View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Massey Research Online

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

The Development of a Java Image Processing Framework

A thesis presented in partial fulfillment of the requirements for the degree of Master of Technology in Computer Systems Engineering at Massey University, Palmerston North, New Zealand.

Jesse Louis McLaughlin

2000

Abstract

Practical computer-based teaching methods are often used in conjunction with theory-based lecture sessions and textbooks when teaching image processing. In kind, electronic or on-line image processing courses commonly provide both theoretical and interactive components, however these are often disparate in that the software use to provide each component is independent rather than integrated. It is less common to find electronic instructional resources for image processing that integrate theoretical textual and practical interactive content together into one seamless package. An integrated approach has the advantage that the concepts are more easily conveyed and reinforced when taught 'side-by-side' this way.

The World Wide Web offers an attractive medium for delivering an integrated instructional resource on image processing. Applets written in Java may be seamlessly integrated into a hypertext environment. These applets can provide practical demonstrations of image processing concepts along side the relevant hypertext-based theoretical content. One of the major barriers to realising this kind of resource is the development effort required to create the necessary applets. This research demonstrates that the provision of a software framework can significantly reduce the burden of developing these applets. Such a framework provides a common code base that can be drawn upon during applet development, thereby avoiding the need to start from scratch each time a new applet is needed.

The framework's design is modelled on a dataflow view of image processing, allowing applets to be built in terms of interconnections between operations. This design is intended to provide the developer with an intuitive and easy-to-use application programming interface (API) for developing applets. The framework also provides APIs for the programmer to implement new operations and data types, thereby extending the capabilities of the framework. Further, the framework's design is general enough to allow it to be used for developing general purpose image processing programs, or other programs that lend themselves to development using a dataflow language. This thesis shows that the proposed framework achieves its aims through an example application of the development of an applet that demonstrates a thresholding operation.

Table of Contents

Abstract iii

Acknowledgments vii

1. Introduction 1

1.1 Image Processing 1

1.2 Web-Based Instruction 3

1.3 Electronic Textbooks 4

1.4 The Java Programming Language 6

1.5 A Framework 7

2. Framework Requirements 11

2.1 Image Processing Software 11

2.1.1 Categories of Image Processing Software 12

2.1.2 Software for Interactive Image Processing Instruction 13

2.1.3 Java Applets for Interactive Image Processing Instruction 15

2.1.4 The Basis of an Image Processing Framework 16

2.2 Framework Components 17

2.2.1 Extensible Image Processing Operation Library 18

2.2.2 Data Types 19

2.2.3 A Supervisory System 20

2.2.4 Applet Support 24

2.3 A Data Oriented Model 25

2.3.1 Algorithm Representation 26

2.3.2 A Dataflow Framework 27

2.4 Summary 29

3. Framework Design 33

3.1 Introduction 33

3.1.1 Tools 33

3.1.2 UML Notation 35

3.1.3 Methodology 38

3.1.4 Definitions 39

3.2 Operations 39

3.2.1 Operator 39

3.2.2 Primitive 42

3.2.3 Meta 44

3.2.4 Summary 45

3.3 Data Types 46

3.3.1 Data 46

3.3.2 Common Types 48

3.3.3 Summary 51

3.4 Data Passing 52

3.4.1 Input and Output 52

3.4.2 Progress 53

3.4.3 InputManager and OutputManager 57

3.4.4 DataManager 70

3.4.5 Port 77

3.4.6 FeedThru 79

3.4.7 Binding 79
3.4.8 Inputs, Outputs, Sources and Sinks 84
3.4.9 Summary 87
3.5 Scheduling 89
3.5.1 Blocking and Unblocking 90
3.5.2 Primitive's Main Loop 92
3.5.3 ControlFlowException 94
3.5.4 Shared Threads 95
3.5.5 Workspace 97
3.5.6 Sequencing 98
3.5.7 Summary 100
3.6 Applet Support 101
3.6.1 Hiperlet 101
3.6.2 Display 102
3.7 Additional Features 103
3.7.1 IllegalActionException 103
3.7.2 Looping Constructs 104
3.7.3 Macro 106

4. Application Example 109
4.1 Concept 109
4.2 Design 110
4.3 Implementation 111
4.3.1 Threshold 112
4.3.2 Control 114
4.3.3 DisplayImage 116
4.3.4 ThreshDemo 118
4.4 Conclusions 121

5. Summary and Conclusions 123
5.1 Future Work 126

References 132

Appendix 1: Optimally Accessing Image Data in Java **135**
A1.1 Introduction 135
A1.2 Storage/Access Patterns 136
A1.2.1 Array Configuration 136
A1.2.2 Primitive Type Selection 137
A1.2.3 Access Strategy 138
A1.3 Experimental Design 139
A1.3.1 Evaluation Criteria 139
A1.3.2 Patterns in the Experiment 139
A1.3.3 Experimental Conditions 140
A1.4 Results and Discussion 141
A1.5 Conclusions 143
A1.6 Future Work 144

Acknowledgments

First and foremost, thanks must go to my two very capable supervisors, Donald Bailey and Wyatt Page, each of whom made very different but equally important contributions to the research presented in this thesis. Thanks must go to Donald for his constant ability to provide expert advice and sound common sense in the face of my more fantastic and quick-conceived ideas. He was, after all, the originator of this research and therefore best qualified to provide the direction for the project, both overall and in its every detail. I would also like to thank Donald for inspiring me in the art of devising useful analogies, which itself proved useful when it came time to write this thesis. Wyatt, for his part (and to our fortune) had been less involved at the time the project was initially formulated, and so was able to bring a more objective perspective to the nature of the problem we had set out to solve. His speciality expertise in a variety of niche areas also proved invaluable on more than one occasion.

I am grateful to acknowledge Massey University's assistance, in the form of a Massey Masterate Scholarship. In kind, Donald's financial contribution to the project is also gratefully acknowledged in full, the sum of which afforded me an additional level of support that I might not have otherwise enjoyed.

I would like to give special thanks to my family. To my parents, for being forever interested in everything I do, and to my extended family, for being such a kind, generous and supportive bunch.

My friend and co-conspirator, David Orchiston, is deserving of thanks; his threats mixed with encouragement were of increasing value as the deadline for this thesis approached.

Finally, I would like to warmly thank Rachelle, not only for the constant companionship and much needed distraction she provided during this project, but also for being brave enough to love an aspiring computer systems engineer.