THE UNIVERSITY OF WESTERN AUSTRALIA

Achieve International Excellence

Number Representations and Precision in Vector Graphics

Implementation of an Arbitrary Precision SVG Viewer

Sam Moore

Supervisors: Tim French, Rowan Davies

October 9, 2014

Contents

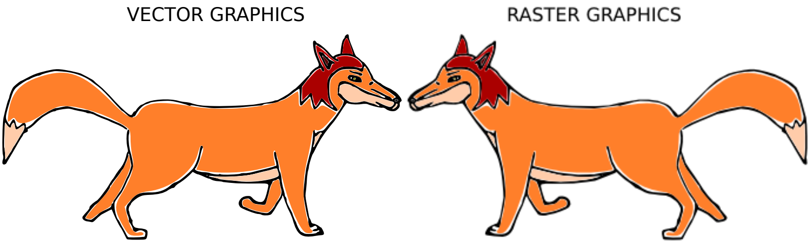
Summary

- Vector graphics allow scaling but not arbitrary scaling
- We implemented a vector graphics viewer that does allow arbitrary scaling
- ... but it will take an arbitrary amount of time

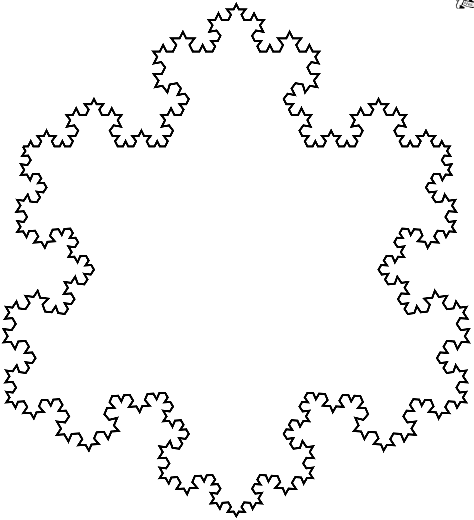
Graphics Formats

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

- Document formats (eg: PDF and SVG) are formats for vector graphics
- Vector graphics scale better than raster graphics



Why is there a zoom limit?



Why is there a zoom limit?

- SVG, PostScript, PDF specify IEEE-754 *single* floating point number representations
- Range of values: $\approx 3 \times 10^{-38} \rightarrow 3 \times 10^{+38}$
- Rough Floating Point Definition[1]:

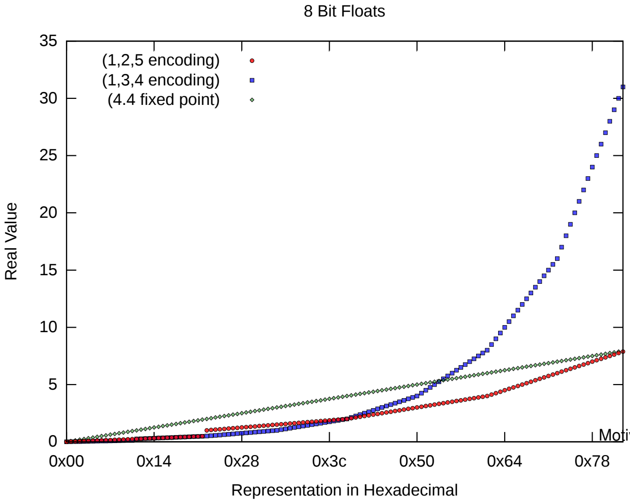
$$X = m \times 2^{E} \tag{1}$$

- m and E are encoded in a *fixed length* string of bits
- Floating Point $\approx$ Scientific Notation for computers

[1] IEEE-754 is more complicated

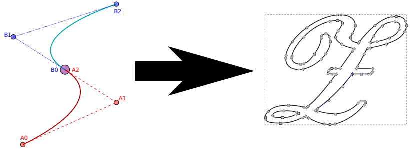
Visualisation of Floats

- With total length of *m* and *E* limited to 7 bits (1 sign bit)
- Operations are inexact (in general)



Structure of Vector Graphics

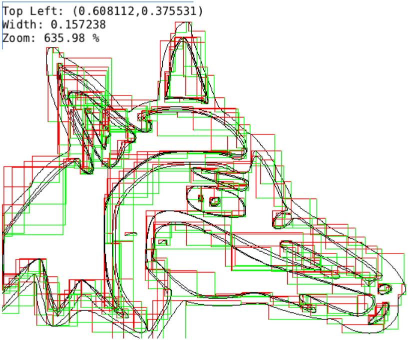
- Bézier Curve (Quadratic or Cubic Parametric Polynomial)
- Path of Bézier Curves → Shapes (with fill)
- Shapes include font glyphs, like this $\mathscr{L}$



Structure of Vector Graphics III

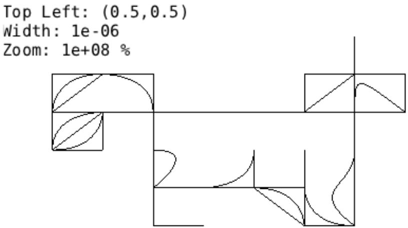
THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

- Rectangles show individual Béziers forming outline of the Fox



Floating point calculations go wrong

- Scaled to 1×10^{-6}, the fox is very sick



Arbitrary Precision Rationals

$$Q = \frac{N}{D} \quad (2)$$

- N and D are arbitrary precision integers

$$N = \sum_{i=0}^{S} d_i \beta^i \quad (3)$$

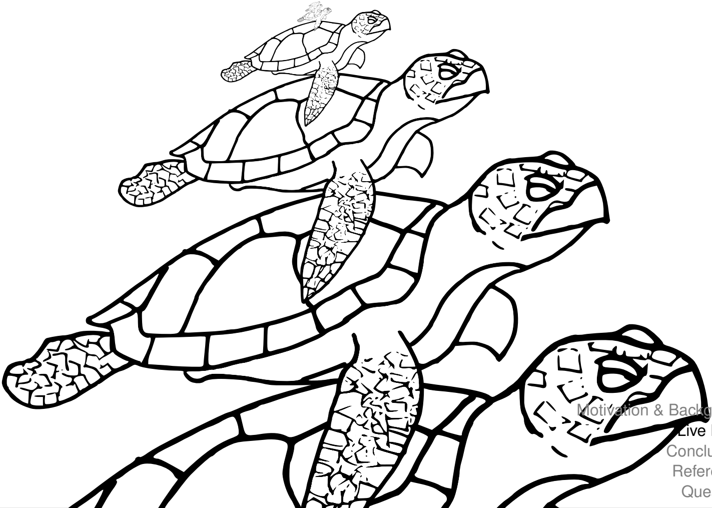
- d_i are fixed size integers, $\beta = 2^{64}$
- Size S grows as needed
- Operations are always exact
- Implemented by GNU Multiple Precision Library

Replace floats with rationals?

- Rationals are *slow*
- Screen coordinates always in range $0 \rightarrow 1$
- Introduce intermediate coordinate system
  - Many Béziers contained in a Path
  - Use Rationals for bounds of the Path
  - Use floats to transform Bézier coordinates

Live Demo

- We can import standard SVGs wherever we want
- If we are willing to wait long enough
- "... But, asks the scientist, what does that turtle stand on? To which the lady triumphantly answers: 'You're very clever, young man, but it's no use – it's turtles all the way down!'."



Conclusions

The University of Western Australia
Achieve International Excellence

- What we have done?
  - Implemented a basic SVG viewer
  - Demonstrated how precision affects rendering vector graphics
  - Using GMP rationals, demonstrated the ability to render SVGs scaled to an arbitrary position in a document
- Possible future work
  - Implement more of the SVG standard
  - Trial alternative number representations
  - Allow for saving and loading SVGs with arbitrary precision

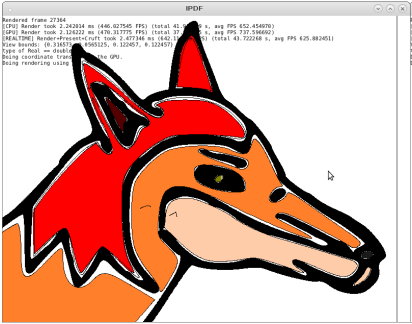
References & More information

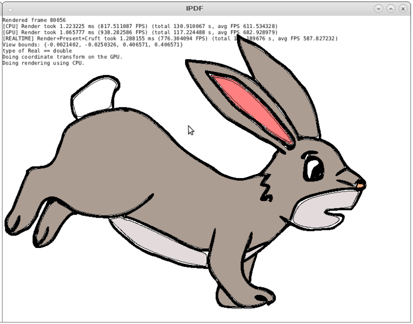
THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

- Work on SVG viewer collaborative with David Gow
  - See David Gow's presentation about Quadtrees
- Muller et al, *Handbook of Floating Point Arithmetic*,
- Hearn, Baker *Computer Graphics*
- Kahan et al, *IEEE-754* (1985 and 2008 revision)
- Dahlstóm et al, *SVG WC3 Recommendation 2011*
- Grunland et al, *GNU Multiple Precision Manual 6.0.0a*
- Kahan's website
  `http://http.cs.berkeley.edu/~wkahan`

Q: Why don’t you have colour?

- We do![2]
- A complete implementation of SVG is “future work”





[2] If you are willing to wait long enough

Q: Why not just use doubles?

- Any fixed precision format will still give inexact results
- But the inexact results will appear slower

Q: Arbitrary precision floats?

$$X = m \times 2^E \tag{4}$$

- m and E are of arbitrary size
- Implemented by MPFR or GMP
- Difficulties:
  - Need to manually set precision (size) of m
  - Some operations require infinite precision:

$$\frac{1}{3} = 0.33333333333333333333333\ldots \times 10^0 \tag{5}$$

  - How do you choose when to increase precision?

Floating Point calculations go wrong

- Plank Length: 1.61×10^{-35} metres $> 3 \times 10^{-38}$
- Size of Universe: 4.3×10^{26} metres $<< 3 \times 10^{38}$
- Why isn't this good enough for 1×10^{-6}

Floating point calculations go wrong

- Transforming from document $(x, y) \rightarrow$ screen (X, Y)
- View is at (v_x, v_y) in document, has dimensions (v_w, v_h)

$$X = \frac{x - v_x}{v_w}, \qquad Y = \frac{y - v_y}{v_h} \tag{6}$$

- Division by $v_w \approx 10^{-6}$ increases the error due to $x - v_x$
- Using *double* precision, render correctly down to $v_w \approx 10^{-37}$