matfyz

Computer Graphics Charles University

Image morphing

© 1998-2015 Josef Pelikán

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

Morphing – „image transition”

- **transition between two raster images**
  - continuous transformation
  - „in-between” frame generation
- **geometric deformation** (warping)
  - image space warp (domain transform)
- **change of attribute** (image) **function**
  - color/attribute transition (range transform)

Diagram

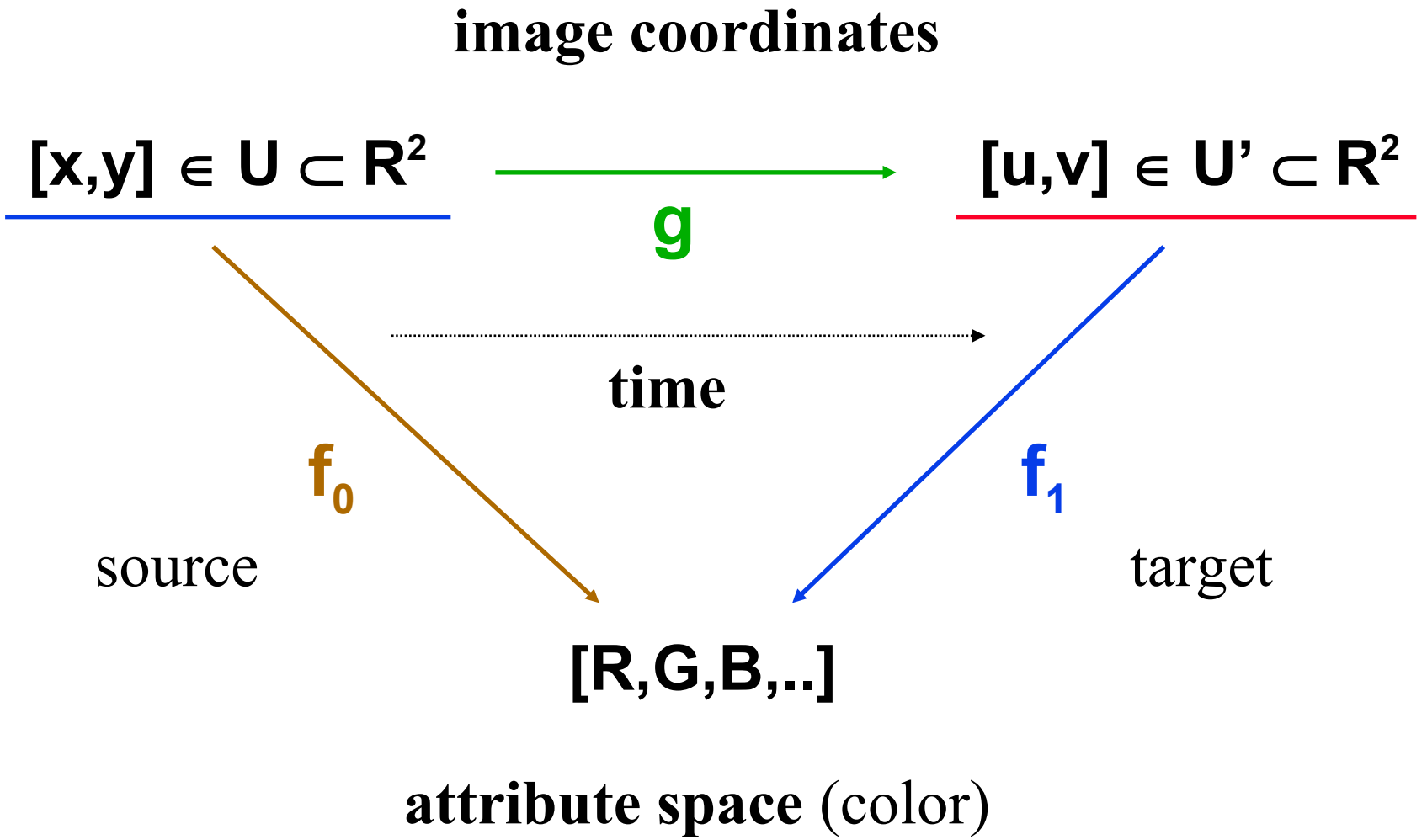


image coordinates
[x,y] ∈ U ⊂ R²
g
[u,v] ∈ U' ⊂ R²
time
f0
f1
source
target
[R,G,B,..]
attribute space (color)

Interpolation of deform. function

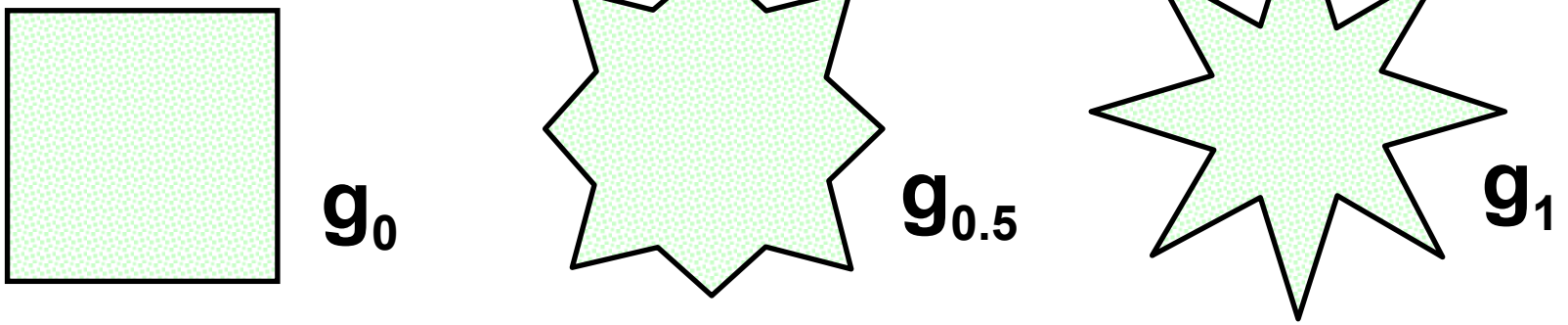
Introducing **time variable** into **g** function:

$$g_t(x,y) \qquad 0 \le t \le 1, \quad [x,y] \in U \subset R^2$$

Boundary conditions:

$$g_0(x,y) = [x,y]$$

$$g_1(x,y) = g(x,y) = [u,v]$$



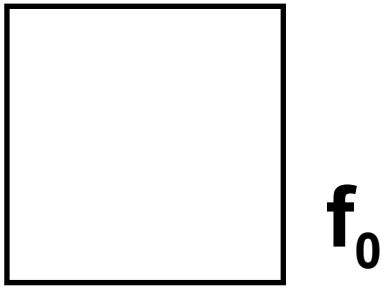
Interpolation of image function

Image function in time **t**:

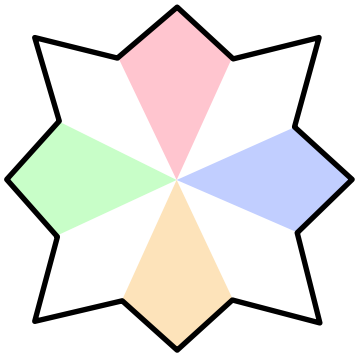
$$f_t[g_t(x,y)] = (1-t)\cdot f_0(x,y) + t\cdot f_1[g_1(x,y)]$$

$$0 \le t \le 1, \quad [x,y] \in U \subset R^2$$

can be precomputed



f_0



$f_{0.5}$

f_1

Deformation

- defined by **[triangle] meshes**
  - individual vertices can move in time (triangle mesh, spline net, ..)
- defined by a **set of features**
  - interpolation of arrow endpoints
  - better: arrow centers, lengths and orientations
- deformation of a **planar polygon**
  - transition function should minimize deformation energy

Planar polygon blending

- shape transition between two **planar polygons**
  - need not to have the same number of vertices
  - continuous transform (for „in-betweening”)
- method based on **physics**
  - elastic wire model
  - deformation work minimization
  - „stretching” and „bending” work

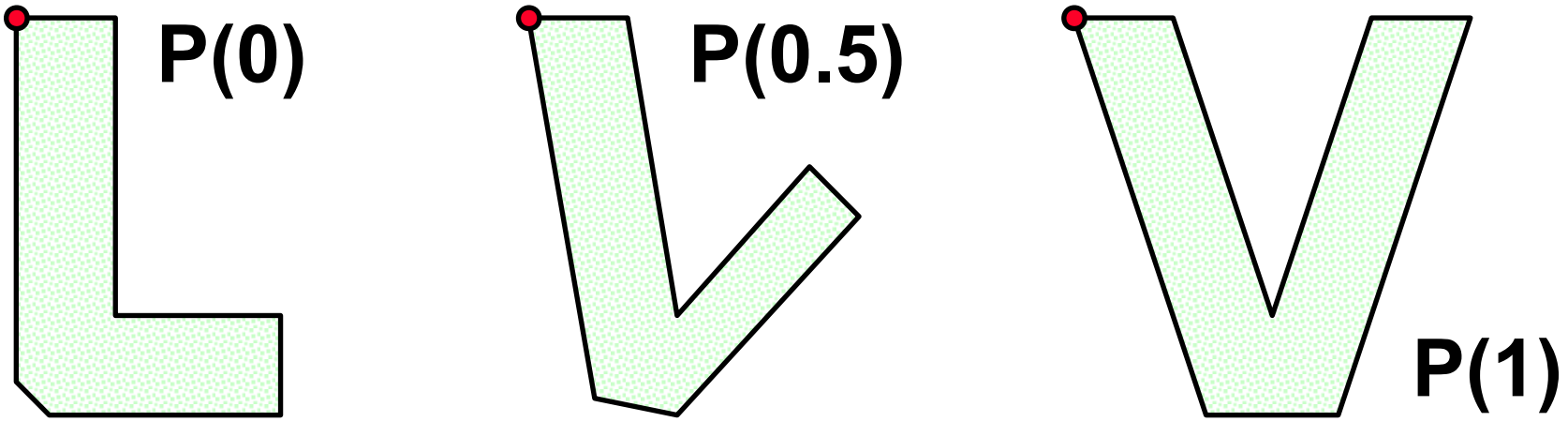
Linear vertex-position interpolation

Polygons with equal number of vertices:

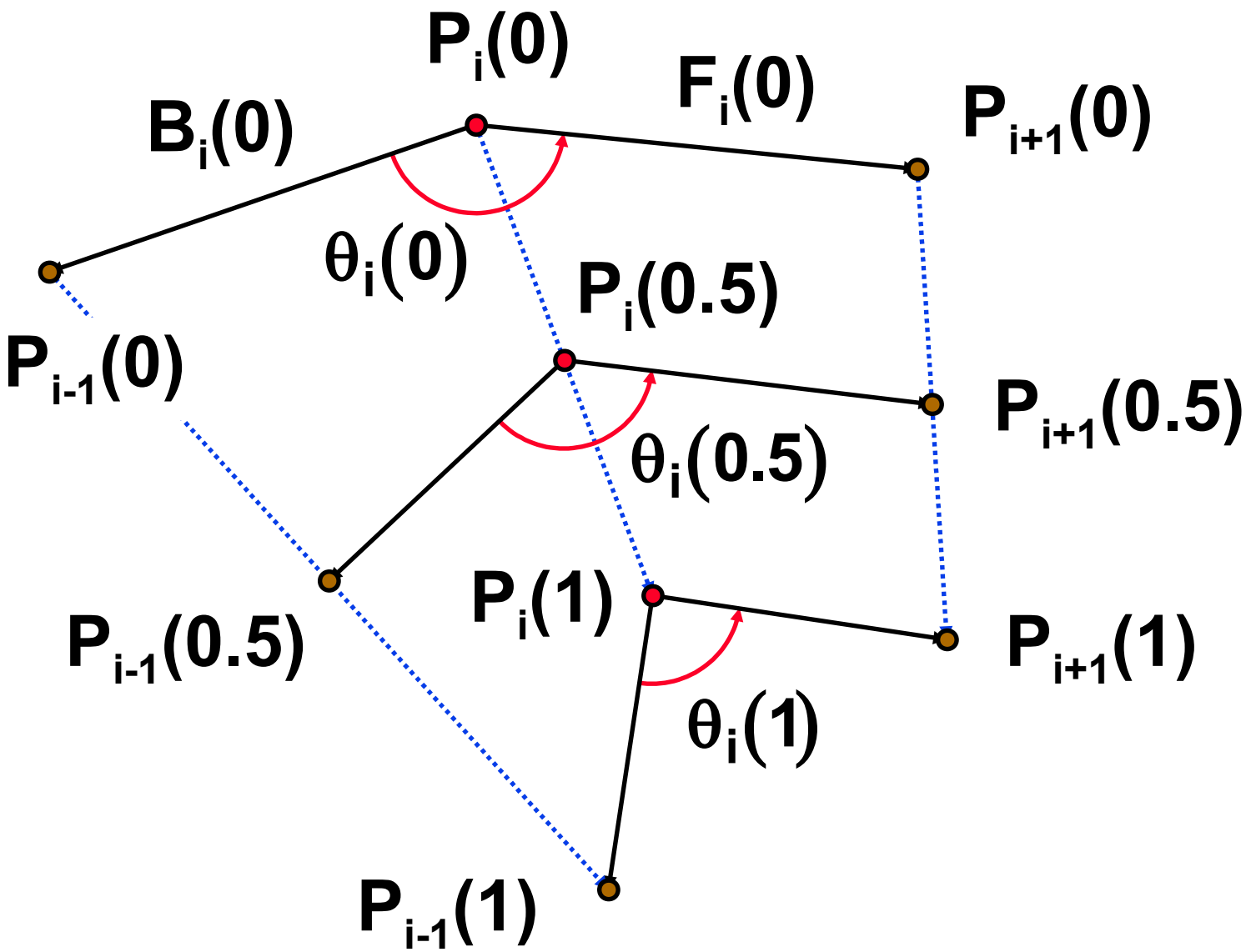
$$P(0) = [P_0(0), P_1(0), \ldots P_N(0) = P_0(0)]$$

$$P(1) = [P_0(1), P_1(1), \ldots P_N(1) = P_0(1)]$$

Intermediate position of P_i: $P_i(t) = (1-t) \cdot P_i(0) + t \cdot P_i(1)$



Angle interpolation



Inner angles

$$\theta_i(t) = \angle P_{i-1}(t), P_i(t), P_{i+1}(t)$$
$$= \angle B_i(t), 0, F_i(t)$$

For angle between two adjacent line segments:

$$\tan(\angle A, 0, B) = \frac{A \times B}{A \cdot B} = \frac{x_A y_B - x_B y_A}{x_A x_B + y_A y_B}$$

$$\tan\theta_i(t) = \frac{B_i(t) \times F_i(t)}{B_i(t) \cdot F_i(t)} = \frac{y_0(1-t)^2 + y_1 2t(1-t) + y_2 t^2}{x_0(1-t)^2 + x_1 2t(1-t) + x_2 t^2}$$

Inner angles

$$Q_0 = [x_0, y_0] = [F_i(0) \cdot B_i(0), F_i(0) \times B_i(0)]$$

$$Q_1 = [x_1, y_1] = \begin{bmatrix} \frac{1}{2}(F_i(1) \cdot B_i(0) + F_i(0) \cdot B_i(1)), \\ \frac{1}{2}(F_i(1) \times B_i(0) + F_i(0) \times B_i(1)) \end{bmatrix}$$

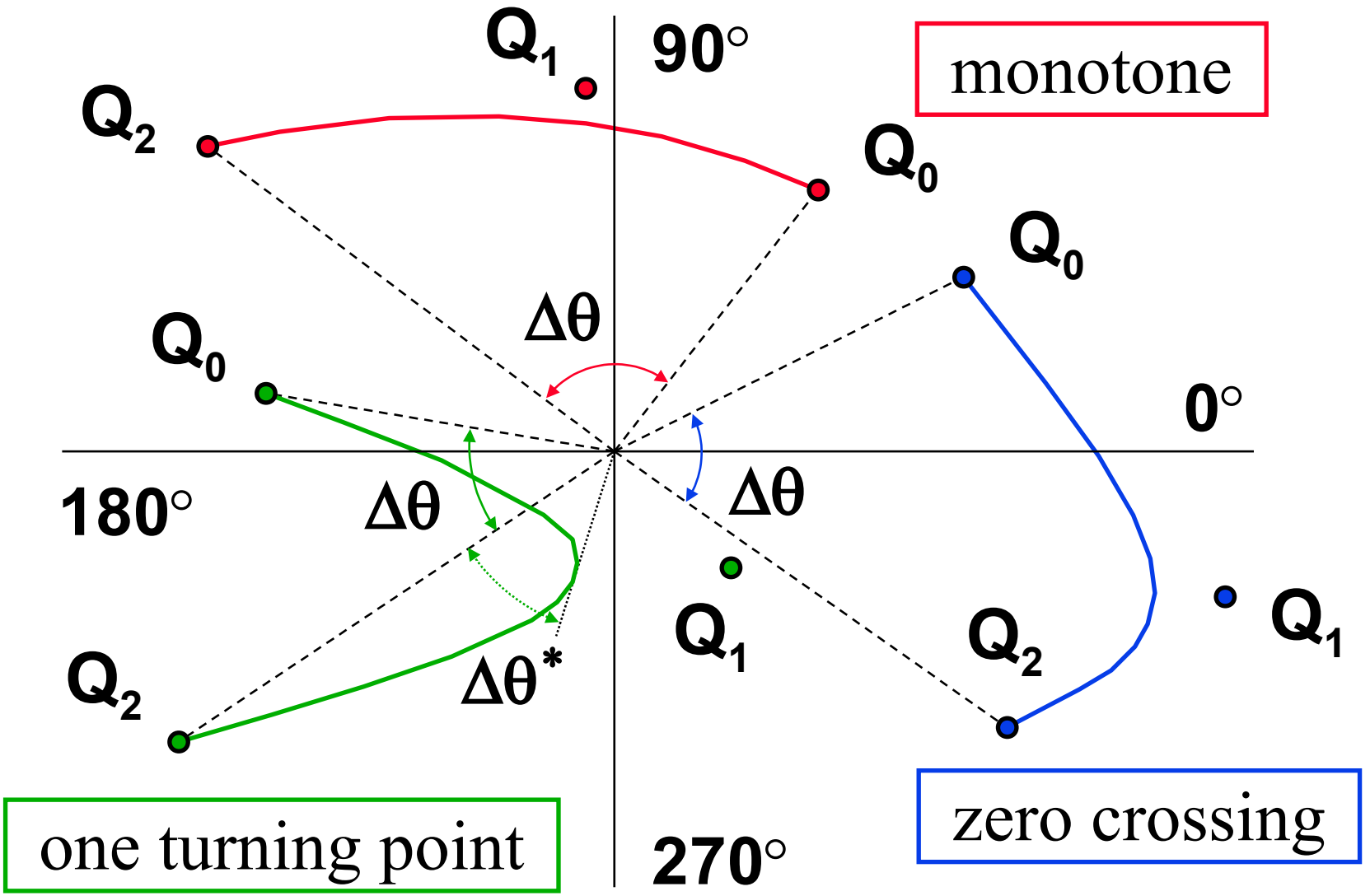
$$Q_2 = [x_2, y_2] = [F_i(1) \cdot B_i(1), F_i(1) \times B_i(1)]$$

Quadratic **Bézier curve** defined by control points Q_i:

$$Q(t) = Q_0 \cdot (1-t)^2 + Q_1 \cdot 2t(1-t) + Q_2 \cdot t^2$$

while this holds: $\theta_i(t) = \angle[1,0],[0,0],Q(t)$

Evolution of an inner angle



„Stretching" work

From physics we have:

$$W = \frac{\delta^2 \cdot A \cdot E}{2 \cdot L_0}$$

A	cross section
E	elastic modulus
$\mathbf{L_0}$	original length of the segment
δ	(absolute) stretching

There would be **infinite work** to shrinking a segment into **one point**!

„Stretching” work

Modified formula:

$$W_S = \frac{k_S \cdot \left| L_1 - L_0 \right|^{e_S}}{(1 - c_S) \cdot \min\{L_0, L_1\} + c_S \cdot \max\{L_0, L_1\}}$$

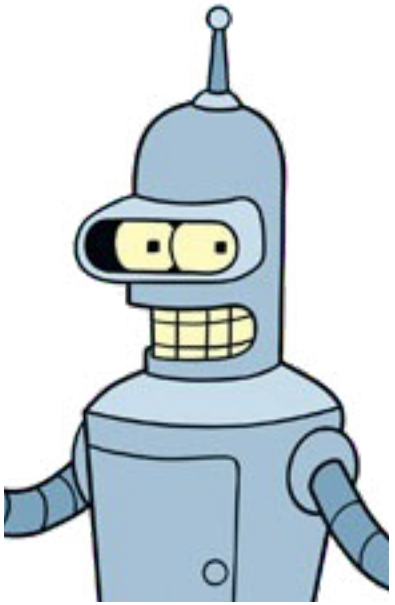
$k_S = A \cdot E$ elastic modulus times cross-section

e_S elastic „exponent“ (1 .. plastic, 2 .. elastic)

c_S degeneration penalty

„Bending" work

$$W_B = k_B \cdot \left(\Delta\theta + m_B \cdot \Delta\theta^*\right)^{e_B} + p_B(Q)$$



k_B	bending elastic modulus
m_B	penalty for non-monotony
$\Delta\theta$	total inner angle change
$\Delta\theta^*$	„fold" of the inner angle
p_B	penalty for zero-crossing („turnover"), **zero** if **Q(t)** does not cross zero

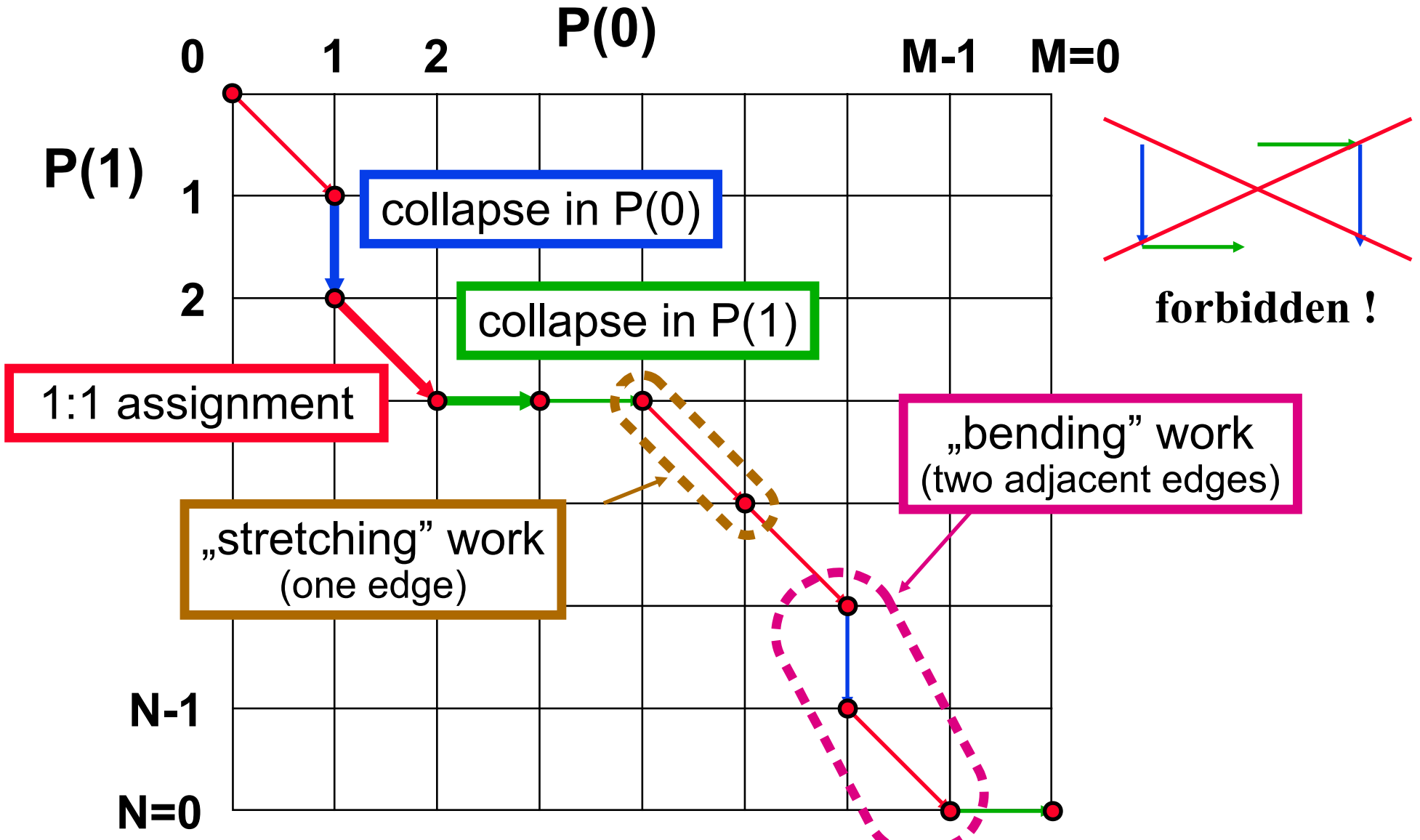
Total deformation work

- sum of all the „**stretching**” and „**bending**” components
  - „stretching” work for each segment
  - „bending” work for each vertex
- polygon **size normalization**
  - „stretching” work is not scale-invariant
  - uniform scaling to the same size of bounding rectangle's largest side

Global optimization

- looking for an optimal **source-to-target vertex assignment**
  - transition of vertices is given (linear)
  - to minimize total deformation work
  - not expecting equal number of vertices ⇒ edge degenerations
  - too large edges could be split in preprocessing
- **dynamic algorithm .. O(MN)**
  - assignment of short sequences ⇒ largest sequences

Assignment graph



Algorithm

- looking for a **path from [0,0] to [M,N]** with minimal cost
  - the path cannot return or turn by 90°
- each edge has its „**stretching**“ **work**
- each couple of adjacent edges define „**bending**” **work**
- **dynamic programming** – each path prefix has three values in the working matrix
  - if it went **E**, **SE** or **S**

General algorithm

- looking for a **path from [i,0] to [i,N]** with minimal cost
  - trying all potential **starting points [i,0]**
  - **cyclic connection** of left/right boundaries (toroidal topology)
- one level **higher complexity**
  - **O(M×N×M)** working matrix

The End

More info:

- **J. Gomes et al.**: ***Warping and Morphing of Graphical Objects***, C.N., SIGGRAPH'95
- **T. Sederberg, E. Greenwood**: ***A Physically Based Approach to 2D Shape Blending***, SIGGRAPH'92