CS 557 Winter 2001
Project 5: 2-D Free-Form Deformation
Due: Friday, 6 April

Dr. Thomas W. Sederberg

Enhance your CPLOT package by including the capability of performing 2D free-form deformation. Your program should respond to the following new comands:

DEFO(RMATION)
$X_{min}, Y_{min}, X_{max}, Y_{max}, m, n, iabs$
$\delta x_{00},\ \delta y_{00},\ w_{00},\ \delta x_{01},\ \delta y_{01},\ w_{01},\ \ldots\ , \delta x_{0m},\ \delta y_{0m},\ w_{0m}$
$\delta x_{10},\ \delta y_{10},\ w_{10},\ \delta x_{11},\ \delta y_{11},\ w_{11},\ \ldots\ , \delta x_{1m},\ \delta y_{1m},\ w_{1m}$
.
.
.
$\delta x_{n0},\ \delta y_{n0},\ w_{n0},\ \delta x_{n1},\ \delta y_{n1},\ w_{n1},\ \ldots\ , \delta x_{nm},\ \delta y_{nm},\ w_{nm}$

(X_{min}, Y_{min}) and (X_{max}, Y_{max}) define the corners of a deformation region, and m and n specify how many control points are to be used: there are to be $m+1$ vertical columns and $n+1$ horizontal rows of control points. Control point $(0,0)$ should lie on the lower left corner, and control point (m,n) on the upper right corner. If *iabs* is zero, $(\delta x_{ij}, \delta y_{ij})$ specifies the **relative** displacement that control point i,j is to receive, and w_{ij} is the weight. If *iabs* is one, $(\delta x_{ij}, \delta y_{ij})$ specifies the **absolute** displacement that control point i,j is to receive, and w_{ij} is the weight.

The deformation is defined in terms of a **rational bivariate tensor product Bernstein polynomial** which takes the form

$$\mathbf{X}(s,t) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij} B_j^m(s) B_i^n(t) \mathbf{P}_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} w_{ij} B_j^m(s) B_i^n(t)} \tag{1}$$

where $B_i^n(t)$ and $B_j^m(s)$ are Bézier blending functions, and s and t are the local coordinates of a point with respect to the deformation region.

The s and t coordinates of a point in the deformation region range between 0 and 1. Thus, for a point (x,y) within the rectangular region,

$$s = \frac{x - X_{min}}{X_{max} - X_{min}}, \qquad t = \frac{y - Y_{min}}{Y_{max} - Y_{min}}. \tag{2}$$

The control point values $\mathbf{P}_{ij}$ are the actual (x,y) coordinates of the displaced control point i,j. For example, $\mathbf{P}_{00} = (X_{min} + \delta x_{00},\ Y_{min} + \delta y_{00})$. If *iabs* is zero,

$$\mathbf{P}_{ij} = (X_{min} + \frac{i}{m}(X_{max} - X_{min}) + \delta x_{ij},\ Y_{min} + \frac{j}{n}(Y_{max} - Y_{min}) + \delta y_{ij}). \tag{3}$$

Otherwise, if *iabs* is one,

$$\mathbf{P}_{ij} = (\delta x_{ij}, \delta y_{ij}). \tag{4}$$

To compute the location of a point which undergoes deformation, first compute its (s,t) coordinates using equation 2. If they are within the range $[0,1]$, then compute the deformed (x,y) coordinates using equation 1. If the point is not within the deformed region, do not deform it. Deformation should be imposed on the endpoints of all line segments immediately prior to sending then through the pipeline.

Other commands that should be implemented are:

PDEF
$iflag,\ rad$

If $iflag = 1$, then for all subsequent **DEFO** commands, plot the control points as circles with radius rad, and plot line segments between neighboring control points. If $iflag = 0$, disable the plotting of control points.

GRID
$X_{min}, Y_{min}, X_{max}, Y_{max}, nx, ny, nseg$

Plot a grid of lines over a rectangular region whose corners are (X_{min}, Y_{min}) and (X_{max}, Y_{max}). There should be $nx + 1$ vertical lines and $ny + 1$ horizontal lines. Each line should be divided into $nseg$ pieces to be sent through the deformation and pipeline.

Hand in:

1. A plot showing a circle and a grid deformed using an FFD with $m = n = 2$ and $iabs = 0$.
2a. A plot showing your name deformed into an arc using an FFD with $m = 4$, $n = 1$, and $iabs = 1$.
2b. Your name deformed using an FFD in which only the weights are changed, not the control point locations.
3. A circle deformed so that half of it lies inside the deformation region. Do one deformation in which the deformed region is G^0 with the undeformed region, and another in which the deformed region is G^1 with the undeformed region.
4. Through trial and error, find an FFD that flattens a quadratic polynomial Bézier curve with control points

$$\mathbf{P}_0 = (1,1); \qquad \mathbf{P}_1 = (3,2); \qquad \mathbf{P}_2 = (4,1)$$

into a straight line. Show the curve and a grid before and after deformation.
5. A square deformed into a circle.
6. An FFD example of your design.