

Proposal for a book covering imaging topics

Michael Still

Table of Contents

Introduction	3
What Will the Book Be Good For?.....	3
The Market for the Book	3
Licensing issues	3
The Outline	3
Schedule	7
Writing Sample.....	8
The format of the manuscript.....	8
Who Am I?.....	8

Introduction

This document outlines my proposal to develop a book covering an introduction to image programming techniques and libraries for O'Reilly. I would be prepared to answer any questions this proposal might raise, either via email to mikal@stillhq.com, or alternatively by calling +6414 382 568.

What Will the Book Be Good For?

This book is aimed at developers who are already competent in c programming, but now need to include imaging capabilities into their applications. The book will describe enough of the technical aspects of each format covered to provide a good grounding in the fundamentals, before moving on to discuss the intricacies of using the various APIs associated with the imaging formats. This is the book that I wished I had been able to buy when I first started image programming.

The Market for the Book

There are a large number of programmers who develop code which manipulates images. Many of these developers do not understand adequately the intricacies of the formats they are using, and this results in programs which don't work well. In my experience, many of the developers I have worked with could have made good use of a book such as the one I propose.

For example, the PDF format is quite fashionable at the moment. One of my previous employers, the Australian Patent Office (IPAustralia) has over 10 developers writing imaging software of some form. My new employer, TOWER Software, an independent software vendor, has several developers which write code that handles images of various formats. The comp.text.pdf USENET news group had around 1,000 posts in August 2002.

There is no competing title at this time.

Licensing issues

Some of the basis for the TIFF chapter of the proposed book was developed at the expense of IBM for the IBM DeveloperWorks series of TIFF articles completed in the first half of 2002. The contract with IBM imposes no restrictive terms on the use of this content, other than that a note containing the text "First published by IBM DeveloperWorks" appear on any subsequent uses. The sample book already includes this statement at the end of the TIFF chapter.

The sample work, which was produced for the Australian Unix User's Group annual winter conference, is licensed under the terms of the GNU Open Publication License, and the sample code is licensed under the terms of the GNU General Public License. It is possible that the license for the text of the book can be renegotiated if this is required by O'Reilly, although dual licensing would be preferable.

Michael Still, the author of this proposal, currently owns all rights to the sample book submitted to O'Reilly.

The Outline

I see the outline of the proposed book being along the lines of the following chapters (with indicative contents and length) ¹

It should be noted that all the APIs discussed in the book are covered by open source licensing of some form, and that all of these APIs are usable on win32 platforms as well as various unices including Linux.

Table 1. Proposed book chapters

Chapter name and description of content	Current length	Target
Introduction. Explanation of the motivation for the book, descriptions of the topics covered by the book, definition of assumed knowledge, license for the book content and sample code.	11	13
Concepts. Introduction to fundamental imaging concepts which need to be explained before further discussion of imaging can occur. For example: anti-aliasing, encryption, gray scale conversion, pixels, rasters, palettes, samples per pixel, bits per sample, et cetera.	20	40
TIFF. A discussion of the TIFF file format, including discussion of the pros and cons of the implementation decisions made by the TIFF standards committee, use of libtiff including examples. This chapter currently includes 70 pages of libtiff man pages, which I now believe should be removed from the manuscript. This chapter is the most complete in the sample book associated with this proposal, although it does still need work.	110	80
GIF. A discussion of the GIF file format, the history of why it was once popular, and is less popular now, and examples of how to use Eric Raymond's libgif and libungif.	6	40

Chapter name and description of content	Current length	Target
PNG. Comparison the PNG file format with GIF, the advantages of PNG over GIF, and discussion of how PNG is technically superior to GIF. An extensive discussion of how to use the libpng API, which comparisons with the libtiff examples from the TIFF chapter where appropriate. This chapter currently includes libpng documentation, which will be removed and replaced by a better discussion of the API and more examples.	70	80
JPEG. There is currently no JPEG chapter in the sample book. This chapter will discuss the JPEG format, and show examples of how to use the libjpeg API. This will include comparison of the API with the previously discussed libtiff TIFF API, and the libpng PNG API.	0	80
SVG. Introduction of non raster image formats, and a discussion of how SVG is an increasingly common vector graphics format. Discussion of the SVG specification and comparison with the various raster formats previously discussed. Samples of SVG files.	0	80

Chapter name and description of content	Current length	Target
PDF. Discussion of the PDF file format and the various versions of the specification (with attention to the different levels of functionality offered by each of these). Discussion of the fact that PDF is a hybrid raster / vector format. Examples of how to use Panda, my open source PDF generation API, and limited comparison with other PDF generation APIs. This chapter will also discuss the lexical parsing requirements of PDF. This chapter currently includes documentation for many of the Panda function calls, and these will be rationalized and reorganized.	90	150
ImageMagick. A relatively short chapter describing the ImageMagick suite of image conversion and manipulation utilities. This chapter will include example uses of the tools.	0	20
aalib. A short chapter describing aalib, a library which renders raster graphics into ASCII art, including examples.	0	10
SVGAlib. A short chapter discussing the most important functionality offered by SVGAlib, a library for drawing graphics without using X Windows.	0	40
Freetype. This chapter will discuss the Freetype TrueType rasterization API. It will include a limited discussion of theory, and include examples of how to use Freetype and integrate it with the raster imaging APIs discussed in the book.	0	40

Chapter name and description of content	Current length	Target
Compression. This chapter will discuss the various compression options offered by the raster formats in the book. This will enabled readers to make informed choices about what image format is appropriate for their needs. This will not be a mathematically comprehensive discussion however.	0	20
Conclusion. A summary of the information presented in the book.	1	5
Appendix: Selecting an appropriate image format. In conjunction with the compression chapter, this appendix will enable readers to be able to select the most appropriate imaging format for their needs. This appendix will be in the form of a checklist, or flowchart.	0	10
Index	0	Unknown
<i>Totals</i>	323	Approx 708

Rationale for the organization of the book

The book starts by introducing concepts which will need to be understood for the remainder of the book to make sense. It then moves on to discuss various image formats, starting with possibly the most common commercial image format, TIFF, and then comparing that format with it's various competitors. It then wraps up by providing assistance in selecting a format for your application.

Use of illustrations

Illustrations will be frequently used.

Use of color in illustrations

A facility for a series color plates grouped somewhere within the book would be preferred, but this is not absolutely essential.

Schedule

I intend to develop one chapter of the book at a time. This is the way the sample book was developed, and I have found that this has worked well.

In the process of determining the schedule for the development of the book, I have determined the rate at which I write in the following way:

1. I determined the average number of words per page for the sample book: $59,900 / 323 = 185$ words per page (not including source code) ²
2. The TIFF articles for IBM DeveloperWorks were 5,790 words, and took two weeks to develop. Therefore I was writing at the rate of 412 words a day

My proposed schedule is therefore as follows ³:

Table 2. Proposed schedule

Chapter (in proposed development order)	Days
TIFF. 40 new pages	$40 * 185 / 412 = 17$ days
GIF. 34 new pages	$34 * 185 / 412 = 15$ days
PNG. 56 new pages	$56 * 185 / 412 = 25$ days
JPEG. 80 new pages	$80 * 185 / 412 = 35$ days
SVG. 80 new pages	$80 * 185 / 412 = 35$ days
PDF. 60 new pages	$60 * 185 / 412 = 26$ days
ImageMagick. 20 new pages	$20 * 185 / 412 = 8$ days
aalib. 10 new pages	$10 * 185 / 412 = 4$ days
SVGLib. 40 new pages	$40 * 185 / 412 = 17$ days
Freetype. 40 new pages	$40 * 185 / 412 = 17$ days
Compression. 20 new pages	$20 * 185 / 412 = 8$ days
Introduction. 2 new pages	$2 * 185 / 412 = 1$ day
Concepts. 20 new pages	$20 * 185 / 412 = 8$ days
Conclusion. 4 new pages	$4 * 185 / 412 = 2$ days
Appendix. 10 new pages	$10 * 185 / 412 = 4$ days
<i>Total</i>	222 days

Note that this estimate does not include time for reworking the manuscript following editorial comments.

Writing Sample

A large portion of this book (approximately 30 percent) has already been developed for various magazine articles and conferences. This is available as a writing sample on request, however it should be noted that it is not ready for printing at this time.

The format of the manuscript

The current draft of the book has been developed using the DocBook SGML DTD (version 4.1). It is my intention that this use of DocBook would continue through the life cycle of the book. It should be noted that portions of the DocBook representation of the book is generated by a series of scripts written by myself. These scripts are publically available from my website at <http://www.stillhq.com>.

Who Am I?

I have been an application developer for about four years now. Prior to that I administered networks of Unix and Windows machines in a variety of academic and government environments. For the majority of that time I have been working on developing image processing systems, some of which have had hundreds of users. I am also the author of a variety of open source applications and libraries, including a PDF generation API. I have also supplied patches to a variety of other open source applications. I am also the maintainer of the comp.text.pdf USENET FAQ.

My goal with this proposed book is to ease the pain that comes with a first foray into image programming. This is the book that I wished that I had owned when I first started in the field. The book documents some of the "gotchas" I have experienced in my years undertaking this type of development, as well as a solid introduction to the various formats available, so that developers can make informed decisions.

I have previous experience writing to a professional standard. This includes a variety of conference presentations, and a recent series of two articles published by IBM DeveloperWorks (<http://www-109.ibm.com/cgi-bin/click.pl?url=http://www-106.ibm.com/developerworks/library/l-libtiff/&qry=tiff>, and <http://www-109.ibm.com/cgi-bin/click.pl?url=http://www-106.ibm.com/developerworks/library/l-libtiff2/&qry=tiff>).

Notes

1. Chapter lengths are given in A4 pages.
2. This relatively low word count is partially caused by the large number of illustrations and example source code in the sample book. This figure will also therefore cover time generating illustrations and example source code.
3. I have inserted times here as days, as opposed to calendar dates, as I am not aware of when the proposal will be approved.

