Digital illumination

Dr. Alun Moon
School of Informatics
University of Northumbria
Newcastle upon Tyne, UK
alun.moon@unn.ac.uk

Abstract

Donald Knuth has given us *Digital Typography*, and through METAFONT, *Digital Calligraphy*; this paper explores how these tools can be used for *Digital Illumination*. It follows from my interest as an amateur calligrapher in Celtic artwork. Two examples of my work are in figures 1 and 2. Compare with a sketch of an element from the Lindesfarne Gospels in 3 (Bain, 1989, pg. 67), which is the destination I'm reaching for. This has been a very good exercise in learning to write macros for METAPOST.

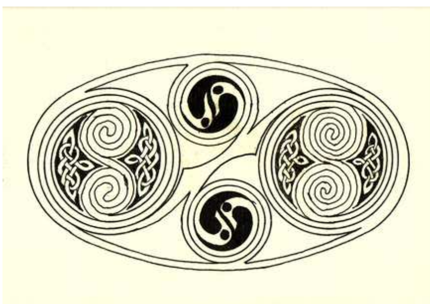


Figure 1: A cartouche with spiral inserts.

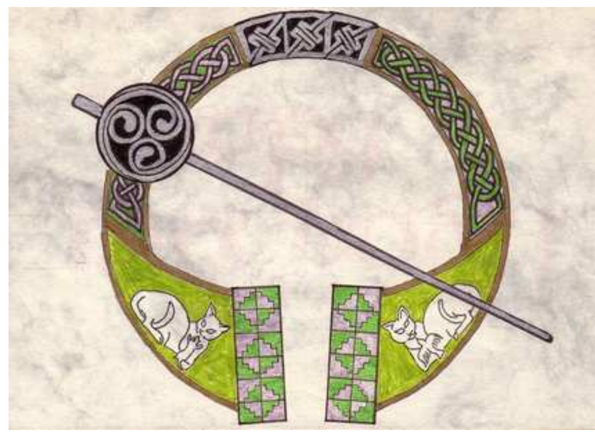


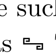
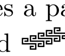
Figure 2: A brooch inspired by elements of the Tara brooch.

Some background

Celtic artwork in Britain covers the period from the 7th century BC through the 7th century AD. During that time examples can be found in stone carving, intricate metalwork, and, towards the end of the period, in illuminated manuscripts. The Celtic monastic scribes produced such masterpieces as *The Book of Kells* and *The Lindesfarne Gospels*. Lindesfarne itself is about 60 miles north of Newcastle; the Gospels are thought to have been produced at Jarrow on the south bank of the Tyne. These show a highly developed artistic style, with very fine, intricate detail. There are three main styles considered here: knots, keypatterns, and spirals.

Knots and keypatterns can be drawn from block elements treated as characters, and large carpet pages built from these standard elements. However, the Celtic scribes show a high degree of geometry and geometrical construction in their work.

A knot can be described as one or more strands that loop, cross and re-cross many times. Can the curves be described and then a METAFONT algorithm used to split them up to generate the under–over–under–... pattern?

A keypattern does have a base form that is then tiled to form the page. The base pattern does have a simple sequence of numbers that define it. A sequence such as $(1, 1, 2, 2, 7, 2, 2, 1, 1)$ gives a pattern such as  This pattern can then be tiled .

Can these simple sequences be used to program METAFONT to generate larger patterns?

Similarly, spiral patterns can be constructed using a pair of compasses. How can METAFONT's geometrical programming be used as a digital pair of compasses to create these beautiful patterns?

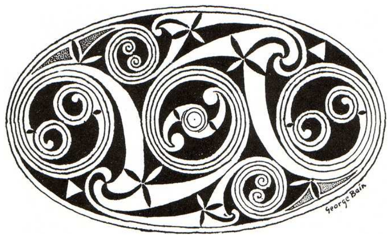


Figure 3: Sketch of a cartouche in the Lindesfarne Gospels (2 in long).

Knotwork

Interlacing patterns of weaving cords is possibly the best known and most widely recognised form of Celtic artwork.

Given a set of curves, once the intersections are known and sorted along the paths, drawing the intersections is easy. For an array of paths $p[]$, two numeric arrays are needed. One holds the times of the intersections $p[]t[]$, the other a count of intersections $p[]t\#$. This way p_1t_3 is the $3^{\rm rd}$ intersection on the $1^{\rm st}$ curve, and $p_3t\#$ is the number of intersections on the $3^{\rm rd}$ curve. The intersections can be found with the `intersectiontimes` operator in METAPOST and METAFONT.

A function `crossings` takes a suffix parameter and a text parameter. The suffix is the path for which the crossings are to be found, and the text is a list of paths to test for (see figure 4).

Intersection times The `intersectiontimes` operator tends to generate points at the beginning of the path. To iterate along the path a series of subpaths are used. Each one starts just past the last intersection (time plus `epsilon`), up to the end of the path.

There is a small problem using subpaths with the `intersectiontimes` operator; the time returned is the time for the subpath. The path $z_1..z_2..z_3..z_4$ has length 4, while the subpath $[.75, 1.25]$ has length 2. Points have been added to the beginning and end where there are not points of the original path. The intersection time on the subpath (t_s) can be converted to a time on the full path as follows:

- if $t_s < 1$, use t_s to interpolate between the beginning of the subpath (a) and the next point on the curve (ceiling of a).
- if $t_s \geq 1$, add it to the last point on the curve before the subpath (floor a).

For a simple knot the global intersection-times for one of the paths is shown in figure 5.

```
vardef crossings@#(text others) =
  save lastpt, tmp;
  p@#t[0]:=0;
  p@#t#:=0;
  forsuffixes $=others:
    numeric lastpt;
    lastpt := epsilon;
    forever:
      numeric tmp;
      (tmp,whatever)=
        subpath (lastpt,length(p@#)-epsilon)
        of p@#
      intersectiontimes p$;
      exitif (tmp<=0);
      p@#t[incr p@#t#] := if(tmp<1):
        tmp[lastpt,ceil(lastpt)]
      else:
        floor(lastpt)+tmp
      fi;
      lastpt := p@#t[p@#t#]+epsilon;
    endfor
  endfor;
  sort.p@#t;
enddef;
```

Figure 4: `crossings` function.

Drawing the crossings The crossings are drawn using the `erase draw` technique, as described in *The METAFONTbook* (pg. 113). The erasing segment is drawn between the midpoints of the sections on either side of the crossing point, the line is then drawn slightly longer. This avoids gaps in the lines where the erasing began.

Examples With the crossing macros, any knotwork pattern can be drawn, as long as there are the paths $p[]$ defined. There is one caveat, each path *must* start so that its first crossing is over the path it crosses. This makes all the crossing times the odd numbers in the array of times.

The trefoil The trefoil is a simple knot using four paths (figure 6). Some people claim it symbolises the Holy Trinity, or wholeness (I like it because it is the motif used for my wedding).

Border A common theme is a knotwork border, a simple example is shown in 7 after (Bain, 1989, pg. 29). Once the paths are specified, application of the `crossings` and `drawcrossings` macros generate the knots.

Better knots Because a circular pen is used for both erasing and drawing the lines, there is a limit to how wide the line can be before the ends of the strokes become visible.

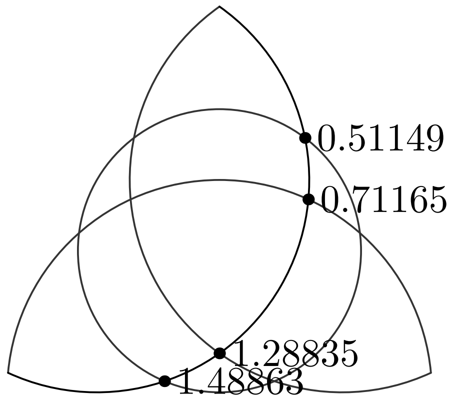


Figure 5: Times for intersection points.

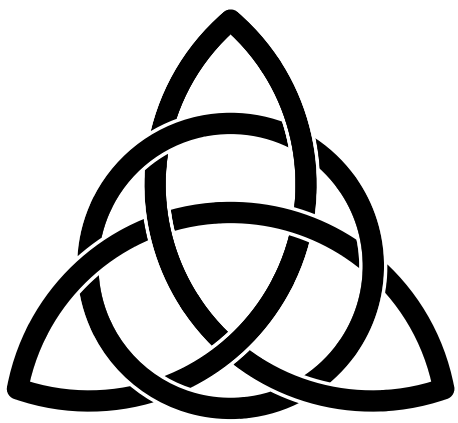


Figure 6: The trefoil.

A better method would be to generate the points that form the end of each stroke. This can be combined with the `penpos` and `penstroke` macros. This requires a little more mathematics. Once the intersection is known, a time on the path is needed to give a point a given distance from the intersection.

Keypatterns

Keypatterns are a common border or filling element. Usually the base element is a C or S spiral, which is then repeated to fill the space. The edges use a separate pattern that fills in around the basic shape.

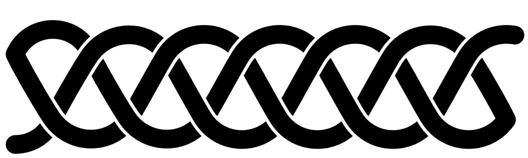


Figure 7: A knotwork border.

Bain (1989) uses a numeric notation to describe the core patterns, the inspiration for one approach.

S-spiral generation A sequence of numbers such as $(1,2,3,4,9,4,3,2,1)$ defines a curve; each number is the length of the segment. Each new segment is drawn at right-angles to the last. Turning anti-clockwise in the first half, where the lengths are increasing. Clockwise in the second, where the lengths decrease.

The macro is shown in figure 8. It maintains a copy of the maximum length drawn, to test whether to turn clockwise or anti-clockwise.

```
def keySspiral(text tail) :=
  begingroup
    save direct,lastpoint,maxlength;
    pair direct,lastpoint;
    direct := up rotated -90;
    lastpoint := origin;
    maxlength := 0;
    origin
    for p=tail: --
      begingroup
        direct := direct
          rotated if (maxlength<=p):
            begingroup maxlength := p;
            90 endgroup
          else:
            -90
          fi;
        lastpoint := lastpoint + direct*p;
        lastlength := p;
        lastpoint
      endgroup
    endfor
  endgroup
enddef;
```

Figure 8: S-spiral generator macro.

Tessellated curves The sequence $(1,2,3,4,8,4,3,2,1)$ gives a curve that tessellates to fill a region (figure 9). Typically a blank border surrounds the keypattern; the `clip` operator in METAPOST serves well to form the border.

Programmed variation A curve that can tessellate in such a way that it interlocks with itself (figure 10) can be generated by the sequence $(1,2,3,4,9,4,3,2,1)$.

With METAFONT we have a powerful programming tool that can vary the pattern as it is drawn, in ways that the seventh century scribes could not easily do. Figure 12 shows the border from figure 10 varying in intensity; figure 11 shows the width of the line varying.

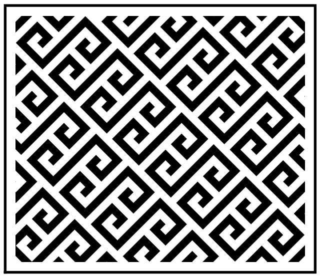


Figure 9: Space-filling keypattern.

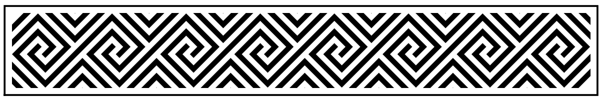


Figure 10: Interlocking keypattern.

Figure 12 is also interesting as the scale of the lengths (and pen width) is 1 pt. It looks fine on a monitor, but may test the limits of a printer. METAFONT allows even finer detail, limited only by the resolution of the printer.

Spirals

Spirals are another signature element of Celtic artwork. Meehan (1993b) shows how the spiral elements can be drawn using two, three or four offset centres and a pair of compasses. METAFONT draws three point paths ($z_0..z_1..z_2$) as close to a circular arc as possible (Knuth, 2000, pg. 128).

Given an initial point, a pair of centres, and a number of turns, the `spiral` macro is a very simple recursive function (figure 13). Although it could be just as simple with a loop, swapping the centres over is easier to do with the recursive call.

Figure 14 shows a cartouche inspired by figure 3. Each pair of spirals is joined by a path connecting the outer points. This path really should follow a common tangent to the two curves forming the outer end of the spiral. A macro is needed to find the points on the two curves giving the common tangent.

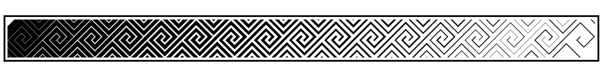


Figure 11: Keypattern varying with line width.

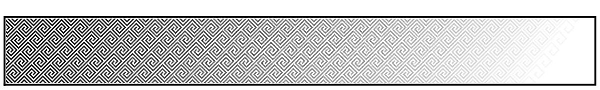


Figure 12: Keypattern varying with colour.

```
def spiral(expr a,b,$)(expr turns) =
  $
   .. $ rotatedaround(a, 90)
   .. $ rotatedaround(a, 180)
  if( turns>1 ):
    & spiral(b,a,
                 $ rotatedaround(a, 180))
                 (turns-1)
  fi
enddef;
```

Figure 13: Spiral macro.

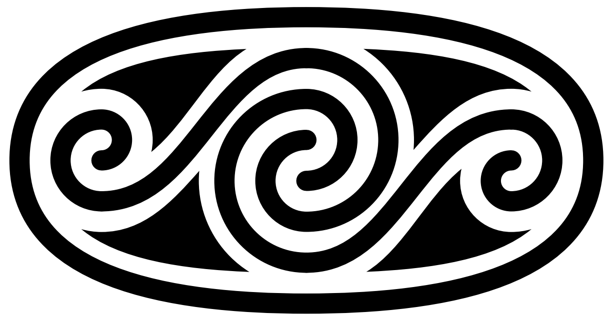


Figure 14: Cartouche inspired by figure 3.

Some resources

Over the years I have found several books to be of use. George Bain (Bain, 1989) is often cited as the key work. He has collected a wide range of material from the Gospels, Book of Kells, jewelry, artifacts and stone carvings. It doesn't go into many of the construction techniques, but is a very good source of inspiration. Aidan Meehan has produced a series of books (Meehan, 1993a; Meehan, 1993b), which give a step-by-step approach and give some good ideas as to the construction of the geometry. Sheila Sturrock's book (Sturrock, 2001) has a different approach to building the keypatterns and clearly shows how the borders develop.

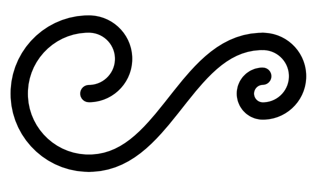


Figure 15: Joined spirals.

Andy Sloss has two books with a radically different technique (Sloss, 1997a; Sloss, 1997b). He enumerates all possible combinations of crossings, characterised by the four entry and exit directions of the two strands. These can then be laid out on a grid. This would be eminently suitable for conversion to a font (assuming enough characters). Perhaps a task for the long winter evenings.

Finally

What new Illumination can be produced by a tool as highly versatile as METAFONT? Can the transformations in Metafont implement conformal mapping and be applied to patterns generated as above? Preliminary experiments suggest it can, but the trick is finding the mapping to use. If the patterns can be described in a parametric form, can an Escher-like tiling be achieved where the pattern changes across the page? Again yes, but finding a shape that scales and still fits together is difficult.

The so-called "Dark Ages" produced a flowering of the work of Celtic scribes, culminating in the "Golden Age" of the Scribes' art. Knuth has given us tools to usher in a Golden Age of Typesetting and Digital Illumination.

References

Bain, George. *Celtic Art: The Methods of Construction*. Constable and Company Ltd, 1989. ISBN 0 09 461830 5.

Knuth, Donald. *The METAFONTbook*. Addison Wesley, 2000.

Meehan, Aidan. *A Beginner'S manual*. Celtic Design. Thames and Hudson, 1993a. ISBN 0 500 27629 3.

Meehan, Aidan. *Spiral Patterns*. Celtic Design. Thames and Hudson, 1993b. ISBN 0 500 27705 2.

Sloss, Andy. *How to Draw Celtic Key Patterns*. Blandford, 1997a. ISBN 0 7137 2652 0.

Sloss, Andy. *How to Draw Celtic Knotwork*. Blandford, 1997b. ISBN 0 7137 2492 7.

Sturrock, Sheila. *Celtic Spirals and Other Designs*. Guild of Master Craftsman Publications Ltd, 2001. ISBN 1 86108 159 6.