Geometry Processing

6 Deformation

Ludwig-Maximilians-Universität München
Announcements

● Guest Talk: Industrial Modeling (scheduled at 22.02.2021, will send email for further information)
  ○ From WAY Engineering Group
  ○ Good opportunities to learn about real-world industrial level meshes (large and messy geometric data)
  ○ Connection and maybe intern

● No more coding projects, just focus on polish your proposed individual project :)
  ○ But still open for submission, but no template provided

● Register to the "exam" (project presentation, just for reporting the grades to examination office)
  ○ Bachelor: https://uni2work.ifi.lmu.de/course/W20/IfI/GP/exams/GP-Exam/show
  ○ Master: https://uni2work.ifi.lmu.de/course/W20/IfI/PGP/exams/PGP-Exam/show
Session 6: Deformation

● Motivation
● Surface Deformation
● Space Deformation
● Skinning
● Summary
Large Existing Applications

Character animations

Image editing

"Memoji"

...
Problem Settings

The deformation of a given surface $S$ into the desired surface $S'$

- A displacement function $d(p)$ on each vertex $p \in S$
- Desired surface is determined by the displacement $S' = \{ p + d(p) | p \in S \}$

User inputs (constraints):

- Handle region $\mathcal{H}$ such that $d(p_i) = \bar{d}_i, \forall p_i \in \mathcal{H}$
- Fixed region $\mathcal{F}$ such that $d(p_i) = 0, \forall p_i \in \mathcal{F}$

Optimization goal: **Determine the displacement for remaining region**

$$d(p_i), \forall p_i \in \mathcal{R} = S \setminus (\mathcal{H} \cup \mathcal{F})$$
Approaches

Surface deformation

- Shape is an empty shell: Curve for 2D and surface for 3D deformation
- Deformation only defined on the shape
- Deformation coupled with shape representation

Space deformation

- Shape is volumetric: Planar domain in 2D and Polyhedral domain in 3D
- Deformation defined in neighborhood of shape
- Can be applied to any shape representation
Session 6: Deformation

- Motivation
- Surface Deformation
- Space Deformation
- Skinning
- Summary
Transformation Propagation

Basic Idea: Smooth blending between the transformed handle and the fixed region

Controlled by a scalar field:

- Smoothly blends between 0 (fixed region) and 1 (the handle)

Distance function:

\[
d_{\mathcal{F}}(p) \quad \text{distance from vertex } p \text{ to the fixed region}
\]

\[
d_{\mathcal{H}}(p) \quad \text{distance from vertex } p \text{ to the handle region}
\]

Scalar field:

\[
s(p) = \frac{d_{\mathcal{F}}(p)}{d_{\mathcal{F}}(p) + d_{\mathcal{H}}(p)}
\]

[Botsch et al. 2008]
Transformation Propagation (2) Smooth Blending

Recall Laplacian smooth, consider the scalar field as the harmonic field on the surface, we have:

\[ \Delta s(p_i) = 0, p_i \in \mathcal{R} \]
\[ s(p_i) = 1, p_i \in \mathcal{H} \]
\[ s(p_i) = 0, p_i \in \mathcal{F} \]

Replace the Laplacian via Laplace-Beltrami operator, that turns into a linear system.

The resulting scalar field is used to damp the transformation of the handle for each vertex in the remaining region:

\[ p'_i = s(p_i) T_{\mathcal{H}}(p_i) + (1 - s(p_i)) p_i \]

Handle Transformation

*Easy to implement but typically not result in geometrically intuitive solution.*
Multi-Scale Deformation

Basic concept: decompose the object into multiple frequency and deform low frequencies (global shape) while preserving the high frequency details

- Low frequencies correspond to the smooth global shape
- High frequencies correspond to the fine-scale details

Not a specific approach but a general framework of doing deformation tasks

displacement with regard to global coordinate system and local tangent plane
**Laplacian Editing** [Sorkine et al. 2004]

Manipulate per-vertex Laplacian

1. Compute initial Laplacian (scalar)
2. Manipulate Laplacian coordinates (local transformation)
3. Find new coordinates that match the target Laplacian coordinates

Similar approach uses gradient in Poisson gradient mesh editing [Yu et al. 2004]
As-Rigid-As-Possible Deformation (ARAP) [Sorkine et al. 2007]

Basic idea: deformed object should only apply rotation and translation (rigid), no scaling and shearing.

Define and minimize energy function:

\[
E = \sum_{i=1}^{N} A_i \sum_{j} w_{ij} \left\| (p'_i - p'_j) - R_i (p_i - p_j) \right\|
\]

all verts  one ring  rotation matrix

Use alternating minimization technique (EM algorithm in machine learning) to minimize the energy.
Interactive Deformation

Deformation often has higher demand on interactivity, so that either designer can iterating their idea quickly or renderer can manipulate geometry in real-time.

Implementation thinking:

- "Interaction cost": Designing minimum user inputs (select handle and fixed region, drag handle)
- "Intuitive deformation": Transformation works as "expected", global deformation that preserves local details
Session 6: Deformation

- Motivation
- Surface Deformation

**Space Deformation**

- Skinning
- Summary
Space-based Deformation

Space-based deformations (meshless mapping) deforms the ambient space and thus implicitly deform the embedded objects.

Two classic approaches

- Lattice-based freeform deformation
- Cage-based freeform deformation
Lattice-based Freeform Deformation

Space deformation represented by a trivariate tensor-product spline function

\[ d(u) = \sum_{l=1}^{n} \delta c_l N_l(u) \]

Each original vertex \( p_i \in S \) has a corresponding parameter value \( u_i = (u_i, v_i, w_i) \) such that \( p_i = \sum_l c_l N_l(u_i) \)

The deformation can be considered as the transformation of vertices:

\[ p_i' = p_i + d(u_i) \]

The remaining problem is to solve the displacement function.
Cage-based Freeform Deformation

A generalization of the lattice-based freeform deformation

Control cages are typically coarse, arbitrary triangle mesh enclosing the object to be modified

The vertices \( p_i \) of original mesh \( S \) can be represented as linear combinations of the cage's control vertices \( c_l \) by

\[
p_i = \sum_{l=1}^{n} c_l \varphi_l(p_i)
\]

where \( \varphi_l(p_i) \) are generalized barycentric coordinates, such as mean value coordinates.
Session 6: Deformation

- Motivation
- Surface Deformation
- Space Deformation
- Skinning
- Summary
Linear Blend Skinning

Direct skeletal shape deformation

Input data assumption

- **Rest pose shape**: original undeformed polygon mesh
- **Bone transformation**: a list of transformation matrices
- **Skinning weights**: amount of influence of a bone on a vertex

\[ \mathbf{v}_i' = \left( \sum_{j=1}^{m} w_{ij} \mathbf{T}_j \right) \mathbf{v}_i \]

Similar terminologies: skeleton-\textit{subspace} deformation, (single-weight-)enveloping, matrix-palette skinning
LBS and More

Multilinear skinning

Nonlinear skinning

...
Delta Mush (DM) [Mancewicz et al. 2014]

- Rigid binding using global large-scale solver
- Mush = Smoothing
  - Laplacian smooth
  - Shrink geometry
  - Lose surface detail
- Delta = Rest Pose - Rest Pose Mush
  - Encode surface's details as displacements
  - Stored at local frame defined by the mush
  - Pre-computed once
- Delta Mush = Delta + Deformed Mush
Direct Delta Mush [Le et al. 2019]

Real-time acceleration via

- Direct computation (runtime)
- Pre-computation (once)

Delta Mush like quality for easy authoring without cleft and bulging
Session 6: Deformation

- Motivation
- Surface Deformation
- Space Deformation
- Skinning

- Summary
Summary

- Deformation is a surface to surface mapping process
- Surface-based approach applies directly on surface and offer precise control on surface
- Space-based approach interpolates space and mostly meshless
- Skinning combines surface-based and space-based approaches that enable more flexibility of authoring in character modeling process
Homework 6: Delta Mush (optional)

Implement the Delta Mush in halfedge representation.

1. Compute Laplacian smooth on rest pose
2. Compute delta of vertices
3. Apply bone transformation
4. Apply Laplacian smooth on transformed object
5. Recover vertex positions

Implementation details on handling user inputs:

1. Create skeleton on rest pose
2. Interactive bone transformation controller
Further Readings: Deformation (1)


Further Readings: Deformation (2)


Thanks! What are your questions?

Next session: Data-driven Approach