

The problem is to find the shortest distance from Bézier curve defined by the control points  $\{x_0, y_0\}, \{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}$  to point  $\{x_4, y_4\}$ . The Bézier curve is defined in terms of  $u$ .

$$\begin{aligned} x &= a_3 u^3 + a_2 u^2 + a_1 u + a_0 \\ y &= b_3 u^3 + b_2 u^2 + b_1 u + b_0 \end{aligned}$$

Where  $a_3, a_2, a_1, a_0, b_3, b_2, b_1$ , and  $b_0$  are defined in terms of the control points.

$$a_3 = \frac{x_3 + 3(x_1 - x_2) - x_0}{8}$$

$$a_2 = \frac{3(x_3 - x_2 - x_1 + x_0)}{8}$$

$$a_1 = \frac{x_3 - x_0}{2} - a_3$$

$$a_0 = \frac{x_3 + x_0}{2} - a_2$$

$$b_3 = \frac{y_3 + 3(y_1 - y_2) - y_0}{8}$$

$$b_2 = \frac{3(y_3 - y_2 - y_1 + y_0)}{8}$$

$$b_1 = \frac{y_3 - y_0}{2} - b_3$$

$$b_0 = \frac{y_3 + y_0}{2} - b_2$$

The distance is  $s$ .

$$s = \sqrt{(x - x_4)^2 + (y - y_4)^2}$$

$$s^2 = (x - x_4)^2 + (y - y_4)^2$$

When the distance is minimum, the rate of change of distance is zero.

$$\frac{ds}{du} = 0$$

$$2s \frac{ds}{du} = 2(x - x_4) \frac{dx}{du} + 2(y - y_4) \frac{dy}{du}$$

$$0 = (x - x_4) \frac{dx}{du} + (y - y_4) \frac{dy}{du}$$

$$\frac{dx}{du} = 3a_3 u^2 + 2a_2 u + a_1$$

$$\frac{dy}{du} = 3b_3 u^2 + 2b_2 u + b_1$$

The order of this equation five. The solution requires trial and error methods. My favorite method is interpolation. Here linear interpolation will work very well if we start close to the correct answer.

$$z = (x - x_4) \frac{dx}{du} + (y - y_4) \frac{dy}{du}$$

Assume  $z$  is a linear function of  $u$ .

$$z = a u + b$$

$$z_1 = a u_1 + b$$

$$z_2 = a u_2 + b$$

$$z - z_1 = a (u - u_1)$$

$$z_2 - z_1 = a (u_2 - u_1)$$

$$\frac{z - z_1}{z_2 - z_1} = \frac{u - u_1}{u_2 - u_1}$$

Now  $z = 0$ .

$$\frac{-z_1}{z_2 - z_1} = \frac{u - u_1}{u_2 - u_1}$$

$$\frac{z_2 u_1 - z_1 u_2}{z_2 - z_1} = u$$

The following BASIC program uses the method of finding distance. The program also searches for the minimum squared distance between points and a curve.

```

REM BEZIER.BAS JIM 20DEC92 12:37
DATA 2,3,5,8,8,14,11,17,14,17,16,15,18,11,-1
DATA 2,10,5,12,8,11,11,8,14,6,17,5,19,10,-1
DATA 2,5,5,7,8,8,12,12,13,14,12,17,10,18,8,17,7,14,8,12,12,8,15,7,18,5,-1
OPEN "BEZIER.OUT" FOR OUTPUT AS #1
OPEN "BEZ.ps" FOR OUTPUT AS #2
CLS
psscale = 20
FOR example% = 1 TO 3
  REDIM rawdata(32)
  FOR I% = 0 TO 32
    READ rawdata(I%)
    IF rawdata(I%) < 0! THEN EXIT FOR
  NEXT I%
  n% = I% - 1
  PRINT "Example "; example%; (n% + 1) \ 2; " points"
  PRINT #1, ""
  PRINT #1, "Example "; example%; (n% + 1) \ 2; " points"
  PRINT #1, " #   x     y"
  J% = 0
  FOR I% = 0 TO n% STEP 2
    J% = J% + 1
    PRINT #1, USING "###.###.###.###.###"; J%; rawdata(I%); rawdata(I% + 1)
    LPRINT USING "###.###.###.### 3 0 360 arc fill"; rawdata(I%) * psscale; rawdata(I% + 1) * psscale
    PRINT #2, USING "###.###.###.### 3 0 360 arc fill"; rawdata(I%) * psscale; rawdata(I% + 1) * psscale
  NEXT I%
  x0 = rawdata(0)
  y0 = rawdata(1)
  x1 = rawdata(2)
  y1 = rawdata(3)
  x2 = rawdata(n% - 3)
  y2 = rawdata(n% - 2)
  x3 = rawdata(n% - 1)
  y3 = rawdata(n%)
  IF example% = 3 THEN
    'special guess for loop
    x1 = 8 * x1 - 7 * x0
    y1 = 8 * y1 - 7 * y0
    x2 = 8 * x2 - 7 * x3
    y2 = 8 * y2 - 7 * y3
  ELSE
    x1 = 2 * x1 - x0
    y1 = 2 * y1 - y0
    x2 = 2 * x2 - x3
    y2 = 2 * y2 - y3
  END IF
  GOSUB distance
  LPRINT ".1 setlinewidth"
  PRINT #2, ".1 setlinewidth"
  GOSUB curveto
  e1 = totalerror
  FOR Retry% = 1 TO 6
    PRINT
    PRINT "Retry "; Retry%
    PRINT #1, "Retry "; Retry%
    PRINT #1, " x1   y1   x2   y2   error"
  e3 = .5

```

```

x1a = x1
DO
  x1 = x1 + (x1 - x0) * e3
  GOSUB distance
  e2 = totalerror
  IF e2 = e1 THEN
    EXIT DO
  ELSEIF e2 > e1 THEN
    x1 = x1a
    e3 = -e3 / 3
    IF ABS(e3) < .001 THEN EXIT DO
  ELSE
    e1 = e2
    x1a = x1
  END IF
LOOP
e3 = .5
y1a = y1
DO
  y1 = y1 + (y1 - y0) * e3
  GOSUB distance
  e2 = totalerror
  IF e2 = e1 THEN
    EXIT DO
  ELSEIF e2 > e1 THEN
    y1 = y1a
    e3 = -e3 / 3
    IF ABS(e3) < .01 THEN EXIT DO
  ELSE
    e1 = e2
    y1a = y1
  END IF
LOOP
e3 = .5
x2a = x2
DO
  x2 = x2 + (x2 - x3) * e3
  GOSUB distance
  e2 = totalerror
  IF e2 = e1 THEN
    EXIT DO
  ELSEIF e2 > e1 THEN
    x2 = x2a
    e3 = -e3 / 3
    IF ABS(e3) < .01 THEN EXIT DO
  ELSE
    e1 = e2
    x2a = x2
  END IF
LOOP
e3 = .5
y2a = y2
DO
  y2 = y2 + (y2 - y3) * e3
  GOSUB distance
  e2 = totalerror
  IF e2 = e1 THEN
    EXIT DO
  ELSEIF e2 > e1 THEN
    y2 = y2a
    e3 = -e3 / 3
    IF ABS(e3) < .01 THEN EXIT DO

```

```

ELSE
  e1 = e2
  y2a = y2
END IF
LOOP
IF Retry% = 6 THEN
  LPRINT "1 setlinewidth"
  PRINT #2, "1 setlinewidth"
END IF
GOSUB curveto
NEXT Retry%
LPRINT "100 200 translate"
PRINT #2, "100 200 translate"
NEXT example%
LPRINT "showpage"
PRINT #2, "showpage"
CLOSE #1
CLOSE #2
END
,

Bezier:
x = a0 + u * (a1 + u * (a2 + u * a3))
y = b0 + u * (b1 + u * (b2 + u * b3))
dx4 = x - x4: dy4 = y - y4
dx = a1 + u * (2 * a2 + u * 3 * a3)
dy = b1 + u * (2 * b2 + u * 3 * b3)
z = dx * dx4 + dy * dy4
s = dx4 * dx4 + dy4 * dy4
RETURN
,

distance:
totalerror = 0!
a3 = (x3 - x0 + 3 * (x1 - x2)) / 8
b3 = (y3 - y0 + 3 * (y1 - y2)) / 8
a2 = (x3 + x0 - x1 - x2) * 3 / 8
b2 = (y3 + y0 - y1 - y2) * 3 / 8
a1 = (x3 - x0) / 2 - a3
b1 = (y3 - y0) / 2 - b3
a0 = (x3 + x0) / 2 - a2
b0 = (y3 + y0) / 2 - b2
FOR !% = 2 TO n% - 2 STEP 2
  x4 = rawdata(!%)
  y4 = rawdata(!% + 1)
  stepsize = 2 / (n% + 1)
  FOR u = -1! TO 1.01 STEP stepsize
    GOSUB Bezier
    IF s = 0! THEN u1 = u: z1 = z: s1 = s: EXIT FOR
    IF u = -1! THEN u1 = u: z1 = z: s1 = s
    IF s < s1 THEN u1 = u: z1 = z: s1 = s
  NEXT u
  IF s1 <> 0! THEN
    u = u1 + stepsize
    IF u > 1! THEN u = 1! - stepsize
  DO
    GOSUB Bezier
    IF s = 0! THEN EXIT DO
    IF z = 0! THEN EXIT DO
    u2 = u
    z2 = z
    temp = z2 - z1
    IF temp <> 0! THEN
      u = (z2 * u1 - z1 * u2) / temp
    
```

```

ELSE
  u = (u1 + u2) / 2!
END IF
IF u > 1! THEN
  u = 1!
ELSEIF u < -1! THEN
  u = -1!
END IF
IF ABS(u - u2) < .0001 THEN EXIT DO
u1 = u2
z1 = z2
LOOP
END IF
totalerror = totalerror + s
NEXT I%
PRINT totalerror;
PRINT #1, USING "#####.### #####.### #####.### #####.### #####.###"; x1; y1; x2; y2; totalerror
RETURN
,

curveto:
LPRINT USING "#####.### #####.### moveto"; x0 * psscale; y0 * psscale
PRINT #2, USING "#####.### #####.### moveto"; x0 * psscale; y0 * psscale
F$ = "#####.### #####.### #####.### #####.### #####.### #####.### curveto stroke"
LPRINT USING F$; x1 * psscale; y1 * psscale; x2 * psscale; y2 * psscale; x3 * psscale; y3 * psscale
PRINT #2, USING F$; x1 * psscale; y1 * psscale; x2 * psscale; y2 * psscale; x3 * psscale; y3 * psscale
RETURN

```

