CSE 554
Lecture 10: Extrinsic Deformations

Fall 2016
Review

- Non-rigid deformation
  - **Intrinsic** methods: deforming the boundary points
  - An optimization problem
    - Minimize shape distortion
    - Maximize fit
  - Example: Laplacian-based deformation

![Diagram showing Before and After deformation](image)
Extrinsic Deformation

- Given source and target point pairs (handles)
- Compute deformation of any point in the plane or volume
  - Not just points on the boundary curve or surface
Extrinsic Deformation

- Applications
  - Registering contents between images and volumes
  - Interactive animation
Techniques

- Thin-plate spline deformation
- Free form deformation
- Cage-based deformation
Thin-Plate Spline

- A minimization problem
  - Minimizing distances between source and target points
  - Minimizing distortion of the space
- There is a closed-form solution
  - Solving a linear system of equations
Thin-Plate Spline

- Input
  - Source points: $p_1, \ldots, p_n$
  - Corresponding target points: $q_1, \ldots, q_n$

- Output
  - A deformation function $f[p]$ ($p$ is any point)
Thin-Plate Spline

- Minimization formulation

\[ E = E_f + \lambda E_d \]

- \( E_f \): fitting term
  - Measures how close is the deformed source to the target
- \( E_d \): distortion term
  - Measures how much the space is warped
- \( \lambda \): weight
  - Controls how much distortion is allowed
Thin-Plate Spline

- Fitting term
  - Minimizing sum of squared distances between deformed source points and target points

\[ E_f = \sum_{i=1}^{n} \| f[p_i] - q_i \|^2 \]
Thin-Plate Spline

- Distortion term
  - Penalizes non-linear deformation:

\[
E_d = \int \int \left( \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 \right)^2 \, dx \, dy
\]

- The energy is zero when the deformation is a linear transformation
  - Translation, rotation, scaling, shearing
Thin-Plate Spline

- Finding the minimizer for \( E = E_f + \lambda E_d \)
  - Uniquely exists, and has a closed form:

\[
f[p] = M \cdot p + \sum_{i=1}^{n} \phi[\|p - p_i\|] v_i
\]

where \( \phi[r] = r^2 \log[r] \)

- \( M \): a global transformation
- \( v_i \): translation vectors (one per source point)
- Both \( M \) and \( v_i \) are determined by \( p_i, q_i, \lambda \)
  - By solving a linear equation system
**Thin-Plate Spline**

- **Result**
  - At higher $\lambda$, the deformation is closer to an affine transformation

$$E = E_f + \lambda E_d$$

Credits: Sprengel et al, EMBS (1996)
Thin-Plate Spline

- Application: landmark-based image registration
  - Manual or automatic detection of landmarks and correspondences

Source  Target  Deformed source

Credits: Rohr et al, TMI (2001)
Thin-Plate Spline

- Application: landmark-based image registration
  - Manual or automatic detection of landmarks and correspondences

Credits: Rohr et al, TMI (2001)
Thin-Plate Spline

- **Pros**
  - Only requires scattered point correspondences
  - Closed-form solution makes it efficient to compute the deformation
    - Once the global transformation ($M$) and local vectors ($v_i$) are solved

- **Cons**
  - Solving $M$, $v_i$ could be time-consuming for large number of handles
    - Not for interactive deformation (e.g., user moving the handles in real-time)
Free Form Deformation

- Uses a control lattice that embeds the shape
- Deforming the lattice points (control points) warps the embedded shape

Credits: Sederberg and Parry, SIGGRAPH (1986)
Free Form Deformation

- Blending the deformation at the control points (handles)
  - Each deformed point is a weighted sum of deformed lattice points
Free Form Deformation

- **Input**
  - Source control points: $p_1, \ldots, p_n$
  - Target control points: $q_1, \ldots, q_n$

- **Output**
  - A deformation function $f[p]$ for any point $p$ within the space of the grid.

$$f[p] = \sum_{i=1}^{n} w_i[p] q_i$$

- $w_i[p]$: "influence" of $p_i$ on $p$ (regardless of $q_i$, so it can be pre-computed)
Free Form Deformation

- Desirable properties of the weights $w_i[p]$
  - Decreases with distance from $p$ to $p_i$
    - So that the influence of each control point is local
  - Smoothly varies with location of $p$
    - So that the deformation is smooth
  - $1 = \sum_{i=1}^{n} w_i[p]$
    - So that $f[p] = \sum w_i[p] q_i$ is an affine combination of $q_i$
  - $p = \sum_{i=1}^{n} w_i[p] p_i$
    - So that $f[p] = p$ if the lattice stays unchanged
Free Form Deformation

- Finding weights (2D)
  - Let the lattice points be $p_{ij}$ for $i=0,\ldots,k$ and $j=0,\ldots,l$
  - Compute $p$'s relative location in the grid $(s,t)$
    - Let $(x_{\text{min}},x_{\text{max}}), (y_{\text{min}},y_{\text{max}})$ be the range of grid
      
      $$s = \frac{p_x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$
      $$t = \frac{p_y - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}}$$
Free Form Deformation

• Finding weights (2D)
  - Let the lattice points be \( p_{ij} \) for \( i=0,\ldots,k \) and \( j=0,\ldots,l \)
  - Compute \( p \)'s relative location in the grid \((s,t)\)
  - The weight \( w_{ij} \) for lattice point \( p_{ij} \) is:
    \[
    u_{i,j}[p] = \lambda_{i,j} B_i^k[s] B_j^l[t] \\
    w_{i,j}[p] = \frac{u_{i,j}[p]}{\sum_{i=0}^{k} \sum_{j=0}^{l} u_{i,j}[p]} 
    \]
  - \( \lambda_{ij} \): importance of \( p_{ij} \)
  - \( B \): Bernstein basis function: \( B_i^k[s] = \binom{k}{i} s^i (1-s)^{k-i} \)
Free Form Deformation

- Finding weights (2D)
  - Weight distribution for one control point (max at that control point):

```
\begin{align*}
\text{w}_{1,1}[p] & \quad \text{at control point } p_{1,1}
\end{align*}
```
Free Form Deformation

- A deformation example
Free Form Deformation

- Image registration
  - Embed the source in a lattice
  - Compute new lattice positions over the target
    - Manually or automatically
  - Deform each source pixel using FFD
Free Form Deformation

- Pros
  - Fast updates when control points are moved

- Cons
  - Too many control points (a lot of them are interior to the shape)
  - Cartesian grid is too inflexible for complex shapes
Cage-based Deformation

- Use a control mesh ("cage") to embed the shape
- Deforming the cage vertices warps the embedded shape

Credits: Ju, Schaefer, and Warren, SIGGRAPH (2005)
Cage-based Deformation

- “Blending” the deformation at the cage vertices

\[ f[p] = \sum_{i=1}^{n} w_i[p] q_i \]

- \( w_i[p] \): pre-computed “influence” of \( p_i \) on \( p \)
- More challenging to compute than FFD: not regular lattice structure
Cage-based Deformation

- Finding weights (2D)
  - Problem: given a closed polygon (cage) with vertices $p_i$ and an interior point $p$, find smooth weights $w_i[p]$ such that:

1) $1 = \sum_{i=1}^{n} w_i[p]$

2) $p = \sum_{i=1}^{n} w_i[p] \cdot p_i$
Cage-based Deformation

- Finding weights (2D)
  - A simple case: the cage is a triangle
  - The weights are unique (3 eqs, 3 vars)

\[
1 = w_1 + w_2 + w_3
\]

\[
p_x = w_1 p_{1,x} + w_2 p_{2,x} + w_3 p_{3,x}
\]

\[
p_y = w_1 p_{1,y} + w_2 p_{2,y} + w_3 p_{3,y}
\]

- \( w_i \) are known as the **barycentric coordinates** of \( p \)

\[
w_1 = \frac{\text{Area}_{p, p_2, p_3}}{\text{Area}_{p_1, p_2, p_3}}
\]
Cage-based Deformation

• Finding weights (2D)
  – The harder case: the cage is an arbitrary (possibly concave) polygon
  – The weights are not unique (“generalized barycentric coordinates”)
    • A good choice: Mean Value Coordinates (MVC) [Floater, 2003]

\[
\begin{align*}
  u_i(p) &= \frac{\tan[\alpha_i/2] + \tan[\alpha_{i+1}/2]}{||p_i - p||} \\
  w_i(p) &= \frac{u_i(p)}{\sum_{i=1}^{n} u_i(p)}
\end{align*}
\]
Cage-based Deformation

- Finding weights (2D)
  - Weight distribution of one cage vertex in MVC:

\[ w_i[p_i] \]
Cage-based Deformation

- MVC be extended to 3D [Ju, 2005; Floater, 2005]
  - Other types of coordinates: Harmonic coordinates; Green coordinates; etc.
- Application: character animation
Cage-based Deformation

- Registration
  - Embed source in a cage
  - Compute new locations of cage vertices over the target
  - Deform source pixels using MVC or other generalized barycentric coordinates
Cage-based Deformation

• Pros
  – Fast update when control points are moving
  – Fewer control points and better fitting to shape than FFD

• Cons
  – Constructing the cage is mostly a manual process
    • Although there is some recent progress in automation

Sacht et al., Siggraph Asia 2015
Further Readings

- Thin-plate spline deformation
  - “Landmark-Based Elastic Registration Using Approximating Thin-Plate Splines”, by Rohr et al. (2001)

- Free form deformation

- Cage-based deformation
  - “Mean value coordinates for closed triangular meshes”, by Ju et al. (2005)
  - “Harmonic coordinates for character animation”, by Joshi et al. (2007)
  - “Green coordinates”, by Lipman et al. (2008)