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

Colleagues: David Gow

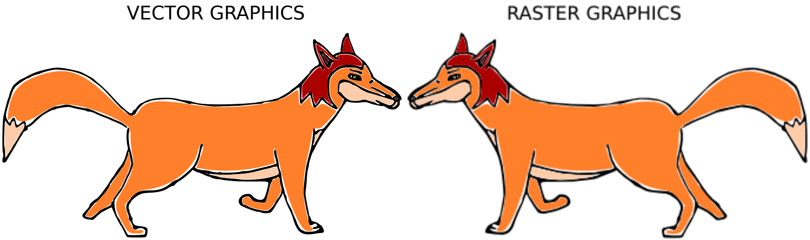
October 14, 2014

Summary

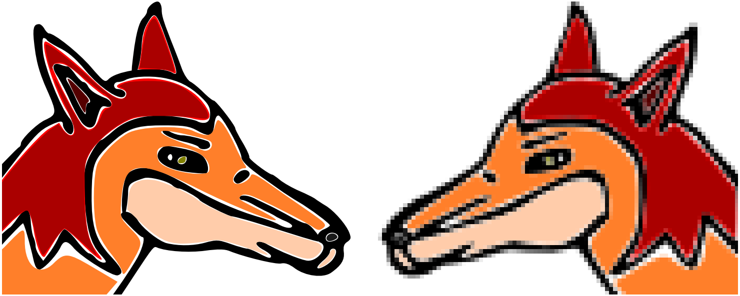
- Vector graphics allow detail to be scaled but not by an arbitrary amount
- We've implemented a vector graphics viewer which does allow arbitrary scaling

Graphics Formats

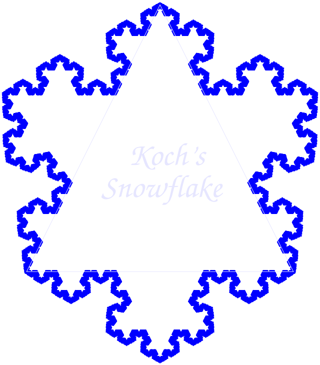
- Document formats (eg: PDF and SVG) are formats for vector graphics
- Vector graphics scale better than raster graphics



- Vector graphics scale better than raster graphics



Is there a zoom limit?



Is there a zoom limit?

- SVG, PostScript, PDF specify IEEE-754 *single* floating point number representations
- Range of values: $\approx 3 \times 10^{-38} \rightarrow 3 \times 10^{+38}$
- Bigger than size of Universe?

Floating Point Definition

- Rough Floating Point Definition[1]:

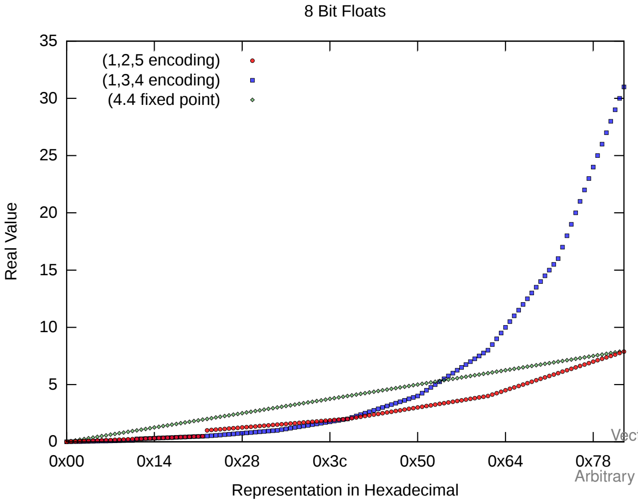
$$X = m \times 2^E \tag{1}$$

- 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

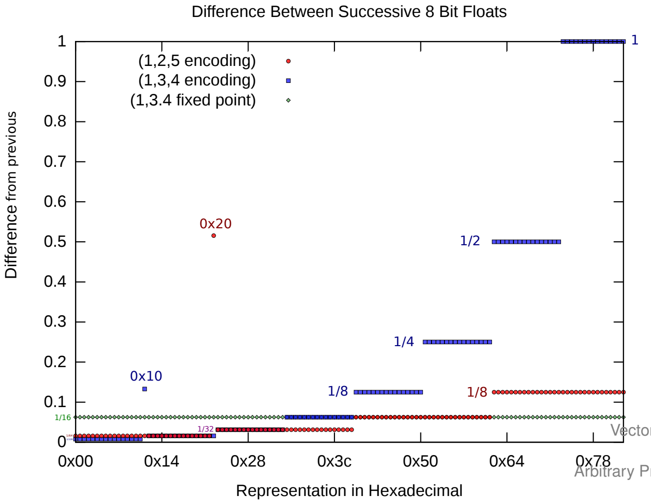
- With total length of *m* and *E* limited to 7 bits (1 sign bit)
- Operations are inexact (in general)



Visualisation of Floats II

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

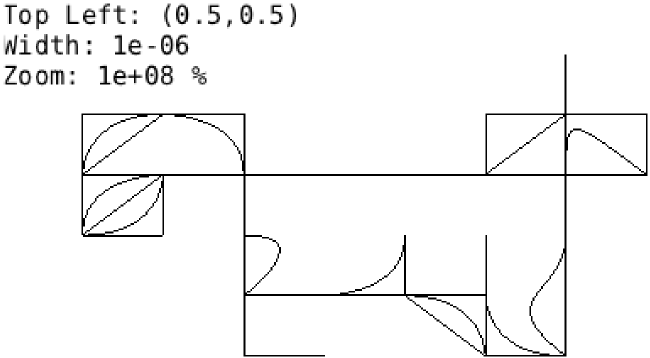
- Difference between successive floats
- Further from origin $\Longrightarrow$ less precision



Motivation
Floating Point
Vector Graphics Viewer
Precision Issues
Arbitrary Precision Rationals
Demonstration
Conclusions

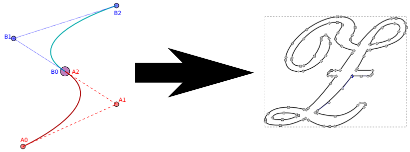
Precision is limited

- Eg: Fox scaled to width of $\approx 10^{-6}$ viewed at zoom of $\approx 10^{8}$
- (Outline should look like images in first slide)



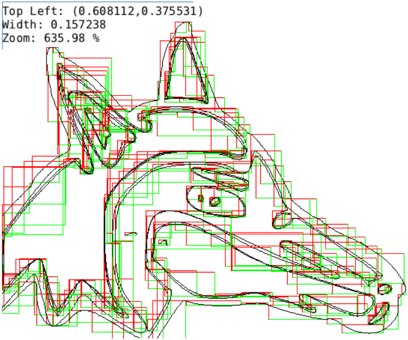
Structure of Vector Graphics

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



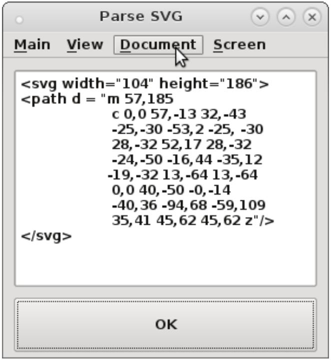
Structure of Vector Graphics II

- Upload *bounding rectangles* of individual objects to renderer (OpenGL)
- Rectangles show individual Béziers forming outline of the Fox



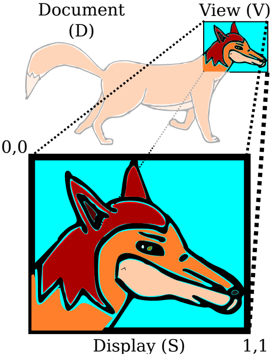
Vector Graphics Viewer Features

- GPU (OpenGL) or CPU (custom) rendering
- Import SVG into current View location
- Control through scripts (Python) or Qt4 GUI



Viewing Vector Graphics

- Tranform coordinates in document → display



Document via View to Display (S)

$$(X, Y) \overset{V}{\longmapsto} (S_X, S_Y)$$

$$S_X = \frac{X - V_x}{V_w} \quad S_Y = \frac{Y - V_y}{V_h}$$

Floating point calculations go wrong

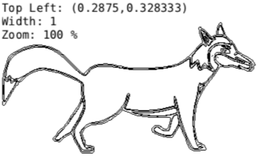
- Example: Insert objects at very small scale

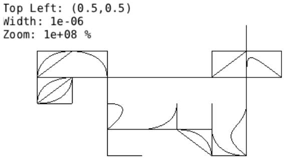
$$X = V_w \times O_x + V_x$$

Round to floats near Vx

$$S_X = \frac{X - V_x}{V_w}$$

Increases error in X





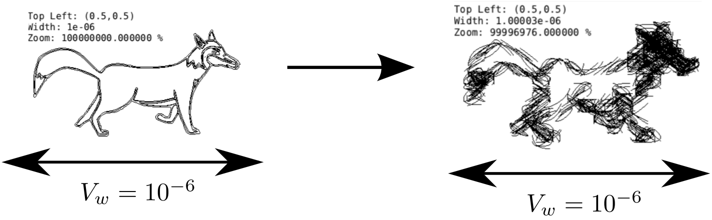
$V_w = 1$ ✓

$V_w = 10^{-6}$ ✗

Reducing error

- Don't apply view transformation directly
- Store object bounds relative to the display
- When modifying the view, modify object bounds
- Detail inserted into the view looks good, But...

Zoom out by a large amount then back in



Inserted SVG

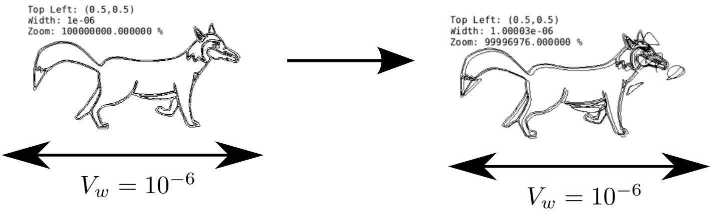
Accumulated Errors

Cumulated Errors with intermediate coord system

THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

- Apply transformations to Paths not individual Béziers
- Paths render correctly, but drift appart

Zoom out by a large amount then back in



Inserted SVG

Drifting of Paths

We only need to transform the Paths with increased precision

Arbitrary Precision Rationals

The University of Western Australia
Achieve International Excellence

$$Q = \frac{N}{D} \tag{2}$$

- N and D are arbitrary precision integers

$$N = \sum_{i=0}^{S} d_i \beta^i \tag{3}$$

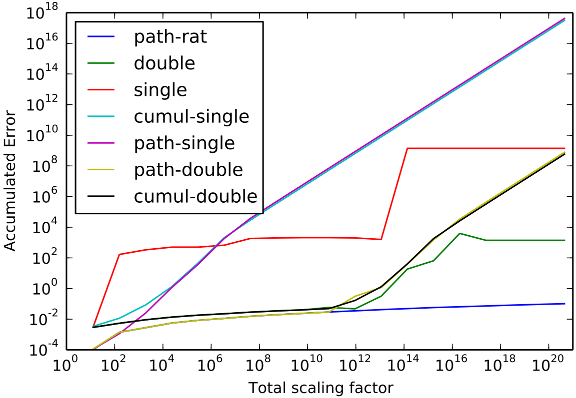
- d_i are fixed size integers, $\beta = 2^{64}$
- **Size S grows as needed**
- Operations are always exact
- Implemented by GNU Multiple Precision Library

Use Rationals to represent Path Coordinates

- Can move view to arbitrary point
- Insert detail (ie: Test SVG image) in Display coordinates
- Move view to another arbitrary point
- Move view back
- Detail is unchanged

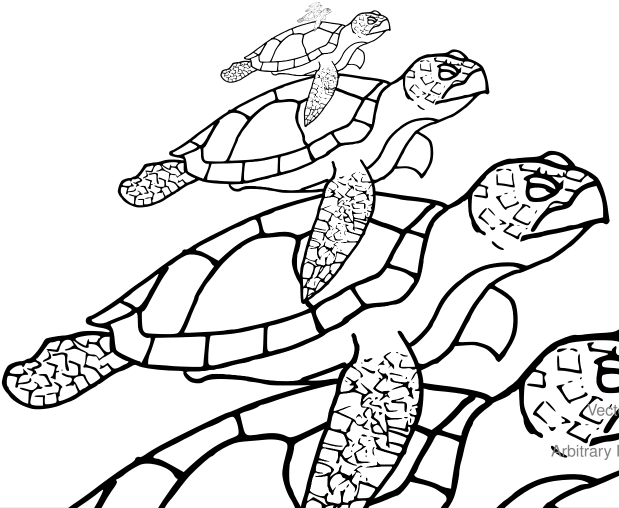
Quantitative Results

- Invariance of grid of lines after scaling



Demonstration

- 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!’.“



What was 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

Future work

- Implement more of the SVG standard
- Trial alternative number representations
- Allow for saving and loading
- Optimisations, eg: clip objects that are not visible
- Compile for Windows (MinGW)

Acknowledgements

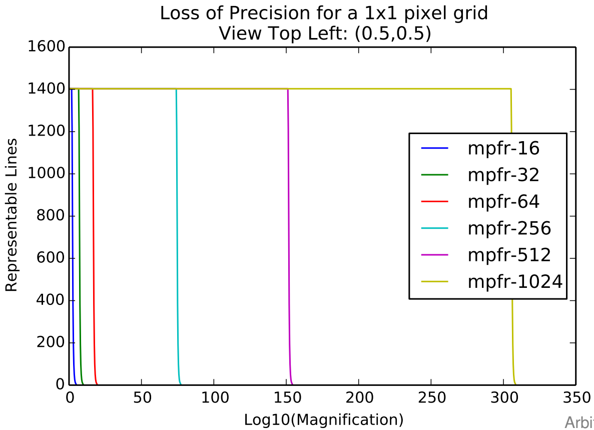
- Work on SVG viewer collaborative with David Gow
  - See David Gow's presentation about Quadtrees
- Supervisors: Tim French and Rowan Davies

References & More information

- 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 not just increase float precision?

- GPU uses singles anyway
- Can use CPU for bounds transforms
- But eventually lose precision for any *fixed* precision float



Q: Arbitrary precision floats?

$$X = m \times 2^E \quad (4)$$

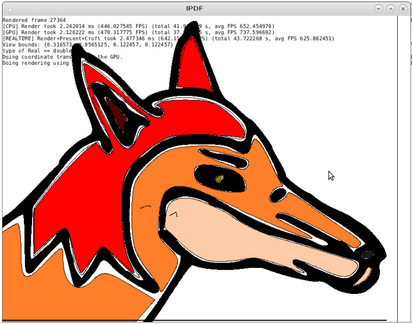
- m and E are of arbitrary size
- Implemented by MPFR (based on GMP)
- Difficulties:
  - Need to manually set precision (size) of m
  - Some operations require infinite precision:

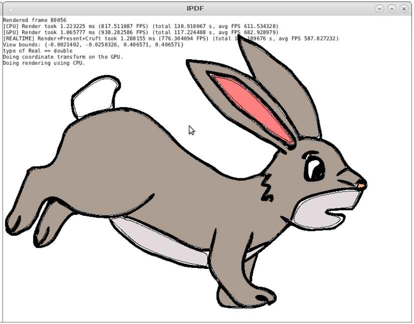
$$\frac{1}{3} = 0.33333333333333333333333\ldots \times 10^0 \quad (5)$$

  - How do you choose when to increase precision?
  - GMP Rational implementation automatically increases size, but MPFR floats do not

Q: Why don't you have colour?

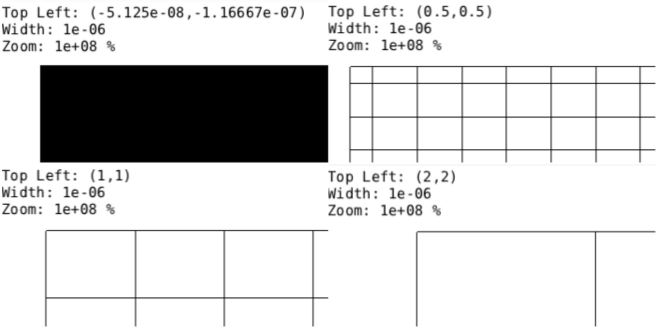
- We do![2]
- A complete implementation of SVG is "future work"



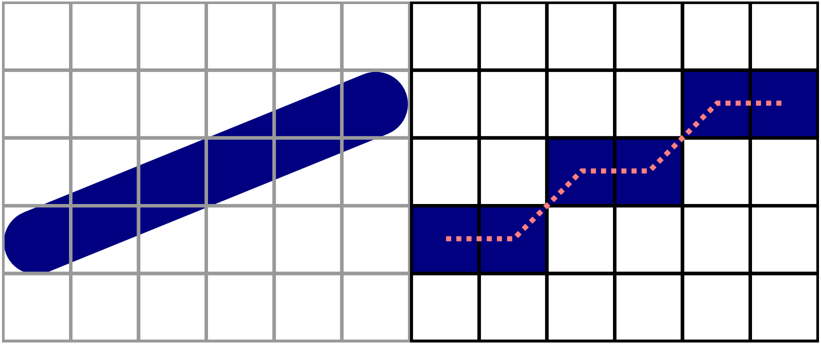


[2] If you are willing to wait long enough

Quantitative?



Bresenham and Wu



Bonus: IEEE-754 on GPUs

- Inconsistent behaviour of calculations on different GPUs
- Eg: $x^2 + y^2 < 1$ (shading a circle) zoomed in on the edge.