



UNIVERSITI PUTRA MALAYSIA

**APPLYING YUV COLOR MODE IN COLOR IMAGE
PROCESSING BY USING JAVA SWING**

SAWSAN KAMEL SHARIAH

FK 2001 8

**APPLYING YUV COLOR MODE IN COLOR IMAGE
PROCESSING BY USING JAVA SWING**

By

SAWSAN KAMEL SHARIAH

**Thesis Submitted in Fulfilment of the Requirement for the Degree of
Master of Science in the Faculty of Engineering
Universiti Putra Malaysia**

September 2001



This work is dedicated to the memory of
my
Late father Dr. Kamel Shariah
God rest his soul.
Whom I miss so much
My lovely mother
for her everlasting love
and
My father in law
Dr. Ismael Abu Shariah
For his endless love
And care



Abstract of thesis presented to the Senate of University Putra Malaysia in fulfilment
of the requirement for the degree of Master of Science

**APPLYING YUV COLOR MODEL IN COLOR IMAGE PROCESSING
BY USING JAVA SWING**

By

SAWSAN KAMEL SHARIAH

September 2001

Chairman: Abd. Rahman Ramli, Ph.D.

Faculty: Engineering

One of the most exciting new areas in computer technology is the development in color image processing field. Image processing is used to improve the visual quality of an image.

Jaguar Java Software is a GUI (Graphical User Interface) software, powerful yet very simple to use, provides the means to filter color images in RGB Color space and YUV Color space, to produce high quality filtered images. YUV Color space model proved to filter images in a smooth way without losing any data, only enhancing the image structure. Different images were applied, and satisfactory results were obtained of multi-applications objectives.

Swing was used as the main programming language, which is part of the Java Foundation Classes (JFC) library. It offers much improved functionality as new components, expanded component features, better event handling.



Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Master Sains

**APLIKASI MODEL WARNA YUV DALAM IMEJ BERWARNA DENGAN
MENGUNAKAN JAVA SWING**

Oleh

SAWSAN KAMEL SHARIAH

September 2001

Tuan Pengerusi : Abd. Rahman Ramli, PhD

Fakulti : Kejuruteraan.

Salah satu perkara yang paling menarik dalam teknologi komputer ialah pembangunan warna dalam bidang pamprosesan imej. Pamprosesan ini penting untuk meningkatkan lagi kualiti visual dalam imej.

Perisian Jaguar Java ialah perisian GUI (Graphical User Interface), yang sering diguna Ia dapat menapis warna imej dalam RGB dan YUV, untuk mengeluarkan imej yang berkualiti tinggi. YUV model telah terbukti tanpa kehilangan data, akan tetapi dapat meningkatkan struktur imej tersebut. Imej-imej berlainan diuji, dan keputusan yang memuaskan dapat dicapai.

Perisian Swing telah digunakan sebagai perisian utama, yang mana serupakan sebahagian dari pustaka JFC (Java Foundation Class). Ia memberi banyak kelebihan sebagai komponen baru, yang dapat dipertingkatkan untuk memberi banyak kemudahan kepada pengguna.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude and sincere to my supervisor Dr. Abd. Rahman Ramli for supporting me in his philosophy and for his guidance and supporting with his scientific knowledge throughout the course, until this research reached the best form.

I would also like to extend a personal word of thanks and greatest appreciation to Dr. Mahmud Hassan, and Dr. Veeraraghavan Prakash, my research co-supervisors, for their discussion comments and encouragement. I am grateful to Dr. Borhanuddin Mohd Ali, Dr. Mohammed Faruqi, for their guidance and advice during my courses.

A special thanks goes to my husband Ph.D candidate Mohammed Ismael Abu Shariah, who provided me with his comment in the CDP seismic reflection subsurface sections, by using Jaguar Java Filters, and was patient enough to complete this research successfully, to him I hold a deep gratitude.

My warmest thanks are paid to my kids, Al-Yasmin, Al-Lieth, Al-Reem, and Al-Leen for being so lovely and patient during my study, until they reached the level of being only satisfied with computer applications and Java programming associated with mom.

I would like to express my deepest thanks and gratitude to my father in law Dr. Ismael I. Abu Shariah, Public service Department Malaysia (Malaysian Technical Cooperation Program, MTCP), Mr. Adnan Abdon (JPA), Mr. Zinal Abdian (JPA), Mrs. Zawiah (JPA) for providing me the financial support and their valuable help during my study.

The last but not the least, my warmest thanks are paid to my loving Mother, A warm thanks and appreciation I hold to my Father in law Dr. Ismael Abu



Shariah, and my Mother in law, for their love and patient they provided me with. I would also like to extend my thanks to my brothers, sister, brothers in law, sisters in law, and my class mates during my study, Mr. Radwan, Miss Saraswathy, Mr. Abu tasnim, Mr. Mohammed Suleiman, Mr. Adi Azmir, Engineering Faculty staff, and GSO staff, to all I hold my graduated.

I hope for all Malaysian people Happiness and health, for their endless hospitality they provided us with during our stay here in Lovely Malaysia.



CONTENTS

	Page
DEDICATIONS	ii
ABSTRACT	iii
ABSTRAK	iv
ACKNOWLEDGEMENT	v
APPROVAL SHEETS	vii
DECLARATION FORM	ix
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS	xix
CHAPTER	
1 INTRODUCTION	1
1.2 General	1
1.3 Objective	2
1.4 Software Methodology	3
1.4.1 Language and platform	3
1.4.2 Java's Architecture	3
1.4.3 Java is platform independent	7
1.4.4 Approaches for applications for the Web	8
1.4.5 Software	8
1.4.6 Hardware	9
1.5 Image Filtering	9
1.6 YUV color space	11
1.7 Why was YUV model chosen	11
1.8 YUV-RGB Conversion	13
1.9 Thesis organization	14
2 COLOR SPACE AND JAVA SWING	16
2.1 Introduction	16
2.2 Color	17
2.2.1 Luminance	20



2.2.2	Hue	21
2.2.3	Saturation	21
2.3	Color space conversions	25
2.3.1	Color space	25
2.3.2	YUV color space	37
2.3.3	Conversion from RGB to YUV	37
2.4	Web browser color space	40
2.5	Java programs.	40
2.6	Java is fulfilling programming language	40
2.6.1	Objective-Oriented	42
2.6.2	Interpreted	43
2.7	Java Swing	43
2.7.1	Swing	43
2.7.2	The JFC (Java Foundation Classes)	44
2.7.3	Swing performance	43
2.8	Designing a user Interface with Swing	44
2.9	Swing Event Handling	47
2.10	New features of Swing	49
2.11	Summary	49
3	ANALYSIS AND DESIGN OF THE PROPOSED SOFTWARE	51
3.1	Introduction	51
3.2	System Domain	51
3.3	System Characteristics	53
3.4	The Structure of the proposed system	53
3.5	Classes Associations (Parts of the proposed System)	54
3.5.1	Jaguar Java Class	54
3.5.2	Desktop Panel Class	55
3.5.3	The DesktopMge Class	55
3.5.4	Menutool Class	55
3.5.5	RGB Frame Class	57

3.5.6	YUVFrame Class	57
3.5.7	Histogram Class	57
3.5.8	FilterPanel	58
3.6	The main components of the proposed software	58
3.6.1	Container	59
3.6.2	Layout	62
3.6.3	Data Components	62
3.7	Summary	65
4	DESKTOP OF JAGUAR JAVA	66
4.1	Introduction	66
4.2	Jaguar Java Software Desktop	66
4.2.1	Toolbar	68
4.2.2	Filter Palette	68
4.2.3	RGB Frame	68
4.2.4	YUV Frame	72
4.2.5	Histogram Frame	75
4.2.6	Look &Feel Support	75
4.3	Foundation classes of Jaguar Java	78
4.3.1	Jaguar Java classes	78
4.3.2	Desktop Panel class	80
4.3.3	Desktop Manager class	80
4.3.4	Menu Toolbar class	81
4.3.5	RGB Frame class	81
4.3.6	YUV Frame class	81
4.3.7	Histogram class	82
4.3.8	Filter Panel class	82
4.4	Pragmatic Model	82
4.5	Filtering	86
4.5.1	Digital Image Processing	86
4.5.2	Spatial Filters	87
4.5.3	Area Processes	89
4.5.4	Convolution	90

4.6	Filters	93
4.6.1	Blur	94
4.6.2	Brightness Adjustment	94
4.6.3	Edge Detector	97
4.6.4	Edge Enhancement	97
4.6.5	High-Pass Filter	100
4.6.6	Noise Filter	100
4.6.7	Sharpen	104
4.6.8	Smoothing	104
4.6.9	Edge Detection with Sobel's Algorithm	104
4.6.10	Prewitt Edge Detector	108
4.6.11	Laplacian Edge Enhancement	111
4.7	Summary	113
5	DISCUSSION	114
5.1	Discussion	114
5.2	Comparing between RGB & YUV Mode Filtering	115
5.3	Applications	118
5.3.1	Geophysical prospecting	118
5.3.2	Medical images	132
5.3.3	Weather lightning	132
5.3.4	Collateral images	138
5.4	Advantages and disadvantages of Jaguar Java Software	143
		147
6	CONCLUSION AND FUTURE WORK	147
6.1	Conclusion	148
6.2	Suggestions for further work	148
	REFERENCES	149
	VITA	155

LIST OF FIGURES

	Pages
Figure 1.1: Java is a cross platform programming language	4
Figure 1.2: The Java Architecture	4
Figure 1.3: The steps required to run a program written in Java	6
Figure 1.4: Showing the Y,U, and V channels	12
Figure 2.1a: The spectrum	18
Figure 2.1b: The relative absorption of color by the human eye	18
Figure 2.1c: The Human Retina [Gauge, 1999].	19
Figure 2.2a: The CIE L*a*b* model	24
Figure 2.2b: The CIE Chromaticity Diagram	24
Figure 2.3: The CIE Color Chart	26
Figure 2.4: The RGB Color Cube	27
Figure 2.5: Neutral Gradient Line.	28
Figure 2.6: Primary-Secondary Gradient Lines.	30
Figure 2.7: Plane of the Primary Colors.	31
Figure 2.8: Plane of the Secondary Colors.	33
Figure 2.9: Primary and Secondary Gradient Vectors.	34
Figure 2.10a: Edge of Saturated Hues	35
Figure 2.10b: RGB value as used with 24-bit color on a computer.	36
Figure 2.11: RGB to YUV, and visa versa.	39
Figure 2.12: 6x6x6 RGB cube and the color spaces it bounds.	41
Figure 2.13: Diagram showing the Java environment and its relationship with other platforms.	45

Figure 2.14: Swing structure (hierarchy)	46
Figure 2.15 Swing stack and its relationship with Java Foundation Classes	48
Figure 3.1 : Classes Diagram	56
Figure 3.2: Viewcomponent, container	60
Figure 3.3: JFrame, Jdesktop pane	60
Figure 3.4: Jinternal Frame, Jpanel Frame.	61
Figure 3.5: Jtabbed, Jsplittedpanel.	61
Figure 3.6: JFrame , Jdesktop pane	63
Figure 3.7: Jflowlayout, Jgridbaglayout	63
Figure 3.8: Jlist, Jcheckbox.	64
Figure 4.1: Jaguar Java Desktop	67
Figure 4.2: Jaguar Java toolbar.	69
Figure 4.3: The Filter Palette	70
Figure 4.4: RGB frame	71
Figure 4.5a: The YUV channels.	73
Figure 4.5b: The YUV frame.	74
Figure 4.6: Jaguar Java Histogram	76
Figure 4.7: Look & Feel support	77
Figure 4.8a: Jaguar Java Pragmatic	83
Figure 4.8b: Diagram illustrates Image Histogram Function in Jaguar Java	84
Figure 4.8c. Java Jaguar pragmatic	85
Figure 4.9a: Shows a convolution kernel	91
Figure 4.9b: The convolution mask.	92
Figure 4.10: The original image upper left, and the blur filtered image right down:	95

Figure 4.11: Brightness filter example	96
Figure 4.12: The upper image is the original, and the lower filtered with Edgedetector filter.	98
Figure 4.13: The upper image is the original, and the lower filtered with Edge Enhancement Filter.	99
Figure 4.14: The upper image is the original, and the lower image filtered with High pass filter.	101
Figure 4.15: The upper image is the original, and the lower of both image filtered with High pass filter by using RGB mode (middle), and YUV mode.	102
Figure 4.16: The upper image is the original, and the lower filtered with Noise cleaner filter.	103
Figure 4.17: The upper image is the original, and the lower filtered with Sharpen filter	105
Figure 4.18: The upper image is the original, and the lower filtered with Smoothing filter.	106
Figure 4.19: The upper image is the original, and the lower filtered with Sobel filter.	109
Figure 4.20: The upper image is the original, and the lower filtered with Prewitt filter.	110
Figure 4.21: The upper image is the original, and the lower filtered with Laplacian filter.	112
Figure 5.1: The original image in the middle, upper right image filtered by Borderized filter in RGB mode, the same image on the lower left filtered by Borderized filter in the YUV mode.	116
Figure 5.2: The original image in the middle, upper right image filtered by Emboss filter RGB mode, the same image on the lower left filtered by Emboss filter in the YUV mode.	117
Figure 5.3: CDM-seismic reflection section of the subsidence and sinkhole development cases	120
Figure 5.4: The region of geohazard in the CDM seismic reflection section	125
Figure 5.5: shows the image about the location of cavity	126

system and its dimension.

Figure 5.6: CDM seismic reflection section filtered by different RGB mode filtering	127
Figure 5.7: CDM seismic reflection section filtered by different RGB mode filtering	128
Figure 5.8: CDM seismic reflection section filtered by different RGB mode filtering	129
Figure 5.9: CDM seismic reflection section filtered by different RGB mode filtering	130
Figure 5.10: CDM seismic reflection section filtered by different RGB mode filtering	131
Figure 5.11: (a) The original image, b) The image filtered by Defocus filter stacking twice, c) The image filtered by Define filter, and d) The image filtered by Borderized	133
Figure 5.12: (a) Original, (b) Dark shadow filter, (c) Dark shadow filter stacking with invert, and (d) Filter Color [3] Filter.	134
Figure 5.13: (a) Original, (b) Programmer defined filter and (c) Stacking dark shadow with invert filter stacking with invert.	135
Figure 5.14: Lightning is more common in storms over land than over oceans, prefers afternoons and varies from one season to the next, [From internet site (Courtesy National Severe Storms Laboratory)].	136
Figure 5.15: (a) Shows the original lightning in image, (b) Shows posterized filter in RGB, (c) Shows posterize in YUV, (d) Shows with programmer defined filter, (e) Shows with overexposed filter, and (f) Shows with prewitt filter.	139
Figure 5.16: (a) The original lightning image is the lower image, surrounding with the other images, filtered by the programmers defined filter, interesting facts can be seen in the pictures.	140
Figure 5.17: All these images were filtered using Filter Colour (a) Shows the original lightning image, (b) and (c) show the image being filtered in the YUV mode.	141

Figure 5.18: The original image is the upper one, the two others were filtered using pasteurized, and color filters respectively. The region of interest was bounded with the red line. [Environmental News Network, 1999]. 142

Figure 5.19: All these images were filtered using Filter Colour (a) Shows the original lightning image, (b) shows filter colour blue invert, (c) shows filter blue remove, (d) shows filter color 5, (e) shows filter color 2, and (f) shows filter green remove. 144

Figure 5.20: The original image in the middle surrounding with different Laplacian filtered with stacking images. 145

ABBREVIATION

AWT :	Abstract Windows Toolkit.
Brightness:	The attribute of a visual sensation according to which an area appears to emit more or less light.
Chromaticity:	Chromaticity diagrams provide a graph of all possible colours for a given system with luminance removed.
CIE :	Commission Internationale de L'Éclairage. An international organisation that deals with standards and other matters related to colorimetry.
CMD:	Common Mid Depth Point reflection technique.
CMY :	Cyan –Magenta-Yellow
Gamut :	The gamut of colors is all colors that can be reproduced using the three primaries. The Lab gamut covers all colors in visible spectrum. The RGB gamut is smaller, hence certain visible colors (e.g. pure yellow, pure cyan) cannot be seen on monitors. The CMYK gamut is the smallest (but not a straight subset of the RGB gamut)
GUI:	Graphical User Interface
JFC :	Java Foundation Classes
JVM	Java Virtual Machine
International Commission On Illumination :	As its name implies, the International Commission on Illumination – abbreviated as CIE from its French title Commission Internationale de l'Éclairage - is an organization devoted to international cooperation and exchange of information among its member countries on all matters relating to the science and art of lighting.
Lab model:	A refined CIE model, named CIE L*a*b in 1976 Luminance: L Chrominance: a -- ranges from green to red, b – ranges from blue to yellow .
NTCS :	National Television Communication System.



PAL :	Phase Alternation Line.
RGB :	A colour specification system that allows a colour to be specified by providing its red, green, and blue components. This system assigns a linear relationship to all of its colours.
RGB Colour Cube:	RGB Colour Cube: The three-dimensional unit cube bounded by (0,0,0) (black) and (1,1,1) (white) with all possible colours of an RGB colour system graphed in between.
SECAM :	Sequential Couleur Avec Mémoire or Sequential Colour with Memory.
SPD :	Spectral Power Distribution.
VOD:	Video-on-demand

CHAPTER ONE

INTRODUCTION

1.1 Introduction

As our society becomes more graphically oriented, Image Processing is becoming a widely acknowledged and very extensive field. One does not have to be a scientist or an engineer to be exposed to image processing. Its applications range from space exploration and remote sensing to document processing and desktop publishing. Even eyeglasses can be considered as a form of image processing.

Image acquisition and processing were once performed exclusively by industry. Image processing has been used for the processing of pictures returned from deep space, as a tool for investigations of Earth's resources, Earth-based astronomy, weather prediction, automated inspection, and robotics-to name just a few applications. The acquisition, manipulation, analysis, and display of images have required an extensive mathematical background to understand and apply the complex algorithms to which image data was typically subjected (Huss, 1999).

For years, the color image processing science confidently predicted that ubiquitous, high performance, powerful applications for enhancing images



was just around the corner. After several delays, this visual enhancing computing environment is finally a reality. It is now possible to improve the visual appearance of images to a human viewer and prepare images for measurement of the features and structures present, by just a click built in a highly powerful application.

The color processing software being built by using Java Swing deals with filtering images in two colors space to get the best possible filtered image. It may help to recall that colors image processing, like word processing, does not reduce the number of data present, but simply rearranges it. Some arrangements may be more appealing to the senses, and some may convey more meaning, but these two criteria may not be identical nor use identical methods.

1.2 Objectives

- Implementing a color image-processing program to provide the user with a basic knowledge of the fundamental techniques of color image filtering.
- Providing the user with an easy to use graphical user interface (GUI), where the user can filter images in two different color spaces, and define the better result for the application.
- Conversion of color image in two different types of formats RGB and YUV
- Generating a histogram of the image being filtered.

1.3 Software Methodology

1.3.1 Language and Platform

The programming Language chose to build with this software was Java. Java is a programming Language that is well suited for designing software that works in conjunction with the Internet. It is also an object-oriented programming language making use of a methodology that is becoming increasingly useful in the world of software design. Additionally, it is a cross-platform language, which means its programs can be designed to run the same way on Microsoft Windows, Apple Macintosh, and most versions of UNIX, including Solaris (Figure 1.1). Java extends beyond desktop to run on devices such as televisions, wristwatches, and cellular phones. JavaStation, Sun's network computer runs the JavaOS operating system and is optimized for the language, (Lemay, 1999).

1.3.2 Java's Architecture

Java's Architecture, shown in (Figure 1.2), consists of a two steps procedure, to run a program:

First, a programmer writes a Java source code and compiles it. The Java compiler (javac) creates an intermediate code, called Byte Code. The Byte Code is an intermediate code, not a specific code for a specific machine; it runs on an abstract machine called the Java Virtual Machine (Java), that is why the code can run in any Java enabled machine. It is sufficient for a

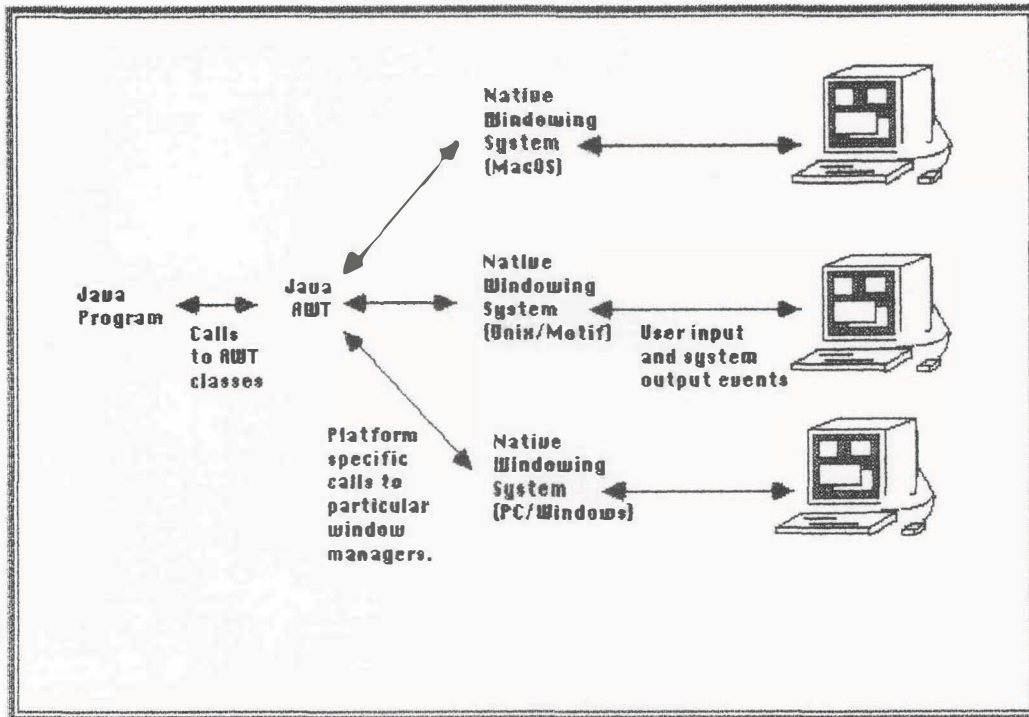


Figure 1.1: Java is a cross platform programming language.

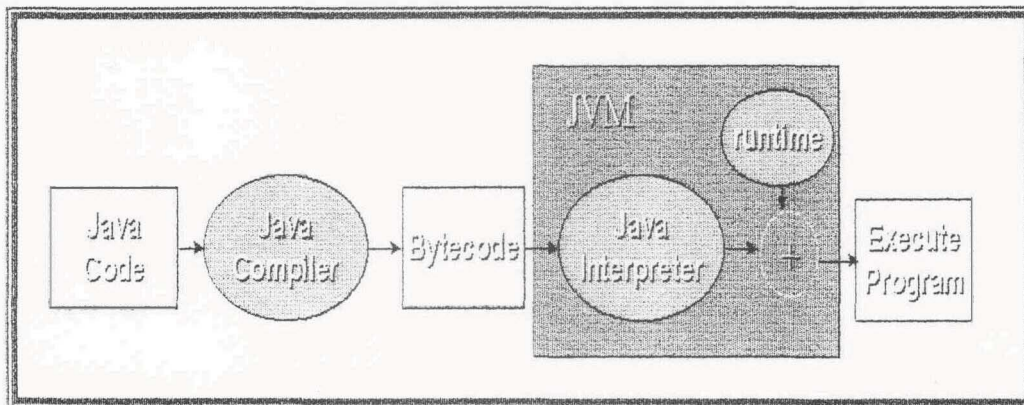


Figure 1.2: The Java Architecture