Kyffin Williams: Digital Analysis of Paintings

Final Report for CS39440 Major Project

Author: Alexander David Brown (adb9@aber.ac.uk)
Supervisor: Hannah Dee (hmd1@aber.ac.uk)

22nd April 2012
Version: 0.0.77 (Draft)

This report was submitted as partial fulfilment of a MEng degree in Software Engineering (G601)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.
- I understand and agree to abide by the University's regulations governing these issues.

Signature .....

Date .....

Consent to share this work

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature .....

Date .....

Acknowledgements

A large thanks to Hannah for being an amazing supervisor.

Abstract

Sir John “Kyffin” Williams was a landscape painter from Wales whose work was predominantly based in Wales and Patagonia. He studied at the Slade, one of Britain’s top art schools, after epilepsy ended his career in the army during the second world war. This epilepsy made Kyffin Williams sensitive to light and is the reason his work gets darker over time(3).

Gareth Lloyd Roderick, a PhD student in the National Library of Wales, has collected data such as the date or location, of these paintings. This data allows for some interesting analysis; particularly that of temporal or geographical classification of a given painting. That is being able to take a painting and decide the year or location in which it was painted from a database of existing, known, works by Kyffin Williams.

Temporal analysis will be the focus of this project as it allows for a diverse range of techniques; from statistical analysis of colour values of the paintings to looking at the length and style of paintbrush strokes. Geographical analysis would likely be very difficult, especially as the locations depicted were often sketched on-site then painted in a studio.

Whilst it would be nice to be able to predict the age of a painting with no known year, it is far more interesting to try to guess the year of paintings for which the date is known. This project will use leave-one-out cross-validation to help measure the effectiveness and validity of the analysis techniques employed in this project. Leave-one-out validation can be used with this project as the data set is small enough not to incur large performance overheads and the overall speed of the program is irrelevant so long as it completes within a decent amount of time.

One major limitation with this is it also includes the technique used for classification, some techniques might work better with K-Nearest Neighbour whilst other might benefit from more complex methods of classification. This means I will either have to stick with a single classification algorithm and hope it’s a good one for all techniques. Or I could find the best machine learning technique for each individual analysis technique, then perform comparison. This does assume that the best machine learning technique for every analysis technique exists within the scope of the project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Background & Objectives

1.1 Sir John "Kyffin" Williams

1.2 Interdisciplinary work with the National Library of Wales

1.2.1 Future Work

1.3 Existing Work

1.3.1 Edge-Orientated Gradients

1.3.2 Brush-stroke Analysis

1.4 Analysis Objectives

1.4.1 Colour-space Analysis

1.4.2 Texture Analysis

1.4.3 Brush-stroke Analysis

1.5 Classification Objectives

1.5.1 Classification

1.5.1.1 Use of Weka

1.5.1.2 Learning Classifier Systems (LCS)

1.5.2 Exemplars

Chapter 2

Development Process

You need to describe briefly the life cycle model that you used. Do not force your project into the waterfall model if it is better described by prototyping or some other evolutionary model. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was felt to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

You should briefly describe the design method you used and any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

2.1 Introduction

Introduce the specific model that you chose to use.

2.2 Modifications

Did you have to modify the model to suit a one-person project. If so, what did you change and why?

Chapter 3

Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

3.1 Overall Architecture

3.2 Some detailed design

3.2.1 Even more detail

3.3 User Interface

3.4 Other relevant sections

Chapter 4

Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex, perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant? Your implementation might well be described in the same chapter as Problems (see below).

Chapter 5

Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Have you tested your system on 'real users'? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

5.1 Overall Approach to Testing

5.2 Automated Testing

5.2.1 Unit Tests

5.2.2 User Interface Testing

5.2.3 Stress Testing

5.2.4 Other types of testing

5.3 Integration Testing

5.4 User Testing

Chapter 6

Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

- Were the requirements correctly identified?
- Were the design decisions correct?
- Could a more suitable set of tools have been chosen?
- How well did the software meet the needs of those who were expecting to use it?
- How well were any other project aims achieved?
- If you were starting again, what would you do differently?

Such material is regarded as the most important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree. You are expected to realise in which ways it falls short of perfection and of things that you did wrong.

Sadly, the critical evaluation is the weakest aspect of most project dissertations. Because of its importance, some examples are provided on the project website.

Chapter 7

Example LaTeX

This chapter includes some example LaTeX.

Ever advancing developments in computational power... mean ever more pictures of kittens on the internet. As you will see in Figure 7.1, some of them are very cute.

Figure 7.1: A picture of a kitten(**?**).

7.1 Overview

In this section I am going to include a spurious label, which appears in the code but has no effect on the display at the time it was inserted. I am also going to include a spurious citation to a journal article (**?**), a citation to a conference paper (**?**), a citation to a book (**?**), and a citation to a website (**?**). All of these citations have been added to the BibTeX (.bib) file which you'll find in

the References directory – they include some tricky stuff (accents and so on) which are explained in the comments in the BibTeX.

7.1.1 A bit of extra text to give the section some bulk

This is a paragraph of extra text just to make this section go over into a second page and to show the use of headers and page numbering that happens automatically once the text flows over to another page.

7.2 A few words of advice on LaTeX

One thing you should be aware of using LaTeX is that LaTeX has its own ideas about where things should be placed and about where page breaks should happen. This minimises the chances of widow and orphan text[1], but it can lead to you feeling like you've lost control if you're used to using software like Word. The best advice for text formatting is to just relax and relinquish control to LaTeX; for figures, it's a good idea to use the float package (already included in this template) and put [H] after your includegraphics command. You can see an example of this usage in the code used to insert Figure 7.1 on Page 7.

In the following paragraph, I've put a pointless equation. This is just so that you can see how to include an equation in a document. Equations are numbered separately, just like tables and figures.

$$X = \sum_{i=1}^{N} x_i + y_i \tag{1}$$

Like tables and figures, if you label an equation you can refer back to it using the ref command (for the number of the equation) or the pageref command (for the page the equation lies on). The source for this document has examples of both types of reference here: Equation 1 lies on Page 8.

7.3 Early work

Year	Kitten frequency	Notes
1993	0.04	World wide web begins to become popular
1995	0.2	Kittens take over
2008	0.34	Cats make a stand

Table 7.1: A pointless table, inserted to show that the list of tables will auto-update

7.3.1 The first signs of this topic

In this section we have a spurious link back to a spurious label, which appeared in Section 7.1.

[1] *Orphan* and *Widow* text are when the last line of a paragraph appears on the following page, or where a header appears on one page and the following text appears on the next

Appendices

0.1 Equations

0.1.1 Statistical Equations

0.1.1.1 Mean

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2}$$

0.1.1.2 Standard Deviation

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3}$$

0.1.2 Distance Equations

0.1.2.1 Manhattan Distance

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=0}^{n} |p_i - q_i| \tag{4}$$

0.1.2.2 Euclidean Distance

$$d_1(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=0}^{n}(q_i - p_i)^2} \tag{5}$$

0.1.3 Filter Equations

0.1.3.1 Gradient Direction

$$\theta = \operatorname{atan2}\left(\frac{\delta f}{\delta x}, \frac{\delta f}{\delta y}\right) \tag{6}$$

0.1.3.2 Gabor Filter

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{7}$$

where:

$$x' = x\cos\theta + y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

Annotated Bibliography

[1] P. Azad, "The integrating vision toolkit (IVT)," Website, 2011. [Online]. Available: http://ivt.sourceforge.net/

Used documentation (http://ivt.sourceforge.net/doxygen/) and examples (http://ivt.sourceforge.net/examples.html) to create a simple image blurring application to test the capabilities and ease of use of the library. Used these resources from the 16 October 2012 to the 24 October 2012.

[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

Used Python (http://opencv.willowgarage.com/documentation/python) and C++ (http://opencv.willowgarage.com/documentation) documentation for library reference and some learning on image processing/computer vision. Used since 11 October 2012.

[3] R. Harris, "How rolf learnt to paint like sir kyffin williams," BBC Broadcast, Feb. 2011. [Online]. Available: http://www.bbc.co.uk/programmes/p00f6nyt

A video on the BBC by Rolf Harris about some of Kyffin Williams' life and about his interesting style of painting.

[4] J. Li, L. Yao, E. Hendriks, and J. Z. Wang, "Rhythmic brushstrokes distinguish van gogh from his contemporaries: Findings via automated brushstroke extraction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 6, pp. 1159–1176, June 2012. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2011.203

Defines a complex method for analysing individual brush strokes which has been used to classify the period of two paintings by Van Gogh. This technique could be very powerful when applied to Kyffin Williams' work. This could be one of the most important techniques for the whole of the Kyffin Williams project.

[5] S. J. D. Prince, *Computer vision : models, learning, and inference*. Cambridge University Press, 2012. [Online]. Available: http://www.computervisionmodels.com/ 9781107011793.

Learning reference for Computer Vision and Machine Learning.

[6] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, June 2012. [Online]. Available: http://dx.doi.org/10.1038/nmeth.2019

Used as part of referencing FIJI documentation to test out of capabilities and ease of use of the library.

[7] SciJava, "Aggregator project for the fiji plugins 2.0.0-SNAPSHOT API." [Online]. Available: http://fiji.sc/javadoc/

Used as reference documentation to create a simple application to blur an image. Used from 11 October 2012 to 24 October 2012

[8] D. Stork and M. Duarte, "Computer vision, image analysis, and master art: Part 3," *IEEE Multimedia*, vol. 14, no. 1, pp. 14–18, 2007. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2007.6

Defines a potential method of analysing regions in a painting, this could be interesting to apply to Kyffin Williams' work as the regions in his earlier work are a lot less well defined as those in his later work.

[9] D. G. Stork, "Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature computer analysis of images and patterns," ser. Lecture Notes in Computer Science, X. Jiang and N. Petkov, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2009, vol. 5702, ch. 2, pp. 9–24. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03767-2_2

Notes a lot of useful literature to look at and some useful terminology and ideas for analysis techniques too.