Curvature combs and harmonized paths in METAPOST

Linus Romer

Abstract
Most font editors offer curvature-related tools. One of these tools is the visualization of curvature via curvature combs. Another tool is the so-called harmonization, which makes the curvature continuous along paths. An implementation of both tools in METAPOST will be presented. Curvature-optimized paths already play a significant role in METAFONT and METAPOST and therefore example METAPOST paths will be examined for their curvature behavior.

1 Curvature
The curvature in a point of a curve is the inverse of the radius of the osculating circle at this point (depicted here as a “curvature vector” on the opposite side of the radius):

For straight segments, the curvature is constantly zero, since the radius of the osculating circle is infinitely large. Vice versa, the curvature becomes infinitely large when the radius of the osculating circle tends to zero.

2 Curvature combs in METAPOST
We can assemble these curvature vectors into a curvature comb:

The curvature may be additionally mapped to a color and the gaps may be filled (all images are grayscaled for print):

A figure like the above can be achieved with path p; p = ⟨path⟩; comb(p, 300); draw p;

Linus Romer
doi.org/10.47397/tb/44-2/tb137romer-curvetools
The condition \( \text{if } i<25: +1 \text{ fi} \) makes the subpath as large as possible to get better accuracy.

A curvature of 0 is mapped to yellow and an infinitely large curvature is mapped to red. This is done by changing the green value between 1 and 0. If the \( .1 \) is increased, the green value tends faster to 0.

### 3 Harmonize paths in METAPOST

In METAPOST, the code

```plaintext
z0 = (70,60); z1 = (0,30); z2 = (20,0);
draw z0{left} .. z1 .. z2{right};
```

produces the following curve:

![Curvature comb in the preceding picture shows that the curvature at \( z_1 \) indeed is not continuous but only near-continuous. When a user sets the direction in the joining knot, METAPOST has no possibility of optimizing the curvature in the joint and the curvature often changes more abruptly at the joint (see the following picture). This case is frequent in type design because knots at horizontal and vertical extrema are preferred over knots with arbitrary direction.]

Fortunately, a METAPOST path can be modified to a continuous-curvature curve with the `harmonize` macro we'll define later. It moves the joining knot along its tangent:

![Curvature combs and harmonized paths in METAPOST](image-url)
We want to choose $g$ such that the curvature is continuous. So the curvature on both sides of $(g, 0)$ must be equal:

$$\frac{2d}{3g^2} = \frac{2l}{3(i-g)^2}$$

Solving for $g$ and obeying the condition $0 \leq g \leq i$ we get

$$g = \frac{\sqrt{d}}{\sqrt{d} + \sqrt{l}} \cdot i.$$ 

If either $d$ or $l$ is zero, $g = \frac{\sqrt{d}}{\sqrt{d} + \sqrt{l}} \cdot i$ becomes either 0 or $i$. That means the joining knot will become collocated with one of its control points, which generally should be avoided. One reason for this avoidance is that the curvature might become infinitely large:

So, we will not alter the paths at all in the case of either $d$ or $l$ being zero. This case occurs also when a straight line goes over to a curve, which is quite frequent in type design:

When the joining knot is a point of inflection, the curvatures $\frac{2d}{3g^2}$ and $\frac{2l}{3(i-g)^2}$ must have different signs.

Hence, a curvature-continuous solution forces $d = l = 0$, so all control points must lie on one line, as in:

In this situation, one could also satisfy further conditions like the preservation of area. On the other hand, having all four control points on the same line as the two affected cubic Bézier segments is critical. Due to rounding errors, such a conversion may add new points of inflection. So, instead of this, we will only guarantee the absolute value of the curvature to be continuous in the case of points of inflection by moving the joining knot in the same manner as before, between its control points:

Finally, the solution of setting

$$g_{\text{new}} = \begin{cases} g_{\text{old}} & \text{if } d = 0 \text{ or } l = 0, \\ \frac{\sqrt{|d|}}{\sqrt{|d|} + \sqrt{|l|}} \cdot i & \text{else} \end{cases}$$

is chosen here and shall be the definition of harmonization. By harmonization the curvatures at the other segment ends will not change.

We define a corresponding macro `harmonize` that returns a harmonized version of a given path $p$. In the generic case, the tangent in the joining knot is not the $x$-axis (as depicted in the preceding figures), so we calculate $d$ and $l$ as the height to the tangent by cross products.
vardef harmonize expr p =
  save t,u,d,l,n,q; pair t,u,q[];
  n = length p;
  for j = if cycle p: 0 else: 1 fi upto n-1:
    q[j] = point j of p;
    t := unitvector(direction j of p);
    u := unitvector(point j of p - precontrol j of p);
    if eps > abs((u dotprod t) - 1): % smooth
      l := abs(t crossprod (precontrol j+1 of p - point j of p));
      d := abs(t crossprod (postcontrol j-1 of p - point j of p));
      if not ( (l = 0) or (d = 0) ):
        q[j] := if (d = l): .5 else:
          ( sqrt(d) / (sqrt(d) + sqrt(l)) ) fi
          [precontrol j of p,postcontrol j of p];
    fi
  fi
  if not cycle p:
    q[0] = point 0 of p;
    q[n] = point n of p;
  fi
  q[n] % start returned path
  for j = 0 upto n-1: % define new path
    .. controls postcontrol j of p
      and precontrol j+1 of p ..
      if (j = n-1)
        cycle else: q[j+1] fi
    endfor
  enddef;

A mostly equivalent algorithm has been published in [2].

5 Examples of harmonization

Should you use harmonization? At least it does no harm to consider it. Most of the time, the changes are subtle, as in the following bulb terminal:

And sometimes they are less subtle, as in the following calligraphic dots:

After harmonization, the dot has become rather more rounded and may have lost its “personality”.

6 Smoothing out paths even more

Since harmonization does not affect off-curve control points nor the curvatures at other joining knots it can be easily used over several cubic Bézier segments. Nonetheless, harmonized paths normally no longer interpolate the knots they were originally meant to. The author once thought it might therefore be a good idea to leave the joining knots and move the control points instead. Then we not only can make the curvature continuous but also the change of curvature. The curve then becomes something that may be termed “supersmooth”.

However, there are some problems that come with this “supersmoothness”: It might introduce additional points of inflection (see below). Furthermore, this will normally change the curvature at other knots and break curvature continuity there.

References


○ Linus Romer
Kantonsschule Glarus
Winkelstrasse 1
8750 Glarus (Switzerland)