

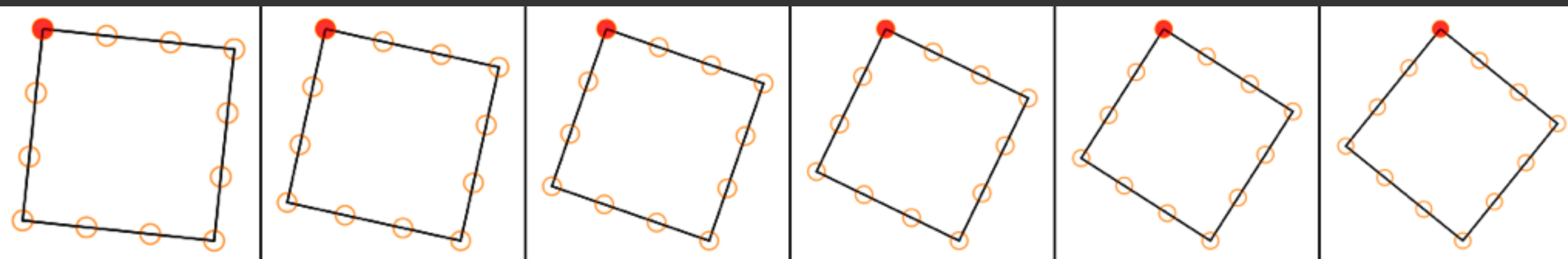
CGPathRef & friends

Leonhard Lichtschlag

github.com/lichtschlag/Cipher
@lichtschlag

CAShapeLayer

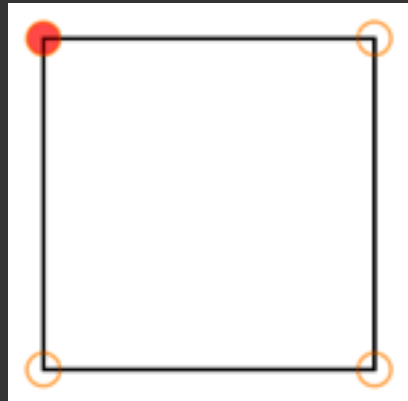
- Renders a CGPathRef, not a CGImageRef
- **path** property is animatable



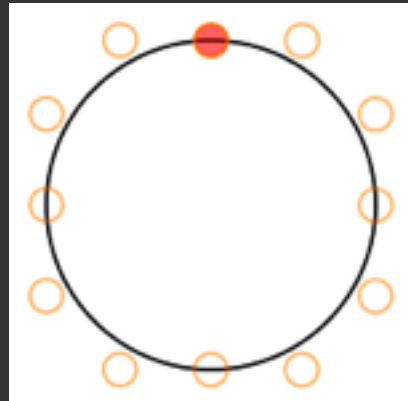
- *“If the two paths have a different number of control points or segments the results are **undefined**”*

Interpolation by hand

- from

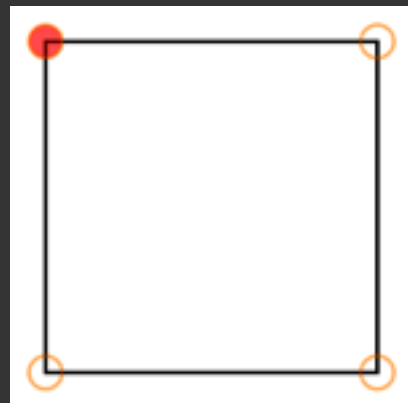


to

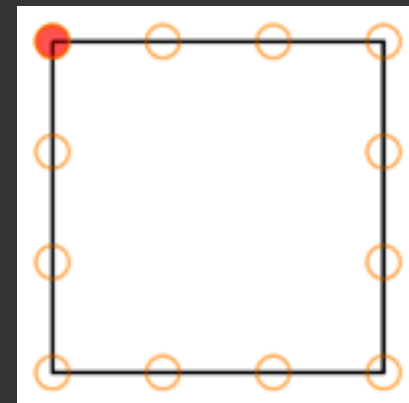


?

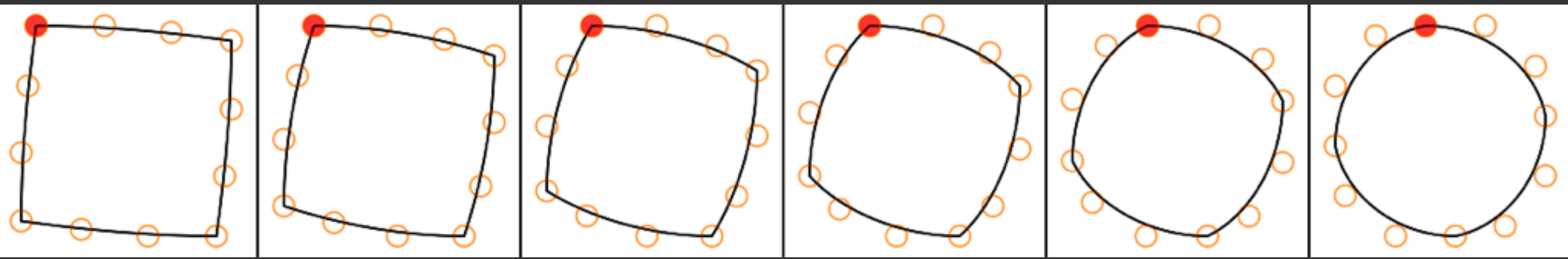
- first normalise



to



- then interpolate all control points



CGPathRef

- cannot count elements
- CGMutablePathRef is write and draw only
- CGPathElementAtIndex: does not exist
- CGPathRemoveElementAtIndex: does not exist
- Well, there is `CGPathApply()`

CGPathApply

```
void CGPathApply (
    CGPathRef path,
    void *info,
    CGPathApplierFunction function
);
```

```
void CGPathElementsCount(void *info,
                        const CGPathElement *element)
{
    NSUInteger *numberOfElementsSoFar
        = (NSUInteger *)info;
    CGPathElementType currentPointType = element->type;

    if (currentPointType != kCGPathElementCloseSubpath)
        (*numberOfElementsSoFar)++;
}
```

```

+ (UIBezierPath *) bezierPathByConvertingPathToCurves:(UIBezierPath *)basePath;
{
    CGMutablePathRef newPath = CGPathCreateMutable();

    [basePath enumeratePathElementsUsingBlock:^(const CGPathElement *element)
    {
        // . . .
        if (currentPointType == kCGPathElementAddLineToPoint)
        {
            CGPoint beginPoint = CGPathGetCurrentPoint(newPath);
            CGPoint endPoint = element->points[0];
            CGPoint controlPoint1 = CGPointMake( beginPoint.x +
                (endPoint.x-beginPoint.x) * 1.0/3.0,
                beginPoint.y +
                (endPoint.y-beginPoint.y) * 1.0/3.0);
            CGPoint controlPoint2 = CGPointMake( beginPoint.x +
                (endPoint.x-beginPoint.x) * 2.0/3.0,
                beginPoint.y +
                (endPoint.y-beginPoint.y) * 2.0/3.0);

            CGPathAddCurveToPoint(newPath, NULL,
                controlPoint1.x, controlPoint1.y,
                controlPoint2.x, controlPoint2.y,
                endPoint.x, endPoint.y);
        }
        // . . .
    }];

    UIBezierPath *result = [UIBezierPath bezierPathWithCGPath:newPath];
    CGPathRelease(newPath);
    return result;
}

```

CGGlyph

- Text typesetting is complicated...
- In short:
 1. Text is broken in to **CTLineRefs** at the border of the container.
 2. Each line is broken into multiple **CTRunRefs** at the points where the text attributes change.
 3. Each run contains an array of **CTGlyphs**. A glyph is position & path.

SVGKit

github.com/SVGKit/SVGKit

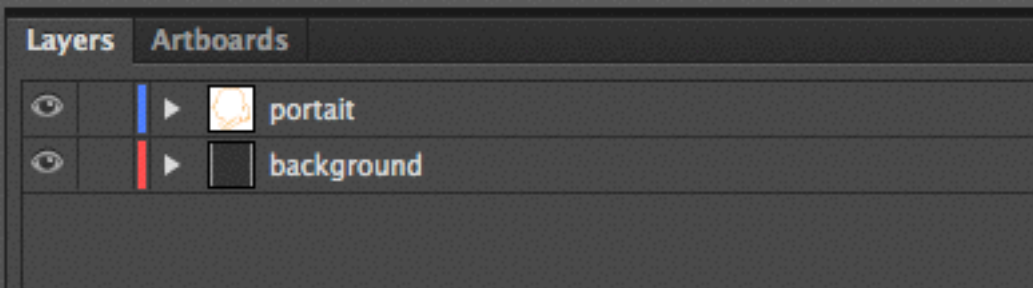


```
<?xml version="1.0" encoding="utf-8"?>
  <!-- Generator: Adobe Illustrator 16.0.4, SVG Export Plug-In . SVG Version: 6.00
  Build 0) -->
  <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/
  Graphics/SVG/1.1/DTD/svg11.dtd">
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://
  www.w3.org/1999/xlink" x="0px" y="0px"
    width="240.96px" height="264.24px" viewBox="0 0 240.96 264.24"
    enable-background="new 0 0 240.96 264.24" xml:space="preserve">

    <g id="background">
      <rect id="background_1" x="0" y="-0.1" fill="#343434"
        width="240.961" height="264.24"/>
    </g>

    <g id="portait">
      <path fill="none" stroke="#FF9300" stroke-miterlimit="10" d="..." />
      <path fill="none" stroke="#FF9300" stroke-miterlimit="10" d="..." />
      <path fill="none" stroke="#FF9300" stroke-miterlimit="10" d="..." />
    </g>

  </svg>
```




```
+ (CipherStroke *) strokeForSVGFileNamed:(NSString *)fileName
    idName:(NSString *)idName
{
    SVGKImage *testDocument =
    [SVGKImage imageNamed:[fileName stringByAppendingPathExtension:@"svg"]];
    Element *group = [testDocument.DOMTree getElementById:idName];
    NodeList *allPathElements = [group getElementsByTagName:@"path"];

    // merge all paths in the file into one
    CGMutablePathRef constructedPath = CGPathCreateMutable();
    for (int i = 0; i < allPathElements.length; i++)
    {
        SVGPathElement *anElement = (SVGPathElement *)[allPathElements item:i];
        CGPathRef aPath = [anElement pathForShapeInRelativeCoords];
        CGPathAddPath(constructedPath, NULL, aPath);
    }

    UIBezierPath *completePath =
        [UIBezierPath bezierPathWithCGPath:constructedPath];

    // . . .
}
```

faster drawing

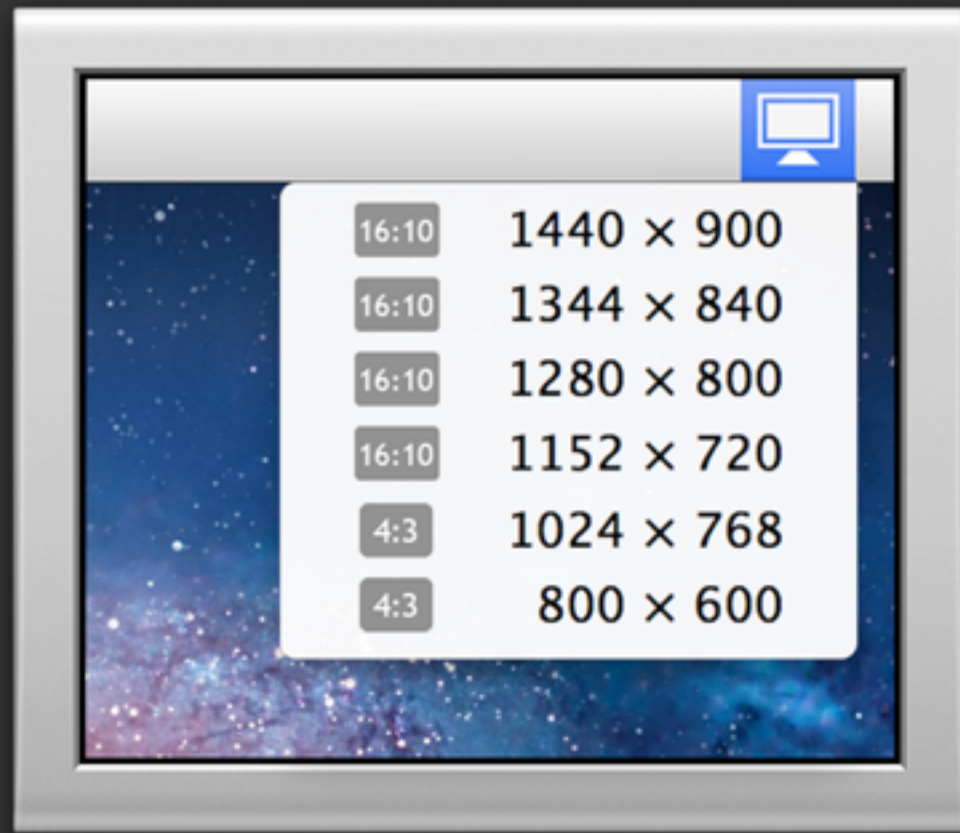
`CAShapeLayer` – good-looking but slow

`Quartz` – worse-looking and also slow

`OpenGL` – fast but cumbersome

`SpriteKit` – no initial success, but I have hope

Project available at github.com/lichtschlag/Cipher



Also, download Thorsten's and my App, [Display Menu](#) (Mac App Store) give it a good rating.