Skeleton-driven Laplacian Mesh Deformations

Shin Yoshizawa, Alexander Belyaev,
and Hans-Peter Seidel

MPI–I–2006–4-005 October 2006

Authors' Addresses

Shin Yoshizawa
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Alexander Belyaev
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Publication Notes

This work is based on presentations in Poster at EUROGRAPHICS Symposium on Geometric Processing, June 26-28, 2006, Sardinia, Italy, and in oral presentation at 1st Int. Workshop on Shapes and Semantics, June 17, 2006, Matsushima, Japan.

Acknowledgements

Models are courtesy of the Stanford University (Armadillo, Dragon), EU AIM@Shape project (Camel, Gargoyle, Homer), and FarField Technology Ltd (hand). This work was supported in part by AIM@SHAPE, a Network of Excellence project (506766) within EU's Sixth Framework Programme.

Abstract

In this report, a new free-form shape deformation approach is proposed. We combine a skeleton-driven mesh deformation technique with discrete differential coordinates in order to create natural-looking global shape deformations. Given a triangle mesh, we first extract a skeletal mesh, a two-sided Voronoi-based approximation of the medial axis. Next the skeletal mesh is modified by free-form deformations. Then a desired global shape deformation is obtained by reconstructing the shape corresponding to the deformed skeletal mesh. The reconstruction is based on using discrete differential coordinates. Our method preserves fine geometric details and original shape thickness because of using discrete differential coordinates and skeleton-driven deformations. We also develop a new mesh evolution technique which allow us to eliminate possible global and local self-intersections of the deformed mesh while preserving fine geometric details. Finally, we present a multiresolution version of our approach in order to simplify and accelerate the deformation process.

Keywords

skeleton-driven deformations, Laplacian deformations, medial ball, self-intersections, shape thickness

Contents

1 Introduction **2**

2 Skeletal Mesh Extraction **5**

3 Skeletal Mesh Editing **7**

4 Skeleton-driven Laplacian Deformation **8**

5 Shape Preserving Self-Intersection Fairing **15**

6 Results **19**

7 Conclusion **25**

1 Introduction

Generating natural-looking deformations of complex shapes has multiple applications in CAGD, computer animation, and geometric modeling. Since the pioneering works [7, 47], developing fast, efficient, and intuitive methods for local and global free-form shape deformations is a subject of intensive research, see the recent works [15, 56, 28] and references therein.

Pierre Bézier (1910 - 1999) introduced the idea of deforming shapes embedded in tensor product splines through a mapping between different configurations of spline control points [46]. This idea, called space deformations, was inherited to the famous FFD (Free-Form Deformation) technique [47] which deforms shapes embedded in regular lattices equipped with Bézier splines. Further developments of the FFD include extensions for non-regular lattices [19], direct manipulations [27], arbitrary lattices [39], mean value coordinates [29], and many others. Besides global deformations [7] as bending, twining, and tapering, space deformations can be employed without lattices. For example, displacement vectors with smooth influence functions [13, 45, 58, 8] produce local bumping and sculpting. In [5] authors proposed to decompose a local sculpting deformation into a sequence of space deformations in order to avoid self-intersections. The technique of [5] is extend for constant volume deformations in [4] by using a set of swirling deformations. Many space deformation schemes are equipped with intermediate control interfaces such as lattices (FFDs), stick-figure skeletons [40, 33], curves [50], triangle meshes [51, 30], and vector field [56] in order to control deformations intuitively and to avoid specifying many local deformations to obtain a large-scale deformation. Most of these deformations are formulated in terms of weighted blending of local coordinate frames attached to the intermediate control interfaces. Parameters of these weighted blending are usually required tedious manual adjustments to obtain natural-looking (intuitive) deformations as mentioned in [35].

Recently skeleton-driven global free-form shape deformations drew a considerable attention [35, 51, 16]. The skeleton-driven deformations are well-suited for large-scale shape deformations and, therefore, can be used in numerous applications in computer game and digital movie industries. Bloomenthal [10, 9] proposed to use the medial axis as skeletal control interface in order to obtain natural-looking deformations by preserving original shape thickness. Unfortunately, fine geometric details of the original shape are often lost during a process of convolving the distance field in [9]. A similar problem arises in the skeleton-driven shape deformation method proposed in [59] because of smoothing fold-overs.

In this paper, we propose a new and powerful approach for free-form skeleton-driven global mesh deformations by using the so-called *discrete differential coordinates*. First a skeletal mesh, a Voronoi-based approximation of the medial axis, is extracted from a given mesh. Next the skeletal mesh is modified by free-form deformations. Then a desired global shape deformation is obtained by reconstructing the shape corresponding to the deformed skeletal mesh. The reconstruction is based on using discrete differential coordinates. The use of the medial axis prevents the so-called *collapsing joint defects* [35] which are thickness changing effects where a large bending or twisting deformation is applied via [40, 33], see Figure 1.1.

The discrete differential coordinates of a mesh vertex are defined by a discrete Laplacian of the vertex. Each coordinate associates with a corresponding element of the Laplacian vector. Since seminal works [1, 37, 60], mesh deformations based on discrete differential coordinates are intensively studied because of their detail preserving ability. These methods first interpolate or propagate the user-specified affine transformations over the mesh, and then the final deformations are obtained by solving the discrete approximations of a Poisson equation. Geometric details of the mesh are embedded into a discrete Laplacian matrix, and solving the Poisson equation diffuses a deformation error over the mesh. Therefore, the fine geometric details are preserved with certain smoothness during their deformations. See a recent review of this topic [52] and references therein. Since the conventional deformation techniques based on discrete differential coordinates do not preserve original shape thickness, our approach based on combining the skeleton-driven with discrete differential coordinates achieves more natural deformations than the conventional methods. To handle large and complex meshes, a multiresolution mesh representation is employed with our deformation technique. Figure 1.2 illustrates an example of our skeleton-driven Laplacian deformation technique.

Furthermore, the use of discrete differential coordinates allows us to achieve shape preserving self-intersection fairing. We develop a new mesh evolution technique which eliminates certain self-intersections of the deformed mesh simultaneously preserving fine geometric details by minimizing the thickness and deformation errors.

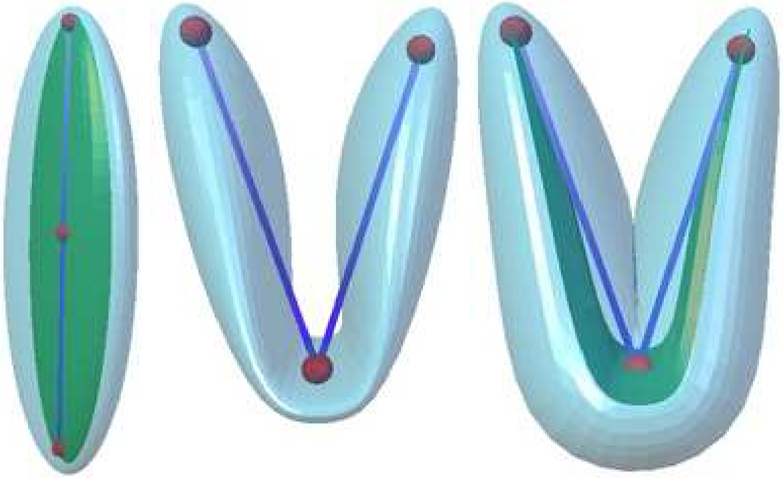
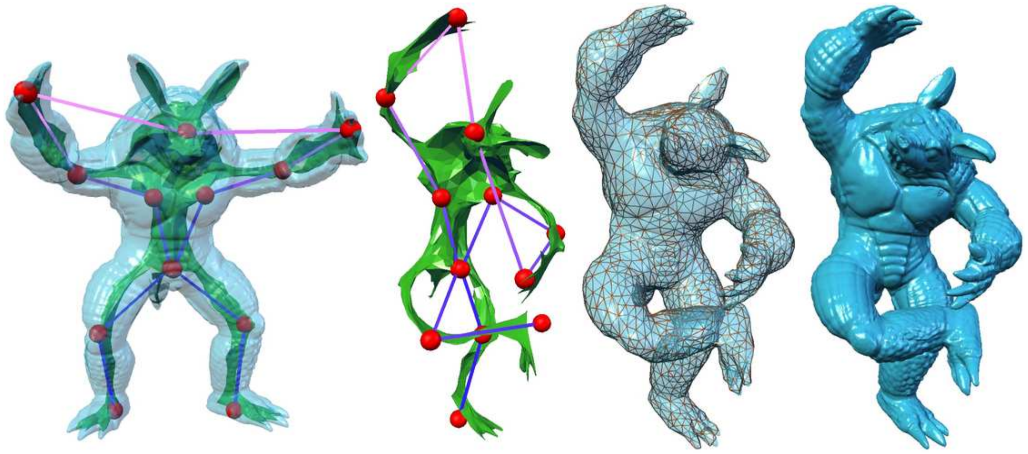


Figure 1.1: Preventing collapsing joint defects. Left: the stick-figure skeleton, the ellipsoid mesh with its skeletal mesh. Deformed meshes via the SSD [MTLT88,LCJ94] (Center) and our method (Right).



(a) (b) (c) (d)

Figure 1.2: A skeleton-driven Laplacian mesh deformation example. (a): Armadillo (332K triangles), its skeletal mesh (5K triangles), and the stick-figure skeleton. (b): A space deformation of the skeletal mesh. (c): A coarse deformed mesh obtained by using discrete differential coordinates from the deformed skeletal mesh. (d): A multiresolutional mesh representation gives us the final dense deformed mesh.

2 Skeletal Mesh Extraction

The medial axis of a bounded figure is a locus of centers of maximal empty balls where the maximal empty ball, also called *medial ball* [3], is completely contained in no other empty ball. The medial axis was introduced by Blum [11, 12] for 2D shape perception and recognition purposes. In 3D, Voronoi-based approximations of the medial axis have been often studied in surface reconstructions [2, 22] and shape manipulations [10, 9, 59].

Given a mesh M, our first goal is to extract an approximate skeletal mesh S such that

$$M = S + d\mathbf{N}, \tag{2.1}$$

where N is the field of unit mesh normals defined at the vertices of M and d is the set of distances from the vertices of M to the corresponding vertices of S along $\mathbf{N}$. The shape representation (2.1) was proposed by [49] for mathematical analysis of the medial axis. The relation (2.1) is the core of our approach. It allows us to edit the original mesh M via modifying its skeletal mesh S. Below we explain how to achieve a robust extraction of the skeletal mesh and build representation (2.1).

In order to extract a skeletal mesh S from a given original mesh M, we employed the Voronoi-based two-sided approximation of the medial axis proposed in [26]. First the Voronoi cells are calculated for all vertices of M by using the Quickhull algorithm [6]. Every vertex of M associates with a Voronoi cell. Consequently, one-to-one correspondences between the vertices of M and the associated vertices of S are established by assigning the Voronoi poles [2] as skeletal mesh vertices. The connectivity of S is copied from the connectivity of M. Hence, each triangle of M also has a one-to-one correspondence with the associated triangle of S. Figure 2.1 illustrates the skeletal mesh extracted from the given 3D mesh. The extracted skeletal mesh is a manifold approximation of the medial axis, therefore, conventional mesh processing methods can be applied to S. As indicated in [26, 59], applying pre-smoothing for M helps us to extract a perceptually good skeletal mesh.

Post-Smoothing. Instead of high precision approximations of the medial axis required in surface reconstructions [2, 22], a spike-less and non-degenerated skeletal mesh is desirable for our purpose because the local frames of our deformation are determined by the associated skeletal mesh triangles.

Inspired by the bilaplacian tangential smoothing proposed in [57], the following special smoothing scheme is employed to S if S is noisy or degenerated. Assume that the normals of M are oriented from S to M. The bilaplacian tangential smoothing is applied for any vertex of S such that the inner product between a smoothing vector of the vertex of S and a one-to-one corresponding vertex normal of M is negative; otherwise, the vertex position of S is not moved by the smoothing.

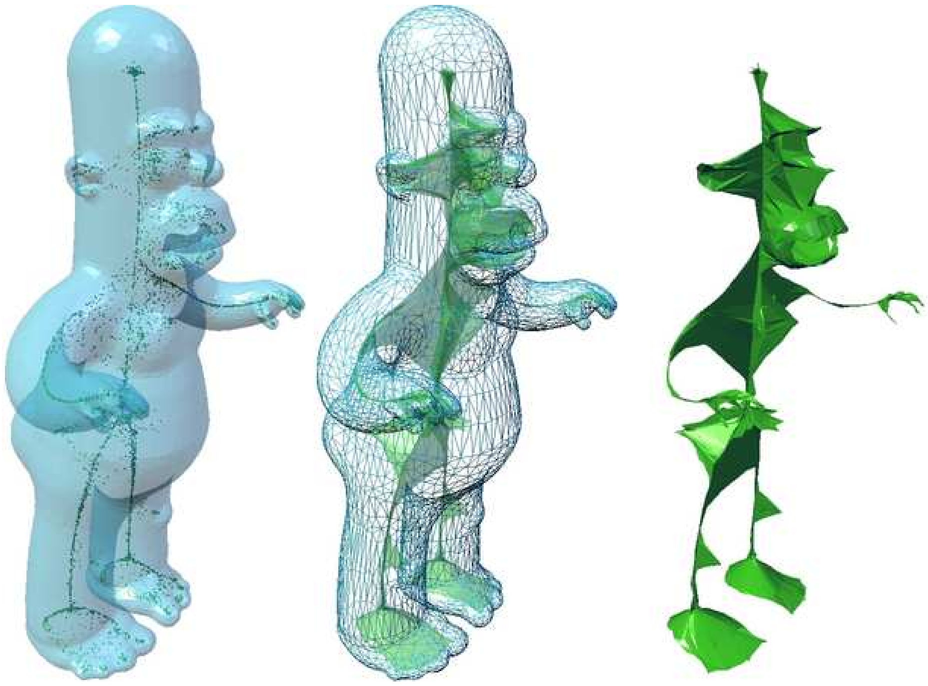


Figure 2.1: Skeletal mesh extraction. Left: Homer mesh and its Voronoi poles. Center: the triangulation of the Voronoi poles by copying Homer mesh connectivity. Right: the skeletal mesh of Homer mesh.

3 Skeletal Mesh Editing

Consider a free-form deformation of the skeletal mesh S. Our method is not restricted by any particular space deformation technique used for editing the skeletal mesh. In our implementation we use a set of free-form deformation tools developed in [58].

Also following [9], we have implemented the skeletal sub-space deformations (SSD) [40, 33]. Let $\mathbf{y}_j$ and $\mathbf{y}_j^d$ be an original local frame origin and a deformed local frame origin, respectively. Any position $\mathbf{z} \in \Re^3$ is transformed according to

$$\sum_j w_j(\mathbf{y}_j^d + A_j(\mathbf{z} - \mathbf{y}_j)), \quad A_j = B_j^d B_j^{-1}, \tag{3.1}$$

where B_j and B_j^d are the original and deformed frames, respectively. The frame is usually composed by three axes (3x3 matrices). Here $\sum_j w_j = 1$ and w_j is a normalized Gaussian-like function of the distance $|\mathbf{z} - \mathbf{y}_j|$. The SSD equation (3.1) can be interpreted as weighted blending of local transformations A_j for $\mathbf{z}$ embedded in the local frame coordinate system.

To implement stick-figure skeletons, one of the frame axes can be assigned to a normalized edge of the stick-figure skeleton. The joints of stick-figure skeletons may associate with more than one frame. The initial frame origins as control points can be picked on the skeletal mesh vertices, see the image (a) of Figure 1.2. Besides the Euclidean distance $|\mathbf{z} - \mathbf{y}_j|$, we use the geodesic distances on original and skeletal meshes.

4 Skeleton-driven Laplacian Deformation

Given a triangle mesh M, consider its skeletal mesh S extracted from M by the method described in Chapter 2, and a deformed skeletal mesh S_d. Here S_d is obtained from S by applying space deformations described in Chapter 3. The basic procedure of our skeleton-driven Laplacian deformation technique is as follows. First, a fragmented mesh M_F is generated by applying local transformations to all triangles of M where the transformations are defined by according to the local frames attached S and S_d. Then a final deformed mesh M_d is obtained by stitching the fragmented mesh triangles based on minimizing a deformation error. Here the error is given by a squared difference between the discrete differential coordinates of M_F and M_d. In [54, 60, 61], similar minimizing strategies are used for stitching fragmented meshes.

Let $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{s}^d$ be the vertices of M, S, and S_d, respectively. Since the deformation from S to S_d does not change the connectivity of S_d, there are one-to-one correspondences between $\mathbf{x}_i \in \mathbf{x}$, $\mathbf{s}_i \in \mathbf{s}$, and $\mathbf{s}_i^d \in \mathbf{s}^d$. Consider a final deformed mesh M_d corresponding to S_d. Let $\mathbf{x}^d$ be the vertices of M_d. Recall that M_d is given by the shape representation (2.1): $M_d = S_d + d\mathbf{N}_d$ where $\mathbf{N}_d$ are the unit normals of M_d, see Figure 4.1.

The radius of medial ball whose center is located on $\mathbf{s}_i$ is given by

$$d(i) = |\mathbf{x}_i - \mathbf{s}_i|. \tag{4.1}$$

Let $\mathbf{n}_i^d \in \mathbf{N}_d$ be a unit normal of M_d at $\mathbf{x}_i^d \in \mathbf{x}^d$. Thus, our shape representation (2.1) can be re-written as $\mathbf{x}_i^d = \mathbf{s}_i^d + d(i)\mathbf{n}_i^d$. In [59], the authors approximate $\mathbf{N}_d$ by decomposing a deformation from M to M_d into a sequence of deformations. Here let us approximate $\mathbf{n}_i^d$ by the following local transformation in order to use discrete differential coordinates.

Local Transformation. Let $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$, $\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}$, and $\{\mathbf{s}_i^d, \mathbf{s}_j^d, \mathbf{s}_k^d\}$ be corresponding triangles of M, S, and S_d, respectively. We attach the original lo-

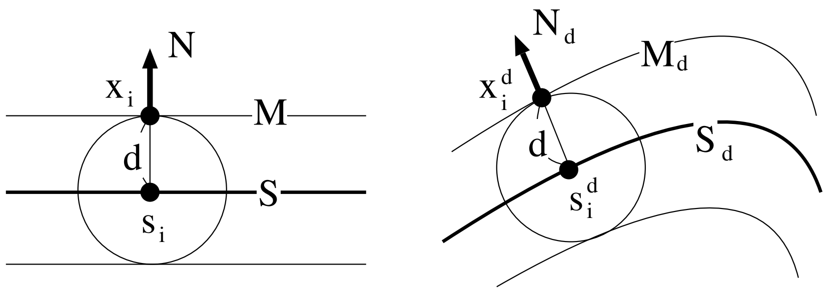


Figure 4.1: Shape representation (2.1).

cal frame $B_0 = (\mathbf{v}_0, \mathbf{t}_0^1, \mathbf{t}_0^2)$ to all triangles of S where $\mathbf{v}_0 = \mathbf{t}_0^1 \times \mathbf{t}_0^2$, $\mathbf{t}_0^1 = (\mathbf{s}_j - \mathbf{s}_i)/|\mathbf{s}_j - \mathbf{s}_i|$, and $\mathbf{t}_0^2 = (\mathbf{s}_k - \mathbf{s}_i)/|\mathbf{s}_k - \mathbf{s}_i|$. The skeletal mesh editing procedure changes the frame from B_0 to the deformed local frame B_d attached to a corresponding triangle of S_d. Here B_d is given by the same calculation procedure of B_0 by using $\{\mathbf{s}_i^d, \mathbf{s}_j^d, \mathbf{s}_k^d\}$ instead of $\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}$. See Figure 4.2.

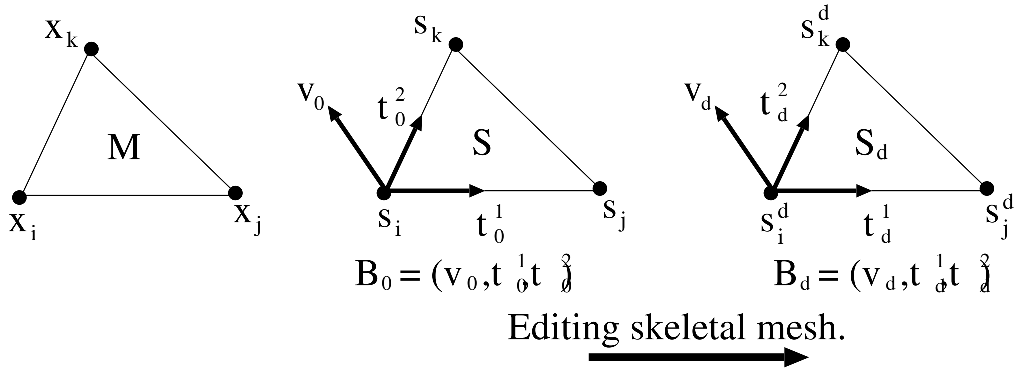


Figure 4.2: Corresponding triangles of M, S, and S_d. Local coordinate frames are attached to the corresponding triangles of S and S_d.

Let $\mathbf{n}_i \in \mathbf{N}$ be a unit normal of M at $\mathbf{x}_i$. Since the displacement $\mathbf{x}_i - \mathbf{s}_i$ represents a normal vector of M [3] at $\mathbf{x}_i$, $\mathbf{n}_i$ is given by $\frac{\mathbf{x}_i - \mathbf{s}_i}{|\mathbf{x}_i - \mathbf{s}_i|}$. Consider a local coordinate representation of $\mathbf{N}$ on S such that the coordinates of $\mathbf{n}_i$ are represented by $B_0^{-1}\mathbf{n}_i$. The coordinate transformation $B_d B_0^{-1}\mathbf{n}_i$ gives us an approximation of $\mathbf{n}_i^d$, the unit normal at $\mathbf{x}_i^d$. Thus, we have

$$\mathbf{n}_i^d \approx \frac{B_d B_0^{-1}(\mathbf{x}_i - \mathbf{s}_i)}{|B_d B_0^{-1}(\mathbf{x}_i - \mathbf{s}_i)|}.$$

Here $B_d B_0^{-1}$ is calculated per triangles. Therefore, applying the following transformation (4.2) to all triangles of M generates a fragmented mesh M_F. See Figure 4.3.

$$\mathbf{x}_l^T = \mathbf{s}_l^d + d(l)\frac{A(\mathbf{x}_l - \mathbf{s}_l)}{|A(\mathbf{x}_l - \mathbf{s}_l)|}, \quad A = B_d B_0^{-1}, \tag{4.2}$$

where $l = i, j, k$, $\mathbf{x}_l^T \in \mathbf{x}^T$ is the fragmented mesh vertex and $d(l)$ is the medial ball radius (4.1). Compared with similar transformations used in other deformations e.g. [54], the transformation (4.2) preserves shape thickness. Also the above equation (4.2) can be considered as a discrete approximation of the shape representation (2.1): $M_d = S_d + d\mathbf{N}_d$.

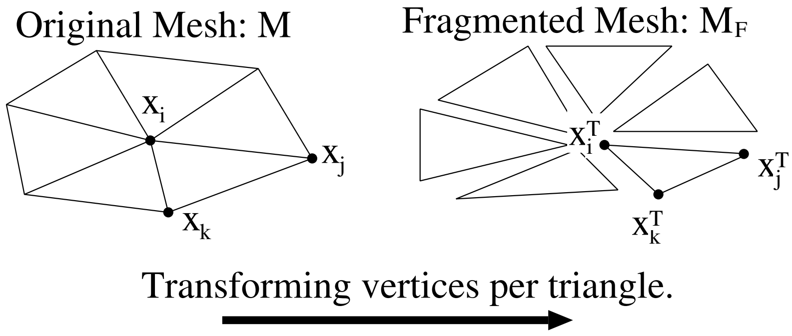


Figure 4.3: A fragmented mesh M_F, triangle soup, is generated via (4.2).

Although the transformation (4.2) generates stretch distortions, it does not destroy fine geometric details compared with shearing transformations in $\Re^3$ as mentioned in [53] because the stretch of (4.2) is locally embedded in the skeletal mesh. In [36] authors embed their local transformations to the original mesh in order to obtain semi-rigid deformations.

In the case of space deformations, e.g. (3.1) or [51, 30, 9], the final position of $\mathbf{x}_i^d$ of M_d is calculated by averaging the fragmented mesh vertices $\mathbf{x}^T$. Averaging the transformed vertices in Euclidean coordinates requires a large influence region which contains a lot of transformed vertices in order to generate nice deformations as indicated in [30]. In our approach, we calculate an average of the fragmented mesh vertices $\mathbf{x}^T$ in discrete differential coordinates where only one-link neighbor vertices are required for the blending.

Discrete Differential Coordinates. Consider two graphs G_M and G_{M_F} corresponding to two pairs of meshes $\{M, S\}$ and $\{M_F, S_d\}$, respectively. Since we have the one-to-one correspondence between $\mathbf{x}$ and $\mathbf{s}$, the graph structures are constructed by adding edges between M and S (M_F and S_d) as illustrated in Figure 4.4. Here the vertex sets of G_M and G_{M_F} consist of $\{\mathbf{x}, \mathbf{s}\}$ and $\{\mathbf{x}^T, \mathbf{s}^d\}$, respectively. Consider also an another graph G_{M_d} whose vertex set consists of $\{\mathbf{x}^d, \mathbf{s}^d\}$. Here G_{M_d} is equipped with the same connectivity of G_M.

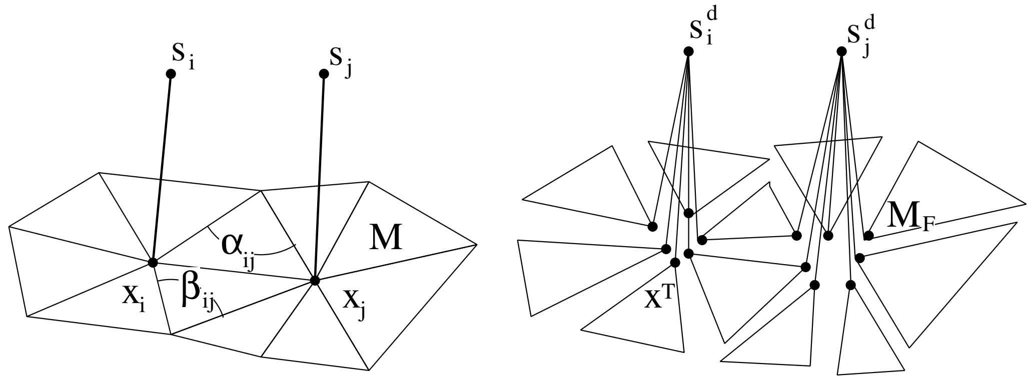


Figure 4.4: Two graphs G_M (Left) and G_{M_F} (Right) are considered.

Consider a weighted graph whose vertices are given by $\{v_1, v_2, ..., v_i, ..\}$. If there is the edge between v_i and v_j then they are adjacent vertices. The ij element of a graph-Laplacian matrix of the graph is defined by

$$\begin{cases} -w_{ij} & \text{if } v_j \text{ and } v_i \text{ are adjacent,} \\ \sum_{k\in N(i)} w_{ik} & \text{if } v_j = v_i, \\ 0 & \text{otherwise} \end{cases}$$

where w_{ij} is a weight associated with the edge between v_i and v_j. Here $N(i)$ is the index set of adjacent vertices of v_i. See [18] for mathematical theory of a graph.

Consider the graph-Laplacian matrix for G_M where the each edge is equipped with a weight w. Here we use standard cotangent weights [44] $w = \cot\alpha_{ij} + \cot\beta_{ij}$ for edges between $\mathbf{x}_i$ and $\mathbf{x}_j$ where the angels α_{ij} and β_{ij} are defined in Figure 4.4. For the weight associated with the edge between $\mathbf{x}_i$ and $\mathbf{s}_i$, we use $w = 1$. Thus the graph-Laplacian matrix associated with G_M for $(\mathbf{x}, \mathbf{s})^*$ is given by

$$\begin{pmatrix} I{+}L_M & -I \\ -I & I \end{pmatrix}, \tag{4.3}$$

where I is the identity matrix, L_M is a mesh Laplacian matrix, and $\mathbf{a}^*$ stands for a transpose of vector $\mathbf{a}$. The ij element of L_M is given by

$$\begin{cases} -(\cot\alpha_{ij}+\cot\beta_{ij}) & j\in N(i) \\ \sum_{k\in N(i)}(\cot\alpha_{ik}+\cot\beta_{ik}) & j=i \\ 0 & \text{otherwise} \end{cases}$$

where $N(i)$ is the index set of the one-link neighborhood vertices of $\mathbf{x}_i$.

Consider the graph-Laplacian operator for G_{M_F} where the each edge is equipped with a weight w. We use $w=\cot\angle\mathbf{x}_i\mathbf{x}_k\mathbf{x}_j$ for the edge between $\mathbf{x}_i^T$ and $\mathbf{x}_j^T$ and $w=\cot\angle\mathbf{x}_i\mathbf{x}_j\mathbf{x}_k$ for the edge between $\mathbf{x}_i^T$ and $\mathbf{x}_k^T$, see the right image of Figure 4.3. Also the weight for the edge between $\mathbf{x}_i^T$ and $\mathbf{s}_i^d$ is equal to $\frac{1}{T_i}$ where T_i is the number of the one-link neighborhood triangles of $\mathbf{x}_i$.

The discrete differential coordinates of G_{M_d} and G_{M_F} are the above graph-Laplacians of their vertices $(\mathbf{x}^d,\mathbf{s}^d)$ and $(\mathbf{x}^T,\mathbf{s}^d)$, respectively.

Stitching Fragmented Mesh. In order to obtain the final deformed mesh M_d, let us minimize a difference between G_{M_d} and G_{M_F} in the discrete differential coordinates subject to the following boundary condition: the boundary of G_{M_d} is fixed to $\mathbf{s}^d$. This boundary condition changes the graph-Laplacian from (4.3) to

$$L_0=\begin{pmatrix} I+L_M & -I \\ 0 & I \end{pmatrix}. \tag{4.4}$$

More precisely, $\mathbf{x}^d$ is given by solving the following sparse linear system

$$\text{minimize } |L_0\mathbf{u}-\mathbf{b}|^2 \Rightarrow \mathbf{u}=L_0^{-1}\mathbf{b}, \tag{4.5}$$

where $\mathbf{u}=(\mathbf{x}^d,\mathbf{s}^d)^*$ and $\mathbf{b}=(\mathbf{x}^f,\mathbf{s}^d)^*$ is the averaged discrete differential coordinates of G_{M_F}. Here the i-th element of $\mathbf{x}^f$ is given by

$$\sum_{j,k\in N(i)} w_1(\mathbf{x}_i^T-\mathbf{x}_j^T)+w_2(\mathbf{x}_i^T-\mathbf{x}_k^T)+\frac{(\mathbf{x}_i^T-\mathbf{s}_i^d)}{T_i}, \tag{4.6}$$

where $w_1=\cot\angle\mathbf{x}_i\mathbf{x}_k\mathbf{x}_j$, $w_2=\cot\angle\mathbf{x}_i\mathbf{x}_j\mathbf{x}_k$, and T_i is the number of the one-link neighborhood triangles of $\mathbf{x}_i$.

In [61] authors evaluate the elements of their mesh Laplacian according to the fragmented mesh coordinates. Besides L_0 is a graph-Laplacian, our angles $\angle\mathbf{x}_i\mathbf{x}_k\mathbf{x}_j$ and $\angle\mathbf{x}_i\mathbf{x}_j\mathbf{x}_k$ do not depend on the fragmented mesh coordinates. Compared with [62] which employed a graph-Laplacian for their deformation technique, our graph structure is much simple and has a nice geometric property as the shape thickness.

Figure 4.5 illustrates the main idea and procedure of our skeleton-driven Laplacian deformation technique described above.

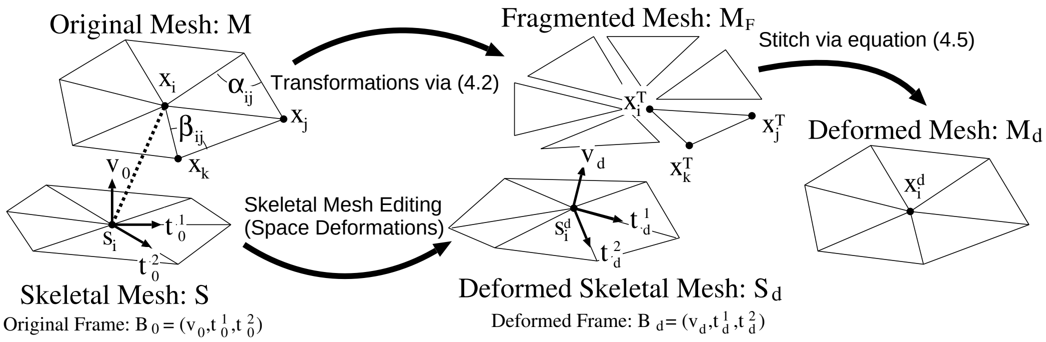


Figure 4.5: Skeleton-driven Laplacian deformation framework.

Solving Linear System: Factorization. Due to recent developments of the direct sparse linear solvers [55, 20], we only need factorization of L_0 once for producing L_0^{-1}. Then every deformation is obtained by updating $\mathbf{b}$ and a backward substitution of the factorized matrix with the updated $\mathbf{b}$. This gives us the linear computational complexity. Unfortunately L_0 is not a symmetric matrix. Consequently, we use the UMFPACK [20] to factorize L_0 instead of the TAUCS [55] which is specialized to a symmetric matrix employed in least-square systems of the Laplacian mesh deformations [37, 53, 36]. Because $\mathbf{s}^d$ is being the fixed boundary, there are no ill-conditioned problems mentioned in [52] caused by too small boundary conditions. According to our numerical experiments, L_0 is invertible as long as M is a non-degenerated manifold mesh.

Multiresolution Representation. We implement a simplified version of DSS (Displaced Subdivision Surface) [34] and combine with our skeleton-driven Laplacian deformations. The skeletal mesh is extracted from the coarse control mesh of the DSS. The control mesh is deformed by our skeleton-driven Laplacian deformations. Then a dense deformed mesh is obtained by using the DSS mechanism: subdividing the deformed control mesh and adding scaler displacements to the subdivided mesh normals.

Our method without DSS preserves fine geometric details, but the skeletal mesh of a textured mesh may have complex geometry. Hence, the skeletal mesh editing of the textured mesh without degenerations could become a very difficult task. Using multiresolution representations helps to reduce excessive complexity of the skeletal mesh. Figure 4.6 illustrates advantages of using the multiresolution representation for our skeleton-driven Laplacian deformations.

Dense Mesh and its Skeletal Mesh (332K triangles)

Control Mesh and its Skeletal Mesh (5.2K triangles)

	Triangle	Editing Skeleton	Update Geometry Factorization + Substitution	DSS	Total
Without DSS:	346K	94s	321s + 702s		1117s
DSS Mesh:	5.2K, 332K	**0.13s**	0.05s + **0.25s**	3.8s	4.23s

Figure 4.6: Advantages of multiresolution skeleton-driven Laplacian deformations. The use of the multiresolution representation reduces excessive complexity of the skeletal mesh and accelerates the deformation process.

5 Shape Preserving Self-Intersection Fairing

The final deformed mesh M_d obtained by solving the system (4.5) may contain self-intersections. Here M_d is an approximation of the envelope of the medial balls of M attached to S_d. For a given smooth surface, the envelope of its medial balls is equivalent to the zero level set of the union of its medial balls. However the medial balls of M attached to S_d are not necessary inscribed maximal empty balls (medial balls) of M_d and, therefore, M_d may have self-intersections. In such cases, the mesh reconstructions based on the envelope and union of the attached medial balls form two different shapes, see Figure 5.1. The differences of them are exactly subsets of the envelope which we would like to eliminate.

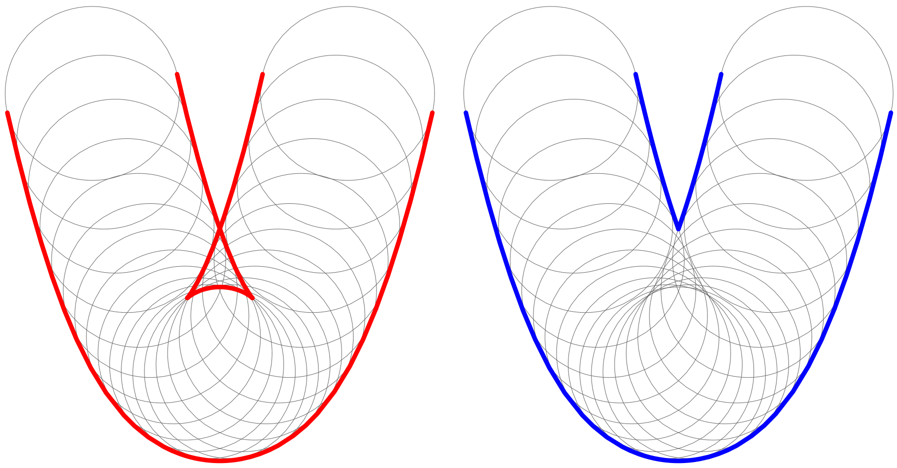


Figure 5.1: Envelope vs. Union. Left: the envelope of the balls. Right: the zero level set of the union of the balls.

The union of medial balls is defined by the signed distance function

$$f(\mathbf{z}) = \min_{\forall j} \left\{ |\mathbf{z} - \mathbf{s}^d_j| - d(j)) \ : \ \mathbf{z} \in \Re^3, \mathbf{s}^d_j \in \mathbf{s}^d \right\}, \tag{5.1}$$

where $d(j)$ is the medial ball radius (4.1).

Below we describe a novel self-intersection fairing method based on minimizing a self-intersection error in the discrete differential coordinates. The self-intersection error is measured by the squared distance $f^2(\mathbf{x}^d)$ from $\mathbf{x}^d$ to the zero level set of the union of the attached medial balls. Here $\mathbf{x}^d$ and f are given by the equations (4.5) and (5.1), respectively. Consider evolutions of a graph

$$\frac{\partial L_0 U(t)}{\partial t} = F(U), \tag{5.2}$$

$$F(U) = \begin{cases} f(\mathbf{z})\nabla f(\mathbf{z}) & \text{If} \quad \mathbf{z} = \mathbf{x}^d \\ 0 & \text{If} \quad \mathbf{z} = \mathbf{s}^d \end{cases}$$

where $U(t)$ is an evolving graph and $U(0) = (\mathbf{x}^d, \mathbf{s}^d)^*$ defined in the previous chapter and L_0 is the graph-Laplacian defined in the equation (4.4). This evolution (5.2) is motivated by the shape preserving effect of L_0 and integrating the deformation and self-intersection errors. Note that $\mathbf{s}^d$ and L_0^{-1} are not changed by the evolution (5.2).

The equation of (5.2) is a gradient descent flow of the self-intersection error $f^2(\mathbf{x}^d)$ in differential coordinates. This gradient descent flow modifies the differential coordinates of $U(0)$. In order to preserve the fine geometric details of M, L_0 is equipped with the original weights of G_M instead of the weights of G_{M_d} or $U(t)$. Hence, the evolution (5.2) minimizes not only the self-intersection error but also the deformation error simultaneously. One can find a similarity between (5.2) and Sobolev gradient flows, see [42, 43] for mathematical theory of Sobolev gradients. In [17], a similar Sobolev gradient flow is employed for describing active contours.

The equation (5.2) is approximated by the following semi-implicit mesh evolutions.

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \epsilon L_0^{-1} F(\mathbf{u}^n) \nabla F(\mathbf{u}^n), \tag{5.3}$$

where $\mathbf{u}^0 = (\mathbf{x}^d, \mathbf{s}^d)^*$.

Our evolution technique (5.3) is more robust and effective than the self-intersection fairing scheme proposed in [59] because of our semi-implicit formulation and the shape preserving effect of L_0^{-1}. Moreover the evolution (5.3) prevents *peeling skin defects* because $\mathbf{s}^d$ plays a role of anchors during the evolutions, see Figure 5.2. This peeling skin defect is a common problem when we consider a mesh evolution on a surface where the evolving mesh may fall into a degenerated solution

Figure 5.2: Preventing peeling skin defects. Self-intersection fairings via the mesh (Left) and graph (Right) Laplacians. The large sphere represents one of medial balls, and the small sphere indicates the corresponding medial ball center.

e.g. the spherical mesh parameterization scheme [25] may produce such a degenerated solution [24, 23]. Compared with [62], the global self-intersections can be eliminated by our evolution (5.3), see Figure 5.3.

Every evolution step increases accuracy of the thickness preservation because of minimizing $f^2(\mathbf{x}^d)$. Therefore, the evolution (5.3) is useful even if there are no self-intersections in M_d, but $f^2(\mathbf{x}^d) \neq 0$.

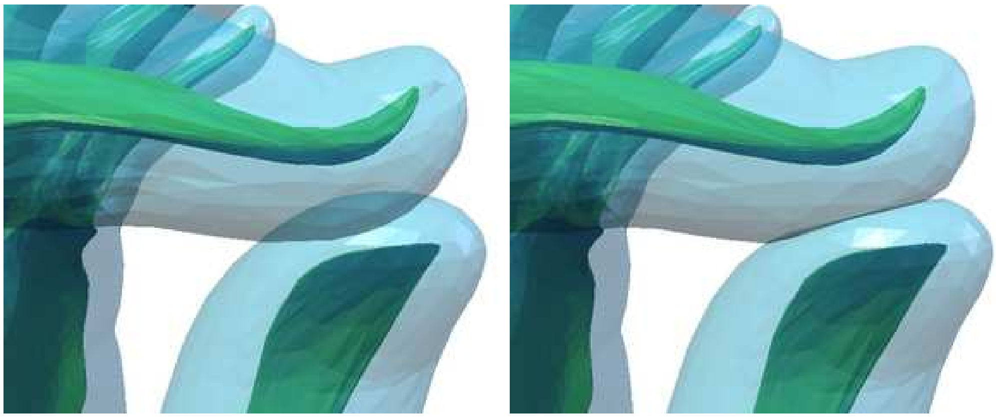


Figure 5.3: Global self-intersection fairing. Left: the deformed mesh with global self-intersections. Right: the fairing result via our evolutions (5.3).

Fast evaluation of f. Although L_0^{-1} is already computed for solving the system (4.5), evaluating the equation (5.1) could be time consuming. We accelerate the evaluation of f taking a union of a sub set of the medial balls instead of all medial balls attached to S_d.

For an evolving graph vertex $\mathbf{u}_i^k = \{\mathbf{x}_i^k, \mathbf{s}_i^d\}$, a maximum bound of $f(\mathbf{x}_i^k)$ is given by $||\mathbf{x}_i^k - \mathbf{s}_i^d| - d(i)|$. For every $\mathbf{u}_i^k$, consider the neighbor vertices $\mathbf{u}_j^k$ of $\mathbf{u}_i^k$ and its index set $j \in E(i)$ such that $|\mathbf{x}_j^k - \mathbf{x}_i^k| \leq d(i)$. Then the union of medial balls with its gradient is approximated by

$$f(\mathbf{x}_i^k)\nabla f(\mathbf{x}_i^k) \approx (||\mathbf{x}_i^k - \mathbf{s}_p^d|| - d(p))(\mathbf{x}_i^k - \mathbf{s}_p^k)$$

where $p = \text{argmin}_j(||\mathbf{x}_i^k - \mathbf{s}_j^d| - d(j)|) : \forall j \in E(i)$. This vertex set $\mathbf{x}_j^k : j \in E(i)$ can be efficiently searched by using a kd-tree. We construct the kd-tree [41] for every evolution of (5.3) if the self-intersection fairing is necessity.

6 Results

Our method is implemented by using the JDK 1.4 and Java3D. The UMFPACK compiled by the GNU gcc 3.3.5 is called by Java Native Interface. Figures 1.1, 1.2, 6.2, and 6.4 demonstrate how well the original shape thickness is preserved during our deformations.

	Ellipsoid	Camel	Homer	Gargoyle	Armadillo	Dragon
V	2.5K	37K	65K	160K	166K	263K
F	4.9K	75K	130K	320K	332K	527K
V_c		2.3K	4.1K	1.5K	2.5K	4.1K
F_c		4.7K	8.2K	5K	5.2K	8.2K
(a)		4.5s	1.6s	4.9s	25.9s	13.9s
(b)	4.5s	4.7s	6.5s	9.9s	5.1s	4.8s
(c)		2	2	3	3	3
(d)		2.8s	4.3s	26s	24.7s	19.7s
(e)		11s	8.8s	200s	161s	103s
(f)	0.08s	0.05s	0.09s	0.05s	0.05s	0.09s
(g)	0.06s	0.09s	0.14s	0.76s	0.13s	0.54s
(h)	0.05s	0.11s	0.06s	0.77s	0.13s	0.32s
(i)	0.16s	0.13s	0.29s	0.79s	0.12s	0.44s
(j)		0.61s	3.86s	4.36s	3.8s	43.8s
(k)	1.1	6.2			1.2	6.2

Table 6.1: Timings. (a): Decimation for DSS. (b): Skeletal Mesh Extraction. (c): Subdivision Level. (d): Subdivision for DSS. (e): Displacement Sampling for DSS. (f): Factorization of L_0. (g): Skeletal Mesh Editing. (h): Transformation (4.2) and Solving (4.5). (i): Self-Intersection Fairing (5.3) (1 step). (j): DSS Reconstruction. (k): Corresponding Figure Numbers.

Timing. Table 6.1 represents timings which are measured on a 1.7 GHz Pentium 4 with 1 GB RAM computer. Here V and F are the vertex and triangle numbers of M, and V_c and F_c are the vertex and triangle numbers of the DSS's coarse control mesh, respectively. The interactive mesh deformations are achieved for the control meshes, see (g) and (h) of Table 6.1.

The multiresolution mesh representation significantly improves the computational time of our approach. For example, deforming Armadillo (V:173K, F:346K) without the DSS requires the following timings: (b):356s, (f):321s, (g):94s, (h):423s, and (i):279s where alphabets are explained in the caption of Table 6.1.

Comparison. We have implemented the conventional skeleton-driven deformations: weighted blending [9] and homotopy method [59] to compare with our approach. Our implementation of weighted blending [9] is summing up the local frame changes within its medial ball radius instead of all local frames. The homotopy method suffers from fold-overs especially twisting, see Figure 6.1. Our approach outperforms [9, 59] in terms of the both speed and quality.

Besides the conventional deformation methods based on discrete differential coordinates which include the most recent ones [36, 61, 62] do not have the skinning ability. The original shape thickness is not preserved during their deformations. We believe that the shape thickness is a natural and intuitive measurement than local volumes [14, 62, 15] or global volume [4, 56] because volume-preserving shape deformations include large shearing effects in $\Re^3$ which may not produce natural-looking mesh deformations as indicated in [53].

Discussion. Our approach depends on the skeletal mesh S. Consequently, if a part of S is degenerated to a space curve or a point (e.g. when M belongs to Dupin's cyclides) then we may need another definition of local frames B_0 and B_d for the degenerated part. A possible solution would be that a frame axis defined from a degenerated point to a neighbor skeletal vertex position is employed to construct the frame.

The graph-Laplacian L_0 defined by the equation (4.4) does not contain the angle information $\angle \mathbf{s}_i \mathbf{x}_i \mathbf{x}_j$ or $\angle \mathbf{s}_i \mathbf{x}_i \mathbf{x}_k$. Further work is required for deeper understanding of discretization effects of L_0.

Our self-intersection fairing (5.3) does not work appropriately if M_d intersects S_d because $\nabla F(U)$ may be oriented to an opposite direction of our desired gradient direction in such cases. Adding smoothing components to the fairing (5.3) would help such cases.

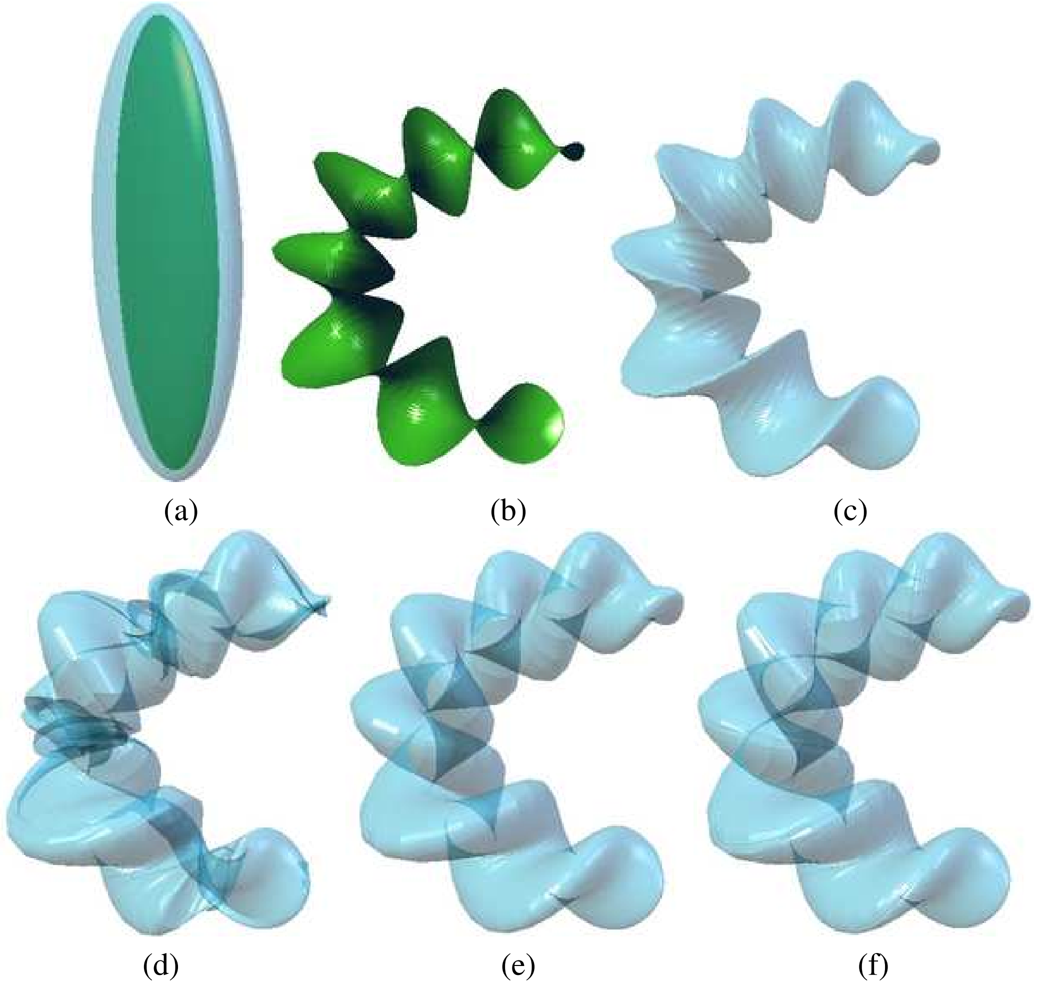


Figure 6.1: Comparison with conventional skeleton-driven deformations. Here f is given by the equation (5.1), and summation $\sum$ is carried out over all mesh vertices. (a): Ellipsoid and its skeletal mesh, V: 9.8K, F: 19.6K. (b): Edited skeletal mesh. (c): SSD [MTLT88,LCJ94], time: 2.3s, $\sum f^2$: 20.2. (d): Homotopy method [YBS03] without DSS, time: 2.7s, $\sum f^2$: 19.6. (e): Weighted blending [Blo02], time: 14.8s, $\sum f^2$: 11.8. (f): Our skeleton-driven Laplacian deformation method without DSS, time 2.9s, $\sum f^2$: 3.8. By using our evolution (5.3) $\sum f^2$ becomes 0.2.

Figure 6.2: Deformation examples. The initial models are represented in Figure 6.3.

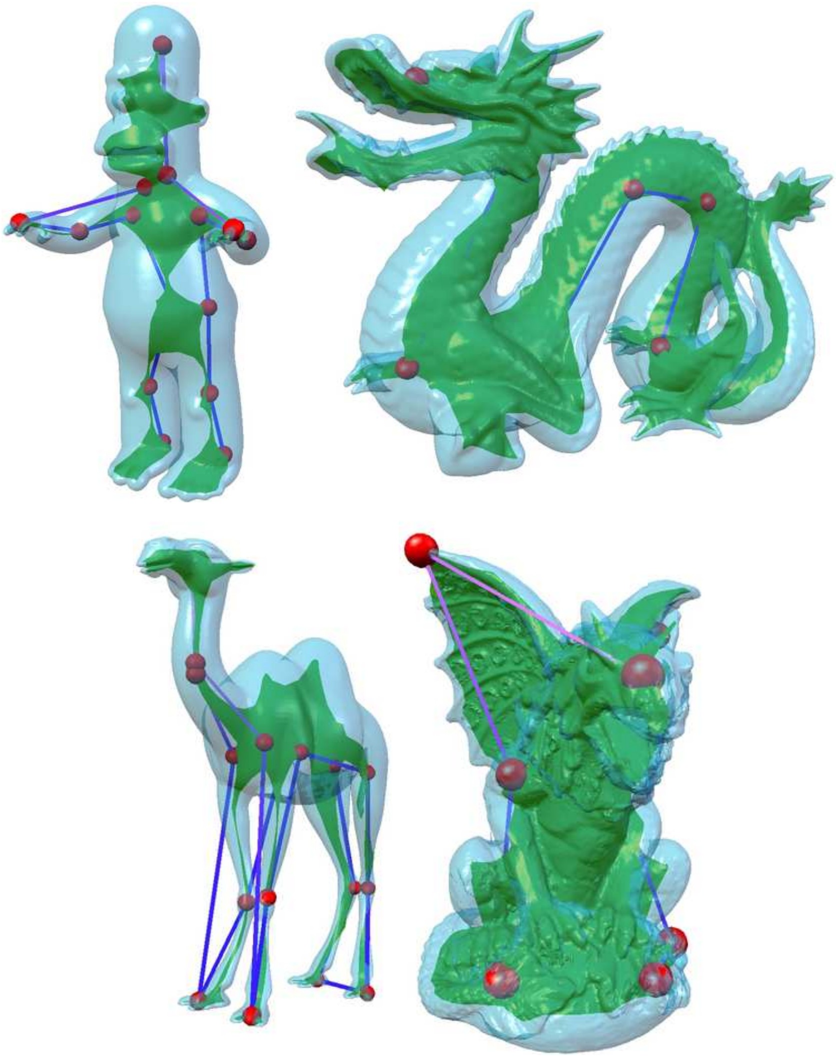


Figure 6.3: Initial meshes, their corresponding coarse skeletal meshes, and stick-figure skeletons.

Figure 6.4: Armadillo's gymnastics. Initial mesh, its coarse skeletal mesh, and the stick-figure skeleton are represented in the image (a) of Figure 1.2.

7 Conclusion

We proposed a novel semi-rigid mesh deformation technique based on combining skeleton-driven deformations with discrete differential coordinates. We have successfully integrated the advantages of both of them without loss of efficiency. Preserving the original shape thickness and fine geometric details while maintaining the smoothness of the surface allows us to generate natural-looking complex mesh deformations. Moreover, a new shape preserving self-intersection fairing is proposed.

A promising future work would be applying discrete differential coordinates to the skeletal mesh editing phase by using a single-sided skeletal mesh.

Bibliography

[1] M. Alexa. As-rigid-as-possible shape interpolation. *Proceedings of ACM SIGGRAPH*, pages 157–164, 2000.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH*, pages 415–421, 1998.

[3] N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.

[4] A. Angelidis, M.-P. Cani, G. Wyvill, and S. King. Swirling-sweepers: Constant-volume modeling. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pages 10–15, Washington, DC, USA, 2004. IEEE Computer Society.

[5] A. Angelidis, G. Wyvill, and M.-P. Cani. Sweepers: Swept user-defined tools for modeling by deformation. In *Proceedings of Shape Modeling International*, pages 63–73, 2004.

[6] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[7] A. H. Barr. Global and local deformations of solid primitives. *Proceedings of ACM SIGGRAPH*, pages 21–130, 1984.

[8] G. H. Bendels and R. Klein. Mesh forging: editing of 3d-meshes using implicitly defined occluders. In *Proceedings of Eurographics Symposium on Geometry Processing*, pages 207–217, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[9] J. Bloomenthal. Skinning: Medial-based vertex deformation. In *Proceedings of ACM SIGGRAPH Symposium on Computer Animation*, pages 147–151, 2002.

[10] J. Bloomenthal and C. Lim. Skeletal methods of shape manipulation. In *Proceedings of Shape Modeling International*, pages 44–47, 1999.

[11] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Symposium on Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[12] H. Blum. Biological shape and visual science. *J. Theor. Biology*, 38:205–287, 1973.

[13] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics*, 13(2):137–155, 1994.

[14] M. Botsch and L. Kobbelt. Multiresolution surface representation based on displacement volumes. *Computer Graphics Forum*, 22(3):483–491, 2003. Eurographics 2003 issue.

[15] M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. Primo: Coupled prisms for intuitive surface modeling. In *Proceedings of Eurographics Symposium on Geometry Processing*, pages 11–20, 2006.

[16] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. Interactive skeleton-driven dynamic deformations. In *Proceedings of ACM SIGGRAPH*, pages 586–593, New York, NY, USA, 2002. ACM Press.

[17] G. Charpiat, R. Keriven, J.-P. Pons, and O. Faugeras. Designing spatially coherent minimizing flows for variational problems based on active contours. In *ICCV '05: Proceedings of the Tenth IEEE International Conferenc e on Computer Vision*, pages 1403–1408, Washington, DC, USA, 2005. IEEE Computer Society.

[18] F. R. K. Chung. *Spectral graph theory*. American Mathematical Society, 1997. Regional Conference Series in Mathematics, number 92.

[19] S. Coquillart. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *Proceedings of ACM SIGGRAPH*, pages 187–196, New York, NY, USA, 1990. ACM Press.

[20] T. A. Davis. UMFPACK - an unsymmetric-pattern multifrontal method with a column pre-ordering strategy. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004.

[21] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proceedings of ACM SIGGRAPH*, pages 317–324, 1999.

[22] T. K. Dey and W. Zhao. Approximate medial axis as a Voronoi subcomplex. In *Proceedings of 7th ACM Symposium on Solid Modeling and Applications*, pages 356–366, Saarbrüken, Germany, June 2002.

[23] I. Friedel, P. Schröder, and M. Desbrun. Unconstrained spherical parameterization. In *ACM SIGGRAPH Technical Sketch*, 2005.

[24] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3D meshes. *ACM Transactions on Graphics*, 22(3):358–363, 2003. Proceedings of ACM SIGGRAPH.

[25] X. Gu and S.-T. Yau. Computing conformal structures of surfaces. *Communications in Information and Systems*, 2:121–146, 2002.

[26] M. Hisada, A. G. Belyaev, and T. L. Kunii. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum*, 21(4):1–12, 2002.

[27] W. M. Hsu, J. F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *Proceedings of ACM SIGGRAPH*, pages 177–184, New York, NY, USA, 1992. ACM Press.

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

[29] T. J., S. Schaefer, and J. Warren. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics*, 24(3):561–566, 2005. Proceedings of ACM SIGGRAPH.

[30] K. G. Kazuya G. Kobayashi and K. Ootsubo. t-ffd: Free-form deformation by using triangular mesh. In *Proceedings of 8th ACM Symposium on Solid Modeling and Applications*, pages 226–234, 2003.

[31] L. Kobbelt, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. *Computer Graphics Forum*, 19(3):249–260, 2000.

[32] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH*, pages 105–114, 1998.

[33] F. Lazarus, S. Coquillart, and P. Jancéne. Axial deformations: an intuitive deformation technique. *Computer-Aided Design*, 26(8):607–613, 1994.

[34] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proceedings of ACM SIGGRAPH*, pages 85–94, 2000.

[35] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of ACM SIGGRAPH*, pages 165–172, 2000.

[36] Y. Lipman, O. Sorkine, D. Cohen-Or, and D. Levin. Linear rotation-invariant coordinates for meshes. In *Proceedings of ACM SIGGRAPH*, pages 479–487, 2005.

[37] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rössl, and H.-P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of Shape Modeling International*, pages 181–190, 2004.

[38] V. D. Liseikin. *A Computational Differential Geometry Approach to Grid Generation*. Springer, 1996.

[39] R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *Proceedings of ACM SIGGRAPH*, pages 181–188, New York, NY, USA, 1996. ACM Press.

[40] N. Magnenat-Thalmann, R. Laperriére, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface*, pages 26–23, 1988.

[41] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. In *www.cs.umd.edu/∼mount/ANN*, 2006.

[42] J. W. Neuberger. *Steepest Descent for General Systems of Linear Differential Equation s in Hilbert Space*. Springer, 1983.

[43] J. W. Neuberger. *Sobolev Gradients and Differential Equations*. Springer-Verlag, 1997.

[44] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[45] R. Raffin, M. Neveu, and F. Jaar. Curvilinear displacement of free-form-based deformation. *The Visual Computer*, 16(1):38–46, 2000.

[46] L. Ramshaw. Béziers and b-splines as multiaffine maps. *Theoretical Foundations of Computer Graphics and CAD*, 40:757–776, 1994. NATO ASI Series F: Computer and Systems Sciences.

[47] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *Proceedings of ACM SIGGRAPH*, pages 151–160, 1986.

[48] J.-K. Seong, G. Elber, and M.-S. Kim. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *Computer-Aided Design*, 38(3):183–193, 2006.

[49] E. C. Sherbrooke, N. M. Patrikalakis, and F.-E. Wolter. Differential and topological properties of medial axis transforms. *CVGIP: Graphical Model and Image Processing*, 58(6):574–592, 1996.

[50] K. Singh and E. Fiume. Wires: A geometric deformation technique. In *Proceedings of ACM SIGGRAPH*, pages 405–414, 1998.

[51] K. Singh and E. Kokkevis. Skinning characters using surface-oriented free-form deformations. In *Proceedings of Graphics Interface*, pages 35–42, 2000.

[52] O. Sorkine. Laplacian mesh processing. In *Eurographics:State-of-the-art report*, 2005.

[53] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry*, pages 179–188, 2004.

[54] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *Proceedings of ACM SIGGRAPH*, pages 399–405, 2004.

[55] S. Toledo, D. Chen, and V. Rotkin. TAUCS: A library of sparse linear solvers. In *www.tau.ac.il/~stoledo/taucs*, 2003.

[56] W. von Funck, H. Theisel, and H.-P. Seidel. Vector-field-based shape deformations. *ACM Transactions on Graphics*, 25(3):1118–1125, 2006. Proceedings of ACM SIGGRAPH.

[57] Z. J. Wood, M. Desbrun, P. Schneider, and D. Green. Semi-regular mesh extraction from volumes. In *Proceedings of IEEE Visualization*, pages 275–282, 2000.

[58] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. A simple approach to interactive free-form shape deformations. In *Proceedings of Pacific Graphics*, pages 471–474, 2002.

[59] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. Free-form skeleton-driven mesh deformations. In *Proceedings of 8th ACM Symposium on Solid Modeling and Applications*, pages 247–253, 2003.

[60] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with poisson-based gradient field manipulation. In *Proceedings of ACM SIGGRAPH*, pages 641–648, 2004.

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

[62] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum. Large mesh deformation using the volumetric graph laplacian. In *Proceedings of ACM SIGGRAPH*, pages 496–503, 2005.

[63] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Proceedings of ACM SIGGRAPH*, pages 259–268, 1997.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2006-RG1-001	S. Hirth, C. Karl, C. Weidenbach	Automatic Infrastructure for ..... Analysis
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A Neighborhood-Based Approach for Clustering of Linked Document Collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment Texture Editing in Monocular Video Sequences based on Color-Coded Printing Patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: Index-access Optimized Top-k Query Processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems
MPI-I-2006-4-010	T. Langer	?
MPI-I-2006-4-009	J. Gall, J. Potthoff, B. Rosenhahn , C. Schnoerr, H. Seidel	Interacting and Annealing Particle Filters: Mathematics and a Recipe for Applications
MPI-I-2006-4-008	I. Albrecht, M. Kipp, M. Neff, H. Seidel	Gesture Modeling and Animation by Imitation
MPI-I-2006-4-007	O. Schall, A. Belyaev, H. Seidel	Feature-preserving Non-local Denoising of Static and Time-varying Range Data
MPI-I-2006-4-006	C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, H. Seidel	Enhanced Dynamic Reflectometry for Relightable Free-Viewpoint Video
MPI-I-2006-4-005	S. Yoshizawa	?
MPI-I-2006-4-004	V. Havran, R. Herzog, H. Seidel	On Fast Construction of Spatial Hierarchies for Ray Tracing
MPI-I-2006-4-003	E. de Aguiar, R. Zayer, C. Theobalt, M. Magnor, H. Seidel	A Framework for Natural Animation of Digitized Models
MPI-I-2006-4-002	G. Ziegler, A. Tevs, C. Theobalt, H. Seidel	GPU Point List Generation through Histogram Pyramids
MPI-I-2006-4-001	R. Mantiuk	?
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On Verifying Complex Properties using Symbolic Shape Analysis
MPI-I-2006-1-007	I. Weber	?
MPI-I-2006-1-006	M. Kerber	Division-Free Computation of Subresultants Using Bezout Matrices
MPI-I-2006-1-004	E. de Aguiar	?
MPI-I-2006-1-001	M. Dimitrios	?
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated Retraining Methods for Document Classification and their Parameter Tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An Emperical Model for Heterogeneous Translucent Objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric Calibration of High Dynamic Range Cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A.,. Magnor, H. Seidel	Joint Motion and Reflectance Capture for Creating Relightable 3D Videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and Design of Discrete Normals and Curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse Meshing of Uncertain and Noisy Surface Scattered Data

MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-004	Y. Kazakov	A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures
MPI-I-2005-2-003	H.d. Nivelle	Using Resolution as a Decision Procedure
MPI-I-2005-2-002	P. Maier, W. Charatonik, L. Georgieva	Bounded Model Checking of Pointer Programs
MPI-I-2005-2-001	J. Hoffmann, C. Gomes, B. Selman	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-008	C. Gotsman, K. Kaligosi, K. Mehlhorn, D. Michail, E. Pyrga	Cycle Bases of Graphs and Sampled Manifolds
MPI-I-2005-1-008	D. Michail	?
MPI-I-2005-1-007	I. Katriel, M. Kutz	A Faster Algorithm for Computing a Longest Common Increasing Subsequence
MPI-I-2005-1-003	S. Baswana, K. Telikepalli	Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs
MPI-I-2005-1-002	I. Katriel, M. Kutz, M. Skutella	Reachability Substitutes for Planar Digraphs
MPI-I-2005-1-001	D. Michail	Rank-Maximal through Maximum Weight Matchings
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H. de Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching