Washington
University in St. Louis
School of Engineering
& Applied Science

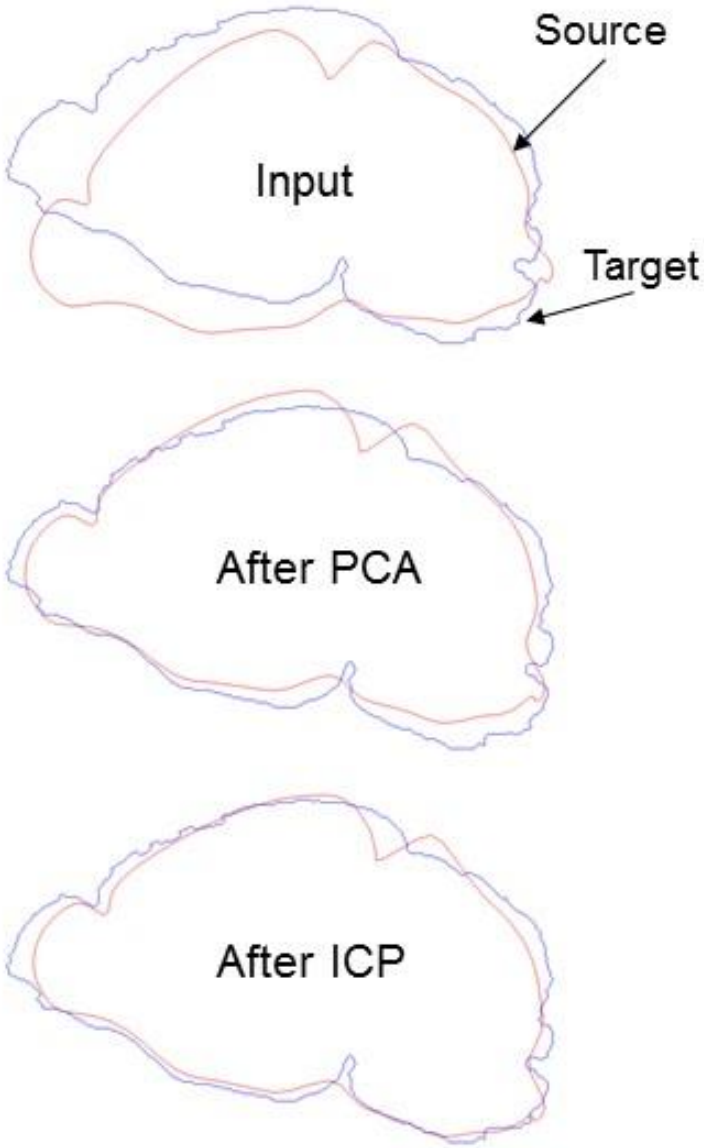
CSE 554

Lecture 9: Laplacian Deformation

Fall 2016

Review

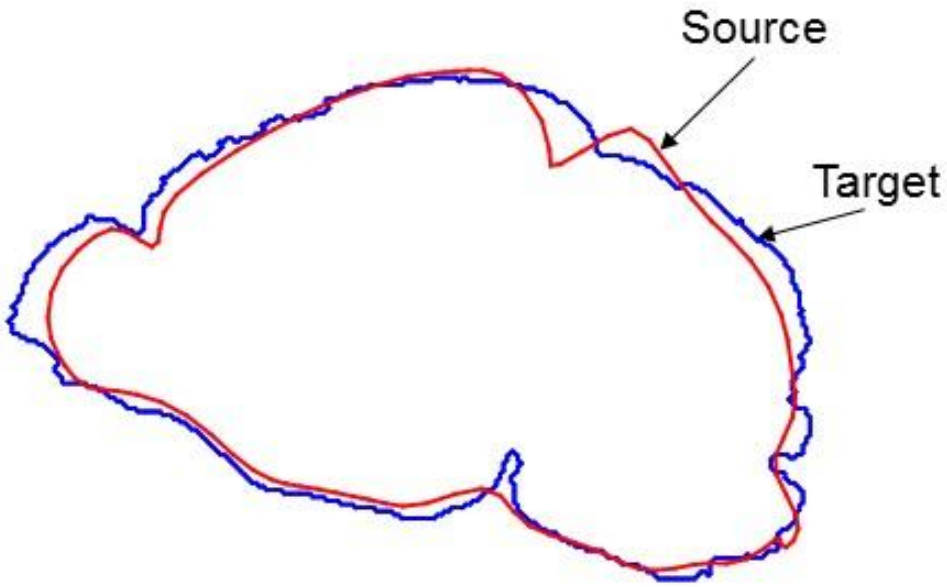
- Alignment
  - Registering source to target by rotation and translation
    - Rigid-body transformations
- Methods
  - Aligning principle directions (PCA)
  - Aligning corresponding points (SVD)
  - Iterative improvement (ICP)
    - Initialize with PCA
    - Alternate between finding correspondences and SVD



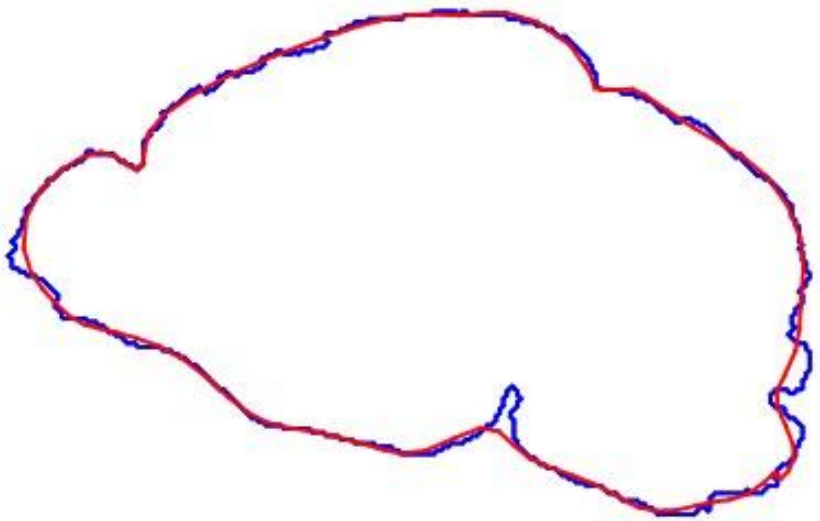
Non-rigid Registration

Washington University in St. Louis
School of Engineering & Applied Science

- Rigid alignment cannot account for shape variance
- Non-rigid deformation is needed for improved fitting



Rigid alignment

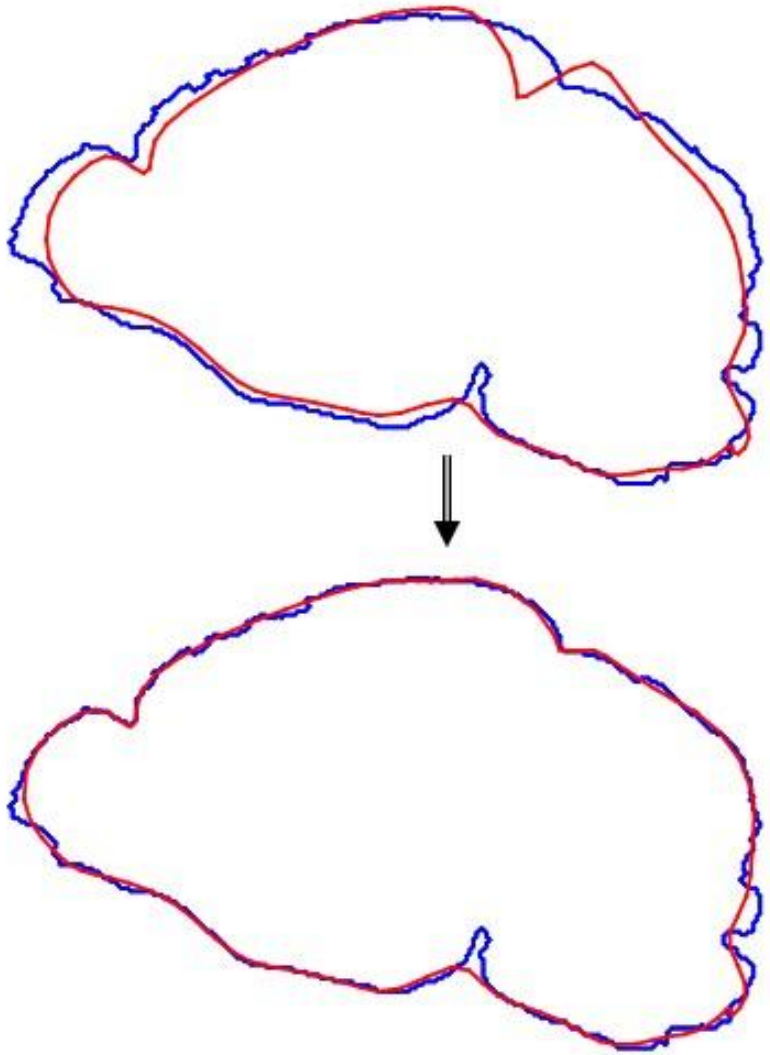


After non-rigid deformation

Non-rigid Registration

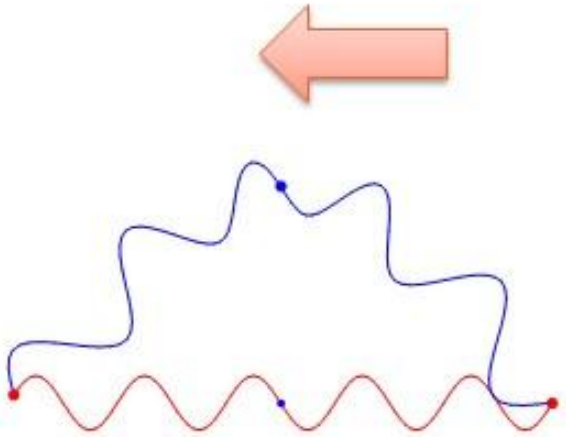
Washington University in St. Louis
School of Engineering & Applied Science

- A minimization problem
  - Minimizing the distance between the deformed source and the target
    - "Fitting term"
  - Minimizing the distortion to the source shape
    - "Distortion term"

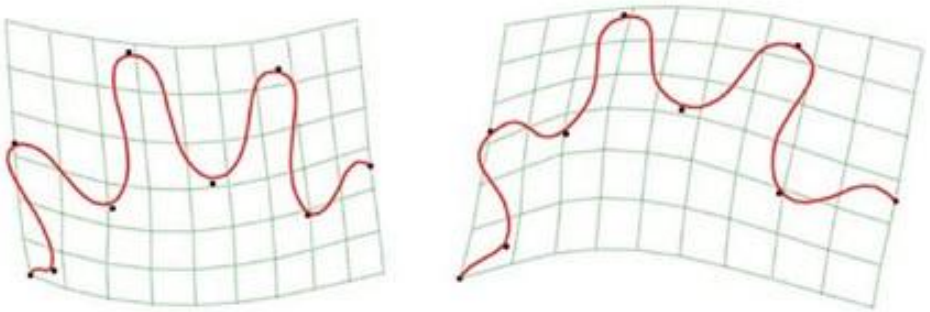


Intrinsic vs. Extrinsic

- Intrinsic methods
  - Deforms points on the source curve/surface
  - App: boundary curve or surface matching



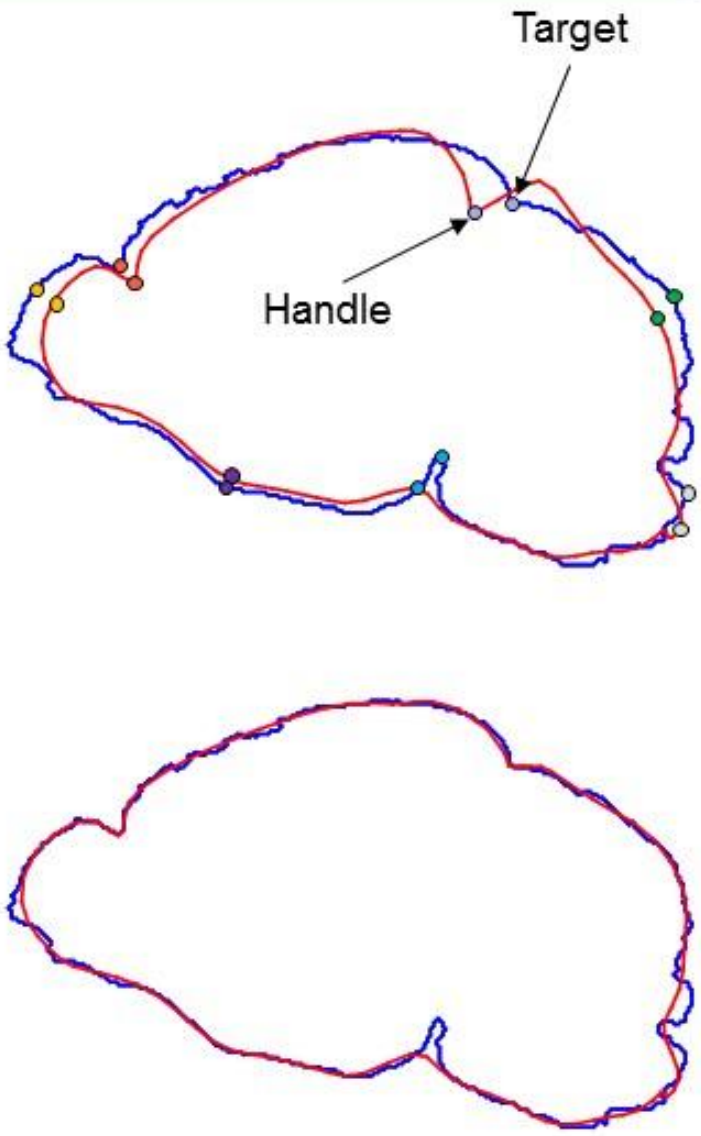
- Extrinsic methods
  - Deforms all points in the space around the source curve/surface
  - App: image or volume matching



Intrinsic Registration

Washington University in St. Louis
School of Engineering & Applied Science

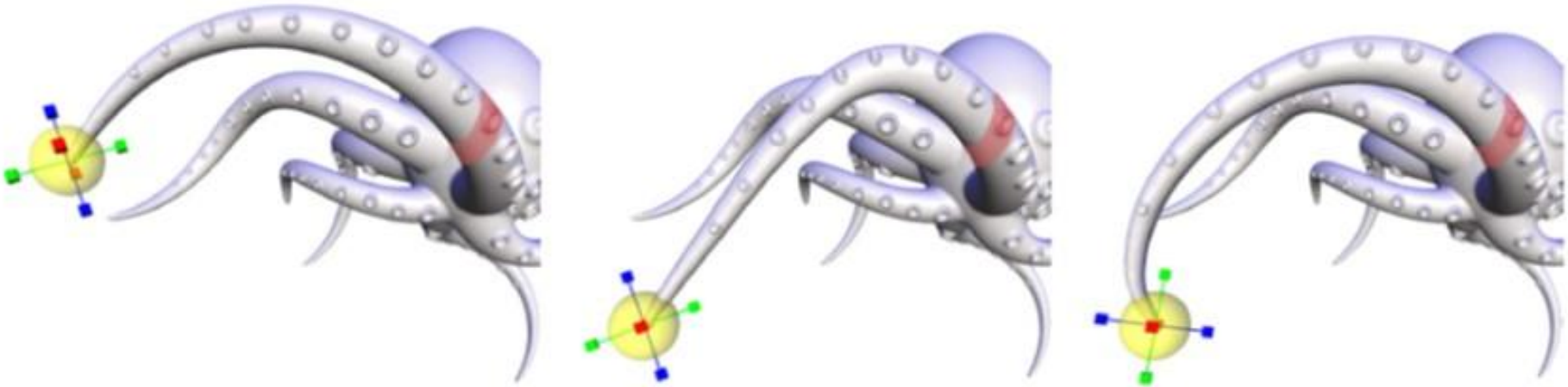
- Given a source and target shape
  - Plus correspondences between some source points (called handles) and target points
- Relocate source points such that:
  - The handles move to their corresponding targets (*fitting term*)
  - The rest of the shape deforms as naturally as possible (*distortion term*)
- ICP-style registration
  - Alternate between finding correspondences (e.g., closest neighbors) and deformation



Laplacian-based Deformation

Washington University in St. Louis
School of Engineering & Applied Science

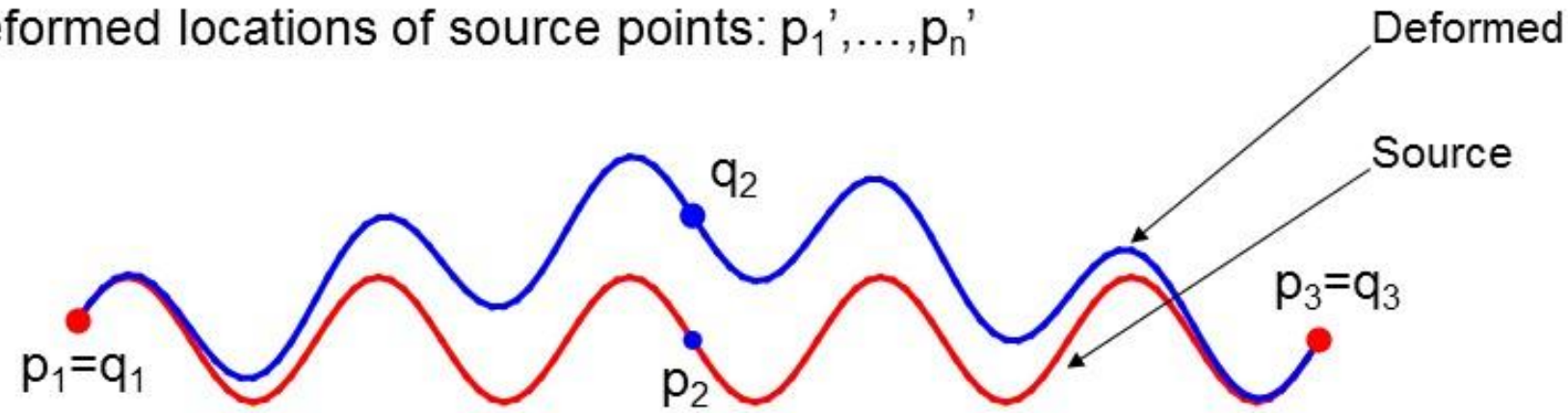
- An intrinsic method used in graphics/animation
  - Simple and efficient (fitting/distortion terms are quadratic)
  - Preserving local shape features



Reference: "Laplacian surface editing", by Sorkine et al., 2004

Setup

- Input
  - Source with n points: $p_1,\ldots,p_n$
    - For notation purpose, assume that the first m points are handles
  - Target location of handles: $q_1,\ldots,q_m$
- Output
  - Deformed locations of source points: $p_1',\ldots,p_n'$



An example with 3 handles, two of which are stationary (red)

Overview

- Finding deformed locations p_i' that minimize:

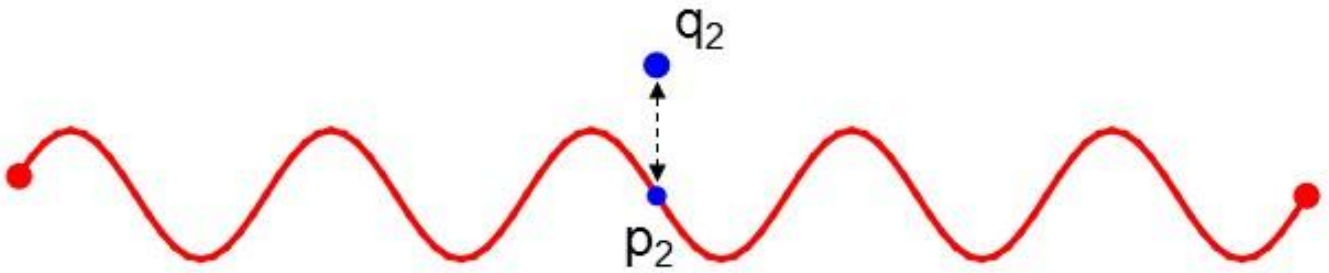
$$\mathbf{E = E_f + E_d}$$

  - E_f: fitting term
    - Measures how close are the deformed handles to the target
  - E_d: distortion term
    - Measures how much the source shape is changed

Fitting Term

- Sum of squared distances to target handle locations

$$E_f = \sum_{i=1}^{m} \|p_i' - q_i\|^2$$

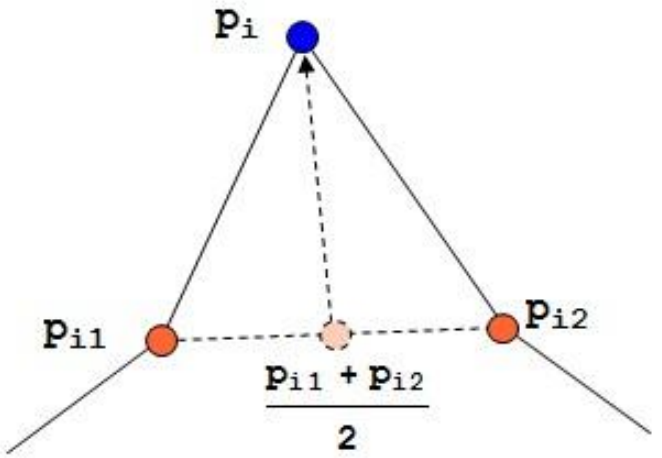


Distortion Term

- Q: How to measure shape locally?
- A: By "bumpiness" at each vertex
  - Laplacian: vector from the centroid of neighbors to the vertex
    - A linear operator over point locations

$$L[p_i] = p_i - \frac{1}{|N_i|} \sum_{j \in N_i} p_j$$

where N_i are indices of neighboring vertices of p_i



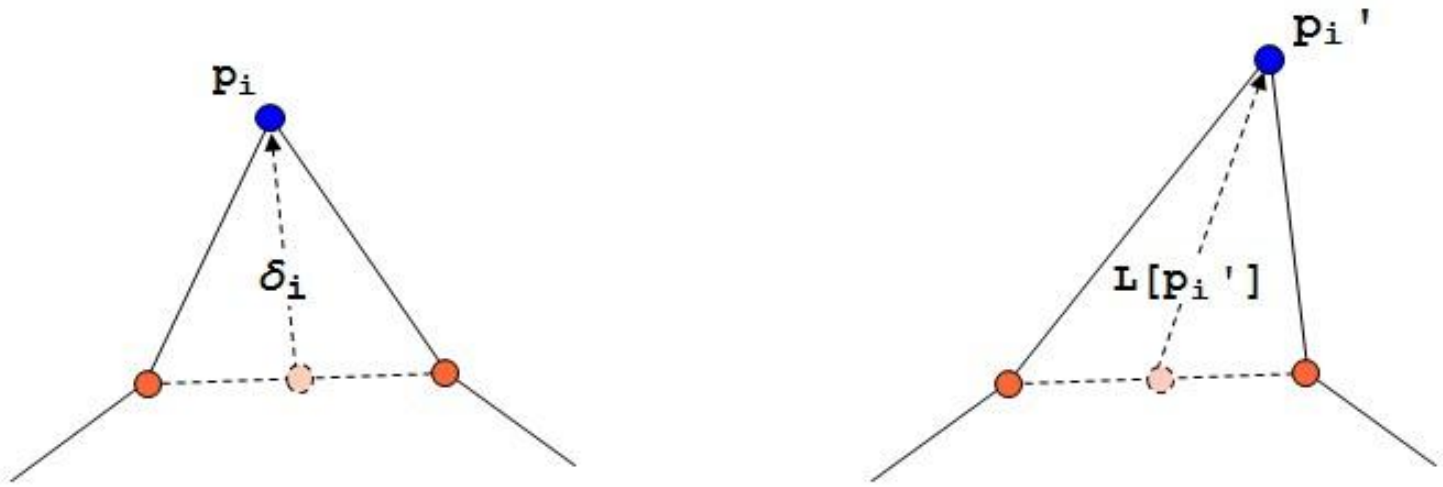
- Recall that in fairing, we minimizes this vector to "smooth out" bumps

Distortion Term

- Minimizing changes in Laplacians during deformation
  - Over all source points

$$E_d = \sum_{i=1}^{n} \| L[p_i'] - \delta_i \|^2$$

δ_i: Laplacian at p_i before deformation



Putting Together

- Finding deformed locations p_i' that minimize:

$$E = E_f + E_d$$

$$= \sum_{i=1}^{m} \|p_i' - q_i\|^2 + \sum_{i=1}^{n} \|L[p_i'] - \delta_i\|^2$$

  - A quadratic equation in terms of variables (p_{ix}', p_{iy}', p_{iz}')
    - q_i, δ_i are constants
    - **L[]** is a linear operator

Quadratic Minimization

- A general form of quadratic minimization:

$$\min \sum_{i=1}^{k} \left\| a_i^T x - b_i \right\|^2$$

  - There are s variables: $x=(x_1,\ldots,x_s)^T$
  - Each $a_1,\ldots, a_k$ is a length-s column vector (linear coefficients)
  - Each $b_1,\ldots, b_k$ is a scalar (constant coefficients)
  - k should be greater than s (so that the problem is over-constrained)

Quadratic Minimization

- Re-writing our minimization in the general form

$$E = E_f + E_d$$

$$= \sum_{i=1}^{m} \|p_i' - q_i\|^2 + \sum_{i=1}^{n} \|L[p_i'] - \delta_i\|^2$$

  - In 2D, there are s=2n variables: $x = (p_{1x}', \ldots, p_{nx}', p_{1y}', \ldots, p_{ny}')^T$
    - In 3D, there are s=3n variables
  - We will next re-write each quadratic term in 2D as $(a_i x - b_i)^2$
    - Can be extended easily to 3D

Quadratic Minimization

- The a_i and b_i in the fitting term

$$E_f = \sum_{i=1}^{m} \|p_i' - q_i\|^2 = \sum_{i=1}^{m} (p_{i_x}' - q_{i_x})^2 + \sum_{i=1}^{m} (p_{i_y}' - q_{i_y})^2$$

  - There are 2m quadratic terms

$$\sum_{i=1}^{2m} (a_i^T x - b_i)^2$$

$$x = (p_{1_x}', \ldots, p_{n_x}', p_{2_y}', \ldots, p_{n_y}')$$

  - In the first set of m terms:
    - For i=1,…,m, $b_i=q_{ix}$, a_i contains all zero, except its (i)th entry is 1.
  - In the second set of m terms:
    - For i=1,…,m, $b_{i+m}=q_{iy}$, a_{i+m} contains all zero, except its (i+n)th entry is 1

Quadratic Minimization

- The a_i and b_i in the fitting term

$$E_f = \sum_{i=1}^{m} \|p_i' - q_i\|^2 = \sum_{i=1}^{m} (p_{i_x}' - q_{i_x})^2 + \sum_{i=1}^{m} (p_{i_y}' - q_{i_y})^2$$

  - There are 2m quadratic terms

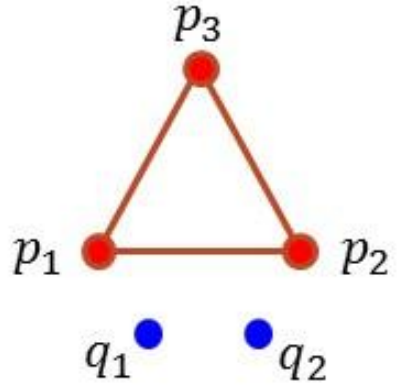
$$\sum_{i=1}^{2m} (a_i^T x - b_i)^2$$

  - Example with 3 vertices and 2 fitting constraints (n=3; m=2):

$$a_1^T = (1\ 0\ 0\ 0\ 0\ 0)$$
$$a_2^T = (0\ 1\ 0\ 0\ 0\ 0)$$
$$a_3^T = (0\ 0\ 0\ 1\ 0\ 0)$$
$$a_4^T = (0\ 0\ 0\ 0\ 1\ 0)$$

$$x = \begin{pmatrix} p_{1_x}' \\ p_{2_x}' \\ p_{3_x}' \\ p_{1_y}' \\ p_{2_y}' \\ p_{3_y}' \end{pmatrix}$$

$$b_1 = q_{1_x}$$
$$b_2 = q_{2_x}$$
$$b_3 = q_{1_y}$$
$$b_4 = q_{2_y}$$



Quadratic Minimization

- The a_i and b_i in the distortion term:

$$\mathbf{L}[\mathbf{p}_i] = \mathbf{p}_i - \frac{1}{|\mathbf{N}_i|}\sum_{j\in\mathbf{N}_i}\mathbf{p}_j$$

$$\mathbf{E}_d = \sum_{i=1}^{n}\|\mathbf{L}[\mathbf{p}_i'] - \delta_i\|^2 = \sum_{i=1}^{n}(\mathbf{L}[\mathbf{p}_{i_x}'] - \delta_{i_x})^2 + \sum_{i=1}^{n}\left(\mathbf{L}[\mathbf{p}_{i_y}'] - \delta_{i_y}\right)^2$$

  - There are 2n quadratic terms $\sum_{i=1}^{2n}\left(\mathbf{a}_i^T\mathbf{x} - \mathbf{b}_i\right)^2$
  - The first set of n terms:

$$\mathbf{x} = (\mathbf{p}_{1_x}', \ldots, \mathbf{p}_{n_x}', \mathbf{p}_{2_y}', \ldots, \mathbf{p}_{n_y}')$$

    - For i=1,…,n, a_i is all zero except the (i)th entry is 1, the (j)th entries are $-1/|N_i|$ for all $j\in N_i$, and $b_i=\delta_{ix}$
  - The second set of n terms:
    - For i=1,…,n, a_{i+n} is all zero except the (i+n)th entry is 1, the (j+n)th entries are $-1/|N_i|$ for all $j\in N_i$, and $b_{i+n}=\delta_{iy}$

Quadratic Minimization

- The a_i and b_i in the distortion term:

$$L[p_i] = p_i - \frac{1}{|N_i|} \sum_{j \in N_i} p_j$$

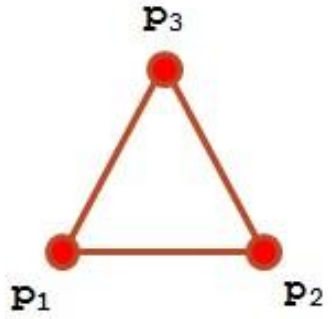
$$E_d = \sum_{i=1}^{n} \| L[p_i'] - \delta_i \|^2 = \sum_{i=1}^{n} (L[p_{i_x}'] - \delta_{i_x})^2 + \sum_{i=1}^{n} (L[p_{i_y}'] - \delta_{i_y})^2$$

  - There are 2n quadratic terms $\sum_{i=1}^{2n} (a_i^T x - b_i)^2$
  - Example with 3 vertices (n=3):

$$a_1^T = \left(1 \;\; -\tfrac{1}{2} \;\; -\tfrac{1}{2} \;\; 0 \;\; 0 \;\; 0\right)$$
$$a_2^T = \left(-\tfrac{1}{2} \;\; 1 \;\; -\tfrac{1}{2} \;\; 0 \;\; 0 \;\; 0\right)$$
$$a_3^T = \left(-\tfrac{1}{2} \;\; -\tfrac{1}{2} \;\; 1 \;\; 0 \;\; 0 \;\; 0\right)$$
$$a_4^T = \left(0 \;\; 0 \;\; 0 \;\; 1 \;\; -\tfrac{1}{2} \;\; -\tfrac{1}{2}\right)$$
$$a_5^T = \left(0 \;\; 0 \;\; 0 \;\; -\tfrac{1}{2} \;\; 1 \;\; -\tfrac{1}{2}\right)$$
$$a_6^T = \left(0 \;\; 0 \;\; 0 \;\; -\tfrac{1}{2} \;\; -\tfrac{1}{2} \;\; 1\right)$$

$$x = \begin{pmatrix} p_{1_x}' \\ p_{2_x}' \\ p_{3_x}' \\ p_{1_y}' \\ p_{2_y}' \\ p_{3_y}' \end{pmatrix}$$

$$b_1 = \delta_{1x}$$
$$b_2 = \delta_{2x}$$
$$b_3 = \delta_{3x}$$
$$b_4 = \delta_{1y}$$
$$b_5 = \delta_{2y}$$
$$b_6 = \delta_{3y}$$



Quadratic Minimization

- To solve:

$$\min \sum_{i=1}^{k} \left\| \mathbf{a}_i^T \mathbf{x} - \mathbf{b}_i \right\|^2$$

- Re-write in matrix form:

$$\min \left\| \mathbf{A}\mathbf{x} - \mathbf{B} \right\|^2$$

where $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_k^T \end{pmatrix}$ is a k by s matrix

$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_k \end{pmatrix}$ is a length-k vector

Quadratic Minimization

- The minimizer is where the partial derivatives are all zero

$$0 = \frac{\partial \|Ax - B\|^2}{\partial x}$$
$$= 2A^T Ax - 2A^T B \quad \Longrightarrow \quad A^T Ax = A^T B$$

  - To solve for x in this equation:
    - Taking matrix inverse (good for small s, but numerically unstable for large s)

$$x = \left(A^T A\right)^{-1} A^T B$$

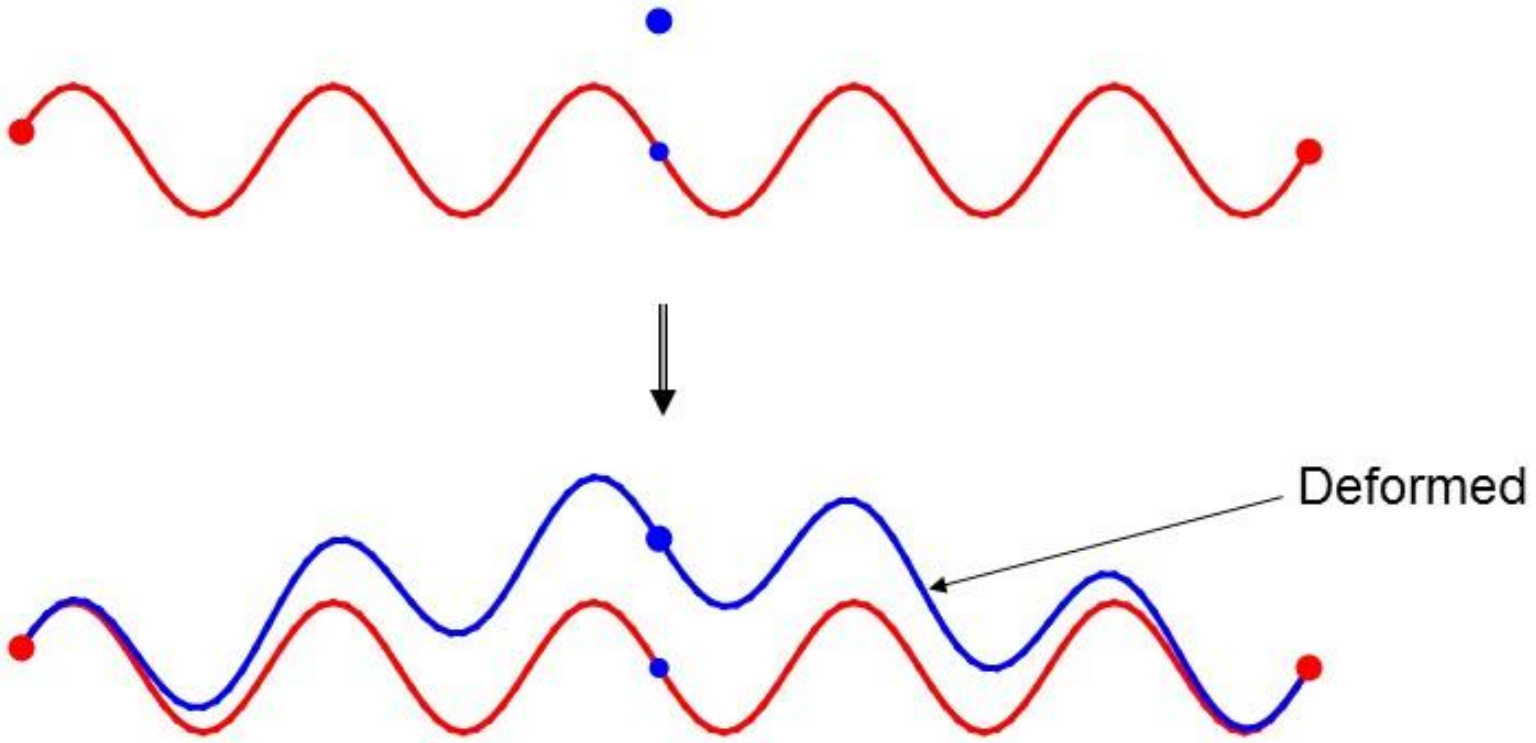
    - Using specialized linear system solver (LinearSolve in *Mathematica*, Eigen/TNT/LAPACK in C)

Summary

- Compute Laplacians (δ_i)
- Construct coefficients (a_i, b_i)
  - Put them into matrices (A,B)
- Solve (x)
  - $\mathbf{A^T A x = A^T B}$

$$\mathbf{x} = (\mathbf{p}_{1_x}', \ldots, \mathbf{p}_{n_x}', \mathbf{p}_{2_y}', \ldots, \mathbf{p}_{n_y}')$$

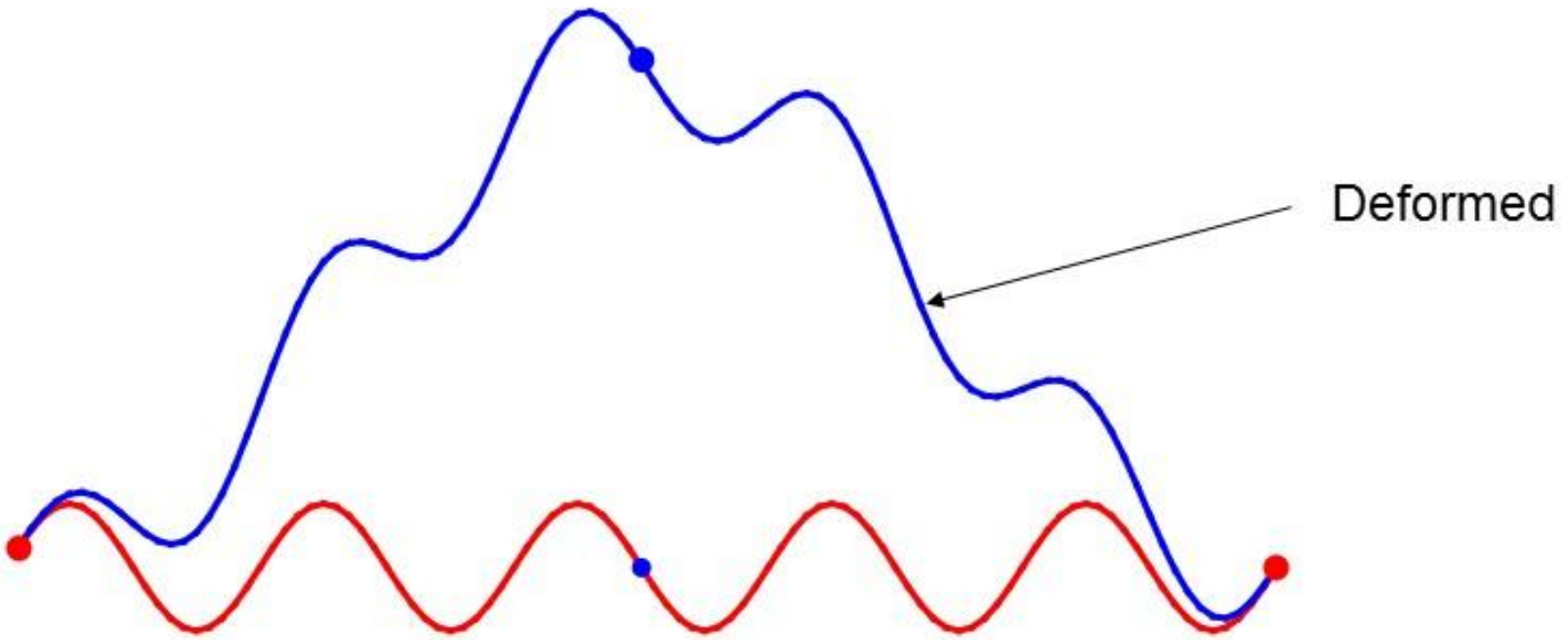
Results



A small deformation

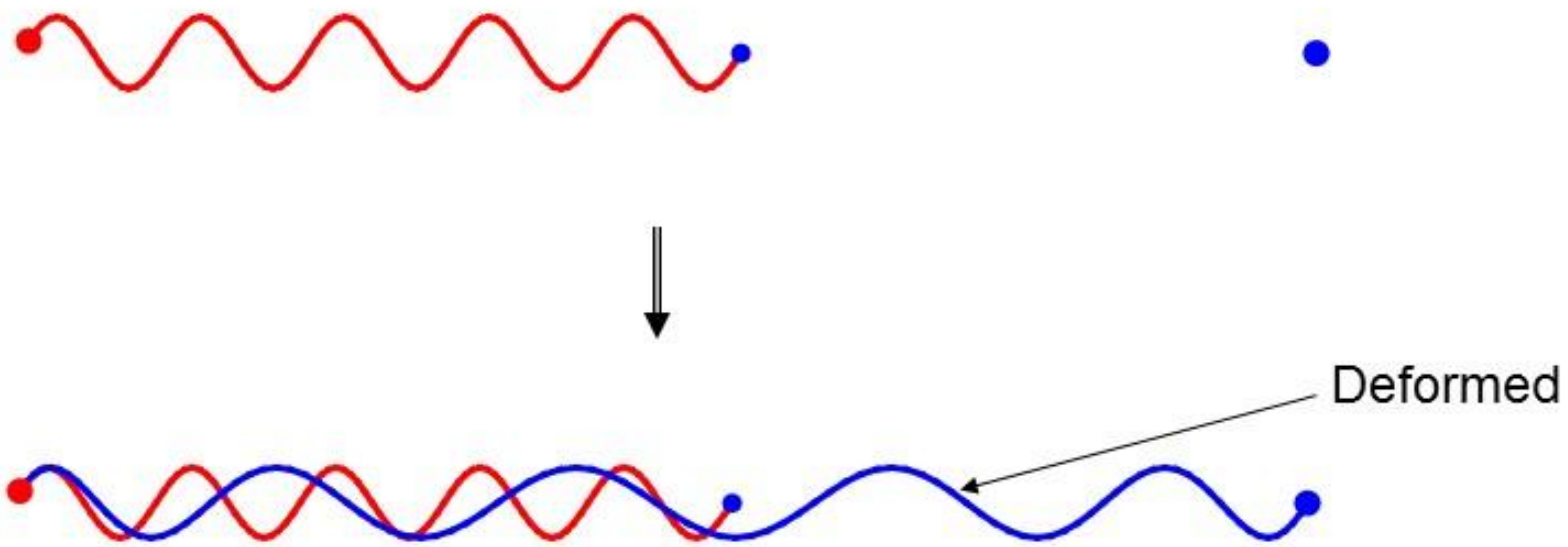
Results

Washington University in St. Louis
School of Engineering & Applied Science



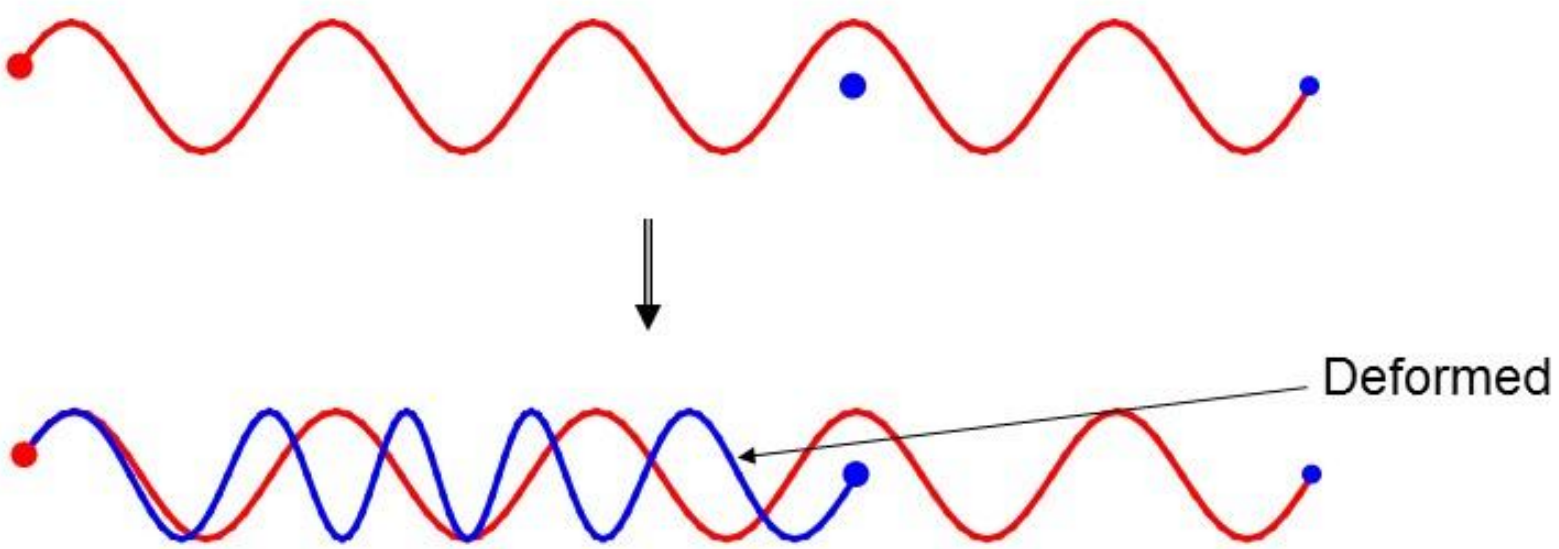
A larger deformation

Results



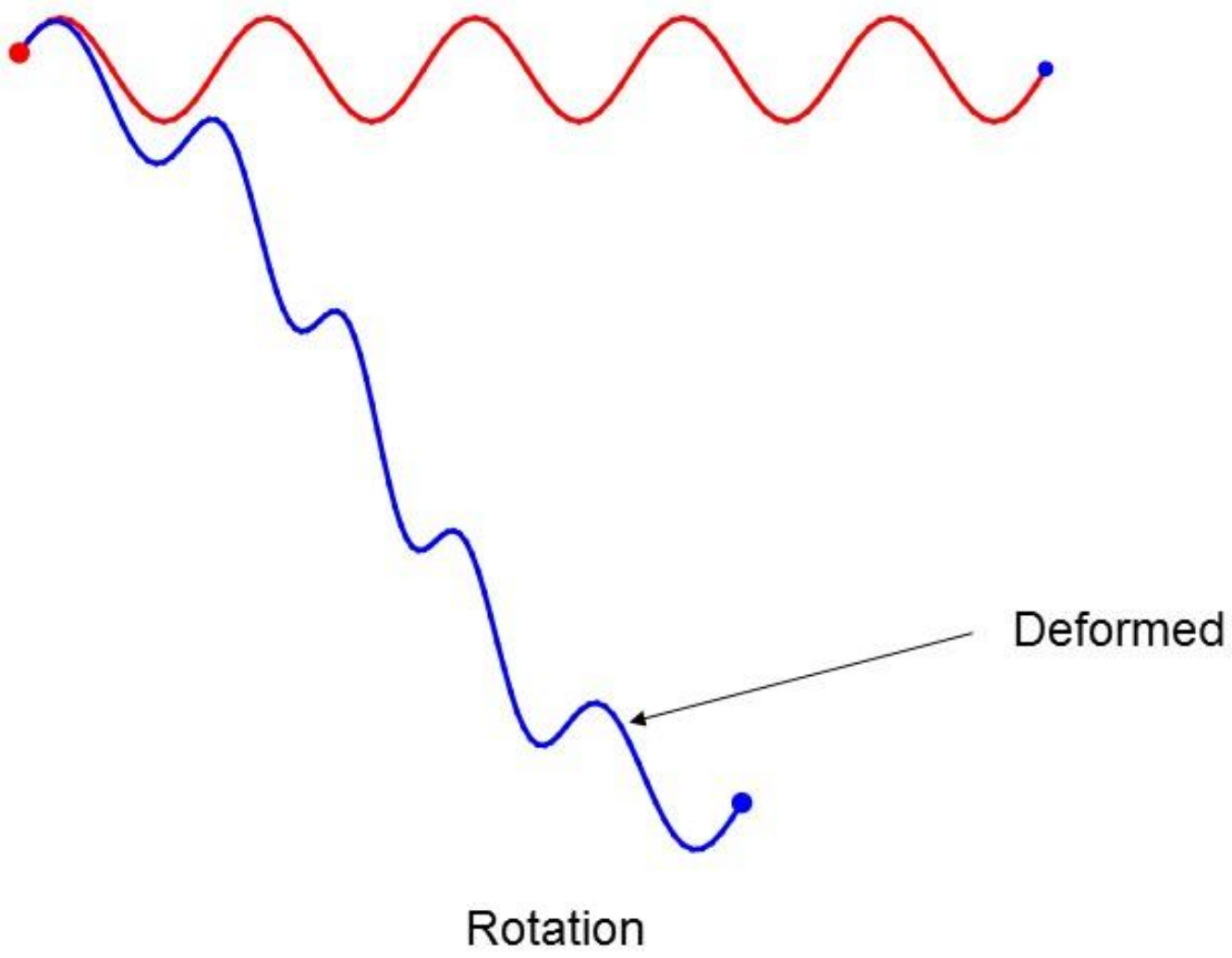
Stretching

Results



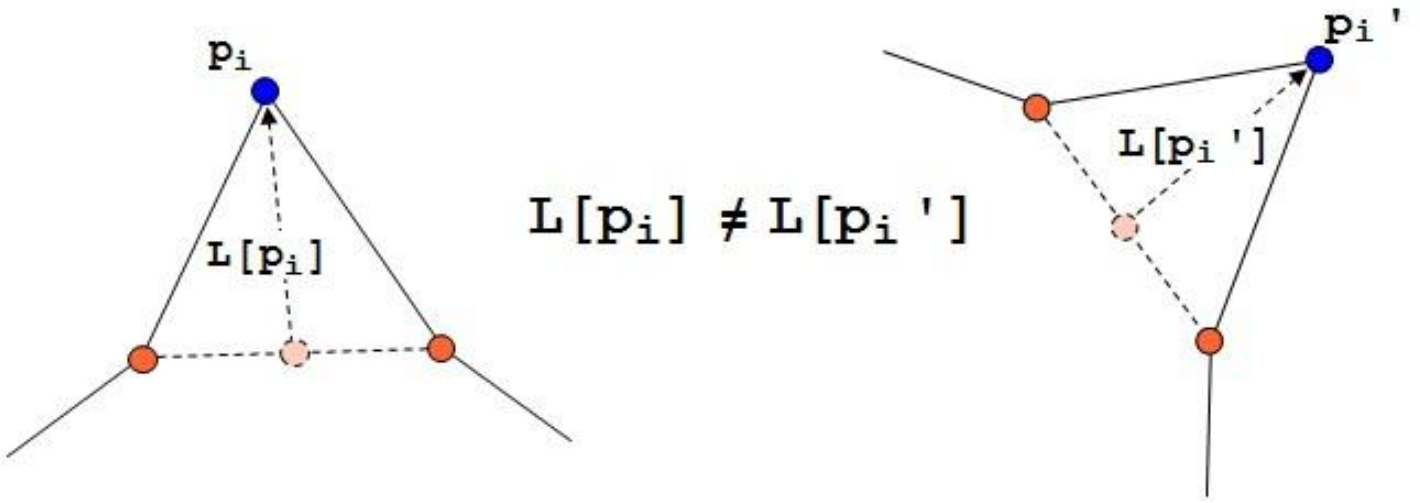
Shrinking

Results



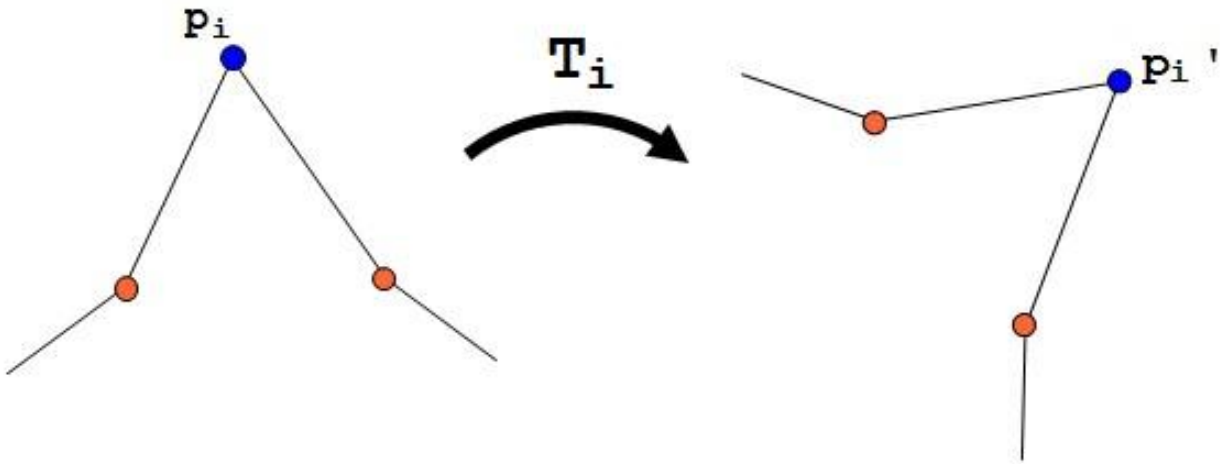
Discussion

- Limitations
  - Local features don't rotate or scale with the model
- Reason: Laplacian is not invariant under rotation or scale
  - Two bumps that differ by rotation or scale result in non-zero distortion



A Better Distortion Term

- Not penalizing rotation and scaling of local features
  - Estimating how the local neighborhood is transformed



  - Transforming the original Laplacian vectors before comparing to the deformed Laplacians

$$E_d = \sum_{i=1}^{n} \| L[p_i'] - \boxed{T_i} \delta_i \|^2$$

Catch 22

- Q: How do we find T_i before we know the deformed shape?
- A: Represent T_i as a function of the unknown variables
  - A linear function of p_i', just like L

$$E_d = \sum_{i=1}^{n} \| L[p_i'] - T_i \delta_i \|^2$$

- We will focus in the derivations of the 2D case
  - 3D results will be briefly presented at the end

Transformation Matrices (2D)

Washington University in St. Louis
School of Engineering & Applied Science

- Homogeneous coordinates
  - A 2D point: (x,y,1)
    - A 2D vector: (x,y,0)
  - A 3D point: (x,y,z,1)
    - A 3D vector: (x,y,z,0)

Transformation Matrices (2D)

- Translation
  - Cartesian coordinates: vector addition

$$\begin{pmatrix} p_x' \\ p_y' \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

  - Homogeneous coordinates: matrix product

$$\begin{pmatrix} p_x' \\ p_y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

Transformation Matrices (2D)

- Isotropic scaling
  - Cartesian coordinates: vector scaling

$$\begin{pmatrix} p_x' \\ p_y' \end{pmatrix} = s * \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

  - Homogeneous coordinates: matrix product

$$\begin{pmatrix} p_x' \\ p_y' \\ 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

Transformation Matrices (2D)

- Rotation
  - Cartesian coordinates: matrix product

$$\begin{pmatrix} p_x' \\ p_y' \end{pmatrix} = \begin{pmatrix} \text{Cos}[\alpha] & -\text{Sin}[\alpha] \\ \text{Sin}[\alpha] & \text{Cos}[\alpha] \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \end{pmatrix}$$

  - Homogeneous coordinates: matrix product

$$\begin{pmatrix} p_x' \\ p_y' \\ 1 \end{pmatrix} = \begin{pmatrix} \text{Cos}[\alpha] & -\text{Sin}[\alpha] & 0 \\ \text{Sin}[\alpha] & \text{Cos}[\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

Transformation Matrices (2D)

- Summary of elementary similarity transformations
  - To perform a sequence of transformations: multiple the corresponding matrices in order

$$\begin{pmatrix} p_x' \\ p_y' \\ 1 \end{pmatrix} = M \cdot \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}$$

$$\mathrm{Trs}[v] = \begin{pmatrix} 1 & 0 & v_x \\ 0 & 1 & v_y \\ 0 & 0 & 1 \end{pmatrix}$$ Translation by vector v

$$\mathrm{Scl}[s] = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ Scaling by scalar s

$$\mathrm{Rot}[\alpha] = \begin{pmatrix} \mathrm{Cos}[\alpha] & -\mathrm{Sin}[\alpha] & 0 \\ \mathrm{Sin}[\alpha] & \mathrm{Cos}[\alpha] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ Rotation by angle α

Similarity Transforms (2D)

- General similarity transformations

$$T = \begin{pmatrix} a & w & t_x \\ -w & a & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

  - The product of any set of similarity matrices can be written this way
  - Any choice of (a, w, t_x, t_y) can be written as a sequence of rotation, isotropic scaling and translation

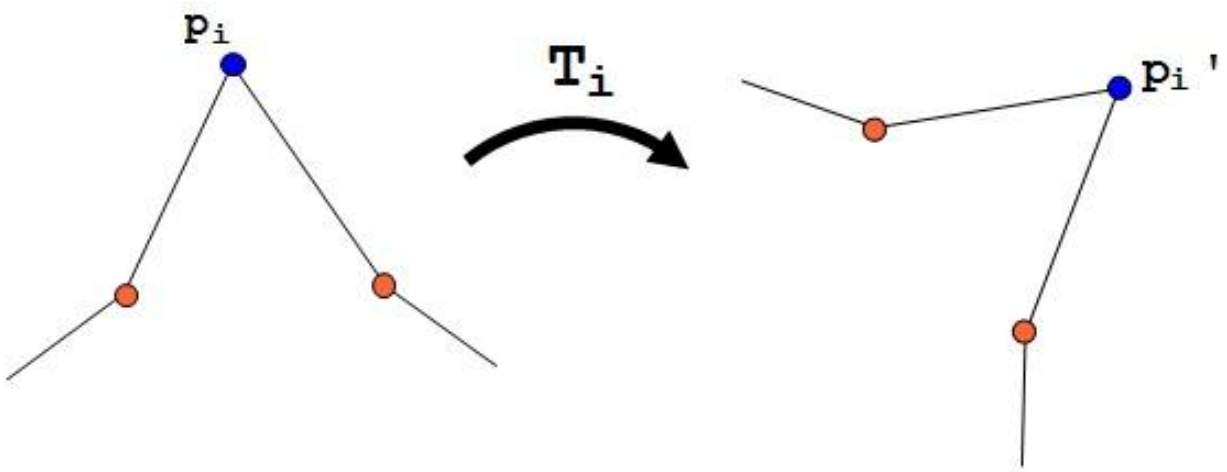
$$\begin{pmatrix} a & w & t_x \\ -w & a & t_y \\ 0 & 0 & 1 \end{pmatrix} = \text{Trs}[t_x, t_y].\text{Scl}\left[\sqrt{a^2 + w^2}\right].\text{Rot}\left[\text{ArcTan}\left[\frac{a}{-w}\right]\right]$$

    - Note that a and w can't be both zero

Computing T_i (2D)

- Suppose we know the deformed locations p_i'
- Compute T_i as the similarity transform that best fits the neighborhood of p_i to that of p_i'

$$\min \left(\| T_i p_i - p_i' \|^2 + \sum_{j \in N_i} \| T_i p_j - p_j' \|^2 \right)$$



Computing T_i (2D)

- Suppose we know the deformed locations p_i'
- Compute T_i as the similarity transform that best fits the neighborhood of p_i to that of p_i'

$$\min \left(\| T_i p_i - p_i' \|^2 + \sum_{j \in N_i} \| T_i p_j - p_j' \|^2 \right)$$

- This is a quadratic minimization problem for entries of T_i
  - E.g., a, w, t_x, t_y

Computing T_i (2D)

- The matrix form of the minimization is:

$$\min \left\| C \begin{pmatrix} a \\ w \\ t_x \\ t_y \end{pmatrix} - \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix} \right\|^2$$

where $C = \begin{pmatrix} p_{i_x} & p_{i_y} & 1 & 0 \\ p_{i_y} & -p_{i_x} & 0 & 1 \\ p_{i1_x} & p_{i1_y} & 1 & 0 \\ p_{i1_y} & -p_{i1_x} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$ is a $2|N_i|+2$ by 4 matrix, and N_i={i1, i2,…} are indices of neighboring vertices of p_i

Computing T_i (2D)

- By quadratic minimization:

$$\begin{pmatrix} a \\ w \\ t_x \\ t_y \end{pmatrix} = (C^T C)^{-1} C^T \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix}$$

  - Linear expressions of variables (p_{ix}' , p_{iy}')

Distortion Term (2D)

- Two parts of each distortion term: $\|L[p_i'] - T_i\,\delta_i\|^2$
  - Transformed Laplacian:

$$T_i\,\delta_i = D \begin{pmatrix} a \\ w \\ t_x \\ t_y \end{pmatrix} = D\,(C^T C)^{-1} C^T \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix} \quad \text{where } D = \begin{pmatrix} \delta_{i_x} & \delta_{i_y} & 0 & 0 \\ \delta_{i_y} & -\delta_{i_x} & 0 & 0 \end{pmatrix}$$

  - Laplacian of the deformed locations:

$$L[p_i'] = L \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix} \quad \text{where } L = \begin{pmatrix} 1 & 0 & -\frac{1}{|N_i|} & 0 & \cdots \\ 0 & 1 & 0 & -\frac{1}{|N_i|} & \cdots \end{pmatrix}$$

is a 2 by $2|N_i|+2$ matrix

Distortion Term (2D)

- Putting together:

$$E_d = \sum_{i=1}^{n} \| L[p_i'] - T_i \, \delta_i \|^2$$

$$= \sum_{i=1}^{n} \left(H[[1]] \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix} \right)^2 + \sum_{i=1}^{n} \left(H[[2]] \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i1_x}' \\ p_{i1_y}' \\ \vdots \end{pmatrix} \right)^2$$

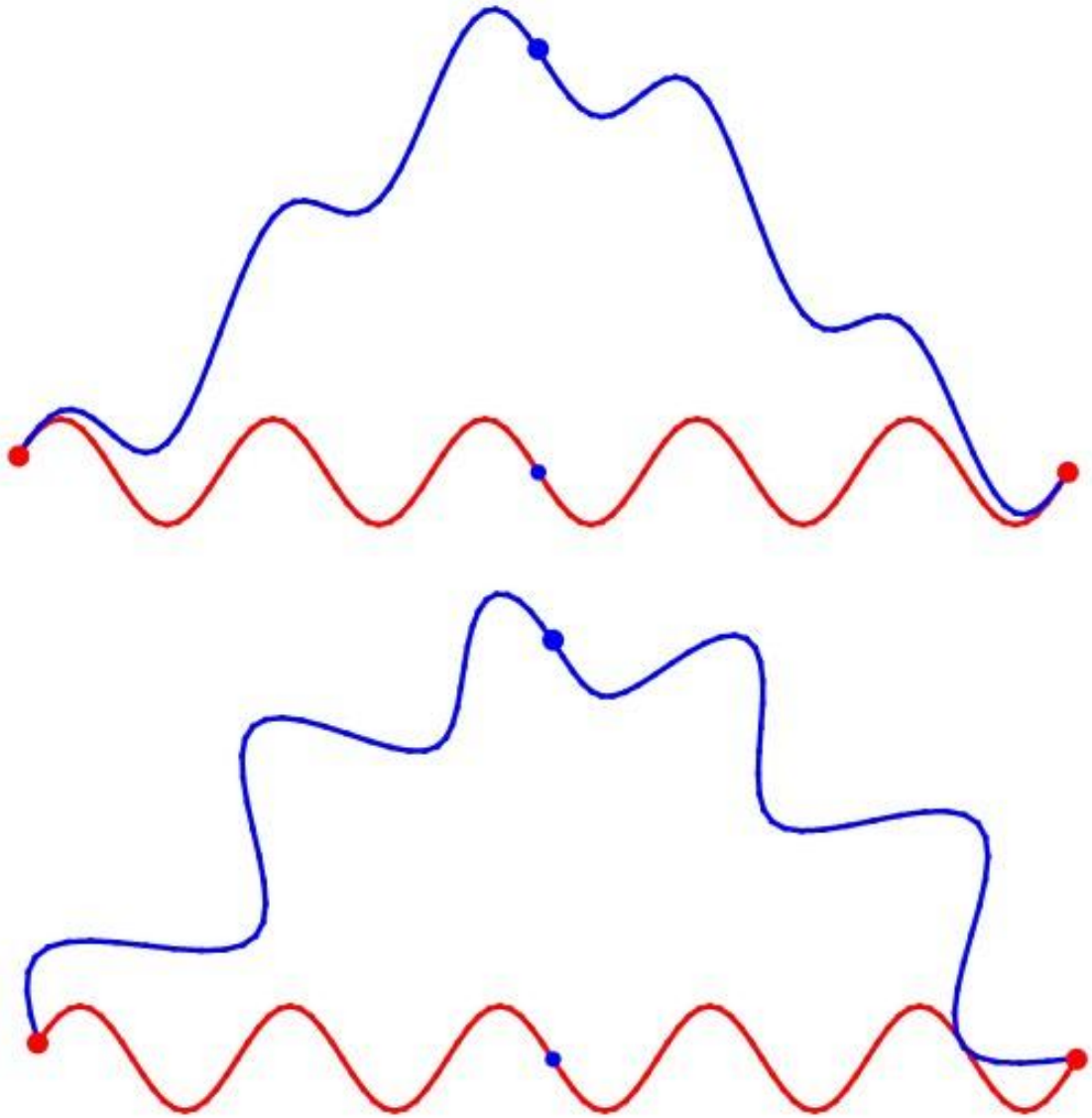
where $H = L - D \left(C^T C \right)^{-1} C^T$

and $H[[1]]$, $H[[2]]$ are its rows

  - They form 2n quadratic terms $(a_i x - b_i)^2$ for $x = (p_{1x}', \ldots, p_{nx}', p_{1y}', \ldots, p_{ny}')^T$
    - All b_i are zero
    - Each a_i can be extracted from H

Results (2D)

Washington University in St. Louis
School of Engineering & Applied Science

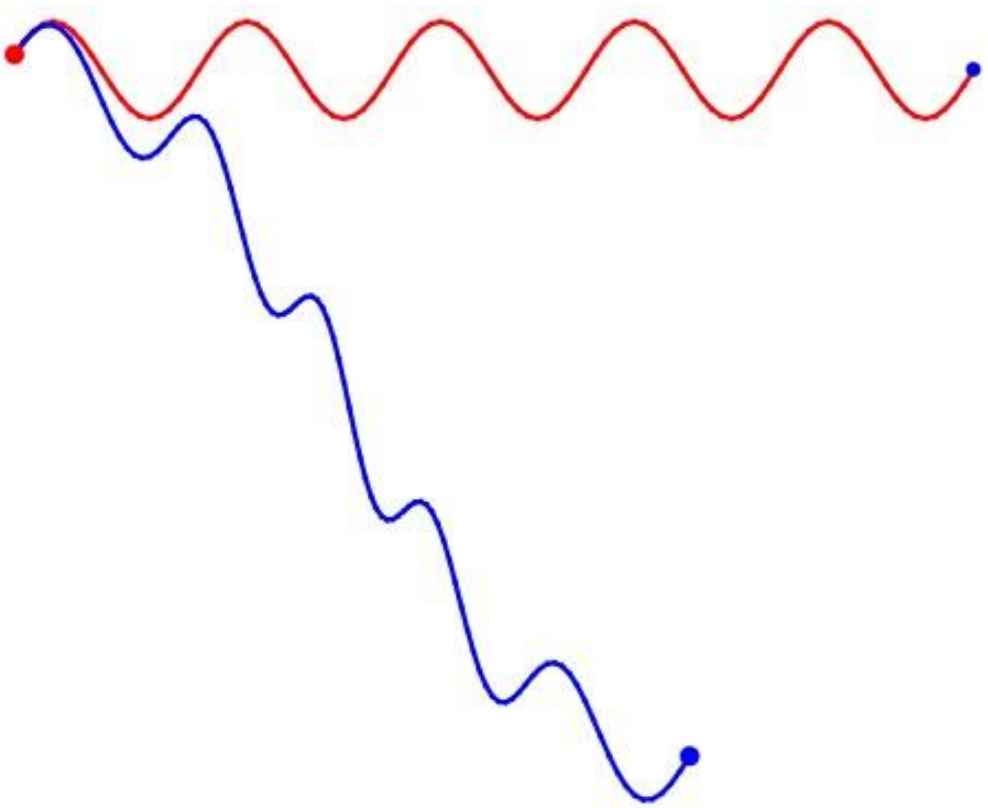


Old distortion term

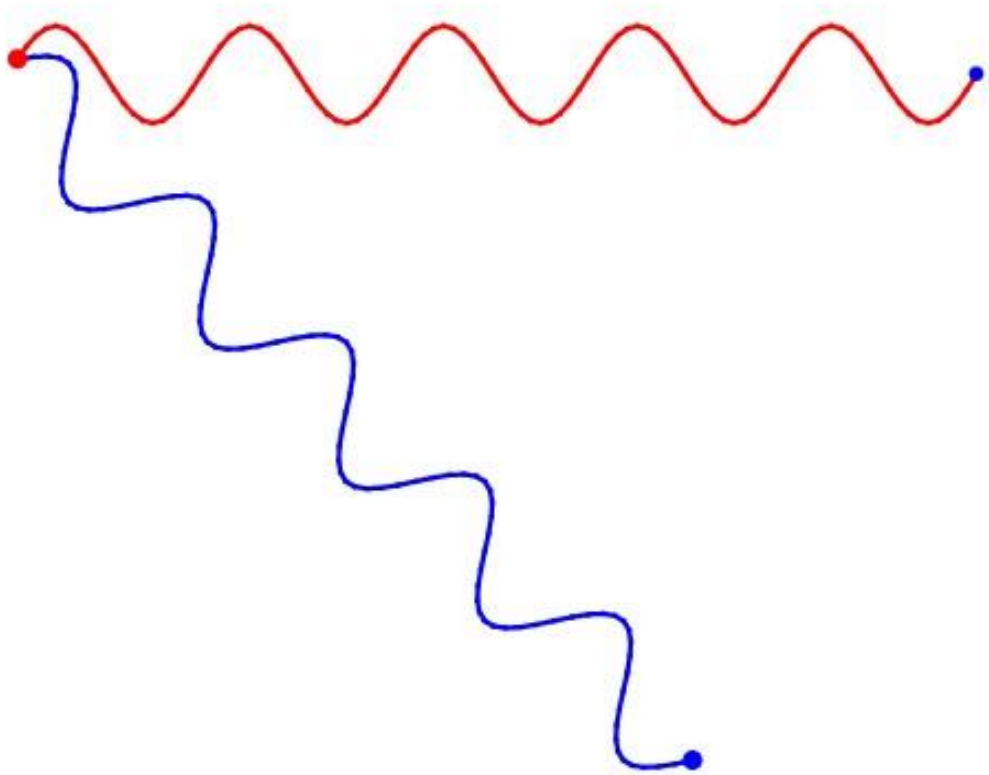
New distortion term

Results (2D)

Washington University in St. Louis
School of Engineering & Applied Science



Old distortion term

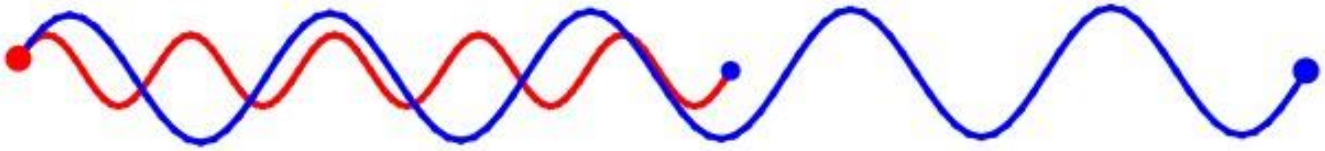


New distortion term

Results (2D)

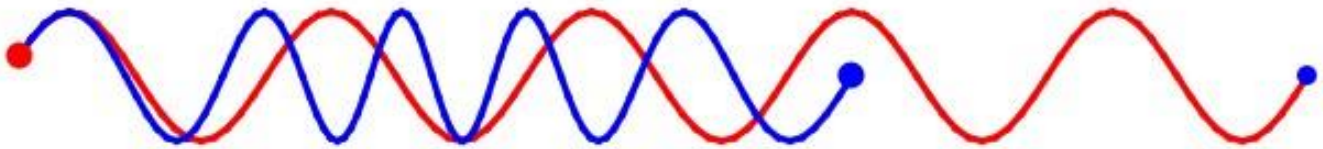


Old distortion term

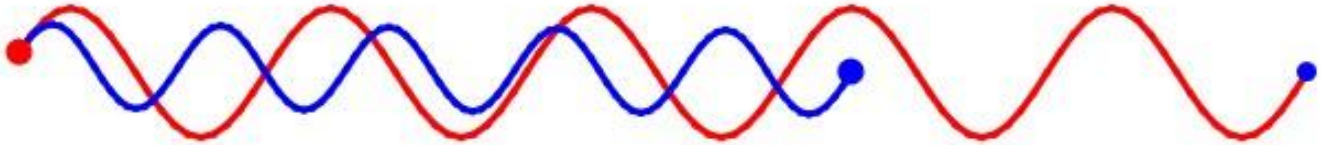


New distortion term

Results (2D)



Old distortion term



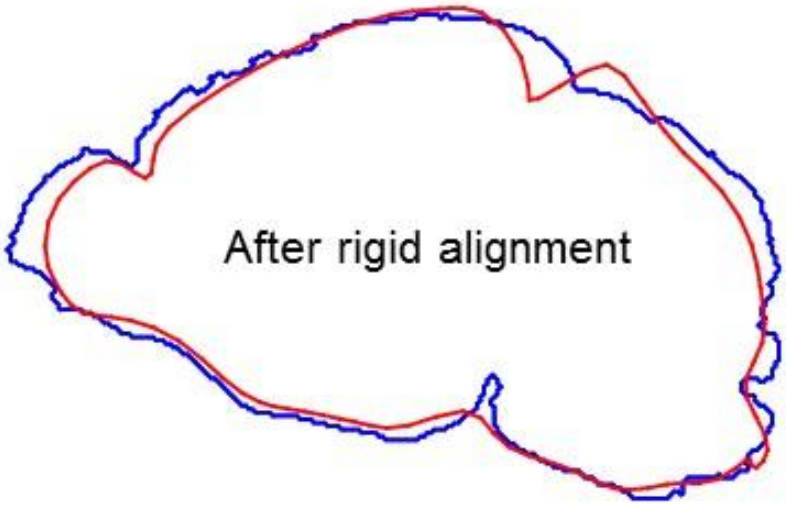
New distortion term

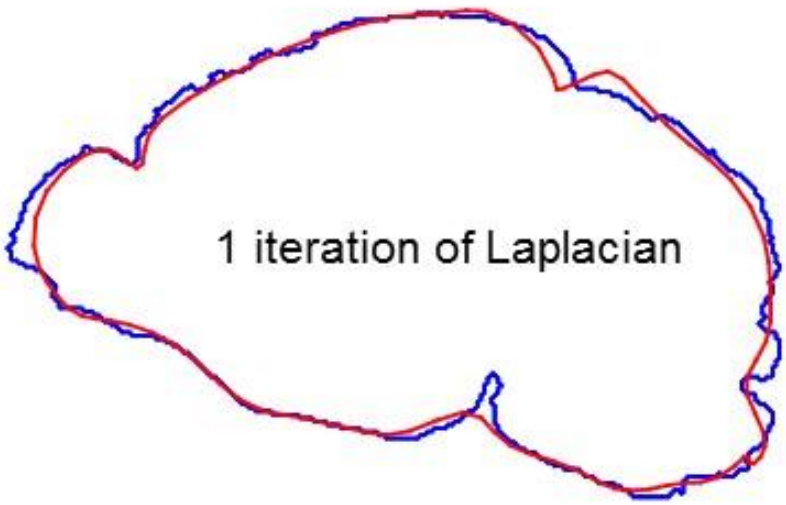
Registration

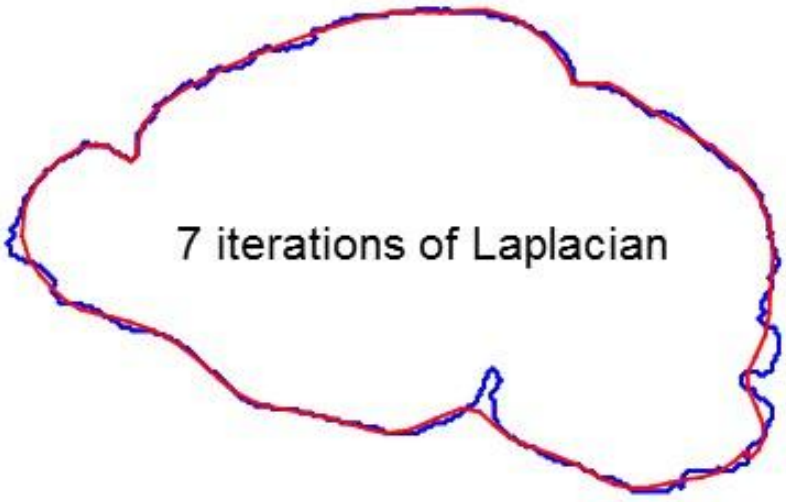
- Use nearest neighbors as corresponding target locations
  - Assuming the source is already close to the target

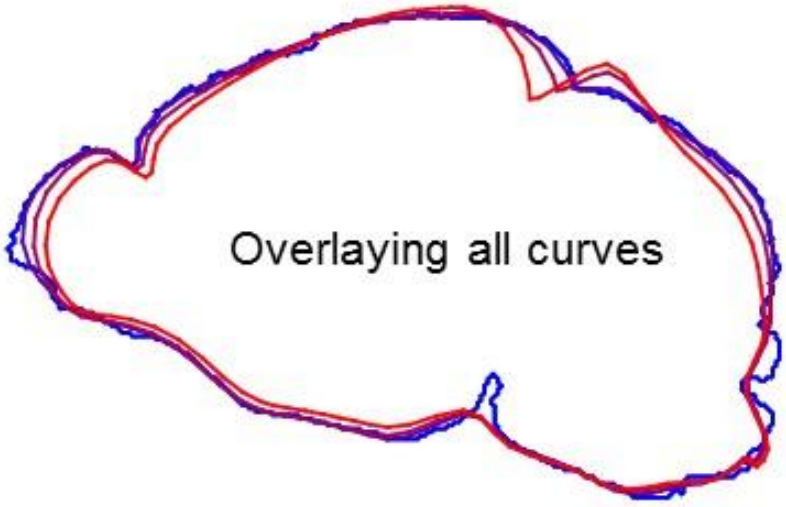
- Iterative closest point (ICP)
  - 1. For each point on the source p_i, treat it as a handle, and assign its closest point on the target as its target location q_i.
  - 2. Compute Laplacian-based deformation.
  - 3. Repeat step (1) until a termination criteria is met.

Result


After rigid alignment


1 iteration of Laplacian

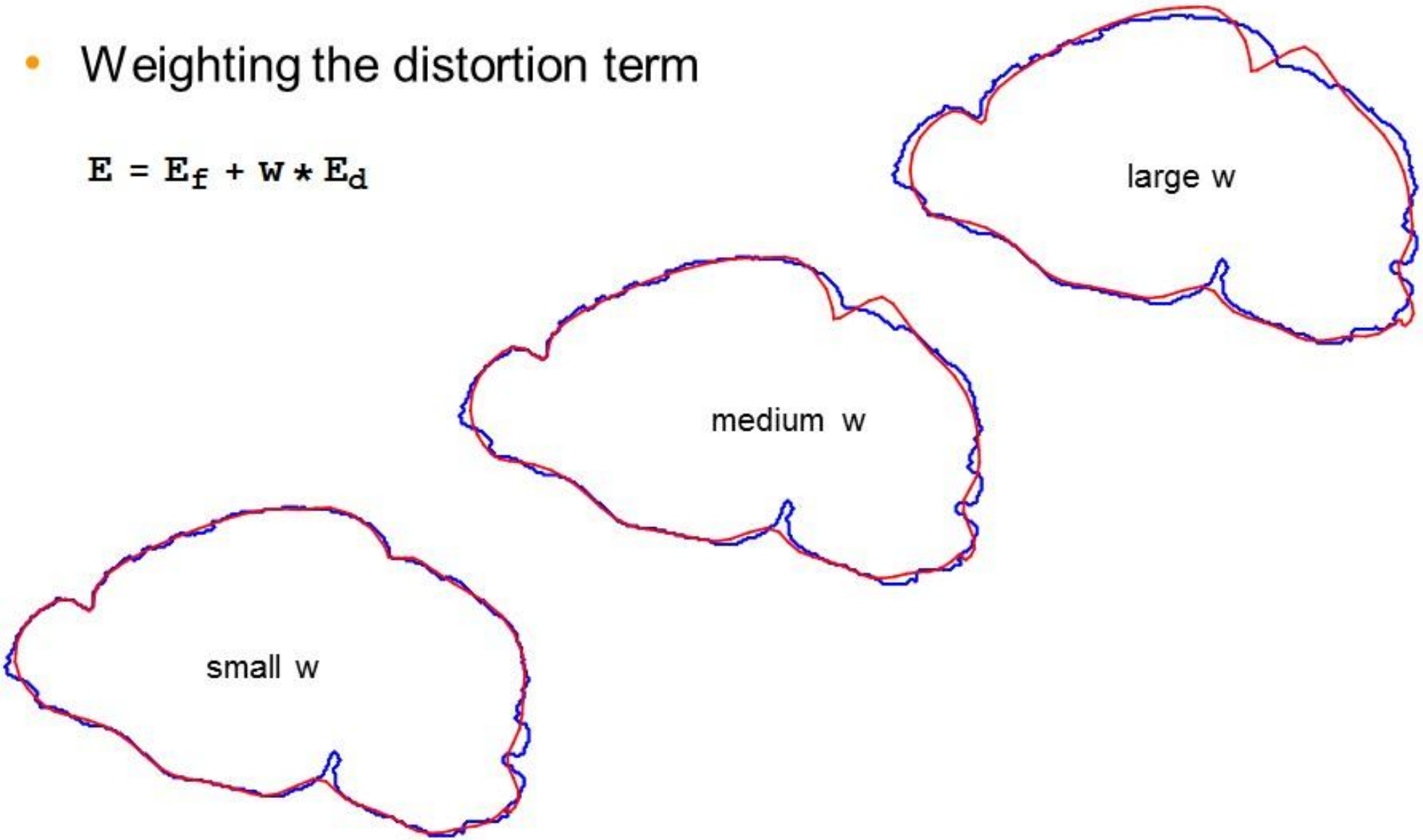

7 iterations of Laplacian


Overlaying all curves

Result

- Weighting the distortion term

$$E = E_f + w * E_d$$



Similarity Transforms (3D)

- Elementary transformation matrices
  - To perform a sequence of transformations: take the product of these matrices

$$\begin{pmatrix} p_x' \\ p_y' \\ p_z' \\ 1 \end{pmatrix} = M \cdot \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

$$\text{Trs}[v] = \begin{pmatrix} 1 & 0 & 0 & v_x \\ 0 & 1 & 0 & v_y \\ 0 & 0 & 1 & v_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Translation by vector v

$$\text{Scl}[s] = \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Scaling by scalar s

$$\text{Rot}[X, \alpha] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}[\alpha] & -\text{Sin}[\alpha] & 0 \\ 0 & \text{Sin}[\alpha] & \text{Cos}[\alpha] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Rotation by angle α around X axis

Similarity Transforms (3D)

- General similarity transformations in 3D

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

  - Approximates the product of a set of elementary matrices
    - Up to a small rotation angle
    - May introduce skewing for large rotations

Computing T_i (3D)

Washington University in St. Louis
School of Engineering & Applied Science

- Assuming known deformation, by quadratic minimization:

$$\begin{pmatrix} s \\ h_1 \\ h_2 \\ h_3 \\ t_x \\ t_y \\ t_z \end{pmatrix} = \left(C^T C\right)^{-1} C^T \begin{pmatrix} p_{i_x}' \\ p_{i_y}' \\ p_{i_z}' \\ p_{i1_x}' \\ p_{i1_y}' \\ p_{i1_z}' \\ \vdots \end{pmatrix} \quad \text{where } C = \begin{pmatrix} p_{i_x} & 0 & p_{i_z} & -p_{i_y} & 1 & 0 & 0 \\ p_{i_y} & -p_{i_z} & 0 & p_{i_x} & 0 & 1 & 0 \\ p_{i_z} & p_{i_y} & -p_{i_x} & 0 & 0 & 0 & 1 \\ p_{i1_x} & 0 & p_{i1_z} & -p_{i1_y} & 1 & 0 & 0 \\ p_{i1_y} & -p_{i1_z} & 0 & p_{i1_x} & 0 & 1 & 0 \\ p_{i1_z} & p_{i1_y} & -p_{i1_x} & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

  - Linear expressions of the deformed points p_i'
    - C is a $3|N_i|+3$ by 7 matrix

Distortion Term (3D)

- Constructing transformed Laplacian:

$$T_i\,\delta_i = D \begin{pmatrix} s \\ h_1 \\ h_2 \\ h_3 \\ t_x \\ t_y \\ t_z \end{pmatrix} = D\left(C^T C\right)^{-1} C^T \begin{pmatrix} p_{i_x}{}' \\ p_{i_y}{}' \\ p_{i_z}{}' \\ p_{i1_x}{}' \\ p_{i1_y}{}' \\ p_{i1_z}{}' \\ \vdots \end{pmatrix}$$

$$\text{where } D = \begin{pmatrix} \delta_{i_x} & 0 & \delta_{i_z} & -\delta_{i_y} & 1 & 0 & 0 \\ \delta_{i_y} & -\delta_{i_z} & 0 & \delta_{i_x} & 0 & 1 & 0 \\ \delta_{i_z} & \delta_{i_y} & -\delta_{i_x} & 0 & 0 & 0 & 1 \end{pmatrix}$$