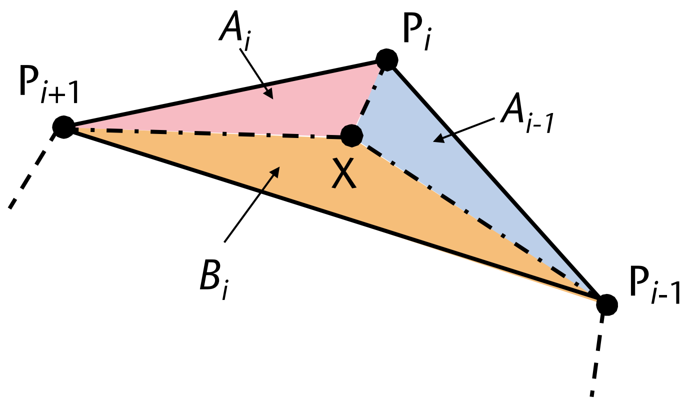
- Reminder:
  - In a triangle, A_i/C_i, B_i/C_i, $A_{i\text{-}1}/C_i$ are the barycentric coordinates; i.e.:

$$\frac{A_i}{C_i}(P_{i-1} - X) + \frac{B_i}{C_i}(P_i - X) + \frac{A_{i-1}}{C_i}(P_{i+1} - X) = 0$$



- Therefore:

$$A_i(P_{i-1} - X) + B_i(P_i - X) + A_{i-1}(P_{i+1} - X) = 0$$

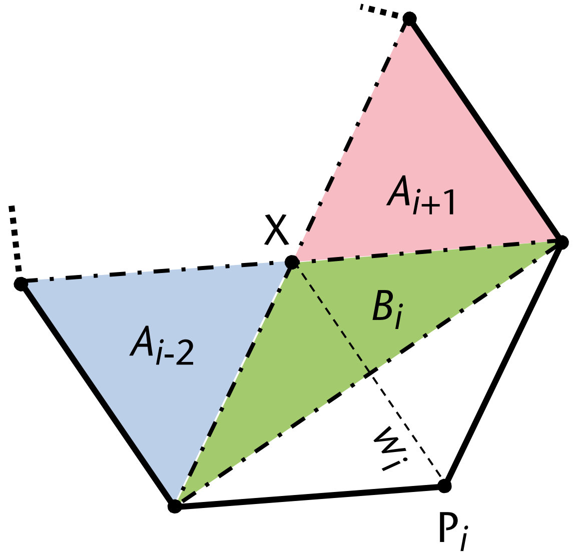
Homogenous barycentric coordinates

- Consider the series of triangles $\Delta\ P_{i-1}\ P_i\ P_{i+1}$
- Approach: compute the weighted average of the (homogeneous) barycentric coordinates w.r.t. each of these triangles:

$$w_i := w_i(X) = \sigma_{i-1} A_{i-2} + \sigma_i B_i + \sigma_{i+1} A_{i+1}$$

where $\sigma_i := \sigma(X)$ can be any function (for the time being)

- Every vertex is involved in 4 or 5 barycentric coordinates, respectively

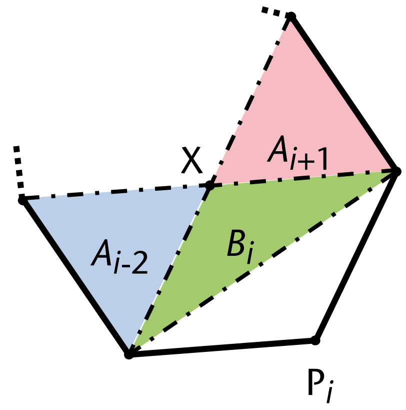


- Proposition 1:

  These

  $$w_i = \sigma_{i-1} A_{i-2} + \sigma_i B_i + \sigma_{i+1} A_{i+1}$$

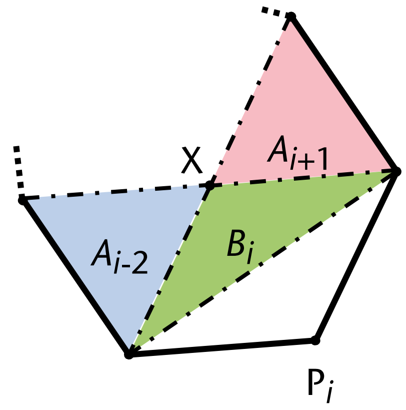
  fulfill condition 1 from the definition of barycentric coordinates.



- Proof:

$$\sum_{i=1}^{n} w_i (P_i - X) = \sum_{i=1}^{n} \sigma_i \big(A_i (P_{i-1} - X) + B_i (P_i - X) + A_{i-1} (P_{i+1} - X) \big) = 0$$

- Proposition 2:
  If the polygon is convex and $\forall i : \sigma_i(X) > 0$
  then $\sum w_i(X) > 0$
  for all values of X in the interior of the polygon.



- Proof:

$$\sum_{i=1}^{n} w_i(X) = \sum_{i=1}^{n} \sigma_i(X) \cdot C_i > 0 \quad , \quad \text{da} \quad \forall i : \; C_i > 0$$

Insert definition of the w_i, change summation indicies appropriately, remember indices are mod n

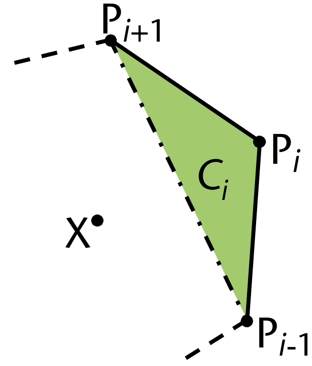
- Note: $\sigma_i > 0$ alone does not guarantee that $\sum w_i(X) > 0$!
  - The convexity of the polygon is crucial...

- Note: with $\sum w_i > 0$, the normalization of the w_i's to get the λ_i 's always works
- Reminder: $w_i > 0$ is a requirement from condition 2 of the definition

- Goal: look for appropriate σ_i, such that $w_i > 0$ and $\sigma_i > 0$

Some Candidates

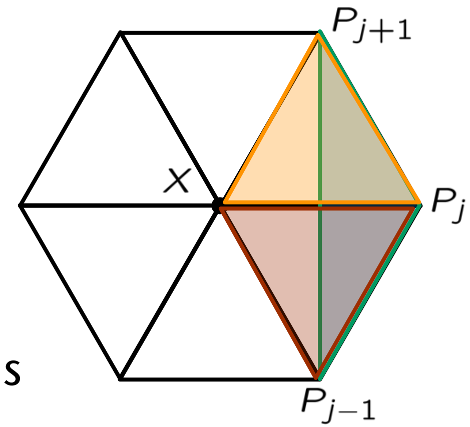
- Naive approach: choose $\sigma_i = \frac{1}{C_i}$
  - Thus $\sum w_i(X) \equiv n$
  - Unfortunately, $w_i(X) > 0$ is not guaranteed
  - Result: the interpolation property doesn't hold ☹



- Wachspress coordinates: choose $\sigma_i(X) = \dfrac{1}{A_{i-1}A_i}$
  - Thus

$$w_i := \frac{\mathcal{F}\left(\Delta P_{i-1}P_iP_{i+1}\right)}{\mathcal{F}\left(\Delta X P_{i-1}P_i\right)\cdot\mathcal{F}\left(\Delta X P_iP_{i+1}\right)}$$

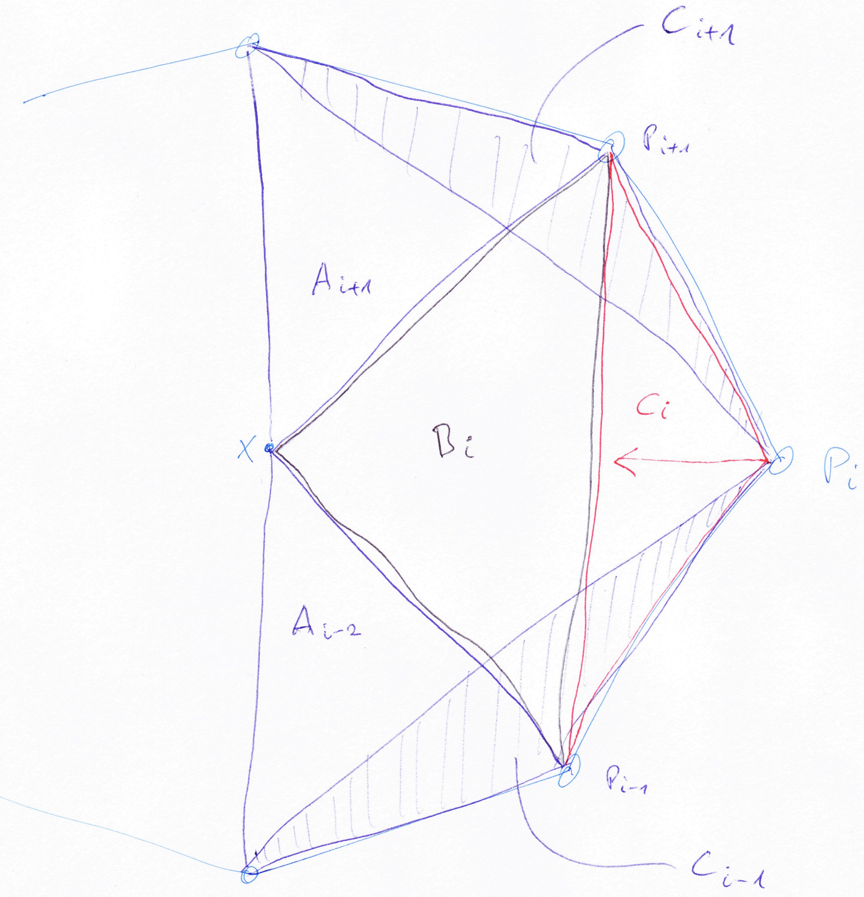
  - Disadvantage: they behave badly in a non-convex polygon, since $\sum w_i(X) = 0$ is possible, which means that the λ_i's have a pole there



- Explanation why $w_i < 0$ is possible with the naïve choice

Mit $\sigma_i = \frac{1}{c_i}$ wird

$$w_i = \frac{A_{i-2}}{c_{i-1}} + \underbrace{\frac{B_i}{c_i}}_{\to -\infty\text{, wenn } P_i \to \overline{P_{i-1}P_{i+1}} \text{ geht}} + \frac{A_{i+1}}{c_{i+1}}$$



The Best Candidate (up until this point)

- Mean value coordinates (MVCs):
  - Choose

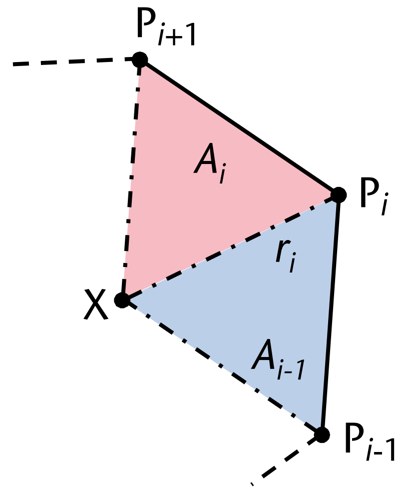
$$\sigma_i = \frac{r_i}{A_{i-1}A_i}$$

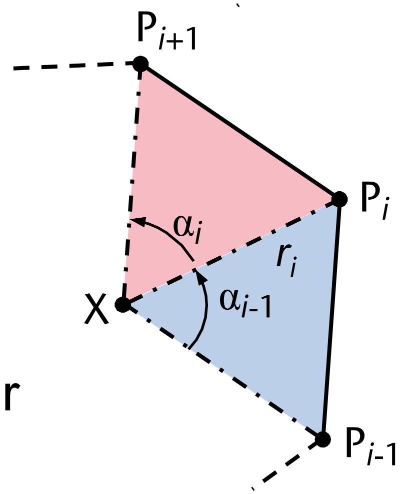
  - Thus

$$w_i(X) = \frac{r_{i-1}A_i + r_i B_i + r_{i+1}A_{i-1}}{A_{i-1}A_i}$$

  - With some trigonometric substitutions:

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{r_i/2}$$





- Proposition: the MVCs are barycentric coordinates for all X in the interior of the polygon
- Obvious, because:

  if X is in the interior $\rightarrow$ all $\sigma_i > 0$ and all $w_i > 0$

- A demonstration that the equation for w_i is correct:

$$
\begin{aligned}
w_i &= \sigma_{i-1} A_{i-2} + \sigma_i B_i + \sigma_{i+1} A_{i+1} \\
&= \frac{r_{i-1}}{A_{i-2}A_{i-1}} A_{i-2} + \frac{r_i}{A_{i-1}A_i} B_i + \frac{r_{i+1}}{A_i A_{i+1}} A_{i+1} \\
&= \frac{r_{i-1}}{A_{i-1}} + \frac{r_i}{A_{i-1}A_i} B_i + \frac{r_{i+1}}{A_i} = \cdots
\end{aligned}
$$

- Then: for A_i and B_i, use the sin-formula for the surface area and use trigonometric identities

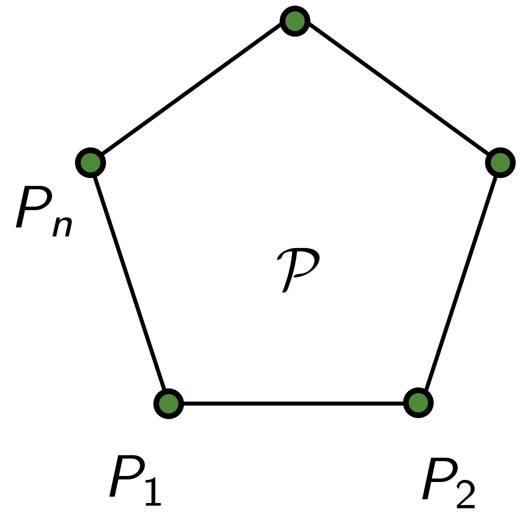
Extension to Non-Convex Polygons

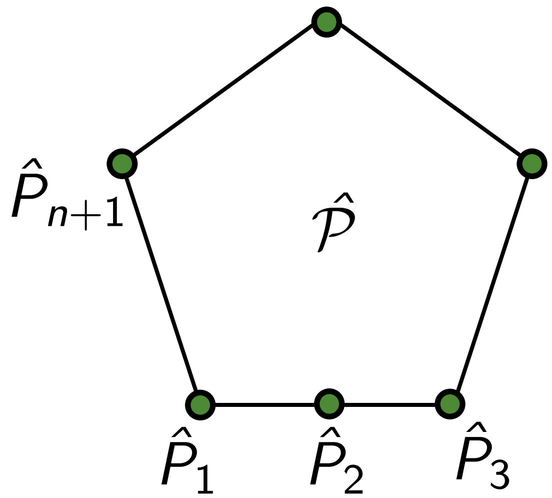
- Lemma (w/o proof):
  Let $\mathcal{P}$ be a given convex polygon.
  Label the MVCs of a point X
  w.r.t. $\mathcal{P}$ with w_i , i=1…n.
  Now refine $\mathcal{P}$ with the insertion of a point.
  Denote this refined polygon by $\hat{\mathcal{P}}$.
  Label the MVCs of X
  w.r.t. $\hat{\mathcal{P}}$ with $\hat{w}_i$, i=1…n+1.

  Then

$$\sum_{i=1}^{n+1} \hat{w}_i = \sum_{i=1}^{n} w_i$$

- Consequence: the $\lambda' s$ are also well-defined for $\hat{\mathcal{P}}$





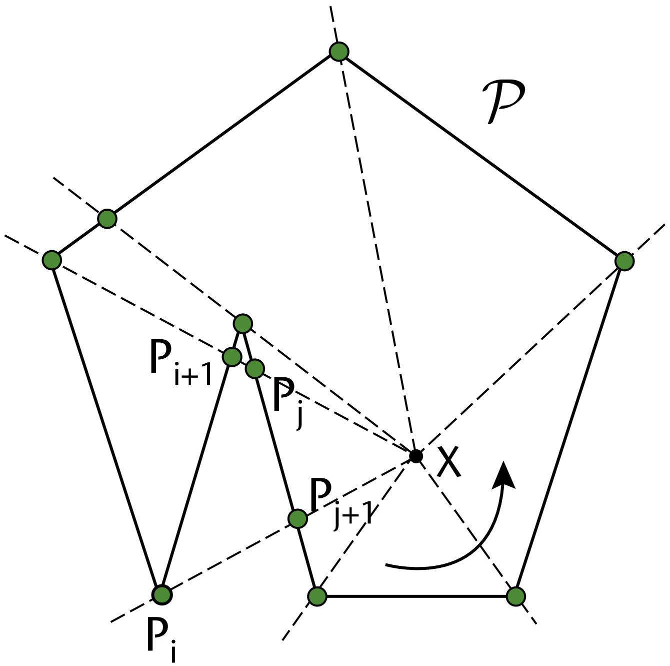
- Theorem:

  Let $\mathcal{P}$ be any simple polygon.

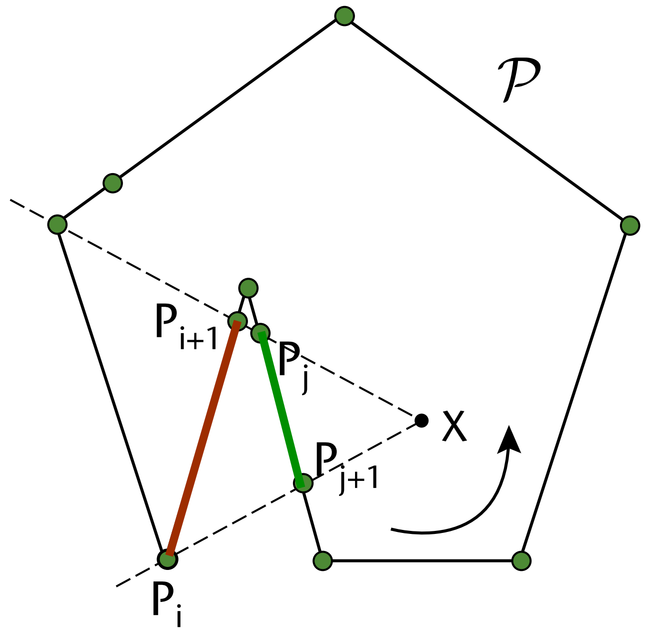
  For all X not located on the edge of the polygon $\mathcal{P}$:

$$\sum w_i(X) \neq 0$$

- Proof:
  - Assumption: X is in the interior of $\mathcal{P}$
  - Draw rays from X through the corners of $\mathcal{P}$ $\rightarrow$ refinement of $\mathcal{P}$
  - Name the refinement $\mathcal{P}$ again and its corners $P_1,\ldots,P_n$.



- Classify edges into
  "entry edge" (red) or "exit edge" (green)
  - Can be done easily either by checking the orientation of the edge w.r.t. *X*, or by following a ray from *X* outward



- Observation:
  For every entry-edge there is an exit-edge closer to *X*

- For every edge P_iP_{i+1}, define the following value

$$k_i = \left(\frac{1}{r_i} + \frac{1}{r_{i+1}}\right) \tan \frac{\alpha_i}{2}$$

where the *signs* of the angles α_i are determined by the orientation of the respective edges

- One sees immediately that: $$\sum k_i = \frac{1}{2} \sum w_i$$

(The summands are combined only a little differently, and the coefficient ½ is with the r_i)

- Thus, for an edge P_iP_{i+1}:

  if exit-edge $\rightarrow k_i > 0$

  if entry-edge $\rightarrow k_i < 0$

- Let P_iP_{i+1} be an entry-edge
- Then a corresponding exit-edge P_jP_{j+1} must also exist, and it is closer to X
- The following holds for their angles: $\alpha_i = -\alpha_j$
- The following applies for their distances:

$$r_j \leq r_{i+1} \;\wedge\; r_{j+1} < r_i \quad \text{oder} \quad r_j < r_{i+1} \;\wedge\; r_{j+1} \leq r_i$$

- With that, we have

$$k_j = \left(\frac{1}{r_j} + \frac{1}{r_{j+1}}\right) \tan\frac{\alpha_j}{2} \quad > \quad \left(\frac{1}{r_i} + \frac{1}{r_{i+1}}\right) \tan\frac{-\alpha_i}{2} = -k_i$$

- In other words: for every k_i of an entry-edge, there is a k_j of an exit-edge so that $k_i + k_j > 0$
- Thus $\sum k_i > 0$

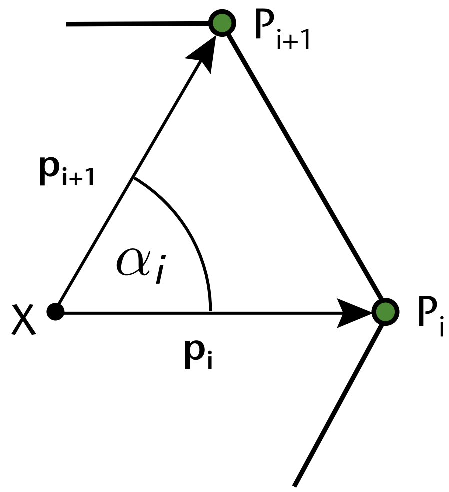
  and with that $\sum w_i > 0$

  for all X in the interior of $\mathcal{P}$

- Furthermore, we can show that for non-convex polygons the *mean value coordinates* have the following properties:
  - λ_i are well-defined for *X* on the edge of the polygon
  - $\lambda_i(P_j) = \delta_{ij}$
  - $\lambda_i \in \mathcal{C}^\infty$ with the exception of those at P_j; there they are only $\mathcal{C}^0$

Implementation

- Practical calculation of $\tan\left(\frac{\alpha_i}{2}\right)$:



$$\tan\frac{\alpha_i}{2} = \frac{\sin\alpha_i}{1+\cos\alpha_i}$$

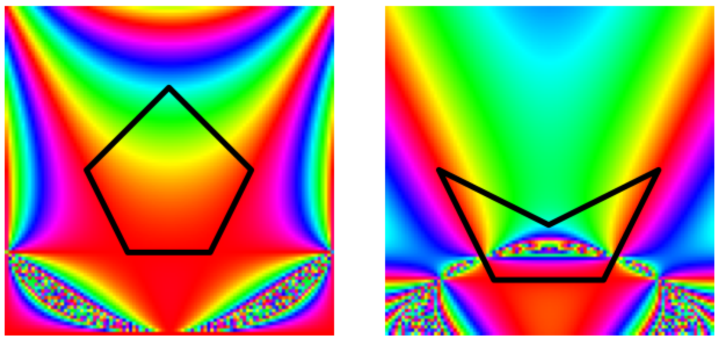
$$\cos\alpha_i = \frac{\mathbf{p}_i \cdot \mathbf{p}_{i+1}}{|\mathbf{p}_i|\cdot|\mathbf{p}_{i+1}|} \qquad \sin\alpha_i = \frac{|\mathbf{p}_i \times \mathbf{p}_{i+1}|}{|\mathbf{p}_i|\cdot|\mathbf{p}_{i+1}|}$$

Thus: $$\tan\frac{\alpha_i}{2} = \frac{|\mathbf{p}_i \times \mathbf{p}_{i+1}|}{|\mathbf{p}_i|\cdot|\mathbf{p}_{i+1}| + \mathbf{p}_i\cdot\mathbf{p}_{i+1}}$$

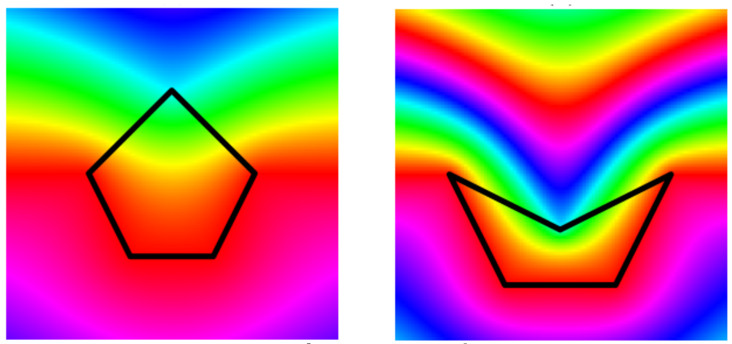
- If $|\mathbf{p}_i \times \mathbf{p}_{i+1}| = 0$, then X is located on an edge
  $\rightarrow$ special treatment:
  1. $X = P_i$ or $X = P_{i+1}$
  2. Otherwise: use linear interpolation between P_i und P_{i+1}

Application: Interpolation of Colors

- Given:
  - A simple polygon (not necessarily convex)
  - A color at every corner
- Task: color the interior of the polygon with "pretty" color gradients (a common task in drawing software, for example)
- Solution:
  - Calculate barycentric coordintates for every pixel in the interior of the given polygon
  - Interpolate the colors of the vertices using these barycentric coords



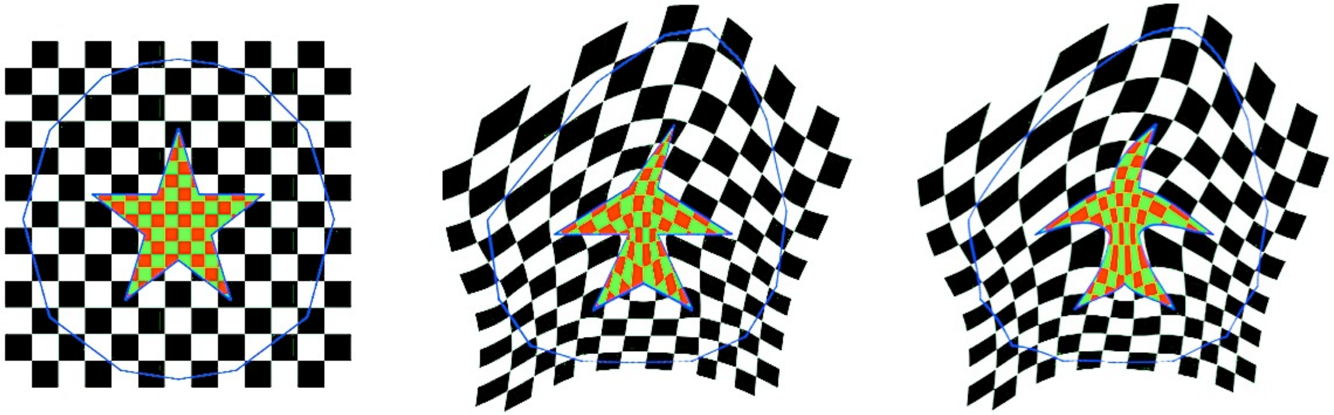
Wachspress



Mean Value Coordinates

Application: Image Warping

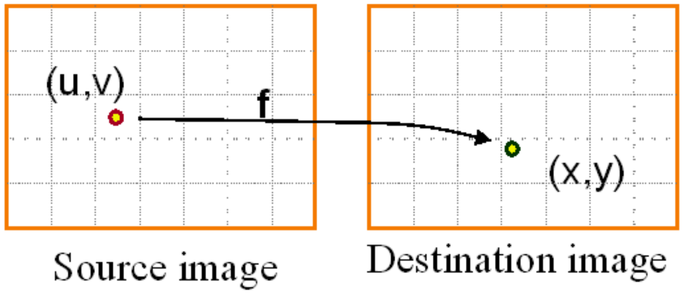
- Task: warp the given image by displacing a few "control polygon"
- Examples:



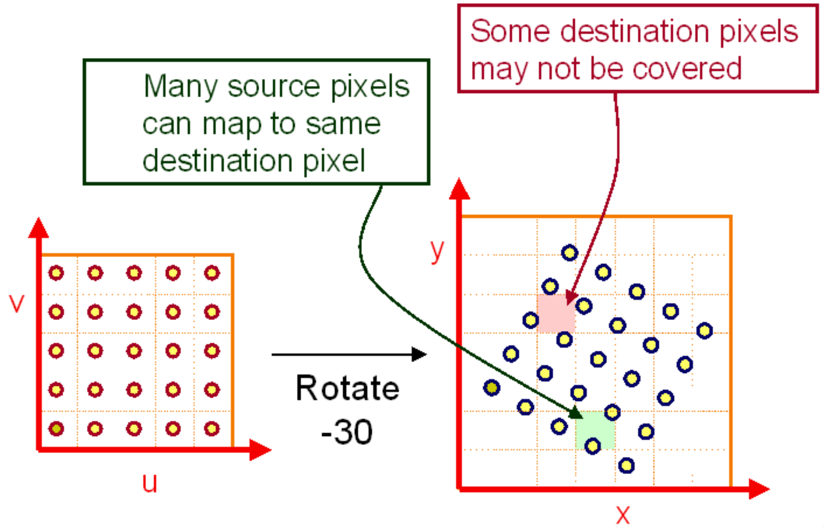
Algorithm

- First idea: *"forward mapping"*

```
for u = 0 .. umax:
  for v = 0 .. vmax
    x, y = f(u,v)
    dst(x,y) ← src(u,v)
```

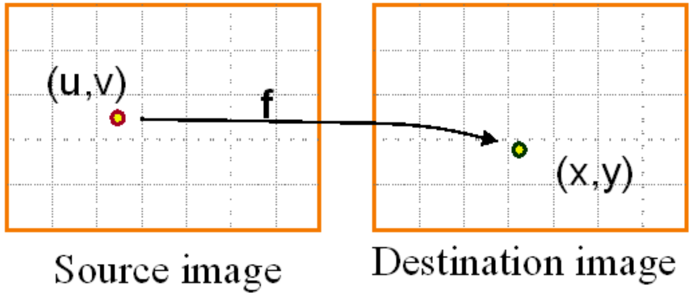


- Construction of *f* :
  - Determine barycentric coordinates of a point *X* w.r.t. the control polygon in the source image
  - Interpolate the *positions* of the vertices of the control polygon in the destination image ⟶ *X'*
- Problems:

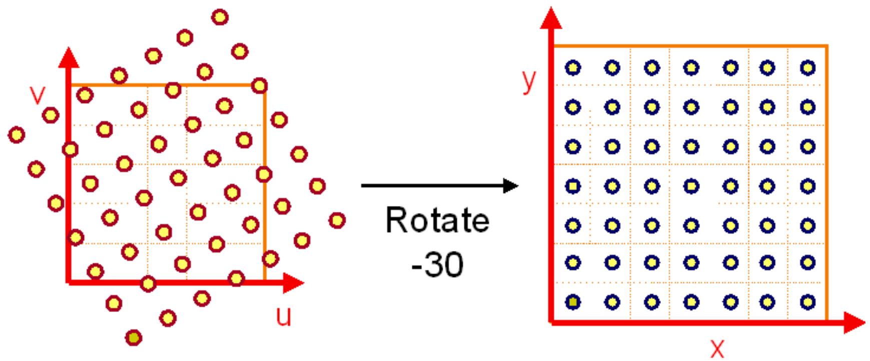


- Better idea: *"reverse mapping"*

```
for x = 0 .. xmax:
  for y = 0 .. ymax
    u, v = f^-1(x,y)
    dst(x,y) ← src(u,v)
```

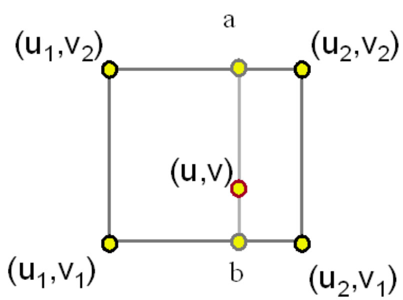


- Use barycentric interpolation again to construct f^{-1}
  - Just swap the roles
- Small problem:
  - (*u*,*v*) are not pixel coords.; rather they are located "between" pixels
  - One has to do *"resampling"* or interpolation in the source image



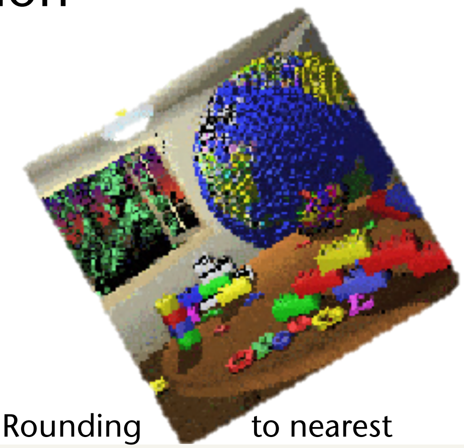
- Simplest solution: rounding
  - Produces big artifacts ("aliasing"; more on this later)
- The second-simplest solution: bi-linear interpolation

```
for x = 0 .. xmax:
  for y = 0 .. ymax
    u, v = f⁻¹(x,y)
    a = lin.interp. between src(u1,v2) and src(u2,v2)
    b = lin.interp. between src(u1,v1) and src(u2,v1)
    c = lin.interp. between a and b
    dest(x,y) ← c
```



- Better yet: Gaussian convolution
- Examples:

Original

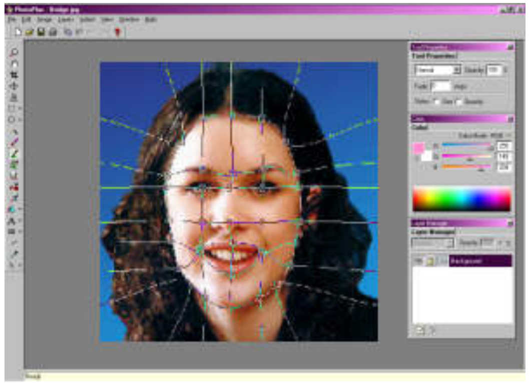


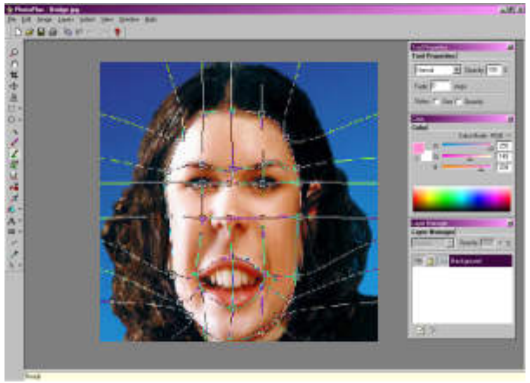
Rounding to nearest

Bilinear

- Additional examples:

- Today fully-integrated in software:





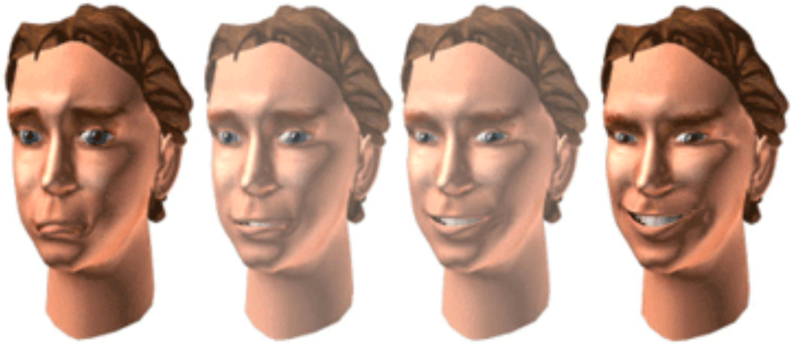
Application: Morphing

- Given: two triangle meshes M_1 und M_2 with...
  - Exactly the same number of vertices und triangles; and
  - A correspondence $\Phi : V_1 \longrightarrow V_2$ so that

$$P, Q, R \text{ ist ein Dreieck in } M_1 \Leftrightarrow$$

$$\Phi(P), \Phi(Q), \Phi(R) \text{ ist ein Dreieck in } M_2$$

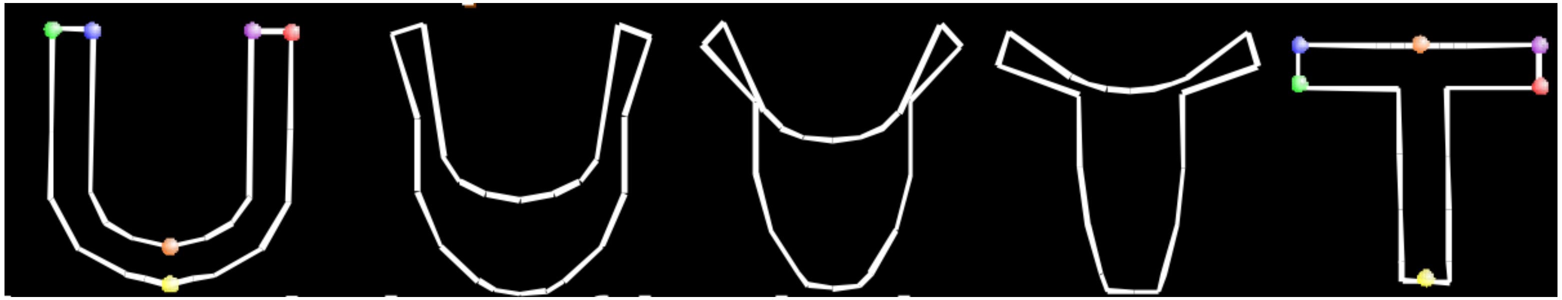
- Task: a uniform "deformaton" of mesh M_1 in M_2
  - Because of the correspondence, it is sufficient to manipulate the coordinates of the vertices from V_1 uniformly (for example, across 1000 time steps), so that in the end V_2 is generated
- Terminology: M_1 and M_2 are also called *"morph targets,"* or *"source"* and *"target"*



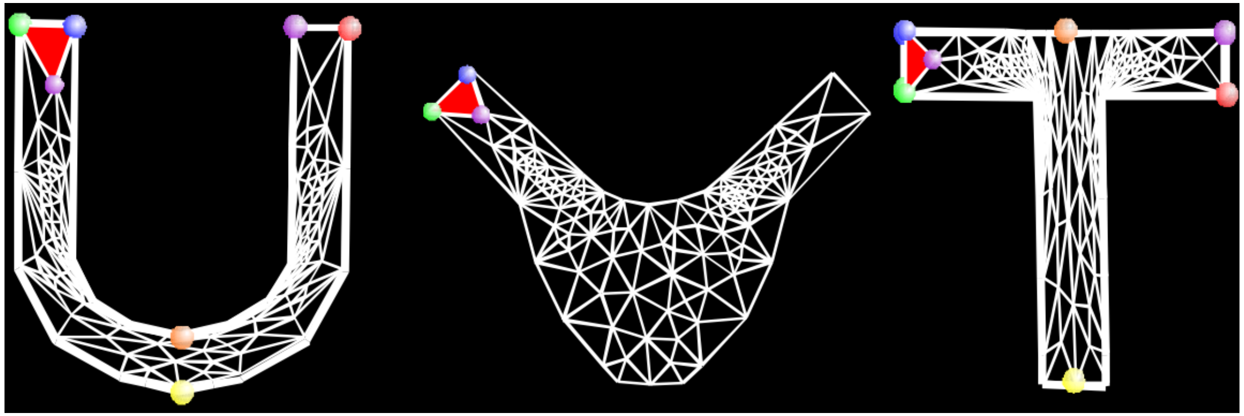
- Given t, the "morph parameter“
- Naive solution: linear interpolation

$$P(t) = (1-t)P + t\Phi(P)$$

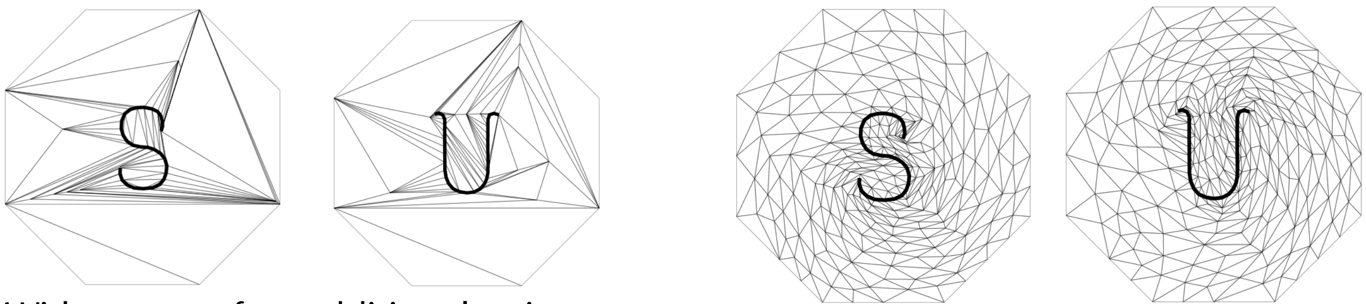
  - Example:



- Assumption: both meshs M_1 and M_2 are flat in the plane



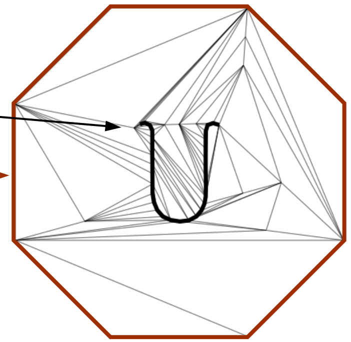
- Enclose both morph targets in a common, fixed polyline:



With none or few additional points

Many additional (Steiner) points

- Terms:
  - Inner vertices $V_I = \{P_i \mid i = 1 \ldots n\}$
  - Bounary points $V_B = \{P_i \mid i = n+1 \ldots n+k\}$
  - $N = n + k$
  - E = set of edges



- Using generalized barycentric coordinates, set up an LES for all vertices (for both M_1 and M_2):
  - For every $P_i \in V_I \;,\; i = 1 \ldots n$

    define $\lambda_{ij} > 0 \; \forall (i,j) \in E$

    and set $\lambda_{ij} = 0 \; \forall (i,j) \notin E$

- Therefore:

$$P_i = \sum_{j=1}^{N} \lambda_{ij} P_j \ , \quad i = 1 \dots n$$

- Written slightly differently:

$$P_i - \sum_{j=1}^{n} \lambda_{ij} P_j = \sum_{j=n+1}^{n+k} \lambda_{ij} P_j \ , \quad i = 1 \dots n$$

- $P_i = (x_i, y_i, z_i)$ yields the 3 LES's:

$$\underbrace{\begin{pmatrix} 1 & -\lambda_{12} & \cdots & -\lambda_{1n} \\ -\lambda_{21} & 1 & \cdots & \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} \lambda_{1,n+1} x_{n+1} + \cdots + \lambda_{1,n+k} x_{n+k} \\ \vdots \\ \vdots \end{pmatrix}}_{\mathbf{b}}$$

  - Analogue for y and z coordinates

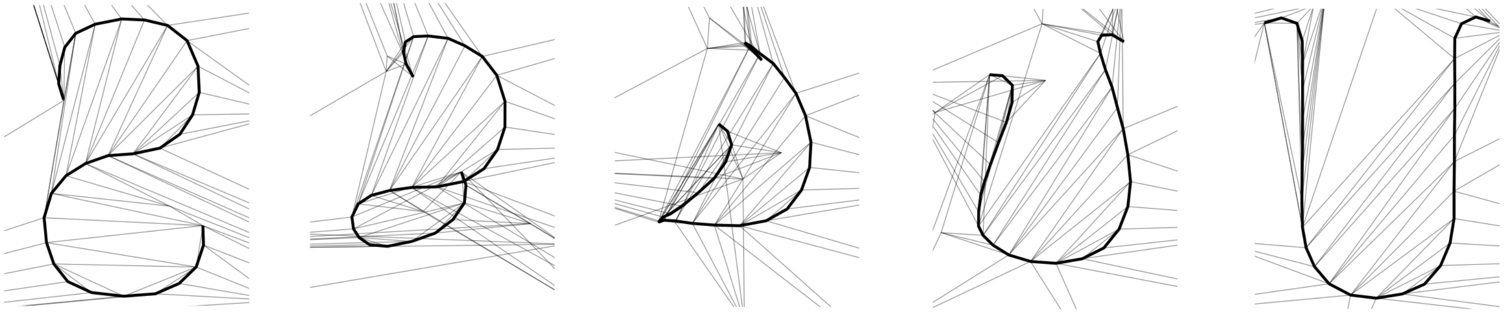
- The (simple) idea:
  1. Interpolate the λ's:
$$\lambda_{ij}^{(t)} = (1-t)\lambda_{ij}^{(1)} + t\lambda_{ij}^{(2)}$$
  2. Solve the 3 LES's for every t → yields P_i, $i = 1 \dots n$
- A less simple idea ("intrinsic morphing"):
  1. Interpolate the α's and r's (= angle & distances in the mesh):
$$\alpha_{ij}^{(t)} = (1-t)\alpha_{ij}^{(1)} + t\alpha_{ij}^{(2)} \qquad r_{ij}^{(t)} = (1-t)r_{ij}^{(1)} + tr_{ij}^{(2)}$$
  2. From that, calculate $\lambda(t)$'s
  3. Solve the 3 LES's
- Exercise: how many variables are interpolated in the three variants (for a specific t)?

Towards Implementation

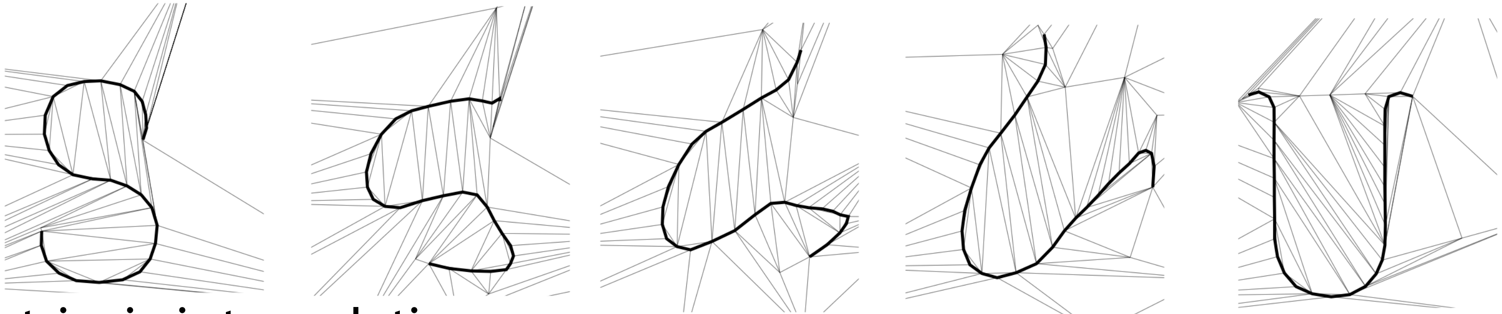
- Note:
  - The matrix *A* is not necessarily symmetrical
  - It is sparsely populated
  - It is, to a large degree, diagonally dominated, but it is not a band matrix
- Use an iterative solver
  - Initialize it with the solution of step *t*-1

Results

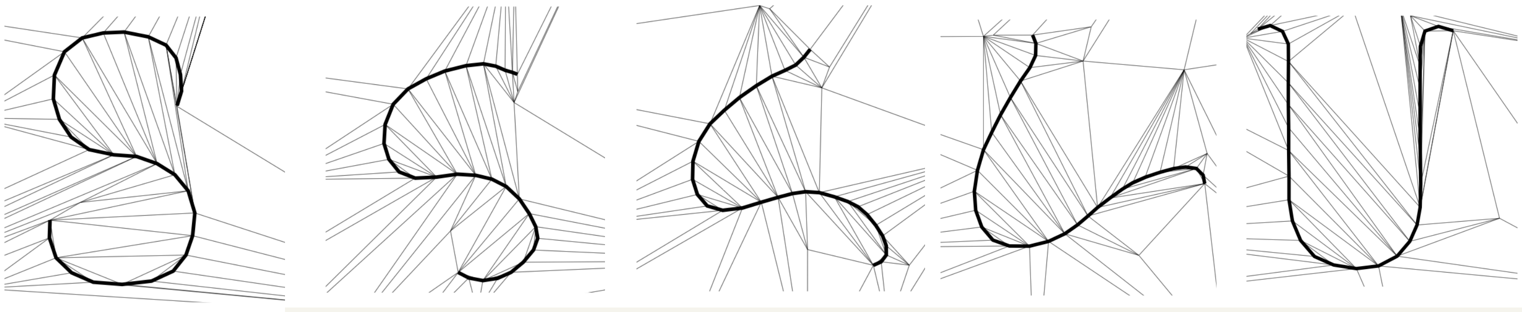
- Linear interpolation of vertices:



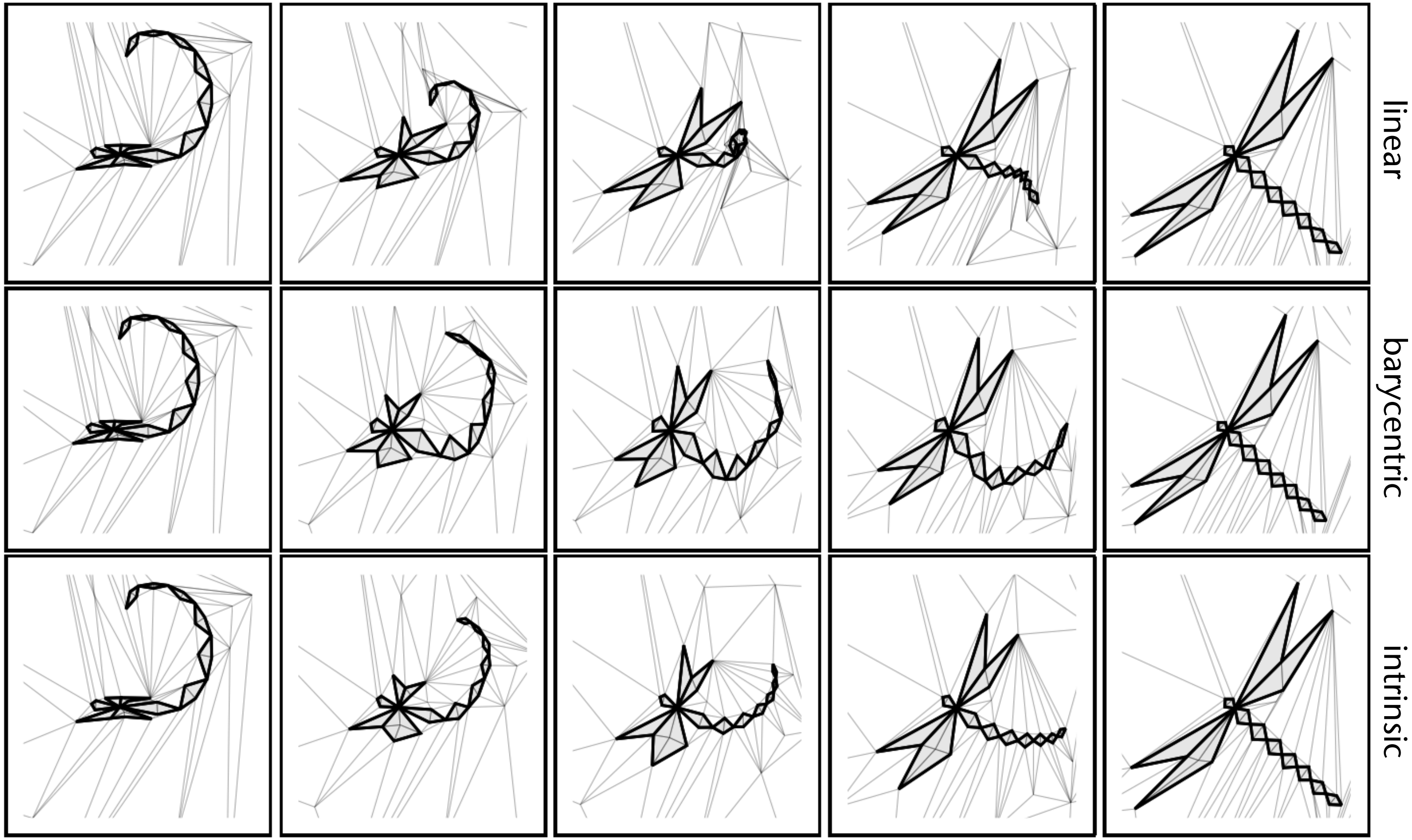
- Linear interpolation of barycentric coordinates:



- Intrinsic interpolation:



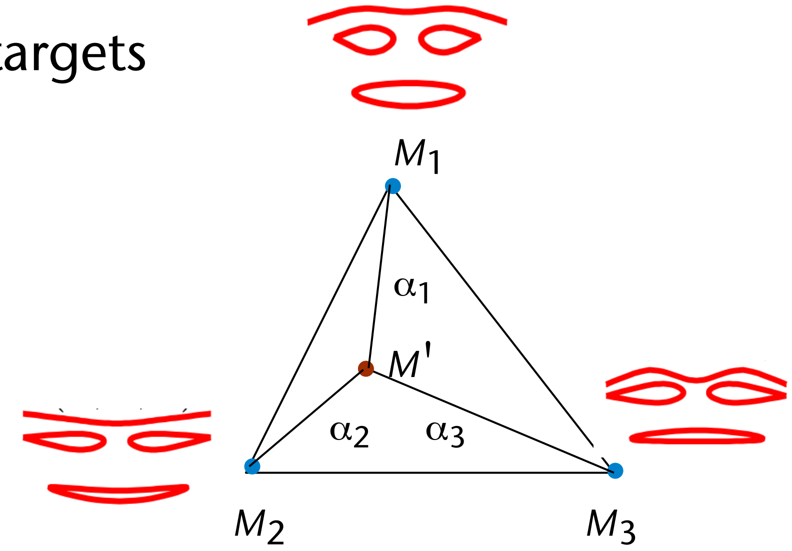
- Additional example:



Going Further

- Simultaneous morphing of multiple targets
- Given n morph targets M_i
- Task: determine an "in-between"

$$M' = \sum \alpha_k M_k$$



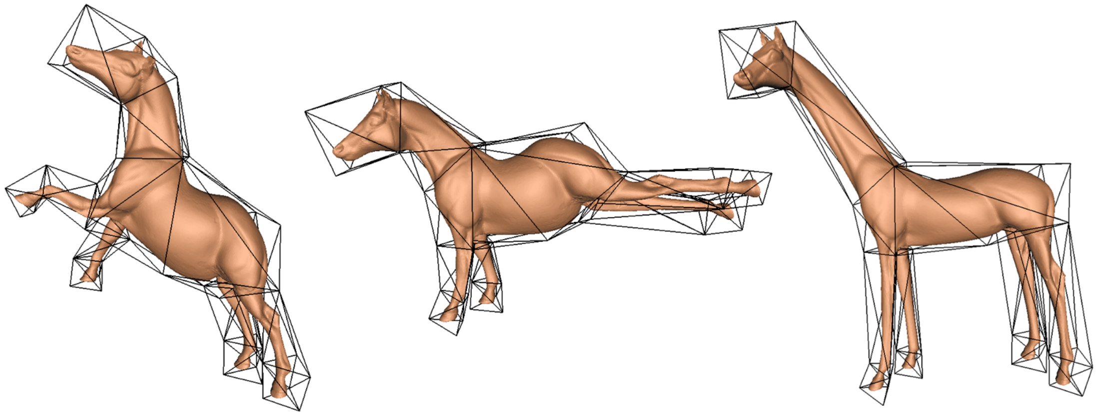
- Idea:
  1. Determine the barycentric coordinates $\lambda_{ij}^{(k)}$ of all M_k with regard to a fixed control polygon (or control polyhedron in 3D)
  2. Interpolate the λ's:

$$\lambda'_{ij} = \sum \alpha_k \lambda_{ij}^{(k)}$$

  3. Solve the LES's

Application: Cage-Based Shape Deformation

- Given: a mesh
- Task: targeted deformation of individual parts of the surface
- Example:



  - The "cages" (a.k.a. "control mesh") determines the deformation (and is set, for example, by the animator)
- Solution: …

Literature

- Take a look at papers on the class‘s homepage!
  - Under "Online-Literatur"