Differential Coordinates for local mesh morphing and deformation

Marc Alexa(TVC 2003)
Presented by:Yannis Atsonios@CS.JHU

Outline

- Introduction
- Local mesh morphing framework
- Mesh deformation approach
- Differential representation of a mesh
- Application to mesh morphing
- Application to free-form modelling
- Conclusion

Introduction

- Meshes are mainstream models for representation of models in graphics.
- We have to ways to change the mesh geometry
- 1)Mesh morphing=>modify shape locally
- 2)Free form deformation
- Lets see more on these techniques…

Introduction

- Mesh morphing is transformation from the one mesh to the another.
- Mesh morphing is a 3 step process
- The first two steps result in one mesh connectivity with two geometries attached to the vertices.
- With this description the user can change parts of the geometry from source to target.
- Albeit we can have problem,because corresponding features might not have the same position in space=>interpolation of (absolute Euclidean)coordinates can insert undesirable effects.

- More on Alexa(2002),Recent Advances in mesh morphing(good tutorial)

Introduction

- Free form deformation gives the freedom to users to transform some parts of a shape.
- Control of deformation by some scattered vertices,in parallel we want the shape to be as close to the original given the additional constraints.
- Ideally we want coordinates to capture the local shape and not the global shape.Deformations can be a global operation.If we didn't have that bottleneck then the fitting problem posed by the (user's) constraints would be easier.

Introduction

- Example of problematic Instance

QuickTime™ and a
decompressor
are needed to see this picture.

Local mesh morphing framework

- A mesh M is described by K connectivity of its vertices,edges and faces and geometric positions V of the vertices.
- We have two meshes M_o and M_1 and by the classical mesh morphing we produce a family M(t),t in[0,1] with the given properties:φ(M(1))=φ (M_1) and φ(M(0))=φ(M_0) .
- The idea is to generate a mesh topology that can be deformed to shapes of the source and target,M(t)={V(t),K}.
- We have three steps for the computation of this family.

Local mesh morphing framework

- Correspondence between the meshes,especially computation of maps ψ_0 and ψ_1 so $\psi_0(V_0)$ in $\varphi(M_1)$ and $\psi_1(V_1)$ in $\varphi(M_0)$=>barycentric coordinate for each vertex with respect to a simplex in the other mesh.This step can be formulated as 2D parameterization of the meshes.
- Compute a new consistent mesh topology K with two geometric positions V(0) ,V(1) for each vertex so the shapes of the original meshes are reproduced.In the case that edges from different meshes cross we insert new vertices.
- We create paths V(t),t in (0,1) for the vertices=>we compute vertex positions from the original V(0),V(1) where t varies.

Local mesh morphing(1st step)

- Finding a 2D parameterization of the vertex-edge graphs of the input meshes=>correspondence between surfaces.
- Isomorphic dissection of meshes into pathes homeomorphic to a disk(because can be parameterized independently in the plane).
- Wealth of algorithms that can carry out this step(fundamentally the same..)

- Feature alignment is CRUCIAL…in order to get pleasing morphing results.Features can be selected by user or can be shape features(curvature,normals,etc).We can have even point to point correspondence via warping methods.

Local mesh morphing(2nd step)

- Having correspondence information we construct a mesh that contains (combines) both meshes.We have two ways:a)we overlay meshes in the parameter domain,b)generate multiresolution representation.
- If we are lucky and parameter domain is plane we have plethora of algorithms from Computational Geometry.
- For non planar parameter domains we need special treatment.
- Hot topic=usage of multiresolution models.Idea is to remesh the shape by a given irregular coarse base mesh with regular connectivity.How?We use geometrical info by the original meshes and mesh topology is tweaked by refinement operator.

Local mesh morphing(3rd step-vertex path)

- We have a global transformation ,so we have o compute the intermediate vertex coordinates of the intermediate shapes.We compute V(t),t in(0,1),t is called transition parameter.
- Classic way by linear interpolation.If in that scheme a rigid or affine transform used prior to the interpolation =>better results.
- General solution is to interpolate an intrinsic description of boundary,have been applied to polygons=>hard in meshes.
- Other methods based on skeleton shape interpolation techniques using not only boundary information(3d is unknown).Other methods based on isomorphic complexes of the shape and other compose optimal simplex morphs to get vertex paths.

Local mesh morphing(spatially non-uniform transition states)

- Local morph control is known from image morphing as transition control,transition of each pixel can be done by warping images.
- Local control requires description of more than a single scalar t.We assign a transition parameter to each vertex…we construct a transition state T.T is a diagonal matrix
- We can get by this matrix the linear interpolation of absolute vertex coordinates (I-T)*V(0)+T*V(1) (convexity).This method tends to be problematic.If we use path generation techniques also we can have problems such as numerically instability.

Local mesh editing

- If we have more than 2 meshes?…No problem.
- We just compute the new transition state description as a linear vertex combination: QuickTime™ and a decompressor are needed to see this picture.
- The idea for that is just reduce all meshes in a common parameter domain(the idea of barycentric representation of vertices as aforementioned)

Mesh deformation approach

- Mesh deformation is another point of view of what we have already discussed.
- User specifies absolute position of mesh vertices.Differential representation of the mesh computes a global fit in a least-square fashion constrained to the specified vertices.
- Current techniques use multiresolution methodologies.They operate(modify) in coarse level large regions of the shape and on detailed level they change shape only locally.

Differential representation of mesh

- Intuition:Representation of vertex coordinates with respect to their neighbors in the mesh.

QuickTime™ and a decompressor are needed to see this picture.

Laplacian Representation

- We compute the centre of mass by the neighbors(local operation)
- The new representation is the difference of the original and the centre of mass…By this "trick" we have increased robustness,in the sense that centre of mass is always defined(and even for non convex shapes,colinear,coplanar etc)
- Forward Transformation(absolute=>relative coordinates).

A :adjacency matrix of mesh graph D:diagonal matrix where d(I,j)=1/|N(I)|.Then the transform is L=I-D*A,L is a Laplacian of the mesh!=>Is a generalized shape representation that can be applied not only to meshes but parametric or implicit functions.

Laplacian Representation(#2)

- Backward transformation(relative=>absolute coordinates) is not unique=ill posed operation.Is uniquely determined up to a translation.L (in m Euclidean space) has rank m-1.DA is stochastic matrix(eigenvalue 1) also is normal then we have multiplicity 1 for all eigenvalues.(Due to the fact that is stochastic can we engage any randomized technique…any markov chain???).L has one eigenvalue of 0.

Affine independent representation

- We represent a vertex v with respect to its neighborhood N(I),this representation is invariant affine transforms is straightforward
- This is linear system(in homogenous coordinates).
- We can solve it by SVD.This strategy provides the optimal in mean square weighting fashion and the best when no exact solution exists.
- If the neighborhood of N(I) is not bases of 3d euclidean space we loose information about the shape.
- We can use Laplacian coordinate but numerically is not a good idea.
- This means that the problem is susceptible to local degradations that lead to global effects.
- The scheme that is used here doesn't suffer from these problems but is restricted also.

Solving for absolute coordinates

- LV=W is hard!!!(needs some sophistication)
- L is singular(theoretical bottleneck),easily can be surpassed
- L has huge dimensions(practical bottleneck)
- L is large and sparse
- LV=W must be solved 3 times(for x,y,z)
- We use approximation techniques for V(euphemism for iterative matrix methods)
- For free-form modeling we use least-square fit.
- Iterative methods like Gauss-Seidel,Jacobi can easily used with mesh data structures and are stable for least square fitting.

Application to mesh morphing

- Idea:Morph by linearly interpolating Laplacian coordinates rather than absolute coordinates..Laplacian are linear in absolute coordinates.Morphing is them same in absolute and Laplacian coordinates.
- Even if the translations are different for subsets of vertices,interpolation of Laplacians yields reasonable results.
- We have a mesh topology K and geometries Vi.We apply a forward transform L which allows computing Laplacian coordinates Wi.
- Ti is a set of transition matrices
- V'=inv(L)(T1W1+T2W2+…..+TnWn)
- To=I-T,T1=T,we have matrices that change over time.
- We can have some several key frames of matrices.
- Interpolation is done in matrix space ,in absolute values should be avoided.

GUI for defining states

- GUI helps defining a region of interest(ROI)
- User specifies a transition state T'' for ROI.Then T'=(1-d)*T''+d*T,where d is a distance which is 0 inside the inner boundary and 1 outside the outside boundary.Dijkstra algorithm is used to compute the distances.
- This technique is used to produce a morph sequence.
- Interesting fact:The eigenvectors of the Laplacian L form a basis which is equivalent to Fourier basis of a mesh….
- So we have a local spectral filter and we can define several band limited versions of shapes.

Spatially dependent transition states

- We can define transition state without using ROI scheme.Transition state is function of absolute coordinates.
- Transition state from absolute coordinates would involve a function from points of 3d space to weights how to combine Laplacian coordinates in that point.Using Laplacian framework global positioning is lost.We use a heuristic to overcome that difficulty(otherwise we have to solve a non-linear system)
- Simple heuristic:Absolute coordinates of vertices for each of the source shapes define a transition state.These transition states are averaged to yield final transition state.
- This method has the advantage that smooth changes in spatial distribution lead to smooth changes in shape.

Application to free-form modeling

- We have a mesh (K,V) this defines a forward transform L,the laplacian representation W.User inserts a set of constraints of vertices
- We try to fit V' to minimize the squares of the actual Laplacian coordinates
- Min (LV'-W)^2. It is computed by setting the gradient over the free vertices to 0.We have a linear system in the free vertices.

QuickTime™ and a
decompressor
are needed to see this picture.

Conclusion

- Differential mesh coordinates introduced,this method helps local mesh editing task.
- Laplacian are advantageous in terms that are independent of transformations of the shape and are rigorously defined.
- Laplacian are sensitive to scaling and rotation of the shape.This can be very problematic and is numerically hard to manage..
- Future direction,to devise an affine invariant representation scheme.

Queries????

Thank you for your time..