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.

Adaptation of Colour Perception through Dynamic ICC Profile Modification

A thesis presented in partial
fulfilment of the requirements
for the degree of

Doctor of Philosophy
in Computer Science

at Massey University,
Albany (Auckland), New Zealand.

Guy Kristoffer Kloß

2010

Copyright © 2010 by Guy Kristoffer Kloss, Some Rights Reserved
This work is licensed under the terms of the
Creative Commons Attribution – Noncommercial 3.0 New Zealand license.
You are free to copy, distribute and transmit the work as well as adapt the work,
providing it is used for non-commercial purposes and it is cited properly.
The license is available at http://creativecommons.org/licenses/by-nc/3.0/nz/

Abstract

Digital colour cameras are dramatically falling in price, making them affordable for ubiquitous appliances in many applications. Change in colour perception with changing light conditions induce errors that may escape a user's awareness.

Colour constancy algorithms are based on inferring light properties (usually the white point) to correct colour. Other attempts using more data for colour correction – such as (ICC based) colour management – characterise a capturing device under given conditions through an input device profile. This profile can be applied to correct for deviating colour perception. But this profile is *only* valid for the specific conditions at the time of the characterisation, but fails with changes in light. This research presents a solution to the problem of long time observations with changes in the scene's illumination for common natural (overcast or clear, blue sky) and artificial sources (incandescent or fluorescent lamps).

Colour measurements for colour based reasoning need to be represented in a robustly defined way. One such suitable and well defined description is given by the CIE LAB colour space, a device-independent, visually linearised colour description. Colour transformations using ICC profile are also based on CIE colour descriptions. Therefore, also the corrective colour processing has been based on ICC based colour management. To verify the viability of CIE LAB based corrective colour processing colour constancy algorithms (White Patch Retinex and Grey World Assumption) have been modified to operate on $L^*a^*b^*$ colour tuples. Results were compared visually and numerically (using colour indexing) against those using the same algorithms operating on RGB colour tuples.

We can take advantage of the fact that we are dealing with image *streams* over time, adding another *dimension* usable for analysis. A solution to the problem of slowly changing light conditions in scenes with a static camera perspective is presented. It takes advantage of the small (frame-to-frame) changes in appearance of colour within the scene over time. Reoccurring objects or (background) areas of the scene are tracked to gather data points for an analysis. As a result, a suitable colour space distortion model has been devised through a first order Taylor approximation (affine transformation). By performing a multidimensional linear regression analysis on the tracked data points, parameterisations for the affine transformations were derived.

Finally, the device profile is updated by amalgamating the corrections from the model into the ICC profile for a single, comprehensive transformation. Following applications of the ICC based colour profiles are very fast and can be used in real-time with the camera's capturing frame rate (for current normal web cameras and low spec desktop computers). As light conditions usually change on a much slower time scale than the capturing rate of a camera, the computationally expensive profile adaptation generally showed to be usable for many frames.

The goal was to set out and find a solution for consistent colour capturing using digital cameras, which is capable of coping with changing light conditions. Theoretical backgrounds and strategies for such a system have been devised and implemented successfully.

Acknowledgements

My first and biggest thanks go to my lovely wife Friederike. Without her love and support the "Project Ph. D." would *not* have been possible. Especially the personal, mental and financial backing through the *W. I. F. E. scholarship scheme* were essential.

Of course, my parents and family have laid the corner stones by raising me (physically and mentally), and providing me with a good education. And my dad has dragged me into the printing industry. This gave me sound knowledge on what colour management is, how it is done and why it matters.

Ansgar: You are my best friend. You pointed out early, that there are life and opportunities also outside of Germany. During the last years, geographical distances were against seeing more of each other. But you were there for me early, you have "reoriented my corner stones" (to my parents' sadness), and I know you will always be there and be a good influence.

On a more academic side, I am thankful to Ken Hawick, for accepting me as a Ph. D. student without prior formal Computer Science education. And of course Napoleon Reyes, my supervisor, for giving me the "academic playground" at Massey that I have been exercising on, and for being a nag to guide me also through the more tedious essentials and red tape of the Ph. D. course.

I would also like to thank Andreas Schreiber, my former "bosslette" (little boss) and friend, who gave me the freedom to learn and "play" with Python on the job at the German Aerospace Centre (DLR) for productive purposes, that laid out the foundations for much of the programming conducted in this thesis. Andreas, I miss the boxing and collective departmental caffeine etiquette! Also, for getting me involved in the Grid Provenance Project, in which I met Omer Rana. Omer has pointed me towards New Zealand and Ken here at Massey University.

Brian Whitworth was (almost) always a very entertaining and inspiring discussional sparring partner in the corridor. Thanks for giving me reasons, to view my academic work beyond pure Computer Science.

Martí Maria and Graeme Gill, your feedback, information and insights into your respective colour management systems, that I have used extensively, were invaluable to the success of this research.

The education in my "original" academic degree in Chemical Engineering at the University of Dortmund has been hard, but fun. Lots of it ... and it has given me the highly valuable scientific background information that I am building on, which enabled me to think beyond business applications. It laid the foundations for the interest in Scientific Computing.

Finally, I would like to thank Pink Floyd for their fabulous music (and Radio Hauraki for playing them, and providing an online live radio feed). This music has provided some moments of sanity, while labouring on the research, to relieve me for a few minutes out of my working pace and to revitalise and give me new clarity.

Now, really ... finally! Many many thanks to all my friends, family, colleagues, reviewers, etc. that did not get a mention on this page. I want to keep this brief, so I am cutting off the list here. But you have helped to make it worth it!

Contents

Abstract iii

Acknowledgements v

Contents x

List of Figures xii

List of Tables xiii

1 Introduction 1

1.1 Problem Domain . . . 2
1.2 Objectives . . . 2
1.3 Significance . . . 3
1.4 Statement of Problem and Research Attempt . . . 4
1.5 Scope . . . 5
1.6 Verification . . . 5
1.7 Summary . . . 6
1.8 Hitch Hiker's Guide to this Thesis . . . 6

2 Colour Background and Theory 9

2.1 Cameras as Measuring Devices . . . 9
2.2 Colour Theory . . . 11
2.2.1 Image Formation and Capturing . . . 11
2.2.2 Human Colour Vision . . . 18
2.2.3 Quantifying (Human) Colour Vision . . . 19
2.3 Colours in the Digital World . . . 21
2.3.1 Colour Spaces . . . 22
2.3.2 Device-Dependent Colour Spaces . . . 23
2.3.3 Device-Independent Colour Spaces . . . 23
2.3.4 Perceptually Linerarised Colour Spaces . . . 24
2.3.5 Colour Management . . . 26
2.3.6 Illuminant Adaptation . . . 26

3 Colour Adaptation 29

3.1 Approaches for Chromatic Adaptation . . . 30
3.1.1 Static Chromatic Adaptation . . . 30
3.1.2 Dynamic Chromatic Adaptation . . . 31
3.1.3 Failing Adaptation . . . 32
3.2 Spectral Adaptation . . . 32
3.3 Computational Colour Constancy . . . 33

3.4 Histogram Colour Correlation . . . 33
3.5 A Priori Knowledge Based Methods . . . 34
3.6 Learning Colour Constancy . . . 35
3.6.1 Biologically Based Approaches . . . 35
3.6.2 Neural Network Based Approaches . . . 35
3.6.3 Probabilistic Learning . . . 36
3.6.4 Genetic Programming Based Approaches . . . 37
3.7 Adaptation Method Comparison . . . 37

4 Colour Constancy with $L^*a^*b^*$ 41
4.1 Introduction . . . 41
4.2 Colour Constancy Algorithms . . . 42
4.2.1 White Patch Retinex . . . 43
4.2.2 Grey World Assumption . . . 44
4.3 Algorithm Modifications . . . 45
4.3.1 White Patch Retinex . . . 45
4.3.2 Grey World Assumption . . . 47
4.3.3 Chromatic Adaptation . . . 47
4.4 Implementation and Evaluation . . . 48
4.4.1 Implementation of Colour Constancy . . . 49
4.4.2 Implementation of Colour Indexing . . . 49
4.5 Experiments . . . 50
4.6 Results . . . 50
4.6.1 Visual Evaluation Results . . . 51
4.6.2 Quantitative Evaluation . . . 55
4.7 Conclusions on Colour Constancy . . . 58

5 Colour Management 61
5.1 Introduction . . . 61
5.2 Hardware . . . 63
5.3 Colour Profiles . . . 64
5.4 Profile Creation . . . 65
5.5 Colour Translation . . . 66
5.6 Illuminant Adaptation . . . 68
5.7 Implementations . . . 68
5.8 Conclusions on Colour Management . . . 69

6 Dynamic Adaptation 71
6.1 Limitations of Colour Constancy . . . 72
6.2 Limitations of Colour Management . . . 72
6.3 Dynamic Profile Adaptation – A Hybrid Approach . . . 73
6.4 Conclusions on Dynamic Adaptation . . . 75

7 ICC Profile Generation 77
7.1 Blueprint of an ICC Profile . . . 77
7.1.1 ICC Tags . . . 78
7.1.2 Rendering Intents . . . 79
7.2 Interpolation . . . 80
7.2.1 Regularised Linear Spline Fitting . . . 81
7.2.2 Numerical Formulation and Solution . . . 85
7.2.3 Approach for Border Problems . . . 87
7.3 Implementation . . . 91

7.4 Results . . . 93
7.4.1 n-D Spline Fitting . . . 93
7.4.2 Colour Profiling in the Presence of Measurement Noise . . . 96
7.5 Conclusions and Future Work on Profile Generation . . . 99

8 Adaptive Colour Transformation 101
8.1 Analysing Light-Induced Colour Shifts . . . 101
8.2 Compensation For Colour Shifts . . . 107
8.2.1 Obtaining the Affine Transformation . . . 109
8.2.2 Quantitative Evaluation . . . 110
8.2.3 Evaluation of the Shift Vector Field . . . 113
8.2.4 Visual Image Evaluation . . . 114
8.3 Determining Colour Shifts . . . 117
8.4 *A priori* Knowledge Based Correction . . . 119
8.4.1 Initial Camera Characterisation/ICC Profiling . . . 120
8.4.2 Colour Adaptation . . . 120
8.4.3 Colour Adaptation Results . . . 122
8.4.4 Colour Segmentation . . . 124
8.5 Static Perspective Based, Continuous Correction . . . 127
8.5.1 Continuous Long Term Correction Results . . . 129
8.5.2 Summary of Continuous Correction Performance . . . 130
8.6 Conclusions on Adaptive Colour Transformations . . . 131

9 ICC Profile Adaptation 133
9.1 Correcting ICC Profiles . . . 133
9.2 Profile Adaptation Process . . . 134
9.3 Implementation . . . 136
9.4 Results . . . 136
9.5 Conclusions and Future Work for Profile Adaptation . . . 137

10 Implementation 139
10.1 CMS Tool Evaluation . . . 139
10.1.1 Argyll CMS . . . 139
10.1.2 SampleICC . . . 140
10.1.3 Little CMS . . . 141
10.2 Approach . . . 142
10.3 Versions Used . . . 145
10.4 Outcome . . . 146
10.5 Conclusions on Implementation . . . 146

11 Discussion 149
11.1 Contributions . . . 149
11.1.1 Colour Constancy with $L^*a^*b^*$ . . . 149
11.1.2 Colour Management and ICC Profiling . . . 151
11.1.3 Adaptive Colour Transformation . . . 151
11.1.4 ICC Profile Adaptation . . . 153
11.1.5 Implementation . . . 153
11.2 Summary of Findings . . . 154
11.2.1 Colour Spaces . . . 155
11.2.2 Dynamic Colour Adaptation . . . 155
11.2.3 Integration with Colour Management . . . 156
11.3 Context . . . 156

11.3.1 Limitations (and Potential Solutions) . . . 156
11.3.2 Implications . . . 157
11.3.3 Applications . . . 157
11.4 Future Research Potential . . . 158

12 Conclusions 161
12.1 Contributions . . . 162
12.2 Future Work . . . 163

Glossary 167

A Python/Native Code Integration 169
A.1 Wrapping Little CMS with Ctypes . . . 170
A.2 The Example . . . 171
A.3 Code Generation . . . 171
A.3.1 Parsing the Header File . . . 172
A.3.2 Generating the Wrapper . . . 173
A.3.3 Automating the Generator . . . 173
A.4 Refining the C API . . . 174
A.4.1 Creating the Basic Wrapper . . . 174
A.4.2 `c_lcms` Example . . . 175
A.5 A Pythonic API . . . 175
A.5.1 `littlecms` Example . . . 176
A.6 Conclusions on Python/Native Code Integration . . . 176

B Applications in Industry Projects 179
B.1 Baggage Identification . . . 179
B.1.1 Image Normalization . . . 181
B.1.2 Determination of Visual Fingerprint . . . 181
B.1.3 Matching Algorithm . . . 182
B.1.4 Process Control . . . 183
B.1.5 Industry Partners . . . 183
B.2 Mobile Farm Management . . . 183
B.2.1 Industry Partner . . . 184
B.3 Manufacturing Quality Control . . . 184
B.3.1 Industry Partner . . . 185

C Simulation Project 187
C.1 Software Architecture . . . 187
C.2 Implementation . . . 191
C.3 Conclusions for Simulation Project . . . 192

D Various Python Tools and Techniques 193
D.1 Parallel and Distributed Programming . . . 193
D.1.1 Parallelisation Theory . . . 194
D.1.2 Process Based Parallelisation . . . 195
D.1.3 Inter-Process Communication . . . 198
D.1.4 Parallel and Distributed Programming Conclusions . . . 200
D.2 Data Plotting and Visualisation . . . 201
D.2.1 Two Dimensional Tools . . . 201
D.2.2 Three Dimensional Tools . . . 205
D.2.3 Data Plotting and Visualisation Conclusions . . . 210

List of Figures

1.1 The Hitch Hiker's Guide to this Thesis. 7

2.1 Simple model of colour image formation. 12
2.2 Spectral power distribution of standardised illuminants D_{65} and F2. 13
2.3 A bright red surface's spectral reflectance. 13
2.4 Colour sensing characteristics of human cone cells. 14
2.5 Relative RGB sensitivities of a typical digital camera. 14
2.6 The dichromatic reflection model. 16
2.7 RGB colour matching functions. 20
2.8 XYZ colour matching functions. 21
2.9 Comparison of the size of different RGB colour spaces. 24

4.1 Basic White Patch Retinex algorithm on RGB colours. 44
4.2 Basic Grey World Assumption algorithm on RGB colours. 45
4.3 Visualisation of the brightest $L^*a^*b^*$ colours in an image. 46
4.4 White point estimation according to the White Patch Retinex algorithm. . . 46
4.5 White point estimation according to the Grey World Assumption algorithm. . 47
4.6 Channel scaling transformation adapted for $L^*a^*b^*$. 48
4.7 Variations of colour constancy algorithms applied to old photograph. 51
4.8 Variations of colour constancy algorithms applied to under water photograph. 52
4.9 Variations of colour constancy algorithms applied to "Barnard's ball" picture. 53
4.10 Variations of colour constancy algorithms applied to "Barnard's ColorChecker" picture. 54
4.11 Comparison of chromaticity distributions used for colour indexing. 58

5.1 Colour transformation using input and output profiles. 67

6.1 Workflow for static ICC profile generation. 73
6.2 Workflow for cooperative image processing and profiling. 74

7.1 Chain of processing elements for an "AToBx" ICC tag. 79
7.2 The mechanical analogue of the modelling process. 82
7.3 Spline interpolations in 1-D and 2-D. 94
7.4 3-D rendering of the interpolation volumes for an ICC profile. 95
7.5 Location and error of RGB measurements of a test chart. 96
7.6 Comparison of ICC profile corrected colour images. 97

8.1 Visual evaluation of scene's colour distribution through histogramming. . . . 102
8.2 Camera setup for analysis of colour shift samples series. 105
8.3 Shift vector field for illuminant "`65000000`" and "`65650000`" against "`65270000`". 107

8.4 Shift vector field for illuminant “fluoresc” and “Inca____” against “65270000”. 108

8.5 Shift vector field of uncorrected vs. corrected colour samples “65000000” and “65650000”. 115

8.6 Shift vector field of uncorrected vs. corrected colour samples “fluoresc” and “Inca____”. 116

8.7 Shift vector field of corrected colour sample “27270000” with distorted greys. 116

8.8 Reference image “65270000”. 117

8.9 Visual comparison of colour correction approaches with different illuminants. 118

8.10 Colour samples and characterisation target under reference conditions. 121

8.11 Individual steps of the colour correction. 122

8.12 ΔE^*_{ab} for different shutter speeds with/without colour correction. 123

8.13 Fitness function for genetic colour classifier training. 125

8.14 Scores for fitness of colour classifier training. 126

8.15 Scores for applied colour classifiers. 127

8.16 Scenes used for long term static perspective colour correction. 128

8.17 Colour correction results for long term, static perspective observation. 129

9.1 Pseudo code of the algorithm updating the CLUT values in an ICC profile. . 135

A.1 Example in C using the *LittleCMS* library directly. 172

A.2 Essential parts of the code generator script. 173

A.3 Lines to be patched into the generated module _lcms. 174

A.4 Extract from module _setup.py. 174

A.5 Example using the basic API of the c_lcms module. 175

A.6 Example using the object oriented API of the littlecms module. 176

B.1 UML activity diagram for baggage identification system. 180

C.1 Screen shot of simulation results at a point of time. 188

C.2 Class diagram of composition for the simulation implementation. 189

C.3 Sequence diagram of actions during a simulation time step. 190

D.1 Gnuplot example, adding values on every call to update() to a list for plotting. 203

D.2 Example for matplotlib, similar to the previous Gnuplot example. 204

D.3 Surface plot from irregularly sampled data created with Mayavi. 206

D.4 Example of a surface plot from irregularly sampled data using Mayavi. 206

D.5 Example simulating a Brownian point cloud using VPython. 207

D.6 Some sample plots using VPython. 209

D.7 Adaptations to imports and run() method for Mayavi visual module. 210

List of Tables

4.1 Colour correction on un-clipped pixels only, colour indexing applied to all pixels. 56
4.2 Colour correction and colour indexing applied to un-clipped pixels only. . . . 57

7.1 Comparison of profiles' fit to measured characterisation data. 98

8.1 Camera settings for colour shift samples series. 104
8.2 Description of illuminant/filter combinations for illumination cases. 105
8.3 Relative white points of used illuminants in $L^*a^*b^*$. 106
8.4 Comparison of affine colour correction results using all data points. 111
8.5 Averages of transformation results for all illuminants using all data points. . . 112
8.6 Comparison of affine colour correction results computed using six data points. 113
8.7 Averages of transformation results for all illuminants using six data points. . 114

10.1 Exact versions of codes, tools and libraries used or discussed for this research. 145

D.1 Abstraction levels of explicitness for parallel computing models. 195