Interactive Shape Manipulation Based on Space Deformation with Harmonic-Guided Clustering

Kai Xu
National University of Defense Technology
{kevin.kai.xu, yanzhen.wang}@gmail.com

Yanzhen Wang
National University of Defense Technology

Yueshan Xiong
National University of Defense Technology
ysxiong@nudt.edu.cn

Zhi-Quan Cheng
National University of Defense Technology
cheng.chiquan@gmail.com

Abstract

We present an efficient and effective deformation algorithm for interactive shape manipulation. To obtain the advantages of both surface- and space-based deformation, we propose to maximally incorporate surface geometry information into space deformation framework while preventing the dependence on surface representation. Our deformation model significantly reduces the problem size through sampling the shape surface and then clustering the sampled points into deformation clusters under the guidance of handle-based harmonic field. The deformation computed on deformation clusters is transferred to the embedded shape through interpolation based on harmonic values. With the surface-base clustering and interpolation, our method enables fine-grained deformation control and provides skeletal deformation without requiring skeleton information. We introduce a method for efficiently evaluating harmonic fields on point clouds. Our deformation model works well with the approximate harmonic fields and hence can handle a wide range of surface representations by sampling them into point clouds.

Keywords: shape manipulation, surface-based deformation, space deformation, reduced model, harmonic field, shape matching

1. Introduction

Recent years have seen the remarkable development of detail-persevering shape deformation techniques. In the most intuitive setting, detail preservation is inherently a nonlinear problem [1].

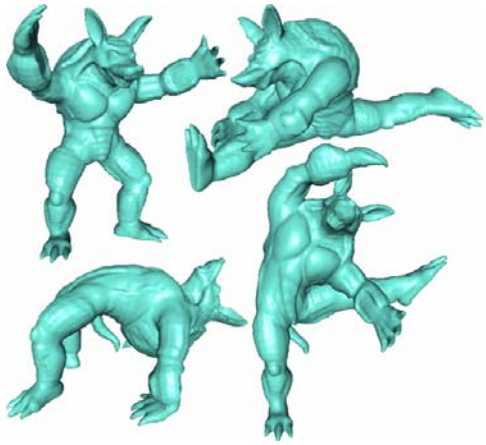

Figure 1: The Armadillo model (left top) is manipulated interactively to take various dance poses with our deformation framework. Only 12 to 16 deformation clusters are used by our method to produce the plausible deformation efficiently.

To achieve scalability, reduced deformation models approximately formulate and solve for the detail-preserving deformation through projecting the problem to a reduced domain. Shape deformation is obtained by interpolating the deformations computed in the reduced domain. The formulation of detail preservation and the interpolation (projection) scheme are crucial to the design of an effective reduced model. Both of them can be done in either a surface-based or a space-based manner.

Surface-based detail preservation is widely employed in designing reduced deformation models. The resultant reduced models inherit the representation-dependent nature of the unreduced surface-based methods. In surface-based methods, spatial interpolation generally

can not provide as fine-grained deformation control as surface-based interpolation.

In space deformation, both detail preservation and interpolation are preformed in a spatial manner. Due to its representation-independent feature, space deformation attracted many attentions lately. However, previous methods have failed to make use of geometry information to reduce of the computation complexity. Besides, space deformation shares the fine-grained control problem.

1.1 Approach and Contributions

To obtain the advantages of both surface- and space-based deformation, we maximally incorporate surface geometry information into space deformation. A harmonic field is first computed based on the user selected handles. In our framework, we utilize the harmonic field to enhance our space deformation in two aspects. First, we build a reduced deformation model through sampling the surface geometry and then clustering the sampled points under the guidance of harmonic field. With this reasonable clustering, the problem size of our space deformation is much smaller than most previous approaches. As a result, our method is very efficient and robust (Figure 1). Second, the deformation computed on the reduced model is transferred to the shape through linear interpolation based on harmonic values. Respecting surface geometry in both clustering and interpolation provides our framework with the ability of fine-grained deformation control like surface-based methods.

To achieve representation-independence, we evaluate harmonic fields on shapes given by other representations. We show that harmonic field can be computed using a meshless method on point cloud models. Other kinds of representations can be handled by sampling the surface into a point cloud.

Efficient and robust. By using harmonic field as the guidance of clustering, our method effectively exploits the spatial coherence in shape matching. The number of unknowns in our nonlinear optimization is linearly related to that of deformation clusters. As a result, our method is very efficient and robust.

Representation-independent. The detail preservation of our space deformation does not depend on the underlying surface representation. With the proposed method for evaluating harmonic field on point cloud, our method can handle many other representations by sampling them into point clouds.

Fine-grained deformation. Our method can provide fine-grained deformation control similar to the surface-based methods through integrating surface harmonic field in both deformation clusters building and deformation interpolation.

Direct manipulation. Our method enables direct manipulation by employing a local shape matching on the manipulated handle. The user can pick any point within the region of manipulated handle and move it around to control the deformation.

2. Related Work

Reduced surface-based deformation. Huang et al. [4] introduce the subspace deformation where the subspace is defined by a coarse control mesh surrounding the original mesh. The positions of control mesh vertices serve as the control variables. Shi et al. [5] build the so-called tetrabones to encode the transformations of mesh skeleton. The control variables, skeleton position and weights for rigging, are computed with a cascading optimization process. Zhou et al. [6] propose the direct manipulation of subdivision surfaces. In their work, the smooth surface can be seen as a reduce model used for computing an initial estimation for the highly nonlinear components of the deformation energy.

Instead of vertex positions, Au et al. [7] use transformations associated to isolines of harmonic fields as control variables. Although more related, our method is quite different with their work in two aspects. First, the method in [7] is surface-based and tightly coupled with mesh representation. Our method adopts space deformation framework and hence facilitates representation-independence.

Space deformation. Space deformation obtain scalability by decoupling the complexity of deformation from the surface complexity. Botsch et al. [8] propose a space deformation method by adaptively discretizing the space into rigid cells. Sumner et al. [9] design the deformation graph to represent space deformation. Both the rigid cells and the deformation graph can be seen as space-based reduced structures. Space deformation often uses transformations as control variables,

which are associated with the control elements, e.g. rigid cells and deformation graph nodes.

One of the main problems of space deformation is the resolution of the control structure must be high enough in order to accurately represent the deformation of the embedded shapes. However, large number of control variables inevitably leads to high computational cost [8] calling for effective models to explore spatial coherence. Rivers and James [11] perform shape matching on regions formed by clustering the lattices [10] to improve their matching performance. While the clustering used by [11] is oblivious to surface geometry, our method clusters the control elements under the guidance of surface harmonic field, which is more reasonable and effective. Space-based interpolation makes the fine-grained deformation control difficult for space deformation methods. Sumner et al. [9] suggest to "cut" the undesirable influence with the user assistant, which is a tedious task for models with complex shape. Our method addresses this problem through surface-based interpolation with harmonic fields.

3. Set-Up and Nomenclatures

3.1 Handle-Aware Manipulation and Harmonic Fields

Handle is a popular metaphor in shape editing. Our framework also employs the handle-based paradigm, where the moved handle is referred to as manipulated handle while all other handles are called fixed handles.

Handle-based surface harmonic field is computed by solving the Laplace equation $\nabla^2\varphi = 0$ under the boundary constraints of 1 for manipulated handle and 0 for fixed handles. One of the most important features of the handle-based harmonic fields is that all the surface vertices with the same harmonic value receive the same degree of transformation propagation from the manipulated handle [12]. This feature motivates us to reduce the complexity of space deformation methods through clustering the control elements according to harmonic values.

3.2 Harmonic-Guided Clustering for Space Deformation

Our method builds the control elements by first sampling the surface and then clustering the sampled points with the same harmonic value into a point set which we call deformation cluster in this paper. We associate to each deformation cluster a transformation **T** (consisting of a rotation **R**, and a translation **t**) as control variable. The embedded shape is deformed through linearly interpolating the transformations of the clusters with harmonic values.

We design two kinds of deformation clusters: isocluster and range-cluster. Isocluster, associated with a single harmonic value φ_{C_j}, is composed of a set (IC) of sampled points with that harmonic value. Range-cluster, associated with an interval of harmonic value $[\varphi_{C_j}^{\min}, \varphi_{C_j}^{\max}]$, contains the sampled points whose value is within that interval. A serial of isoclusters form a "spine" of the embedded shape and produce rubber like deformation. Range-cluster can be used to represent the bone of an articulated shape in implementing skeletal deformation.

Our method uses the linear interpolation:

$$\mathbf{T}_i = \begin{cases} \mathbf{T}_{C_j}, & \exists C_j, \varphi_i \in (\varphi_{C_j}^{\min}, \varphi_{C_j}^{\max}] \\ (1-\alpha)\mathbf{T}_{C_j} + \alpha\mathbf{T}_{C_{j+1}}, & \text{else}, \varphi_i \in (\varphi_{C_j}, \varphi_{C_{j+1}}] \end{cases} \quad (1)$$

where $\alpha = \varphi_i - \varphi_{C_j} \big/ \varphi_{C_{j+1}} - \varphi_{C_j}$ and $\mathbf{T}_i$ is the transformation of vertex *i* with harmonic value φ_i. If a vertex belongs to a range-cluster as the first case in (1), it uses the transformation of that cluster. Otherwise, its transformation is computed by linearly interpolating the transformations of its neighboring clusters. For range-cluster, φ_{C_j} denotes its mean harmonic value.

By sampling the handle regions, handles are also treated as deformation clusters, named manipulated handle cluster and fixed handle cluster. Figure 2 shows the deformation clusters for a bar containing 2 handle clusters and 10 isoclusters.

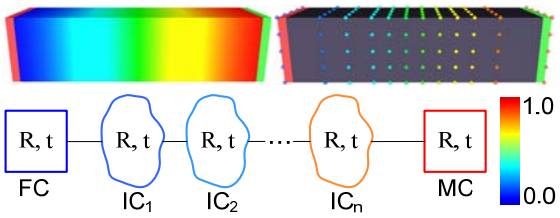

Figure 2 : Deformation clusters on Bar model. Isoclusters (IC), fixed handle clusters (FC) and manipulated handle cluster (MC) are extracted based on the harmonic field (top left).

4. Deformation Energy

In this section, we describe the constraints for detail preservation, transformation propagation, direction manipulation and length preservation. Each constraint is implemented as an energy term of the global deformation energy.

4.1 Detail Preservation

Detail preservation is achieved by ensuring the transformation of each deformation cluster to be close to a pure rotation. In order for a 3×3 matrix **R** to represent a rotation in SO(3), it must satisfy six conditions [13]: each of its three columns must be unit length, and all columns must be orthogonal to one another. Similar to [9], the squared deviation from these constraints for each affine transformation is:

$$\begin{aligned}\mathrm{Rot}(\mathbf{R}) = &(\mathbf{c}_1 \bullet \mathbf{c}_2)^2 + (\mathbf{c}_1 \bullet \mathbf{c}_3)^2 + (\mathbf{c}_2 \bullet \mathbf{c}_3)^2 + (\mathbf{c}_1 \bullet \mathbf{c}_1 - 1)^2 \\ &+ (\mathbf{c}_2 \bullet \mathbf{c}_2 - 1)^2 + (\mathbf{c}_3 \bullet \mathbf{c}_3 - 1)^2 \end{aligned} \quad (2)$$

where $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ are the 3×1 column vectors of **R**. This constraint is nonlinear. The energy term $\mathrm{E}_{\mathrm{Detail}}$ sums the rotation error over all transformations of the n_{C} clusters:

$$\mathrm{E}_{\mathrm{Detail}} = \sum_{i=1}^{n_{\mathrm{C}}} \mathrm{Rot}(\mathbf{R}_{\mathrm{C}_i}) \quad (3)$$

Since the user's control is applied to the manipulated handle, the constraint (2) exerted on the manipulated handle cluster will present larger error than other clusters. In the nonlinear least-squares problem of (3), the residual vector is dominated by this constraint, which we refer to as *manipulated handle constraint* (*MHC*). In order to improve the stability of our system, we put heavier weight on MHC since the rotation guarantee for the manipulated handle cluster is more important than for other clusters. We re-write the energy terms in (3) with preferential weight on MHC:

$$\tilde{\mathrm{E}}_{\mathrm{Detail}} = w_{\mathrm{MHC}} \mathrm{Rot}(\mathbf{R}_{\mathrm{MC}}) + \sum_{i=1}^{n_{\mathrm{IC}}} \mathrm{Rot}(\mathbf{R}_{\mathrm{IC}_i}) \quad (4)$$

where n_{IC} is the number of isoclusters. our experiments show $w_{\mathrm{RHC}} = 1.2 \times n_{\mathrm{IC}}$ is a relatively good choice. Figure 3 shows the comparison between the MHC-preferential weighting scheme of our method and the uniform scheme used by embedded deformation [9]. The Dinosaur model is dragged by a large distance. While the embedded deformation presents visible artifacts near the manipulated handle, our method can robustly deform the model in a nature way.

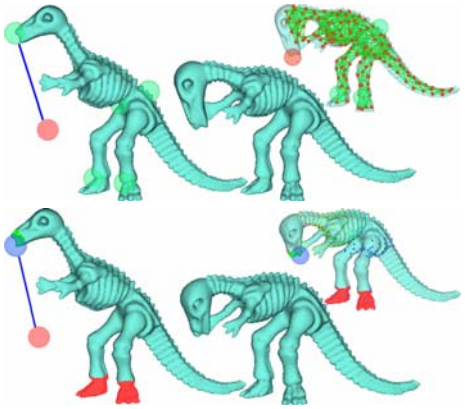

Figure 3: Stability comparison between the uniform scheme of [9] (top) and our MHC-preferential scheme (bottom).

4.2 Transformation Propagation

For any two adjacent deformation clusters, their transformations should be as consistent as possible to each other. This consistency is dedicated to propagate the transformations from manipulated handle to other deformation clusters. To achieve this, we adopt the shape matching technique. For all the points of two adjacent clusters, their positions transformed by one cluster should match those transformed by the other one (Figure 4).

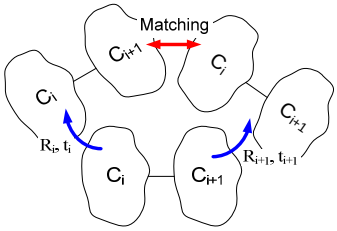

Figure 4: Shape matching on two adjacent clusters, C_i and C_{i+1}.

This matching is performed over all adjacent cluster pairs and introduces the following energy term:

$$\mathrm{E}_{\mathrm{Prop}} = \sum_{(i,j) \in E_{\mathrm{C}}} \int_{\mathrm{C}_i \cup \mathrm{C}_j} \left\| \mathbf{T}_{\mathrm{C}_i} \mathbf{x} - \mathbf{T}_{\mathrm{C}_j} \mathbf{x} \right\|^2 \mathrm{d}\mathbf{x} \quad (5)$$

where E_{C} is the set of all adjacent cluster pairs.

4.3 Direct Manipulation

The user controls the deformation through direct manipulation of the manipulated handle. Any point in the region of manipulated handle can be picked as the manipulated point. We use shape matching [10] to compute correct position and orientation for the manipulated handle

under the positional constraint of manipulated point (Figure 5).

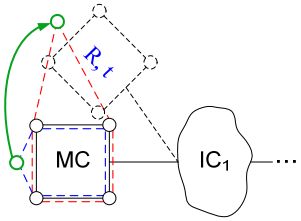

Figure 5: When the user drags a point (the green circle) in region of the manipulated handle, the shape matching matches the new configuration (the red polygon) with the original one (the blue polygon).

This shape matching problem can be stated as follows: Given initial and manipulated positions of points of the manipulated handle, $\mathbf{x}_k^{(0)}$ and $\mathbf{x}_k$. Find the affine transformation for the manipulated handle cluster, $\mathbf{R}_{\mathrm{MC}}$ and $\mathbf{t}_{\mathrm{MC}}$, which minimize:

$$\sum_{k\in MC} w_k(\mathbf{R}_{\mathrm{MC}}(\mathbf{x}_k^{(0)}-\mathbf{c}_{\mathrm{MC}}^{(0)})+\mathbf{t}_{\mathrm{MC}}-\mathbf{x}_k)^2 \qquad (6)$$

where w_k are weights of individual points. To achieve accurate manipulation, the weights are set to 10 for manipulated point and 1 for other points in all of our examples. The optimal translation $\mathbf{t}_{\mathrm{MC}}$ turns out to be the vector from barycenter of the initial configuration, $\mathbf{c}_{\mathrm{MC}}^{(0)}$, to that of the manipulated configuration, $\mathbf{c}_{\mathrm{MC}}$:

$$\mathbf{t}_{\mathrm{MC}}=\mathbf{c}_{\mathrm{MC}}-\mathbf{c}_{\mathrm{MC}}^{(0)}=\frac{\sum_k w_k\mathbf{x}_k}{\sum_k w_k}-\frac{\sum_k w_k\mathbf{x}_k^{(0)}}{\sum_k w_k} \qquad (7)$$

The optimal rotation is found by matching the two configurations in their local coordinate reference frame respectively, which results the following energy term:

$$\mathrm{E}_{\mathrm{Manip}}=\sum_{k\in MC} w_k(\mathbf{R}_{\mathrm{MC}}(\mathbf{x}_k^{(0)}-\mathbf{c}_{\mathrm{MC}}^{(0)})+(\mathbf{x}_k-\mathbf{c}_{\mathrm{MC}}))^2 \quad (8)$$

Minimizing both (8) and (3) simultaneously ensures $\mathbf{R}_{\mathrm{MC}}$ is the desirable rotation.

4.4 Length Preservation

When dealing with humanoid figures, it is particularly useful to preserve the length of models. We add a distance constraint for each pair of adjacent clusters to as follows:

$$\mathrm{E}_{\mathrm{Length}}=\sum_{(i,j)\in E_{\mathrm{C}}}^{n_{\mathrm{C}}}\left(\left\|(\mathbf{c}_i+\mathbf{t}_i)-(\mathbf{c}_j+\mathbf{t}_j)\right\|-l_{ij}^{(0)}\right)^2 \quad (9)$$

where $l_{ij}^{(0)}$ is the original distance of the barycenter of two adjacent clusters *i* and *j*.

4.5 Optimization

Our shape manipulation framework solves for the deformation through minimizing the weighted sum of all the above-stated energy terms, which results the following nonlinear least-squares (NLS) problem:

$$\min_{\mathbf{R}_{\mathrm{C}},\mathbf{t}_{\mathrm{C}}} w_{\mathrm{Detail}}\mathrm{E}_{\mathrm{Detail}}+w_{\mathrm{Prop}}\mathrm{E}_{\mathrm{Prop}}+w_{\mathrm{Manip}}\mathrm{E}_{\mathrm{Manip}}+w_{\mathrm{Length}}\mathrm{E}_{\mathrm{Length}}.$$
$$\text{s.t. } \mathbf{R}_{\mathrm{FC}}=\mathbf{I},\ \mathbf{t}_{\mathrm{FC}}=\mathbf{0} \qquad (10)$$

in which we use w_{Detail}=1, w_{Prop}=10, w_{Length}=20, w_{Manip}=100 for all our examples. The fixed constraint in (10) is handled trivially by treating $\mathbf{R}_{\mathrm{FC}}$ and $\mathbf{t}_{\mathrm{FC}}$ as constants, leaving $12n_{\mathrm{IC}}$+12 free variables. We implement the iterative Gauss-Newton algorithm to solve the unconstrained NLS problem [14]. The number of unknowns is small so the solver can robustly converge within 4 iterations for all of the examples.

5. Skeletal Deformation

In deforming articulated figures, it is more desirable to constraint the parts corresponding to bones of the figures to be unbendable. We propose a simple method to provide skeletal deformation. Our method only needs the user to designate the joint parts (instead of provide the whole skeleton).

Our method is based on our key observation: Skeletal deformation is handle-aware. As was pointed in [7], the gradient magnitude of harmonic field at all vertices forms a rigidity field. Therefore, the joints located at relatively large gradient regions have low rigidity and hence act more effectively than others. We call this kind of joints *effective joints* in this paper. As a result, the skeletal deformation is determined only by effective joints. An effective joint is generally bent along the gradient direction of harmonic field, so it can be modeled with a serial of isoclusters with consecutive harmonic values. Consequently, we can partition shape surface into bone regions and joint regions according to harmonic values.

The user selects several joint regions on shape surface. Our approach first calculates the upper and lower bound of harmonic values for each region. Then the regions are sampled into a serial of isoclusters. The rest parts of the surface will form several range-clusters representing bone regions (Figure 6).

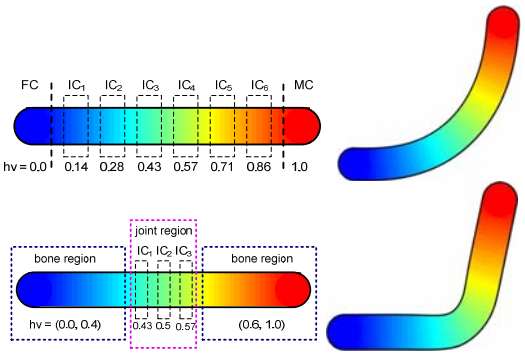

Figure 6: Deformation clusters for "spine" (top) and skeletal (bottom) deformation on a 2D bar.

An important feature of this approach is that it does not introduce new constraint to the optimization and hence is very efficient and easy-implemented. Figure 7 shows the skeletal deformation result on Cylinder model. The user selects a joint part. Our method computes deformation clusters with respect to the selected joint region and provides natural skeletal deformation.

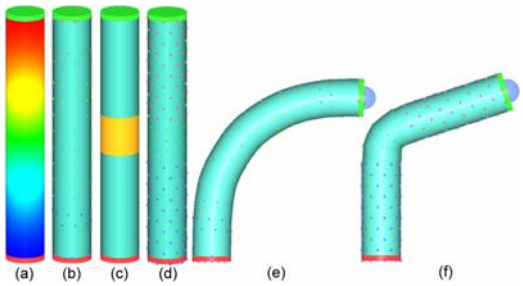

Figure 7: "Spine" deformation (e) vs. skeletal deformation (f) on Cylinder model. The deformation clusters are computed for both "spine" (b) and skeletal (d) deformation with a joint at the central part shown in (c).

6. Representation-Independent Deformation

It is obvious that almost all kinds of geometry, such as polygon soup, point set surface, and even implicit surface, can be easily sampled into a point cloud. For this reason, instead of seeking a method for each kind of geometry, we focus on the evaluation of harmonic field on point-sampled surfaces.

Our deformation clusters are computed by point sampling, which is not sensitive to the accuracy of the harmonic field. As a result, an approximate harmonic field for point-sampled surface is enough for our task.

We propose a simple method which finds approximate 1-ring neighbors for each point of the point-sampled surface and use the found neighbors to evaluate Laplacian approximately. Suppose we have obtained the neighborhood information for each point, a quasi-Laplacian matrix $\hat{\mathbf{L}}$ with uniform weighting scheme is defined as follows:

$$(\hat{\mathbf{L}})_{ij} = \begin{cases} 1 & i = j \\ -\#(\hat{N}_1(i)) & j \in \hat{N}_1(i) \\ 0 & \text{otherwise.} \end{cases}$$

For finding approximate neighbors, we employ the k-nearest neighbors searching together with normal information of each point. Normal is used to remove from the k-nearest neighbors the points which have a reverse orientation with the center point. Belkin [16] has shown that with the sampling rate increasing, this approximate graph Laplacian of point cloud converges to the Laplace-Beltrami operator of the underlying manifold.

7. Results

Besides the direct manipulation, our system also provides a frame-based metaphor to facilitate the regular editing tasks which are hard to accomplish by freely manipulating with mouse, such as the example in Figure 8. This metaphor is quite easy to implement by eliminating the local shape matching in (8) and treating the manipulated handle as a fixed one.

Detail preservation. Figure 1 and 8 demonstrates that our deformation method can preserve features of the shape well. A bumpy plane is edited by fixing the left edge of the place and translating the right edge upward. Although this manipulation is purely translational, our algorithm successfully finds for deformation clusters the transformations which are as close as possible to pure rotations under the positional constraints on the two edges. As a result, the bumps on the plane deform in a natural fashion without shearing artifacts.

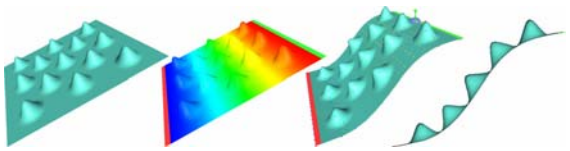

Figure 8: Deforming a bumpy plane.

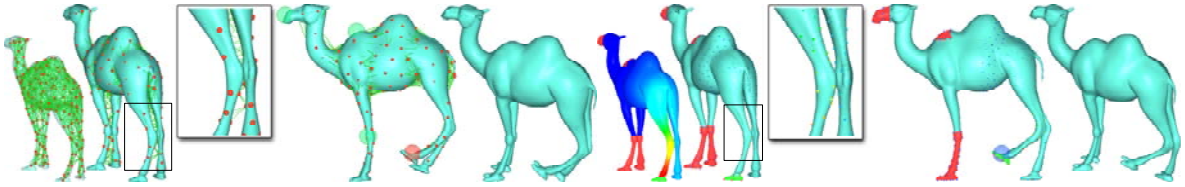

Figure 9: Comparison between our method and the embedded deformation [9] in fine-grained deformation control. The deformation graph (left) of the embedded deformation presents many undesired connections at the two hind legs of the camel (see the zoomed depiction), which can result undesirable inter-influencing among the two legs. Our method, however, avoids this phenomenon and obtains correct results (right).

Fine-grained deformation control. Figure 9 demonstrates our fine-grained control ability through comparing with the space deformation method, embedded deformation [9]. As shown in Figure 9, the two hind legs of the camel are very close to each other. The purely spatial nature of embedded deformation can result in undesired inter-influencing between the two legs, which is shown by the undesired connections among the deformation graph nodes located at two different legs. When the user drags one of the hind legs with the fixed handles shown in the figure, the other hind leg follows the former one undesirably. In contrast, our method nicely resolves this problem by respecting surface geometry in both clusters construction and deformation interpolation.

Efficiency. Table 1 gives the statistics on mesh and deformation clusters, as well as the timing for clusters building and online manipulation, measured in milliseconds on a 2.4GHz Intel Core 2 Duo PC with 2GB of RAM.

Model	#Vert	#C	Build	Solve	Def
Armadillo	172,974	16	1,564 ms	5 ms	26 ms
Armadillo	51,893	16	302 ms	4 ms	8 ms
Buddha	243,652	14	2,659 ms	6 ms	40 ms
Dinosaur	56,194	16	472 ms	5 ms	10 ms
Dinosaur	56,194	100	498 ms	36 ms	16 ms
Dragon	437,645	12	5,035 ms	7 ms	59 ms

Table 1: The column “Build” shows the time statistics for building #C deformation clusters on mesh with #Vert vertices. The column “Solve” gives the time required for each Gauss-Newton iteration. The time for interpolating the deformation of clusters to surface vertices is listed in the column “Def”.

As dictated in the table, our implementation achieves both fast building of reduced model and interactive deformation. The time for building deformation clusters is mainly spent on harmonic field computation. Our method can produce satisfying results with generally 12 to 20 deformation clusters. For the Dinosaur model, we show timing statistics for as many as 100 deformation clusters, which still maintains interactive performance with only 36ms required for each iteration. Since we adopt the linear interpolation, the deformation transferring from clusters to shape is fast.

8. Conclusions

We presented a natural and intuitive deformation algorithm for interactive shape manipulation. By incorporating surface harmonic field information into space deformation framework, our method obtains both efficiency and the fine-grained control feature of surface-based models.

From the current limitations of our system, we see two avenues for future work. First, we will try to propose a multi-resolution method for efficiently evaluating harmonic fields on large-scale models. Second, due to the linear interpolation on transformations, our method shares the linear effect problem with [7] and hence can not provide smooth enough deformation when few deformation clusters are used. We will consider integrating some more advanced interpolation methods, such as the radial basis functions, into our framework.

Acknowledgements

This research was supported in part by 863 program of China (No. 2007AA01Z313), National Natural Science Foundation of China (No. 60773022 and No. 60773020), and Natural Science Foundation of Beijing (No. 4062034).

References

[1] M. Botsch and O. Sorkine. On linear variational surface deformation methods. IEEE Transactions on Visualization and Computer Graphics, 14(1):213–230, 2008

[2] V. Kraevoy and A. Sheffer. Mean-value geometry encoding. International Journal of Shape Modeling, 12(1):29–46, 2006

[3] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. PriMo: Coupled prisms for intuitive surface modeling. In Proc. of Eurographics/ACM SIGGRAPH SGP, pp. 11–20, 2006

[4] J. Huang, X. Shi, X. Liu, K. Zhou, L.-Y. Wei, S.-H. Teng, H. Bao, B. Guo, and H.-Y. Shum. Subspace gradient domain mesh deformation. ACM Transactions on Graphics, 25(3):1126–1134, 2006

[5] X. Shi, K. Zhou, Y. Tong, M. Desbrun, H. Bao, and B. Guo. Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. ACM Transactions on Graphics, 26, 3, Article 81, 2007

[6] K. Zhou, X. Huang, W. Xu, B. Guo, and H.-Y. Shum. Direct manipulation of subdivision surfaces on GPUs. ACM Transactions on Graphics, 26, 3, Article 91, 2007

[7] O. K.-C. Au, H. Fu, C.-L. Tai, and D. Cohen-Or. Handle-aware isolines for scalable shape editing. ACM Transactions on Graphics, 26, 3, Article 83, 2007

[8] M. Botsch, M. Pauly, M. Wicke, and M.Gross. Adaptive space deformations based on rigid cells. Computer Graphics Forum, 26(3):339-347, 2007

[9] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. ACM Transactions on Graphics, 26, 3, Article 80, 2007

[10] M. Müller, B. Heidelberger, M. Teschner, M. Gross. Meshless deformations based on shape matching. ACM Transactions on Graphics, 24(3):471–478, 2005

[11] A. R. Rivers and D. L. James. FastLSM: Fast lattice shape matching for robust real-time deformation. ACM Transactions on Graphics, 26, 3, Article 82, 2007

[12] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. Computer Graphics Forum, 24(3):601–609, 2005

[13] F. S. Grassia, Practical parameterization of rotations using the exponential map. J. Graph. Tools, 3, 3, 1998

[14] K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark. 2004

[15] U. Clarenz, M. Rumpf, and A. Telea. Finite elements on point based surfaces. In Proc. of Symposium on Point-Based Graphics 04, pp. 201–211, 2004

[16] M. Belkin, and P. Niyogi. Towards a Theoretical foundation for Laplacian-based manifold methods. Journal of Computer and System Sciences, Special Issue on Learning Theory, to appear, 2008