Interactive Shape Editing

LUO XIAWU

***Abstract*— Three-dimensional geometric models are the base data for applications in computer graphics, computer aided design, visualization, multimedia, and other related fields. This report surveys state-of-the-art interactive shape editing techniques. It covers free-form deformation, mean value coordinates discrete Laplace, Poisson-based gradient algorithm and differential coordinates techniques for creating, manipulating, editing and analyzing digital geometry models, mainly on computerized modeling of discrete (digital) geometry, in particular polygonal meshes.**

***Index Terms*— mesh editing, surface deformation, free-form deformation, mean value coordinates, discrete Laplace, Poisson-based gradient algorithm, differential coordinates**

I. Introduction

Recently, mesh editing and interactive shape deformation have become a popular field of research in computer graphics.

Most researches are mainly focusing on the following topics:

- How do we interactively edit the arbitrary meshes?
- How do we avoid changing the connectivity and positions of vertices?
- How do we efficiently edit on non-linear surface?
- How do we avoid the distortion during the deformation of the shape?
- How do we preserve the local details during deformation?
- How do we avoid the too costly computation in interactive mesh editing?
- How do we invent fast and robust techniques to tackle the complicated models?
- How do we tackle the issue: neighbouring vertex does not always define a local frame (due to linear dependency)?
- How do we prevent local self-intersection in the reconstructed surface?

Consequently, many techniques/solutions have been proposed to deal with the challenges listed above, such as: free-form deformation, re-meshing, multi-resolution, non-linear differential coordinate, detail editing and transfer, non-linear Poisson algorithm, manipulate vertex positions explicitly, non-linear handle-aware isoline technique, Boolean operations, Laplacian smoothing, multi-band decompositions, Laplacian coordinates; and extended free-form deformation.

In order to deeply understand this active field, we choose the following five papers to survey:

1) Free-Form Deformation of Solid Geometric Models
2) Mesh Editing with Poisson-Based Gradient Field Manipulation
3) Mean Value Coordinates for Closed Triangular Meshes
4) Mesh Editing based on Discrete Laplace and Poisson Models
5) Differential Coordinates for Interactive Mesh Editing

Each paper is discussed in a new section commencing from II. We focus mainly on learning its objectives, technologies involved and presenting some results. In section VII, we perform some simple comparison on the methods and algorithms involved in each paper, and conclusion is made in section VIII.

II. Free-Form Deformation of Solid Geometric Models [1]

A. *Objectives:*

- It uses FFD technique to deal with solid geometric modeling.
- It can be applied to any geometric model, such as: deformed polygonal data sphere intersected by a plane.
- It keeps continuity, C^0, C^1, C^2.
- It will have volume changes.
- The volume is preserved in FFD.
- It has flexibility applications, like:
  ① Molding a rounded bar into the telephone handset.
  ② It can be applied to the common boundary of the two patches, results in the slope continuous between them.
  ③ It creates handles by using single FFD for Trophy.
  ④ It can be used as fairing/duct surfaces.
  ⑤ Artist display of solid objects has the property of preserving volume.

This paper provides a free-form manner which based on trivariate Bernstein Polynomials to deform solid geometric models. The techniques involved can be used in CSG or B-rep solid modeling systems, to deform planes, quadrics, surfaces (B-spline parametric patches and implicit surface).

One of its applications is to apply the local deformation to any desired degree curves. It also has a property of preserving volume during deformations.

The author summarized the background of solid modeling and surface modeling technologies in the past fifteen years. In the paper, the author also indicated that free-form surfaces are mostly used in surface modeling, and discussed the category of the problems in defining a solid geometric model with free-form surfaces into three phases:

- Combining existing free-form surface and solid modeling techniques
- Trivariate parametric hyperpatch
- Implicit surfaces

B. Technology it involved:

1.) Tensor product trivariate Bernstein polynomial.
2.) Local coordinate system on a parallel piped region.

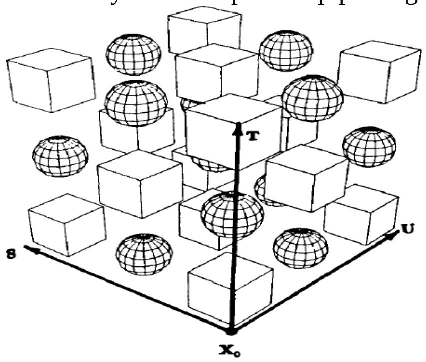

Fig.1 [1]

- Definitions:

① Point X has (s, t, u) coordinates.

$$X = X_0 + sS + tT + uU$$

② By using the linear algebra, we can get the (s, t, u) coordinates of X, as below:

$$s = \frac{T \times U \cdot (X - X_0)}{T \times U \cdot S}, t = \frac{S \times U \cdot (X - X_0)}{S \times U \cdot T}, u = \frac{S \times T \cdot (X - X_0)}{S \times T \cdot U}$$

(0<s<1, 0<t<1 and 0<u<1)

③ A grid of control points P_{ijk} on the parallelepiped with $l+1$ planes in S direction, $m + 1$ planes in T direction and $k + 1$ planes in U direction.

$$P_{ijk} = X_0 + \frac{i}{l}S + \frac{j}{m}T + \frac{k}{m}U$$

④ During deformation, the new point X_{ffd} of the arbitrary point X can be calculated by its (s, t, u) coordinates and trivariate Bernstein polynomial:

$$X_{ffd} = \sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^t (\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j (\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k P_{ijk}))$$

- Continuity control

Maintaining the cross-boundary derivative continuity for two or more FFDs in a piecewise manner is possible, which can be described in local surface parameterization.

After denoting the local parameters by v, w, we express the surface as: $(s, t, u) = (s\,(v, w), t\,(v, w), u\,(v, w))$, and assume two FFDs $X_1(s_1, t_1, u_1)$ and $X_2(s_2, t_2, u_2)$ share a common boundary $s_1 = s_2 = 0$, then the first derivative can be expressed as:

$$\frac{\partial X_1(v,w)}{\partial v} = \frac{\partial X_1}{\partial s} \cdot \frac{\partial s}{\partial v} + \frac{\partial X_1}{\partial t} \cdot \frac{\partial t}{\partial v} + \frac{\partial X_1}{\partial u} \cdot \frac{\partial u}{\partial v}$$

$$\frac{\partial X_1(v,w)}{\partial w} = \frac{\partial X_1}{\partial s} \cdot \frac{\partial s}{\partial w} + \frac{\partial X_1}{\partial t} \cdot \frac{\partial t}{\partial w} + \frac{\partial X_1}{\partial u} \cdot \frac{\partial u}{\partial w}$$

And the sufficient condition for the first derivative is:

$$\frac{\partial X_1(0,t,u)}{\partial s} = \frac{\partial X_2(0,t,u)}{\partial s}$$

$$\frac{\partial X_1(0,t,u)}{\partial t} = \frac{\partial X_2(0,t,u)}{\partial t}$$

$$\frac{\partial X_1(0,t,u)}{\partial u} = \frac{\partial X_2(0,t,u)}{\partial u}$$

The author indicated the first derivative continuity and higher derivative continuity can be shown to be straightforward extension of the continuity conditions for Bezier curves and tensor product Bezier surfaces.

- Local Deformations

Reference [1] provides an example to demonstrate the continuity condition can perform local, isolated deformation. It is easy to show that sufficient conditions for a C^k local deformation are that the control points on the K planes adjacent to the interface planes will not be moved.

This local deformation can be used hierarchically.

- Volume Change

Another technology highlighted in this paper is the Jacobian of FFD, which can be used to control over the volume change.

Let us assume the FFD is defined as below:

$$F(x, y, z) = (F(x, y, z), G(x, y, z), H(x, y, z))$$

Then its determinant is the Jacobian.

$$Jac(F) = \begin{pmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} & \frac{\partial F}{\partial z} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \\ \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} & \frac{\partial H}{\partial z} \end{pmatrix}$$

The volume will change to $J_{ac}(F(x, y, z)) \cdot d_x \cdot d_y \cdot d_z$ from the original $d_x \cdot d_y \cdot d_z$ after the deformation.

The volume of the whole deformed solid is simply the triple integral of the above differential volume over the volume enclosed with the undeformed surface, so we can get the bound on the $J_{ac}(F)$ over the region of deformation, we will have a bound on the volume change.

The bound is also easy to get if $J_{ac}(F)$ is expressed as a trivariate Bernstein polynomial, the largest and smallest polynomial coefficients are the upper and lower bounds on the volume change.

Another property of FFD is the volume will be preserved if $J_{ac}(F) \equiv 1$. The solid model will keep the original volume under such a transformation.

C. Present some results:

Reference [1] demonstrates the technologies are robust to define the FFD solid models; this model has several advantages, as below:

- It is easy to use, especially for strong sculpturing metaphor.
- It can be used with any solid modeling scheme, as well as applied to surfaces or polygonal models.

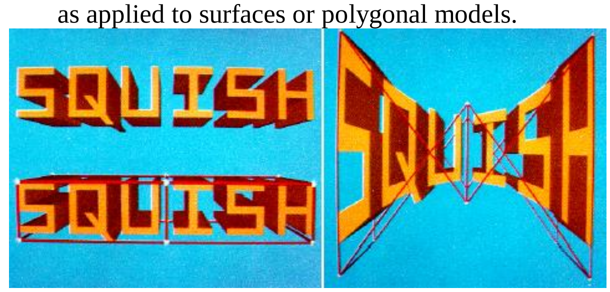

Fig.2 [1]

- It has properties on the degree of volume change, and volume preserving.
- Under FFD, parametric curves and surfaces remain parametric characters.

Fig.3 [1]

- It works with surfaces of any degree or formulation.
- It can be applied globally or locally, with derivative continuity.

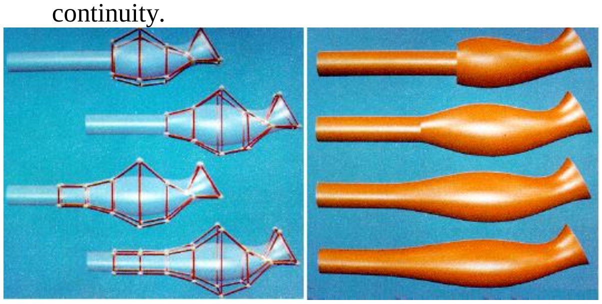

Fig.4 [1]

- It can be used widely, aesthetic surfaces, many fairing surfaces and some functional surfaces.

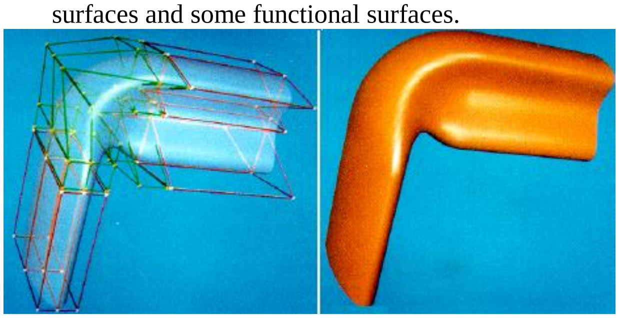

Fig.5 [1]

FFD has some limitations:

- It does not work for general filleting and blending.
- In tackling the arbitrary boundary curve, it will cost much more, because user will have to do duplicate deformation.
- Modeling operations, like subdivision on trivariate Bernstein polynomials, cost much more than on bivariates.

III. Mesh Editing with Poisson-Based Gradient Field Manipulation [2]

A. *Objectives:*

In order to use Poisson equation, we need to fix two issues:

- First, how do we apply Poisson equation to mesh geometry? As the unknown in the Poisson equation is a scalar function, while the mesh is a vector function with unique differential properties.
- Second, how do we modify two types information of gradient fields and boundary condition which are mathematical concepts in Poisson equation to achieve desirable mesh editing effects?

Reference [2] introduces a method to mesh editing with Poisson equation, with the gradient field manipulation, which implicitly modifies the vertex positions.

Editing a function can be achieved by modifying the gradient field and boundary condition, then performing a reconstruction using the Poisson equation.

B. *Technology it involved:*

1. Mathematic Techniques

The Poisson Equation:

$$\nabla^2 f = \nabla \cdot \boldsymbol{w}, \qquad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Vector Field Decomposition:

Poissson equation is closely related to Helmholtz-Hodge vector field decomposition:

$$\boldsymbol{w} = \nabla_{\emptyset} + \nabla \times v + \boldsymbol{h}$$

The scalar potential field $\emptyset$ from this decomposition happens to be the solution of the following least-squares minimization:

$$\min_{\emptyset} \iint_{\Omega} \|\nabla_{\emptyset} - w\|^2 \, dA$$

Discrete Fields and Divergence:

$$(Div\boldsymbol{w})(v_i) = \sum_{T_k \in N(i)} \nabla B_{ik} \cdot \boldsymbol{w} |T_k|$$

For the descriptions of all the parameters in above formulas, please see [2].

2. Three Key Technologies

There are mainly three key technologies/contributions that are introduced in [2]:

- A basic mesh solver based on the Poisson equation, which has following two characters:
  ① It can reconstruct a scalar function from a guidance vector field and a boundary condition.
  ② Poisson equation is an alternative formulation of a least-squares minimization.
- A gradient field manipulation scheme function by using local transforms
- The generalized boundary condition representation based on local frames.

3. A basic Poisson Mesh Solver

There are three coordinates which can be defined as three scalar fields in a parameterization, the piecewise linear mesh defined with the discrete potential field.

The same topologies are located in the target and parameterization meshes, and the vertices in the mesh have one-to-one correspondence.

We can answer the question: How do we solve an unknown target mesh with unknown geometry but known topology?

For each of the three coordinates, the Poisson equation requires a discrete guidance vector field, which can be got by the parameterization mesh, and then its divergence can be computed at a vertex in the parameterization mesh.

b can be obtained from the collection of divergence values of all vertices, co-efficient matrix A is independent of the guidance filed, which can only be obtained from the parameterization mesh.

We can get one specific coordinate for all vertices at the same time by solving the result of the linear system, and can use Poisson equation to get the coordinates, for 3D, we need to calculate 3 times.

Different parameterization meshes are mapped to different target meshes, in calculating the edited mesh, the general rule is that guidance vectors associated with larger triangles in the parameterization mesh are better approximated than those associated with the smaller triangles.

The role for the area of the triangles is: it will be used as the weighting scheme, the input is the given parameterization meshes and the goal is to obtain an edited mesh, without any 2D parameterization.

4. Gradient Field Editing Using Local Transforms

Manipulating mesh gradient fields is a key component of the mesh editing system. It has following four applications.

Mesh Deformation: It is divided into two types on the boundary conditions. The first is fixed boundary condition and the other is editable boundary condition.

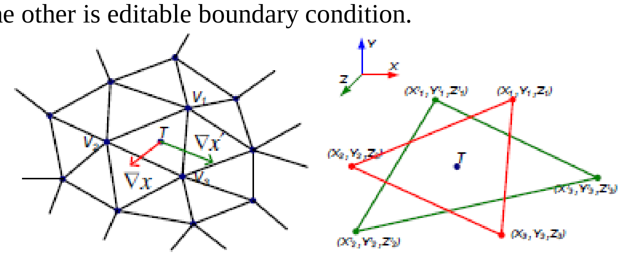

Fig.6 [2]

The vertices in the fixed boundary condition have the identify matrix as their local transform.

The vertices on the same editable curve can be manipulated either individually or simultaneously.

It includes simultaneous translation and the rotation, which mean all the vertex normals around their respective tangent directions with the same degree.

By default, the Poisson equation only keeps modified vertex positions in the boundary condition, but suppresses changes of the orientation and scale at the free vertices, while the technique in this paper can preserve small-scale features.

Acceleration for interactive deformation

How do we tackle the case when there are much larger vertices in the mesh? It will be impossible to use the original resolution to solve the equation. There are two ways to accelerate this process.

- Pre-compute A^{-1} by using *LU* decomposition at every frame to dynamically execute the back substitution step. However *LU* decomposition has two shortcomings of less stable and do not preserve the sparse structure of A.
- Building multi-resolution meshes pyramid acceleration for large meshes.

Mesh merging and Assembly

There are seven steps to merge two meshes:
① Obtaining each mesh's boundary and vertex correspondence.
② Computing the local frames along the two boundaries.
③ Obtaining an intermediate boundary, including both vertex positions and local frame by simply choosing one of the original two boundaries or interpolating the original two vertex correspondence.
④ Modifying the mesh connectivity along two meshes' boundaries according to the intermediate boundary.
⑤ Obtaining two sets of quaternions by comparing the local frames at the intermediate boundaries and the ones at the original two boundaries.
⑥ Propagating the two sets of quaternions respectively towards the interior of both meshes.
⑦ Constructing the linear system for all the vertices of both meshes and solving it to get a merged mesh.

Since it is not automatically available for the correspondence between the boundaries, there are three interactive tools are designed for the user to choose.

- Obtain the second boundary by projecting the first mesh's boundary onto the second mesh.
- Next obtain the planar parameterization of the boundary curve on the first mesh. The using mapping scheme to map this 2D boundary onto the second mesh.
- Lastly obtain the dense correspondence after interactively defining sparse key vertex correspondences between the two boundaries.

The first tool is most restrictive but it needs the little users' interaction; the last one is most powerful but it needs much users' interaction.

Advantages:

- Two mesh boundaries can be any different shapes, sizes and roughness.
- The shapes become more compatible with each other because the propagation of the local frame changes can adjust the two meshes globally.

Mesh Smoothing and De-noising

The algorithm user provides the bilateral filtering on the normal instead of vertex, which makes the feature preserving mesh smoothing Bilateral filters have two parameters σ_f and σ_g, the former one controls the spatial weight and the latter one defines the amount of allowed normal variation.

The algorithm will shift vertex positions by using the Poisson equation to reflect the altered normal once the smoothed normals have been obtained.

The engine facilitates a linear method to obtain the vertex positions from normals.

A local rotation matrix is defined from the minimal rotation angle and its associated axis that can transform the original normal to the new one.

A new triangle and its new gradient vectors will be obtained by applying the local rotation to the original triangle.

Advantages:

- This way is more intuitive since the normal typically changes abruptly at edges and creases.
- This algorithm can be applied once or multiple iterations to a mesh.
- The solution obtained can be used directly or used as the initialization for further no-linear optimization.
- It can be used for mesh denoising and smoothing. In the smoothing, the bilateral filter will be replaced with a regular Gaussian filter for normals which can avoid preserving small feature and artifacts.

C. Present some results:

Advantages:

The approach introduced in this paper has a perfect feature of modifying the original mesh geometry implicitly through gradient field manipulation.

It can produce acceptable and desirable results for both global and local editing operations, like: smoothing, object merging and deformation.

- Simultaneous translation.

Fig.7 [2]

- Simultaneous rotation of all the vertex normal around their respective tangent directions by the same degree.

Fig.8 [2]

- Comparison: from left to right. Left: technology introduced in this paper, Middle: Naïve Poisson. Right: WIRE.

Fig.9 [2]

The Poisson mesh solver based interactive tool is robust to use for high-end application which needs superior results.

- Detailed editing by individually manipulating vertices on curves.

Detailed editing is applied the eyes, eyebrows and lips.

Local smoothing is applied to the cheeks and the groove.

Fig.10 [2]

- Mesh merging and Assembly, by projection scheme.

(1) The WING (2000 faces), the HORSE (100K faces), running time = 400 ms.

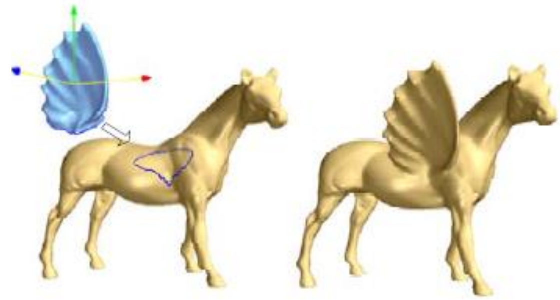

Fig.10 [2]

(2) Two meshes components are merged at their jagged boundaries.

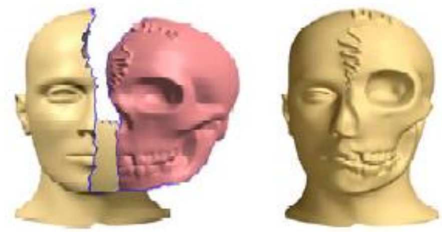

Fig.11 [2]

- Object merging using mapping.

(1) GARGOYLE has 4000 faces, TEAPOT has 2000 faces, and the running time is 890 ms.

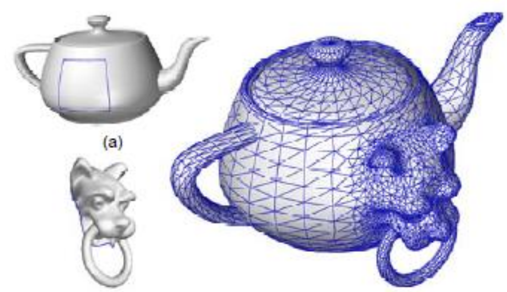

Fig.12 [2]

(2) The front facing half of the DRAGON model (18K faces), CYLINDER (60K faces), the running time is 5 seconds.

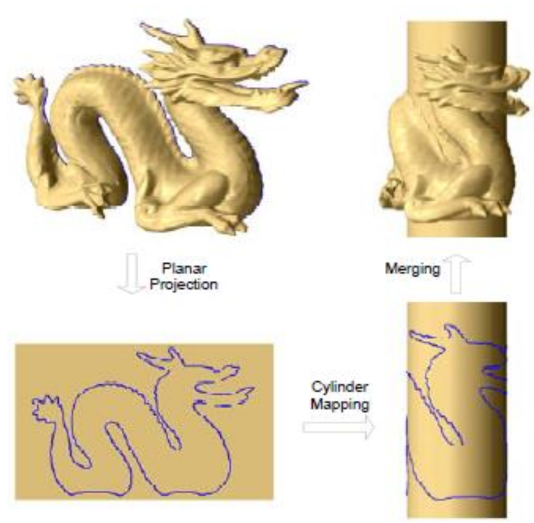

Fig.13 [2]

- Mesh Smoothing and Denoising

(1) Original DRAGONAL model (150K vertices), with only one iteration with σ_f=4.0 and σ_q=0.2π.

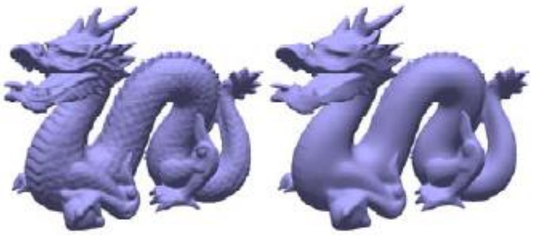

Fig.14 [2]

(2) Gaussian noise mode, with three iterations with σ_f=3.0 and σ_q=0.2π.

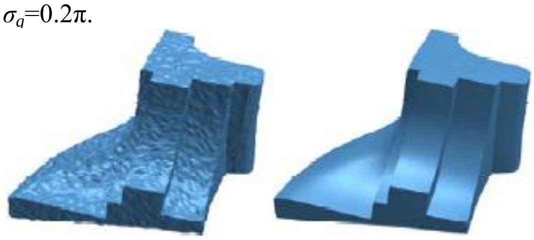

Fig.15 [2]

(3) Smoothing merging boundary.

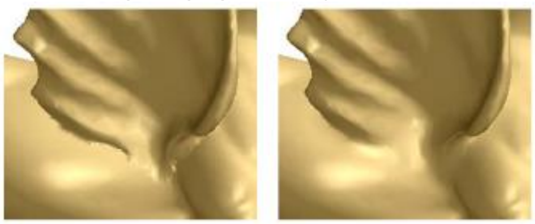

Fig.16 [2]

Limitations:

The Poisson equation can only guarantee C^0 continuity between constrained and free vertices.

The performance of the user-guided deformation is not perfect. There is a need for improvement.

IV. Mean Value Coordinates for Closed Triangular Meshes [3]

A. *Objectives:*

Mean value coordinates is an excellent method to construct an interpolation for closed polygons. In this paper, the author generalized mean value coordinates from closed 2D polygons to closed triangular meshes, and constructs 3D mesh value coordinates.

In terms of the positions and associated values of meshes vertices, an efficient and stable method is created for evaluating the resulting mean value interpolant.

The coordinates in a given a mesh P are continuous everywhere and smooth on the interior of P.

They provide a generalization for arbitrary closed surfaces; the resulting interpolant has linear precision and are well-behaved.

B. *Technology it involved:*

1. Gouraud shading computes intensities at the vertices of a triangle and extends these intensities to the interior by using linear interpolation.

$$\hat{f}[v] = \frac{\sum_j w_j f_j}{\sum_j w_j}$$

2. Mesh parameterization and freeform deformation methods also use above type interpolate, set the data values f_j to be their associated vertex positions P_j, the interpolate will reproduce linear functions:

$$v = \frac{\sum_j w_j p_j}{\sum_j w_j}$$

Coordinate function $\frac{w_j}{\sum_j w_j}$ are the desired affine combination.

Wachspress' interpolate does not generalize to non-convex polygons, and the interpolate it yields will have poles.

3. Floater proposed a new type of interpolate based on the mean value theorem, which generates smooth coordinates for star-shaped polygons.

$$w_j = \frac{\tan\left[\frac{\alpha_{j-1}}{2}\right] + \tan\left[\frac{\alpha_j}{2}\right]}{|p_j - v|}$$

Hormann shows the mean value coordinates reproduce linear functions everywhere.

Mean Value Interpolation

We construct a function $\hat{f}[v]$ where $v \in R^3$ that interpolates $f[x]$ on P for all x.

① Projecting a point $p[x]$ of P onto the unit sphere S_v centered at v.

②Weighting the point's associated value $f[x]$ by $\frac{1}{|p[x]-v|}$, and integrate this weighted function over S_v, by this way, it can ensure affine invariance of the resulting interpolate.

③ Gathering all pieces together, then getting the Mean Value Interpolate:

$$\hat{f}[v] = \frac{\int_x w[x,v] f[x] dS_v}{\int_x w[x,v] dS_v}$$

④ $w[x,v]$ is exactly $\frac{1}{|p[x]-v|}$

There are 3 important properties of the resulting mean value interpolate:

- Interpolation
- Smoothness
- Linear precision

⑤ The integral of the unit normal over a sphere is exactly zero (because of the symmetry property)

$$\int_x \frac{p[x]-v}{|p[x]-v|} dS_v = 0$$

⑥ v value can be obtained as below, because of $\frac{p[x]-v}{|p[x]-v|}$ is the unit normal to S_v at parameter value x.

$$v = \int_x \frac{p[x]}{|p[x]-v|} dS_v \Big/ \int_x \frac{1}{|p[x]-v|} dS_v$$

⑦ Shortcomings: if P is a convex shape, then the coordinate functions are positive for all v inside P, otherwise it is negative because the orientation of ds_v is negative.

Coordinates for Piecewise Linear Shape

In terms of vertex positions and their associated function values, a simple and closed form solution for piecewise linear shapes is created.

1. Mean Value coordinates for closed polygon.

- The edge of a closed polygon P with vertices $\{p1, p2\}$ and associated values $\{f_1, f_2\}$ can be linear parameterized by:

$$p[x] = \sum_i \emptyset_i [x] p_i$$

- The data values f_1 and f_2 can also be linearly parameterized by:

$$f[x] = \sum_i \emptyset_i [x] f_i$$

- Projecting the edge E onto the unit circle to form the circular arc $\bar{E}$.

$$\frac{\int_x w[x,v] f[x] d\bar{E}}{\int_x w[x,v] d\bar{E}} = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

- From the following knowledge,

$$\sum_i w_i (p_i - v) = m$$

$$m = \tan[\alpha/2]\left(\frac{(p_1 - v)}{|p_1 - v|} + \frac{(p_2 - v)}{|p_2 - v|}\right)$$

We can get the value for w_i

$$w_i = \tan[\alpha/2]/|p_i - v|$$

- The interpolate for all edges can be obtained:

$$\hat{f}[v] = \frac{\sum_k \sum_i w_i^k \cdot f_i^k}{\sum_k \sum_i w_i^k}$$

w_i^k and f_i^k are weights and values associated with edge E_k.

2. Mean Value coordinates for closed triangular meshes.

We use the same process as what we did for the closed polygons, we can get the formulas for tackling close triangular meshes T, with vertices $\{p_1, p_2, p_3\}$, and associated values $\{f_1, f_2, f_3\}$

$$w_i = \int_x \frac{\phi_i[x]}{|p[x]-v|} d\bar{T}$$

Derive the value for w_i by inverting equation in the matrix form:

$$\{w_1, w_{21}, w_3\} = m\{p_1 - v, p_2 - v, p_3 - v\}^{-1}$$

By using the theorem: the integral of outward unit normal over a closed surface is always exactly zero [3], the value for m can be obtained:

$$m = \sum_i \frac{1}{2} \theta_i n_i$$

Then we can get the value for w_i:

$$w_i = \frac{n_i \cdot m}{n_i \cdot (p_i - v)}$$

There might be a possibility of obtaining a negative value for w_i, because projecting a triangle T onto S_v may reverse its orientation.

The mean vector points towards the projected spherical triangle $\bar{T}$ when the orientation of $\bar{T}$ is positive, otherwise points away from the $\bar{T}$.

Robust mean value interpolation

There are three steps for calculating the mean value interpolation on triangular meshes.

- Compute the mean vector m.
- Compute the weights w_i.
- Respectively by adding $\sum_i w_i$ and $\sum_i w_i f_i$.

There are two obstacles:

1) During calculating w_i, the denominator will have a chance of zero when the point v lies on the same plane of the face T.
2) Numerical instability in computing w_i for triangle T with small projected area.

The following are ways to tackle the above mentioned issue:

1) There are two different cases

- If v lies inside T, the algorithm implies $\hat{f}[v]$ converges to $f[x]$. We can use whether $\sum_i \theta_i = 2\pi$ to test this.
- Otherwise w_i will be zero, because the integral definition $\int \frac{\phi_i[x]}{|p[x]-v|} d\bar{T}$ will be zero, can be tested by checking whether any of the dihedral angles ψ_i or $sin[\psi_i]$ are zero.

2) We uses a stable formula to compute weights:

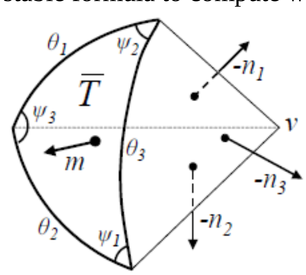

Fig.17 [3]

$$w_i = \frac{\theta_i - \cos[\psi_{i+1}]\theta_{i-1} - \cos[\psi_{i-1}]\theta_{i+1}}{2\sin[\psi_{i+1}]\sin[\theta_{i-1}]|p_i^k - v|}$$

Here, we use half-angle formula for spherical triangles.

$$\cos[\psi_i] = \frac{2\ \sin[h]\sin[h-\theta_i]}{\sin[\theta_{i+1}]\sin[\theta_{i-1}]} - 1$$

C. *Present some results:*

It is widely used, such as:

① Performing boundary value interpolation.

- Function values are associated with the vertices of the mesh.
- The function is smooth, and a linear function on the faces of the triangles.

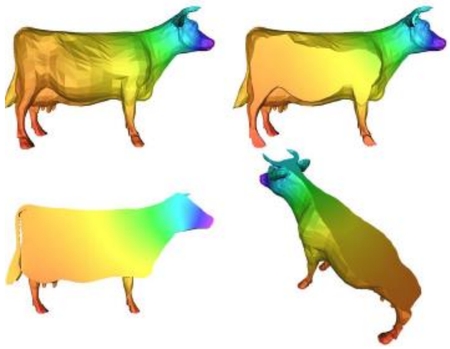

Fig.18 [3]

② Constructing volumetric textures

- The texture coordinates (u_i, v_i) will be used as *fi* for each vertex.
- It will interpolate the texture coordinates on the vertices and along the polygons of the mesh.
- Constructing a volumetric texture.

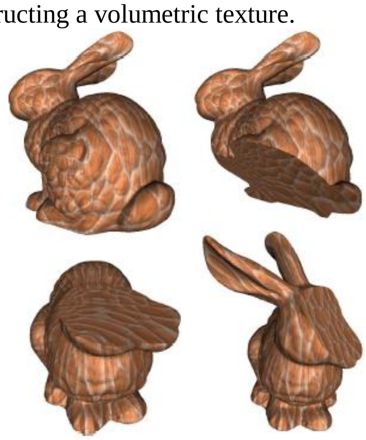

Fig.19 [3]

③ Surface deformation.

⑴ Deformation are applied to the HORSE model.

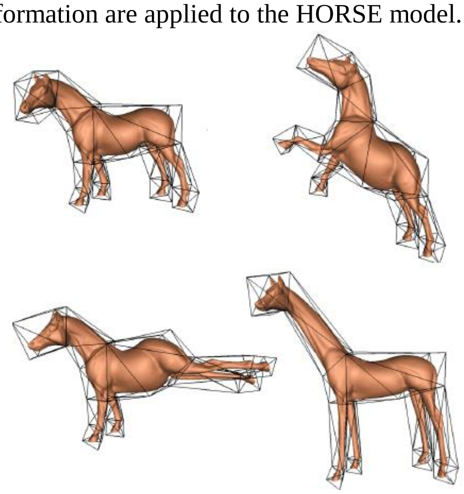

Fig.20 [3]

⑵ Deformation are applied to the monsters model

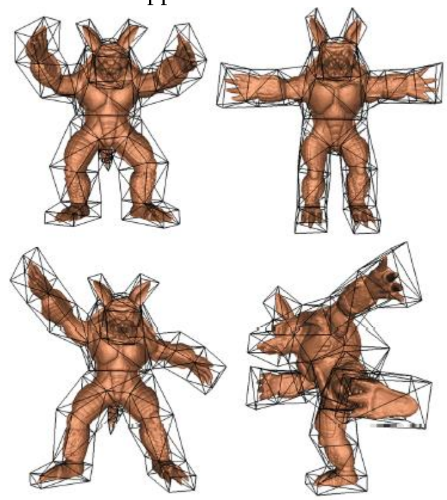

Fig.21 [3]

- Computing the mean value weight function w_j with respect to each vertex p_j in the un-deformed control mesh, for each vertex v in the model.
- Moving the vertices of the control mesh to include the deformation on the original surface.

$$\hat{v} = \frac{\sum_j w_j \hat{p}_j}{\sum_j w_j}.$$

④ Pre-compute as much information as possible can speed up the computations, as shown in the below table, the experiments are performed on 3GHz Intel Pentium 4 computer.

Model	Tris	Verts	Eval/s
Horse control mesh (fig 1)	98	51	16281
Armadillo control mesh (fig 7)	216	111	7644
Cow (fig 5)	5804	2903	328
Bunny (fig 6)	69630	34817	20

Fig.22 [3]

Advantages:

- It is well defined on both the interior and exterior of the mesh.

Futures' work:

- Exploring the feature: perform a job of extrapolating value outside of the mesh.
- Investigating the relationship between Wachspress coordinates and the mean value coordinate.

Limitation:

- It only considers the meshes that have triangular faces, does not start work on the mean value coordinates for piecewise linear mesh with arbitrary closed polygons as faces yet.

V. Mesh Editing Based on Discrete Laplace and Poisson Models [4]

A. *Objectives:*

Explicit representation: based on points, vertices or nodes are described by absolute Euclidean coordinates.

Implicit representations: describe the shape as the level set of a function defined in Euclidean space.

The author raised a concept that geometric detail is an intrinsic property of the surface, so the surface editing will be best got from operating on an intrinsic surface representation, which can be derived from differential properties of mesh, and this way will result in a Poisson model for the modeling process which poses the nonzero boundary constraints.

After discussing the shortcomings of the multi-resolution mesh method, the author raised to use differential coordinate to encode the geometric details, by this way, the reconstruction can preserve the intrinsic geometry very well.

Laplacian representation is used to implement above idea, involved in developing a solution of a sparse linear system. This will be applied in three aspects:

- Interactive free-form deformation in ROI based on the transformation of the handle.
- Mixing and transferring of geometric details between two surfaces.
- Transplanting a partial surface mesh into another surface.

B. Technology it involved:

It mainly includes two parts: Transitional regions and mapping.

The Laplacian representation

1. The discretized version of the Laplace operator for meshes is:

$$\delta_i = \mathbf{v}_i - c_{ij} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j \quad , \quad \sum_j c_{ij} = 1$$

2. To solve a linear system of equations.

Let $C = \{c_{ij}\}$, then the transformation between V and Δ is:

$$\Delta = (I - C)V$$

Mesh Modeling Framework

1. The deformed geometry V' is defined as:

$$\min_{V'} \sum_{i=1}^{n} \left\| \delta_i - \left(\mathbf{v}'_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \mathbf{v}'_j \right) \right\|^2$$

With above least square sense function, the mesh modeling framework will preserve differential properties of the original geometry, it has two cases:

- If the original surface was a membrane, the constraints will become: $L^2 V = 0$.
- Otherwise, the right-hand side is non-zero, and it will become a variant of the discrete Poisson modeling way.

2. The easiest way in implementing the approach by $\mathbf{A}\mathbf{V}' = \mathbf{b}$ and $\mathbf{A}^T\mathbf{A}\mathbf{V}' = \mathbf{A}^T\mathbf{b}$, yields:

$$w_i \| \mathbf{v}'_i = \hat{\mathbf{v}}_i$$

3. for the arbitrary point on the triangle mesh, the positional constraints $\hat{\mathbf{v}}_{ij}$ is:

$$(1-\lambda)v'_i + \lambda v'_j = \hat{\mathbf{v}}_{ij}$$

4. The differentials can be prescribed as below:

$$\mathbf{v}'_i - \sum_{j \in \mathcal{N}_i} c_{ij} \mathbf{v}'_j = \hat{\delta}_i$$

δ_i points roughly to the normal direction of vertex i, and its length is proportional to the mean curvature.

Modeling Operation

The operations are mainly focusing on ROI, and adding additional anchor vertices on several layers.

In the system matrix A, the anchor vertices generate the positional constraints $\mathbf{v}'_i = \hat{\mathbf{v}}_i$ which warrants a soft transition between the fixed part of the mesh and the altered ROI.

Incorporating linear transformations

1. We can assign each vertex i an individual transformation T_i, then transform each local Laplacian δ_i with T_i, as below:

$$\min_{V'} \sum_{i=1}^{n} \left\| T_i \delta_i - \left(\mathbf{v}'_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \mathbf{v}'_j \right) \right\|^2$$

2. $T_i\delta_i$ is contained in the right-hand side column vector b in the function $\mathbf{A}\mathbf{V}' = \mathbf{b}$. The system A is independent to the T_i, which means T_i can be changed during interactive modeling.

There are four types usage for T_i:

1) Prescribing the transformations: with the formula $T_i = R_i S_i$, each local rotation and scale are a distance-weighted average of given transformation; rotation and scale are treated independently.

2) The transformations from the membrane solution.

- Solving $\Delta V' = 0$ firstly.
- Computing each transformation T_i based on comparing the one-rings in $\mathbf{V}$ and $\mathbf{V'}$ of vertex i.
- T_i can be got from:

$$\min_{T_i} \left(\| T_i \mathbf{v}_i - \mathbf{v}'_i \|^2 + \sum_{j \in \mathcal{N}_i} \| T_i \mathbf{v}_j - \mathbf{v}'_j \|^2 \right)$$

- The minimizer is a linear function of V'.
- A has to be factored once.
- T_i will be computed.
- b is modified accordingly.
- Using back-substitution to get the final positions V'.

3) The linearized implicit transformations

1. for each vertex i, we compute an appropriate transformation T_i based on the eventual characters of the new vertices V', so $T_i(V')$ is a function of V'.

- T_i will be got by solving for V', if the coefficients of T_i are a linear function.
- If T_i is unconstrained, we can use the following representation of the exponential to avoid geometric details losing.

$$s \exp H = s(\alpha I + \beta H + \gamma \mathbf{h}^T \mathbf{h})$$

- T_i can be described as below because the constrained transformations are a linear approximation.

$$T_i = \begin{pmatrix} s & h_1 & -h_2 & t_x \\ -h_1 & s & h_3 & t_y \\ h_2 & -h_3 & s & t_y \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- We need to minimize the following function when we express the vector of the unknowns in T_i as $(s_i, \mathbf{h}_i, \mathbf{t}_i)^T$.

$$\| A_i(s_i, \mathbf{h}_i, \mathbf{t}_i)^T - \mathbf{b}_i \|^2$$

- Then the linear square problems will be solved by the following function

$$(s_i, \mathbf{h}_i, \mathbf{t}_i)^T = \left(A_i^T A_i\right)^{-1} A_i^T \mathbf{b}_i$$

When A_i and $\boldsymbol{b}_i$ are constructed by the known geometry values.

$$A_i = \begin{pmatrix} v_{k_x} & v_{k_y} & -v_{k_z} & 0 & 1 & 0 & 0 \\ v_{k_y} & -v_{k_x} & 0 & v_{k_z} & 0 & 1 & 0 \\ v_{k_z} & 0 & v_{k_x} & -v_{k_y} & 0 & 0 & 1 \\ \vdots & & & & & & \end{pmatrix}, k \in \{i\} \cup N_i$$

$$\mathbf{b_i} = \begin{pmatrix} v'_{k_x} \\ v'_{k_y} \\ v'_{k_z} \\ \vdots \end{pmatrix}, k \in \{i\} \cup N_i$$

4) T_i can be adjusted by following steps:

- Computing $\{T_i\}$ from V and V'.
- Inspecting T_i, which will update the corresponding Laplacian coordinates δ_i appropriately according to the effect achieved.
- Solving the system again.

Mesh Editing

The editing steps are listed here from the user side:

- Defining ROI (Region of interest) for editing.
- Selecting the handle vertex.
- Optionally defining stationary anchors vertices, which are transition vertices between untouched part of the mesh and the ROI.
- Moving the handle vertex

The steps from the algorithm point of view are also listed:

Once ROI, vertices for the stationary anchors and handle vertex are defined. Vertices will be divided into two groups: the modified vertices and the untouched vertices of the mesh.

- The handle vertex will be updated at once.
- The unmodified vertices will be forced to follow the user inputs to represent the overall shape.
- Use the least-squares solution method to generate a soft blend between the ROI and the fix part of the mesh.
- Choose several layers of anchors to improve the smoothness; these anchors have the weights which are proportional to the geodesic distance from their handle.
- Then the edited surface will be reconstructed from the locally rotated differential coordinates, but it needs a priori estimation on the normal of the editing result.
- Reconstruct the surface by solving the linear least-square system.

In summary, the computational kernel of the algorithm is the sparse linear solver for the least-squares problems; and minimizing $\|A\mathbf{x}-\mathbf{b}\|$ over the modified region of the surface.

Detail Transfer

1. It extracts the surface from the source geometry and transfers it into the target surface.

2. The details of a surface are encoded based on the Laplacian representation.

$$\xi_i = \delta_i - \tilde{\delta}_i$$

δ_i and $\tilde{\delta}_i$ are the Laplacian coordinates of the vertex i in S and $\tilde{S}$, where $\tilde{S}$ is the smooth version of S.

3. S can be reconstructed with inverse Laplacian transform L^{-1} by the following formula. By this way, the original details can be recovered.

$$S = L^{-1}(\tilde{\delta} + \xi)$$

4. It uses the following important property of the Laplacian coordinates to tackle the case, adding the details ξ onto an arbitrary surface U.

$$R \cdot L^{-1}(\delta_j) = L^{-1}(R \cdot \delta_j)$$

5. Under the assumption that the target surface T and the source surface S share the same connectivity, and the correspondence between the different geometries, we will generalize this for the arbitrary surfaces.

6. The detail transfer from S onto U will be expressed as followings, then the new surface U' will have the details of U.

$$U' = L^{-1}(\Delta + \xi')$$

Assume we have had the following knowledge:

- The rotation operator R_i will be defined by $\mathbf{n}_s \times \mathbf{n}_u$ and $\mathbf{n}_u = R_i(n_s)$ where $\mathbf{n}_s$ and $\mathbf{n}_u$ are the normals related to the orientations of i in S and U respectively.
- The rotated detail encoding of vertex i is defined by $\xi'_i = R_i(\xi_i)$.
- The Laplacian coordinates of the vertices of U is denoted as Δ.

Mapping and Resampling

1. In order to sample the Laplacian coordinates, the mapping between the two surfaces need to be defined.

2. The mapping is generated by parameterizing the meshes over a unit circle or the unit square, because the patches are assumed to be homeomorphisms.

3. In this paper the author chooses the mean-value coordinates parameterization because it produces a quasi-conformal mapping efficiently and it is valid for convex domains.

4. A radial basis function elastic warp is used to tackle the case where it requires a more careful correspondence.

- Firstly mapping the 1-ring of i onto S using the parameterization.
- Computing the Laplacian from the above mapped 1-ring.
- Getting the similar results, while it leads to some 'blurring'.
- It is simpler and it has no other distortion.
- It doesn't need the more special treatment.

5. The corresponding ξ'_i is the difference of the sampled Laplacian coordinates in S and $\tilde{S}$.

Mixing Details

Apply the above mentioned transfer method onto the surfaces of the two given meshes with different details and the same connectivity, we can generate a third target mesh. The global shape of this target mesh is deformed respectively and the details are gradually mixed.

Transplanting surface patches

1. The operation includes: topological and geometrical.
 - Topological operation creates a consistent triangulation from the two sub-meshes' connectivity.
 - Geometrical operation creates a gradual change of the geometrical structure of one shape into the other. This is based on Laplacian coordinates and the reconstruction method.

2. It requires the user to register the two parts in world coordinates.

3. Selecting a region $U_0 \in U$. cutting it, and then the remained boundary is homeomorphic of S boundary.

4. Projecting the boundary of S to U_0.

5. The connectivity of the resulting mesh D is created when the two boundary loops are zipped.

6. Creating the transitional regions:

7. Creating the mapping: parameterize both meshes over the unit square.

8. Sampling the Laplacian coordinates of S' and U' over D'. It also has more precision ways by defining the coordinate along the 'height' axis.

Implementation details

Solving a sparse linear least-squares problem is the main computational core of the surface reconstruction.

Firstly we can compute a sparse triangular factorization of the normal equations by using a direct solver,

Then find the minimizer by the back-substitution.

Lastly, perform only once per ROI will enhance the efficiency.

C. Present some results:

The local shape can be preserved as much as possible in considering the geometry which is essentially encoded using differential properties of the surface.

The technology introduced in this paper can tackle many cases, like below:

- Mesh editing

(1) Different handle manipulations. Top-right: Translation of the handle. Down-left and down-right: Rotation of the handle.

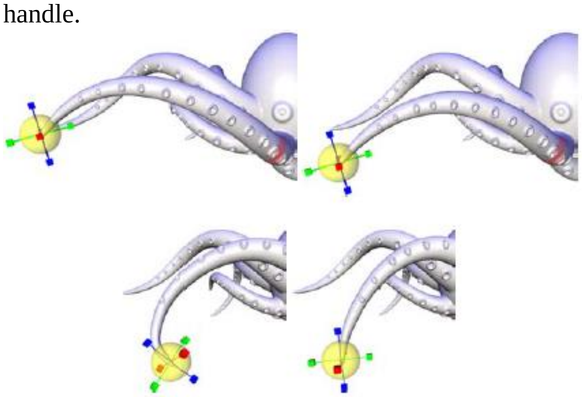

Fig.23 [4]

(2) User can select the ROI on the DRAGON to do mesh editing.

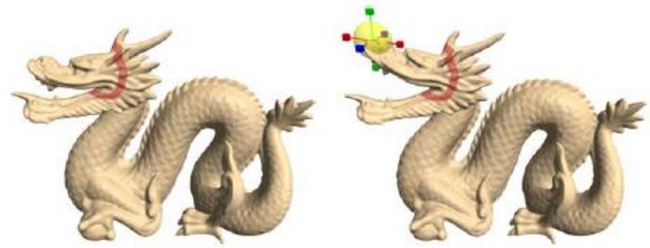

Fig.24 [4]

- Mixed detail using Laplacian coordinates.

Fig.25 [4]

- Detailed transfer, mapping and resampling

(1) The details of the left model is transferred to the middle model, the result model is on the right.

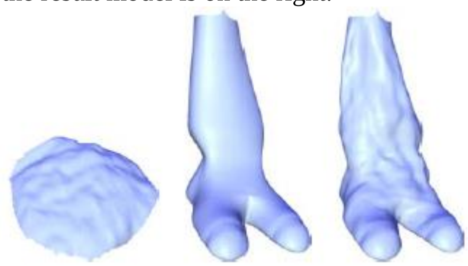

Fig.26 [4]

(2) Detailed transfer with different levels of smoothing.

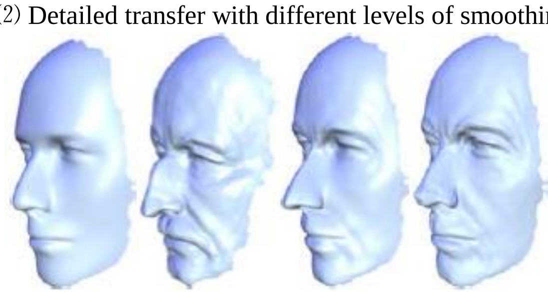

Fig.27 [4]

(3) The source surface is significantly smoothed to generate the smooth surface [middle model], and then transfer onto the right model.

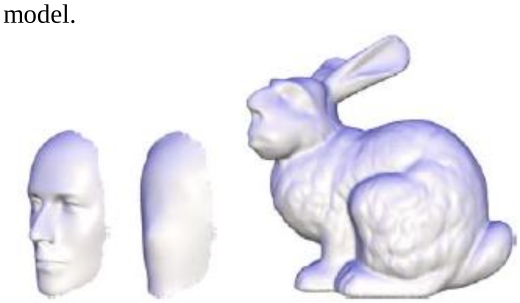

Fig.28 [4]

(4) The transferred detail vector are rotated to match the orientation of the corresponding points in the bottom two models.

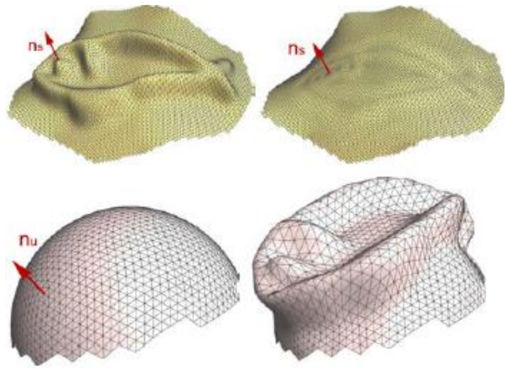

Fig.29 [4]

- Transplanting surface patches.

Transplant of FELINE’s wings onto the Bunny.

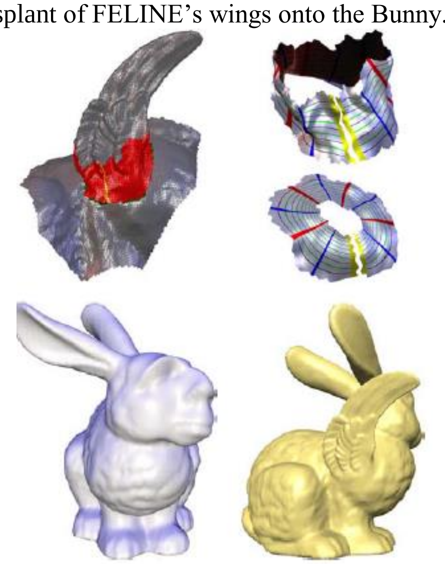

Fig.30 [4]

Advantages:

The followings are the comparison of the multi-resolution approach and Laplacian approach:

- Users can choose the editing region and model arbitrary boundary constraints freely.
- The solution of the linear system will be required in computing the absolute coordinates.
- In multi-resolution representations, the non-local bases limit the choice of the editing region and the boundary constraints.
- The absolute coordinates are computed much faster and simpler by the hierarchy.

Disadvantage:

The Laplacian is not invariant to scaling and rotations, and there are several ways to fix this.

- It can be defined and prescribed according to modeling operation.
- It can be deduced from the membrane solution: $LV' = 0$.
- It can be implicitly defined by the solution; under the condition that the rotation part is linearized.

Future’s work:

The author suggested we can investigate on these aspects:

Modeling geometry should include other surface properties, like textures.

The well-known Poisson equations which are effective in editing image, so we can look into applying it to perform on the combined differential geometry/texture representation.

VI. Differential Coordinates for Interactive Mesh Editing [5]

A. Objectives:

This paper proved how a differential representation of vertex coordinates can be exploited for the editing of arbitrary triangle meshes.

We can use the mean of the linear differential coordinates to preserve the high-frequency detail of the surface. The differential coordinates represent the details of the vertex coordinates and they are defined by a linear transformation of the mesh vertices.

It is not rotation-invariant because it is defined in a global coordinate system, and by rotating them with an approximated local frame to compensate for this. Under a pre-computed factorization of the coefficient matrix, this linear least square system can perform efficiently enough to guarantee interactive response time.

It proposes to use the differential coordinates as an alternative representation for the vertex coordinates. The authors develop the affine-invariant coordinates in mesh editing by proposing a procedure to explicitly manipulate the differential coordinates, which are only translate-invariant, no rotation-invariant.

B. Technology it involved:

It is going to resolve the issue: the differential coordinates are not rotation-invariant.

By rotating the differential coordinates according the rotation of an approximated local frame, it will rectify the natural orientation of the details.

Differential Coordinates

1. Let S is a scheme approximating vertices $\boldsymbol{p}_j \in \boldsymbol{V}$ by linear combination of some other vertices:

$$\boldsymbol{p}_j \approx S\left(\boldsymbol{p}_j\right) = \sum_{i \in supp(j), i \neq j} \alpha_{ji} \boldsymbol{p}_i$$

2. The linear transformation which is defined as linear differential mesh operator is created by above S.

$$D(\boldsymbol{p}_j) = \boldsymbol{p}_j - S(\boldsymbol{p}_j)$$

3. The differential representation of the mesh is created by above S.

$$D(V) = V - S(V)$$

4. The mesh Laplacian operator:

$$D(\boldsymbol{p}_j) = L\left(\boldsymbol{p}_j\right) = \boldsymbol{p}_j - \frac{1}{d_j} \sum_{i:(j,i)\in E} \boldsymbol{p}_i$$

5. d_j is the valency of vertex j.

6. Approximation scheme S is denoted as below:

$$S(\boldsymbol{p}_j) = \frac{1}{d_j} \sum_{i:(j,i)\in E} \boldsymbol{p}_i$$

7. D measures the deviation of a vertex from the centroid of its neighbors and thus captures local detail properties of the surface, which will need to preserve during the implementation.

8. D is linear and will be represented by ($\mathbf{n} \times \mathbf{n}$) matrix M, where n=$|V|$:

$$M_{ij} \begin{cases} 1 & i = j \\ -\alpha_{ij} & j \in supp(i) \\ 0 & otherwise \end{cases}$$

9. D ($\boldsymbol{p}_j$) is the differential coordinates of vertex j.

$$(\delta^{(x)}, \delta^{(y)}, \delta^{(z)}) = M(\mathbf{p}^{(x)}, \mathbf{p}^{(y)}, \mathbf{p}^{(z)})$$

$\boldsymbol{\delta}^{(x)}$ is the n-vector of x components of $D(\boldsymbol{p}_j)$.

10. If $|sup(j)|$ is less, M is a sparse matrix.

11. The absolute coordinates of the mesh geometry can be got by solving the system $MX = \delta^{(x)}$.

12. It uses spatial constraints to the system to get a unique lest-square reconstruction and control the surface's shape.

13. It adds the equation $w_i x_i = w_i c_i$ to system to put a constraint on the position of vertex i.

14. The resulting system $\boldsymbol{AX} = \boldsymbol{b}$ can be solved in the least-squares sense.

- It is constructed from the extension of the constrained vertex positions equations and the basic differential operator matrix.
- For the $\boldsymbol{b}$, it contains the rotated differential coordinates, the stationary anchors and the constrained locations of the handle.
- It has much freedom to choose an appropriate differential operator D, like the first and second orders of the Laplacian.
- The higher-order operation will have the larger support and result in a less sparse system.

Preserve the orientation of the details

Since the differential coordinates are defined in a global coordinate system, so they are not rotation invariant. Which result in the orientation of the reconstructed details will lose, the details will not rotate with deformed surfaces.

The resolving way is in keeping representing the vectors in the global coordinate system, rotating the vectors will represent the differential coordinates. The rotation is taken to be the local estimation of the transformation in the low frequency surface.

There are four main steps to reconstruct the shape from the rotated differential coordinate.

The differential coordinate has the following important properties:

- R is a global rotation; D is the transformation from absolute to differential coordinate.

$$R \cdot D(\boldsymbol{p}_i) = D(R \cdot \boldsymbol{p}_i)$$

The editing transformation T introduces different local rotations on the surface;

The original frame is defined as:

$$D(\mathbf{p}_j) = \alpha \mathbf{n}_j + \beta \mathbf{u}_{ij} + \gamma(\mathbf{n}_j \times \mathbf{u}_{ji})$$

1. Apply a rough deformation T to the mesh.

$$D'(\mathbf{p}'_j) = \alpha \mathbf{n}'_j + \beta \mathbf{u}'_{ji} + \gamma(\mathbf{n}'_j \times \mathbf{u}'_{ji})$$

2. Approximate local rotations R_j.

Local rotation at vertex j is approximated by the rotation of an orthogonal frame:

$$\{\mathbf{n}'_j, \mathbf{u}'_{ji}, \mathbf{n}'_j \times \mathbf{u}'_{ji}\}$$

And $\mathbf{u}_{ji}$ is:

$$\mathbf{u}_{ji} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

Where:

$$\mathbf{v} = (\mathbf{p}_i - \mathbf{p}_j) - \langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{n}_j \rangle \mathbf{n}_j$$

The rotation of the normal component is defined as:

$$\mathbf{n}_j \leftrightarrow \mathbf{n}'_j$$

It is best to choose the one whose direction is the closet to being orthogonal to $\boldsymbol{n_j}$

If $\|\boldsymbol{R_1} - \boldsymbol{R_2}\| \approx 0$, assume $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are similar.

Try to make the value $\|\boldsymbol{R}i - \boldsymbol{R}_{i+1}\|$ to be small for vertices i in the neighborhood of j.

3. Rotate each differential coordinate $D(\boldsymbol{p}_j)$ by $\boldsymbol{R}_j$.

$$R \cdot D(\boldsymbol{p}_i) = D(R \cdot \boldsymbol{p}_i)$$

The normal directions of nearby points over a low-frequency surface do not deviate rapidly.

4. Solve the system of $R_j(D(\boldsymbol{p}_j))$ to reconstruct the edited surface.

Editing Using Differential Coordinates

The editing process steps from the user side:

- Define ROI (Region of interest) for editing.
- Select handle vertex.
- Optionally define stationary anchors vertices, which are transition vertices between untouched part of the mesh and the ROI.
- Move the handle vertex

The steps from the algorithm point of view:

Once ROI, vertices for the stationary anchors and handle vertex are defined. Vertices are divided into two groups: the modified vertices and the untouched vertices of the mesh.

- The handle vertex is updated at once.
- The unmodified vertices will be forced to follow the user inputs to represent the overall shape.
- Use the least-squares solution method to generate a soft blend between the ROI and the fix part of the mesh.
- Choose several layers of anchors to improve the smoothness; these anchors have the weights which are proportional to the geodesic distance from their handle.
- Then the edited surface will be reconstructed from the locally rotated differential coordinates, but it needs a priori estimate on the normal of the editing result.
- Reconstruct the surface by solving the linear least-square system.

In summary, the computational kernel of the algorithm is the sparse linear solver for the least-squares problems; minimize $\|A\mathbf{x}\text{-}\mathbf{b}\|$ over the modified region of the surface.

Normal Estimation

The normal can be calculated by the smooth weighting scheme, in this way, the weights decrease with the distance from the estimated vertex:

$$\mathbf{n} = \sum_{i \in W_j} w_{ij} \mathbf{n}_i$$

$$w_{ij} = p(\mathrm{dist}(\mathbf{p}_j, \mathbf{p}_i))$$

In order to overcome the shortcoming of the difficulties and expensive computation, it is refined to use the De Casteljau algorithms, by computing the length of the weighted shortest path between $\boldsymbol{p}_j$ and $\boldsymbol{p}_i$.

The above mentioned normal schema gives a smoother approximation of normals.

C. *Present some results:*

Representing the geometric information of a triangle mesh in differential form enables the detail preserving interactive shape modeling.

By solving a sparse system, it can reconstruct the absolute vertex positions from their relative coordinates. Factorization is applied only once per ROI. The local rotations are applied to the differential coordinates to preserve the orientation of the details; otherwise the result will have distortion. The method can work for fairly complex models.

It is recommended to choose the differential mesh operator *D* as uniform discretization of the Laplacian technology. The order of the Laplacian affects the local support of the operator, then the sparseness of the system.

The software consists of two components:

- Triangle mesh
- Sparse linear solver with a matrix package.

This method does not need multi-resolution analysis and synthesis to provide different scale in interactive edits.

It has following applications.

- Editing the model with different 'locality' effect.

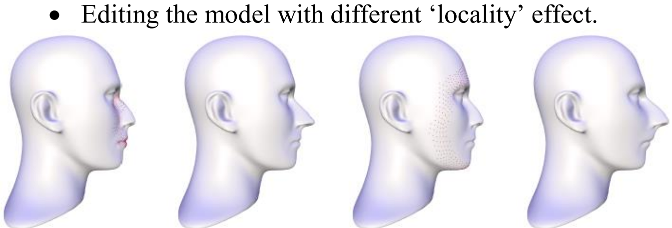

Fig.31 [5]

- Apply local rotation to the differential coordinates.

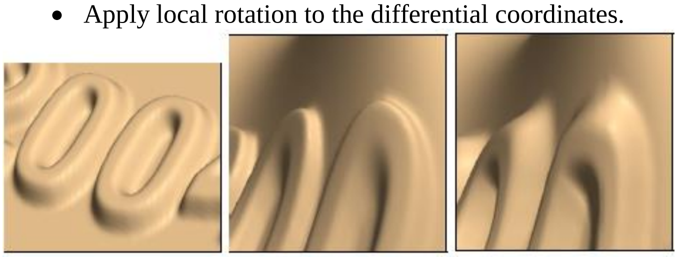

Fig.32 [5]

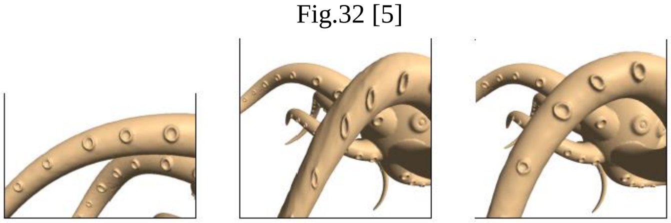

Fig.33 [5]

- Constraining curves and handle regions by appropriately grouping handle vertices.

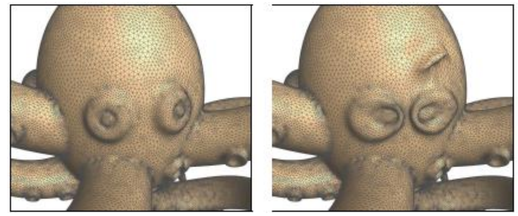

Fig.34 [5]

- Define different ROIs will have different deformation.

(1) Small ROIs, result in the local change of the arm.

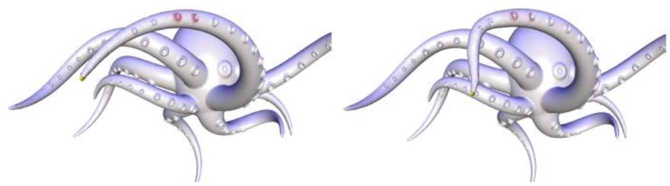

Fig.35 [5]

(2) Large ROIs, allows for a more global deformation.

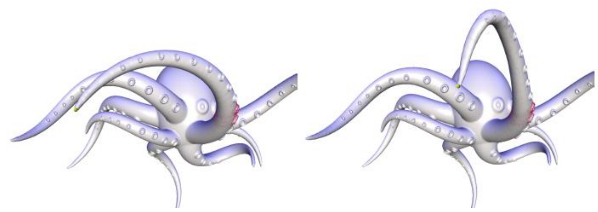

Fig.36 [5]

Author also did some experiments:

- Obtained the running times of solving the linear least-squares system for the different editing regions, as below:

Model	ROI	Factor	Solve
Mannequin (Figure 1(a))	1,201	0.031	0.002
Mannequin (Figure 1(c))	3,395	0.051	0.003
Octopus (Figure 7, top)	4,685	0.092	0.005
Octopus (Figure 7, bottom)	12,774	0.568	0.020
Octopus (Figure 6, bottom)	16,792	0.804	0.030
Height field (Figure 6, top)	32,280	1.863	0.069

Fig.37 [5]

The data in the above table indicates the direct solver performs quite fast even on large editing regions.

- The weighting scheme leads to a smoother approximation of the normals.

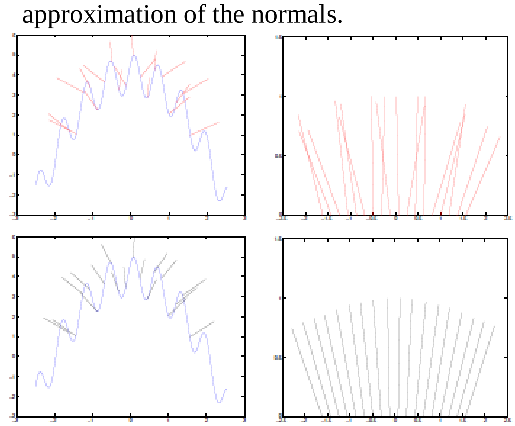

Fig.38 [5]

- If we use the second-order Laplacian (without applying explicit rotations to the differential coordinates), it will exhibit smoother transition between the stationary vertices and the ROI.

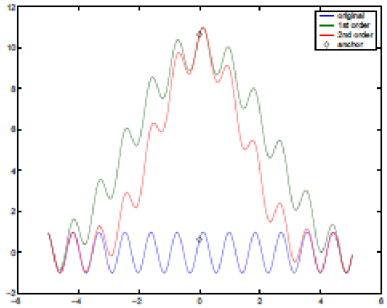

Fig.39 [5]

Advantages:

- It is efficient, inactively and intuitive shape modeling includes the local control and detail preservation.
- It is an effective and efficient method for fairly complex input meshes.
- The differential coordinates are defined respect to a common global coordinate system, which can represent the geometric details.
- By solving a linear least squares system, the presentation allows a direct detail-preserving reconstruction of the modified mesh.
- It allows editing arbitrary triangle meshes.
- It enables flexible, intuitive and powerful for interactive feature-preserved shape modeling method.
- It is conceptually simple and straightforward to implement.
- It avoids explicit multi-resolution representation of the shape to allow editing in different scale.

Disadvantages:

- The differential coordinate are not rotation invariant because they are defined in a global coordinate frame.
- It is limited to only express the differential coordinates in linear terms.

Implementation Issues:

1.) In solving the sparse linear least-square solver system, the calculation speed is high, but it is base on pre-factorization of the coefficient matrix.

The author uses a direct solver for the normal equations:

$$A^T A\mathbf{x} = A^T \mathbf{b}$$

The triangular factorization for the matrix $A^T A$ is computed by an upper triangular matrix *R* with the formula: $A^T A = R^T R$ which is most time-costing.

By only calculating once per defined edited region, it can solve the issue.

2.) It uses De Casteljau's algorithm to compute the discrete shortest path to resolve the issue: it is difficult and time-consuming in the computation for smooth normals estimation.

Future's work:

We can look into the tradeoff between additional computational costs and the benefit for editing.

We can investigate the more potential properties of the differential coordinates in digital geometry processing. Like to employ differential operators used in [6] to mesh editing process.

We also can investigate to replace an arbitrary subset of handle vertices with general FFD.

VII. Comparison of All Above Methods

A. *Free Form Deformations*

Free Form Deformations (FFD), is a powerful modeling technique that enables the object's deformation by modifying the space around them.

It requires volumetric cells specified in the interior of the control mesh, it is dependent on how the control mesh is decomposed into volumetric cells, which are always defined with multi-dimensional spline, by manipulating control points of the spline, an interface will reflect the underlying mathematics of the modeling method to implement the shape deformation.

FFD has following advantages:

- Free-form deformation enables high-quality space modeling by directly manipulating the 3D space.
- It is independent on the complexity of the object which is manipulated.
- It can be applied to any parametric or polygonal model, and is therefore will not be restricted to any class of objects.

FFD also has following shortcomings:

- They typically fail to reproduce correct deformation results if only a small number of constraints are used.
- The placement and control of the lattice used to define the deformation are non-trivial.
- It may be contra-intuitive, because the control points' move according to the user's input, the result of a subsequent edit will depend on a new control lattice.
- It forces the user to firstly define control points around the region of space to deform and then manipulate these control points. It will be unnatural and tiring in case there are many of them.
- Tri-variate volumes are based on multiplication of three basis functions, which will be too costly for interactive applications.
- It will restrict the user to creating control meshes with quadrilateral faces.
- The deformations are defined by 3D splines parametric functions whose values are determined by the location of control points. It is sometimes difficult.
- The control point movement only indicated the type of deformation the object will be changed to, the interface will confuse users.
- Some shapes are not intuitive to form.
- Complex deformation operations often require a large number of control points, which will cause the screen clutter.

- Sometimes it is difficult to select or operate the control points efficiently because they tend to get buried during the deformation.
- 3D volume of control points for FFDs (in contrast to the 2D mesh of control points for spline surfaces) exacerbates the difficulties of deciding how an aggregate move should be performed.
- The control lattice used to manipulate the underlying space won't directly relate to the object being deformed. Then when a control point which are close to the object's surface, it may be far from the object surface after the deformation. Thus, these methods may surprise a user who does not understand the distinction between the object and the space in which it lies.

B. Mean Value Coordinates for Closed Triangular Meshes

Advantages:

- It is well defined on both the interior and exterior of the mesh.

Futures' work:

- We can explore the feature: perform a job of extrapolating value outside of the mesh.
- We can investigate the relationship between Wachspress coordinates and the mean value coordinate.

Limitation:

- The technology only considers the meshes those have triangular faces, it does not start working on the mean value coordinates for piecewise linear mesh with arbitrary closed polygons as faces yet.

C. Mesh Editing based on Discrete Laplace and Poisson Models

Advantages:

The comparison of the multi-resolution approaches and Laplacian approach are as below:

- The user can choose the editing region and model arbitrary boundary constraints freely.
- The solution of the linear system is required in computing the absolute coordinates.
- In multi-resolution representations, the non-local bases limit the choice of the editing region and boundary constraints.
- The absolute coordinates are computed much faster and simpler by the hierarchy.

Disadvantage:

The Laplacian is not invariant to scaling and rotations. There are several ways to fix this:

- It can be defined and prescribed according to the modeling operation.
- It can be deduced from the membrane solution: $LV' = 0$.
- It can be implicitly defined by the solution; under the condition of the rotation part is linearized.

D. Differential Coordinates for Interactive Mesh Editing

It can be used in the arbitrary 3D mesh structures for progressive and compressed geometry coding.

It can derive a geometry compression algorithm which has benefits from much quantization.

It will discretize the Poisson equation with Dirichlet boundary conditions.

E. Mesh Editing based on Discrete Laplace and Poisson Models

This technology combined previous two papers together to summarize because they used almost the same algorithm: differential coordinates, and the Laplacian coordinates which is the simplest form of differential coordinates.

Advantages:

The Laplacian coordinates can be effective for morphing.

Disadvantages:

In order to preserve the orientation of the local details, it needs to rotate the local frames that define the Laplacians, there are two ways:

- The local rotations of the frames will be estimated on the underlying smooth surface,
- Applying the rotation of the editing handle to all the vertices of the region of interest.

It will be more suitable to constrain under a global deformation of the mesh to describe a local mesh with the differential coordinates.

VIII. Conclusions

Mesh editing and interactive shape deformation are very active fields of research in computer graphics. From the in-depth survey of the above mentioned five papers, we know that every technology/algorithm involves has its advantages and disadvantages, and we can find answers to some questions listed in Section I. There are also other many technologies that are invented and published in the papers [7]-[15], we can get answers for all the rest of the questions once we read through them, We will have a better understanding on what have been developed in this field.

Every detailed method listed in sections II-VI is robust or efficient in tackling some specific cases. It is not easy to propose a general way that can tackle all cases.

In the future, we may focus on the methodology and algorithms to generate a more general model to deal with any mesh editing or shape editing requests. We can implement this gradually, e.g., at the first stage, we can divide all cases into several main standard categories, and then invent several main solutions/formulations to tackle requests for those main categories respectively. If this can be accomplished, we will continue to work on the more general model to tackle all cases in the next stage.

References

[1] Thomas W., Scott R. "*Free-Form Deformation of Solid Geometric Models*" ACM0-89791-196-2/86/008/0151

[2] Yizhou Yu, "*Mesh Editing with Poisson-Based Gradient Field Manipulation*" 644-651 SIGGRAPH 2004. SIGGRAPH '04 ACM SIGGRAPH 2004 Papers Pages 644-651, Volume 23, Number 3, August 2004

[3] Tao Ju, Scott Schaefer, Joe Warren, "*Mean Value Coordinates for Closed Triangular Meshes*" SIGGRAPH '05 ACM SIGGRAPH 2005 Papers, Pages 561-566 , ACM Transactions on Graphics (TOG) - Proceedings of

ACM SIGGRAPH 2005 TOG, Volume 24 Issue 3, July 2005, Pages 561-566

[4] Marc Alexa, Andrew Nealen, "*Mesh Editing based on Discrete Laplace and Poisson Models*", Advances in Computer Graphics and Computer Vision, Communications in Computer and Information Science Volume 4, 2007, pp 3-28

[5] Yaron Lipman , Olga Sorkine , Daniel Cohen-or , David Levin , Christian Rössl , Hans-peter Seidel, "*Differential Coordinates for Interactive Mesh Editing*", Proceeding of SMI '04 Proceedings table of contents ISBN:0-7695-2075-8 doi>10.1109/SMI.2004.30, P. 181 - 190, IEEE

[6] P. P´erez, M. Gangnet, and A. Blake. "*Poisson image editing"*. In Proceedings of ACM SIGGRAPH 2003, pages 313–318, 2003.

[7] Seungyong Lee "*Interactive Multi-resolution Editing of Arbitrary Meshes*", EUROGRAPHICS '99 / P. Brunet and R. Scopigno, Volume 18 (1999), Number 3

[8] Denis. Z etl, "*Interactive Multi-resolution Mesh Editing*",SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques, Pages 259-268

[9] Ron MacCracken, Kenneth I. Joy ,"*Free-Form Deformations With Lattices of Arbitrary Topology*" ,SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, Pages 181-188

[10] Mark Meyery, Haeyoung Lee, "*Generalized Barycentric Coordinates on Irregular Polygons*" ,Journal of Graphics Tools archive Volume 7 Issue 1, November 2002, Pages 13-22

[11] Mario Botsch, Leif Kobbelt , "*Multi-resolution Surface Representation Based on Displacement Volumes*" ,Computer Graphics Forum Volume 22, Issue 3, pages 483–491, September 2003

[12] Ken Museth, David E. Breen, "*Level Set Surface Editing Operators*" ,SIGGRAPH '02 Proceedings of the 29th annual conference on Computer graphics and interactive techniques, Pages 330-338

[13] Michael S. Floater, G etl , "*Mean value coordinates in 3D*" ,Computer Aided Geometric Design,Volume 22, Issue 7, October 2005, Pages 623–631

[14] Michael S. Floater ,"*Parametrization and smooth approximation of surface triangulations*" ,Computer Aided Geometric Design archive, Volume 14 Issue 3, April 1997,Pages 231-250

[15] Karan Singh, Eugene Fiume, Aliasjwavefront ,"*Wires: A Geometric Deformation Technique*" ,SIGGRAPH '98 Proceedings of the 25th annual conference on Computer graphics and interactive techniques, Pages 405-414