Rapid Termination Evaluation for Recursive Subdivision of Bezier Curves

Thomas F. Hain
School of Computer and Information Sciences,
University of South Alabama, Mobile, AL, U.S.A.

Abstract – *Bézier curve flattening by recursive subdivision requires that the maximum excursion of the subdivided curve segment be known so that recursion can be terminated once this value drops below the specified flatness criterion. A much more accurate method than the most commonly used techniques to evaluate this distance is presented. This method stops recursion sooner, significantly reducing the number of generated straight line segments that are used to approximate the curve to within the given flatness. The incremental computational overhead is minimal.*[1]

Key words: *recursive subdivision, Bézier curves, flattening.*

1 Introduction

The fast graphical rendering of curves usually involves the reduction of the curve to an approximating polyline—a process called "flattening" the curve. The required goodness of fit is specified by a single scalar parameter, d_{flat}, called the "flatness". The mathematical curve should not deviate from the straight-line segments of the approximating polyline by a distance greater than the flatness.

In the case of two-dimensional cubic Bézier curves, flattening can be achieved by forward differencing [2], or by recursive subdivision [1]. The first technique, while fast, has the disadvantage that the number of line segments in the generated polyline is fixed, rather than depending on the curvature and size of the curve being rendered. Attempts at adapting the step size in forward differencing (by doubling or halving the step size along the path) are described in [3]. Recursive subdivision requires recursively bisecting the curve until each of the resulting Bézier curve segments lies completely within a specified distance of the line segment joining its endpoints, and then replacing each of the resulting Bezier curves by the line segment joining its endpoints. The stopping criterion for the recursion requires a computationally expensive calculation of the actual maximum perpendicular distance, $d_{\max}$, that a cubic Bézier curve, specified by its control points $P_0,\cdots,P_3$, deviates from the line through P_0P_3. Alternatively, a reasonably good but inexpensive estimate of $d_{\max}$, denoted by $\breve{d}_{\max}$, such that $\breve{d}_{\max} \le d_{\max}$, is desired. We may stop the recursion when $d_{\max} \le d_{flat}$, or, because of the previous requirement, when $\breve{d}_{\max} \le d_{flat}$.

A method commonly used to estimate the maximum deviation of the curve from the line through P_0P_3 is to calculate either the x- or y-component of the vector from I_1, the point closest to control point P_1 on line through P_0P_3, to P_1 (see Figure 1—a similar figure could be drawn with both inner control points being on the same side of the chord), depending on whether the magnitude of the slope of $\overline{P_oP_3}$ is greater or less than one respectively. The appropriate component of $\overline{I_2P_2}$ is also calculated. This forms the estimate $\breve{d}_{\max}$ for this method; i.e., if either of these components is greater than the flatness, the curve is subdivided, and each resulting Bézier segment is recursively flattened. This method gives a good estimate only if

1. the magnitude of the slope of $\overline{P_oP_3}$ is close to one, and
2. the two control points are equidistant from the line $\overline{P_oP_3}$, and on the same side.

The first condition will not be true in general, but the second will tend to be true when the subdivi-

This research was funded by a grant from QMS Inc., Mobile, Alabama.

sion has occurred often enough to be close to recursion termination. Here $d_{\max}$ can be calculated as $0.75\times\max(|A|,|B|)$, which is not too far from the estimated value of $\frac{1}{\sqrt{2}}X = 0.707\times\max(|A|,|B|)$. However, the estimate is about 41% too high if the slope of $\overline{P_oP_3}$ is either vertical or horizontal, and nearly 350% too large in that case if $A=-B$. The excessively conservative estimate by this technique causes unnecessary recursion depth, and consequently more line segments/vertices are generated than would be necessary to provide a curve flattened to the given flatness.

A final problem with most rendering implementations is that if $|A|$ and $|B|$ are within the flatness criterion, but I_1 or I_2 are outside the line segment $\overline{P_oP_3}$, recursion is continued until I_1 and I_2 are within a distance of P_0 or P_3 equal to the flatness. Thus, a series of vertices are generated that are collinear, and essentially redundant. However, this is done to ensure that the curve undergoes the appropriate excursion in the direction of $\overline{P_oP_3}$.

2 Calculation of maximum excursion

In this work, an expression for $d_{\max}$, the maximum deviation of the curve from the line through P_0P_3, in terms of *A* and *B* is developed. It is shown that this expression can be approximated to within 1.2% of the actual values by a quadratic in $\frac{E}{F}$, where $E=\min\{|A|,|B|\}$, and $F=\max\{|A|,|B|\}$. In particular, the approximating expression is more accurate at more critical values of $\frac{E}{F}$. This solves the first (and major) problem. Using this approach, tests have shown that the number of vertices generated during Bézier drawing is reduced by about 26%.

The second problem, that of generating many redundant collinear vertices when the curve is close to the line defined by P_0 and P_3, but when the convex hull extends to beyond the limits of $\overline{P_oP_3}$, has been solved by reducing the problem into a number of cases, and solving for the extremum point or points on the curve explicitly. These extremum points are resolved to vertices without the need for any recursion.

3 Mathematical derivations

We are given the four control points of a Bézier curve, $P_0,\cdots,P_3$. We will take closed Bézier curves (where the first and last control points at coincident, i.e., $P_0 \approx P_3$) as a special case, by unconditionally recursing once.

Consider a normalized coordinate system with *s*- and *r*-axes such that the origin is at P_0 and the *r*-axis is in the direction $\overline{P_0P_3}$, and the *s*-axis is orthogonal to this, in a right-handed sense. The control point P_3 should be at (1,0). It should be noted that orthogonal component cubic equations of Bézier curves are completely separable.

The normalized coordinates of an arbitrary point $P_c=(r,s)$ in this new system are calculated as follows,

$$r=\frac{(P_{0_y}-P_{c_y})(P_{0_y}-P_{3_y})-(P_{0_x}-P_{c_x})(P_{3_x}-P_{0_x})}{L^2}$$

$$s=\frac{(P_{0_y}-P_{c_y})(P_{3_x}-P_{0_x})-(P_{0_x}-P_{c_x})(P_{3_y}-P_{0_y})}{L^2}$$

where,

$$L=\|P_0P_3\|=\sqrt{(P_{3_x}-P_{0_x})^2+(P_{3_y}-P_{0_y})^2}$$

Note that the denominator of the equation for *s* can be omitted since we will be using a function of the ratio of two *s*-values (L^2 cancels out) to evaluate a normalized (not involving *L*) maximum excursion, l_{norm}, of the curve from the straight line through P_0P_3.

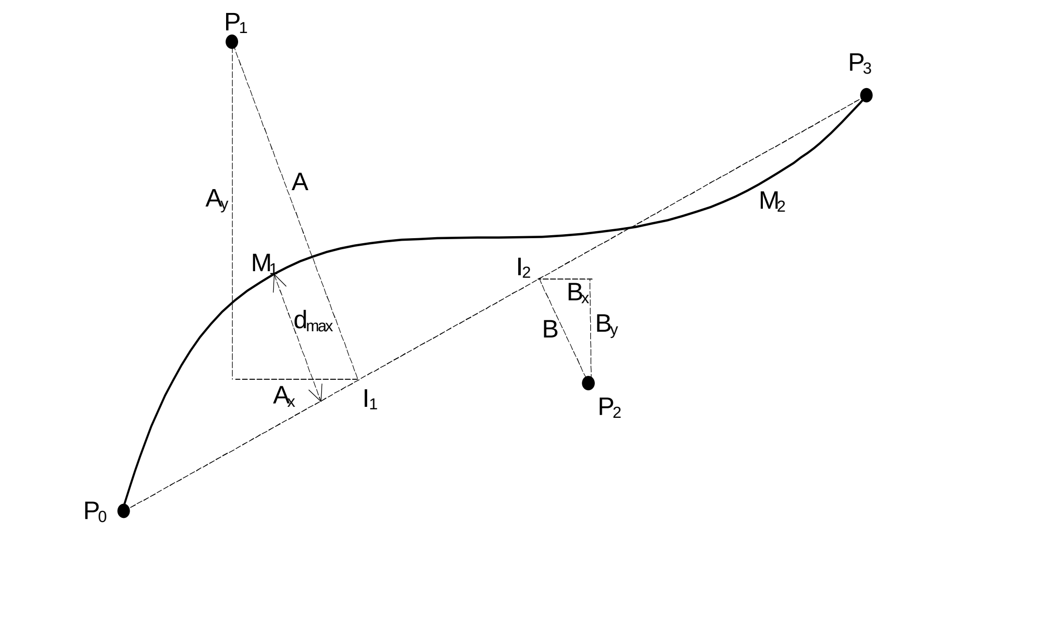

Figure 1 Bézier with P_1 and P_2 on opposite side of $\overline{P_0P_3}$.

To get actual distances from these coordinates, we must multiply by L. When it comes to the flatness decision, we can compare $(l_{norm}L)^2 = l_{norm}{}^2L^2$ with $l_{flat}{}^2$ (where l_{flat} is the flatness criterion distance), thereby avoiding a square root evaluation.

3.1 Transverse component of curve

The general equation for the *s*-component of points on the curve as a function of the parametric value *t*, where $0 \le t \le 1$, is

$$s(t) = (1-t)^3 s_0 + 3t(1-t)^2 s_1 + 3t^2(1-t)s_2 + t^3 s_3$$

With the selected coordinate system, $s_0 = s_3 = 0$ (i.e., curve endpoints lie on the *r*-axis), so

$$\begin{aligned} s(t) &= 3t(1-2t+t^2)s_1 + 3t^2(1-t)s_2 \\ &= (3t-6t^2+3t^3)s_1 + (3t^2-3t^3)s_2 \end{aligned}$$

We want to find maximum (in fact, there may be two local maxima) excursion of the curve from the *r*-axis. At these points,

$$\frac{ds}{dt} = (3-12t+9t^2)s_1 + (6t-9t^2)s_2 = 0$$

Let us rescale the *s*-axis such that $s_i = 1$, where $|s_i| = \max(|s_1|,|s_2|)$, and $i = 1,2$. That is, we rescale the *s*-axis so that the control point furthest from the line through P_0P_3 (the chord) is at distance +1 (on the left of the chord). Since the maximum excursion of the curve away from the chord is independent of the order of the inner control points P_1, and P_2, we can swap the *s*-coordinates of P_1 and P_2 such that $i = 1$. That is, $s_1 = 1$, and $s_2 = v$, where $-1 \le v \le +1$. Thus,

$$\begin{aligned} \frac{ds}{dt} &= 3-12t+9t^2+6tv-9t^2v \\ &= 3+(6v-12)t+9(1-v)t^2 \\ &= 0 \end{aligned} \tag{1}$$

We now solve the equation

$$1+2(v-2)t+3(1-v)t^2 = 0$$

for *t*,

$$
\begin{aligned}
t &= \frac{\cancel{2}(2-v) \pm \sqrt{\cancel{4}(v-2)^2 - \cancel{4} \times 3(1-v)}}{\cancel{2} \times 3(1-v)} \\
&= \frac{(2-v) \pm \sqrt{r^2 - 4v + 4 - 3 + 3v}}{3(1-v)} \\
&= \frac{(2-v) \pm \sqrt{v^2 - v + 1}}{3(1-v)}
\end{aligned}
$$

The maximum excursion of the curve will occur nearer to the P_0 end (where $t = 0$) since we had arranged that $s_1 \geq s_2$. Thus, the solution for *t* that we are interested in is,

$$t_{\max} = \frac{(2-v) - \sqrt{v^2 - v + 1}}{3(1-v)}$$

Now, we can substitute this value of *t* back into equation (1) to get the normalized (scaled) distance of the point on the curve of maximum excursion from the line through $P_0 P_3$.

$$d_{norm}(v) = s\left(t_{\max}(v)\right), \quad -1 \leq v \leq +1$$

To avoid the computational overhead of evaluating this function, the function was empirically approximated by a quadratic,

$$\tilde{d}_{norm}(v) = 0.07200(v + 3.180556)v + 0.449000$$

The constants were derived such that $\max\{\tilde{d}_{norm}(v) - d_{norm}(v) \mid -1 \leq v \leq +1\}$ is minimized, and $\tilde{d}_{norm}(v) - d_{norm}(v) \geq 0, \quad -1 \leq v \leq +1$, and $\tilde{d}_{norm}(v) = d_{norm}(v)$ at $v = 1$. The constants were chosen such that that $\tilde{d}_{norm}(v)$ is a good estimate (the proportional error is never greater than 1.155%), and never an underestimate. In practical Bézier curves, where the curvature is much larger than the flatness criterion, the value of *v* at the depth in the recursive subdivision when flatness has almost been achieved, the expected value of *v* will be close to 1. This is where the estimating function is most accurate.

The maximum excursions (both actual and estimated) of the curve from a straight line are obtained by dividing these distances by the magnitude of the scale factor used above,

$$
\begin{aligned}
\tilde{d} &= \tilde{d}_{norm} s_i \\
d &= d_{norm} s_i
\end{aligned}
$$

although this evaluation would not actually be needed in any implementation.

3.2 Longitudinal component of the curve

If the curve is “flat” within the given tolerance, we may still need to find the points of maximum excursion of the curve in the longitudinal (i.e., *r*-axis) direction (see Figure 2—the *s*-axis direction has been scaled up for the sake of clarity; $P_{\max}$ is projected onto $P_0 P_3$, the *r*-axis). This situation is usually handled by continued multiple recursions. We will call the point of maximum excursion on the opposite side of P_0 from P_3 (if such a point exists) the minimum point, $P_{\min}$, and the point on the opposite side of P_3 from P_0 (if such a point exists) the maximum point, $P_{\max}$. We need to find either (or both) turning points if they occur outside the segment $\overline{P_0 P_3}$ since the approximating polylines must pass through them. If both these points are inside $\overline{P_0 P_3}$, then the (flat) curve is covered by simply joining P_0 and P_3 directly. Thus, the flattened multiline is passes through a sequence of vertices which may be one of (P_0, P_3), $(P_0, P_{\min}, P_3)$, $(P_0, P_{\max}, P_3)$ (e.g., Figure 3), $(P_0, P_{\min}, P_{\max}, P_3)$, or $(P_0, P_{\max}, P_{\min}, P_3)$ (see below.)

The general parametric equation of the *r*-component of a Bézier curve is,

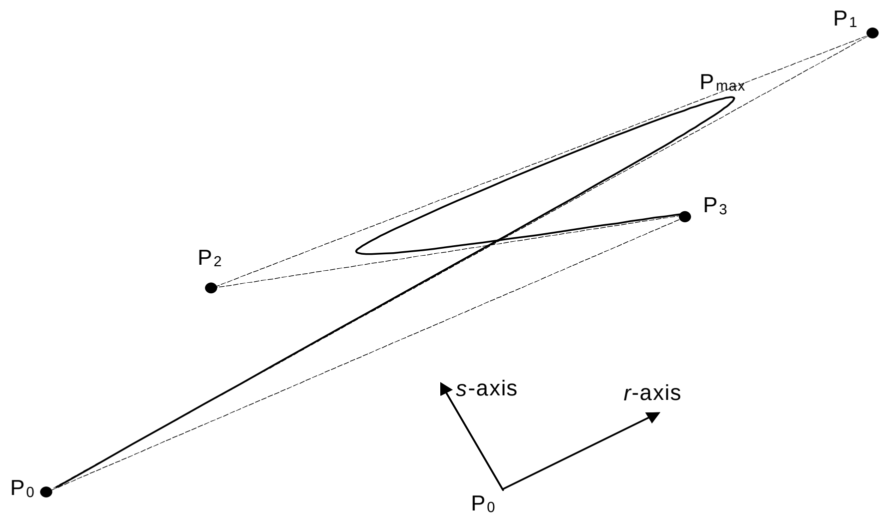

Figure 2 Longitudinal curve approximated by line segments $\overline{P_0 P_{\max}}$ and $\overline{P_{\max} P_3}$ (Think of all points projected onto line through $P_0 P_3$.)

$$r(t) = (1-t)^3 r_0 + 3t(1-t)^2 r_1 + 3t^2(1-t) r_2 + t^3 r_3$$
$$= r_0 + 3(r_1 - r_0)t + 3(r_0 - 2r_1 + r_2)t^2 + (3r_1 - 3r_2 + r_3 - r_0)t^3$$

$$\frac{dr}{dt} = 3(r_1 - r_0) + 6(r_0 - 2r_1 + r_2)t + 3(3r_1 - 3r_2 + r_3 - r_0)t^2$$
$$= (r_1 - r_0) + 2(r_0 - 2r_1 + r_2)t + (3r_1 - 3r_2 + r_3 - r_0)t^2$$
$$= 0$$

$$t = \frac{-2(r_0 - 2r_1 + r_2) \pm 2\sqrt{(r_0 - 2r_1 + r_2) - (r_1 - r_0)(3r_1 - 3r_2 + r_3 - r_0)}}{2(3y_1 - 3y_2 + y_3 - y_0)}$$
$$= \frac{-(r_0 - 2r_1 + r_2) \pm \sqrt{r_1(r_1 - r_2) + r_2(r_2 - r_0) - r_3(r_1 - r_0)}}{3(r_1 - r_2) + r_3 - r_0}$$

Since the r-coordinate as defined above is normalized so that $r = 0$ at P_0 and $r = 1$ at P_3, i.e., $r_0 = 0$, and $r_3 = 1$, then

$$t_{\min} = \frac{(2r_1 - r_2) + \sqrt{r_1(r_1 - r_2 - 1) + r_2{}^2}}{3(r_1 - r_2) + 1}$$
$$t_{\max} = \frac{(2r_1 - r_2) - \sqrt{r_1(r_1 - r_2 - 1) + r_2{}^2}}{3(r_1 - r_2) + 1}$$

(2)

These values of t can be substituted into

$$r(t) = (1-t)^3 r_0 + 3t(1-t)^2 r_1 + 3t^2(1-t) r_2 + t^3 r_3$$
$$= t(3(1-t)(r_1 - t(r_1 - r_2)) + t^2) \quad \text{with } r_0 = 0, \text{ and } r_3 = 1$$

to yield $r_{\min}$ and $r_{\max}$. The coordinates of the turning points (if they exist) are then

$$P_{\min} = \left[P_{0_x} + (P_{3_x} - P_{0_x}) r_{\min}, P_{0_y} + (P_{3_y} - P_{0_y}) r_{\min} \right]$$
$$P_{\max} = \left[P_{0_x} + (P_{3_x} - P_{0_x}) r_{\max}, P_{0_y} + (P_{3_y} - P_{0_y}) r_{\max} \right]$$

The condition that $P_{\min}$ and $P_{\max}$ exist and should be added are as follows:

```
if ($r_1 \le r_2$)
{
   if ($r_1 < 0$) add $P_{\min}$
   if ($r_2 > 1$) add $P_{\max}$
} else
{
   if ($r_1 < 0$) add $P_{\min}$
   else if ($r_2 > 1$) add $P_{\max}$
   else if (discriminator > 0)
   {
      if ($q_{\max} > 1$) add $P_{\max}$
      if ($q_{\min} < 0$) add $P_{\min}$
   }
}
```

where $r_{1,2}$ are the normalized r-coordinates of the inner control points, $q_{\min,\max}$ are the normalized *r*-coordinates of the turning points, and discriminator is the expression under the square root in equation (2).

4 Conclusion.

In conclusion, the new method produces an average of 26% fewer vertices for Bézier segments (measured over a very large test-suite of PostScript pages containing typical vector graphics), while maintaining the prescribed flatness criterion. Curves whose convex hull is narrower than the flatness distance but longer than $\overline{P_o P_3}$ are not recursively generated (producing redundant collinear line segments), but extremum vertices are explicitly calculated. The flattening procedure is faster overall since the number of generated vertices in the flattened curve approximation is smaller. The time to fill areas bounded by Bézier segments will be also reduced since the number of vertices is smaller. Finally, the size of the display list is concomitantly reduced, saving memory resources.

References

[1] Foley, J,D., A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics Principles and Practice*, Addison Wesley, 1995.

[2] Lane, J., and L. Riesenfeld, “A theoretical development for the computer generation of piecewise polynomial surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(1), Jan 1980, 36–46.

[3] Shantz, M., and S. Lien, “Rendering trimmed NURBS with adaptive forward differencing” *SIGGRAPH* 89, 189–198.