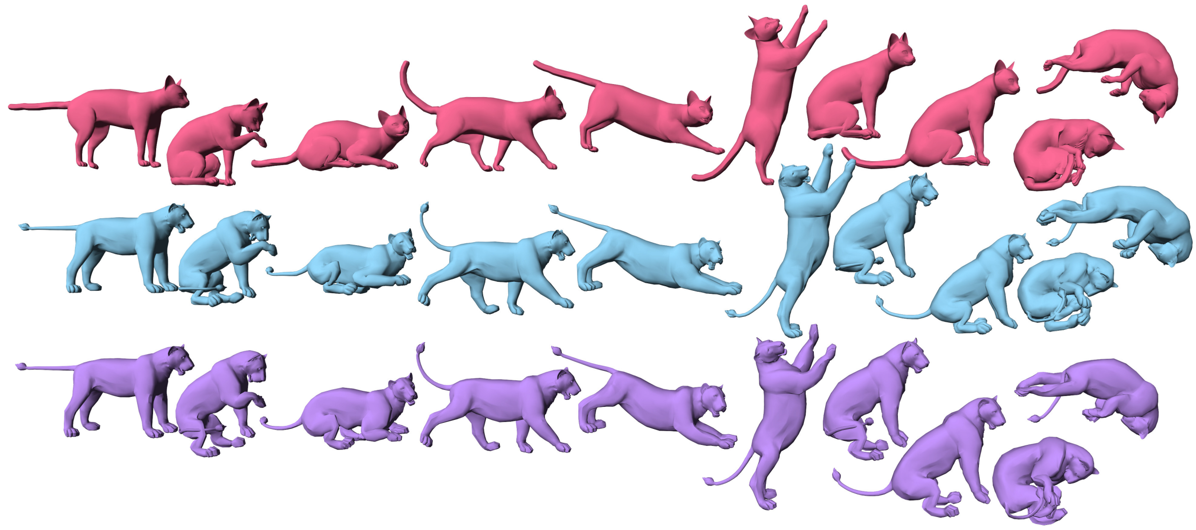

Figure 4: *Comparison of our spatial deformation transfer method, with DT [SP04], on a manifold triangular mesh. (top row) Source poses. (middle row) Result of DT. (Bottom row) Our result. Left most poses on each row are the reference poses.*

Although cage-based deformation methods are quite popular [JSW05, JMD*07, LLCO08, BCWG09], to the best of our knowledge there are no published methods for automatically generating a reasonable cage from an input shape.

The cage generation problem can be posed as follows. Given a set of points $p_i \in R^3$, and a constant integer α, find a closed polyhedron $C = \{V,F\}$, where V are its vertices, and F its faces, which encloses the volume Ω, such that $|F| < \alpha$, $p_i \in \Omega$ for all i, and the volume of Ω is minimal. The volume requirement aims at keeping the cage close to the outer surface of the shape, as this will generally generate pleasing deformations. The upper bound on the number of faces is necessary, since the complexity of the deformation depends on the complexity of the cage. For example, a cage having a few thousand vertices is prohibitive in conjunction with existing deformation methods.

Our cage generation method does not solve the posed problem in the general case, but does provide a good heuristic for automatic cage generation. It proceeds as follows:

1. ***Points and normals.*** Create from the input shape a set of points with normal directions which represent the shape. If the shape is given as a collection of triangles (polygon soup, multiple component mesh or manifold mesh), this can be done by sampling the input triangles, and assigning to each sample the normal to the face it was sampled from. If the input shape is a tet-mesh, the same procedure can be done on the outer surface of the volume mesh.
2. ***Envelope.*** Create an "envelope" E of the input shape, by applying a reconstruction algorithm such as [KBH06] to the points and normals found in step 1.
3. ***Simplify.*** Simplify E, within a tolerance ε.
4. ***Offset.*** Compute an offset position for each vertex of E, by moving it in the normal direction by step size s. The normal direction is computed as the area-weighted average of the normals of the neighboring faces.
5. ***Repeat.*** Repeat steps 2,3,4 until the simplification step achieves the required number of faces.

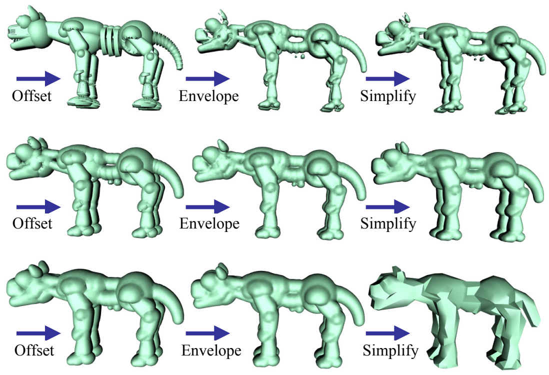

Figure 5: *Cage generation for the robot-dog model. Every row shows one iteration of the offset-reconstruction-simplification steps. The bottom right model is the cage used for the deformation transfer of Figure 3*

Each time we offset the surface of the cage, the geometry of the shape becomes less complex, and it is easier to approximate it with a small number of faces. The ideal way to solve the problem would have been to compute the minimal offset required to approximate the offset surface within tolerance ε, with less than α faces. However, this is a difficult problem and our iterated reconstruction/simplification/offset method simulates this process somewhat. For the cages used in our experiments, a handful of such iterations were sufficient to generate a decent cage.

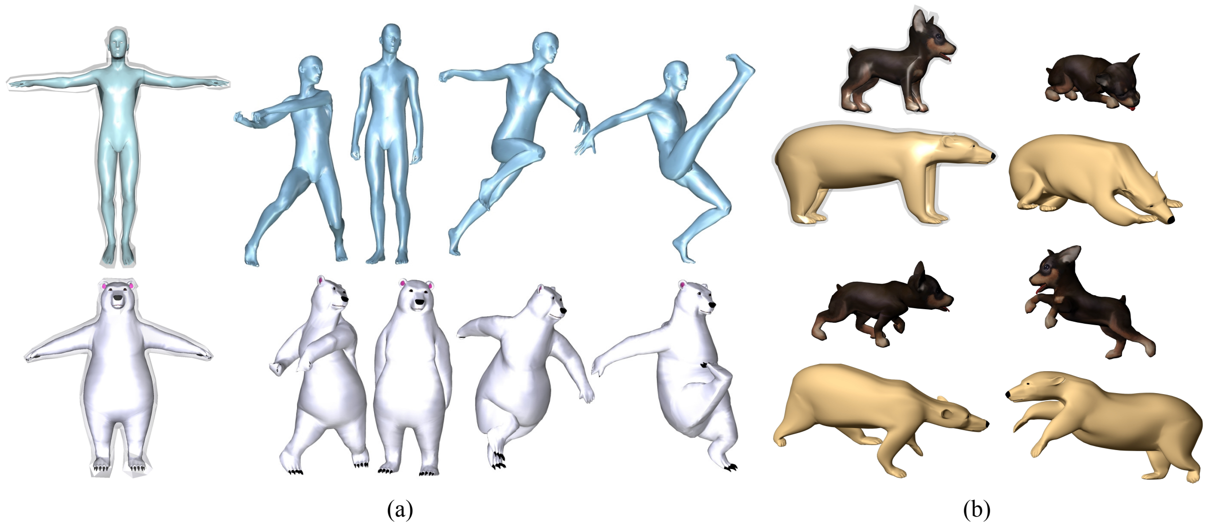

Figure 6: *(a) Transferring deformations of a person to a multiple component polar bear. (b) Transferring deformations of a pincher to a multiple component bear. In both cases the reference poses are shown inside their cages.*

Figure 5 shows a few steps in the cage generation process, as well as the final cage. All of the cages shown in this paper, except the person in Figure 6, and the skeleton cat in the teaser, were generated this way.

In some inputs, however, using the same offset for all points of the model is not the right way to go. For example, the feet of the person in Figure 6 are very close to each other, hence a simple offsetting method will create a cage with fused feet, which is certain to cause artifacts in the resulting deformation. Thus, a better cage generation algorithm should use a varying offset, which might have to be user-defined.

4.2 Deformation Transfer Results

It is natural to compare the performance of our algorithm to that of the original DT algorithm when the source and target meshes are manifold meshes. We state upfront that there are some classes of shapes for which DT's results will be superior to ours. For example, our method generates less pleasing results for facial animations. This is because the source deformation typically modifies the fine details of the surface, and this cannot be approximated well using our harmonic basis functions, without using a prohibitively large number of vertices for the cage.

Figure 4 compares the two methods when transferring a deformation of a cat to a lion. As is evident in the figure, on this manifold mesh our results are comparable to DT. Our method, however, does not require a full correspondence between the meshes (as opposed to the DT method), and a set of only 40 landmarks suffices to transfer the gradient information from the source to the target meshes.

In addition to matching landmarks on the source and target shapes, the user can also create a correspondence between a *line segment* on the medial axis of the source shape (for example, an edge of the extracted skeleton), and a line segment on the target shape. These are then translated to point landmarks by sampling both lines. Figure 6 (a) shows the results of transferring deformations of a person to a multiple component polar bear. The deformation was transferred using only 20 line segments on the medial axis of the shapes, from which 80 corresponding landmarks were sampled.

Figure 6 (b) and the teaser figure show more deformation transfers, also generated by matching corresponding line segments. Note that the pose of the tail of the reference shapes is slightly different in the cat and the skeleton cat. This reflects in all the deformed poses, as they are relative to the reference pose. This phenomenon is common to all deformation transfer methods, including DT.

Our spatial deformation transfer method requires only the solution of linear systems of equations, hence is very efficient. As the accompanying video demonstrates, it is possible to deform a source shape using any standard deformation scheme (such as directly manipulating the vertices, or through a skeleton rig), while *simultaneously* transferring the deformation to a target shape at interactive rates. To give a feel for the times involved, transforming a deformation from the horse to the robot-dog model requires 12 ms per deformed pose. The accompanying video also shows the resulting animations of the skeleton cat and the bear models.

5. Conclusions and discussion

We have presented a method to extend the basic deformation transfer (DT) technique, so that it is applicable to a wide variety of shape representations, beyond single component manifold triangle meshes. We showed how to analyze a given source deformation by projecting it to a linear subspace of harmonic functions, and how to adapt the VHM space deformation method to generate the target deformation from a sparse set of Jacobian constraints. In addition, we have demonstrated that our method is extremely efficient, allowing deformation transfer at interactive rates.

As the original DT method, our approach does not use positional constraints (except one for fixing the translational degree of freedom), hence some artifacts may be present in the resulting target deformation. For example, feet may not touch the ground. This can be rectified, in a somewhat brute-force manner, by adding positional constraints to equation (2) – one constraint for each foot, as we did for the animations of the cat and the bear shown in the accompanying video. However, this might result in artifacts in the temporal domain. A better solution would be to extend VHM so that the deformation transfer problem is solved in the space-time domain, similarly to what has been done in [Gle98] for motion retargeting.

An alternative to spatial deformation transfer by transferring Jacobians, can be to apply the DT [SP04] method to the cages, and then interpolate the deformation to the interior using some barycentric coordinate scheme. This idea was recently proposed in [ZLXP09], where the cages are defined as tetrahedral meshes. An interesting direction for future work would be to apply this approach in the context of deformation with VHM.

Finally, the VHM basis functions might find application in the compression of animation sequences. For example, a deformed shape can be represented using the coefficients of the VHM basis functions, together with the reconstruction errors, if a lossless representation is required. This can greatly reduced the cost of storing a large number of deformed models constituting the keyframes of an animation sequence.

Acknowledgments

We would like to thank NVIDIA for the donation of the graphics hardware, Oscar Kin-Chung Au for providing his skeleton extraction software, Samsung Europe for providing the pincher model and animation, Robert Sumner for the cat, lion, and galloping horse models, Franck Hetroy and Christine Depraz for the black cat model and animation, and Yohai Devir and Michael Bronstein for the person models.

References

[ATC*08] AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., LEE, T.-Y: Skeleton extraction by mesh contraction. *ACM Trans. Graph.* 27, 3, (2008), 1-10.

[BCWG09] BEN-CHEN, M., WEBER, O., GOTSMAN, C.: Variational harmonic maps for space deformation. *ACM Trans. Graph.* 28, 3, (2009)

[BPWG07] BOTSCH, M., PAULY, M., WICKE, M., GROSS, M.: Adaptive space deformations based on rigid cells. *Computer Graphics Forum* 26, 3, (2007), 339-347.

[BSPG06] BOTSCH, M., SUMNER, R., PAULY, M., GROSS, M.: Deformation transfer for detail-preserving surface editing. In *Proc. of Vision, Modeling and Visualization*, (2006), 357-364.

[Gle98] GLEICHER, M.: Retargetting motion to new characters. In *Proc. SIGGRAPH*, (1998), 33-42.

[JMD*07] JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., SANOCKI, T.: Harmonic coordinates for character articulation. *ACM Trans. Graph.* 26, 3, (2007), 71.

[JSW05] JU, T., SCHAEFER, S., WARREN, J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* 24, 3, (2005), 561-566.

[KBH06] KAZHDAN, M., BOLITHO, M., HOPPE, H.: Poisson surface reconstruction. In *Proc. Symposium on Geometry Processing*, (2006), 61-70.

[Kyt95] KYTHE, K., P.: *An Introduction to Boundary Element Methods*. CRC Press (1995).

[LCOG*07] LIPMAN, Y., COHEN-OR, D., GAL, R., LEVIN, D.: Volume and shape preservation via moving frame manipulation. *ACM Trans. Graph.* 26, 1, (2007), 5.

[LLCO08] LIPMAN, Y., LEVIN, D., COHEN-OR, D.: Green coordinates. *ACM Trans. Graph.* 27, 3, (2008), 1-10.

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

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

[SSP07] SUMNER, R. W., SCHMID, J., PAULY, M.: Embedded deformation for shape manipulation. *ACM Trans. Graph.* 26, 3, (2007), 80.

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

[WBCG09] WEBER, O., BEN-CHEN, M., GOTSMAN, C.: Complex barycentric coordinates with applications to planar shape deformation. *Computer Graphics Forum* 28, 2, (2009), 587-597.

[WSLG07] WEBER, O., SORKINE, O., LIPMAN, Y., GOTSMAN, C.: Context-aware skeletal shape deformation. *Computer Graphics Forum* 26, 3, (2007), 265-274.

[XZY*07] XU, W., ZHOU, K., YU, Y., TAN, Q., PENG, Q., GUO, B: Gradient domain editing of deforming mesh sequences. *ACM Trans. Graph.* 26, 3, (2007), 84.

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

[ZLXP09] ZHAO, Y., LIU, X., XIAO, C., PENG, Q.: A unified shape editing framework based on tetrahedral control mesh. *Computer Animation and Virtual Worlds* 20, 2-3, (2009), 301-310.

[ZRKS05] ZAYER, R., RÖSSL, C., KARNI, Z., SEIDEL, H.-P.: Harmonic guidance for surface deformation. *Computer Graphics Forum* 24, 3, (2005), 601-609.