B.Eng. Final Year Project

Research Proposal

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

Infinite Precision Document Formats

Keywords: **fractals, arbitrary precision, vector graphics, postscript, pdf, OpenGL**
Superviser: Prof. Tim French (UWA)
Team Members: Sam Moore (Mechatronics Eng), David Gow (Software Eng)[1]

1 Background

Early electronic document formats such as PostScript were motivated by a need to print documents onto a paper medium. In the PostScript standard, this lead to a model of the document as a program; a series of instructions to be executed by an interpreter which would result in "ink" being placed on "pages" of a fixed size[2]. The ubiquitous Portable Document Format (PDF) standard provides many enhancements to PostScript taking into account desktop publishing requirements[3], but it is still fundamentally based on the same imaging model[4]. This idea of a document as a static "page" has lead to limited precision in these and other traditional document formats.

The emergence of the internet, web browsers, XML/HTML and related technologies has seen a revolution in the ways in which information can be presented digitally, and the PDF standard itself has moved well beyond static text and figures[5, 6] but all modern document formats are still designed with the intention of showing information at a single, fixed level of detail.

As most digital display devices are smaller than physical paper medium, all useful viewers are able to "zoom" to a subset of the document. Vector graphics formats including PostScript and PDF support rasterisation at different zoom levels[2, 4], but the use of fixed precision floating point numbers causes problems due to imprecision either far from the origin, or at a high level of detail[7, 8].

We are now seeing a widespread use of mobile computing devices with touch screens, where the display size is typically much smaller than paper pages and traditional computer monitors; it seems that there is much to be gained by breaking free of the restricted precision of traditional document formats.

2 Aim

In this project, we will explore the state of the art of current document formats including PDF, PostScript, SVG, HTML, and the limitations of each in terms of precision. We will consider designs for a document format allowing graphics primitives at an arbitrary level of zoom with no loss of detail. A viewer and editor will be implemented as a proof of concept.

The goal is to be able to render a primitive at the same level of detail it is specified. For example, the coordinates of primitives drawn in a graphical editor will be limited by the resolution of the display on which they are drawn, but not by the viewer.

There are many possible applications for documents in which precision is unlimited. Several areas of use include: visualisation of extremely large or infinite data sets; digital artwork; computer aided design; and maps.

3 Methods

Initial research and software development will be conducted in collaboration with David Gow[1]. Once a simple testbed application has been developed, we will individually explore approaches for introducing arbitrary levels of precision; these approaches will be implemented as alternate versions of the same software. The focus will be on drawing simple primitives (lines, polygons, circles). However, if time permits we will explore adding more complicated primitives (font glyphs, bezier curves, embedded bitmaps).

At this stage we have identified two possible areas for individual research:

1. **Arbitrary Precision real valued numbers** — Sam Moore

   We plan to investigate algorithms related to the representation of real values to an arbitary degree of precision. This would allow for a document to be represented using a single global coordinate system. However, we would expect a decrease in performance with increased complexity of the data structure used to represent a real value. We will also need to consider limitations imposed by performing calculations on the GPU or CPU.

   Possible starting points for research in this area are Priest's 1991 paper, "Algorithms for Arbitrary Precision Floating Point Arithmetic"[9], and Goldberg's 1992 paper "The design of floating point data types"[8].

2. **Local coordinate systems** — David Gow [1]

   An alternative approach involves segmenting the document into different regions using fixed precision floats to define primitives within each region. A quadtree or similar data structure could be employed to identify and render those regions currently visible in the document viewer.

We aim to compare these and any additional implementations considered using the following metrics:

1. **Performance vs Number of Primitives**

   As it is clearly desirable to include more objects in a document, this is a natural metric for the usefulness of an implementation. We will compare the performance of rendering different implementations, using several "standard" test documents.

2. **Performance vs Visible Primitives**

   There will inevitably be an overhead to all primitives in the document, whether drawn or not. As the structure of the document format and rendering algorithms may be designed independently, we will repeat the above tests considering only the number of visible primitives.

3. **Performance vs Zoom Level**

   We will also consider the performance of rendering at zoom levels that include primitives on both small and large scales, since these are the cases under which floating point precision causes problems in the PostScript and PDF standards.

4. **Performance whilst translation and scaling**

   Whilst changing the view, it is ideal that the document be re-rendered as efficiently as possible, to avoid disorienting and confusing the user. We will therefore compare the speed of rendering as the standard documents are translated or scaled at a constant rate.

5. **Artifacts and Limitations on Precision**

   As we are unlikely to achieve truly "infinite" precision, qualitative comparisons of the accuracy of rendering under different implementations should be made.

4 Software and Hardware Requirements

Due to the relative immaturity and inconsistency of graphics drivers on mobile devices, our proof of concept will be developed for a conventional GNU/Linux desktop or laptop computer using OpenGL. However, the techniques explored could easily be extended to other platforms and libraries.

5 Timeline

Deadlines enforced by the faculty of Engineering Computing and Mathematics are *italicised*.[1].

Date	**Milestone**
17th April	Draft Literature Review completed.
1st May	Testbed Software (basic document format and viewer) completed and approaches for extending to allow infinite precision identified.
26th May	*Progress Report and Revised Literature Review due.*
9th June	Demonstrations of limitations of floating point precision in the Testbed software.
1st July	At least one implementation of infinite precision for basic primitives (lines, polygons, curves) completed. Other implementations, advanced features, and areas for more detailed research identified.
1st August	Experiments and comparison of various infinite precision implementations completed.
1st September	Advanced features implemented and tested, work underway on Final Report.
TBA	*Conference Abstract and Presentation due.*
10th October	*Draft of Final Report due.*
27th October	*Final Report due.*

[1]David Gow is being assessed under the 2014 rules for a BEng (Software) Final Year Project, whilst the author is being assessed under the 2014 rules for a BEng (Mechatronics) Final Year Project; deadlines and requirements as shown in Gow's proposal[1] may differ

References

[1] David Gow. Infinite-precision document formats (project proposal). http://davidgow.net/stuff/ProjectProposal.pdf, 2014.

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

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

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

[6] 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.

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

[8] David Goldberg. The design of floating-point data types. *ACM Lett. Program. Lang. Syst.*, 1(2):138–151, June 1992.

[9] D.M. Priest. Algorithms for arbitrary precision floating point arithmetic. In *Computer Arithmetic, 1991. Proceedings., 10th IEEE Symposium on*, pages 132–143, Jun 1991.

This proposal also available at:
`http://szmoore.net/ipdf/documents/ProjectProposalSam.pdf`.