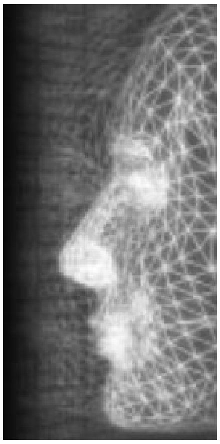
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2004; **15:** 201–210 (DOI: 10.1002/cav.22)

Robust morphing of point-sampled geometry



By Chunxia Xiao, Wenting Zheng, Qunsheng Peng and A. R. Forrest*

We propose a novel morphing algorithm for objects represented by point-sampled geometry. The fundamental problem of point-sampled geometry morphing is how to set the correspondence between points of the two objects which are usually of different size. The two objects are first parameterized by projecting the sample points onto a common parametric domain. As both objects are densely sampled, we present a novel accelerated parameterization algorithm employing the technique of LOD. The common parameter domain is then split recursively into clusters. The correspondence between sample points of the two objects is established by performing a local mapping in each cluster. As for complex geometries, the establishment of correspondence is facilitated by decomposing the geometry into patches using geodesic decomposition curves.

To preserve the features during morphing, a process of features assignment is incorporated. By re-sampling the in-between object dynamically and adaptively, the cracks that would occasionally occur during morphing are successfully eliminated. Experiment results show that our algorithms are fast, stable and easy to implement. High-quality morphing is produced. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: point-sampled geometry; morphing; covariance analysis; dynamic sampling decomposition; parameterization

Introduction

As an effective graphic primitive for modeling and rendering of highly complex sculptured objects, point primitives have experienced a major 'renaissance' in recent years, considerable research has been devoted to the efficient representation,[1,2] modeling,[3] processing,[4–6] and rendering of point-sampled geometry.[1,2,6] These efforts have spawned a new field called point-based computer graphics.

Morphing of point-sampled geometry is one important research area in point-based computer graphics. Morphing techniques aim at transforming a given source object into a target object. Morphing techniques have various applications ranging from special effects in television and movies to medical imaging and scientific visualization. Since meshes are the most popular representation form of models in computer graphics, not surprisingly, morphing between objects represented by meshes, has received a lot of attention.[8] Kent *et al.*[9] introduced the idea of topological merging as a way of establishing correspondence for a wide class of genus-0 polyhedra. Related studies were also conducted on how to decompose the model into patches each homeomorphic to a disk,[10,11] and on spherical parameterization.[12–14] For a good survey on mesh morphing, readers are referred to Alexa.[8]

Comparatively little work has been carried out on morphing of point-sampled geometry. Cmolik[15] presented work on point cloud morphing. Firstly, two BSP (binary space partition) trees are created by clustering the source model and the target model. In the next stage the clusters in the trees are assigned to each other depending on a certain chosen metric function. Morphing is then performed. The main problem with this algorithm is the occurrence of cracks whilst dealing with the non-convex models.

Point cloud is an unstructured set of point samples. Each point sample is specified by its location in 3D space, normal vector, color and size. Unlike most of other surface representations, these discrete points do not possess any continuity or belong to certain

*Correspondence to: Chunxia Xiao, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027 P. R. China. E-mail: cxxiao@cad.zju.edu.cn homology groups. It is therefore difficult to build the correspondence of points between two models. Nevertheless, working on point clouds offers a number of advantages.

Compared with mesh morphing, no complex data structure addressing surface continuity has to be built and maintained during the process of parameterization and correspondence assignment. For mesh morphing it is troublesome to check if the parameterization is valid (check if the orientations of the faces are legal), and to check if it will introduce foldovers during morphing. Since point-based morphing is independent of the topology of the models, it is never affected by these problems. Thus the algorithms for morphing point-based geometries are time and memory efficient and easier to implement than those for mesh morphing.

In this paper, we propose a novel morphing algorithm for objects represented by point-sampled geometry. The fundamental problem of point cloud morphing is how to set the correspondence between the points of the source object and the target object. Since the number of sample points of the two models may be greatly different, it is certainly not a trivial task to establish the correspondence. Cmolik[15] set the correspondence based on space partitioning. However, his method is not applicable to high-quality morphing of non-convex shapes nor for feature preserving morphing. In contrast to[15], our approach projects the sample points of the two objects onto a common parameter domain. These projection points are then clustered within the unified parametric domain. The correspondence between point sets of the two objects is then established by performing a local mapping in each cluster. For objects with complex geometries, we first decompose the geometry into several patches then build the correspondence between each pair of patches.

Point-sampled geometry is usually densely sampled. The number of points in each geometry is normally very large. The parameterization algorithm proposed in Floater[18] is too slow to deal with such large-scale data point models. A novel algorithm for accelerating parameterization is presented in this paper employing the level of detail (LOD) technique. In addition, we incorporate a features assignment process for point-sample geometry to preserve features during the morphing.

Due to the variation of the local sampling density during morphing, some small cracks might occasionally occur. To eliminate the cracks, a dynamic and adaptive re-sampling method is proposed. Experimental results show that our algorithms produce fast and high-quality morphing.

The rest of the paper is organized as follows: Section 2 proposes an efficient method for parameterizing a point-sampled patch. Details of our morphing algorithms regarding two point-sampled patches are presented in section 3. In section 4 we show how to deal with complex models, while in section 5 we demonstrate the experimental results. Conclusions and some topics of future work are given in section 6.

Parameterization of Point-sampled Geometry

Parameterization of point-sample geometry is the fundamental step of our morphing algorithm. As preliminaries, in the following, we first give a brief introduction to covariance analysis.

Covariance Analysis

Covariance analysis can be used to estimate various local surface properties, such as the normal vector of the approximation surface or surface variation. The covariance matrix C of the point cluster P is defined as

$$C = \begin{bmatrix} p_{i_1} - \bar{p} \\ p_{i_k} - \bar{p} \end{bmatrix}^T \cdot \begin{bmatrix} p_{i_1} - \bar{p} \\ p_{i_k} - \bar{p} \end{bmatrix}$$

where $\bar{p}$ is the centroid of point set P. Since C is symmetric and positive semi-definite, all eigenvalues $\lambda_i (i = 0, 1, 2)$ are real-valued and the eigenvectors $v_i (i = 0, 1, 2)$ form an orthogonal basis. The eigenvalues λ_i measure the variation of the point set along the direction of the corresponding eigenvectors.

If we assume that $\lambda_0 \leq \lambda_1 \leq \lambda_2$, the plane $(x - \bar{p}) . v_0 = 0$ minimizes the sum of squared distance from the neighbors of $\bar{p}$ to the plane, if P is the set of points representing the surface, then the normal v_0 of this plane approximately represents the normal of surface at $\bar{p}$. The eigenvalue λ_0 expresses in this particular case the variation of the surface along the normal v_0, or in other words, it estimates how much the points of surface deviate from the tangent plane. For a more detailed description, see[16].

Fast Parameterization Algorithm

Extending previous work on parameterization of triangle meshes, Levy,[17] Zigelman,[7] Zwicker[5] designed a novel method for distortion minimal distortion parameterization of point clouds. Floater[18] also proposed a method called meshless parameterization for unorganized point sets of a 'single patch'. The points are projected onto a planar parameter domain by solving a global sparse linear system, whose equations arise from demanding that each interior parameter point be a convex combination of some neighboring points. As our approach is an improvement to Floater's method, we briefly review meshless parameterization.

Given a sequence of distinct points $X = (x_1, \dots, x_N)$ in $\mathbb{R}^3$, which are assumed to be sampled from a patch of some unknown surface in $\mathbb{R}^3$, we assume that the set X can be split into two disjoint subsets: $X_I = \{x_1, \dots x_n\}$, the set of interior points, and $X_B = \{x_{n+1}, \dots, x_N\}$, the set of boundary points, where the points $x_{n+1}, \dots, x_N$ are ordered consecutively along the boundary, then parameterization is accomplished by two steps.

In the first step the boundary points $x_{n+1}, \dots, x_N$ are mapped into the boundary of some convex polygon D in the plane, by choosing the corresponding parameter points $u_{n+1}, \dots, u_N$ to lie around ∂D in some anticlockwise order, for example let $u_{n+1}, \dots, u_N$ to lie on unit circle and determine the distribution of $u_{n+1}, \dots, u_N$ along ∂D by some standard polygonal parameterization, such as uniform or chord length.

In the second step, a neighborhood $N_i = \{j : 0 < \|x_j - x_i\| < r\}$, a set of points in $X \backslash \{x_i\}$, is chosen for each interior point $x_i \in X_I$. Then we choose a set of strictly positive weights $\lambda_{i,j}$, for $j \in N_i$, such that

$$\sum_{j \in N_j} \lambda_{ij} = 1 \qquad (1)$$

then n parameter points $u_1, \dots, u_n \in \mathbb{R}^2$ corresponding to the interior points $x_1, \dots, x_n \in \mathbb{R}^3$ can be found by solving the linear system of n equations

$$u_i = \sum_{j \in N_i} \lambda_{i,j} u_j, \quad i = 1, \dots, n. \qquad (2)$$

The linear system (2) may be written in the form $Au = b$, where $A = (a_{i,j})$ is the square $n \times n$ matrix with $a_{ii} = 1$ and $a_{ij} = -\lambda_{ij}$ for $i \neq j, u$ is the column vector $(u_1, \dots, u_n)^T$, and $b = (b_1, \dots, b_n)^T$ is the column vector with

$$b_i = \sum_{j=n+1}^{N} \lambda_{ij} u_j. \qquad (3)$$

Floater[18] derives a weak sufficient condition under which the linear system (2) is uniquely solvable. Unfortunately, models represented by point clouds contain usually thousands, possibly millions of sample points, and it is therefore time-consuming to solve the linear system $Au = b$. Furthermore, the huge memory cost for the storage of the non-zero terms a_{ij} of a very large sparse matrix A is also unacceptable. So to parameterize such dense surfaces, we have to find a faster and more effective parameterization approach.

The main idea of our approach is to simplify the densely sampled patch O into a sparsely sampled patch S, then the meshless parameterization method is applied to parameterize S into an unit disk. The point samples of original model are embedded into the corresponding position in the disk. Our fast algorithm is accomplished in three successive steps:

1. We generate a simplified model S of the original model O by Hierarchical Clustering.[16] The point cloud O is split recursively into clusters along the direction of greatest variation until the size of each sub-cluster is less than a given threshold. The normal vector of the split plane can be defined by the centroid of current cluster and the largest eigenvector of its covariance matrix (which is v_2). O is now split into clusters $\{Ci\}$. Each cluster C_i is replaced by its centroid y_i. Suppose S consists of the point set $Y = (y_1, \dots, y_t)$ and $Y_I = (y_1, \dots, y_s)$ is the set of interior points, $Y_B = (y_{s+1}, \dots, y_t)$ is the set of boundary points.
2. We perform meshless parameterization to parameterize the simplified model S, and get a set of projection points $V = (v_1, \dots, v_t)$ on the unit disk.
3. We calculate the parameter for each original sample point $x_k \in C_i$ by convex interpolation. First we find neighbors $N_i = \{j : 0 < \|y_j - y_i\| < r\}$ for each interior point $y_i \in Y$. And locate y_{i1}, y_{i2}, y_{i3} in N_i which are the nearest to x_k. Suppose y_{i1}, y_{i2}, y_{i3} have corresponding projections v_{i1}, v_{i2}, v_{i3} in the unit disk. Let λ_{ij} be the reciprocal distance weights $\lambda_{ij} = (1/\|x_k - y_{ij}\|) / \sum_{m=1,2,3} (1/\|x_k - y_{im}\|)$, and then x_k have a corresponding point $u_k = \sum_{j=1,2,3} \lambda_{ij} v_{ij}$ in the parameter domain.

The ordered boundary points can be acquired by the methods proposed by Floater[18] or by user interaction. Using a 3D grid data structure, the neighborhood can be computed efficiently in constant time. In our approach we use the reciprocal distance weights for the linear system (2). Figure 1 shows the parameterization step of our method.

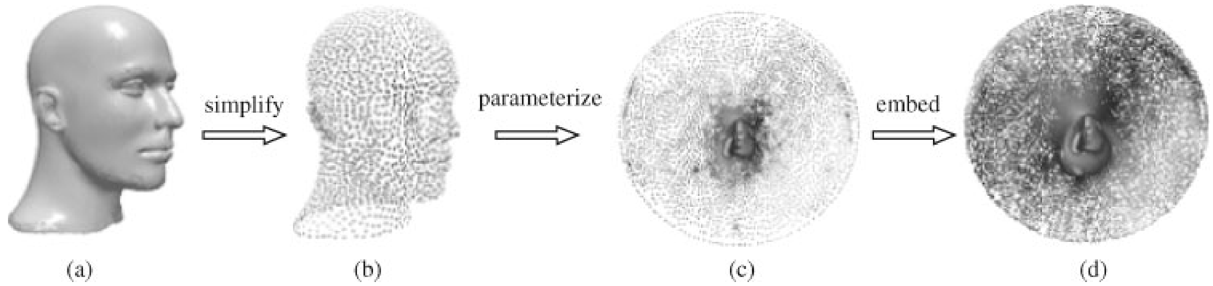


Figure 1. Fast parameterization algorithms. (a) Original model with 98,250 points, (b) The model is simplified to 8,856 points, (c) Parameterization of the simplified model to a disk (d) Parameterization of the original model.

By simplifying the original patch, the size of the matrix A is greatly reduced. So compared to[18], our algorithm is superior in both time and space complexity.

Points-sampled Geometry Morphing

Commonly, there are two problems that must be solved for morphing between source object and target object. The first is how to establish the correspondence between sample points on the two objects. The second is how to create the morphing paths for each pair of corresponding points. In general, the first problem is more difficult to tackle.

Our approach establishes the correspondence between two models in two steps. The first step is the alignment of corresponding features on the two objects. In the second step, the two parameter domains are merged into a common parameter domain to let all the correspondent features coincide. The merged parameter domain is split into clusters. In each cluster, mapping of points between two models can be set up.

Finally, a morph sequence is obtained by interpolating the positions of two mapped points.

Feature Alignment

Alexa[12] addressed the problem of feature alignment for genus-0 polyhedron mesh models on a unit sphere. Similarly to mesh morphing, it is necessary to align prominent features between the source object and the target object represented by point-based geometry. By projecting the corresponding feature points to the same position on the common parameter domain, we can easily produce a morphing sequence demonstrating a natural transition of features of both objects.

A polygonal mesh is generally described by its geometry and connectivity. As the connectivity between adjacent triangles is kept during the morphing, no cracking will happen. Nevertheless, point-sampled geometry is represented by discrete points and no connectivity information is preserved. The occasional occurrence of cracks on the surface during the morphing (see Figure 4 (a)) becomes a problem.

Suppose two embeddings with points U^1, U^2 on the unit disk, and two ordered sets of features which are represented as a number of indices in the set F^1, F^2 respectively (we assume $|F^1| = |F^2|$). We denote the *ith* element of the point sets as u_i^1, u_i^2, and the *ith* element of the feature sets as f_i^1, f_i^2. The goal of feature alignment can be written as $\forall i.u^1_{f_i^1} = u^2_{f_i^2}$.

Our algorithms are similar to Alexa[12] with some differences. Face foldover need not be checked in point-based morphing and the features are aligned on a unit disk not sphere.

An overview for the algorithm is given as follows:

1. Rotate the first disk so that the summed squared distance $S = \sum_i \|u^1_{f_i^1} - u^2_{f_i^2}\|^2$ is minimized.
2. Define $p_i^1 = u^1_{f_i^1} - u^2_{f_i^2}$ regarding the ith feature of f_i^1 then apply the following mapping to all points u_j^1 on U^1

$$\forall j.u_j^1 = \begin{cases} u_j^1 + p_i^1(d - \|u_j^1 - u^1_{f_i^1}\|)/(2d) & \|u_j^1 - u^1_{f_i^1}\| < d \\ u_j^1 & \|u_j^1 - u^1_{f_i^1} \geq d \end{cases} \tag{4}$$

   where d is the radius of influence of each feature point on the parametric domain. Apply the above process to all features f_i^1.
3. Perform step 2 with respect to all features of f_i^2.
4. Repeat steps 2 and 3 if $\max_i \|u^1_{f_i^1} - u^2_{f_i^2}\| > error$.

Note that features alignment may cause the occurrence of cracks during morphing. Assume that the original projection of f_i^1 on the parametric domain in unit disk is moved according to (4). If d is larger than $p_i^1/2$, all points u_j^1 which are located within the circle of influence will remain inside of the circle of influence after features alignment, resulting in a one-to-one map between any sample point x_j and its projection on U^1. But the density of the projection points may be extremely non-uniform when d/p_i^1 is close to 1/2. If d is less than $p_i^1/2$, the wrapped influence circle will overlap the same area outside the original influence circle. Either of the last two cases may cause cracks.

In order to eliminate cracks, we would like to take a larger d. As shown in Figure 2, a larger d works effectively. The techniques for how to eliminate the cracks will be discussed further.

Point Pairs Assignment

In this section we discuss how to set the correspondence between other sample points of the source object and the target object.

As is known, the actual morphing is the position transition between assigned point pairs (s_i, t_i) $s_i \in S$ and $t_i \in T$ where S is the point set of the source model and T is the point set of the target model. In reality *S and T* might contain different numbers of points, and one point from the source set might be assigned to multiple points in the target set and vice versa.

To find the correspondence between points of the two objects, point set U^1, U^2 are merged into point set U, a unified parameter domain. After the process of feature alignment, the corresponding features should coincide in U. We then split U into clusters, such that the number of the projection points from either object in each cluster does not exceed a threshold but includes at least one point. Mapping of points between S and T is performed cluster by cluster.

Unit U can then be hierarchically clustered based on covariance analysis. Each cluster C_i consists of set $A_i = (a_1, a_2, \ldots, a_k)$ and $B_i = (b_1, b_2, \ldots, b_l)$, $A_i \subset S, B_i \subset T$ and $k \geq 1, l \geq 1$. The size of C_i should be less than a threshold σ, for example $\sigma = 10$.

The next step is to yield a mapping for points within cluster C_i. The mapping generates the set of m assignments (a_i, b_j), where $m = \max(k, l)$.

Assume that $k > l$, the mapping can be described as surjective mapping $\varphi : A \rightarrow B$

The metric error can be expressed as $\sum_{i=1}^{k} |\varphi(a_i) - a_i|^2$. A good mapping is to minimize the metric error and run fast.

We propose an approximate minimal metric error algorithm as follows. Assume $k > l$, for each point $b_i \in B$, we find a corresponding point $a_i \in A$ which is not marked and nearest to b_i, and then mark it. Assignment (a_i, b_i) is built. Then for each unmarked point in A, a corresponding point is found in B which is nearest to it. The point couple is also added to assigned point pairs. As n is usually small (for example, if $\sigma = 10$, then $n < 9$), this approach is fast and acceptable in metric error. By applying this algorithm to all clusters, the point couples for both models are acquired.

Morphing Path

Once we have established the correspondence between points of the source object and the target object, the problem of morphing between two complicated shapes is reduced to a trivial problem of morphing between the assigned point pairs (s_i, t_i). The coordinates of points of transition for the point cloud can be determined as below

$$path_i(t) = f(s_i, t_i, t) = (1 - t)s_i + t \cdot t_i, \quad t \in (0, 1) \qquad (5)$$

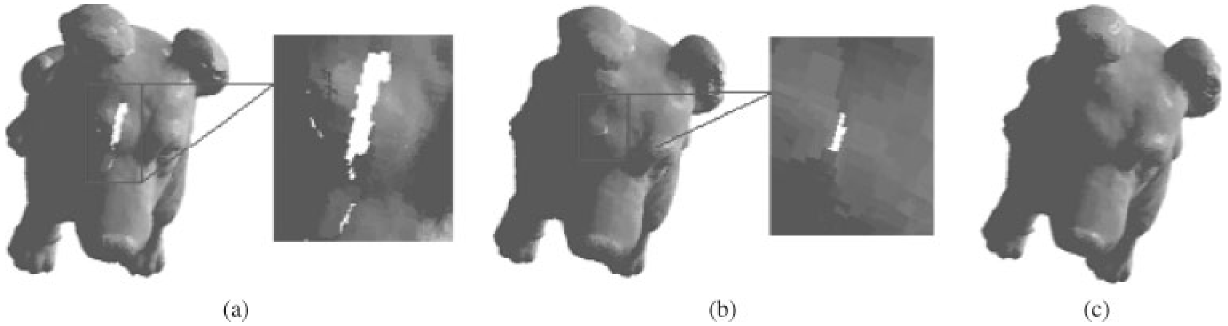


Figure 2. Interpolate the corresponding points at $t = 0.4$, (a) A relatively large cracks with $d = 2\|p\|$, (b) A small cracks with $d = 5\|p\|$, (c) No cracks with $d = 8\|p\|$.

We can also apply (5) to other properties of the point samples, such as color, normal, transparency or size. Note that the interpolated normal vector should be normalized during morphing. With the assignments M and morphing path $path_i$ we can perform morphing between two different-sized point -sampled geometries.

Dynamic Re-sampling

In this section we focus on preventing the occasional occurrence of cracks during morphing.

Above, a larger d is adopted to reduce the cracks. However, due to the projection distortion resultant from parameterization or the non-uniform mapping introduced by features alignment, some small cracks may still occur. To eliminate the small cracks, we need to detect the regions with insufficient sampling density, then insert new sampled points.

Alexa *et al.*[6] proposed a point-sampled geometry representation by fitting a local polynomial approximation to the point set using a moving least squares method (MLS). The result of the MLS-fitting is a smooth, 2-manifold surface for any point set. They then used MLS-surfaces to dynamically up-sample the point set to obtain a high quality smooth surface.

Assume that the point-sampled geometry consists of an unstructured point cloud $P = \{p_i = (x_i, y_i, z_i) \, |1 \le i \le n\}$. The continuous MLS surface S is defined implicitly as the stationary set of a projection operator $\psi(p,r)$ that projects a point onto the MLS surface

$$S = \{x \in \mathbb{R}^3 | (\Psi(p,x) = x\} \tag{6}$$

To evaluate $\Psi(p,r)$ we first compute a local reference plane

$$H = \{x \in \mathbb{R}^3 | x \cdot n - D = 0\} \tag{7}$$

by minimizing the weighted sum of squared distances

$$\sum_{p \in P} (p \cdot n - D)^2 \phi(\|p - q\|), \tag{8}$$

where q is the projection of r onto H and ϕ is the MLS kernel function. After transforming all points into the local frame defined by H, a second least squares optimization yields a bivariate polynomial $g(u,v)$ that locally approximates the surface. The projection of r onto S is then given as $\psi(p,r) = q + g(0,0) \cdot n$ (for more details see[6]). The characteristics of the surface can be controlled by the kernel function ϕ. Typically, a Gaussian $\phi(t) = \exp(-t^2/h^2)$ is chosen, where h is a global scale parameter that determines the feature size of the resulting surface. Rather than following Alexa,[6] to get a stable solution to (7), we use the Powell iteration with the initial values n, t; where $t = 0$, and the normal v_0 generated by covariance analysis is regarded as a good initial value for n.

Up-sampling can be also considered as a reconstruction technique since it prevents the appearance of cracks by assuring a sufficiently high density of points, as shown in Figure 3. A problem associated with this approach is that point samples have to be generated dynamically during morphing.

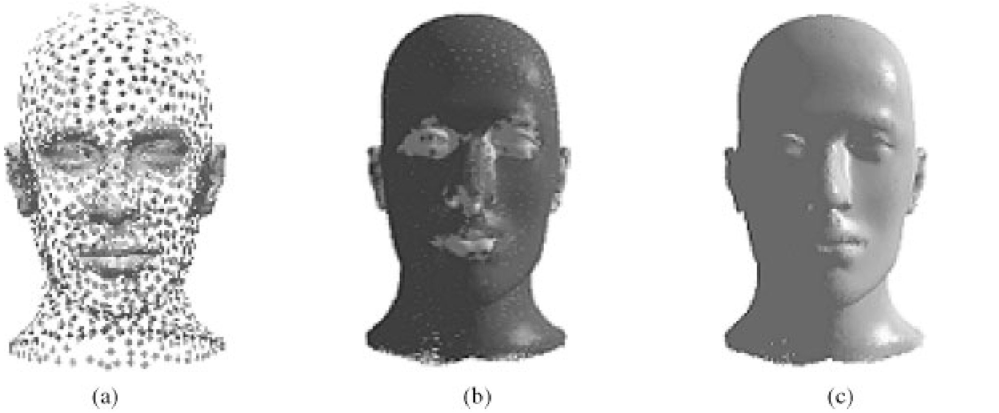


Figure 3. MLS up-sampling of the Manhead. (a) Input point-based surface, (b) Up-sampling the regions with insufficient sampling density, (c) reconstructed MLS surface.

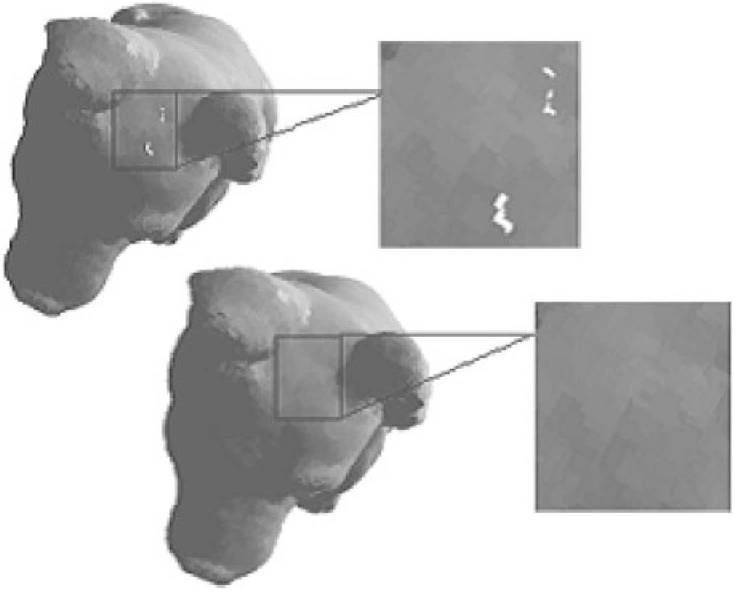


Figure 4. Top, surface with cracks at $t = 0.5$, Bottom, eliminating the small cracks by up-sampling the surface.

To detect regions of insufficient sampling density dynamically, we estimate and record the local sampling density for each point of the source and target object. We estimate the local sampling density ρ_i for each $p_i \in P$ by finding the sphere with minimum radius r_i centered at p_i that contains the k-nearest neighbors to p_i. Then ρ_i is defined as $\rho_i = k/r_i^2$. According to (5), let $path_i(t)$ be a point in the transition point cloud, ρ_i is the sampling density of $path_i(t)$. Let ρ'_i be the sampling density of s_i, ρ''_i be the sampling density of t_i. Let Ω be the threshold value for up-sampling, if $\rho_i < ((1 - t)\rho'_i + t \cdot \rho''_i) * \Omega$, then new sample points must be inserted.

To up-sample the surface, firstly, a local linear approximation is built and points nearby are projected onto the plane. A bounding rectangle is established for the projected points. Points are uniformly re-sampled in the rectangle according to a user specified threshold. The MLS surface is constructed and the re-sampled points are projected onto it to get the final up-sampling result. The normal at each sample point can be obtained directly by evaluating the gradient of the polynomial. As shown in Figure 4, the up-sampling eliminates the cracks.

Morphing of Complex Models

In previous sections we present a morphing technique for point-sampled geometry between two surface patches. In this section, we will extend the techniques to dealing with complex point models.

Considering two objects of genus-0, initially, the user selects some feature on the geometry and these features compose an ordered sequence. The geodesic curves of the geometry between the sequenced features are computed.[19,20] These geodesic curves make up a closed decomposition curve on the surface. The genus-0 geometry can then be decomposed into two patches, as shown in Figure 5(b). The points on the decomposition curve automatically generate the ordered set of boundary points which is demanded by the parameterization step. The set of interior points of each patch is acquired by applying the algorithms based on a level set method.[19] The surface decomposition operation is applied to both objects. The correspondence between patches is established by user interaction. The methods described in previous sections are then applied. An example of morphing genus-0 objects is shown in Figure 9.

In a similar way, the homeomorphic non-simple point-sampled geometries that are not genus-0 can be handled.

Implementation and Results

We have implemented our algorithms in VC++6.0 on a PC with PIII 1.0G, 512M memory. Table 1 shows the number of iterations and CPU time for each of the following two component systems:

$$Au^1 = b^1, \quad Au^2 = b^2 \tag{9}$$

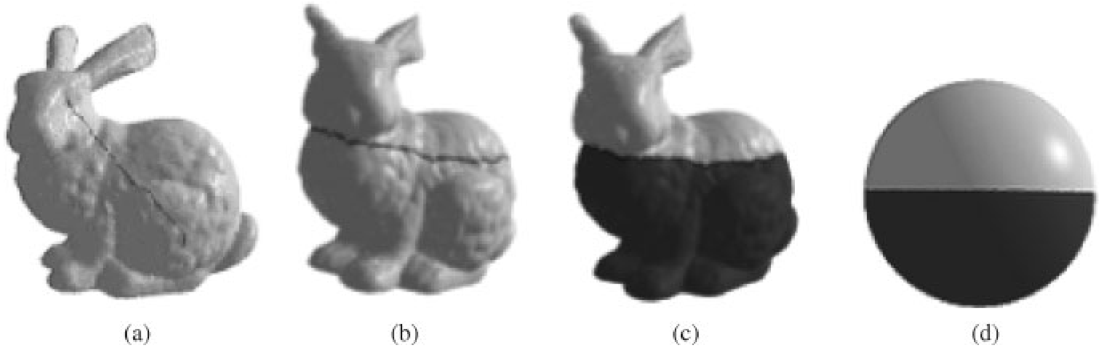


Figure 5. (a) Geodesic on the surface, (b) Generating the decomposition curve, (c), (d) Patch selection, the blue patch in (c) corresponds to the blue patch in (d).

Models	Num. *F*	#(O)	#(S)	Num. Iterations.	CPU time (Sec.)
Venus\\Manhead	5	40818\\109450	40818\\10580	1920/1890\\250/231	180.41/178.21\\11.11/10.08
Dog\\Cat	5	98250\\84348	10814\\12513	220/251\\260/281	10.11/15.08\\13.11/15.08
Isis\\Rabbit	4	118250\\34234	11214\\34234	243/231\\1685/1599	12.11/11.01\\161.41/158.31

Table I. Performance of our parameterization method based on simplified model. Num. *F* is the Num. of features, #(O) is the Num. of points of both original models, #(S) is the Num. of points of their respective simplified model. The Num. Iterations and CPU time of the two component systems iterations for each simplified models are illustrated

Figures 6 to 8 show the morphing sequence of models with sophisticated boundary surfaces. No crack appears in Figures 6 and 7. In Figure 8, little cracks occur during morphing, the detail of the local area is shown in Figure 4(a). These cracks are eliminated by dynamically up-sampling the surface.

Conclusions and Future Work

Many investigations have focused on meshes morphing, whilst research on point-sample geometry morphing is in its original phase. In this paper, we have presented a novel morphing method for objects represented by point-sampled geometry. By mapping the sample points of each object onto a unit disk then merging the two disks, the correspondence between points of the source model and the target model can be established. The occasional little cracks during morphing are eliminated by up-sampling the surface derived by MLS methods. Our methods are fast, robust, easy to implement, and can deal with complex point clouds.

A closed genus-0 point-sampled manifold is topologically equivalent to a sphere. The sphere is the natural parameter domain for them. There has been much work on spherical parameterization of 3D-meshes.[11,14,21] In

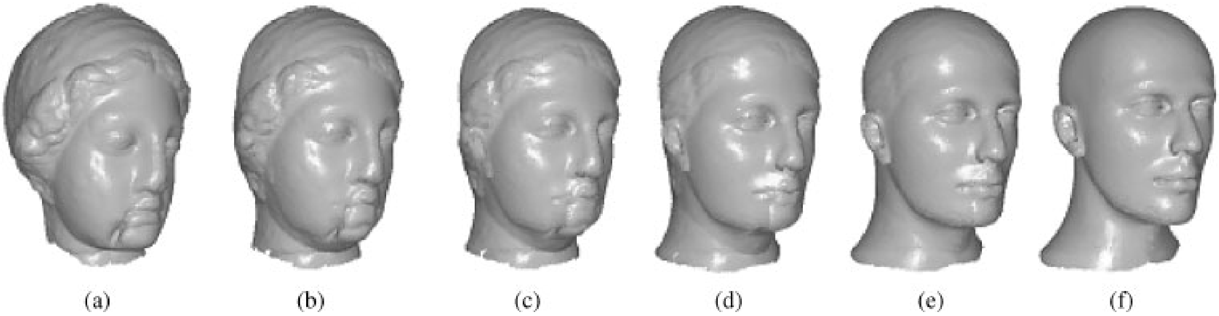


Figure 6. Morphing from Venus to Manhead with 5 features, (a) Venus (40,818 points), (f) Manhead (109,450 points).

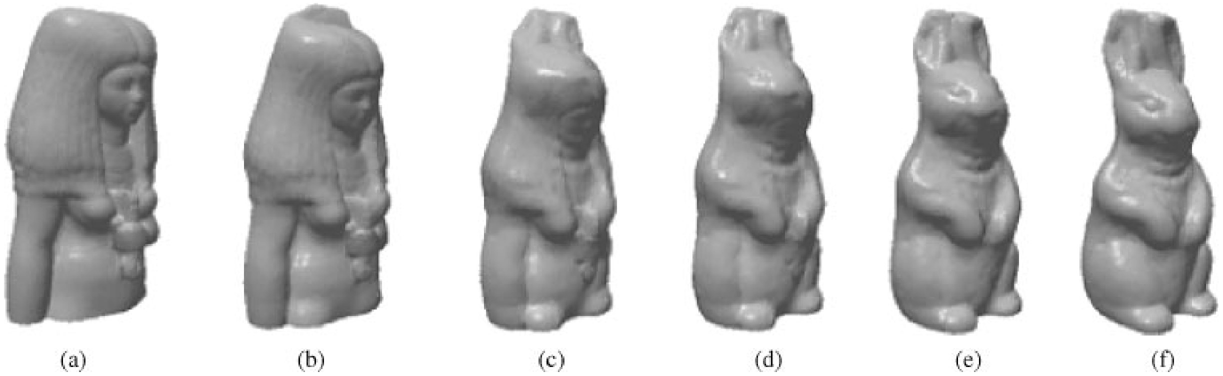


Figure 7. Morphing from Isis to Rabbit with 5 features, (a) Isis (118,250 points), (f) Rabbit (34,234 points).

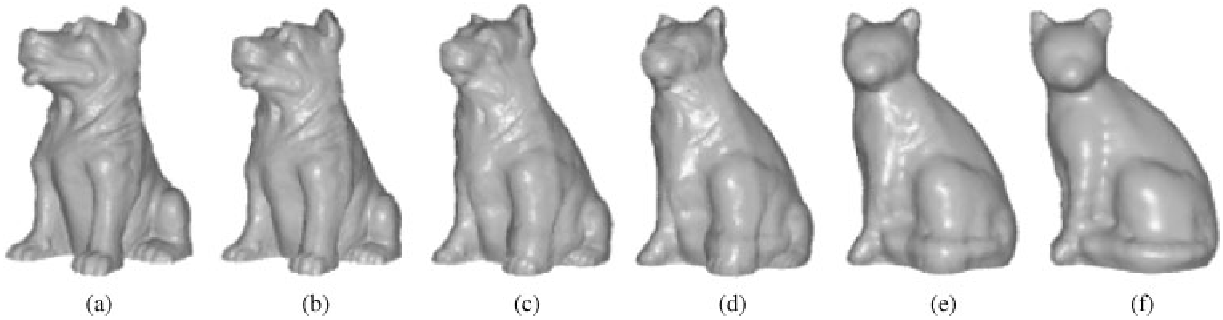


Figure 8. Morphing from Dog to Cat with 5 features (a) Dog (98,250 points), (f) Cat (84,348 points).

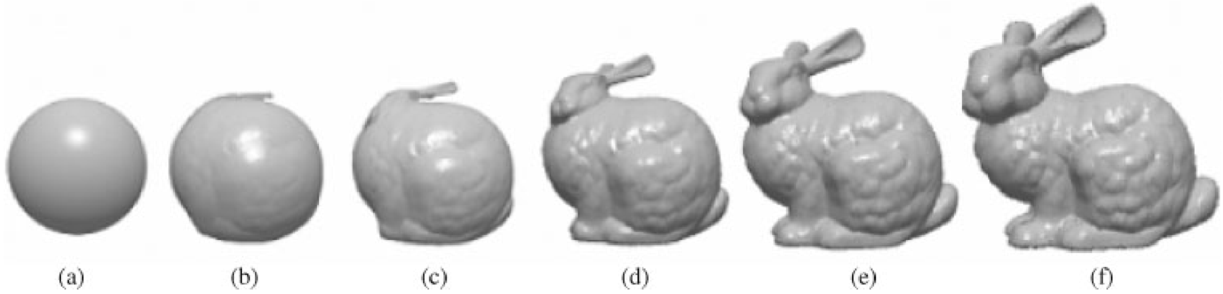


Figure 9. Morphing based on surface decomposition. Note there are no cracks during morphing.

future, we plan to embed the genus-0 geometry into the unit sphere, then build mappings for the source and target models without requiring decomposition.

ACKNOWLEDGMENTS

This work was supported by 973 program of China (No. 2002CB312101, 2002CB312102), NSFC grant (NO. 60103017) and Key project supported by NSFC (NO. 60033010). We are grateful to Yanwen Guo and Guofei Hu for their help.

References

1. Pfister H, Zwicker M, van Baar J, Gross M. Surfels: surface elements as rendering primitives. In *Proceedings of SIGGRAPH 2000*, New Orleans, Louisiana, 23–28 July 2000; 335–342.
2. Rusinkiewicz S, Levoy M. QSplat: a multiresolution point rendering system for large meshes. In *Proceedings of SIGGRAPH 2000*, New Orleans, Louisiana, 23–28 July 2000; 343–352.
3. Pauly M, Keiser R, Kobbelt L, Gross M. Shape modeling with point-sampled geometry. In *Proceedings of SIGGRAPH*

2003, San Diego, California, 27–31 July 2003; *ACM Transactions on Graphics* 2003; **22**(3): 641–650.

4. Adams B, Dutré P. Interactive Boolean operations on surfel-bounded solids. In *Proceedings of ACM SIGGRAPH 2003*, 2003; 651–656.
5. Zwicker M, Pauly M, Knoll O, Gross M. Pointshop 3D: an interactive system for point-based surface editing. In *Proceedings of SIGGRAPH 2002*, San Antonio, Texas, 21–26 July 2002; *ACM Transactions on Graphics* 2002; **21**(3): 322–329.
6. Alexa M, Behr J, Cohen-or D, Fleishman S, Levin D, Silva CT. Point set surfaces. In Ertl T, Joy KI, Varshney A (eds). *Proceedings of IEEE Visualization 2001*, San Diego, California, 21–26 October 2001, IEEE, Piscataway, New Jersey, 2001, 21–28, 537.
7. Zigelman G, Kimmel R, Kiryati N. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics* 2002; **8**(2): 198–207.
8. Alexa M. Recent advances in mesh morphing. *Computer Graphics Forum* 2002; **21**(2): 173–196.
9. Kent JR, Carlson WE, Parent RE. Shape transformation for polyhedral objects. *Computer Graphics (Proceedings of SIGGRAPH 92)* 1992; **26**(2): 47–54.
10. DeCarlo D, Gallier J. Topological evolution of surfaces. In *Proceedings of Graphics Interface'96*, Canadian Human-Computer Communications Society, Toronto: Canada, 1996; 194–203.
11. Gregory A, Stat A, Lin MC, Manocha D, Livingston MA. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Proceedings of Computer Animation '98*, Philadelphia, 1998.
12. Alexa M. Merging polyhedral shapes with scattered features. *The Visual Computer* 2000; **16**(1): 26–37.
13. Shapiro A, Tal A. Polyhedron realization for shape transformation. *The Visual Computer* 1998; **14**(8/9): 429–444.
14. Praun E, Hoppe H. Spherical parameterization and remeshing. In *Proceedings of SIGGRAPH 2003*, San Diego, California, 27–31 July 2003; *ACM Transactions on Graphics* 2003; **22**(3): 340–349.
15. Cmolik L, Uller M. Point cloud morphing. http://www.cg.tuwien.ac.at/studentwork/CESCG/CESCG-2003/LCmolik/paper.pdf
16. Pauly M, Gross M, Kobbelt L. Efficient simplification of point-sampled surfaces. In *Proceedings of IEEE Visualization 2002*, Boston, Massachusetts, 27 October–1 November, 2002; 163–170.
17. Levy B. Constrained texture mapping for polygonal meshes. In *Proceedings of SIGGRAPH 2001*, Los Angeles, California, 12–17 August 2001; 417–424.
18. Floater MS, Reimers M. Meshless parameterization for surface reconstruction. *Computer Aided Geometric Design* 2001; **18**(2): 77–92.
19. Xiao C, Peng Q. Point-based surface decomposition and patch selection based on level set methods. Submitted.
20. Memoli F, Sapiro G. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *Journal of Computational Physics* 2001; **173**(2): 730–764.
21. Gotsman C, Gu X, Sheffer A. Fundamentals of spherical parameterization for 3D meshes. In *Proceedings of SIGGRAPH 2003*, San Diego, California, 27–31 July 2003; *ACM Transactions on Graphics* 2003; **22**(3): 358–363.