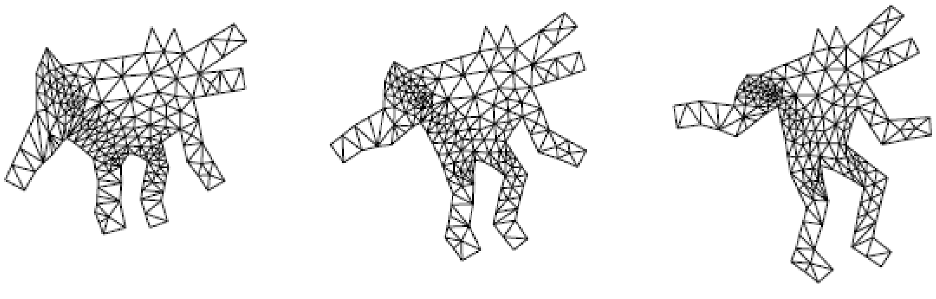
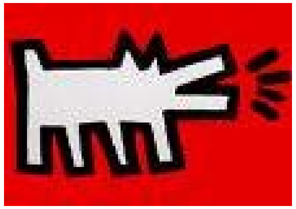
Rigid Morphing

14 May 2015

As-rigid-as-possible shape interpolation

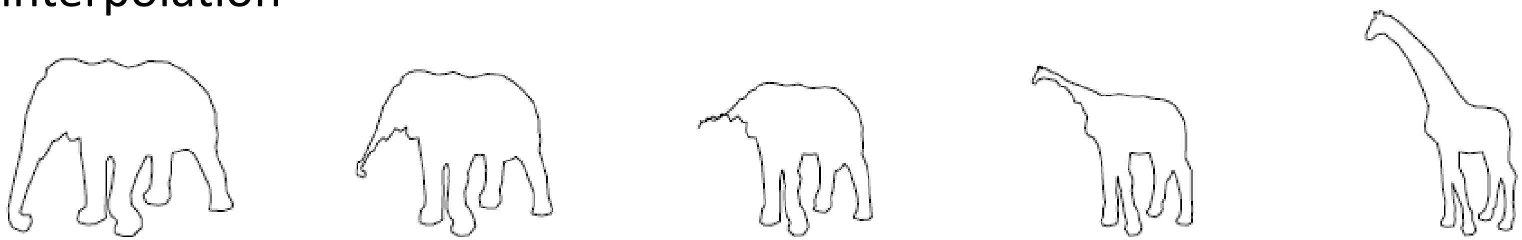
- *Alexa et al.*

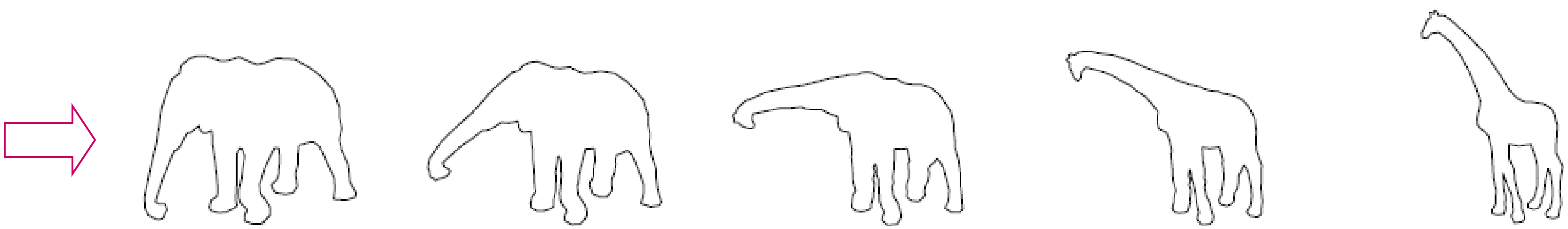


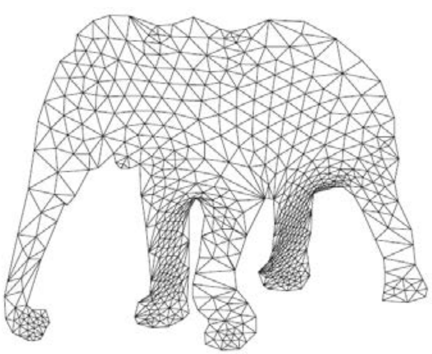


As Rigid As Possible

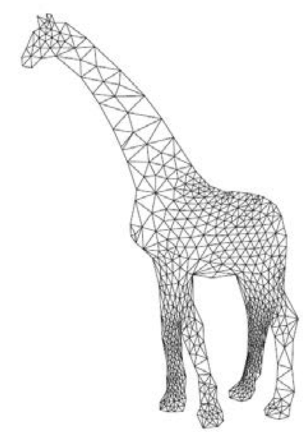
linear interpolation





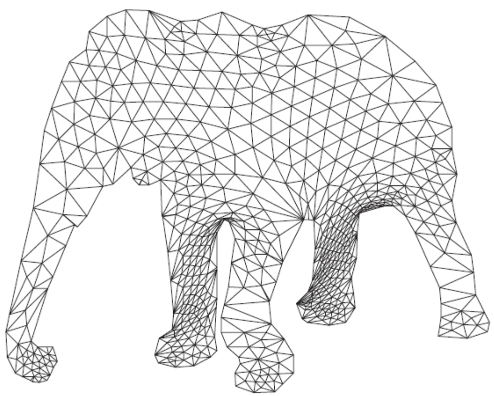


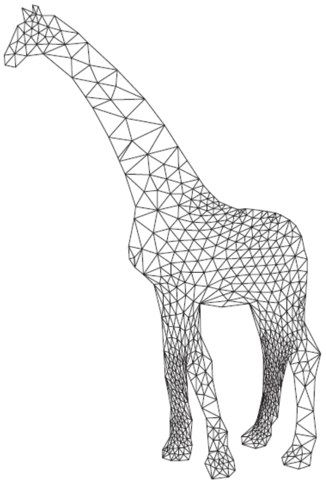
least-distorting morphing



Isomorphic Dissections of Shapes

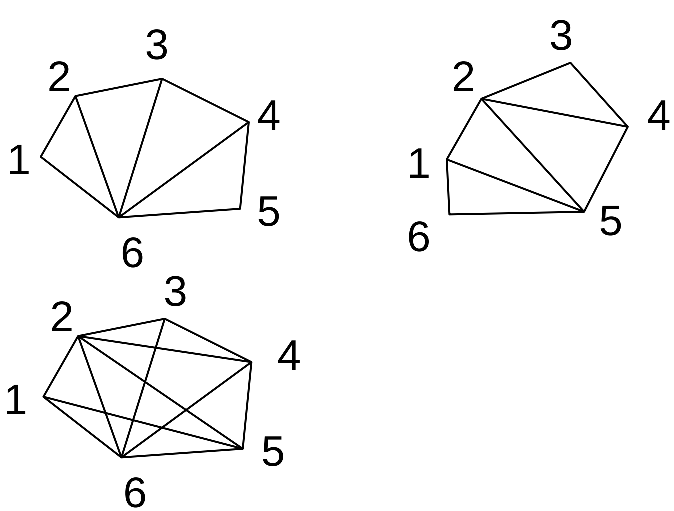
- Compatible triangulation
  - Constructing a common triangulation for two given polygons

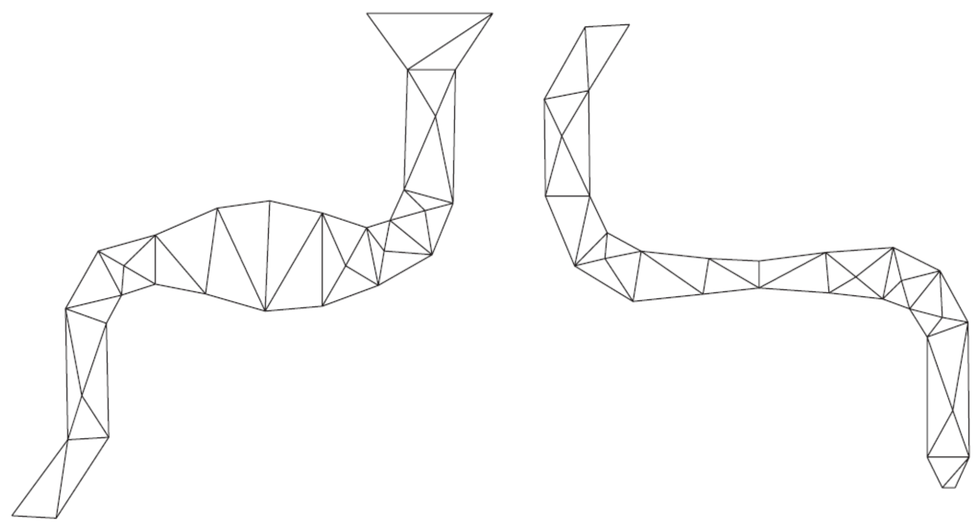




Basic Algorithm

- 1. Triangulate each polygon independently, using Delaunay triangulations
- 2. Map both polygons to a regular n-gon so that **corresponding boundary vertices coincide**
- 3. The compatible triangulation is established by **overlaying the two edge sets** in the convex n-gon. The resulting new interior vertices are then mapped back into the original polygons.

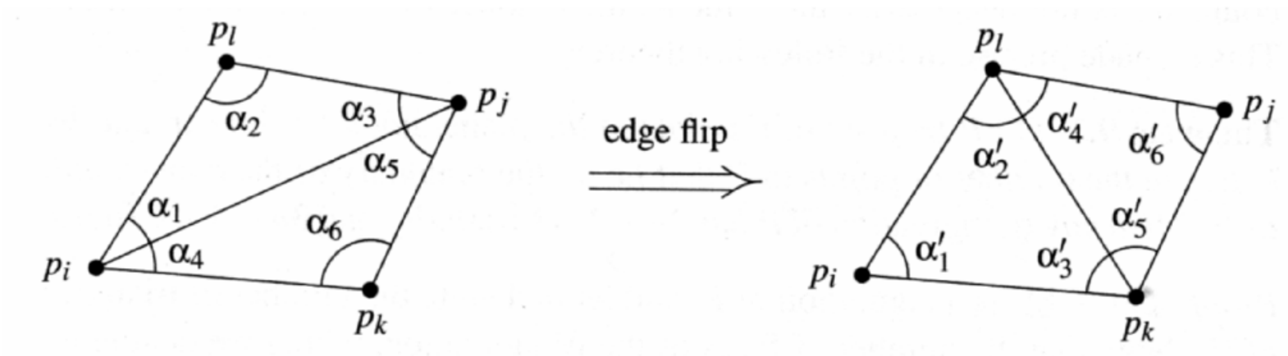


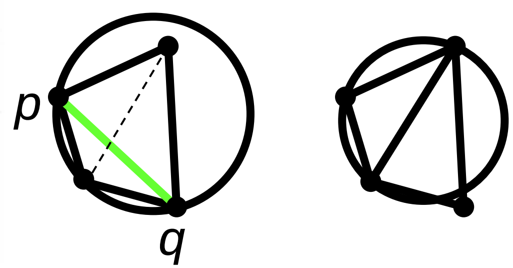


Mesh Refinement

- Smoothing, applying Delaunay criterion
  - **Moving interior vertices** to find vertex positions that **maximize the minimum angle** for a given triangulation

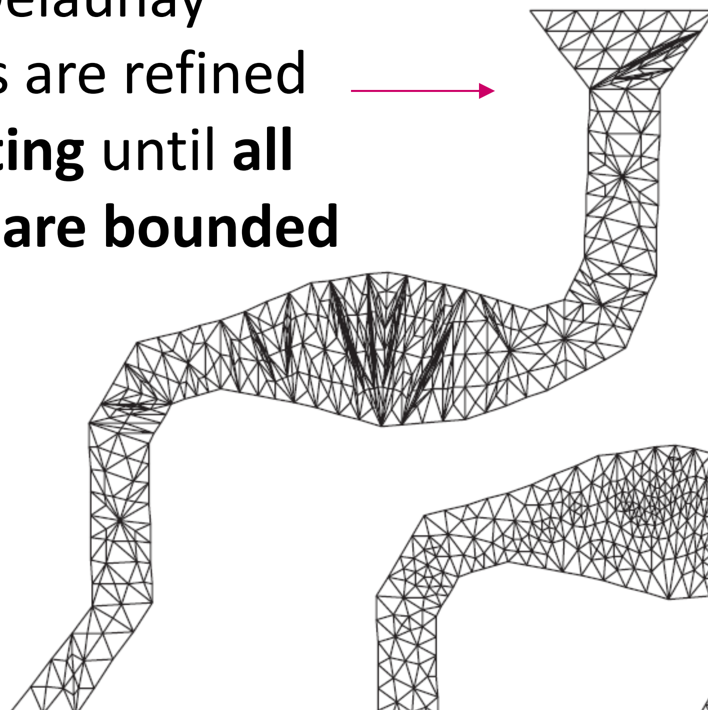
  - **Flipping interior edges** simultaneously in both triangulations. This procedure follows the edge flip criteria used in Delaunay triangulation.



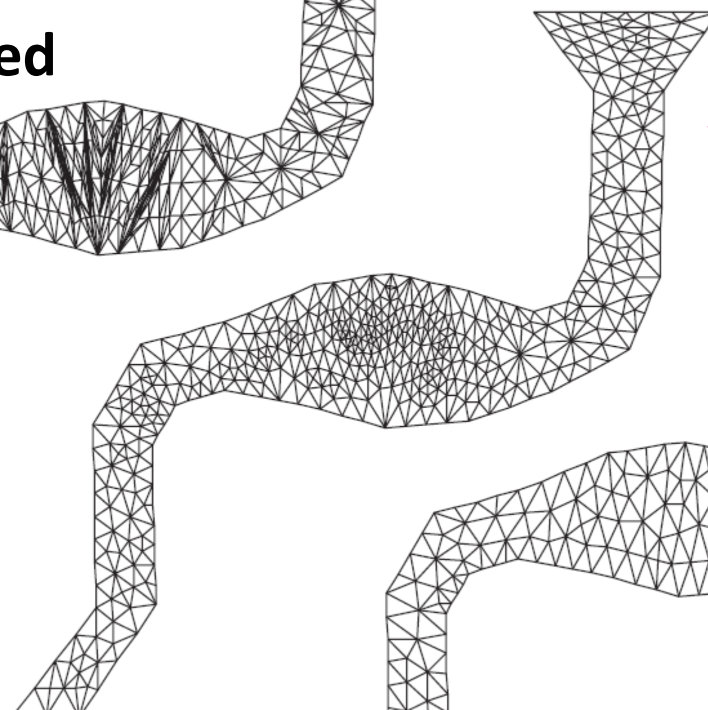


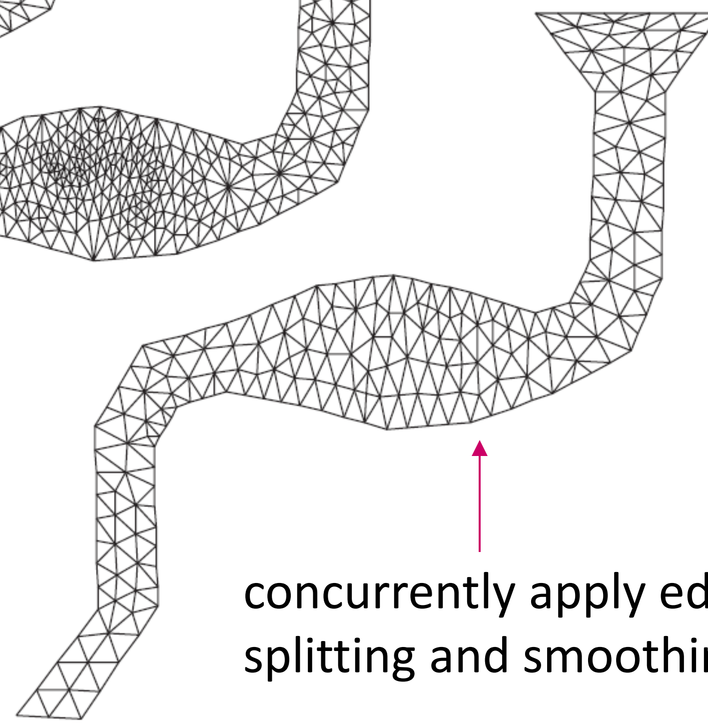
Edge Splitting

the merged Delaunay triangulations are refined by **edge splitting** until **all edge lengths are bounded**



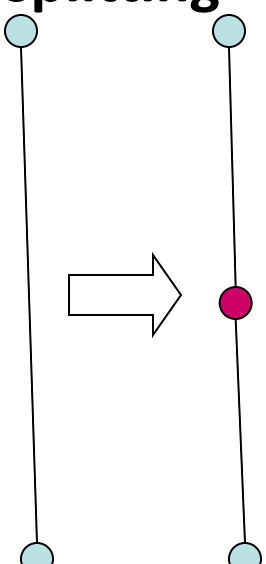
edge splitting and then smoothing



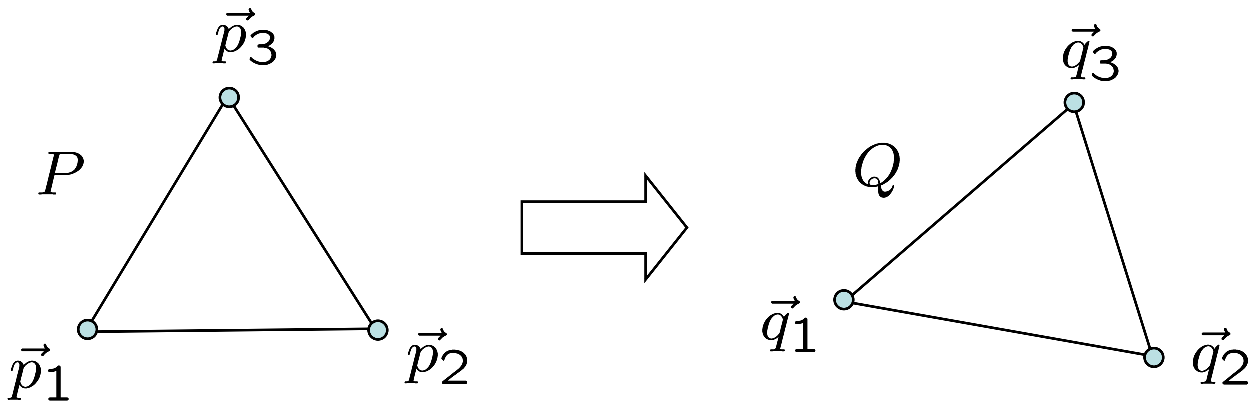


concurrently apply edge splitting and smoothing

edge splitting



Least-Distorting Triangle-to-Triangle Morphing



affine transformation

$$A\vec{p_i} + \vec{l} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \vec{p_i} + \begin{pmatrix} l_x \\ l_y \end{pmatrix} = \vec{q_i}, \quad i \in \{1, 2, 3\}$$

ignore translation

$$V(t) = (\vec{v_1}(t), \vec{v_2}(t), \vec{v_3}(t)) \qquad V(t) = A(t)P$$

How to Define $A(t)$ Reasonably?

- The transformation should be symmetric with respect to t
- The rotational angle and scale should change linearly
- The triangle should keep its orientation (not be reflected)
- The resulting vertices' paths should be **simple**

Singular Value Decomposition (SVD)

$$A\vec{p_i} + \vec{l} = \underbrace{\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}} \vec{p_i} + \begin{pmatrix} l_x \\ l_y \end{pmatrix} = \vec{q_i}, \quad i \in \{1, 2, 3\}$$

rotation matrix is orthogonal $R^{-1} = R^T$

SVD

$$A = R_\alpha D R_\beta = R_\alpha \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} R_\beta, \quad s_x, s_y > 0$$

SVD: $A = UDV^T$

```
[Ra, D, Rb] = svd(A);
Rb = Rb';
```

$$A^T A = (VDU^T)(UDV^T) = VDV^T$$

$$AA^T = (UDV^T)(VDU^T) = UDU^T$$

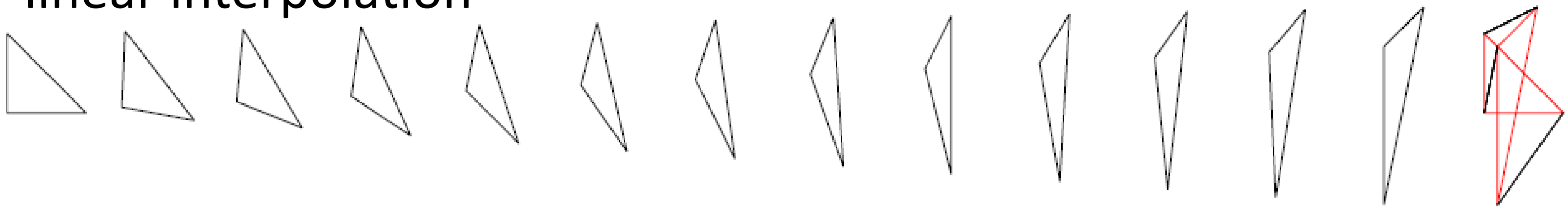
Rotation + Scale, Shear

$$
\begin{aligned}
A &= R_\alpha D R_\beta = R_\alpha (R_\beta R_\beta^T) D R_\beta = \\
& (R_\alpha R_\beta)(R_\beta^T D R_\beta) = R_\gamma S = R_\gamma \begin{pmatrix} s_x & s_h \\ s_h & s_y \end{pmatrix}
\end{aligned}
$$

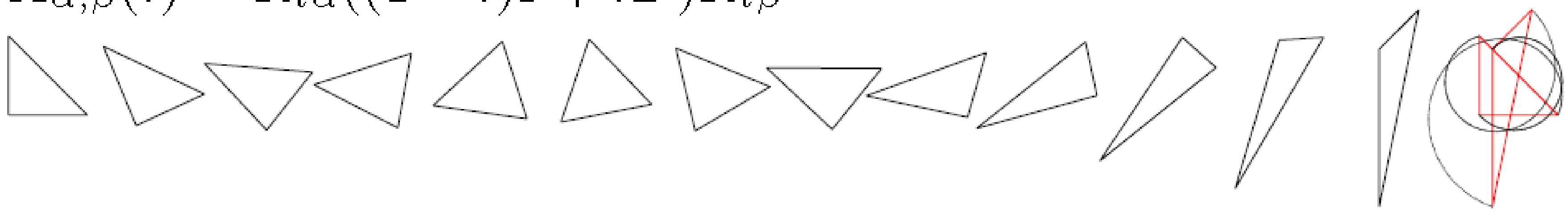
```
[Ra, D, Rb] = svd(A);
Rb = Rb';

Rc = Ra*Rb;
S = Rb'*D*Rb;
theta = atan2(-Rc(1,2), Rc(1,1));
```

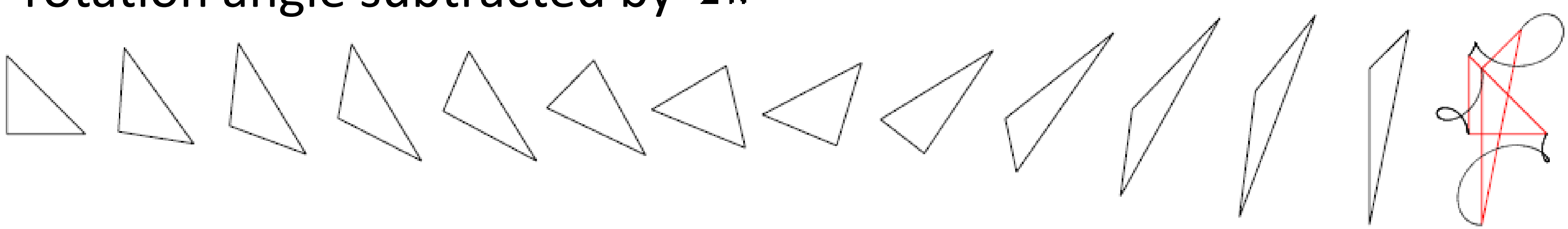
linear interpolation



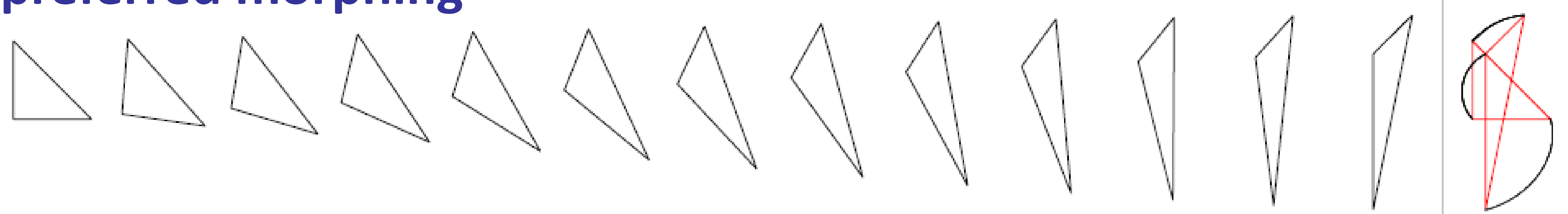
$$A_{\alpha,\beta}(t) = R_{t\alpha}((1-t)I + tD)R_{t\beta}$$



rotation angle subtracted by 2π



preferred morphing



$$A_{\gamma}(t) = R_{t\gamma}((1-t)I + tS) \qquad R_{t\gamma} = \begin{bmatrix} \cos(\theta t) & -\sin(\theta t) \\ \sin(\theta t) & \cos(\theta t) \end{bmatrix}$$

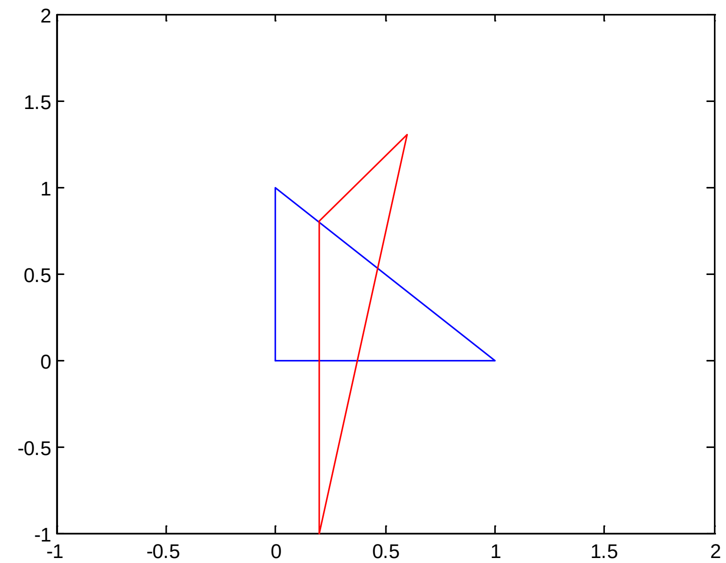
Recover the Affine Transformation

six unknowns

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & l_x \\ a_3 & a_4 & l_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

three vertices, six equations

$$\begin{bmatrix} p_{1x} & p_{1y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{1x} & p_{1y} & 1 \\ p_{2x} & p_{2y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{2x} & p_{2y} & 1 \\ p_{3x} & p_{3y} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{3x} & p_{3y} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ l_x \\ a_3 \\ a_4 \\ l_y \end{bmatrix} = \begin{bmatrix} q_{1x} \\ q_{1y} \\ q_{2x} \\ q_{2y} \\ q_{3x} \\ q_{3y} \end{bmatrix}$$




```
P = [0, 0; 1, 0; 0, 1];
Q = [0.2, 0.8; 0.2, -1; 0.6, 1.3];
figure; hold on
plot([P(:,1); P(1,1)], ...
     [P(:,2); P(1,2)], 'b');
plot([Q(:,1); Q(1,1)], ...
     [Q(:,2); Q(1,2)], 'r');
axis([-1 2 -1 2])
```

```
M = zeros(6, 6);
for k = 1 : 3
    M(2*k-1, 1:2) = P(k, :);
    M(2*k-1, 3) = 1;
    M(2*k, 4:5) = P(k, :);
    M(2*k, 6) = 1;
end
b = reshape(Q', 6, 1);
a = M\b;
A = reshape(a, 3, 2)';
L = A(1:2, 3)'; % translation
A = A(1:2,1:2); % rotation+scaling+shearing
```

```
[Ra, D, Rb] = svd(A);
Rb = Rb';
Rc = Ra*Rb;
S = Rb'*D*Rb;
theta = atan2(-Rc(1,2), Rc(1,1));
figure;
for t = 0:0.1:1
    thetat = theta*t;
    Rct = [cos(thetat) -sin(thetat); ...
           sin(thetat) cos(thetat)];
    At = Rct*((1-t)*eye(2) + t*S);
    V = At*P';
    V = V';
    V = V + t*repmat(L, 3, 1);
    plot([V(:,1); V(1,1)], [V(:,2); V(1,2)], 'b')
    axis([-1 2 -1 2])
    pause(0.2)
end
```

Closed-Form Vertex Paths for a Triangulation

triangulation

$\mathcal{T} = \{T_{\{i,j,k\}}\}$

triangles

$$P_{\{i,j,k\}} = (\vec{p_i}, \vec{p_j}, \vec{p_k})$$

$$Q_{\{i,j,k\}} = (\vec{q_i}, \vec{q_j}, \vec{q_k})$$

$$A_{\{i,j,k\}}(t) \Rightarrow V(t) = (\vec{v_1}(t), \ldots, \vec{v_n}(t)), t \in [0,1]$$

$$V(0) = (\vec{p_1}, \ldots, \vec{p_n}) \quad V(1) = (\vec{q_1}, \ldots, \vec{q_n})$$

desired affine transformation

$$B_{\{i,j,k\}}(t)\vec{p_f} + \vec{l} = \vec{v_f}(t), \quad f \in \{i,j,k\}$$

minimize $E_{V(t)} = \sum_{\{i,j,k\} \in \mathcal{T}} \left\| A_{\{i,j,k\}}(t) - B_{\{i,j,k\}}(t) \right\|^2$

actual

$\|\cdot\|$ is the Frobenius norm

Matrix Form

minimize $E_{V(t)} = \sum_{\{i,j,k\}\in\mathcal{T}} \left\| A_{\{i,j,k\}}(t) - B_{\{i,j,k\}}(t) \right\|^2$

actual

$\|\cdot\|$ is the Frobenius norm

$$u^T = (1, v_{2_x}(t), v_{2_y}(t), \ldots, v_{n_x}(t), v_{n_y}(t))$$

$$E_{V(t)} = u^T \begin{pmatrix} c & G^T \\ G & H \end{pmatrix} u$$

$G \in \mathbb{R}^{2n\times 1}$

$H \in \mathbb{R}^{2n\times 2n}$

$$H \begin{pmatrix} v_{2_x}(t) \\ v_{2_y}(t) \\ \vdots \end{pmatrix} = -G$$

can be solved by "\"

$$V(t) = -H^{-1}G(t)$$

Good Properties

- For a given t, the solution is unique

- The solution requires only **one matrix inversion** for a specific source and target shape. Every intermediate shape is found by multiplying the inverted matrix by a vector $G(t)$

- The vertex path is infinitely smooth, starts exactly in the source shape, and ends exactly in the target shape

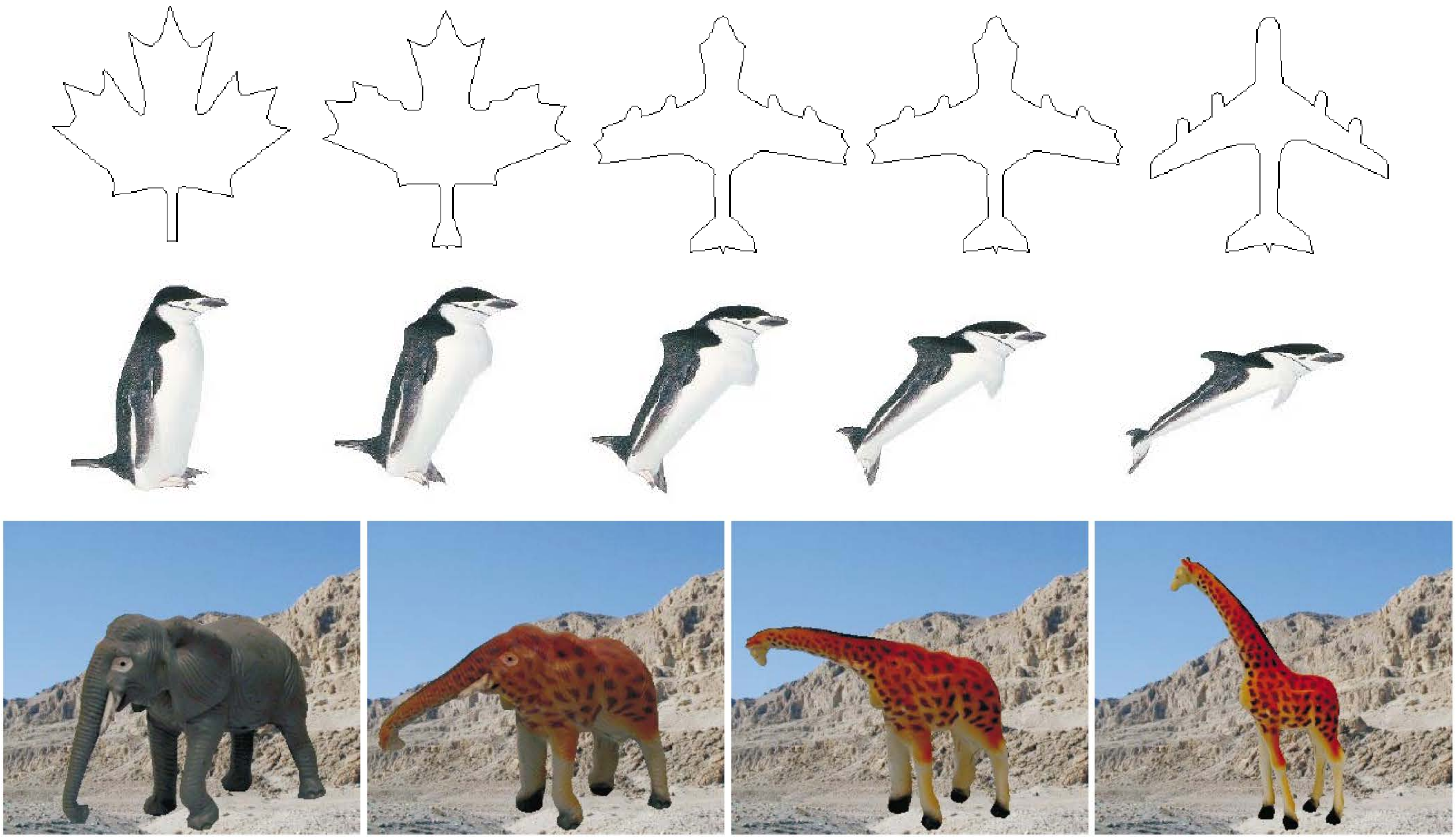
Symmetric Solutions

- Consider both directions

$$E_{V(t)} = (1-t)E^{\rightarrow}_{V(t)} + tE^{\leftarrow}_{V(t)}$$

$$\begin{aligned} E^{\rightarrow}_{V(t)} &= \sum_{f \in \mathrm{Tri}} \left\| A^{\rightarrow}_{f}(t) - B^{\rightarrow}_{f}(t) \right\|^2 \\ E^{\leftarrow}_{V(t)} &= \sum_{f \in \mathrm{Tri}} \left\| A^{\leftarrow}_{f}(1-t) - B^{\leftarrow}_{f}(1-t) \right\|^2 \end{aligned}$$

Results



Feature-Aware Warping

[Gal, Sorkine, and Cohen-Or]

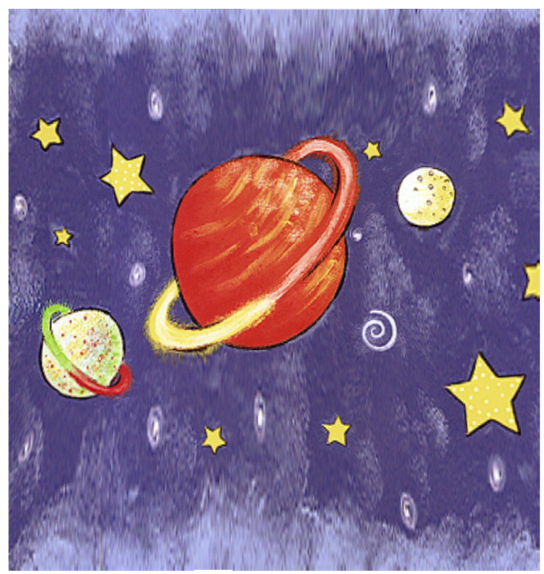
input image

standard swirl

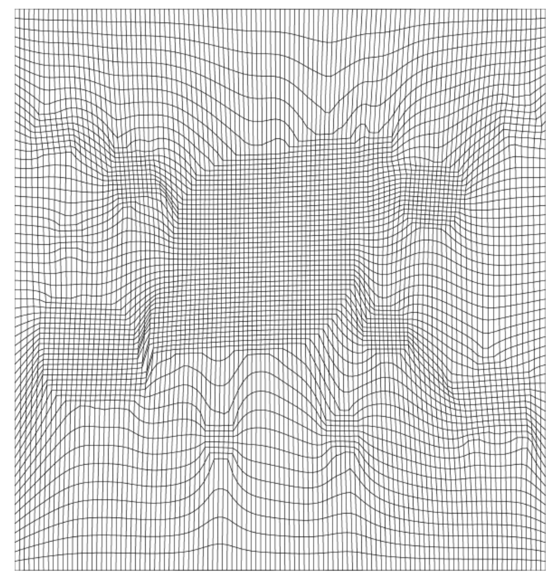
feature-aware swirl (rigid)

feature-aware swirl (similarity)

input image and its feature mask



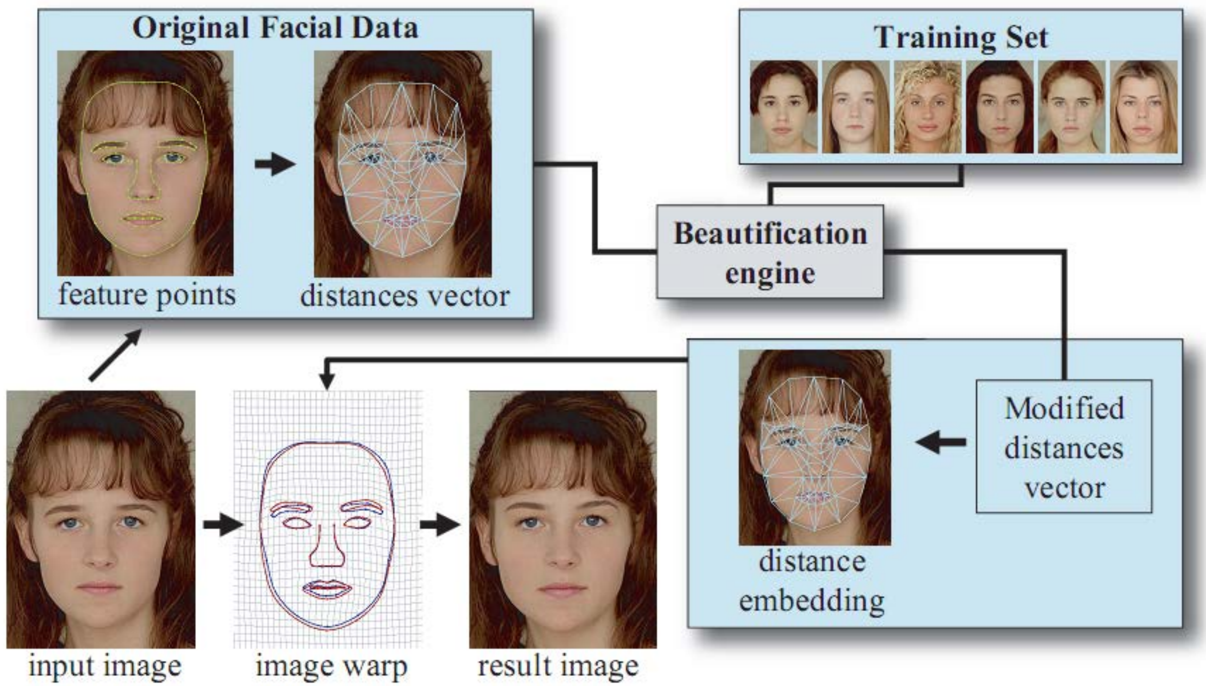
vertical stretch $\times$ 2



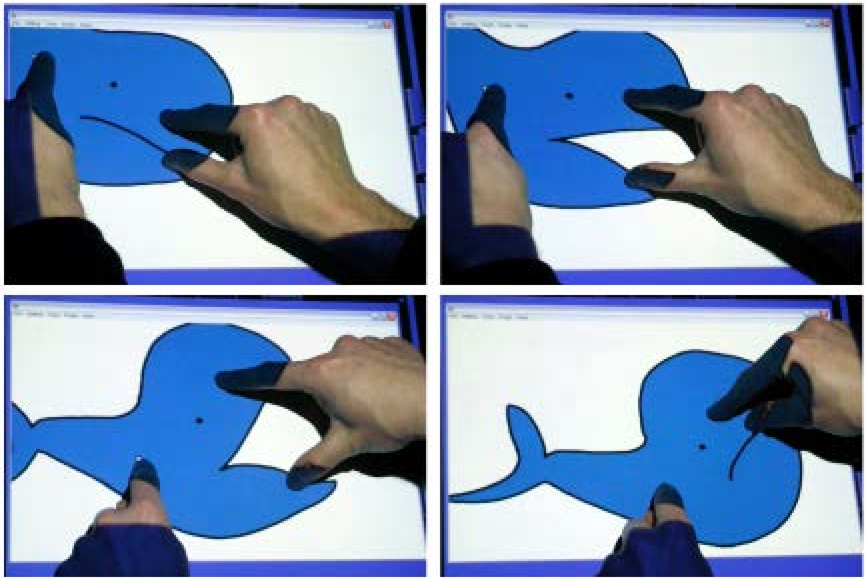
underlying grid

Data-Driven Enhancement of Facial Attractiveness

- Leyvand et al., SIGGRAPH 2008
- http://www.youtube.com/watch?v=lVbrUuwK-8g



As-Rigid-as-Possible Shape Manipulation



- Demo Java Applets
  - http://www-ui.is.s.u-tokyo.ac.jp/~takeo/research/rigid/index.html

Regenerative Morphing

- Shechtman, Rav-Acha, Irani, and Seitz
  - CVPR 2010
- Bidirectional similarity
- PatchMatch

Being John Malkovich

- Kemelmacher-Shlizerman, Sankar, Shechtman, and Seitz
  - ECCV 2010
- Puppeteering a celebrity
- Video or photo collection of celebrity
- Image based
- Local Binary Pattern (LBP)