B.Eng. Final Year Project

Literature Notes

Sam Moore, David Gow
Faculty of Engineering, Computing and Mathematics, University of Western Australia
March 2014

Contents

1 Postscript Language Reference Manual[1]

Adobe's official reference manual for PostScript.

It is big.

2 Portable Document Format Reference Manual[2]

Adobe's official reference for PDF.

It is also big.

3 Portable Document Format (PDF) — Finally...[3]

This is not spectacularly useful, is basically an advertisement for Adobe software.

Intro

- Visual communications has been revolutionised by computing
- BUT there have always been problems in exchanging formats
- Filetypes like text, rich text, IGES, DXF, TIFF, JPEG, GIFF solve problems for particular types of files only
- PDF solves everything for everyone; can include text, images, animation, sound, etc

PDF Features

- Raster Image Process (RIP) — For printing (presumably also displaying on screen)
- Originally needed to convert to PS then RIP, with PS 3 can now RIP directly.
- Reduced filesize due to compression
- Four major applications - Stoy 1999[**?**]
  1. Download files from internet
  2. Files on CDs
  3. Files for outputing to printers
  4. Conventional [commercial scale?] printing
- List of various (Adobe) PDF related software
  - Includes software for PS that converts to/from PDF
  - So PS was obviously pretty popular before PDF
- Can Optimize for screen/printer [not clear how]
- Can compress for size

4 Pixels or Perish [4]

"The art of scientific illustration will have to adapt to the new age of online publishing" And therefore, JavaScript libraries (D^3) are the future.

The point is that we need to change from thinking about documents as paper to thinking of them as pixels. This kind of makes it related to our paper, because it is the same way we are justifying our project. It does mention precision, but doesn't say we need to get more of it.

I get the feeling from this that Web based documents are a whole bunch of completely different design philosophies hacked together with JavaScript.

This paper uses Metaphors a lot. I never met a phor that didn't over extend itself.

Intro

- Drawings/Pictures are ornaments in science but they are not just ornamental
- Processes have changed a lot; eg: photographic plates → digital images
- "we are about to turn the page — if not close the book — on yet another chapter in publishing history." (HO HO HO)
- It would be cool to have animated figures in documents (eg: Population pyramid; changes with time); not just as "supplements"
- In the beginning, there was PostScript, 1970s and 1980s, John Warnock and Charles Geschke, Adobe Systems
- PS is a language for vector graphics; objects are constructed from geometric primitives rather than a discrete array of pixels
- PS is a complete programming language; an image is also a program; can exploit this to control how images are created based on data (eg: Faces)
- PDF is "flattened" PS. No longer programable. Aspires to be "virtual paper".
- But why are we using such powerful computing machines just to emulate sheets paper? (the author asks)

Web based Documents

- HTML, CSS, JavaScript - The Axis of Web Documents
  - HTML - Defines document structure
  - CSS - Defines presentation of elements in document
  - JavaScript - Encodes actions, allows dynamic content (change the HTML/CSS)
- `<canvas>` will let you draw anything (So in principle don't even need all of HTML/CSS)
  - Not device independent
  - "Coordinates can be specified with precision finer than pixel resolution" **(TODO: Investigate this?)**
  - JavaScript operators to draw things on canvas are very similar to the PostScript model
- SVG — Same structure (Document Object Model (DOM)) as HTML
  - "Noun language"
  - Nouns define lines/curves etc, rather than paragraphs/lists
  - Also borrows things from PostScript (eg: line caps and joints)
  - IS device independent, "very high precision" **(TODO: Investigate)**
  - JavaScript can be used to interact with SVG too
- D^3 (Data Driven Documents) - A JavaScript library
  - Idea is to create or modify elements of a DOM document using supplied data
  - `https://github.com/mbostock/d3/wiki`
- We are in a new Golden Age of data visualisation
- Why do we still use PDFs?
  - PDFs are "owned" by the author/reader; you download it, store it, you can print it, etc
  - HTML documents are normally on websites. They are not self contained. They often rely on remote content from other websites (annoying to download the whole document).
- **Conclusion** Someone should open up PDF to accept things like D^3 and other graphics formats (links nicely with [5])
- Also, Harry Potter reference

5 Embedding and Publishing Interactive, 3D Figures in PDF Files[5]

- Linkes well with [4]; I heard you liked figures so I put a figure in your PDF
- Title pretty much summarises it; similar to [4] except these guys actually did something practical

6 27 Bits are not enough for 8 digit accuracy[?]

Proves with maths, that rounding errors mean that you need at least q bits for p decimal digits. $10^p < 2^{q-1}$

- Eg: For 8 decimal digits, since $10^8 < 2^{27}$ would expect to be able to represent with 27 binary digits
- But: Integer part requires digits bits (regardless of fixed or floating point represenetation)
- Trade-off between precision and range
  - $9000000.0 \rightarrow 9999999.9$ needs 24 digits for the integer part $2^{23} = 83886008$
- Floating point zero = smallest possible machine exponent
- Floating point representation:

$$y = 0.y_1y_2...y_q \times 2^n$$

- Can eliminate a bit by considering whether $n = -e$ for $-e$ the smallest machine exponent (???)
  - Get very small numbers with the same precision
  - Get large numbers with the extra bit of precision

7 What every computer scientist should know about floating-point arithmetic[6]

- Book: *Floating Point Computation* by Pat Sterbenz (out of print... in 1991)
- IEEE floating point standard becoming popular (introduced in 1987, this is 1991)
  - As well as structure, defines the algorithms for addition, multiplication, division and square root
  - Makes things portable because results of operations are the same on all machines (following the standard)
  - Alternatives to floating point: Floating slasi and Signed Logarithm (TODO: Look at these, although they will probably not be useful)
- Base β and precision p (number of digits to represent with) - powers of the base can be represented exactly.
- Largest and smallest exponents e_{min} and e_{max}
- Need bits for exponent and fraction, plus one for sign
- "Floating point number" is one that can be represented exactly.
- Representations are not unique! $0.01 \times 10^1 = 1.00 \times 10^{-1}$ Leading digit of one $\implies$ "normalised"
- Requiring the representation to be normalised makes it unique, **but means it is impossible to represent zero**.
  - Represent zero as $1 \times \beta^{e_{min}-1}$ - requires extra bit in the exponent
- **Rounding Error**
  - "Units in the last place" eg: 0.0314159 compared to 0.0314 has ulp error of 0.159
  - If calculation is the nearest floating point number to the result, it will still be as much as 1/2 ulp in error
  - Relative error corresponding to 1/2 ulp can vary by a factor of β "wobble". Written in terms of ϵ
  - Maths $\implies$ **Relative error is always bounded by** $\epsilon = (\beta/2)\beta^{-p}$
  - Fixed relative error $\implies$ ulp can vary by a factor of β . Vice versa
  - Larger $\beta \implies$ larger errors
- **Guard Digits**
  - In subtraction: Could compute exact difference and then round; this is expensive

  - Keep fixed number of digits but shift operand right; discard precision. Lead to relative error up to $\beta - 1$
  - Guard digit: Add extra digits before truncating. Leads to relative error of less than 2ϵ. This also applies to addition
- **Catastrophic Cancellation** - Operands are subject to rounding errors - multiplication
- **Benign Cancellation** - Subtractions. Error $< 2\epsilon$
- Rearrange formula to avoid catastrophic cancellation
- Historical interest only - speculation on why IBM used $\beta = 16$ for the system/370 - increased range? Avoids shifting
- Precision: IEEE defines extended precision (a lower bound only)
- Discussion of the IEEE standard for operations (TODO: Go over in more detail)
- NaN allow continuing with underflow and Infinity with overflow
- "Incidentally, some people think that the solution to such anomalies is never to compare floating-point numbers for equality but instead to consider them equal if they are within some error bound E. This is hardly a cure all, because it raises as many questions as it answers." - On equality of floating point numbers

References

[1] Adobe Systems Incorporated. *PostScript Language Reference*. Addison-Wesley Publishing Company, 3rd edition, 1985 - 1999.

[2] Adobe Systems Incorporated. *PDF Reference*. Adobe Systems Incorporated, 6th edition, 2006.

[3] Michael A. Wan-Lee Cheng. Portable document format (pdf) – finally, a universal document exchange technology. *Journal of Technology Studies*, 28(1):59 – 63, 2002.

[4] Brian Hayes. Pixels or perish. *American Scientist*, 100(2):106 – 111, 2012.

[5] David G. Barnes, Michail Vidiassov, Bernhard Ruthensteiner, Christopher J. Fluke, Michelle R. Quayle, and Colin R. McHenry. Embedding and publishing interactive, 3-dimensional, scientific figures in portable document format (pdf) files. *PLoS ONE*, 8(9):1 – 15, 2013.

[6] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.*, 23(1):5–48, March 1991.