School of Science & Engineering
Department of Electrical & Computer Engineering

Non-Visual Representation of Complex Documents for Use in Digital Talking Books

Azadeh Nazemi

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

July 2015

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: Azadeh Nazemi

Date: 08/07/2015

Parts of this thesis have been previously published as listed below:

1. Azadeh Nazemi, Cesar Ortega-Sanchez & Iain Murray. (2011). Digital Talking Book Player for the Visually Impaired Using FPGAs: Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs, RECONFIG '11. (P.493-496): IEEE Computer Society Washington Conference Publications, DC, USA ©2011 ISBN: 978-0-7695-4551-6. doi>10.1109/ReConFig.2011.28

2. Azadeh Nazemi, Iain Murray & Nazanin Mohammad. (2012). Mathspeak: An audio method for presenting mathematical formulae to blind students: Proceeding of the 5th International Conference on Human System Interactions (HSI).(p. 48-58).

3. Azadeh Nazemi & Iain Murray. (2012). A Novel Complete Reading Embedded System for the Vision Impaired: Proceeding of the 3rd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2012). Singapore.(Best Student Paper Award Winner).

4. Azadeh Nazemi & Iain Murray. (2013). A method to provide accessibility for visual components to vision impaired. International Journal of human computer interaction. 4(1), 2013.

5. Azadeh Nazemi & Iain Murray. (2013). Mathematical Formula Recognition and Transformation to a Linear Format Suitable for Vocalization. International journal on computer science and engineering. 5(9).

6. Azadeh Nazemi & Iain Murray. (2013). An Open Source Reading System For Print Disabilities. International journal of Information Technology and Computer Science (IJITCS).12(2).

7. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2014). Layout Analysis for Scanned PDF and Transformation to the Structured PDF Suitable for Vocalization and Navigation. Computer and Information Science.7(1), 162-172. Publisher: Canadian Centre of Science and Education.

8. Azadeh Nazemi Iain Murray & David A. McMeekin. (2014). Multilingual Text to Speech in embedded systems using RC8660. International Journal of Computers & Technology.13(4), Publisher: Council For Innovative Research.

9. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2014). A Method to Provide High Volume Transaction Outputs Accessibility to Vision Impaired Using Layout Analysis. Transaction on machine learning and artificial intelligence.2(3), Publisher: scholar publishing.

10. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2014). Practical segmentation methods for logical and geometric layout analysis to improve scanned PDF accessibility to Vision Impaired. International Journal of Signal Processing, Image Processing and Pattern Recognition, August 2014 issue of IJSIP.

11. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2014). Mathematical Information Retrieval (MIR) from Scanned PDF Documents and MathML Conversion, Transactions on Computer Vision and Application. Volume 7(10.12.2014) IPSJ.

12. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2014). A Method to Implement DAISY Online Delivery Protocol. Proceeding of the 8th International Conference on Telecommunication System, Services, and Application (TSSA2014).| Kuta Bali, IEEE Conference Publications.

13. Azadeh Nazemi, Iain Murray & David A. McMeekin. (2015). Unbalanced Chemical Equations Conversion to Mark-up Format and Representation to Vision Impaired Students. International Journal Computer Applications in Engineering Education.23(2), 4/2015, Publisher: Wiley.

Acknowledgments

I express sincere appreciation and thanks to my Supervisor, Dr. Iain Murray who has been a tremendous mentor for me. He was the reason why I decided to go to pursue a career in research.I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my life have been priceless. I would also like to thank Co-supervisor, Dr. Cesar Ortega Sanchez for scientific advice and knowledge and many insightful discussions and suggestions and Associate Supervisor Dr. David McMeekin for helping me immensely. I would especially like to thank respondents of my surveys including CAVI students. I would like to thank Mr Russel Wilkinson for technical resources and help.

A special thanks to my family. Words cannot express how grateful I am to my husband, Pejman and my son Hiva for all of the sacrifices that they have made on my behalf. Your prayer for me was what sustained me thus far. My special thanks are due to my mother, father, brother and sister-in-law for their constant support and unconditional love.

I would also like to thank all of my friends, especially Chandrika who supported me in writing and help me to strive towards my goal.

Abstract

According to a World Intellectual Property Organization (WIPO) estimation, only 5% of the world's one million print titles that are published every year are accessible to the approximately 340 million blind, visually impaired or print disabled people. Equal access to information is a basic right of all people. Essential information such as flyers, brochures, event calendars, programs, catalogues and booking information needs to be accessible by everyone. Information helps people to make decisions, be involved in society and live independent lives. Article 21, Section 4.2. of the United Nation's Convention on the rights of people with disabilities advocates the right of blind and partially sighted people to take control of their own lives. However, this entitlement is not always available to them without access to information. Today, electronic documents have become pervasive. For vision-impaired people electronic documents need to be available in specific formats to be accessible. If these formats are not made available, vision-impaired people are greatly disadvantaged when compared to the general population. Therefore, addressing electronic document accessibility for them is an extremely important concern. In order to address the accessibility issues of electronic documents, this research aims to design an affordable, portable, stand-alone and simple to use "Complete Reading System" to provide accessible electronic documents to the vision-impaired.

Contents

Acknowledgments **ix**

Abstract **xi**

List of Figures **xix**

List of Tables **xxiii**

1 Introduction **1**

1.1 Introduction 1
1.2 Statement of the Problem 1
1.3 Aims of the Research 2
1.4 Outcome of the Research 2
1.5 Thesis Overview 3
1.6 Summary 4

2 Current State of Technology **5**

2.1 Introduction 5
2.2 Print Disability 5
2.3 Reading and Learning Methods 6
2.3.1 Different Types of Reading and Learning Methods 6
2.3.2 Learning Styles for the Sighted Print Disabled 7
2.3.3 Learning Styles for the Vision-Impaired 7
2.4 Traditional Alternate Access 8
2.4.1 Braille 8
2.4.2 Analogue Talking Books 8
2.5 Adaptive and Assistive Technologies for Vision-Impaired 9
2.5.1 Assistive Technologies based on Tactile Methods 9
2.5.2 Assistive Technologies based on Audio Methods 10

2.5.2.1 Screen Reader 10
2.5.2.2 Sonification 10
2.5.2.3 Digital Accessible Information System (DAISY) 11
2.5.2.4 electronic publication (ePub) 12
2.6 Documents Accessibility Issues 12
2.7 Categories of Electronic Documents Accessibility 13
2.8 Portable Document Format (PDF) 14
2.9 Categories of PDF Documents Accessibility 14
2.9.1 Scanned PDF 14
2.9.2 Structured PDF 15
2.9.3 Tagged PDF 15
2.9.4 PDF/Universal Accessibility 15
2.10 Components of PDF Documents 16
2.10.1 Accessibility of Non-Textual Components 16
2.10.2 Approaches for Accessibility of Non-Textual Components 17
2.10.3 Graphical Components Accessibility Requirements 20
2.10.4 Alternate Access Methods for Visual Printed Material . . . 20
2.10.5 Mathematical Expressions 21
2.10.6 Tactile Approaches for Accessibility of Mathematical Expressions 23
2.10.7 Audio Approaches for Accessibility of Mathematical Expressions Using Mark-up Formats 24
2.10.8 Accessibility of High Volume Transaction Output (HVTO) 24
2.11 Accessibilty Issues of Chemical Equation Presentation 26
2.11.1 Represent Chemical Equations to Vision-Impaired Students 26
2.11.2 Concepts of Chemical Equation and Balancing 26
2.11.3 Current Methods to Represent Chemical Equation to Vision-Impaired Students 27
2.12 Summary 28

3 Methodology **31**
3.1 Introduction 31
3.2 Objectives 31
3.3 Research Methodologies 32
3.4 Alternative Methodologies for Software Development 32
3.4.1 Waterfall 32
3.4.2 Agile 33

3.5 Agile Methodology and this Research 34
3.6 Summary 34

4 Requirements Analysis 35
4.1 Introduction 35
4.2 Requirement Analysis Using Survey Results 35
4.2.1 Accessibility of Electronic Documents 35
4.2.2 Communication with non-Textual Components in Electronic Document 41
4.3 Design Documents and Prototype Using Licensing Approaches . 47
4.4 Summary 49

5 System Components 51
5.1 Introduction 51
5.2 Overview of CRS Modules 51
5.3 DAISY Player 52
5.3.1 Introduction 52
5.3.2 Developed DAISY Player 53
5.3.3 Features of Developed DAISY Player 55
5.4 Creation of Accessible and Navigable Document from Scanned PDF 56
5.4.1 html OCR (hOCR) 56
5.4.2 Segmentation and Layout Analysis 57
5.4.3 Segmentation by Recognition by Adaptive Subdivision of Transformation (RAST) and Voronoi Methods 58
5.4.4 Preprocessing 59
5.4.4.1 Format Conversion 59
5.4.4.2 Convert to Binary 60
5.4.4.3 Scaling 60
5.4.4.4 Black Borders Removal 60
5.4.4.5 Margin Removal 60
5.4.4.6 Skew Detection and Correction 61
5.4.5 Block Segmentation 62
5.4.6 Text/non-Text Segmentation 64
5.4.7 Line Segmentation 66
5.5 Mathematical Information Retrieval (MIR) from Scanned PDF . . 70
5.5.1 MIR Overview 70
5.5.2 Accessible Formats 72

5.5.3 Mathematical Expression Accessibility Issues in Scanned PDF Documents 73
5.5.4 Previous Methods of Mathematical Formulae Extraction . 75
5.5.5 Global Line Labelling by Feature Extraction and SVM . . 76
5.5.6 EF Extraction Using Word Segmentation 81
5.5.7 Recursive Symbol Segmentation and Bracket Rule 83
5.5.8 Generating a Symbol Dictionary Using InftyMDB-1 86
5.5.9 Mathematical Symbol Recognition Using K-Nearest Neighbours (kNN) Based on Binary Vector 88
5.5.10 Symbol Layout or Structural Semantic Analysis 90
5.5.11 Merging Primitive Components and MathML Generation . 92
5.5.12 Rendering Math-TEX Family with MATHSPEAK 97
5.6 Mathematical Graph Accessibility with MathGraphReader 100
5.6.1 Current Method 101
5.6.2 Proposed Method 101
5.6.2.1 Examined Graphs 102
5.6.2.2 General Concepts 104
5.6.2.3 Data Extraction 104
5.6.3 Testing and Evaluation 109
5.7 Not-in-Order Components: Tables and TableReader 110
5.7.1 Tables Categories 112
5.7.1.1 Regular table 112
5.7.1.2 Irregular table 114
5.8 High Volume Transactional Output (HVTO) Segmentation . . . 116
5.9 Non-Textual (Graphical) Components Accessibility 118
5.9.1 ChartRecognition 118
5.10 Examined Non-textual Components in this Research 123
5.10.1 Bar Chart and BarChartReader 123
5.10.2 Line Chart and LineChartReader 128
5.10.3 Pie Chart and PieChartReader 130
5.10.4 GNUPLOT Evaluation Tool 132
5.11 A Method to Present Chemical Equations to Vision-Impaired Students 134
5.11.1 Representing Chemical Equation in Markup Format . . . 134
5.11.2 Species Classification and Reactants/Products Extraction . 135
5.11.3 Chemical Elements Extraction and Symbol Replacement . 136

5.11.4 Calculation of Total Quantity of Element at Left and Right Side and Comparison . . . 137
5.11.5 Insert Unknown Coefficients Before Species Including Reactants and Products . . . 140
5.11.6 Defining Algebraic Equations to Obtain Coefficients and Balance Chemical Equation . . . 141
5.11.7 Tagging Classified Information and Markup Format Generation . . . 141
5.12 Summary . . . 143

6 Implementation of Hardware Platform 145
6.1 Introduction . . . 145
6.2 FPGA Prototype . . . 145
6.2.1 FPGA Prototype Requirements . . . 145
6.2.2 FPGA DTB Books Player Functionality . . . 146
6.2.3 FPGA DTB Player Hardware Components . . . 146
6.2.4 FPGA DTB Player . . . 147
6.3 Embedded Platform for Reading System . . . 150
6.4 User Interaction Methods . . . 152
6.4.1 Introduction . . . 152
6.4.2 Speech Recognition . . . 152
6.4.3 Customised Tactile Keypad . . . 154
6.4.4 Users Feedback about USB keypad . . . 156
6.4.5 Joysticks . . . 157
6.4.6 Text to Speech (TTS) . . . 158
6.4.7 Braille Terminal . . . 160
6.5 Summary . . . 162

7 Testing for Evaluation and Usability 163
7.1 Introduction . . . 163
7.2 Block Segmentation . . . 163
7.3 Chart Recognition . . . 164
7.4 BarChartReader . . . 167
7.5 PieChartReader . . . 169
7.6 LineChartReader . . . 170
7.7 MathGraphReader . . . 172
7.8 MATHSPEAK . . . 175

7.9 Alternative Text Description for Chemical Equation 180
7.10 Mathematical Handwritten Documents Segmentation Results . . . 183
7.11 Summary 183

8 Conclusion 185

8.1 Future Work 193
8.2 Final Reflection 194

List of Figures

2.1 The PHANToM (Murray, 2008) 19
2.2 2-Dimensional mathematical equations contain subscripts and superscripts 22

4.1 Survey participants age group (left) and survey participants gender (right) 36
4.2 Respondents' perception of the most common document format . 37
4.3 Respondents' Most preferred format 38
4.4 Important factors in selecting a document format 39
4.5 Accessibility Tool 40
4.6 Most inaccessible graphical components 42
4.7 Frequent encounter inaccessible graphical representations 43
4.8 Best perceived accessibility method for graphical components . . . 44
4.9 Navigation importance for access to graphical components 45
4.10 Preferred type of tactile method 45
4.11 Frequent usage of Haptic device 46

5.1 Overview of CRS modules 52
5.2 Three DAISY standards comparison 54
5.3 Steps of reading session by developed DAISY player 55
5.4 An image before (left) and after (right) margin removal 61
5.5 Rotated PDF document 62
5.6 Top-left: a multiple columns document including sidebar. Top-right: the result of using Voronoi segmentation. Bottom-left: a multiple columns document. Bottom-right result of using Horizontal/Vertical segmentation (blocks are specified with blue lines separator) 64
5.7 Left to right: A sample Binary-Block-Segment, dilated, non-Textual and Text-Only images 65
5.8 Text/non-Text Segmentation 66
5.9 Different types of lines in mathematical documents:Top: Text, Middle: EF and Bottom: IF 66

5.10 A Fully-Segmented EF line 67
5.11 A 2-D IF line (top) and over-segmentation result (middle and bottom) 68
5.12 Two 2-D IF lines (top) and under- segmented IF 2-D line result (bottom) 68
5.13 EF 2-D line (top) and broken symbol (bottom) 68
5.14 Mathematical PDF document (top) and its line segmentation result (bottom) 70
5.15 Overview of MIR 72
5.16 A sample of mathematical document (left) and extracted features of its segmented lines (right). These features are: left margin (LM), right margin (RM), height (H), width (W), vertical space (VS) and aspect ratio (AR) 77
5.17 Threshold value=Mode(x) (left), Threshold value=Median(x)(right) 78
5.18 Global Line Labeling 80
5.19 An EF line sample (top) and word segmentation result of the sample line (bottom). 82
5.20 Voronoi result for sample of EF 84
5.21 VCE and HCE result for a sample 85
5.22 Slopes between two adjacent symbols 90
5.23 Relationship between two adjacent mathematical symbols Height-Ratio 91
5.24 8-different relationships between a symbol and its neighbours . . . 91
5.25 A Mathematical graph and its similar product by GNUPLOT . . 109
5.26 From top to bottom: Illustration of a regular table (Binary-Table.png), vertical borders (vertical.png), horizontal borders (horizontal.png), vertical/horizontal borders intersection (intersection.png) and the values of cells bounding boxes 114
5.27 Sample of an irregular table and its cells bounding box values . . 116
5.28 HVTO Segmentation: The sample of a bill (British Gas, 2013) (left) and result of horizontal/vertical segmentation (right) 117
5.29 Chart-Legend Segmentation Results: Left: from top to bottom a sample of Pie Chart, Pie-Chart-Only and Legend-Only images. Right: from top to bottom a sample of Bar Chart, Bar-Chart-Only and Legend-Only images 120
5.30 Chart classification 122
5.31 Horizontal/Vertical lines removal and binarizing result for samples of pie chart and bar chart. Left: top to bottom, Pie-Chart-Only and its Binarized-Removed-Horizontal-Vertical-Lines. Right: top to bottom Bar-Chart-Only and its Binarized-Removed-Horizontal-Vertical-Lines 123

5.32 Horizontal/Vertical Axes Segmentation 126
5.33 a) Bar chart; b) Chart-only; c) Legend-only; d) Vertical axis; e) Horizontal axis; and f) Binary-Bars-Only images by BarChart Reader. 127
5.34 a) A line chart sample; b) Horizontal axis; c) Vertical axis; d) Horizontal grid lines removal result; and e) Marker-Only images by LineChartReader 130
5.35 From left to right: a) A sample of Pie Chart; b) Pie-Only; c) Legend-Only; d) Transparent-Background-Pie-Only; and e) Color-Reduced-Ttransparent-Pie-Only 132
5.36 A sample line chart (left) and generated line chart by GNUPLOT based on data extracted by LineChartReader (right) 133
5.37 A sample bar chart (left) and generated bar chart by GNUPLOT based on data extracted by BarChartReader(right) 134
5.38 Algebraic finding to balance three samples 143

6.1 Finite state machine for PmodDA2 148
6.2 FPGA-based embedded system for DTB player 150
6.3 Block diagram of customized board for reading systems 151
6.4 Arduino Hex Keypad Schematic 155
6.5 CRS customised keypad 156
6.6 Mobile keypad arrangement 157
6.7 NLS Digital Talking Book Player (Maine State Library, 2013) . . 157
6.8 Reading system connected to Braille terminal 162

7.1 Engineering drawing 164
7.2 Recognised as pie chart 165
7.3 Recognised as bar chart 165
7.4 Recognised as line chart 166
7.5 Examined bar charts by BarChartReader. Maximum / Minimum Bars, total number of bars were propery recognised. 168
7.6 Bar chart samples which were examined by BarChartReader . . . 169
7.7 Sample of Pie charts,their slices percentage and equivalent bar charts generated by PiechartReader 170
7.8 Line Chart Samples processed by LineChartReader 172
7.9 Sample1 Mathematical graph 173
7.10 Sample2 Mathematical graph 174
7.11 Sample3 Mathematical graph 175
7.12 Sample4 Mathematical graph 175
7.13 Handwritten mathematical sample (left), line segmentation result (middle) and character segmentation of two lines (right). 183

8.1 Scanned PDF processing with CRS 192

List of Tables

4.1 Results summary for survey 1 41
4.2 Results summary for survey 2 47

5.1 hOCR Existing tags and recommended tags by this research . . . 57
5.2 Mathematical Symbol Categories Based on symbol Aspect Ratio . 88
5.3 Predefined size for three symbol categories 89
5.4 Different relations between two adjacent symbols in mathematical expressions 92
5.5 Graph location in plane 107
5.6 Supported functions with GNUPLOT and MathGraphReader application 108
5.7 Evaluation responses 110
5.8 Category 1 of symbols for chemical elements 137
5.9 Category 2 of symbols for chemical elements 138
5.10 Algebraic equations 141
5.11 Mark-up Format Tags 142

6.1 Push button values for different functions 147
6.2 Basic Functions in VOCA file in Voice Recognition System for CRS 153
6.3 Using joystick as a user interface 158
6.4 TTS control commands 160

7.1 Chart recognition features 167

8.1 Research objectives, proposed methods and used techniques -1 . . 188
8.2 Research objectives, proposed methods and used techniques-2 . . 189
8.3 Research objectives and proposed methods to achieve them-3 . . . 190

List of Abbreviations

ASCII American Standard Code for Infor mation Interchange

ASTER Audio System for Technical Readings

BNF Backup Normal Form

CRMFS Compressed ROM File System

CRS Complete Reading System

DAC Digital to Analogue

DAISY Digital Accessible Information System

DAS Document Accessibility Services

DFA Deterministic Finite Automaton

DFU Device Firmware Update

dpi Dot Per Inch

DRM Digital rights management

DTB Digital Talking Book

DTE/DCE Data Terminal Equipment/Data Circuit Equipment

ePUB Electronic Publication

FPGA Field Programmable Gate Array

FTP File Transfer Protocol

GPIO General Purpose Input/Output

HMM Hidden Markov Model

HTML Hyper Text Markup Language

HVTO High Volume Transactional Output

ICAD the International Community for Auditory Display

I/O Input/Output

kNN k nearest neighbours

LVCSR Large Vocabulary Continuous Speech Recognition

MAC Media Access Controller

MathML Mathematical Markup Language

MIR Mathematical Information Retrieval

MMU Memory Managment Unit

MPC Magick Persistent Cache image file format

NCX Navigation Center eXtended

NLS National Library Service

OCR Optical Character Recognition

OPF Open Packaging Format

OS Operating System

PCM Pulse Code Modulation

PDF Portable Document Format

PLB Processor Local Bus

PPM Portable Pixel Map

RAST Recognition by Adaptive Subdivision of Transformation

SIGHT Summarizing Information Graphics Textually

SMIL Synchronised Multimedia Integration Language

STEM science, technology, engineering and math

SBC Single Board Computer

SVM Support Vector Machine

TFTP Trivial File Transfer Protocol

TTS Text to Speech

UART Universal Asynchronous Receiver Transmitter

VEM Visual Extraction Module

WHO World Health Organisation

XED Structured Electronic Document

XCDF Structured Canonical Document Format (XCDF)

XHTML Extensible Hyper Text Markup Language

XML Extensible Markup Language

Chapter 1

Introduction

1.1 Introduction

Today technology advancement leads to pervasive usage of electronic documents. These electronic documents include a wide range of daily life information for people such as flyers, brochures, news, statement presentment in the financial services, insurance, utilities, and government sectors such as bills or bank statements and learning materials such as reference books or user manuals. Thus, the accessibility to electronic documents for vision-impaired is a noteworthy concern which is considered the major problem that this research will address.

1.2 Statement of the Problem

The right to access information is internationally recognised in Article 21 of the United Nation Convention on the Rights of Persons with Disabilities (United Nations, 2006). According to the World Health Organisation (WHO) statistics updated by October 2013, 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 million have low vision. Approximately 90% of visually impaired people live in developing countries (Geoffrey et al., 2010) and 65% of them are aged 50 or older. This age group comprises approximately 20% of world's population. There are 19 million visually impaired children, out of which 1.4 million are irreversibly blind for the rest of their lives and need social, vocational, economic and educational support (WHO, 2013). They experience great difficulty in accessing printed material. Having access to information and printed resources is an essential requirement for independent and constructive lives. In addition, more people will be at risk of age-related visual impairment with increasing elderly population in many countries (WHO, 2011). The lack of accessible information is one of the most critical obstacles for vision-impaired students to complete post secondary education and undertake academic studies in science, technology and engineering fields which may lead to lower career success.

1.3 Aims of the Research

This research is focused on the accessibility of printed and electronic documents including websites, books, invoices, letters and leaflets to a broad, vision-impaired audience. It aims to study the issues in accessing the most commonly used electronic documents to formulate the definition of the research problem and address the accessibility issues of different components of electronic documents such as charts, graphs, tables, mathematical expressions and chemical equations.To achieve this goal design and implement of Complete Reading System (CRS) for vision-impaired was considered. The CRS is a fully functional, simply operated, standalone, and affordable system which provides opportunity to access, navigate and bookmark documents through reading sessions for the vision-impaired. CRS supports vision-impaired users specifically students in high school and high level education to access reference books and user manuals just like sighted people by using navigation, bookmarking and searching abilities. The CRS is a modular software system.

1.4 Outcome of the Research

An initial prototype has been designed and implemented for the system to provide accessibility to the most commonly used electronic documents for vision-impaired people during this research. In designing the CRS, efforts have been made to offer electronic document accessibility and fully featured multimedia reading experience to vision-impaired people, even for those with little or no computer experience. The CRS is an embedded device with a software application which provides features and flexibility, and a hardware platform that provides cost effective performance without the requirement for expensive computers or smart phones. The findings support making a standalone, low cost and affordable CRS for vision-impaired that:

- Enables electronic documents such as HTML, PDF, POSTSCRIPT, plain text, DOC, DOCX, and ODT to be accessible to vision-impaired using audio method;
- Plays Digital Accessible Information System (DAISY) with navigation, searching and bookmarking abilities for reading user manuals, reference books and encyclopedias ;
- Performs layout analysis;
- Runs multilayer segmentation on document as follows:
    1. Block segmentation
    2. Text/non-Text segmentation

3. Line segmentation
4. Word segmentation
5. Character segmentation;

- Generates comprehensive text descriptions from non-textual components and converts text descriptions to accessible audio output;
- Keeps author reading order or structure of not-in-order components such papers with more than one column;
- Converts mathematical formula images to mark-up format using Mathematical Optical Character Recognition (MOCR);
- Converts High Volume Transaction Output (HVTO) to accessible formats for vision-impaired customers; and
- Converts text format of chemical equations to alternative mark-up and audio description to help vision-impaired students to balance chemical equations.

1.5 Thesis Overview

The structure of the thesis consists of 8 chapters as follows, Chapter 2 provides a literature review of current knowledge, findings, definitions and theoretical and methodological contributions to the electronic documents accessibility to vision-impaired users. Chapter 3 compares the Waterfall and Agile methodologies as two methods for software development. Chapter 4 presents results of two surveys conducted among vision-impaired people to perceive their accessibility to electronic documents and communication with non-textual components. Chapter 5 explains different software modules as approaches to solve issues regarding electronic document accessibility. Chapter 6 describes hardware implementation methods for the Complete Reading System followed by finding and driving appropriate input/output interface to communicate system and users. Chapter 7 includes outcomes of applying the developed modules in Chapter 5 for random samples to evaluate the system.Chapter 8 recapitulates the research that has been carried and Appendix B contains codes of developed software modules in Chapter 5.

To demonstrate the contributions of this research complete alternative text descriptions were provided for all figures in this thesis to be accessible to vision-impaired users who utilize screen readers.

1.6 Summary

Electronic documents may contain non-textual components (such as charts and graphs), non-linear components such as tables, not-in-order (papers with more than one column), and multidimensional components (mathematical expressions and chemicals equations). These documents have a number of accessibility issues for those who are vision-impaired and using assistive technology. The next chapter is the literature review of established knowledge and ideas about electronic document accessibility issues and the current solutions to address them.

Chapter 2

Current State of Technology

2.1 Introduction

This chapter includes the current knowledge, findings, definitions and theoretical and methodological contributions to electronic document accessibility for vision-impaired users.

2.2 Print Disability

George Kerscher coined the term print disabled (circa 1988-1989) to describe persons who could not access print (Kendrick, 2001). The definition is as follows: "A print-disabled person is a person who cannot effectively read print because of a visual, physical, perceptual, developmental, cognitive, or learning disability" (Reading Rights Coalition, 1989). A print disability prevents a person from gaining information from printed material in the standard way and requires them to utilize alternative methods to access that information. Print disabilities include visual impairments, learning disabilities, or physical disabilities that impede the ability to manipulate a book in some way (Learning Ally, 2012).

Additionally, the Higher Education Opportunity Act defines print disabled as a student with a disability who experiences barriers to accessing instructional material in non-specialized formats (Title 17 of the Copyright Act - Council for Exceptional Children, 2008).

The Google Library Project Settlement defines print disabled as a user who is unable to read standard printed material due to blindness, visual disability, physical limitations, organic dysfunction or dyslexia (Reading Rights Coalition, 1989).

The reasons for print disability vary but may include:

- Vision impairment or blindness;

- Physical dexterity problems such as multiple sclerosis, Parkinson's disease, arthritis or paralysis;
- Learning disabilities, such as dyslexia brain injury or cognitive impairment;
- Literacy difficulties; and
- Early dementia (Vision Australia, 2012).

2.3 Reading and Learning Methods

In active learning environment learners can restructure the new information and their prior knowledge into new knowledge about the content and to practice using it such as visual aids, demonstrations or integrated into class presentation learners. In passive learning area can get only what they are told such as listen to tape recorder (Herr, 2006). Study by and for educators identify three basic styles of learning (Barbe et al., 1979). These three styles are used by sighted, print disabled and vision-impaired differently.

2.3.1 Different Types of Reading and Learning Methods

The three basic learning styles are:

- Tactile/kinetic: This type of learner learns best through moving, doing and touching. They prefer hands-on approach, actively exploring the physical world around them. They remember best what was done, but may have difficulty recalling what was said or seen. They need direct involvement and may find it hard to sit still for long periods and may become distracted when not able to move; by their need for activity and exploration. These learners often need to take frequent breaks or move around when bored (Baxter, 2013).
- Auditory: Auditory learners learn best through listening lectures, discussions, talking things through and listening to what others have to say. Auditory learners focus in on tone of voice, pitch, speed and other aspects of verbal presentations. Written information may have little meaning until it is heard. These learners prefer to sit where they can hear, but may not pay attention to what is happening up front. They may talk to themselves or others when bored. Auditory learners often benefit from reading text aloud and using a tape recorder. The learners using this style would rather listen to things being explained than read about them. However, other noises may become a distraction resulting in a need for a relatively quiet place (Baxter, 2013).

- Visual: This category of learner learns best by looking at graphics, watching a demonstration, or reading. For the visual learner, it is easy to look at charts and graphs, but they may have difficulty focusing while listening to an explanation. Visual learners often think in pictures and prefer graphical representations of concepts through charts, diagrams, or tables (Baxter, 2013).

Although most people use a combination of these three learning styles, there is usually a clear preference for one (Manktelow & Carlson, 2014). Knowing and understanding the types of learning styles is important for students of any age. A balanced, intelligent child is able to develop all three types of learning styles (Felder & Brent, 2005). Just because a child has dominant learning style does not mean that the other learning styles cannot be improved (Cleaver, 2011). People may have to adapt to new learning styles as their lifestyles change. For example, a visual learner who is experiencing the effects of ageing on their eyesight may need to shift toward the auditory learning style. Conversely, a youngster who has successfully learned through hands-on, tactile methods may need to adapt to the visual and auditory learning as they enter higher education.

2.3.2 Learning Styles for the Sighted Print Disabled

Dyslexia is a type of sighted print disability, which causes difficulties with specific language skills, particularly reading. Students with dyslexia may experience difficulties in other language skills such as spelling, writing and speaking. Moreover, people with dyslexia have problems with discriminating sounds within a word, which is a key factor in their reading. It is important for these individuals to be taught by a method that involves several senses (hearing, seeing, touching) at the same time. They can benefit from listening to books-on-tape and from writing on computers (Ramus, 2003; Silverman, 2000).

2.3.3 Learning Styles for the Vision-Impaired

Individuals cannot be categorized into these three simple learning styles, they may require a combination of two styles to understand and comprehend new material (James & Gardner , 1995). While all students learn in three methods: visual, audio and tactile or kinaesthetic methods, most blind and vision-impaired students learn by both audio and tactile methods (Gardier et al., 1997). Students who are blind or vision-impaired often prefer using printed course materials that have been converted into Braille, large print and digital recordings (Miner et al., 2001).

2.4 Traditional Alternate Access

Since learning styles for the vision-impaired are tactile and audio methods, printed documents need to be available in specific alternate formats. Traditionally alternative formats such as Braille, large print or audio are used by the vision-impaired to read material.

2.4.1 Braille

In 1829 Louis Braille the inventor of the Braille system stated that “Braille is knowledge, and knowledge is power”. Braille is a tactile form of reading and writing used by people who are blind or vision impaired. It is based on a six dot cell with two columns of three, like the six on a dice. The dots in the first column are numbered 1, 2 and 3 from the top down; and the dots in the second column are numbered 4, 5 and 6 from the top down. By using any number of these six dots 63 different patterns can be formed (Vision Australia, 2012)

It is a method of writing words, music and plain song by means of dots, for use by the blind and arranged by them (Sullivan, 2002). The range of paper-based Braille material is limited. Braille books, whilst of great value, particularly in conveying mathematics or musical score, are heavy and bulky. For the vision-impaired person over the age of 85 has newly acquired sight loss, the adaption to a new way of reading may be difficult and learning Braille may be problematic (European Blind Union, 2011). Effective use of Braille requires tactile sensitivity and all vision-impaired people may not have this sensitivity (Murray, 2008).

2.4.2 Analogue Talking Books

One of the most common ways of the audio method is using traditional talking book, which is an analogue representation of a print publication, usually human read and recorded on cassette tape or record (Friedmann, 2008). These media suffer several significant disadvantages.

- Distribution of physical media is costly;
- They are easily damaged and wear over time; and
- They offer only sequential access to information and extremely limited navigation. This navigation is sometimes available but needs cassette number, side and tone number.

2.5 Adaptive and Assistive Technologies for Vision-Impaired

Print disability occurs in different forms and ranges of severity (Miner et al., 2001). Consequently, assistive technology which includes any device, piece of equipment or system to help them must be flexible. Assistive technology provides alternative strategies to compensate for weakness and capitalize talents. Print material needs to be reproduced in a format such as Braille, large print hard copy, sound recording of text narration, or digital files for people with a print disability (Australian Copyright Council, 2007).

A wide range of adaptive and assistive software and devices are available for people with a print disability. Many adaptations are simple and readily available (University of Washington, 2013). Most technologies use tactile and/or audio methods to convey information.

A screen magnifier can make monitors easier to read (Macular Degeneration Foundation, 2012). Software reconfiguration or special software can reverse images on computer screens from the conventional black on white to white on black (or other combinations) for individuals who are light-sensitive. For many partially sighted readers, well-designed print information using a minimum 12-point size font on good quality non-shiny paper makes the document accessible (European Blind Union, 2011). Additionally; accessibility features in computer operating systems and other programs are useful for those with vision disabilities (National Center Accessible Information Technology Education University Washington, 2013). Some of the available computer programs to support electronic documents accessibility are:

- Text to Speech using computer-synthesised voice and screen readers can be used to read text on the screen such as Voice Over for Macintosh (Schwarzenegger, 1991) and JAWS for Windows (Henter, 1989);
- Text to Braille, translate text into refreshable Braille such as Duxbury in Windows (Gilda et al., 2014) and liblouis and libbrlapi using Brltty in Linux (Boyer, 2002);
- Screen magnifier software to enlarge text on the screen; and
- Text recognition applications such as Optical Character Recognition (OCR).

2.5.1 Assistive Technologies based on Tactile Methods

Braille embossers provide hard copy output which is too bulky. In addition the Braille embosser is noisy and expensive. Nowadays, Braille users can read computer screens and other electronic devices using a Braille Terminal or refreshable Braille display. A refreshable Braille display is a piece of hardware that provides

Braille output from computer input. The 8-dot refreshable Braille cells change, or refresh, according to the part of the screen that has the computer's attention and is used with screen reading software, such as JAWS for Windows (Assistive Technology and Accessibility Centers , 2007). Refreshable Braille displays provide word-by-word translation of text on the screen into Braille on a separate display. The display reproduces words in the format of vertical pins that raise and lower to form Braille characters in real time as the text is scanned (Miner et al., 2001). Picture in a Flash (PIAF) is a Tactile Graphic Maker which makes raised line drawings on special paper. Tactile graphics, including tactile pictures, tactile diagrams, tactile maps and tactile graphs, are images with raised surfaces to convey non-textual information such as maps, paintings, graphs and diagrams. Production of tactile maps is one of the most common uses for tactile graphics (McCallum et al., 2005).

Adaptive software and devices, such as refreshable Braille display can be extremely expensive. Even people using free or inexpensive software need computer equipment with adequate memory, processing speed, as well as assistance with learning new software. Thus people who are not in full-time employment, may not be able to afford such equipment. Consequently many of them are unable to use most of sophisticated adaptive technology (Australian Copyright Council, 2007).

2.5.2 Assistive Technologies based on Audio Methods

2.5.2.1 Screen Reader

Screen readers convert screen information to audio using computer speech synthesizers and allow vision-impaired people to use the computer, create a document using a word processor like MS Word, read any article on the internet, communicate through instant messaging software, create a blog post and write an email.

Vision-impaired people listen to a screen reader reading the text displayed on the screen (Theofanos & Redish, 2003).

Screen reader users do not usually have the chance to learn the correct spelling of certain words, especially uncommon ones such as medical terms. They can make a screen reader read a word character by character after they hear a word that they do not know the spelling but it is very time consuming and in some cases, they cannot recognise S's and F's (Bohman, 2014).

2.5.2.2 Sonification

Sonification or Non-speech audio is also used to convey information or perceptual data. It is used to represent the behavior of the graphed equation by presenting mathematical concepts in a different way. Sonification is useful for all students regardless of disability or learning difficulties in a web browser or reading tool that supports audio (Hermann et al., 2011). This auditory perception has advantages in temporal, amplitude and frequency resolution that open possibilities as an alternative or complement to visualization techniques (Kramer, 1993). A Geiger detector which conveys information about the level of radiation or a church bell which conveys the current time are very basic examples for Sonification (Flowers, 2005). Though many experiments with data Sonification have been explored in forums such as the International Community for Auditory Display (ICAD), Sonification faces many challenges to widespread use for presenting and analysing data (Brown & Brewster, 2003). Many Sonification attempts are coded from scratch due to the lack of a flexible tool for Sonification research and data exploration (Matheson, 2013). In some cases, it is difficult to provide adequate context for interpreting data of Sonification. Sonification has the potential to bring computing to a new level of naturalness and depth of experience for the user. Sonification requires that interfaces use audio in the first place. It faces the problem that certain interfaces perform poorly at the outset and may need more user engagement with practicing and longer learning period. To evaluate Sonification , instead of comparisons of interactive visual versus interactive auditory displays. Possibly, the better way is thinking of that whether the interactive sound can improve a user's performance in a combined audio visual display (Hermann et al., 2011).

2.5.2.3 Digital Accessible Information System (DAISY)

DAISY is an international standard for digital talking books and multimedia representation of print publication that can be either human read, utilize synthetic speech or contain both to support audio methods (Leith, 2006). Unlike analogue talking books, an important feature of DAISY books is the easy and rapid navigation ability within sequential and hierarchical structures consisting of marked-up text by such elements as sentence, paragraph and page including specific page numbers. DAISY synchronises text with audio that allows the user to navigate via multiple levels, search, bookmark, annotate, and alter playback speed as well as many other features (Kerscher, 2003). Using the DAISY, people with a print disability can locate particular chapters or page numbers in a digital text or sound recording file almost as easily as a sighted person (Russo, 2010). This standard is increasingly being adopted internationally (Tank & Kerscher, 2007).

By synchronising audio, images and text, DAISY multimedia can address the needs of each type of learner. Full-text/Full-audio DAISY books synchronise the audio playback with written text displayed on a computer screen to the benefit of visual learners. Easy navigation of information produced in DAISY offers tactile/kinetic learners the opportunity to:

- Explore documents;
- Interact with information;
- Retain attention; and

- Improve learning skill.

A DAISY player is a device for people with print disabilities to read, search, navigate, annotate and bookmark materials such as novels, reference books and user manuals. DAISY players are implemented in either hardware or software. Hardware DAISY players, like CD players or MP3 players, can be of great assistance to auditory learners who benefit from audio playback, whether presented through a text-to-speech feature or human narration. Hardware DAISY players offer better portability and can access online content and download books, streamed over the internet, or copied into player via a USB port. Software DAISY players enable DAISY books to be played on a computer or mobile devices such as iPads and mobile phones (Nazemi et al., 2014). Different DAISY players offer different functionality levels. Some are very basic, only offering access to the audio and navigational structure of the DAISY book while other players offer enhanced functionality, such as the ability to search text, record audio notes for future reference, synchronise audio and text during play time and view text on the screen, which may be helpful for people with dyslexia. Both hardware and software playback devices have significant drawbacks. Hardware players tend to be expensive due to the relatively limited market (compared to mainstream consumer devices) and software requires the user to be competent with a computer.

2.5.2.4 electronic publication (ePub)

ePub is a standard for digitized text and publishing an enhanced feature eBook. Semantic markups make it possible to access scientific and mathematical expressions in more meaningful and useful way. ePub uses Structure in contents, image descriptions and alt text and MathML for mathematical expressions. Each section in the document is marked up using appropriate styles such as h1, h2 . So it has a correct hierarchy of sections and page numbers in precisely the same manner as Daisy. ePub uses proper and complete markup for text and tabular data. Images and content embedded in image which are not accessible to vision-impaired has a description, caption or alt text . ePub borrows heavily from the DAISY Standards and W3C & Web Accessibility Initiative (WAI) specifications to the point where the ePub and Daisy standard organisations are working towards a merging the two very similar standards in the near future. All of the features in ePub3 reading systems have been part of DAISY readers since version.

2.6 Documents Accessibility Issues

Most print disability organisations use various forms of software to produce copies in accessible formats. Some organisations may ask the publisher for a digital file of the work, or may use other methods for getting a digital file, such as scanning the text. Publishers in Australia are not legally obliged to supply digital files (Australian Copyright Council, 2007). Access to the digital format greatly reduces the time and expense of converting a text into Braille or other alternative formats. Where a digital file has been provided in an image format such as scanned PDF or Quark (Barrett, 2007), the print disability organisation must extract the text using specialised software. This process generates errors such as incorporation of page numbers into text, substitution of letters and words as well as displacement of sections of text. Therefore, the editing process required after text extraction consists of:

- Error corrections;
- Checking for correct reading order;
- Writing a text content description of an illustration or diagram;
- Incorporation of text and visual cues; and
- Formatting.

Thus, the editing task is time consuming and needs an editor to work on each file. This task is sometimes outsourced to specialist contractors, for subjects such as mathematics and science. Where symbols are used, editors must have detailed knowledge of the subject. In such a case, the original file produced by the publisher and the edited file delivered to vision-impaired individuals by a print disability organisation usually look quite different. In some cases, results are not visually attractive, even hard to use for sighted people and unreadable by blind or partially sighted people (Vystrcil et al., 2011).

2.7 Categories of Electronic Documents Accessibility

In terms of accessibility to assistive technology, electronic documents have been divided into three categories:

- Accessible and navigable due to containing mark-up tags such as DAISY, ePub, HTML, tagged PDF. Information that was previously unavailable to blind people such as newspapers, encyclopedias or telephone directories, becomes now accessible on the internet using HTML (European Blind Union, 2011). The barrier of access to journals has been significantly reduced with the immediate online availability of journals in electronic formats like HTML and PDF files, which can be read by screen output programs.
- Accessible due to being text or text convertible but not navigable such as plain text, structured untagged PDF documents.
- Not accessible, nor navigable, requiring image processing and/or OCR to extract text such as images and scanned PDF documents.

2.8 Portable Document Format (PDF)

PDF is a common way for organisations to publish documents, the most used electronic format for online presentations and the most popular after HTML file (Harris, 2001). PDF documents preserve fonts, images, graphics and layout of any source document and are ideal for printing exactly as the author intended. Text information, pictures and signatures can be scanned into a PDF and easily emailed to recipients. Upon the document's arrival, the receiver can open and view it using a vast array of different PDF viewing applications such as Adobe Reader and Apple Preview. As one of the most common digital document, PDF has historically been a major challenge for the vision-impaired community. A lack of standards and the growing variety of PDF export programs, which do not support tagging, has created major accessibility issues for this community (Wild, 2010). PDF content is presented to assistive technology as a textual representation of the document. Accessibility of PDF documents by assistive technologies depends on the manufacturers of those technologies incorporating PDF support into their products. Several manufacturers have done this with recent versions of their products, but for the many users of earlier versions of the technology, the PDF will remain inaccessible (Hudson, 2004). In addition secured PDF is another issue regarding PDF accessibility. During PDF creation, some authors add restrictions to prevent users from printing, copying, extracting, commenting or editing text and these restrictions can interfere with a screen reader's ability to convert text to speech. Digital Rights Management (DRM) technology that publishers use to control or restrict digital media content on electronic devices can have negative impact on access by users with vision impairment. As a result, documents may not automatically be accessible to screen readers and may require conversion tools (Adobe Systems Incorporate, 2008). In spite of recent changes by Adobe Acrobat in order to optimize accessibility of the secured PDF for assistive technology, it seems many are still inaccessible. Content authors are rarely familiar with the requirements of publishing with successful accessibility and have little reason to learn (Johnson, 2004).

2.9 Categories of PDF Documents Accessibility

2.9.1 Scanned PDF

Scanned PDF documents represent the most inaccessible type of document in terms of accessibility to users of assistive technology. The scanned PDF document is an image file and contains images of text not the real text. Text on the page is not searchable or selectable. To the assistive technology user, the document appears completely blank, tags are not available and images are not identified through alternative text (ALT-text) which conveys the same essential information as the image. Although the page can be viewed with a PDF viewer, screen readers cannot recognise the content. Since the scanned PDF is an image format, it is inaccessible to assistive technologies such as a screen reader which reads plain text. Therefore, the information retrieval requires Optical Character Recognition (OCR) (O'Brein, 2012). The OCR software scans the scanned PDF file and through text extraction generates an editable text formatted document. To make a scanned document accessible, it must be converted from the image of the document into selectable and scalable text. This text document can then be edited, formatted, searched and indexed as well as translated or converted to speech. A problem that the OCR software does not solve is the accurate regeneration of the full text layout. Text obtained from OCR comprises of unexpected segments and these segments may be out-of-order in terms of the expected document reading order(Bailey, 2005). Additionally, the image with less than 75 dpi may not be converted accurately into text by OCR (Neal, 2011).

2.9.2 Structured PDF

Structured PDF is somewhat accessible and is best for documents without complex structure such as columns, tables, footers and side bars (Bailey, 2005). The document has no tags, images have no ALT-text. The columns, rows and headers of the table are not defined, and screen-readers may read text out-of-order, skip or incorrectly interpret sections of text or read tables across rather than by cell.

2.9.3 Tagged PDF

Tagged PDF is fully accessible to the software that supports PDF interpretation. Documents are tagged using many elements of the tag structure to identify sections, divisions, captions, tables and images (Bailey, 2005). Tables are fully rendered using tags to identify columns, rows, headers and data content. Reading order is identified throughout the document. Tagging in PDF is designed to provide a structure similar to HTML. Tagged PDFs are created with built-in accessibility like HTML, not added on (Bohman, 2002). Adding tags creates a document's duplicate that is marked-up for accessibility. PDF tags provide a hidden structure and have no visible effect on the file. Structured PDF documents are made fully accessible and navigable for screen reader users by tagging certain elements within the document. Screen reader users can often understand a properly tagged PDF as well as an HTML document. HTML tags and PDF tags often use similar tag names and organisation structures. Improper PDF tagging, lack of tools and misunderstanding of usability, causes these electronic documents to be inaccessible for users with vision impairments (Xenos Group Inc, 2010).

2.9.4 PDF/Universal Accessibility

The PDF/UA standard defines technical requirements for universally-accessible PDF documents by identifying a set of relevant PDF functions (including text content, images, form fields, comments, bookmarks and metadata) based on ISO 32000-1 (PDF 1.7) and specifies how they must be used in PDF/UA-compliant documents. It does not address elements which have no direct impact on accessibility, such as the compression algorithms used for image data. (Drümmer & Chang, 2012).Consequently, as soon as the PDF file (created by PDF/UA) is converted to scanned PDF, its accessibility features do not work.

In addition for some components such as mathematical expressions and chemical equations PDF/UA needs to be equip with sets of MathMl and ChemMl. Otherwise the results of assistive technology have usability issues due to multidimensional and non-linearity nature of these components.

2.10 Components of PDF Documents

2.10.1 Accessibility of Non-Textual Components

PDF documents may contain textual, non-textual components, tables, mathematical expressions, chemical equations and non-alphanumeric symbols. Non-textual or graphical components such as figures, charts, diagrams, and graphs enable readers to easily acquire the nature of the underlying information (Lin et al., 2012). The use of non-textual graphical information such as line graphs, bar charts, and pie charts is rapidly increasing in digital scientific literature and business reports. These graphical components are commonly used to present data in an easy-to-interpret way. Graphs are frequently used in economics, mathematics and other scientific subjects. A vast amount of science, technology, engineering and mathematics (STEM) information is usually presented visually. Illustrations are often easier to understand for sighted people. These graphics are widely used in newspapers, text books, web pages, metro maps and instruction manuals. They provide significant cognitive benefits over text. These graphical components have an important role in conveying, clarifying and simplifying information (McCathieNevile & Koivunen, 2000). The majority of information in graphics that appear in formal reports, newspapers and magazines are intended to convey a message or communicative intention (Elzer et al., 2007). Traditionally, charts are used to display trends and relationship and, communicate processes or display complicated data simply. These charts may be designed for the experts and trained users for data visualization or in popular media without complicated scientific reasoning (Greenbacker et al., 2011).

Charts are typically intended to convey a message that is an important part of the document and this information generally not repeated in the article (Carberry et al., 2006). Since these components are inaccessible, data visualization techniques are not useful for vision-impaired users and they miss all conveyed information in images. A partially sighted person sees the author schema or flow chart but cannot decipher the labels. A colour blind person sees a pie chart but will not understand it if only colour is used to indicate each section. Students and professionals in the STEM fields who are blind or have low vision must find other ways to access this data. In many cases, they still rely on sighted people to read and describe images for them. This creates dependency which can be inefficient and time consuming. They are unable to see and understand this graphical parts and lose important parts of information. They are unable to learn about the processes involved in reading, analysing, and interpreting information presented in data visual graph and charts, which are frequently used in mathematics and scientific materials to present and summarize data. It is fair to say that lack of access to diagrams and other graphical content significantly limits educational and workplace opportunities for people with vision impairment. This is in contrast with textual content in which assistive technology have improved access (Diagram2012, 2012). They require a complete equivalent in text as a short description, or a text alternative to be accessible.

2.10.2 Approaches for Accessibility of Non-Textual Components

Scanners with interaction compatible OCR software can be used to read printed materials and store them electronically on computers, for later access. Such systems provide independent access to abstracts, journals, syllabus and homework assignments, however many OCR reading machine packages are not able to convert technical information like chemical and mathematical equations into text and are not capable of providing verbal descriptions of pictures and other graphical information (Miner et al., 2001).

There are several approaches to address the accessibility of charts using alternative methods. Many projects have attempted to make graphic components accessible to vision-impaired users by reproducing the image in an alternative medium, such as audio (Meijer, 1992), touch (Ina, 1996) or a combination of the two (Ramloll et al., 2000; Roth et al., 2001).

Graphics like bar and line graphs can be printed in raised dots or Braille. Traditionally, graphs and diagrams are presented in Braille, or raised dots and lines on the swell-paper (Yu & Brewster, 2002). Tactile graphics are images that use raised surfaces and vision-impaired users can feel them. They are used to convey non-textual information such as maps, paintings, graphs and diagrams. Picture in Flash is a tactile graphs method (PIAF).The general shape of the graph can be understood by touching it carefully, but hardware is needed to generate tactile charts (Goncu & Marriott, 2008). Discriminating ability and searching ability are the main effective restriction factors in this method which must be considered for tactile symbols in charts (Watanabe et al., 2012). Tactile symbols without these properties could not help to explore concepts of charts by vision-impaired people. Several problems are associated with Braille and Tactile technique:

- The cost of translating into an accessible graphics format, use of expensive tactile graphics and peripheral devices and lack of congruence with the original visual graphic cause limitation for tactile graphs usage (Diagram 2012, 2012).
- Only a small proportion of blind people can use Braille, because reading it requires sufficient tactile sensitivity which not all vision impaired may have (Murray, 2008).
- Blind people can only get a rough idea or estimation about the content (Yu & Brewster, 2002).
- Tactile diagrams are not durable.
- It is not easy to make changes to tactile diagrams.

Non-textual components can be made accessible to the vision-impaired in verbal description and audio format. Viable alternatives include generation of a tactile graph, delivering the information in text, interaction with an audio graph, or a combination tactile and audio approach (NCAM, 2009). When describing chart or figures within the text, the reader should state the figure and caption numbers before starting a verbal description of the image. After completing a verbal description of the image, the reader should return to the text. These techniques help to express graphical data in non-visual ways.

Haptic feedback is tactile feedback technology which recreates the sense of touch by applying forces, vibrations, or motions to the user.This technology is useful for guidance and assisting users navigation on the graph but it is not efficient to present exact data values to the user. Figure 2.1 shows a Haptic PHANToM device.

Figure 2.1: The PHANToM (Murray, 2008)

Due to nature of human touch, conveying large amounts of information through the touch channel is difficult and the narrow bandwidth can be easily overloaded. As a result haptic and tactile signals could be used to represent variables that don't change frequently but require attention. In addition it may take users some time to familiarize themselves with the new interface. The limitations of force feedback devices hinder users' exploration of the graphs (Yu & Brewster, 2002).

Sonification is conveying data via sound pitch and 2-dimensional acoustics (Brown & Brewster, 2003). It is difficult to convey data accurately with the acoustic method (non-speech sound), and moreover, since acoustics are volatile, information can easily be misheard.

A research by McMullen and Fitzpatrick aimed to make talking tactile diagrams viable as a method of delivering graphical material to visually impaired students at a distance, explained the merits and deficiencies of the system described and also provided psychological observations into how blind learners approach tactile diagrams and the cognitive processes that are used in their comprehension (McMullen & Fitzpatrick, 2008)

The Interactive SIGHT (Summarizing Information GrapHics Textually) system provides high-level knowledge for the vision-impaired that one would gain from viewing. SIGHT uses image processing techniques to extract communicative signals from a chart (Elzer et al., 2007), but it is still limited to present information of the bar chart within web pages. In this research the accessibility and usability of bar charts, line charts and pie charts have been reviewed.

Previous studies by Elzer et al recommends Visual Extraction Module (VEM) to provide chart accessibility. VEM is responsible for analyzing the graphic's image file and producing an XML representation containing information such as the graphic type (bar chart, pie chart) and the textual pieces of the graphic (such as its caption). For a bar chart, the representation includes the number of bars in the graph, the labels of the axes, and information for each bar such as the label, the height of the bar, and the colour of the bar (Elzer et al., 2007).

2.10.3 Graphical Components Accessibility Requirements

Providing alternative access to content is one of the primary ways that authors can make their documents accessible to people with disabilities. Text equivalents are always required for graphical information. Alternative content for users with disabilities will do the same as the primary content does for users without any disabilities.

The factors which must be focused on making alternative for graphical components are:

- The purposes for using graphs, task characteristics and discipline characteristics;
- The differences between presenting information visually and aurally;
- How and what graphical parts are represented in the mind (Friel et al., 2001);
- The problems of non-visual exploration of graphical components;
- Accurate understating of the ways in which non-textual parts benefit sighted people (Brown et al., 2013; Brown, 2007); and
- Obtaining comprehensive information and advanced techniques for the graphical understanding (Huang & Tan, 2007).

2.10.4 Alternate Access Methods for Visual Printed Material

Considering learning styles for the vision-impaired, alternate access to represent visual printed material to the vision-impaired are divided into two categories tactile the method and the audio method.

Tactile representation is active or dynamic. The graphs and diagrams are presented in Braille or raised dots and lines on the swell-paper (Yu et al., 2001).

Tactile graphics are images that use raised surfaces which a vision-impaired person can feel. They are used to convey non-textual information such as maps, paintings, graphs and diagrams and the user can explore the graphic (Cohen et al., 2006).

Audio representation is passive or static and the user is presented with a representation of the entire visual part at one time with limited user input (Conrod, 1996).

2.10.5 Mathematical Expressions

Mathematical Expressions are one of the most significant components within scientific and engineering PDF documents. Students with vision impairment encounter barriers in studying mathematics particularly in higher education levels (Murray, 2008). Accessing and doing mathematics, is one of the biggest obstacles for them in school and at the university. The lack of easy access to mathematical resources is a barrier to higher education for many vision-impaired students and puts them at an unfair disadvantage in school, academia, and industry (Jayant, 2006). Results from the National Assessment of Educational Progress show that there is great disparity between the mathematical skills of students with disabilities and students without disabilities (Noble, 2008). Students with print disabilities must be offered an equal chance with sighted students in mathematics subjects. For students who are blind or vision-impaired, providing an alternative text-only format from PDF documents is very important. They can easily access text format with screen-reader software that converts text into audio but making mathematics accessible to the vision-impaired users is a complicated process. Traditionally, to check accuracy of mathematical description an specialist helps with the aid of a Braille reader, which is a time consuming process.

There are some PDF documents containing mathematical expressions, which can be accessed via standard assistive technology. Although screen reader convert mathematical expressions into audio, this result cannot convey the conceptual meaning of the original mathematical expressions. The language of mathematics is not purely descriptive and sequential. In most cases making a text description can cause ambiguity.

A screen reader reads mathematical documents in a linear way from left to right but due to the multidimensional nature of some materials, such as mathematical formulae containing subscripts and superscripts, the result would be ambiguous. As it is observed from following mathematical expressions (Figure 2.2), there are three different expressions, that the screen reader reads them from left to right in linear manner and produces one result for all of them such as: $Xk1 + Xk2 + Xk3 + ... Xkn = 0$

$$X_1^K + X_2^K + X_3^K + \ldots + X_n^K = 0$$

$$X^{k^1} + X^{k^2} + X^{k^3} + \ldots + X^{k^n} = 0$$

$$X^{k_1} + X^{k_2} + X^{k_3} + \ldots + X^{k_n} = 0$$

Figure 2.2: 2-Dimensional mathematical equations contain subscripts and superscripts

Reading and writing mathematics is inherently different from reading and writing text. Mathematics can even be considered a language of its own (Karshmer et al., 1999).

Mathematical formulae presentation in an accessible form is very complex. If mathematical equations contain fraction bars, to prevent ambiguity, it is important to indicate the numerator and the denominator and be clear about the quantities being multiplied, divided, added, or subtracted (Karshmer & Bledsoe, 2002). Students with vision impairments can learn mathematics when they have access to the proper combination of computer hardware, software and other assistive technologies.

In contrast with image-based mathematical expressions there is MathJax which:

- is an open-source JavaScript display engine for LaTeX, MathML, and AsciiMath notation that works in all modern browsers,
- requires no setup on the part of the user (no plugins to download or software to install), so the page author can write web documents that include mathematics and be confident that users will be able to view it naturally and easily,
- uses web-based fonts (in those browsers that support it) to produce high-quality typesetting that scales and prints at full resolution (unlike mathematics included as images),
- allows page authors to write formulas using TeX and LaTeX notation, MathML, a World Wide Web Consortium standard for representing mathematics in XML format, or AsciiMath notation,
- converts TeX notation into MathML, so that it can be rendered more quickly by those browsers that support MathML natively, searchable and editable in other program,
- is highly configurable, allowing authors to customize it for the special requirements of their web sites; and

- has a rich application programming interface (API) that can be used to make the mathematics on your web pages interactive and dynamic.

2.10.6 Tactile Approaches for Accessibility of Mathematical Expressions

6-dot (Braille) and 8-dot (Dots Plus) are physically raised representations of characters on a flat surface. Conveying mathematical equations in Braille is done using specialized mathematical Braille codes. The Braille notations for mathematical characters have no international standards so printing mathematics in Braille can only be done by local specialists (Larsson, 2008). This manual process is time consuming, causing significant delays for the person requiring access to the mathematical subjects. Nemeth Braille code is a generally accepted method for conveying mathematics in the most countries. The Nemeth Braille Code is used to encode mathematical and scientific formulae, using standard 6-dot Braille cells for tactile reading by the vision-impaired people. The Nemeth Braille Code is an example of a compact human-readable mark-up language (dotlessbraille.org, 2002). Braille is suitable for text representation, which is linear in nature. Mathematical formulae are not linear. Non-Linearity and limitation of character sets are problematic in the use of Braille for mathematical formulae representation. In comparison with linear-nature text, mathematical equations are multidimensional; they may contain fractions, superscripts and subscripts, simple expressions, algebra, square roots, logarithms, series, sums, products, matrices, integrals, limits, trigonometry, and exponential. Thus, by increasing complexity of mathematical equations, Braille representation complexity will be increased. Additionally, Braille text is represented by a limited number of characters including upper and lowercase letters, the 10 digits, various punctuation marks and a small set of special characters, which are made by the $2^6 = 64$ combinations of possible 6-dot placements. Equations may contain any of the normal text characters, plus a large number of special characters. It is obvious that Braille mathematical representation with a 64-character set is still very difficult. This limitation causes Braille difficulties in dealing with mathematical characters and distinguishing between numerous meanings for a single cell (dotlessbraille.org, 2002).

Another tactile method is 8-dot Braille or Dots Plus representation. By using 8-dot, the basic character set is extended from 64 to 256 characters (Dixon, 2007). Most 8-dot systems attempt to have some sort of graphics associated with the dot pattern allowing for clearer representation (Karshmer et al., 2007). However, learning and remembering this character set may be problematic for users. Dots Plus is a two-dimensional spatial tactile mathematics notation, which is a composite of standard Braille with raised lines and symbols. Creating a Dots Plus document requires embossing, which can be done on a Tiger Braille Printer (University of Washington, 2013). Braille mathematical materials can often be created from electronic files with Braille translation software. Many of the symbols and spatial typesetting conventions used in mathematical equations cannot be replicated with the American Standard Code for Information Interchange (ASCII) character set, consequently are not recognised by standard OCR systems. Thus, additional processing is often required to identify the issues regarding non-recognised mathematical symbols.

Tactile output is very helpful for blind mathematics students in representing the graphical output of a function as shown by Hapatic technology (Sahyun et al., 1998), but this technology is still too expensive to be widely implemented.

2.10.7 Audio Approaches for Accessibility of Mathematical Expressions Using Mark-up Formats

Most publishers do not generally produce specialized alternative-format materials such as Braille. Many textbook publishers create electronic versions that can be used with many types of assistive technologies (University of Washington, 2013). In such a case, once a submission has been accepted for publication, text and mathematics are converted to a mark-up language which is then used to create the article for printing. For example, Mathematical Markup Language (MathML) is a format for universally designed mathematics. MathML can be used for mathematical equations within standard of extensible Hyper Text Markup Language (XHTML) or extensible Markup Language (XML) content. Unlike image formats, using MathML enables mathematics to be served, received, and processed in digital environments such as the World Wide Web (Ion, 2000). MathML also provides standard tagging content with an information structure that can be navigated by assistive technology. MathML equations can be accessed via standard assistive technology applications, such as synthetic speech. Text to Speech (TTS) technology can generate the audio form of an equation provided by MathML. This type of audio has been one of the most popular and successful output media for blind students in understanding the process of mathematics (Nazemi et al., 2014). Audio System for Technical Readings (ASTER) and Talking Emacs (EmacSpeak) are two projects aiming to represent mathematical expressions to blind students through audio output. ASTER reads complex equations with special intonation to emphasize parts of the equation (Raman, 1994). EmacSpeak reads C code to the user, providing descriptive information (Raman, 1995). A comprehensive system of notation is essential to be capable of expressing all mathematical relationships clearly and intelligibility.

2.10.8 Accessibility of High Volume Transaction Output (HVTO)

The financial services, insurance, utilities and government sectors typically provide high volumes of PDF documents, which are stored or archived from seven to ten years. Since it is unlikely that all recipients are prepared to accept electronic delivery of their documents, a large portion of the documents is still printed as PDFs. In an online billing system, bills are sent to the email accounts of customers as PDF attachments or HTML links. These bills, in the most cases; are not accessible through assistive technologies or usable by vision-impaired customers (Nazemi et al., 2014).

PDF documents have several features which make it popular for HVTO generation:

- PDF viewers are platform independent (Windows, Mac, Linux, and even portable devices);
- PDF supports compression to make the file smaller; and
- PDF is page independent, which means there is no need to process pages 1 to 999 in order to process page 1000. Each page stands on its own. This is valuable when it comes to printing performance. If necessary, multiple processors can be employed to process pages in parallel (Crawford Technologies Inc, 2011).

As HVTO demands put increasing pressure on online presentment capabilities, accessibility has become a growing concern particularly when these files present to vision-impaired people using assistive technologies. Addressing accessibility in an HVTO environment such as financial services can be difficult, but is certainly achievable. The industry has already made great strides to address web sites and content portals (such as an online banking interface). To date, financial institutions and other HVTO statement generators deliver alternative format statements to their vision-impaired clients using internal consultants or Document Accessibility Services (DAS). These statements typically come in the form of Braille, large-print documents, or audio. But using current outsourcing options to cover an important and growing customer segment is neither cost-effective nor the preferred method to represent information to vision-impaired customers. Moreover, specialized statements generally deliver information to vision-impaired clients with significant delay (Nazemi et al., 2014). While most outsourcers promise turnarounds of 48 hours or more, market research indicates most vision-impaired customers receive their statements a month after normal delivery. Such delays can influence an organisation's ability to deliver equitable access to all customers and may be seen as discriminatory towards vision-impaired customers. This issue affects equal access to all customer segments.

The majority of the vision-impaired community has multiple bank accounts, credit cards, savings and investments which require monthly statements. Although current options may meet existing standards, their delivery delay, cost and complexity are problematic. In dealing with HVTO such as bank statements, insurance benefits or tax forms, properly tagged and accessible PDF for vision-impaired people is more preferable than using alternate formats such as Braille (Actuate Company, 2010).

Unlike information in ordinary documents which come from left to right and top to bottom, most HVTO represent information not in this order. Screen reader users would not use the document in a typical left to right format. Account numbers, overdue notices and charts will interfere with a screen reader's ability to convey information in an appropriate order.

2.11 Accessibilty Issues of Chemical Equation Presentation

2.11.1 Represent Chemical Equations to Vision-Impaired Students

Chemical representations refer to various types of formulae, structures, and symbols used to represent chemical processes and conceptual entities such as molecules and atoms (Wu et al.,2001). Chemistry at the symbolic level is represented by symbols, numbers, formulae, equations, and structures (Gabel, 1998). Studies of Ben-Zvi et.al have shown that understanding symbolic representations is especially difficult for all students because these representations are invisible and abstract while students rely heavily on sensory information. In addition, without substantial conceptual knowledge and visual-spatial ability, vision-impaired students are unable to translate chemical representations (Ben-Zvi et.al, 1987). Mayo's (Mayo, 2004) research results indicate that these student had difficulty interpreting common representations of the structures encountered in introductory chemistry and organic chemistry courses. They have problems with balancing equations and are unable to interpret subscripts, coefficients, and implicit meanings of the equations to produce accurate pictorial representations (Yarroch, 1985).

For studying science, especially chemistry, the role of visual and symbolic modes of representations is extremely important (Sahin & Yorek, 2009). However, students with visual impairments cannot utilize these forms of visual and symbolic representations because of their impairment. Hence, the study of chemistry becomes an area that is virtually unattainable for them. According to the results of the research undertaken by Bodner et al, current pedagogical practices need to be revised to enhance the conceptual student understanding and develop symbolic representations used to describe chemical reactions (Bodner et al., 2013).

2.11.2 Concepts of Chemical Equation and Balancing

Chemistry is the study of the properties of materials and the changes that those properties undergo. A chemical equation is the symbolic representation of a chemical reaction wherein the reactant entities are given on the left-hand side of the equation and the product entities are given on the right-hand side of the equation. According to the Law of Conservation of Mass and the Law of Conservation of Charge, the quantity of each element does not change in a chemical reaction. Thus, each side of the chemical equation must represent the same quantity of any particular element. Also, the same charge must be present on both sides of the balanced equation. Chemical equation balancing plays a critical role in understanding the structure of the chemical elements involved in chemical reactions. The ability to balance chemical equations in terms of mass and charge is a key skill that must be mastered by the serious chemistry student. Each reactant or product contains one, or more than one element and each element has a different quantity. In the result, the balancing task must be performed so that equal quantities of each element are on the left and right sides of the equation. Balancing a chemical equation involves multiple steps:

1. Recognition of the elements in each reactant on left side of the equation;
2. Recognition of the elements in each product on the right side of the equation;
3. Finding the quantity of each element on left and right sides;
4. Comparing the quantity of each element on left and right sides; and
5. Finding the appropriate coefficients to insert before each reactant and products to make entire equation balanced.

These coefficients are the absolute values of the stoichiometric numbers and lead to chemical equation balancing. For balancing, students need to have sufficient knowledge of the chemical element structure as well as the mathematical equation solving skills. There are two methods for balancing chemical equations or finding coefficients:

1. The inspection method is the usual method of balancing chemical equations. It is fast but confusing for complicated equations. In addition, the final result must be double checked to make sure that the chemical equation is indeed balanced.
2. The algebraic method uses algebraic equations. This method is systematic and can be applied to difficult reactions. It can be easily used with equation solvers but it is time consuming to define corrected algebraic equations (Jamin, 2013).

2.11.3 Current Methods to Represent Chemical Equation to Vision-Impaired Students

Traditionally, there are two methods for representing symbolic level or chemical equations to vision impaired students. They are as follows:

1. Tactile method: The use of molecular models to teach balancing equations, which gives a concrete rather than abstract description of the problem, is an efficient approach. Using Braille or embossed paper is another tactile representation of chemical equations and students can actively interact with the implicit information (Edman, 1992). Generally, the molecular formula shows the composition of a compound and the number of each type of atom in a molecule. For example, the molecular formula for benzene is: C_6H_6 . In terms of representation, molecular formulae due to have subscript(s) are not entirely linear in nature. Since Braille is suitable for text representation, which is linear in nature, it is not suitable for molecular chemical formulae representation.

2. Audio method or using assistive technologies such as Text to Speech (TTS). In audio representation, students passively hear all information from the left to right hand side without opportunity to rehear a specific part and/or interact with it.

As it is perceived from previous section(s), balancing can be challenging even for sighted students who are able to use visual aids to communicate with and understand the equation's meaning, hence for vision-impaired students the challenge is greatly increased.

2.12 Summary

It can be seen from the literature review that there are three learning styles, namely, tactile, auditory and visual. Vision-impaired learners cannot utilize visual styles. Effective use of tactile methods requires tactile sensitivity. However not all vision-impaired people have this sensitivity. Screen readers are assisitive technology based on auditory style to convey electronic documents information to vision-impaired. Tactile methods provide active accessibility however audio methods represent passive accessibility.

PDF as a commonly used electronic document still encounters some restrictions regarding accessibility to vision-impaired users. Screen readers cannot read scanned PDF due to inability to deal with images and scanned PDF is complete inaccessible to them. PDF may contain non-textual components (charts and graphs), non-linear components (tables), not-in-order (multiple columns papers, HVTO) and multidimensional components (mathematical expressions and chemical equations). Braille is not suitable for representing multidimensional mathematical expressions and chemical equations due to its linear nature.

A comprehensive system of notation is essential to be capable of expressing all mathematical relationships clearly and intelligibility to convert mathematical expressions to unambiguous audio format using TTS.

Existing tactile methods to access graphical components are Braille, tactile graphs and Hapatic devices which are dependent on tactile sensitivity, not durable and expensive respectively. The vision-impaired people are prevented from certain career paths where non-textual, non linear, multidimensional components, graphs and graphics play a major role in electronic documents. One of the most prominent targets of this project is providing opportunity for the vision-impaired students to attend mathematical and chemical courses (in symbolic representation level), full access to scientific documents containing graphical components which supports them to gain knowledge, build skills, partake in further education and improve career and employment.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the main objectives that were addressed through the research described within this thesis. The methodology to find the approaches in order to achieve the research objectives are also presented here.

3.2 Objectives

This research has three main objectives as follows:

1. Study the issues in accessing the most commonly used electronic documents to formulate the definition of the research problem;

2. Investigate the accessibility of scientific documents including graphs, charts, mathematical formulae and chemical equations to provide educational opportunities for vision-impaired students; and

3. Design and implement the CRS for the vision-impaired users specifically students in high school and tertiary education.

   The CRS is expected to perform the following tasks:

   - Search through materials such as user manuals and references books containing several levels or navigate through different levels;
   - Bookmark or save the ending point of reading session; and
   - Represent essential information and conceptual meaning of non-textual components (charts and graphs), not-in-order (multiple columns papers) and multidimensional or non- linear components (mathematical expressions and chemical equations) in alternative text description format to convey them to vision-impaired users.

3.3 Research Methodologies

Research methodologies are generally divided into two main categories. These categories are:

1. Social sciences methodology; and
2. Science and engineering methodology.

Social science methodology consists of researching social issues; concepts and ideas. Science and engineering approaches involve developing practical solutions for identified issues. The scientific method is used in algorithm design and software development. It creates a descriptive model, develops hypotheses using the model, validates hypotheses running experiments and refines model and repeat (Sedgewick, 2010).

3.4 Alternative Methodologies for Software Development

The major goal of this research was the designing and implementation a software system. For this reason science and engineering methodology was chosen to be used. To select appropriate methodology within this research, it is important to describe and compare two different software development methodologies. There are several ways to develop software, however currently two of the most prominent methods are the Waterfall and the Agile methods(Kasia, 2013).

3.4.1 Waterfall

- The Waterfall Model is based on a traditional engineering process. In the Waterfall method, the next stage of the process does not commence until the previous stage has been completed. Royce who first discussed the model in 1970 mentioned that this model should not be adopted in its current form, due to its limitations. He continued to describe the model using a feedback mechanism at each stage (McMeekin, 2010). The Waterfall model can essentially be described as a linear model of software design. Development flows sequentially from the start point to the end point, with several different stages such as conception, initiation, analysis, design, construction, testing, implementation and maintenance (Melonfire, 2006). As this process is sequential, once a step has been completed, developers can go back to a previous step only with scratching the whole project and starting from the beginning. There is no room for change or error correction, so a project outcome and an extensive plan must be set in the beginning and then followed carefully. Thus, the clients do not have the ability to change the scope of the project once it has begun. Waterfall methodology can be used if the key to success is definition, not speed or if there is a clear picture of what the final product should be. The Waterfall method has the following overarching characteristics:

- Once a step has been completed, developers cannot go back to a previous stage and make changes.
- It relies heavily on initial requirements. However, if these requirements are faulty in any manner, then there is a high possibility for the project will fail. If a requirement error is found, or a change needs to be made, the project has to start from the beginning.
- The whole product is only tested at the end. If bugs are generated early, but discovered late, their existence may have affected other codes and more errors could have propagated through the system.

3.4.2 Agile

The Agile is a new software development methodology. This method was defined by seventeen software engineering experts in the 2001 meeting to discuss about an alternative for documentation driven and heavyweight software development processes (Highsmith, 2001). The Agile manifesto indicates highlighted points as follows:

- Individuals and interactions over processes and tools;
- Working software over comprehensive documentation;
- Customer collaboration over contract negotiation; and
- Responding to change over following a plan

In this method active user involvement is imperative, feedback is necessary and testing is integrated throughout the project life cycle. It:

- Follows an incremental approach;
- Evolves requirements with fixed timescale;
- Has adaptive planning;
- Focuses on early and frequent delivery of products;
- Improves continuously with small developing;
- Releases incrementally and iterates;
- Responds to change flexibly and quickly; and
- Completes each feature before moving onto the next (Waters, 2007).

3.5 Agile Methodology and this Research

The overarching aim of this project is to design and implement a system for vision-impaired people which allows them to access common electronic documents. For this purpose several software application modules needed to be developed, tested, fixed, integrated and ran again. The style of this research methodology is very similar to the Agile software development methodology.

During this research software was developed repeatedly in small modules. Since the aim of this research is to achieve on satisfactory end-user experience, thus any and all user requests for changes/editions were incorporated irrespective of implementation complexity.

The adoption of Agile methodology addressed in this research are :

- Requirement analysis (Requirement Analysis Using Survey-Chapter 4)
- Designing documents and prototype (Licensing Approaches-Chapter 4)
- Coding (Appendix B)
- Defect identification contains accuracy check based on data mining and resolve bugs (Testing-Chapter 7)
- Demonstration and Deployment (System Components and implementation-Chapters 5 and 6)

3.6 Summary

As stated in this chapter, two main research methodology categories are social science and science and engineering. Waterfall and Agile are two distinct methods for software developers. Since the Waterfall process is sequential, once a step has been completed, there is no room for change or error, so a project outcome and an extensive plan must be set in the beginning and then carefully followed. Compared to Waterfall, the Agile method, continuous improvement and iteration are performed based on active users involvement and their feedback, iterations occurred in the design of this research. This chapter concluded that the agile methodology is a better method for this research due to compatibility of this research criteria with agile specifications.

Chapter 4

Requirements Analysis

4.1 Introduction

The results of two conducted surveys within this research are analysed to define the research problem within this chapter.

The Requirements Analysis and Design Documents which are overtly defined within the Agile Methodology are also described in this chapter.

4.2 Requirement Analysis Using Survey Results

Two surveys were conducted in order to:

1. Determine the perceived accessibility of electronic documents to the vision-impaired; and

2. Discover vision-impaired users' communication quality with graphical components in electronic documents.

4.2.1 Accessibility of Electronic Documents

In July 2011, a survey was conducted among 131 students of the Cisco Academy for the Vision-Impaired (CAVI) (www.ciscovision.org). All respondents had significant though not defined vision impairment. The entry requirement to CAVI is that the student must be defined as "legally" blind in their locality. The gender and age of respondents was collected during the survey, however, the most important factors were familiarity with assistive technology and knowing how to use a computer. The majority of participants were between the ages of 18 and 65, almost 55% were male and the 45% were female as two pie charts in Figure 4.1 illustrate.

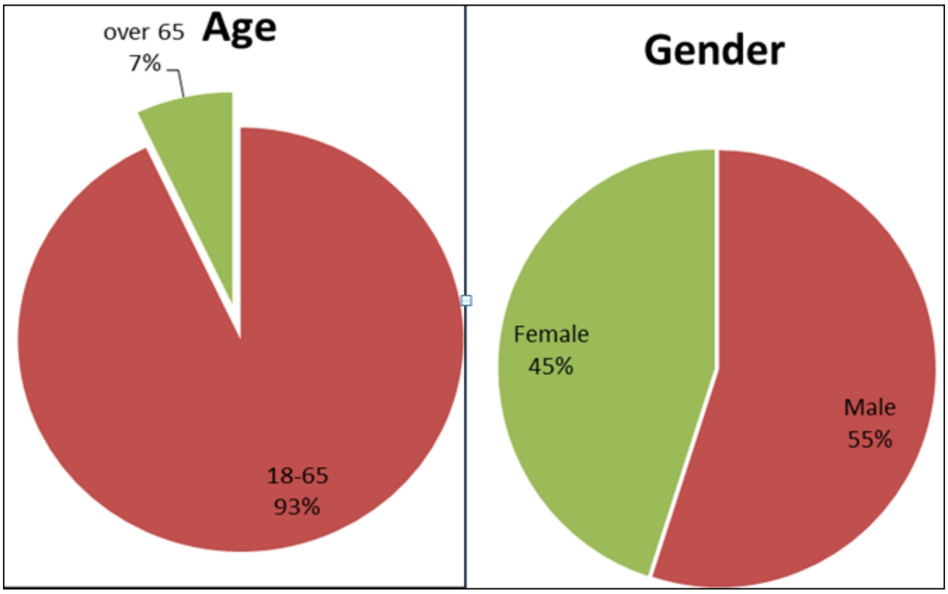
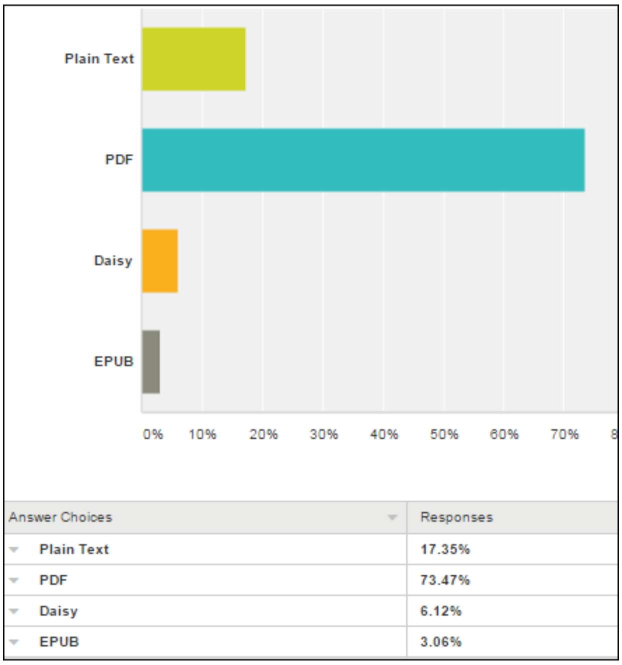


Figure 4.1: Survey participants age group (left) and survey participants gender (right)

According to the demographic and due to the nature of CAVI, most of the students were moderate to highly skilled computer users. The purpose of the survey was to determine the perceived accessibility of electronic documents for the vision-impaired community. The questions that were asked in this survey are available in Appendix A. All bar chart figures in this chapter were extracted from Survey Monkey web pages related to this research surveys.

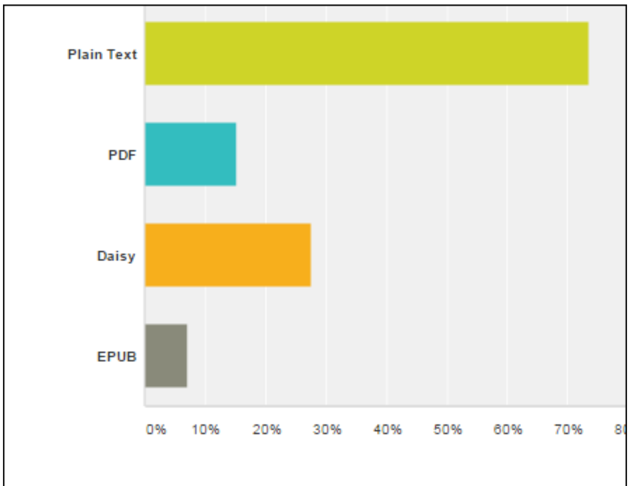
The participants responded to a question about frequent usage of electronic documents. 70% of participants read electronic documents more than 10 times per week, whereas 30% of them read these documents between 5 to 10 times per week. Results showed that PDF was perceived as the most common format utilized for document distribution since 73.47% of the population questioned believed it In the opinion of 17.35% of participants, plain text is the most common. 6.12% mentioned DAISY and 3.06% indicated ePub. Figure 4.2 illustrates this breakdown of vision-impaired peoples' perception of the most commonly available format of digital documents.



Answer Choices	Responses
Plain Text	17.35%
PDF	73.47%
Daisy	6.12%
EPUB	3.06%

Figure 4.2: Respondents' perception of the most common document format

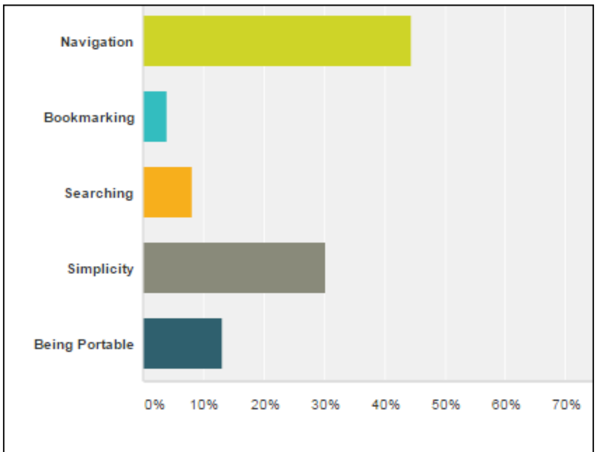
In the second section, 73.47% of people regularly read plain text, while 27.55% regularly use DAISY. Figure 4.3 illustrates this preference, it shows less than 20% prefer PDF and less than 10% prefer the ePub format. According to the results of sections 1 and 2 of the survey, plain text is most preferred format. PDF is the most commonly used and the most inaccessible electronic document. It is worth noting that in some cases respondents selected more than one format.



Answer Choices	Responses
Plain Text	73.47%
PDF	15.31%
Daisy	27.55%
EPUB	7.14%

Figure 4.3: Respondents' Most preferred format

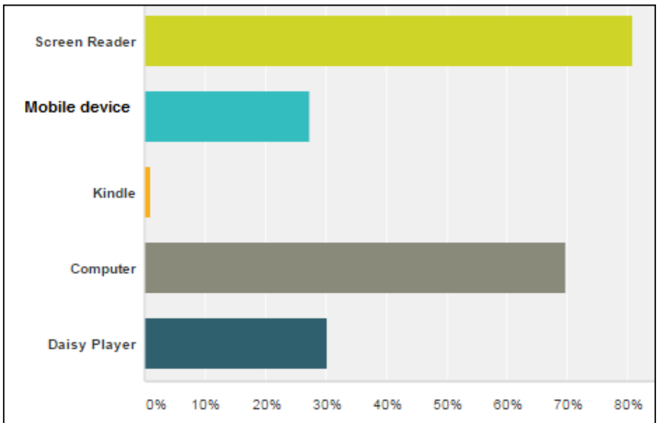
For 44.4% of respondents an important factor in choosing electronic documents is the navigation ability and for 30.3% it is simplicity. For 13.13% of participants portability is important factor. Bookmarking and searching are significant for 4.04% and 8.08% respectively. Figure 4.4 graphically represents these choices of preference.



Answer Choices	Responses
Navigation	44.44%
Bookmarking	4.04%
Searching	8.08%
Simplicity	30.30%
Being Portable	13.13%
Total	

Figure 4.4: Important factors in selecting a document format

Finally, when asked how they access electronic documents, 80.8% of respondents answered that they use a screen reader, 30.3% use a DAISY player and 27.27% use a mobile device. The total exceeds 100% as several respondents mentioned more than one access method. Figure 4.5 illustrates these results and shows that accessing electronic documents is also done via computer.



Answer Choices	Responses
Screen Reader	80.81%
Mobile device	27.27%
Kindle	1.01%
Computer	69.70%
Daisy Player	30.30%

Figure 4.5: Accessibility Tool

114 persons provided an open text response to the final question. The majority stated that the greatest barrier to the full access of electronic documents is secured and restricted PDFs, which are not accessible for most, screen readers.

The conclusions that can be drawn from the survey are the followings:

- For a significant number of respondents, PDF is perceived as the most common format (Figure 4.2) and is also perceived as the most inaccessible format.
- It was shown (Figure 4.3) that plain text was the preferred format. Assistive technologies, such as screen readers can easily read text to a blind or low-vision user. Thus, many electronic documents are converted to text to assist in the accessibility.
- The majority of people who participated in this survey would like to use formats that offer navigation ability (Figure 4.4), but according to the open text responses, complicated and expensive existing players hinder them to use these kinds of format. Hence DAISY was not commonly used, but considered the most accessible.

According to these results, accessibility of PDF and navigation ability of DAISY are important issues for vision-impaired students. Therefore, providing accessible PDFs and DAISY player implementation were considered in designing the CRS for vision-impaired users. The results of the survey are illustrated in Figures 4.2, 4.3, 4.4and 4.5 and summarized in Table 4.1.

Table 4.1: Results summary for survey 1

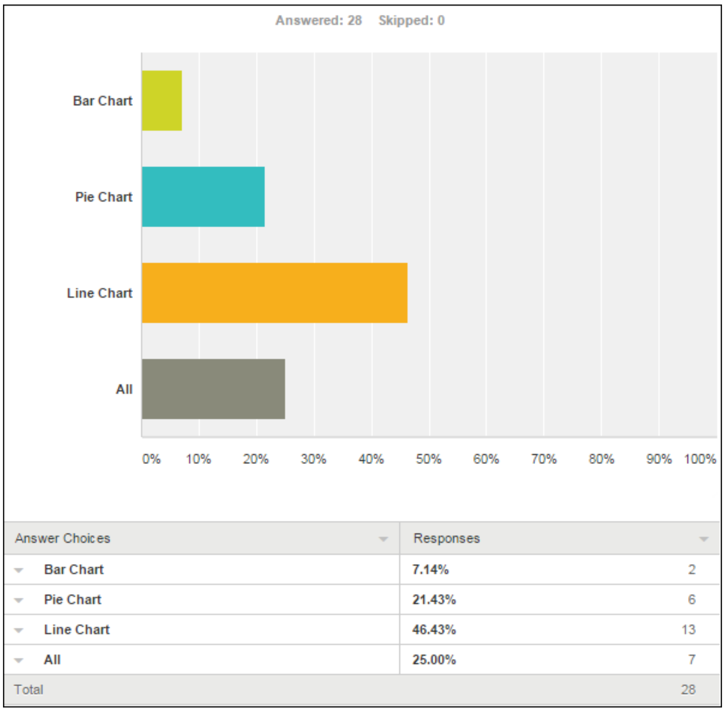
Question	Response				
Gender	55% Male	45% Female			
Frequent usage of electronic documents times/week.	70% More than 10	30% 5 to 10			
Most common format	73.47% PDF	17.35% Plain text	6.12% DAISY	3.06% ePub	
Most preferred format	73.47% Plain text	27.55% DAISY	15.31 PDF	7.14 ePub	
Important factors in selecting a document format	44.4% Navigation	30.3% Simplicity	12% Portability	7% Searching	3% Book marking
Accessibility Tool	80.8% Screen Reader	30.3% DAISY player	27.27% Mobile device		

4.2.2 Communication with non-Textual Components in Electronic Document

In July 2013, a survey was conducted among 28 vision-impaired users. The purpose of the survey was to discover how the vision-impaired communities communicate with non-textual components in electronic documents. The following questions were asked in this survey:

Question one: Which one of these graph styles do you find most difficult to understand? (Please select no more than one option).

As Figure 4.6 shows, for most vision-impaired respondents, line chart is the most inaccessible graphical style.

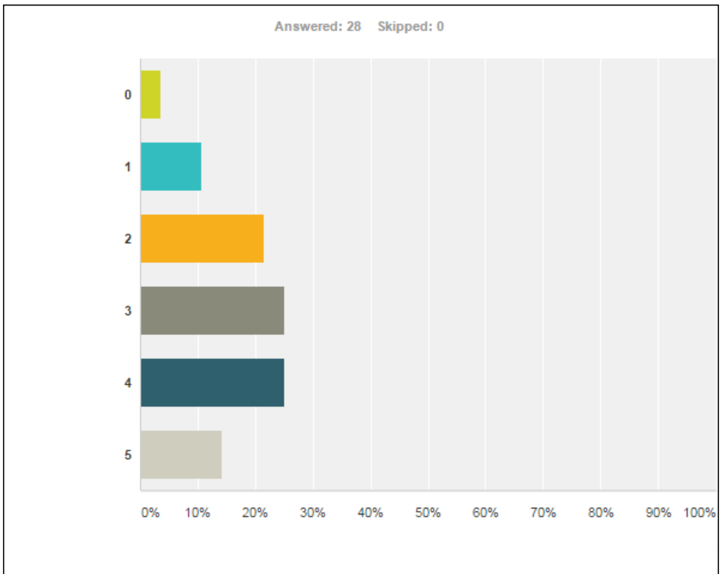


Answer Choices	Responses	
Bar Chart	7.14%	2
Pie Chart	21.43%	6
Line Chart	46.43%	13
All	25.00%	7
Total		28

Figure 4.6: Most inaccessible graphical components

Question two: How often do you encounter inaccessible graphical representations of data in documents? (Please select no more one option. Use the 0- 5 scale, 0 = never 5 = very often).

Figure 4.7 shows that how frequently they encounter inaccessible graphical representations.Almost 50% of participants regularly encounter inaccessible graphical representations of data in documents

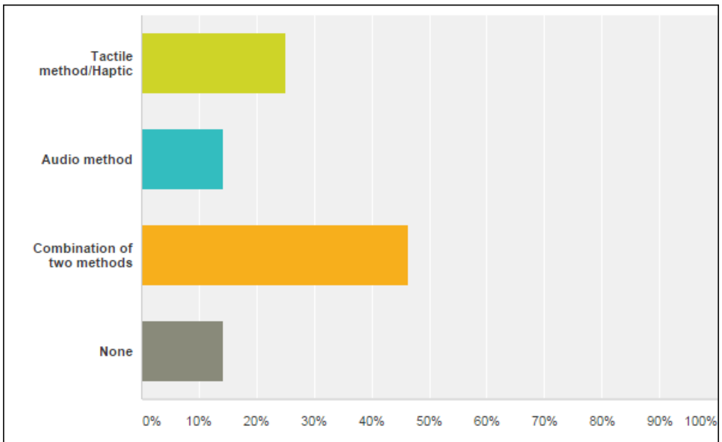


Answer Choices	Responses	
0	3.57%	1
1	10.71%	3
2	21.43%	6
3	25.00%	7
4	25.00%	7
5	14.29%	4
Total		28

Figure 4.7: Frequent encounter inaccessible graphical representations

Question three: In your opinion, what is the best current solution for accessing bar, pie and line graphs within documents?(Please select no more one option).

Figure 4.8 shows the responses to this question where 25% prefer tactile/haptic methods and 14.29% believe audio methods are superior. In the opinion of 46.43% of participants, a combination of the two methods would be effective and 14.8% believe none of these methods are helpful. Therefore, a combination of the two methods can be recommended for this situation.

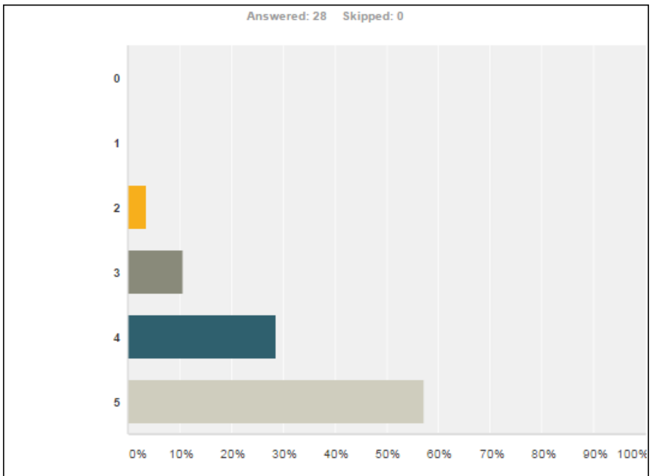


Answer Choices	Responses	
Tactile method/Haptic	25.00%	7
Audio method	14.29%	4
Combination of two methods	46.43%	13
None	14.29%	4
Total		28

Figure 4.8: Best perceived accessibility method for graphical components

Question four: How significant would full navigation of non textual components in documents be?(Use 0-5 scale, 0 totally unimportant, 5 extremely important).

In the view of 3.57% of respondents, it is not important to provide full navigation ability of non-textual components in documents. Nevertheless, for 10.71% it is a considerable factor, for 28.57% very significant and for 57.14% it is an extremely important feature. Figure 4.9 depicts these results clearly showing how important full navigation in accessing non-textual components. Therefore, ability to navigate non-textual components is a required feature for CRS.

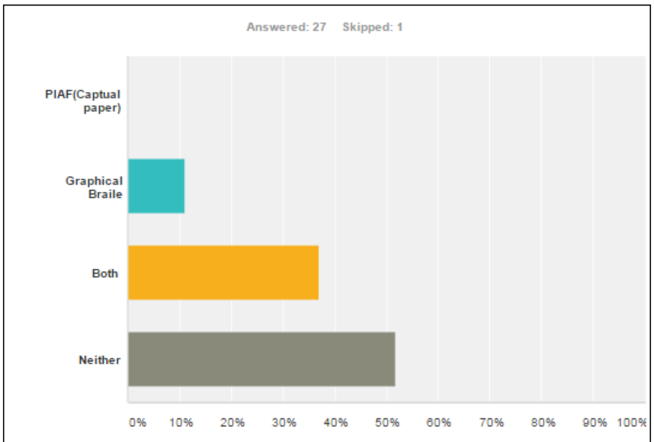


Answer Choices	Responses	
0	0.00%	0
1	0.00%	0
2	3.57%	1
3	10.71%	3
4	28.57%	8
5	57.14%	16
Total		28

Figure 4.9: Navigation importance for access to graphical components

Question five: What kind of tactile representation do you use?

In response to this question regarding preferred type of tactile method, most of them do not neither use PIAF nor Graphical Braille as shown in figure 4.10



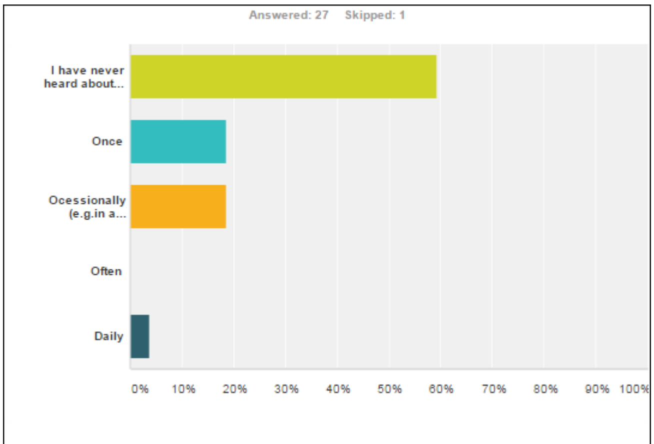
Answer Choices	Responses	
PIAF(Captual paper)	0.00%	0
Graphical Braile	11.11%	3
Both	37.04%	10
Neither	51.85%	14
Total		27

Figure 4.10: Preferred type of tactile method

Question six: How often do you use a haptic device ? (e.g.PHANToM the device which allows you to feel the graphs)

In response to question about frequent usage of haptic device, approximately 60% of respondents have never heard about this device (Figure 4.11).

The survey results for all six questions have been summarized in Table 4.2.



Answer Choices	Responses	
I have never heard about this device	59.26%	16
Once	18.52%	5
Ocessionally (e.g.in a trial)	18.52%	5
Often	0.00%	0
Daily	3.70%	1
Total		27

Figure 4.11: Frequent usage of Haptic device

Table 4.2: Results summary for survey 2

Question	Response			
Most inaccessible graphical components	46.43% Line charts	21.43% Pie charts	25% All	7.14% Bar charts
Best perceived accessibility method for graphical components	46.43% Combination	25 % Tactile	14.29% Audio	14.29 % None
Navigation importance for access to graphical components	57% Extremely effective	28.57 % Very significant	10.71% Considerable	3.57% Ineffective
Preferred Kind of tactile method	0% PIAF	11.11% Graphical Braille	37.04% Both	51.85% Neither
Frequent usage of Haptic device	59.26% Never heard about this device	18.52% Occasionally	3.7 Daily	

4.3 Design Documents and Prototype Using Licensing Approaches

The results from sections 4.1 and 4.2 were taken into account for the deign and prototyping of the CRS. To reduce the total cost of the CRS the OS kernel must be open source (modifiable), free of license constraints and light weight, which requires lower memory and power. As the CRS software application runs under Linux environment, then Linux compatible open source software packages can be used in the development of applications associated with different modules. A selection of utilized packages is listed below:

- ImageMagick is an open source software for displaying, converting and editing raster image files (Still, 2005). ImageMagick is a powerful image processing package and was used for PDF layout analysis, page segmentation and developing applications to extract information from charts.

- docx2txt.sh converts .docx to text. docx2txt is a tool that attempts to generate equivalent (ASCII) text files from Microsoft .docx documents, preserving some formatting and document information with appropriate character conversions for a good (ASCII) text experience. It is a platform independent solution consisting of (core) Perl and (wrapper) Unix/Windows shell scripts and a configuration file to control the output text appearance to fair extent (Kumar, 2014). The following command converts .docx to .txt:

```
./docx2txt.sh $fil > ${fil:0:$((${#fil}-5))}.txt
```

- odt2txt is a Python script that converts Open Document Text (.odt) files to plain text. The output text is marked up using Markdown syntax, which preserves some of the most important formatting (Stosberg, 2006). The following commands convert .odt to .txt:

```
odt2txt.py Doc.odt > tmp.txt
markdown.py tmp.txt > Doc.html
```

- AbiWord is a free and open source software word processor written in C++ to convert .doc to text. It is a free word processing program similar to Microsoft® Word. It is suitable for a wide variety of word processing tasks (Lachowicz, 2009).The following command converts .doc .txt:

```
abiword --to=txt sample.doc
```

- html2text converts html and xml to text.
- Tesseract, Cuneiform, Gocr and Ocrad are open source OCR packages which extract text from scanned PDF.
- Festival, Flite and Espaek are open source TTS packages which convert the plain text generated by CRS modules to audio output.
- Julian is open source voice recognition package which was used to implement the speech recognition as user interface.
- poppler-utils (pdftohtml, pdfminer, pdftops and pstopdf) is open source PDF converter packages to provide PDF accessibility through assistive technology. Most of the eBooks which are presented in secured PDF format are not accessible to screen readers such as JAWS and NVDA, however use of these packages facilitates the owner of DRM-protected content to make them accessible by removing DRM restrictions. To apply these techniques to DRM-protected content the user must own the content.
- OCRopus™ is an OCR system written in Python, NumPy, and SciPy focusing on the use of large scale machine learning for addressing problems in document analysis (Shafait, 2009).
- mplayer is used to play various audio formats such as WAV, MP3 and OGG.

- brltty is a background process (daemon) which provides access to the Linux/Unix console (when in text mode) for a blind person using a refreshable Braille display. It drives the Braille display, and provides complete screen review functionality (Mielke, 1996).

- Liblouis is an open-source Braille translator and back-translator named in honor of Louis Braille (Boyer,2002). Its features support for computer and literary Braille and translation for many languages. New languages can easily be added through tables that support a rule or dictionary based approach. Tools for testing and debugging tables are included.

- Jstest is the joystick driver for Linux provides support for a variety of joysticks and similar devices. Joystick is considered as an alternate input device in designing CRS.

- Machine Learning Package (mlpack) is a powerful machine learning package to find k-Nearest-Neighbours (kNN) (Curtin et al., 2011).

- Ruby is a dynamic, interpreted, open source programming language with a focus on simplicity and productivity. It is used to generate data table from mathematical graph to produce comprehensive text description.

- Support Vector Machine (SVM) was used for global line labelling in mathematical documents.

4.4 Summary

This chapter studied the results of two surveys which were conducted to understand and analyse the problems that vision-impaired people encounter with accessibility of electronic documents and the communication quality to access graphical components. Results of these two surveys showed in the opinion of majority vision-impaired participants:

- Plain text is most preferred format;

- PDF is most commonly used and yet the most inaccessible electronic document;

- Navigation ability is effective during reading session;

- Use of tactile methods to access graphical components are not common;

- Line chart is most inaccessible chart compared to pie chart and bar chart;

- The combination of tactile and audio method to represent charts is most preferred;

- PIAF and Graphical Braille are not preferred; and
- The use of haptic device to access graphs is not common.

Thus, accessibility of PDF and navigation ability of DAISY are major concern for the vision-impired people. Since plain text is most preferred format due to accessibility through screen reader, other formats such as .odt, .doc and .docx need to be converted to text.

Chapter 5

System Components

5.1 Introduction

This chapter describes the approaches used in this research to resolve issues regarding the following points:

1. Playing DAISY format;
2. Accessibility of Scanned PDF documents;
3. Accessibility of mathematical expressions within the scanned PDF documents;
4. Accessibility of non-textual components within the scanned PDF documents;
5. Accessibility of not-in-order components within the scanned PDF documents; and
6. Method to represent chemical equations.

5.2 Overview of CRS Modules

The CRS reads an electronic document through an input port such as USB port, classifies it considering its extension and assigns it to the appropriate virtual bookshelf to investigate with the correct software. Audio formats are played using “mplayer”. DAISY folders are sent to DAISY Player. All documents such as .doc, .docx, .odt are converted to text and then sent to TTS to be converted to audio or converted to Braille and sent to Braille Terminal. PDF documents are processed with PDF layout analysis module and then scanned PDF documents are converted to text using OCR. PDF documents which are restricted with DRM are converted to html first and then through "html2text" are converted to text to be accessible with screen readers. During PDF layout analysis PDF documents are segmented into blocks, lines and words and non-textual or mathematical components are extracted and sent to Visual Extraction Module (VEM) or Mathematical Information Retrieval (MIR) respectively. Figure 5.1 illustrates an overview of main modules and components of CRS.

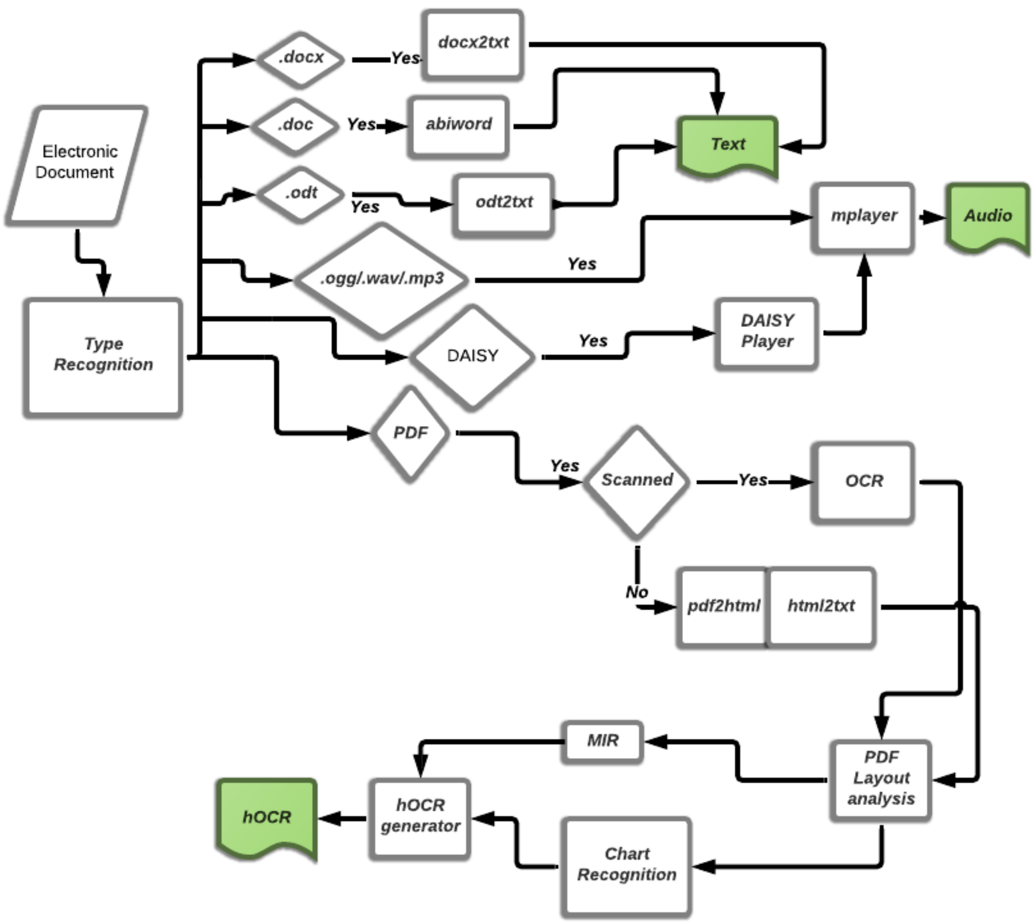


Figure 5.1: Overview of CRS modules

5.3 DAISY Player

5.3.1 Introduction

DAISY is designed for vision-impaired users to access electronic documents. It is a complete audio substitute for print material and technical standard based on MP3 and XML. In addition to traditional audio book, DAISY has advanced features such as searching, place bookmarking and navigation ability by chapter, section, subsection, and page. It allows vision-impaired listeners to navigate something as complex as an encyclopedia or textbook. Readers can read or skip footnotes, sidebars, or information added specifically for users of the audio version.

In terms of accessibility method DAISY has three types:

1. Text-only
2. Audio-only
3. Full text/full audio

It has three formats:

- DAISY 2.02
- DAISY NISO 2002
- DAISY 3/ ANSI/NISO Z39.86 - 2005 (DAISY/.org)

To read DAISY, a standalone DAISY player or computers using DAISY playback software can be used. A DAISY player reads DAISY books in sequential (page by page) or hierarchical manner using structures consisting of (marked-up) text synchronised with audio includes navigation, bookmarking, searching and annotating abilities.

5.3.2 Developed DAISY Player

The developed DAISY player is one major module of CRS and utilities information in Figure 5.2. This figure demonstrates specification and tagging system for three DAISY standards. As ePub structure is similar to DAISY, this research did not consider it as a components of CRS and investigated its accessibility implicitly in DAISY Player.This thesis was determined to consider Daisy as a playback format and not dwell on ePub in too greater detail.

DAISY Standards Comparision	DAISY 2.02	DAISY ANSI/NISO Z39.86 - 2002	DAISY 3 /ANSI/NISO Z39.86 – 2005
Specification	ncc.html	Open Packaging Format (OPF)	Open Packaging Format (OPF)
Format	<meta name="dc:format" content="Daisy 2.02"/>	<dc:Format>ANSI/NISO Z39.86-2002</dc:Format>	<dc:Format xmlns:dc="http://purl.org/dc/elements/1.1/">ANSI/NISO Z39.86-2005</dc:Format>
Tittle	<meta name="dc:title" content=TITLE"/>	<dc:Title>The Coun</dc:TITLE>	<dc:Title xmlns:dc="http://purl.org/dc/elements/1.1/">TITLE</dc:Title>
Duration	<meta name="ncc:totalTime" content="H:M:S" scheme="hh:mm:ss"/>	<meta name="dtb:totalTime" content="H:M:S." />	<meta content="H:M:S." name="dtb:totalTime" />
Navigation	ncc.html	Navigation Center eXtended (NCX)	Navigation Center eXtended (NCX)
Number of navigation levels	<meta name="ncc:depth" content="NUMBER"/>	<meta name="dtb:depth" content="NUMBER" />	<meta content="NUMBER" name="dtb:depth" />
Marks-up for navigation levels		<navPoint class="h1" to h6	<navPoint class="h1" to h6
Number of countable pages	<meta name="ncc:pageNormal" content="NUMBER"/>	<meta name="dtb:pageNormal" content="NUMBER" />	<meta content="NUMBER" name="dtb:totalPageCount" />
Marks-up for pages		<navTarget id="ID " class="pagenum"	<pageTarget class="pagenum" id="ID" playOrder="NMUMBER"
Marks-up for sidebars	<meta name="ncc:sidebars" content="0"/>	Not found	<navTarget class="sidebar" id="ID" playOrder="NUMBER">
Links the elements of the book and ultimately facilitates navigation	Synchronised Multimedia Integration Language (SMIL)	SMIL	SMIL
Marks-up for sequential elements such as paragraph or phrase	<seq> <par><seq></seq></par> </seq>	<par></par>	<par></par>

Figure 5.2: Three DAISY standards comparison

A complete reading session offered by this DAISY player consists of the following steps as shown in Figure 5.3:

- Asking user to connect USB cartridge containing DAISY books to USB port of hardware device;
- Offering available books in cartridge one by one using talking menu to user. User can choose a book by pressing play key or browse talking menu using forward and rewind;
- Offering four different time intervals to user for sleep mode. User can select one by navigation keys or ignore sleep mode by pressing play;
- Recognising the book type, extracting title, total time, heading levels and side bar list;
- Extracting all bookmarks related to chosen books and presenting the Last in First out (LIFO) order to user. A user can choose to read the book from

a bookmark, pressing the play key or browse bookmarks list using forward and rewind.

The end of reading session can occur in two ways:

1. Automatically: when elapsed time passes sleep time; or
2. Manually: by pressing quit key.

The essential information regarding book title, the stop position in book, elapsed time when stopping occurs is appended to the end bookmark table to be available at start of next reading session.

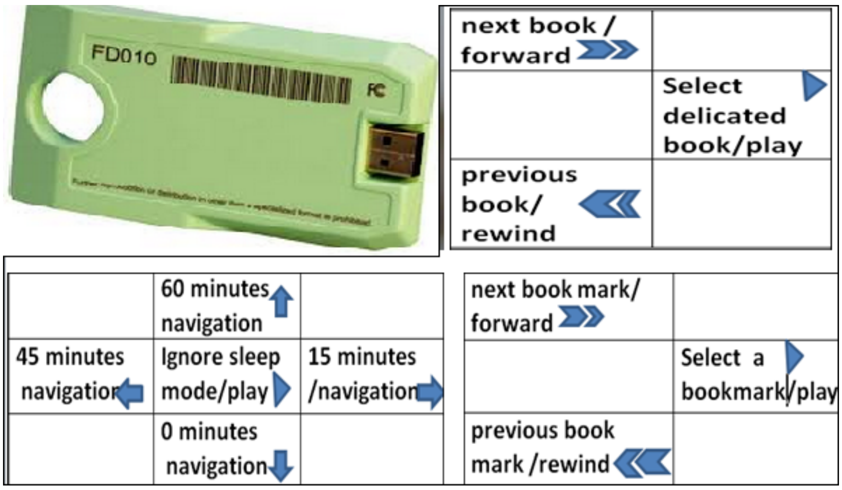


Figure 5.3: Steps of reading session by developed DAISY player

5.3.3 Features of Developed DAISY Player

The developed DAISY player in this research has many features such as:

- Talking user manual, clock and calendar;
- Sleep mode with four different intervals;
- Alternative playback speed and volume;
- Pause and Resume playback function;
- Navigation by levels such as chapter, section, subsection and page;
- Ability to jump forward or backward by specific time interval;
- Info button for announcing title, total time, remaining time and elapsed time;
- Annotation ability to insert, access and delete annotations;

- Bookmarking ability to insert, access and delete bookmark; and
- Using joystick as an interface for the users with motor disability.

5.4 Creation of Accessible and Navigable Document from Scanned PDF

For vision-impaired users, access to electronic documents, including scanned PDF files, has been extremely limited (eGovernment Resource Center, 2007). Even after a scanned document has been processed with OCR software, what is usually left is plain text without tags or mark-up to identify the various components. As this resulting document lacks essential tags, it is not navigable by vision-impaired users. To address this issue and closely preserve the original textual layout, scanned PDF document layout analysis is performed. Several issues that must be considered during layout analysis are the preservation of the correct reading order, and the representation of common logical structured elements such as section headings, line breaks, paragraphs, captions, sidebars, footnotes, running headers, embedded images, graphics, tables and mathematical expressions. PDF layout analysis provides important formatting information that supports PDF component classification as well as creating tagged PDF documents. This classification facilitates the tag generation. Accurate tagging produces a searchable and navigable PDF document. Scanned PDF layout retrieval to provide navigation ability comprises the following steps:

1. Preprocessing, which includes binary conversion by thresholding, scaling, black border removal, margin removal and skew detection and correction;
2. Block segmentation to divide the page into logical blocks and preserve the reading order;
3. Text/non-text segmentation to extract non-textual components such as figures, images and charts; and
4. Line segmentation.

5.4.1 html OCR (hOCR)

To successfully embed the classified data obtained by layout analysis into the text, the final OCR output should be represented in open source standard html, namely, hOCR (html Optical Character Recognition). hOCR is a logical format for representing the output of OCR systems which aims to embed layout, style and other information into the recognised text. Embedding this data into text in the standard html format is used to improve accessibility and achieve navigation ability.

OCRopus is an OCR software package which saves output results in an html file. However, in some cases, different components within a PDF document are not indicated. For example, there is no specific tag for mathematical formulae in hOCR represented by OCRopus. The logical mark-up available in hOCR is designed for the document logical hierarchy, independent of where or how it is rendered on the page. This kind of mark-up is usable for individual documents such as memos, articles, and compound documents such as newspapers, magazines and collections (Breuel & Kaiserslautern, 2007). Table 5.1 indicates existing hOCR tags (left column) and additional tags which is recommended by this research to generate fully marked-up format (right column).

Table 5.1: hOCR Existing tags and recommended tags by this research

Existing hOCR tags	Recommended tags
ocr_document	ocr_caption
ocr_linear	ocr_table
ocr_title	ocr_image
ocr_abstract	ocr_math
ocr_part	ocr_footbar
ocr_chapter{H1}	ocr_sidebar
ocr_section{H2}	ocr_running _header
ocr_subsection{H3}{H4}	ocr_running _footer
ocr_par (paragraph)	
ocr_author	

5.4.2 Segmentation and Layout Analysis

The document layout information is divided into two categories: 1) the geometric layout; and 2) the logical layout (Haralick, 1994). The geometric layout is determined by the positioning information about segments and allows the segments to be categorised into different logical layouts. The detection and labelling of the different zones or blocks as text body, illustrations, mathematical symbols, and tables embedded in a document is called geometric layout analysis. Semantic labelling and classification text zones based on different logical roles inside the document (titles, captions, footnotes, etc) is called logical layout analysis. Document layout analysis is the process of identifying and categorising the regions of interest in the scanned image of a text document.

The layout analysis is the factor, when working with a scanned PDF document that provides the ability for the document to be searched and navigated. It requires the segmentation of text zones from non-textual ones and the arrangement in their correct reading order. Document layout analysis is typically performed before a document image is sent to an OCR engine. Unique areas such as columns, paragraphs and text lines within the page were represented by the physical layout analysis (Haralick, 1994). It is responsible for identifying page components such as, text columns, text blocks, text lines and reading order. Layout analysis works based on identifying the white space and finding the constrained text line.

Layout is a collection of segments: L = [S1,..., Sn], where L and S represent layout and segment respectively. A segment is a pixel collection encapsulated within a bounding box defined by its lower left and upper right corner pixels: S = (P1, P2), where S and P represent segments and pixels respectively. Each pixel is defined by a coordinate pair: P = (x, y).

The different segmentation layers in this research that support layout analysis are:

- Block segmentation;
- Text/non-Text segmentation;
- Line segmentation; and
- Word segmentation.

Documents need to be tagged by component identification using PDF layout analysis to provide scanned PDF navigation ability for use by vision-impaired people.

Extracting Hidden Structures from Electronic Documents (XED) is a reverse engineering tool for PDF documents (Rigamonti et al., 2005). XED discovers and extracts the original document layout structure, and generates an XCDF hierarchical standard form, which is independent of the document type. Firstly, XED cleans the primitives in the original document, taking into account of all types of embedded resources such as raw images and fonts. Then, it recovers the physical structures and represents them in Structured Canonical Document Format (XCDF). XCDF is able to represent the reorganized document in a structured and unique manner that enables the document content to be accessed easily for further work. However, XED is a closed application that works only under the Windows Operating System. Since XED software is not an open source package, it was not used in developing CRS.

5.4.3 Segmentation by Recognition by Adaptive Subdivision of Transformation (RAST) and Voronoi Methods

RAST and Voronoi are two methods which are typically used for layout analysis (Shafait, 2008):

• The RAST algorithm is capable of processing multiple columns documents. In the RAST image result, the column dividers are yellow and different colours are assigned to different segments (Breuel, 2008). RAST is a developed algorithm, consisting of three steps: 1) finding the columns, 2) finding the text-lines, and 3) determining the reading order. RAST starts by extracting the connected components and then it determines the largest possible (maximal) white space rectangles based on the component bounding boxes. These bounding boxes are sorted based on the quantity of connected components touching each of the major sides. In this way, once the column dividers (or gutters) are found, the connected components are examined and classified as text lines, graphics, or vertical/horizontal rulings depending on their shapes and the fact that they do not cross any gutters.

- Named after Georgy Voronoy who created the Voronoi diagram. The Voronoi method identifies the connected components, then extracts sample points along the boundaries to construct a Voronoi-point diagram (Winder, 2010). A large number of edges are created. The unnecessary edges are deleted in an ascending length-wise order, regardless of their connection to other lines. As a result, the Voronoi-point diagram is converted to a Voronoi-area diagram, which represent the page regions. The Voronoi algorithm starts by identifying connected components and then it is able to segment a small collection of complex layouts with the highest accuracy. The Voronoi algorithm divides the page into regions. Voronoi groups blocks of texts in different colours as a result of the segmentation algorithm, but it does not classify them as text or non-text and in some cases, it tends to over segment non-text regions (Kise et al., 1998).

5.4.4 Preprocessing

The PDF document image can be either coloured or greyscaled. Some images are framed with dark borders. The majority of images have top/bottom, left/right margins. In some cases, during the scanning process, the PDF document may rotate slightly. Image colour, size, borders, margins and skew are factors affecting the segmentation and layout analysis result. Therefore, the preprocessing module before starting segmentation is essential to generate smooth input data for the first segmentation layer. The preprocessing includes 6 sub-modules which are described in the following subsections from 5.4.4.1 to 5.4.4.6.

5.4.4.1 Format Conversion

At this stage, the scanned PDF document is converted to the Magick Persistent Cache (MPC) image file format (Still, 2005). MPC is a native in-memory uncompressed file format. This file format is an identical representation of the image in memory. The MPC file format is not portable and not suitable for archiving. However, it is a suitable as an intermediate step for high-performance image processing. The MPC format requires two files to support a single image. The image attributes are written to one file, with the extension .mpc and the image pixels are written to a second file with the extension .cache. The snippet given below first converts PDF to a Portable Pixel Map format (ppm) and then converts .ppm to .mpc using "convert" ImageMagic command-line tool in Linux:

```
pdftoppm document.pdf to document
```

```
convert document.ppm document.mpc
```

5.4.4.2 Convert to Binary

A coloured or greyscaled image is converted into a binary image using threshold technique which starts by designating a separate threshold for each RGB component in the image and then combining them with a logical AND operation (Pham et al., 2007). The following command is used for this purpose:

```
convert document.mpc -threshold 80% binary.mpc
```

5.4.4.3 Scaling

Reducing a large image document by 50% increases the image processing speed during the segmentation process. The following command is used for scaling:

```
convert binary.mpc -scale by 50% half.mpc
```

5.4.4.4 Black Borders Removal

Scanned PDF pages, in some cases are framed with a black border. Without detecting and removing this black border, the margin is not recognisable and removable in the next step. The following steps are used to remove black border from binary image:

1. Negate binary image;
2. Trim negative image to remove margin; and
3. Negate obtained image by step 2 to transfer image to original status.

The following command demonstrates Black Borders Removal process:

```
convert half.mpc -negate mpc:-|convert mpc:- -trim mpc:-|convert mpc:- -
negate RemovedBlackBorder.mpc
```

If a comparison between the dimensions of the obtained output image and the input image shows any difference, it indicates the examined page has black border. The following command is used to acquire dimensions of obtained file.

```
info=$(identify -format "%@" RemovedBlackBorde.mpc)
```

5.4.4.5 Margin Removal

A PDF document layout analysis is based on block segmentation. Accurate recognition of white space areas produces block segments separators. Top-bottom and left-right margins often interfere with the accurate recognition of white space areas as block segments separators. The Margin Removal module removes all margins before running the block segmentation. The following command is used to remove margin from the binary, scaled and black-borders-removed image (image.png)

```
convert image.png -fuzz 0% -trim +repage RemovedMargin.png
```

Alternatively, Margin Removal is performed by finding the bounding box of the foreground pixels and cropping the input image using the obtained bounding box as it represents by the following snippet :

```
(Xmin-ForegroundPixels, Xmax-ForegroundPixels)
(Ymin-ForegroundPixels, Ymax-ForegroundPixels)
Y=Ymax-Ymin
X=Xmax-Xmin
convert RemovedBlackBorder.mpc -crop X x Y + Xmin +Ymin
  RemovedMargin.mpc
```

The code for Margin Removal is available in Appendix B. Figure 5.4 illustrates an image before (left) and after margin removal (right).

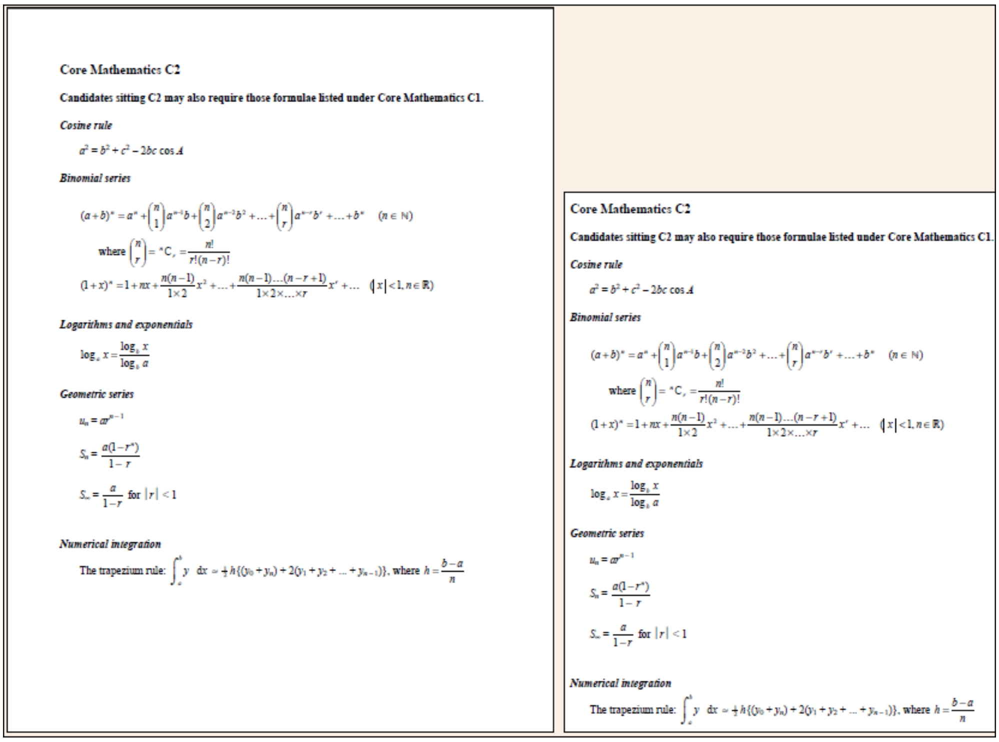


Figure 5.4: An image before (left) and after (right) margin removal

5.4.4.6 Skew Detection and Correction

In some cases, during the process of scanning a PDF document, the document may rotate slightly. This skew reduces the accuracy of PDF layout analysis. To address this issue, skew detection and correction module is applied before running the PDF layout analysis. This module obtains the bounding box of the foreground pixels in the image and extracts the two coordinate values for foreground pixels (X-in-Ymax,Ymax), (Xmin,Y-in-Xmin). Figure 5.5 illustrates how a tangent of rotation angle is obtained by two steps:

1. Subtract "Y-in-Xmin-ForegroundPixels" from "Height of the image"; and

2. Divide result of step 1 to "X-in-Ymax-ForegroundPixel"

The result of applying the tangent of rotation angle to the Arc-tangent function is the rotation angle. After skew detection is completed by obtaining rotation angle, skew correction is performed by rotating original image clock-wise by rotation angle. The following snippet demonstrates skew detection process:

```
Xymax=X-in-Ymax-ForegroundPixels
Yxmin=Y-in-Xmin-ForegroundPixels
H=Height of the image
tang(Rotating-Angle)=(H-Yxmin)/Xymax
Rotating-Angle=arctang(H-Yxmin/Xymax)
```

This command is used for skew correction:

```
convert page.mpc -rotate -$Rotating-Angle SkewCorrected.mpc
```

The code for Skew Detection and Correction is available in Appendix B. Figure 5.5 illustrates a rotated image document and relation between Xymax, Height and Yxmin and Rotating-Angle.

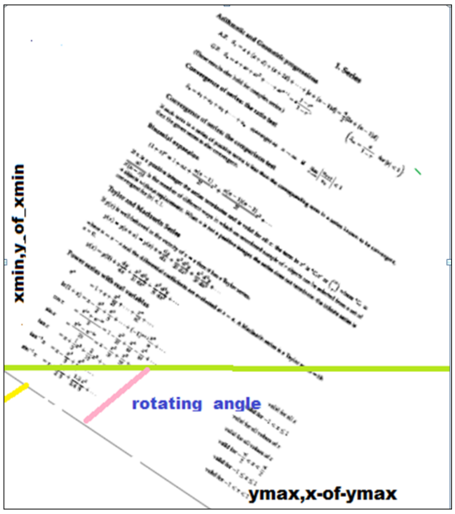


Figure 5.5: Rotated PDF document

5.4.5 Block Segmentation

If the PDF page contains only one column, lines in the OCR result will be presented in order. However in cases where the PDF includes more than one column, there is no guarantee that the OCR result will convey the correct line ordering. In a double column document the OCR results do not keep the reading order. Block segmentation divides the page into logical blocks, preserving the reading order. Voronoi can be used in this module for segmentation purpose. Figure

5.6 (top-right) illustrates over segmentation in some points such as subtitle and sidebar which are represented with more than one colour.

The proposed method is responsible for identifying the extended vertical black lines or the vertical white space. This module utilizes morphological operations for block segmentation to avoid over segmentation.

Image Morphology is a technique which is used for block segmentation in this research. In the morphology method, the structure of shapes within an image can be cleaned up and studied. The method works by comparing each pixel in the image against its neighbours in various ways, either adding/removing or brightening/darkening that pixel. By applying morphology repetitively over a whole image, specific shapes can be found, removed or modified. If a pixel is white and completely surrounded by other white pixels, then that pixel is obviously not on the image edge.

The entire process actually depends on the definition of a structuring element or kernel, which defines what pixels are to be classed as neighbours for each specific morphological method.

The dilate operation returns the maximum value in the neighbourhood. The erode operation returns the minimum value in the neighborhood. Eroded morphology helps to remove non-interested white spaces between text words and recognises remaining white space area as block separators. The following command is used for eroded morphology.

```
convert bin.mpc -morphology erode :4 diamond eroded.mpc
```

Since PDF document may contain different logical layouts such as footer and multiple columns, block segmentation must be done both vertically and horizontally using the following steps:

- Considering the horizontal white space area ;
- Running horizontal segmentation;
- Considering vertical white space area in each extracted horizontal segment; and
- Applying vertical segmentation for all extracted horizontal segments.

Figure 5.6, top-left illustrates a multiple columns document, Figure 5.6, top-right is the result of block segmentation with the Voronoi. As Figure 5.6, top-right shows some over-segmentation occurrences in sub-title and sidebar. This research did not use Voronoi for block segmentation. Figure 5.6, bottom-left is a multiple columns document and bottom-right is Block Segmentation results of proposed method in this research.

Figure 5.6: Top-left: a multiple columns document including sidebar. Top-right: the result of using Voronoi segmentation. Bottom-left: a multiple columns document. Bottom-right result of using Horizontal/Vertical segmentation (blocks are specified with blue lines separator)

The code for the Block Segmentation is available in the Appendix B.

5.4.6 Text/non-Text Segmentation

Morphology is also used in the separating textual components from non-textual components in image documents such as scanned PDF. Non-textual parts contain fatter lines and larger blobs than text parts. Thus non-textual parts can be extracted by dilating the image until all letters are gone. Then, the eroding is performed on the dilated image using the original binary image as the mask until the non-textual part is complete again. The original image is used as a mask to protect non-textual parts of the image from changes. Using original image as mask will restore all shapes that still have some seed part left, so only the non-textual part is left. Now non-textual component is extracted from binary image of document. Subtracting non-textual part from original binary image of document results in text-only image.

Figure 5.7 illustrates this approach step by step.

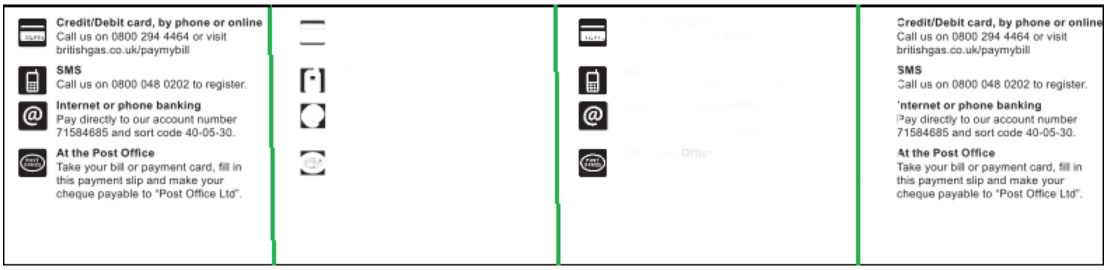


Figure 5.7: Left to right: A sample Binary-Block-Segment, dilated, non-Textual and Text-Only images

The following snippet supports Figure 5.7 and described method for text/non-text segmentation

```
convert Binary-Block-Segment.mpc -morphology dilate: 3 diamond
   dilated.mpc
convert dilated.mpc -morphology erode:20 diamond -clip-mask
   Binary-Block-Segment.mpc non-Textual.mpc
convert non-Textual.mpc -negate Binary-Block-Segment.mpc -compose
    plus -composite Text-Only.mpc
```

Figure 5.8 illustrates a sample PDF containing non-textual component (left), non-Textual segment (middle) by morphology and Textual segment (right) by image subtraction (compose plus composite) techniques.

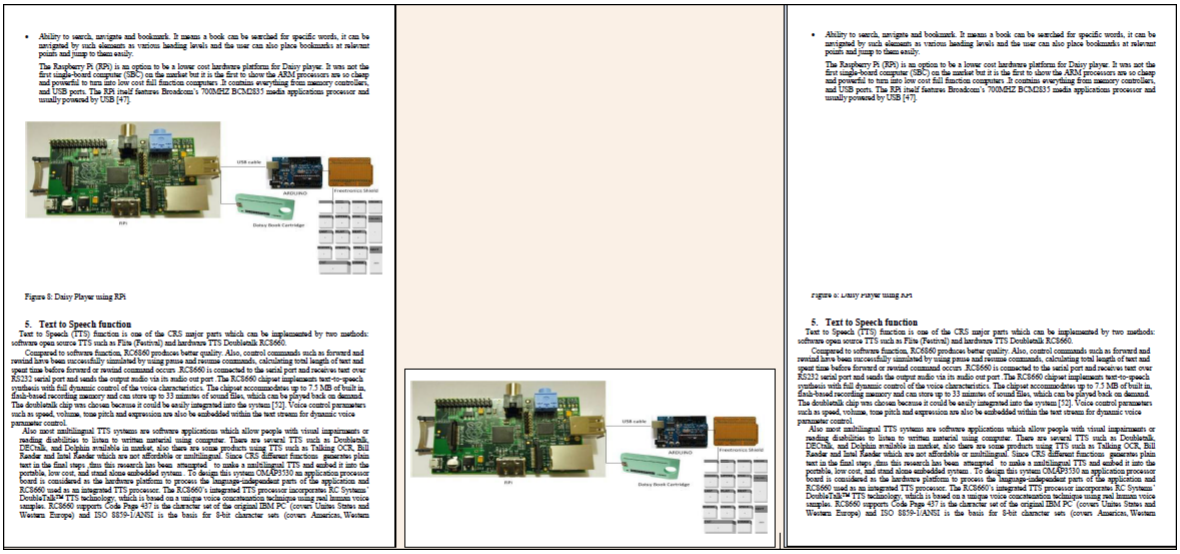


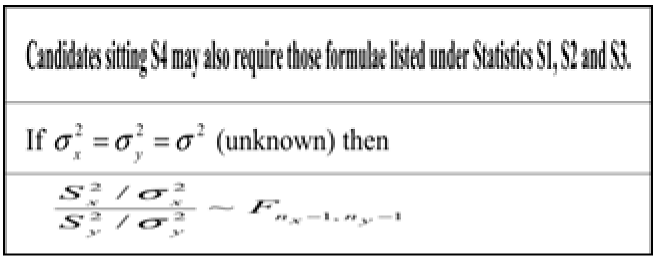
Figure 5.8: Text/non-Text Segmentation

5.4.7 Line Segmentation

The method utilised in this research for PDF line segmentation is One-Column-Projection (1CP) which also supports mathematical documents. When Block Segmentation and Text/non-Text segmentation are completed, 1CP runs on each single block to obtain bounding box of each line. Obtained bounding boxes are used for logical layout analysis.

Lines in Mathematical PDF documents are categorised into three different types:

1. Text lines (the top of Figure 5.9) [1];
2. Embedded Formulae (EF) lines where mathematical expressions are mixed with text (the middle of Figure 5.9) ; and
3. Isolated Formulae (IF) lines (the bottom of Figure 5.9)



Candidates sitting S4 may also require those formulae listed under Statistics S1, S2 and S3.

If $\sigma_x^2 = \sigma_y^2 = \sigma^2$ (unknown) then

$$\frac{S_x^2 / \sigma_x^2}{S_y^2 / \sigma_y^2} \sim F_{n_x - 1, n_y - 1}$$

Figure 5.9: Different types of lines in mathematical documents:Top: Text, Middle: EF and Bottom: IF

[1]Note: The quality of this image is poor(blur) as image was extracted from the real expression in scanned PDF

In terms of line segmentation accuracy in mathematical PDF documents, lines are divided into four categories :

1. Fully-Segmented: the line is completely segmented with its subscript or superscript. (Figure 5.10) [2]

2. Over-Segmented: the line can be split into more than one line, or partially detected. Figure 5.11 illustrates the 2-dimensional IF line, and shows over-segmentation occurrence during ordinary line segmentation, which splits one line into two separate segments. [3]

3. Under-Segmented: the line is merged with some other lines. Figure 5.12 indicates two IF lines and shows under-segmentation occurrence during ordinary line segmentation, which merges the bottom line with the superscript of the previous line and subscript of the next line. [4]

4. Broken symbol or character: Figure 5.13 shows a 2-Dimensioal EF line with a tall symbol, which is broken during ordinary line segmentation. [5]


For a random sample of n observations from $N(\mu, \sigma^2)$

Figure 5.10: A Fully-Segmented EF line

[2]Note: The quality of this image is poor(blur) as image was extracted from the real expression in scanned PDF

[3]Note: The quality of this image is poor(blur) as image was extracted from the real expression in scanned PDF

[4]Note: The quality of this image is poor(blur) as image was extracted from the real expression in scanned PDF

[5]Note: The quality of this image is poor(blur) as image was extracted from the real expression in scanned PDF

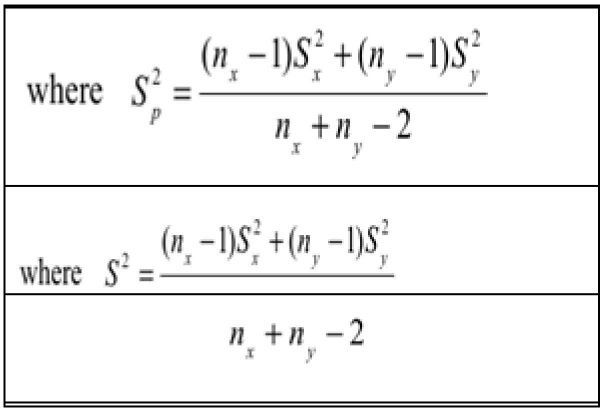


Figure 5.11: A 2-D IF line (top) and over-segmentation result (middle and bottom)

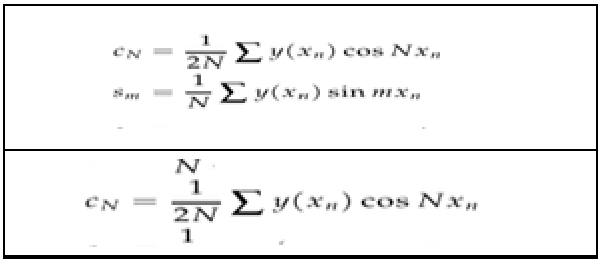


Figure 5.12: Two 2-D IF lines (top) and under- segmented IF 2-D line result (bottom)

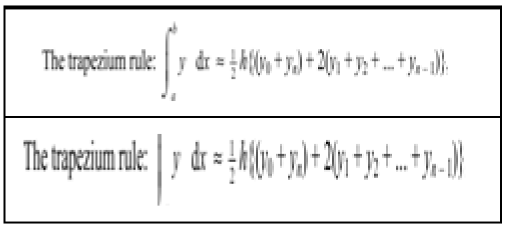


Figure 5.13: EF 2-D line (top) and broken symbol (bottom)

Documents that contain mathematical expressions have some issues which must be considered during line segmentation. These issues are:

- Non-linearity;
- Being two-dimensional including summations, products, integrals, roots, fractions, etc; and
- Symbols of various size and shape such as brackets and fraction bars.

In this research, line segmentation is performed based on specifying the rectangular background (white space) as horizontal gap between text lines in the image by finding the position of all lines which meet the following conditions:

Assuming BL and FL are background lines (gaps) and foreground lines (text) respectively.

BL_i is a text lines horizontal gap:
If length(BL_i)=width(block)=W & L_{i-1} is not similar BL_i
Assuming BL_i and BL_j are two consecutive horizontal gaps
Height(FL_i) $= Y(BL_j) - Y(BL_i)$
FL_i is obtained by cropping original block image:
convert block.mpc Wx *Height(FL_i) +0+$Y(BL_i)$* FL_i.mpc

The line segmentation module performs based on detecting the consecutive foreground pixels which have following conditions:

Assuming n background Pixels as:

$BP_{i,}......BP_n$
if $n = Width(Page)$
for $(i = 1..n)$
if $(j = i + 1$ && $X(BP_j) = X(BP)_i + 1$ && $Y(BP_j) = Y(BP_i))$ then
$Y - BP_i$ is a segmentation position.

Figure 5.14 illustrates a mathematical document (top) and its line segmentation result. In this result segmented lines are separated with blue lines (bottom).[6]

[6]Note: The quality of this image is poor(blur) as image was extracted from real scanned PDF

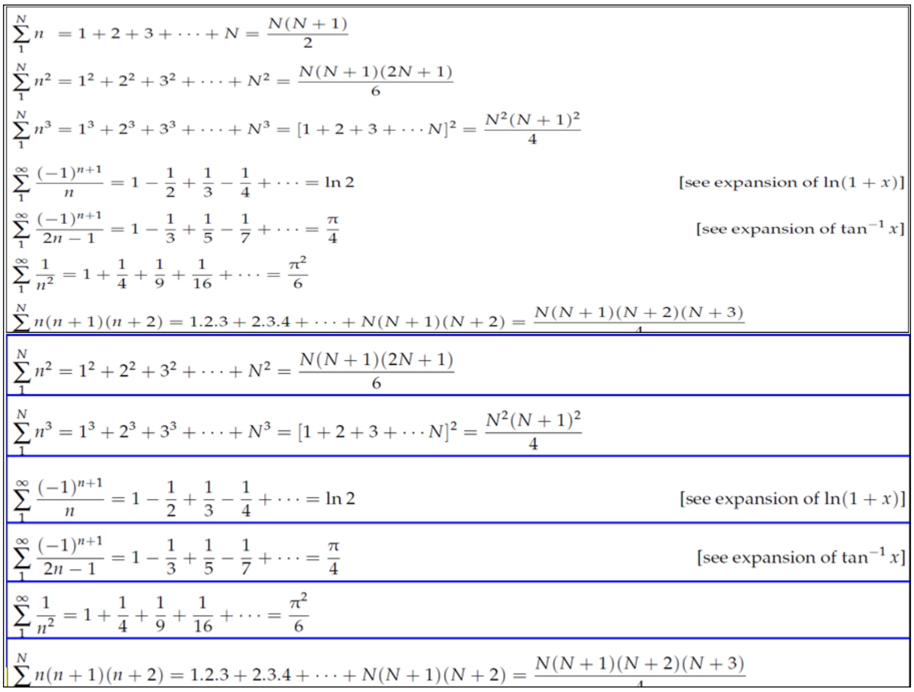


Figure 5.14: Mathematical PDF document (top) and its line segmentation result (bottom)

The proposed line segmentation method solves the problems of mathematical PDF (Figures 5.10 to 5.13). Over-segmentation, under-segmentation and broken symbol issues which occur during mathematical document line segmentation do not happen in this method due to consider the vertical white space with maximum width as separators between lines.

The code for line segmentation is available in Appendix B.

5.5 Mathematical Information Retrieval (MIR) from Scanned PDF

5.5.1 MIR Overview

Mathematical expressions are complicated and require technical expertise to describe. Using images of mathematical or graphical representations create a significant barrier in accessing mathematical documents by vision-impaired students. To avoid graphical representations of the mathematical expressions, designing a practical way to communicate mathematical content to vision-impired is essential. Mathematical Information Retrieval (MIR) from scanned PDF documents is considered to address this issue. MIR is performed after line segmentation. The different steps involved in MIR are:

1. Detecting and extracting mathematical expressions (Garain & Chaudhuri, 2000)
2. Recursive primitive component extraction;
3. Non-alphanumerical symbols recognition;
4. Structural semantic analysis to find the role of each symbol;
5. Merging primitive components to generate linear, non-ambiguous, navigable and detailed text description;
6. Generating mark-up format such as MathML from text description; and
7. Sending mathematical expression in mark-up format or text description to TTS and audio generation.

The illustrated flowchart in Figure 5.15 shows the different modules of MIR.

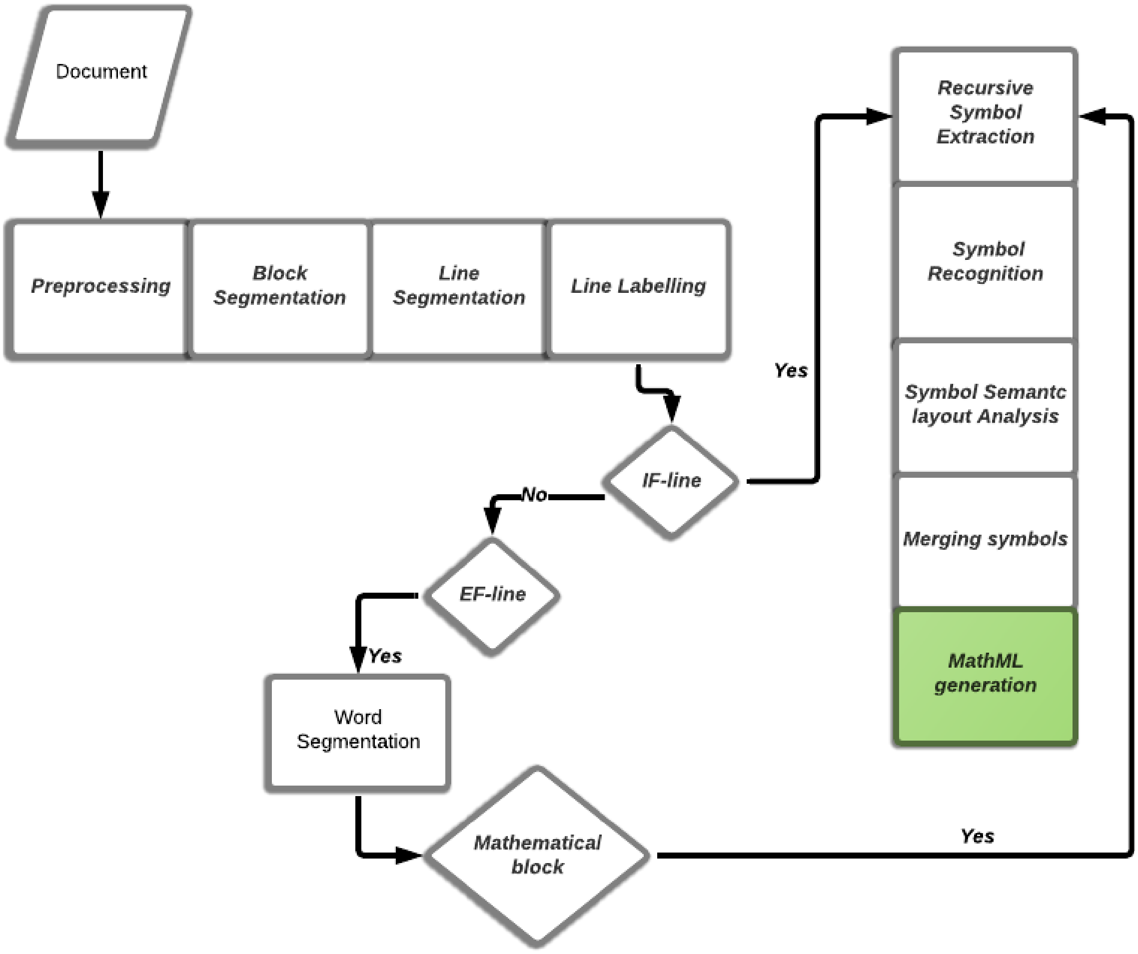


Figure 5.15: Overview of MIR

5.5.2 Accessible Formats

Two of the mark-up formats for presenting mathematical expressions are:

1. LaTeX allows simple construction of mathematical formulae and provides professional look when printed. Typesetting mathematics is one of LaTeX's greatest strengths (Madsen et al., 2013). If a document contains only a few simple mathematical formulae, plain LaTeX may be sufficient and it has most of the required tools. For a scientific document that contains numerous complicated formulae, the Amsmath package introduces several new commands that are more powerful and flexible than the ones provided within the standard LaTeX package. Amsmath is a LaTeX package that provides

miscellaneous enhancements for improving the information structure and printed output of scientific documents containing mathematical formulae (American Mathematical Society, 1999)

2. MathML supports navigation ability by assistive technology. MathML equations can be accessed via standard assistive technology applications, such as synthetic speech technologies. MathML deals with the presentation and the formula meaning. Web pages with MathML embedded in them can be viewed as normal web pages with different web browsers. MathML presentation focuses on the display of an equation, and has approximately 30 elements (Madsen et al, 2013). All names begin with the letter "m". The commonly used MathML elements are:

   - <mi>x</mi>– identifiers
   - <mo>+</mo>– operators
   - <mn>2</mn>– numbers
   - <mtext>non zero</mtext>– text.
   - <mrow>– a horizontal row of items
   - <msup>,<munderover>, and others – superscripts, limits over and under operators like sums
   - <mfrac>– fractions
   - <msqrt>and<mroot>– roots
   - <mfenced>- surrounding content with fences, such as parentheses.
   - Many entities are available for specifying special symbols by name, such as π and & RightArrow;
   - MathML has entities to express normally-invisible operators, such as &Inv

5.5.3 Mathematical Expression Accessibility Issues in Scanned PDF Documents

Screen readers and TTSs read text in linear mode from left to right. In contrast with text, mathematical expressions are not linear and reading them in linear manner causes ambiguity and misrepresentation to result. A problem in using TTS to read mathematical formulae is the two dimensional nature of mathematical expressions which are not conveyed with traditional screen readers. In order to address this issue, mathematical expressions must be rendered and converted to linear textual format before sending to TTS. In addition, often the mathematical expressions are displayed as picture and not text. Dealing with the large range of mathematical symbols in OCR systems is problematic. According to LaTeX documentation, there are approximatlly 500 symbols for mathematical expressions (Berman & Fateman, 1994).

Many mathematical symbols are represented by non-alphanumeric symbols. The main OCR task is to capture information from non-text documents and save them for easy retrieval. Many OCR systems capture only the textual information and leave all or part of mathematical expressions unrecognised (Miller & Viola, 1998). In most cases, even if mathematical expressions are recognised, the output results of TTS still do not make sense. Ambiguity in mathematical expressions makes Mathematical OCR (MOCR) more complicated than normal OCR. There are various reasons for ambiguity in mathematics. Some of the reasons are:

- In literary text, characters are sequentially read from left to right in the case of Latin languages. Mathematical expressions do not necessarily follow this rule. In some cases, finding the first element to be vocalized is problematic;

- Mathematical expressions are not linear. Hence it is problematic to find the relationship between different symbols;

- Some symbols for building formulae have different shapes in different situations but keep the same meaning (Miller & Viola, 1998). In mathematical formulae, the meaning of different elements depend on their shape and spatially occupied position. For example, the root symbol$\sqrt{x}$ grows in height and width to fit its contents, or fraction bars$\frac{a}{b}$ grow in length to fit their numerator or denominator. Multi-Shaping symbols can cause problems in building classifier to recognise mathematical symbols;

- Some similar symbols such as minus and fraction symbols have a similar shape with different meaning and can only be distinguished by neighbourhood information; and

- Many mathematical symbols are used in more than one mathematical area with different usage and meaning. These symbols are Multi-Meaning in various mathematical areas. For example, the meaning of symbol '*' in basic math is multiplication but in Calculus and analysis it is convulsion.The various mathematical areas comprise of the following:

    1. Algebra;
    2. Basic mathematics;
    3. Calculus and analysis;
    4. Geometry;
    5. Probability and statistics; and
    6. Set theory logic.

5.5.4 Previous Methods of Mathematical Formulae Extraction

Currently there are several methods that facilitate mathematical formulae extraction from scanned documents. Jin et al proposed a Mathematical Formulae Extraction method. This method classifies each line as either an Isolated Formulae (IF) or a non-Isolated Formulae (Non-IF) using the Parzen Window Classifier technique (Jin et al., 2003).

The Parzen Window Classification method is a technique for non-parametric density estimation that can also be used for classification (Duda & Hart, 1973). Each class density is separately approximated and a test point, with the maximal posterior probability is assigned to each class. The resulting algorithm is extremely simple and closely related to Support Vector Machines (SVM).

Generally, mathematical expressions are either one-dimensional (1-D) or two-dimensional (2-D). After line classification, the Jin et al. method uses 2-D structure detection. This method does not support 1-D structured Embedded Formulae (EFs.)

A different approach to Jin et al method for Mathematical Formulae Extraction was presented by Kacem et al. This method is based on connected component identification and uses several features, such as bounding boxes for IF. The EF detection and extraction are dependent upon the results from both symbol recognition and OCR (Kacem et al.,2000). This method specifies the location of the most significant symbol and extends it for adjoining symbols using contextual rules. This is continued until the entire formula is delimited, usually through the use of white spaces (Kacem et al, 2001).

Lin et al identified embedded mathematical formulae in PDF documents using SVM (Lin et al., 2012). This method carries out a word segmentation over the scanned document and then classifies words as either ordinary text or as a formula fragment using SVM classifier. Since an isolated expression has a recognisable geometric layout, most of the previous methods for IF detection and extraction are rule-based. However, embedded expressions are often short and separating equations from text is problematic. Character-based separation is a method to recognise the location of embedded expressions. This method works in conjunction with OCR and specifies the location of some symbol position such as position of equal sign within the line. In cases where the line does not contain the specified symbol, the EF's position remains ambiguous.

The method undertaken in this research to extract mathematical expressions from scanned PDF uses the following steps:

- Line segmentation ;
- Line labelling or Classifying line segments to IF and non-IF;
- Applying word segmentation for non-IF lines;
- Classifying word blocks to text block and mathematical block; and

- Sending IF obtained by step 2, and EF by step 4 to Recursive Components Extraction (RCE) to convert them to a linear form.

5.5.5 Global Line Labelling by Feature Extraction and SVM

Completed line segmentation using 1CP provides lines bounding boxes which are defined by lower left and upper right positions. Obtaining line bounding box helps to extract other features such as Height, Left and Right Margin, Vertical Space and Density. Assuming bounding box for a line is represented as:

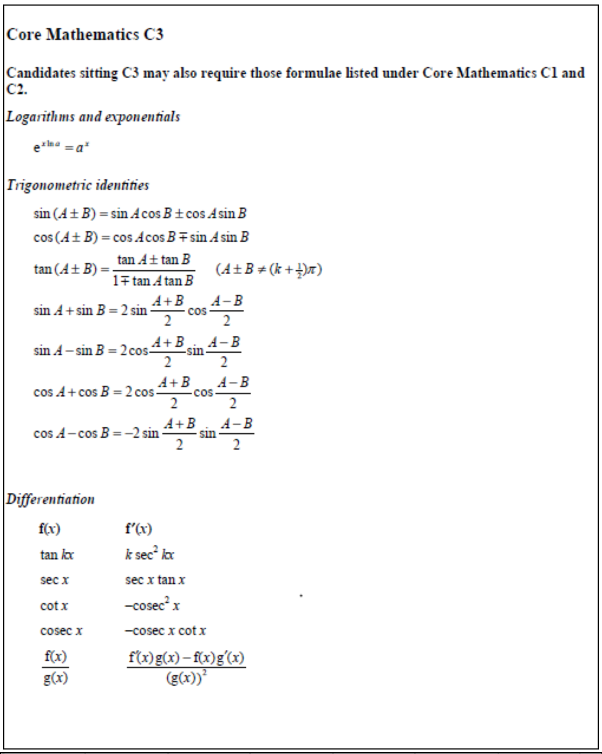
```
Bounding box= (Xmin, Ymax, Xmax, Ymin)
```

Other features are obtained as :

```
Height: H=Ymax-Ymin
Width: W=Xmax-Xmin
Aspect Ratio: AR=H/W
Area=W*H
LeftMargin: LM=Xmin
Number of Black Pixecls=NBP
RightMargin: RM=W-Xmax
VerticalSpace: VS=Ymin(i)-Ymax(i-1) for two adjacent lines (
   Assuming i and i-1 are two consective lines)
Density=NBP/Area
```

Figure 5.16 shows a sample of a mathematical document and extracted features of its segmented lines.[7] These features are: left margin, right margin, height, width. vertical space and aspect ratio.

[7]Note: The quality of image is poor(blur) as image was extracted from real mathematical PDF document



Core Mathematics C3

Candidates sitting C3 may also require those formulae listed under Core Mathematics C1 and C2.

Logarithms and exponentials

$$e^{x \ln a} = a^x$$

Trigonometric identities

$$\sin(A \pm B) = \sin A \cos B \pm \cos A \sin B$$
$$\cos(A \pm B) = \cos A \cos B \mp \sin A \sin B$$
$$\tan(A \pm B) = \frac{\tan A \pm \tan B}{1 \mp \tan A \tan B} \quad (A \pm B \neq (k + \tfrac{1}{2})\pi)$$
$$\sin A + \sin B = 2 \sin \frac{A+B}{2} \cos \frac{A-B}{2}$$
$$\sin A - \sin B = 2 \cos \frac{A+B}{2} \sin \frac{A-B}{2}$$
$$\cos A + \cos B = 2 \cos \frac{A+B}{2} \cos \frac{A-B}{2}$$
$$\cos A - \cos B = -2 \sin \frac{A+B}{2} \sin \frac{A-B}{2}$$

Differentiation

$f(x)$	$f'(x)$
$\tan kx$	$k \sec^2 kx$
$\sec x$	$\sec x \tan x$
$\cot x$	$-\operatorname{cosec}^2 x$
$\operatorname{cosec} x$	$-\operatorname{cosec} x \cot x$
$\frac{f(x)}{g(x)}$	$\frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2}$

Line	LM	RM	H	W	VS	AR
1	13	221	7	74	0	0.094595
2	13	20	8	275	11	0.029091
3	13	286	7	9	0	0.777778
4	13	213	8	82	6	0.097561
5	26	257	7	25	6	0.28
6	14	228	8	66	12	0.121212
7	26	181	8	101	5	0.079208
8	26	180	8	102	3	0.078431
9	26	145	17	137	3	0.124088
10	26	183	17	99	3	0.171717
11	26	184	17	98	2	0.173469
12	26	180	17	102	3	0.166667
13	26	178	17	104	2	0.163462
14	13	254	8	41	20	0.195122
15	29	227	7	52	6	0.134615
16	29	216	9	63	3	0.142857
17	29	211	6	68	6	0.088235
18	29	213	8	66	4	0.121212
19	29	201	6	78	6	0.076923
20	30	182	19	96	5	0.197917

Figure 5.16: A sample of mathematical document (left) and extracted features of its segmented lines (right). These features are: left margin (LM), right margin (RM), height (H), width (W), vertical space (VS) and aspect ratio (AR)

Segmented lines by single column projection technique are divided into several categories. These categories are:

- Text only;
- Linear Embedded Formula;

- Multi-Dimensional Embedded Formula(EF);
- Multi-Dimensional Isolated Formula(IF);
- Linear Isolated Formula;
- Caption;
- Running Footer;
- Running Header; and
- Heading Title.

The lines are classified using features such as: Left Margin, Right Margin, Height, Density, and Vertical Spaces (Xiaoyan et al., 2013). Then the lines which are labelled as IF or EF are sent to the Mathematical Information Retrieval (MIR) module for further investigation. The threshold value of each feature is used to compare features in the data sets. Mean, mode and median are used to find the threshold value. These three values are calculated with the following formulae:

$Mean(x) = \frac{\sum_{i=1}^{n}(x_i)}{n} \quad 1<i<n$

$Median(x) = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$

$Mode(x) = most\ frequent\ x$

If Mode(x) >Mean(x) and Mode(x)>Median, the data set is skewed to the left or it is negatively skewed. In this situation the mean and the median are both less than the mode. Generally, most of the time when the data is skewed to the left, the mean will be less than the median.

If: Mean(x) <=Median(x) <=Mode(x) then Threshold-value(x) =Mode(x) (Figure 5.17 left)

If: Mode (x) <Mean(x) and Mode(x) <Median(x) it means the data set is skewed to the right then Threshold-value(x) = Median(x) (Figure 5.17 right)

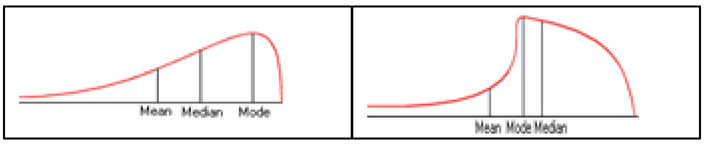


Figure 5.17: Threshold value=Mode(x) (left), Threshold value=Median(x)(right)

Feature i represented by f_i were collected in a file (file.dat) as a data set. In order to obtain the mode, frequency, mean, median, minimum and maximum of f_i, the below snippet is used:

```
Mode=$(cat file.dat| awk '{ a[$1]++}END { for(i in a) print i,a[i
    ]}' |sort -b k1n,1|awk 'END {print $1}')
Freq=$(cat $file,dat| awk '{ a[$1]++}END { for(i in a) print i,a[
    i]}' |sort -b -k2n,2|awk 'END {print $2}')
```

```
Mean=$(awk '{sum+=$1} END { print sum/NR}' file.dat)
Median=$(sort -b -k1n,1 cat file,dat | awk ' { a[i++]=$1; } END {
    x=int((i+1)/2); if (x < (i+1)/2) print (a[x-1]+a[x])/2; else
   print a[x-1]; }')
Min=$(cat file.dat|sort -b -k1n,1|awk 'NR==1{print $1}')
Max=$(cat file.dat|sort -b -k1n,1|awk 'END {print $1}'))
```

The line labelling module first focuses on mathematical expression attribute to classify IF. If a line meets one of the following conditions, it may consider as an IF.

1. If $Height > Height(TextOnlyLine)$. Mathematical expressions include subscript and superscripts (S&S) are often written using 2-3 rows to accommodate them.

2. If the line includes a horizontal line separating the numerator and the denominator

3. If $Density < Threshold(Density)$

Figure 5.18 illustrates the flowchart for global line labelling using following abbreviations.

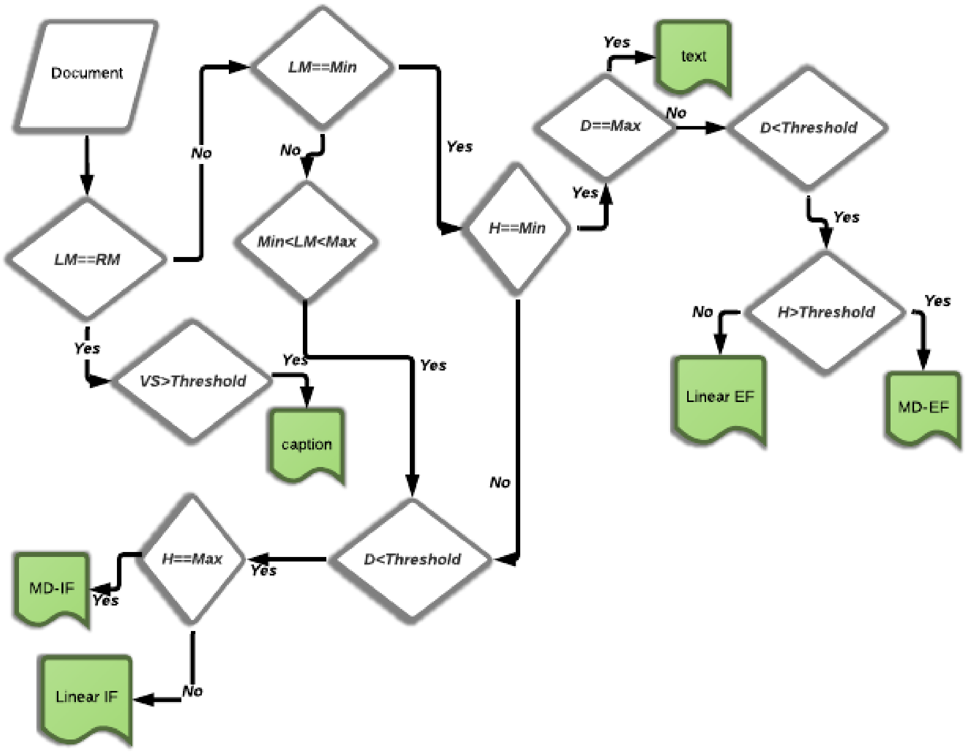


Figure 5.18: Global Line Labeling

```
W=Width(Line)
H=Height(Line)
D=Density
LM=Left-Margin
IF=Isolated-Formula
EF=Embedded-Formula
VS=Vertical-Space
I=Line-Number
MD=Multi-Dimensional
```

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. SVM algorithms calculate the optimal hyperplane to categorise new examples using labelled training data. This is known as supervised learning (Tong & Koller, 2002). Utilizing SVM as a classifier needs a training data file and a model file. In order to generate a training data file for the binary classifier, sample lines were collected from more than 40 pages (about 700 lines) from Mathematical Formula Handbook and the N38210A-GCE Mathematical Formulae Statistical Tables. Extracted features are represented as a vector= [H, LM,RM, VS, Density](Garain, 2009).

Target values are then manually assigned to each line, 1 and -1 values are assigned to non-IF, and IF respectively to classify IF or non-IF. Most of title lines are indented and have more space than threshold of vertical spaces in page. Therefore the lines with the following conditions are considered heading line and labelled as non–IF lines:

$LM \neq Mode(LM)$
$VS > Threshold(VS)$

The next step is the SVM model file generation utilizing svm _learn tool from SVM package under Linux. This tool creates a model file based on collected data in train-data.dat

```
svm_learn train-data.dat model
```

To apply a new line to the classifier, SVM model file and test data file containing feature vector are applied to svm_clasify from SVM package under Linux. This tool generates Result.txt based on information of model and test files.

```
svm_classify data.dat model Result.txt
```

"Result.txt" includes classifier result for data.dat file.

5.5.6 EF Extraction Using Word Segmentation

Isolated formulae line detection is performed based only on the features of the geometric layout and SVM classification because text lines and isolated expressions are significantly different in properties such as height, vertical space, character sizes and symbol layout (Garain, 2009; Yamzaki et al, 2011) whereas position identification of an embedded mathematical expression inserted in between ordinary text is more complicated. Unlike ordinary text, mathematical expressions in most cases are 2-Dimentional. Various types of 2-D mathematical syntax include:

- Fraction $\frac{a}{b}$;
- Root $\sqrt[3]{x}$;
- Integral $\int x$;
- Symbol with superscript and/or subscript(Aly et al., 2008);
- Symbols with right above and/or right under it. $\prod x \sum x$;
- $\lim x$; and
- Accent sign $\bar{y}$.

SVM classification for global line labelling supports the classification of all non-IF labeled line either EF or text only. Labeled lines are sent to next module for further investigation. Generally, an embedded mathematical formulae may contain subscripts and superscripts, Radial symbols, Functional, Integral symbols, Horizontal fraction bar extensions and Binary operator extensions.

This module consists of word segmentation based on the horizontal space between two adjacent unconnected components if it is supposed that PDF documents contain perfect typesetting information.

Assuming BCC and FCC are Background Connected Components (gaps) and Foreground Connected Components (word) in a segmented line respectively.

BCC_i is a word gap:
If height (BCC_i)=height(line)=H & CC_{i-1} is not similar CC_i
Assuming BCC_i andBCC_j are two consecutive horizontal gaps
$Width(FCC_i) = X(BCC_j) - X(BCC_i)$
FCC_i is obtained as a segmented word by cropping segmented line
convert line.mpc Width (FCC_i) x H+$X(BCC_i)$+0 FCC_i.mpc

The word segmentation module performs based on detecting the consecutive foreground pixels which have following conditions:

Assuming n background Pixels as:

$BPi, ...BPn$
If $n = Height(SegmentedLine)$
    For $(i = 1...n)$
        If ($j = i + 1$&& $Y(BP_j) = Y(BP_i) + 1$ && $X(BP_j) = X(BP_i)$)
            $X - BP_i$ is a word segmentation position
        Endif
    Endfor
Endif

In addition, if the current character is one of the special separators, a word should be detected. The special separators include, punctuation marks (e.g., comma, period, dash, colon, etc.), and parentheses (e.g.,(,), [,],etc. The code for word segmentation is available in Appendix B. Figure 5.19 shows an EF line (top) and word segmentation result of the sample line (bottom). Vertical blue lines illustrate word separator.

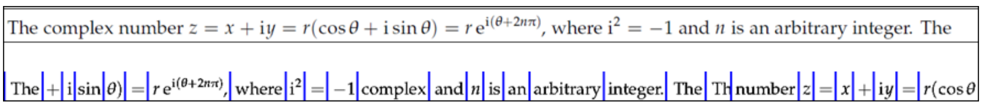


Figure 5.19: An EF line sample (top) and word segmentation result of the sample line (bottom).

Word segmentation is followed by data collection, feature vector generation and SVM word classification to extract the embedded formulae.The features used by SVM in this module include:

```
Width:W= Xmax-Xmin
Height:H= Ymax-Ymin
```

```
Horizontal-Space: HS= Xmin(i)- Xmax(i-1) : Assuming w(i ) and w(i
    +1) are two adjacent words
Aspect-Ratio: AR= H/W
Area=W*H
Density(i= Number of black pixels/Area(i)

x-deviation-word= The amount of variation or disperion of x from
    average
```

$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - mean(x))^2}{n}}$

```
y-deviation-word=The amount of variation or disperion of y from
    average
```

$\sigma_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - mean(y))^2}{n}}$

SVM classification contains the following two steps:

1. Model file creation by training collected data ; and
2. Online word block classification using its feature vectors and model file.

Classified text blocks with SVM are applied to character based classification (Fujiyoshi et al., 2008). If a text block contains one of the following items listed below, it is considered as a part of the mathematical block. This part is needed to run OCR.

1. Greek alphabet
2. Latin alphabet
3. Mathematical standard functions such as: sin, cos, tan, csc, sec, cot, sinh, cosh, tanh, log, ln, det, dim, lim, mod, gcd, lcm

5.5.7 Recursive Symbol Segmentation and Bracket Rule

In many cases, mathematical expressions are multidimensional, hence symbol segmentation or primitive component extraction must be accomplished both vertically and horizontally. "Impervious Component" is a term used in this research for a symbol which can neither vertically nor horizontally be extracted. Several segmentation techniques were compared and evaluated to extract symbols and address Impervious Component issue. Figure 5.20 shows that the Impervious Component Extraction (ICE) issue for some symbols like the radical symbol (Nazemi & Murray, 2013) can be solved using Voronoi. Additionally, the Voronoi result provides bounding boxes and geometric properties for all mathematical components, which are sufficient to symbolize role recognition and semantic structure analysis.

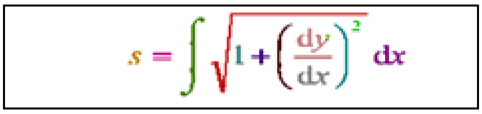


Figure 5.20: Voronoi result for sample of EF

Run Length Smearing Algorithm (RLSA) (Wong et al., 1982) is another segmentation algorithm which was tried within this module. The RLSA works on binary images where white pixels are represented by 0's and black pixels by 1's. The algorithm transforms a binary sequence x into y. The 0's in x are changed to 1's in y, if the number of adjacent 0's is less than or equal to a predefined threshold and the 1's in x are unchanged in y.

Except in cases where mathematical expressions contain impervious components, RLSA provides better results when compared to Voronoi. To improve primitive component extraction accuracy, Recursive Component Extraction (RCE) was developed which contains two sub-modules: Vertical Component Extraction (VCE) and Horizontal Component Extraction (HCE) (Nazemi & Murray, 2013). To reduce ambiguity, each VCE output first must be surrounded by brackets ( ), then the HCE module recursively applied until all components are extracted (Nazemi & Murray, 2013). The brackets Rule or, considering brackets ( ) around each VCE segment, is designed to investigate order, precedence and associativity of operators and address symbol relation issues and ambiguity. The code for recursive component extraction is available in Appendix B.

For example, without the bracket rule, some expressions such as $\frac{a+b}{c}$ and $a + \frac{b}{c}$ are not recognisable. Using the brackets rule changes them to ($\frac{a+b}{c}$) and (a)+($\frac{b}{c}$) to differentiate. The bracket rule covers the following concepts to find order of operators in expressions:

1. Operator range defines legal spatial locations for arguments of an operator (e.g. for '+', or fractions).

2. Operator dominance (Chang, 1970), defines a partial ordering on the application of operators predicates. An operator which nests completely within the range of another operator/relation is called dominated. For example, the + in $\frac{a+b}{c}$ is dominated by the fraction bar. Dominating operators are applied after the operators they dominate. One operator dominates another if and only if the latter is in the range of the former and the converse is false.

3. Operator associativity is employed when two or more of the same operator appear in each others range. For example, addition is normally left-associative: a+ b +c=a+ (b +c).

4. Operator precedence is applied to different operators when they are within each others range. For example:

   a + b ×c is equal to a+(b × c) by ordering the operators '+' and '×'

a + $\frac{b}{c}$ is equal to a +($\frac{b}{c}$) by ordering '+ ' and '/'

Figure 5.21 shows RCE results for a 2D mathematical expression.[8] The first row illustrates the result of applying first round VCE. Vertical red lines represent VCE separators. Second line shows a part of VCE result which is recursively applied to HCE. Horizontal blue lines show HCE separators. Third and forth lines are the result for first round HCE which are recursively applied to VCE. Vertical green lines illustrate second round VCE separators.

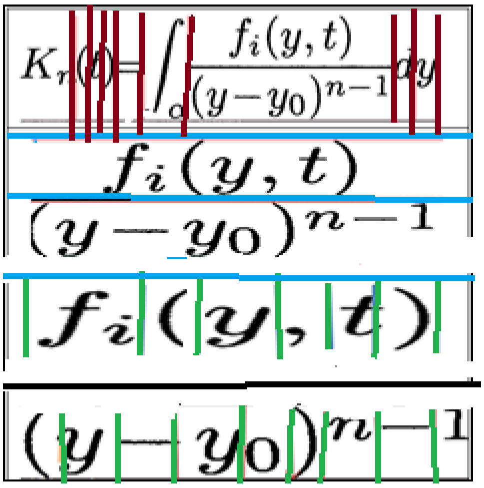


Figure 5.21: VCE and HCE result for a sample

If the extracted mathematical blocks from the previous module are not multidimensional, it can be converted into a series of isolated symbols by finding disconnected pixels inside the mathematical block. This symbol segmentation method is described step by step with relevant codes given below. To improve segmentation performance, a distance morphological function can be used to create a larger space between symbols. This module accepts three parameters: 1) Input (mathematical expression block binary image file name); 2) output; and 3) distance morphological function value. The following snippet is used to find non-transparent pixel in input binary image.

```
in=$1
out=$2
n=$3
i=1
convert "$1" -trim  mpc:-|convert mpc -morphology dilate :$n
   diamond expression.mpc
height=`identify  -format "%h" expression.mpc`
while true; do
```

[8]Note: The quality of image is poor(blur) as image were extracted from real Mathematical PDF documents

```
segment_position=`convert expression.mpc txt:- |grep 000000|
   sed '1d; s/:.*//;s/,/ /g'|sort -b -k1n,1|awk 'NR==1'`
```

Next step after finding non-transparent pixel is to extract segment at this coordinate value. Each output image contains at least one unconnected black segment on a transparent background. The following snippet extracts box of a segment from image by finding bounding box of segment and using "crop"

```
[ -z "$segment_position"  ] && break
convert expression.mpc   -fill none -draw "matte
   $segment_position floodfill"   expression.mpc
width=`identify  -format "%w" expression.mpc`
x1=$(convert expression.mpc txt:-|grep none| sed '1d; s/:.*//;s
   /,/ /g'|sort -b -k1n,1|awk 'NR==1{print $1}')
x2=$(convert expression.mpc txt:-|grep none| sed '1d; s/:.*//;s
   /,/ /g'|sort -b -k1n,1|awk 'END{print $1}')
x=$[$x2-$x1]
xx=$[$width-$x2]`
convert expression.mpc -shave 0x0 -repage $width"x"$height+0+0
   png:-|convert png:- -crop  $x"x0+"$x1"+0" mpc:-|convert mpc:-
   -fill none -opaque black  -flatten -trim $i-$out
```

Finally, this method removes the extracted segment from image. This task is repeated until all non-transparent pixels are processed.

```
convert expression.mpc -shave 0x0 -repage $width"x"$height+0+0
   png:-|convert png:- -crop  $xx"x0+"$x2"+0" temp.mpc
convert temp.mpc -flatten -trim +repage expression.mpc`
i=$[$i+1]
done
```

The line segmentation module and this symbol segmentation module were applied to several handwritten mathematical documents and the results were accurate. Some handwritten mathematical documents which were applied for segmentation have been presented in section 7.10

5.5.8 Generating a Symbol Dictionary Using InftyMDB-1

The InftyMDB-1 database was collated for the Infty project. InftyMDB-1 contains approximately 4400 mathematical formulae which were collected from 32 pure mathematical articles. Each mathematical formula consists of 10 or more symbols (Aly et al., 2009). As a part of this research, RCE was applied for 1000 mathematical expressions from InftyMDB-1 to segment them into primitive symbols. These symbols were collected to provide a mathematical symbol dictionary. This dictionary is utilized for symbol recognition and was divided into three categories considering the symbols aspect ratio (Height/Width). Any symbol can belong to one or two, but never three, categories (Malon et al., 2006).

Aspect-Ratio=Height/Width

Aspect-Ratio>1.3 ==> symbol $\in$ Tall

0.58< Aspect-Ratio<1.7 ==> symbol $\in$ Square

Aspect-Ratio<0.76 ==>symbol $\in$ Short

Table 5.2 shows three categories for 258 different symbols considering their aspect ratio. This categorisation results in 43, 18 and 206 in Tall, Short and Square respectively. Vector-based Symbol Recognition collects the Aspect Ratio, Area, Density and x/y deviation for all segmented symbol. The following formulae are used to extract all features for a symbol.

```
Area (i) =WxH
Density (i) =Number of black pixels/Area (i)
Aspect-Ratio (i)= H/W
```

$$x - deviation = \sigma_x = \sqrt{\frac{\sum_{i=1}^{n} (x_i - mean(x))^2}{n}}$$

$$y - deviation = \sigma_y = \sqrt{\frac{\sum_{i=1}^{n} (y_i - mean(y))^2}{n}}$$

The features of a symbol are saved as vector of features *Vector (i)* = [$\sigma_{x,}, \sigma_y$ *, ar, density*] in trained database (Keshari & Watt, 2007)

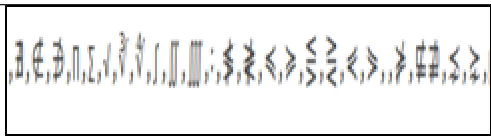
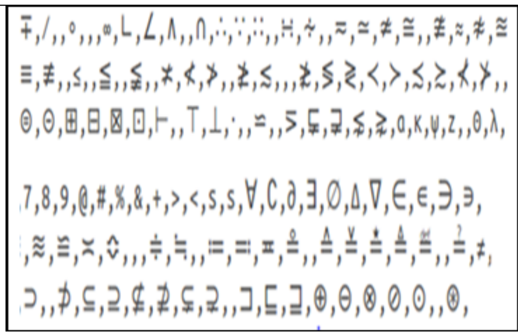
Online symbol recognition for a new symbol is responsible for:

1. specifying the category of candidate symbol;
2. calculating the Euclidean distance between candidate symbol and all symbols in its category trained data file; and
3. comparing distances and find smallest one as the result for candidate symbol recognition.

The following formula specifies Euclidean distance between candidate symbol (i) and a model symbol in trained data file.

$$\sqrt{\begin{array}{c} ((\sigma_x(i) - \sigma_{x,}(model))^2 + (\sigma_y(i) - \sigma_y(model)^2 + \\ (ar(i) - ar(model))^2 + \\ (density(i) - density(model))^2 \end{array}}$$

Table 5.2: Mathematical Symbol Categories Based on symbol Aspect Ratio

Aspect-Ratio /Category	Samples
Aspect-Ratio>1.3/Tall	
0.58<Aspect-Ratio<1.7 /Square	
Aspect-Ratio< 0.78 /Short	

5.5.9 Mathematical Symbol Recognition Using K-Nearest Neighbours (kNN) Based on Binary Vector

All extracted single components obtained by applying RCE for 1000 mathematical expressions in InftyMDB-1 were applied to the Binary Vector Generator which is responsible for:

- Removing possible margin around the single component;
- Calculating aspect ratio of the single component;
- Resizing the single component as given in Table 5.3;
- Converting the single component image which is resized, binarized and without margin to bitmap with "1" representing black and "0" representing white as the following command indicates;

  ```
  cat resized-binary-without-margin-symbol.txt |sed '1d;s/:.*#/ /g;s/,/ /
  g'| sed 's/white/0/g' | sed 's/black/1/g' | awk '{print $1, $2, $4}' >
  bitmap.txt
  ```

- Convert bitmap text file to \$ROW*\$COL matrix. The following snippet is used for this purpose Assuming \$COL*\$ROW is the symbol predefined size; and

  ```
  for((k=0; k<$ROW; k++)); do
  pixel=($(cat bitmap.txt | awk '$2 == '$k' ' | awk '{print $3}'))
  row=${pixel[@]}
  ```

```
echo $row >>TranslatedBitmapMatrix.txt
done
```

- Transfer the matrix to a single vector containing only 0 and.1. Using the following command, the matrix is transferred to a single vector and saved in the file refernces.csv as trained data. For example, if the symbol is 'Tall' and its matrix dimension is 10x16, then the vector of the symbol is a string of 160 zeroes and ones. All 0s and 1s are carrying equal weight.

```
cat TranslatedBitmapMatrix.txt|xargs|sed 's/ /,/g'>>references.csv
```

Table 5.3: Predefined size for three symbol categories

Category	Size
Tall	10x16
Square	16x16
Short	16x10

The search for Nearest Neighbours is a common task performed in machine learning. Previous studies shows algorithms based on kNN have high accuracy but are slow to execute (Zhang et al., 2009). In this study KD-Tree search method was used with the help of "allknn" function of "mlpack" package. The "allknn" calculates all k -Nearest Neighbours of a set of points using KD-Trees or cover trees. Cover tree method is experimental but not necessarily fast. KD-Tree are data structures which are used to store points in k-dimensional space. The leaves of the tree store one or more points of the data set. Each point is stored in one and only one leaf and each leaf stores at least one point. The nodes of the tree splits the space. Each split divides space and data into two distinct parts. Subsequent splits from the root node to one of the leaves remove parts of the data set (and space) until only a single part of the data set (and space) is left. KD-Tree search method allow to perform searches like "all points at distance lower than R from X" or "k nearest neighbours of X" efficiently (Narendra & Fukunaga, 1977).

In the following command, vector-query.csv, referance.csv, neighbors.csv and distanve.csv represent candidate symbol, trained data, 10 nearest neighbours of candidate symbol in trained data and their distance to candidate symbol respectively.

```
allknn -q vector-query.csv -r references.csv -n neighbors.csv -d distance.
csv -k 10 -v
```

As stated before, some mathematical symbols for building formulae have different dimensions in different situations but keep same meaning. These symbols cause some problems in recognising mathematical symbols. Consideration of the unique predefined size for symbol in binary vector method leads to mathematical symbol recognition regardless of its dimensions.

5.5.10 Symbol Layout or Structural Semantic Analysis

Issues regarding mathematical layout analysis include:

1. Two-dimensional structure;

2. No dictionary of all mathematical expressions;

3. Large variations in symbol scale;

4. Non-ordering symbol arrangements; and

5. Many more symbol classes.

The first issue has already been addressed by the previously described approach (RCE) and using the brackets rules. The generated mathematical dictionary, as described in the previous section, covers the issue regarding the lack of available complete and comprehensive dictionaries. Mathematical symbol conversion to binary vector covers the third issue. Last two issues are addressed by using semantic structure analysis and investigating relationships between two adjacent symbols. Mathematical expressions generally represent an application of functions, operators and relations to arguments. Multiple mathematical statements may be represented by a single expression. In other words, mathematical expressions are polysemic (Zanibbi & Blostein, 2012). It means that the definition and role of symbols frequently change. Even when the domain is clear, symbol definitions are often ambiguous. Analysis of the spatial relationships between symbols is the symbol layout or structural semantic analysis. The spatial relationships are baseline, subscript, superscript, upper and lower relations. Since spatial relationships convey the meaning of mathematical expressions, its identification is critical. Symbol semantic analysis is considered for solving the ordering and relationship problems, noting the symbol role and situation concentration instead of symbol isolation. Semantic analysis detects upper, lower, subscripts and superscripts components, assigning them to their parent symbols and merging them as a single unit (Aly et al., 2009). Identifying relationships between each symbol and its previous adjacent symbol has been implemented by measuring the line slope between bounding box corners (Aly et al., 2008) as it is shown in Figure 5.22.

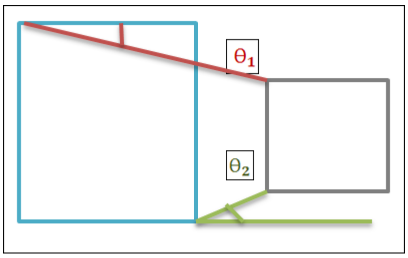


Figure 5.22: Slopes between two adjacent symbols

These slopes are captured by the following formulae: (Labahn et al., 2008)

O_2 =*Green line slope (i)* $= Ymax_i - Ymax_{i-1} / \ Xmin_i - Xmax_{i-1}$

O_1 =*Red line slope (i)* $= Ymin_i - Ymin_{i-1} / \ Xmin_i - Xmin_{i-1}$

Other useful features to recognise relationships between two adjacent symbols are as follows:

HR=Height-Ratio (i) $= \frac{h_i}{h_{i-1}}$

$MLR = middle.line.ratio(i) =$

$\frac{[(Ymax_i - Ymin_i)/2 + Ymin_i] - [(Ymax_{i-1} - Yminx_{i-1})/2 + Ymin_{i-1}]}{h_i} = \frac{(Ymax_i + Ymin_i) - (Ymax_{i-1} + Yminx_{i-1})}{2h_i}$

Figure 5.23 illustrates relationship between height of two adjacent symbols.

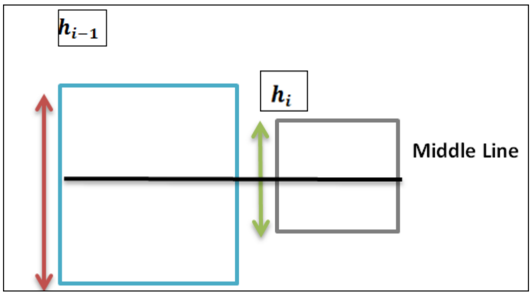


Figure 5.23: Relationship between two adjacent mathematical symbols Height-Ratio

Figure 5.24 illustrates 8 different relations between two adjacent symbols in neighbourhood. These relations are Left, Left above, Left Below, Above, Below, Right Above, Right and Right Below which can be recognised considering mentioned features as it is indicated in Table 5.4.

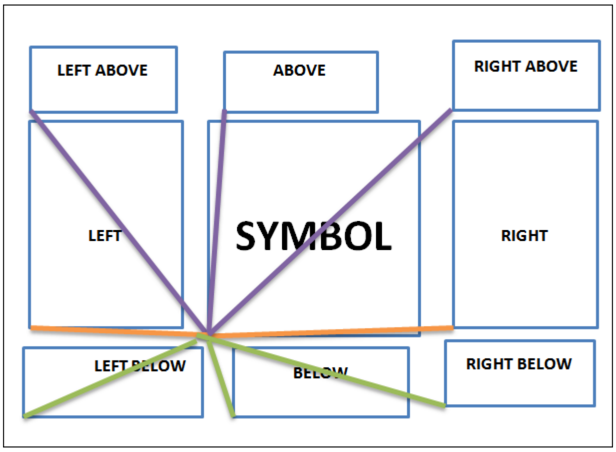
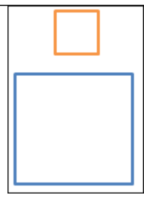
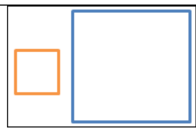
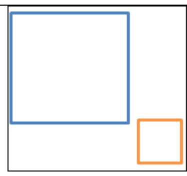
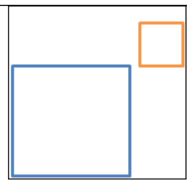
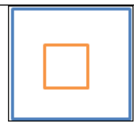


Figure 5.24: 8-different relationships between a symbol and its neighbours

Table 5.4 indicates different relationships between two adjacent symbols based on some extracted feature such as O_1,O_2, HR and MLR.

Table 5.4: Different relations between two adjacent symbols in mathematical expressions

Formula	Relation	Features	Example	Position
$\frac{cc_i}{cc_{i-1}}$	Below	$O_2>1$ $O_1<$-1	$\frac{a}{b}$	
$cc_{i-1}cc_i$	Left	HR=1 MLR=0 $0<O_1<1$ $O_2<$-1	M=	
$cc_{i-1_{cc_i}}$	Above-Left	0<MLR<1 HR<1 $0<O_1<1$ $O_2>1$	D_i	
$cc_{i-1}{}^{cc_i}$	Below-Left	$-1<O_1<0$ $O_2<1$	x^2	
$\sqrt{cc_i}$ when $cc_{i-1}$is $\sqrt{}$	Inside	0<MLR<1 HR<1 $-1<O_1<0$ $O_2<1$	$\sqrt{1+x^2}$	

5.5.11 Merging Primitive Components and MathML Generation

In order to comply with symbol layout analysis, symbols are inspected from left to right (Yonghua et al., 2007). Symbol layout analysis and symbol classification are based on extracted features. One of the most critical tasks in this module is detecting dependent components and merging them as a single unit using brackets to surround them.

Dependent components may be:

- Operands of an individual operator;
- Parameters of a function;
- Subscript and superscript of a symbol; and
- Upper and lower part of a symbol.

Mathematical symbols are divided into 13 classes, this classification supports symbol layout analysis:

1. simple/ordinary A, x
2. prefix operator$\sum \prod \int$
3. binary operator (conjunction) $+$-
4. relation/comparison (verb) $= <$?
5. Open left/opening delimiter ([{
6. Close right/closing delimiter)] }
7. postfix/punctuation . , ; ! and Ellipses $a_0, a_1, \ldots, a_n$
8. SubsNon-ordering symbol arrangements;cripts and superscripts
9. Accents
10. Binomial
11. Matrices
12. Roots
13. Text or Named operator. These operators are represented by a multiple letters abbreviations such as: arccos, arcsin, arctan, arg, cos, cosh, cot, coth, csc, deg, det, dim, exp, gcd, hom, inf, inj, lim, ker, lg, ln, log, max, min, Pr, sec, sin, sinh, sup, tan and tanh. These parts of mathematical expressions can be recognised using OCR.

Symbols of class 3, notably the minus sign, are automatically coerced to class 1, if they do not have a suitable left operand.

$cc_{i-1}cc_icc_{i+1}$ Assuming cc_i is in the prefix class, ifcc_i is located below or left below of cc_{i+1} and above or above left of cc_{i-1},then it can be the parent for cc_{i-1}, cc_{i+1} and so $cc_{i-1}cc_icc_{i+1}$ can be placed in the same bracket.

For operands finding of an operator this algorithm must be run:

- Parse from left to right until meeting class 6 in position A;
- Parse right to left from position A until reaching class 5 in position B;

- Extract the most inner layer between B and A;
- Parse from B to A until getting one of the existing symbols in class 3 in position C;
- Parse back between C and B to get the first operand ; and
- Parse forward between C +LENGTH (operator symbol) and A to get the second operand.

As series of single primitive components for two different expressions:$\frac{x^2+z}{y}$ and $\frac{x^{2+z}}{y}$are same, they seem similar before symbol semantic layout analysis and applying merging rules. The following analysis proves that they are significantly different.

Example 1:$\frac{x^2+z}{y}$

1. VCE and bracket rule convert formula to $(\frac{x^2+z}{y})$. Brackets which are located around only one component are removed and it is converted to :$\frac{x^2+z}{y}$
2. HCE converts$\frac{x^2+z}{y}$ to:
   (a) $x^2 + z$
   (b) fraction bar
   (c) y
3. VCE and bracket rule convert
   (a) $x^2 + z$ to $(x)(^2)(+)(z)$. Brackets which are located around only one component are removed and it is converted to $x,\ ^2, +, z$
   (b) fraction bar
   (c) y
4. Single Component segmentation or RCE is completed now and single components are : $x\ ,^2\ , +, z,\ fraction\, bar,\ y$
5. Parse from left to right until reach class 2, 3, 4 or 5
6. Reach $+$ in class 3
7. Parse back looking for first operand for $+$
8. Reach 2 which is not at same level of x (considering Middle Line Ratio)
9. Parse back
10. Reach x at the same level of $+$, Calculation $\vartheta 1$, $\vartheta 2$ shows x is located in the below left of 2. It means 2 is superscript of x

11. x is parent of 2, so x and 2 should be put in bracket and treated as an unit (x^2)
12. There is nothing before(x^2)
13. (x superscript 2) is tagged for being first operand for +
14. Parse forward after +
15. Reach z at the same level of (x^2) and +
16. Parse forward
17. Reach fraction which is not at same level of +
18. Parse back and tagged z as second operand for +
19. Consider(x^2) +z as an unit and put them in bracket ((x^2)+z)
20. Parse forward after z
21. Reach fraction bar
22. Parse back looking for first operand for fraction bar
23. Reach unit ((x^2) +z) and tag it a as numerator
24. Parse forward after fraction bar
25. Reach y
26. There is nothing after y so y is the denominator.
27. The final description is:
    (a) Numerator ((x^2)+z)
    (b) Fraction bar
    (c) Denominator y

Example 2:$\frac{x^{2+z}}{y}$

1. VCE and bracket rule convert formula to ($\frac{x^{2+z}}{y}$).Brackets which are located around only one component are removed.It is converted to: $\frac{x^{2+z}}{y}$
2. HCE convert it to:
    (a) x^{2+Z}
    (b) Fraction bar
    (c) y

3. VCE and bracket rule convert formula to x, 2,+, z, $fraction\, bar$, y
4. All single component are extracted
5. Parse from left to right until reach class 2, 3, 4, 5
6. Reach + in class3
7. Parse back looking for first operand for +
8. Reach 2 which is at same level of +
9. Parse back reach x
10. x is not at the same as 2
11. 2 is tagged as first operand for +
12. Parse forward after +
13. Reach z at the 2 same level
14. Parse forward
15. Reach $fraction\, bar$, which is not at the same level as z
16. z is second operand for +
17. 2+z tagged as unit and put in brackets (2+z)
18. Calculate O_1, O_2 shows x is located in the below left of (2+z).It means (2+Z) is superscript of x
19. x is parent of (2 +z) and x,(2+z) should be put in bracket and treated as an unit $(x^{(2+Z)})$
20. Parse forward after unit $(x^{(2+Z)})$
21. Reach fraction bar
22. Parse back looking for first operand for fraction bar
23. Reach unit $(x^{(2+Z)})$ and there is nothing behind it so tag it a as numerator
24. Parse forward after fraction bar
25. Reach y
26. There is nothing after y so y is the denominator.
27. The final description is:
    (a) Numerator $(x^{(2+Z)})$
    (b) Fraction bar
    (c) Denominator y

5.5.12 Rendering Math-TEX Family with MATHSPEAK

Rendering of the expression is performed to enable navigation points to access mathematical formulae. Representing mathematical formulae with navigation ability provides the vision-impaired opportunity to rehear parts of the expression to understand it. Currently this module requires input to be in Amsmath format, it can be reconfigured to deal with other mark-up languages in the TEX family.

The rendering module can read mathematical expression input as text files. At this stage, the system recognises all alphanumeric characters, almost all Greek letters and other frequently used symbols in mathematical formulae. It can analyse formulae including fractions, square roots, subscripts, superscripts, integrals, limits, summation and matrices. Some examples are available on Section 7.8 to show how mathematical expression rendering works.

The ability to navigate through different sections of a mathematical formula is one of the most significant results of the rendering system. To aid in the navigation of sections and blocks of mathematical expressions, sections and subsections of the equation must be marked as separate components of the overall expression. Sections may be separated from each other by the '\' character, which is recognisable as a keyword separator, for example: \frac, \sum, \prod, \log, \int, \log, \sin. Each subsection contains all of the elements which are surrounded by open '{' and closed '}' braces.

Two arrow keys ("$\Leftarrow$" and "$\Rightarrow$") on the keyboard were implemented as navigation commands. During the playing of an audio format formula (which is generated by the Text to Speech (TTS) function), the user can change the play position to the previous subsection in the formula by pressing the left arrow $\Leftarrow$" key. If there is no subsection before the current position, play position is changed to the previous main section. Similarly, by pressing the right arrow "$\Rightarrow$", the user is able to set the play position to the next subsection, or in the absence of one, to the following main section.

Traditionally, an experienced mathematics reader might view a complex formula at high-level and then study its various details or components. The rendering module allows a listener to access such a high-level view. This option works based on counting the number of occurrences of each keyword, distinguished by the '\' character. In some cases of mathematical formula, recognition of written notation can be confusing and hard to understand. The absence of brackets can cause this problem, so before sending Amsmath linear text to TTS, further formatting conversions are necessary to make the audio representation closer to the equivalent of the visual representation. Additional formatting fixes some Amsmath conflicts and adds some useful settings, symbols and environments to the Amsmath format construction.

Distinguishing among several different meanings for a symbol in code is the fundamental task of the preparation stage. For example, in Amsmath form, the character underscore '_' is usually used for presenting 'index' meaning, however if this character appears in the formula containing '\sum', '\prod' and '\int' it means 'from', with '\log' it indicates 'to the base of...', and finally with '\lim' it means 'as approaches '. Also the '^' character, which generally means 'to the power of', after '\sum', '\prod', and '\int' means 'to'.

In Amsmath format, a fraction has two elements which are separated from each other by enveloping each element in braces '{ }' and this is the key to solve the ambiguity in some formulae that contain fractions. However, sometimes this causes perturbation in audio output. To reduce ambiguity and perturbation, if a numerator or a denominator of a fraction includes just one element, brackets around this element must be eliminated after the formula is produced by Amsmath.

Another mathematical construction which requires preparation is the matrix. In the Amsmath form of the matrix '\\' denotes 'next row' and the character '&' denotes 'next element'. In some matrices, where all values are displayed for at least one row and at least one column, the number of '\\' characters determines the number of rows and the number of '&' characters in a row determines the number of columns. The size of the matrix is the number of rows times the number of columns. Therefore, at the beginning of matrix reading, it is useful and straight forward to give a construction brief overview to the user by presenting the number of rows and the number of columns. If the matrix is presented in an abbreviated form and contains dots to denote absent elements, extracting the dimensions of the matrix is done by accessing the two subscript indices of the last element of the matrix which lies at the bottom right corner. Navigation through a matrix is done in three ways: 1) by specific row; 2) by specific column; and 3) by specific element.

Distinguishing between mathematical and statistical symbols is another part of the preparation stage, for example the '|' character in statistics indicates 'conditional probability' however, an expression enveloped in the same '|' character in mathematical area, means 'the absolute value of the expression' (Nazemi et al., 2012). The code to convert Amsmath into alternative text descriptive is available in Appendix B.

The following 5 different mathematical expressions were represented with their Amsmath forms and MATHSPEAK Alternative Text Description. If these expressions extracted from images by OCR and screen reader has only access to plain text, in the result of reading in liner mode, it generates the one output for all expressions, however results of accessing and rendering Amsmath forms are totally different.

Elements in equation 5.1 have subscript and superscript. In equation 5.2 elements have superscripts and their superscripts have superscripts. Elements in equation 5.3 expression have superscripts and their superscripts have subscripts. In equation 5.4 expression elements have subscripts and their subscripts have superscripts and in equation 5.5 expression elements have subscripts and their subscripts have subscripts.

$$x_1^k + x_2^k + x_3^k + \cdots + x_n^k = 0 \tag{5.1}$$

```
amsmath: \begin{equation}x_1^k+x_2^k+x_3^k+\cdots+x_n^k = 0\end{
   equation}
MATHSPEAK Alternative Text Description:
<x index 1 power k/>< plus/>
<x index 2 power k/><plus />
<x index 3 power k /><plus/>
<continue  until />
<x index n  power k />
<equal 0 />
```

$$x^{k^1} + x^{k^2} + x^{k^3} + \cdots + x^{k^n} = 0 \tag{5.2}$$

```
amsmath: \begin{equation}x^{k^1}+x^{k^2}+x^{k^3}+\cdots+x^{k^n
   }=0\end{equation}
MATHSPEAK Alternative Text Description:
<x power k power 1/><plus/>
< x power k power 2/><plus/>
< x power k power 3/>< plus/>
 <continue until  />
<x power k power n  />
<equal   0 />
```

$$x^{k_1} + x^{k_2} + x^{k_3} + \cdots + x^{k_n} = 0 \tag{5.3}$$

```
amsmath: \begin{equation}x^{k_1} + x^{k_2} + x^{k_3} + \cdots + x
   ^{k_n} = 0 \end{equation}
MATHSPEAK Alternative Text Description:
<x power  open bracket k index 1  close bracket />< plus/>
<x power  open bracket k index 2  close bracket />< plus/>
<x power  open bracket k index 3  close bracket /><plus />
<continue until />
<x power  open bracket k index n  close bracket />
<equal   0 />
```

$$x_{k^1} + x_{k^2} + x_{k^3} + \cdots + x_{k^n} = 0 \tag{5.4}$$

```
amsmath: \begin{equation}x_{k^1}+x_{k^2}+x_{k^3}+\cdots+x_{k^n}=0
    \end{equation}
MATHSPEAK Alternative Text Description:
<x index k power 1/>< plus/>
<x index k power 2/>< plus/>
<x index k power 3/>< plus/>
<continue until />
```

```
<x index k power n/>
<equal  0 />
```

$$x_{k_1} + x_{k_2} + x_{k_3} + \cdots + x_{k_n} = 0 \tag{5.5}$$

```
amsmath: \begin{equation}x_{k_1}+x_{k_2}+x_{k_3}+\cdots+x_{k_n}=0
    \end{equation}
MATHSPEAK Alternative Text Description:
<x index  open bracket k index 1  close bracket/><plus/>
<x index  open bracket k index 2  close bracket/><plus/>
<x index  open bracket k index 3 close bracket/><plus/>
<continue until />
<x index  open bracket k index n  close bracket/>
<equal  0 />
```

Further samples are available in section: MATHSPEAK, Chapter: Testing (7.8).

5.6 Mathematical Graph Accessibility with MathGraphReader

The mathematical document accessibility issues of vision-impaired students are associated with representation of mathematical expressions and mathematical function graphs. Mathematical function graphs are visual presentation for mathematical information and are useful to illustrate numerical or qualitative information, which are sometimes difficult or even impossible to describe. A graph is a picture that shows how sets of data are related to each other. Mathematical function graphs generally convey the intended message of a mathematical document. Therefore, access to graphs is essential in the mathematics learning process. Finding a solution to represent mathematical function graphs in accessible format is necessary for the vision-impaired.

This section describes an approach to develop an application to address this issue by detecting, extracting and categorising important information from mathematical function graphs using two open source packages ImageMagick and GNUPLOT (Williams & Kelly, 2004).

Accessibility of mathematical function graphs is one of the areas which needs to be facilitated. Currently, Haptic feedback may be used for guidance and assistance in graph navigation (Yu et al.,2001). However, it is not efficient to present exact data values to the user. In addition, it is expensive and may take users some time to familiarize themselves with the new interface. The limitations of the Haptic feedback devices hinder users ability to explore graphs (Yu & Brewster, 2002).

5.6.1 Current Method

Traditionally, there are two methods to represent visual components such as graphs to vision-impaired, namely, tactile and audio methods. In the tactile method graphs and diagrams are presented in Braille or raised dots and lines on the swell-paper which has raised surfaces for a vision-impaired person to feel (Yu et al.,2002). They are used to convey non-textual information such as maps, paintings, graphs and diagrams. The general shape of the graph can be understood by touching it carefully, but hardware is needed to generate tactile charts (Goncu, 2008). Several problems are associated with Braille and tactile techniques. Tactile graphs are not modifiable and durable. Translating information into an accessible graphics form is high cost.

Presenting mathematical graphs audibly is more complicated. Passively hearing the graphed mathematical formulae alone would not be sufficient to transfer and describe complex mathematical constructs. Sonification is a type of audio method which conveying data via sound, pitch, and 2-dimensional acoustics (Brown & Brewster, 2003). Since acoustics are volatile, information can easily be misheard and therefore misunderstood.

5.6.2 Proposed Method

The proposed method in this research provide opportunity for vision-impaired, besides passively listen to extracted classified data from graphs using TTS, actively interact with them by navigating through classified information and digitized data table without requiring another device or peripheral. The framework of mathematical function graph contains two crossed axes which meet at the origin point. The axes divide the plane into four quadrants. Scaled values are increased from left to right on horizontal axis and from down to up on the vertical axis. Data table has two columns to show dependent variable value for each independent variable value. Generally data tables are converted to graphs for better understanding of the relationship between vertical and horizontal values (visualisation). Since vision-impaired students are not able to use visual components such as graphs, the graph must be converted into data table with digitization. Digitization has two responsibilities:

1. Finding the position of pixels of the main graph in the image; and
2. Running mathematical functions to assign a coordinate values to the pixels values.

These mathematical functions are based on the extracted scaling values, size and origin point of graph image. Although the basic concept of converting the image pixel values to scaled values is straight forward, practical considerations such as specifying an accurate vertical and horizontal ratio must be addressed. The developed application by this research performs digitization and extracts all prominent information from the graph using mathematical concepts followed by graph image plotting with GNUPLOT and image processing with ImageMagick. Axes labels, scaled values and the range of each axis, where the function increases or decreases, where the maxima, the minima, the flexes and the points of discontinuity must be collected to provide an alternative text description for mathematical function graph.

This text containing graph description is sent to TTS to represent to user in audio format. This application assumed that mathematical function relevant to the graph is existing and accessible. Regardless of parameter values the shape of the graph depends on the form of the function. Hence, in most cases general shape of a graph can be guessed by a form of a mathematical function.

5.6.2.1 Examined Graphs

The developed Linux command Bash Scrip application was named MathGraphReader. It provides complete text description for linear, quadratic, power, polynomial, exponential, logarithmic, and sinusoidal graphs to be replaced with the graph using the following properties. The application accepts the mathematical function as the input parameter and starts to find some information such as function type or graph shape utilizing text processing of input parameter and the following properties.

1. Linear: $y\ = a\ x + b$
   (a) Graphs of these functions are straight lines.
   (b) a is the slope and b is the y- intercept.
   (c) If a>0 then the line rises.
   (d) If a<0 then the line falls.
2. Quadratic: y= a x^2 $+bx+c$
   (a) The graph is called parabola.
   (b) Graph crosses y-axis at c. The y-intercept point is c.
   (c) If a>0 then the parabola opens upward , the graph will be 'U' shaped.
   (d) If a<0 the parabola opens downward, the graph will be 'n' shaped.
3. Power: $y = ax^b$
   (a) If b>0 then:
   When x = 0 graph meets origin point.
   When x is large and positive they are all large and positive.
   When x is large and negative and b is, even: y is large and positive .
   When x is large and negative and b is odd: y is large and negative.

(b) If b<0 then:

When $x = 0$ these functions suffer a division by zero and therefore are all infinite.

When x is large and positive they are small and positive.

When x is large and negative and b is even: y is small and positive.

When x is large and negative and b is odd: y is small and negative

(c) If b is a fraction between 0 and 1 then:

When $x =$ zero these functions are all zero.

The curves are vertical at the origin and as x increases they increase but curve toward the x axis.

4. Polynomial:$y = a_n .x^n + a_{n-1} .x^{n-1} + ... + a_2 .x^2 + a_1 .x + a_0$,

   (a) The highest power of x is the degree of the polynomial.

   (b) Polynomials of degree greater than 2 can have more than one maximum or minimum value.

   (c) The largest possible number of minimum or maximum points is one less than the degree of the polynomial. The 'n-1 ' gives the maximum number of '*ups and downs*' that the polynomial can have.

5. Exponential: $y = a\ b^x$

   (a) If the base b >1 then the result is exponential growth.

   (b) If the base b<1 then the result is exponential decay.

6. Logarithmic: $y = alog(x) + b$

   (a) For small *x, y is* negative.

   (b) For large *x, y is* positive but stay small.

7. Sinusoidal: $y = asin(bx + c)$

   (a) Graph of Sinusoidal functions has a wave shape with respect to position or time.

   (b) Parameter 'a' is the amplitude which affects the height of the wave.

   (c) Parameter 'b' is the angular velocity, which affects the width of the wave.

   (d) Parameter 'c' is the phase angle that shifts the wave left or right.

5.6.2.2 General Concepts

In general terms, the following conditions are examined for all of the above mentioned graphs:

- If for a specific interval $x_2 > x_1$ and $y_2 > y_1$or $y' > 0$ then in this interval graph is increasing
- If for a specific interval $x_2 > x_1$ and $y_2 > y_1$ or y' <0, then in this interval graph is decreasing
- If for a specific interval $y' = 0$, then in this interval the graph is constant
- If f(x)=-(-x), then the graph is symmetric with respect to y axis
- If f(x)=-f(-x),then the graph is symmetric with respect to the origin.

The following factors are included in the alternative text description of graph:

- Relative maximum: The point(s) on the graph which have maximum y values relative to its neighbourhood on the graph
- Relative minimum: The point(s) on the graph which have minimum y values relative to its neighbourhood on the graph
- Absolute maximum: The point on the graph which has the largest value of y
- Absolute minimum: The point on the graph which has the smallest value of y

Minimum and maximum points are obtained by assigning zero to y' and finding x at this point

If y">0, then the point is maximum, then graph is convex.

If y"<0, then the point is minimum, then graph is concave.

If y"=0,then the point can be both or neither.

5.6.2.3 Data Extraction

Intercept values (meeting points of the graph with axes), inflection point, starting/ending points and the minima/maxima points are calculated and included in the text description. Finding these points in most cases is a complicated process, which involves a large amount of mathematical problem solving. This research to find critical points of the graphs and embed them in text description use reverse engineering technique as follows:

The image of the graph is plotted using GNUPLOT (open source package) and associated graph function with predefined dimensions and specified range for axes. GNUPLOT plots a graph using configuration file (graph.gnu) and relevant mathematical function. The below snippet creates a configuration file for GNUPLOT (graph.gnu) and $in is a text string which indicates relevant mathematical function for graph.

```
#graph.gnu configuration file :
echo "set term png" >graph.gnu
echo set output "output.png">>graph.gnu
echo set xrang restore>>graph.gnu
echo set yrange restore>>graph.gnu
echo show xrange>>graph.gnu
echo show yrange>>graph.gnu
echo plot $in with lines lc rgb green notitle>>graph.gnu
```

This command plots graph, saves it in the file output.png using graph.gnu as configuration file and keeps plotting information in range.txt. According to the last line of graph.gnu, the plotted graph by GNUPLOT is green in colour.

```
gnuplot graph.gnu>&range.txt
```

Dimensions of output.png are extracted by:

```
Height=$(identify -format %h output.png)
Width=$(identify -format %w output.png)
```

Several image processing techniques such as threshold, negate, trim and crop were used for black boarder and white margin removal.

Axes/Graph segmentation means conversion of the original image into three segments x-Axis,y-Axis and graph-only images using morphology and crop technique as the following snippet indicates three steps:

Step 1: obtaining x-Axis position as a horizontal line with minimum height:

```
convert output.png -threshold 70% -negate  -trim -negate -trim -
   morphology dilate rectangle: 10x1 xAxis.txt
xAxis=$(cat xAxis.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep -Ev "#
   FFFFFF"|awk '{print $1,$2}'|sort -b -k2n,2 |awk 'END{print $2
   }')
```

Step 2: obtaining y-Axis position as a vertical line with minimum width:

```
convert output.png -threshold 70% -negate  -trim -negate -trim -
   morphology dilate rectangle :1x10 yAxis.txt
yAxis=$(cat xAxis.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep -Ev "#
   FFFFFF"|awk '{print $1,$2}'|sort -b -k1n,1 |awk 'NR==1{print
   $1}')
```

Step 3: extracting graph-only based on x-Axis and y-Axis positions:

```
X=$[$Width-$yAxis]
convert output.png  -crop $X"x"$xAxis"+"$yAxis"+"0  graph-only.
   png
```

Dimensions of graph-only are extracted by:

```
Height=$(identify -format %h graph-only.png)
Width=$(identify -format %w graph-only.png)
```

The following commands lead to obtain : Xmax, Xmin , Ymax ,Ymin, X-axis range and Y-axis range.

```
Xmin=$(cat range.txt|grep xrange|sed s/:.*//g|sed s/.*\[//g)
Xmax=$(cat range.txt|grep xrange|grep -o -P (?<=:).*(?=]))
Ymin=$(cat range.txt|grep yrange|sed s/:.*//g|sed s/.*\[//g)
Ymax=$(cat range.txt|grep yrange|grep -o -P (?<=:).*(?=]))
X-Axis-Range=$(echo $Xmax-$Xmin|bc)
Y-Axis-Range=$(echo $Ymax-$Ymin|bc)
```

X-ratio and Y-ratio are used to convert values of coordinate points from pixel scaling to real value. They are calculated as follows:

```
X-Ratio=X-Axis-Range/Width
Y-Ratio=Y-Axis-Range/Height
```

The following command removes all pixels except red ones which indicate graph-only pixels.

```
convert output.png txt:-|grep green|sed 's/:.*#/ /g;1d;s/,/ /g'|awk '{print $1,
    $2}' > pixel.txt
```

Position of origin point in the output.png is (Width/2, Height/2)

```
xOrigin=$(echo $Width/2|bc)
YOrigin=$(echo $Height/2|bc)
```

To find graph location in plane Table 5.5 is used. First column of the table indicates that mathematical graph function area is divided into four quadrants. The second column shows the conditions of coordinate values to locate in each quadrant.

Table 5.5: Graph location in plane

Location	Real Graph	GNUPLOT Graph	Command
Quadrant 1	x>0 , y>0	x>Width/2 , y>Height /2	cat pixel.txt|awk ’$1>’$xOrogin’ && $2>’$yOrigin”|wc -l
Quadrant 2	x<0 , y>0	x<Width/2 , y>Height /2	cat pixel.txt|awk ’$1<’$xOrogin’ && $2>’$yOrigin”|wc -l
Quadrant 3	x<0 , y<0	x<Width/2 , y<Height /2	cat pixel.txt|awk ’$1<’$xOrogin’ && $2<’$yOrigin”|wc -l
Quadrant 4	x>0 , y<0	x>Width/2 , y<Height /2	cat pixel.txt|awk ’$1>’$xOrogin’ && $2<’$yOrigin”|wc -l

X-intercept where the graph of an equation crosses the x-axis is usually extracted by assigning zero to y in graph equation and obtaining x.This step requires solving the equation. For example in the following equation:

$ax^2 + bx + c = 0$

The values of x is given by Quadratic Formula, then:

$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$.

As it is usually complicated, this point is extracted by finding green pixel(s) in graph image of GNUPLOT with y=Height/2 as this command show:

```
xIntercept=($(cat pixel.txt|awk '$2=='$yOrigin''
```

- Similarly, Y-Intercept where the graph of an equation crosses the y-axis is extracted by assigning zero to x in graph equation and obtaining y. This point is extracted by finding green pixel(s) in GNUPLOT graph image with x=Width/2.

```
yIntercept=($(cat pixel.txt|awk '$1=='$xOrigin''
```

- Absolute maximum is obtained by finding (X-in-Ymin,Ymin-of-Green-Pixels)

```
Maximum=$(cat pixel.txt|sort -b -k2n,2|awk 'NR==1')
```

- Absolute minimum is obtained by finding (X-in-Ymax,Ymax-of-Green-Pixels).

```
Minimum=$(cat pixel.txt|sort -b -k2n,2|awk 'END{print}')
```

In order to convert all obtained pixel scaling coordinate values to real scale values below commands are used:

```
X-Real=X-Ratio*X-in-GNUPLOT-graph
Y-Real=Y-Ratio*Y-in-GNUPLOT-graph
```

Table 5.6 presents all supported functions with GNUPLOT and this application.

Table 5.6: Supported functions with GNUPLOT and MathGraphReader application

Function	Description
abs(x)	absolute value of x
acos(x)	arc-cosine of x
atan(x)	arc-tangent of x
asin(x)	arc-sine of x
cos(x)	cosine of x, x is in radians.
cosh(x)	hyperbolic cosine of x, x is in radians
exp(x)	exponential function of x, base e
invnorm(x)	inverse normal distribution of x
log10(x)	log of x, base 10
log(x)	log of x, base e
norm(x)	normal Gaussian distribution function
rand(x)	pseudo-random number generator
sin(x)	sine of x, x is in radians
sinh(x)	hyperbolic sine of x, x is in radians
sqrt(x)	the square root of x
tan(x)	tangent of x, x is in radians
tanh(x)	hyperbolic tangent of x, x is in radians

Figure 5.25 illustrates the mathematical graph of function $y = x^4 + 3x^3 + 4$ (left) and generated graph by GNUPLOT (right).

Figure 5.25. A Mathematical graph and its similar product by GNUPLOT

Figure 5.25 (left) indicates that the minimum is occurred at (-2.25 , -4.54)

Figure 5.25 (right) is plotted by GNUPOLT using y=x**4+3*x**3+4.

The following is the data extracted by the developed application from Figure 5.25 (right):

- The degree of this graph is 4;
- It is a polynomial;
- There are 3 possible maxima and minima;
- The y-intercept is (0,4);
- The x-intercept are (-3,0),(-1,0);
- The graph is located at quadrants 1, 2 and 3; and
- Absolute minimum is located at: (-2.14815 ,-4.40191).

There are further examples of mathematical graph description finding by this method in section MathGraphReader, Chapter: Testing (7.7). The code for MathGraphReader is available in section Appendix B.

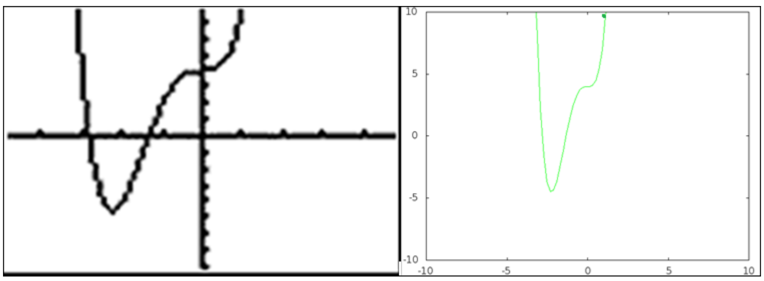


Figure 5.25: A Mathematical graph and its similar product by GNUPLOT

5.6.3 Testing and Evaluation

A survey was conducted using a questionnaire among 28 vision-impaired students. They were asked the following question:

"The following is an alternative description for a mathematical function graph. Please rate how well you believe you understand the graph by listening to this description once?"

- Horizontal axis is labelled X and its range is between -3 to 3
- Vertical axis is labelled Y and its range is between -3 to 3
- Shape of graph is parabola

- Horizontal and vertical step value is 1
- This graph is plotted from the equation: y= x^2-2x
- The graph is started from point (-1, 3) until (3,3)
- The graph is located at 1,2 and 4 quadrants
- x-intercept occurs at points (0,0) and (2,0)
- y- intercept occurs at point (0,0)
- The graph has absolute minimum at point (1,-1)

Obtained results from the participants responses indicated how they communicated with this alternative text description while students listened to text only once and did not use navigation ability to explore data table. As Table 5.7 illustrates 48.15% of participants completely understood it, 37.04% under 50% realised the graph and 14.81% not at all.

Table 5.7: Evaluation responses

percentage of participants	evaluation response
48.15	completely understand
37.04	under 50%
14.81	not at all
0	I do not understand math enough to judge

MathGraphReader, achieved graph data extraction or graph digitization to provide data table and alternative text description by plotting corresponding graph for a known mathematical function. Some image processing techniques such as negating, trimming, morphology and cropping were used for black border/white margin removal and graph/axes segmentation to obtain pure graph, exact axes ranges and origin point position. Finally, accessing x-ratio and y-ratio assists to complete data table present it to vision-impaired student.The results of applying this application for three other mathematical function graph and generate complete text description for them are available in Section 7.7

5.7 Not-in-Order Components: Tables and TableReader

Tabular data arrangement means arranging data in rows and columns. The use of tables is pervasive throughout all communication, research and data analysis. Tables appear in print media, handwritten notes, computer software, architectural ornamentation, traffic signs and many other places. Tables differ significantly in variety, structure, flexibility, notation, representation and use. In books and technical articles, tables are typically presented apart from the main text in numbered and captioned floating blocks. A table consists of rows and columns, the intersection of a row and a column is a cell. The table cells may be grouped, segmented, or arranged in many different ways, and even nested recursively (Zielinski, 2006)

To provide full access to tables, the titles of table and titles of each column or row must be presented before reading the data in cells. It should be left up to the user to decide the most logical one-dimensional way to read materials. The tables are mostly inaccessible in scanned PDF documents. Additionally, there is no guarantee for tables in untagged structured PDF documents to be represented in correct order due to lack of tags.

Table structure often contains vertical lines as column separators and horizontal lines as row separators. In an optimal case, all horizontal and vertical lines which produce table structure would be specified. Accessing each data cell individually is based on finding the related column and row intersection point. TableReader is a developed module in this research for cell segmentation. It uses morphology dilate technique, first removes vertical lines and obtains row positions, then removes horizontal lines and obtains columns positions. TableReader segments table to cells by finding (Xmin,Ymin) and (Xmax,Ymax) for each cell which represent cell bounding box. All cell segments are tagged based on positions and sent individually to OCR software.

TableReader considers horizontal and vertical line borders as the cells separators in table structure. Image processing techniques such as morphology, compose and composite are used to obtain the positions of the borders as follows:

1. "Morphology dilate rectangle:10x1" is used to remove vertical lines and keep horizontal lines. In remaining image foreground pixels Y-values indicate positions of horizontal lines. The following snippet is used to remove vertical lines and obtain horizontal lines position:

   ```
   convert Binary-Table.png -morphology dilate rectangle:10x1
      txt:-|grep -Ev '#FFFFFF'| sed '1d; s/:.*/ /g;s/,/ /g'awk
      '{print $2,$1}'|sort -b -k1n,1|awk 'p{print $1-p,$1,$2}{p
      =$1}{if (NR==1)print $1,$1,$2}'|awk '$1>1'|awk '{print $2
      }'>horizontal.dat
   ```

2. "Morphology dilate rectangle:1x10" is used to remove horizontal lines and keep vertical lines. In remaining image foreground pixels X-values indicate positions of vertical lines. The following snippet is used to remove horizontal lines and obtain horizontal line position:

   ```
   convert Binary-Table.png  -morphology dilate rectangle:1x10
       txt:-|grep -Ev '#FFFFFF'| sed '1d; s/:.*/ /g;s/,/ /g'|
      awk '{print $1,$2}'|sort -b -k1n,1|awk 'p{print $1-p,$1,
      $2}{p=$1}{if (NR==1)print $1,$1,$2}'|awk '$1>1'|awk '{
      print $2}'>vertical.dat
   ```

3. “ Compose lighten” the result of step 1 and 2 to generate an image of common points. This image shows intersections of horizontal and vertical lines.The below command is used to obtain intersection points of horizontal and vertical lines

   ```
   convert horizontal.png vertical.png  -compose lighten -
       composite  txt:-|grep -Ev '#FFFFFF'| sed '1d; s/:.*/ /g;s
       /,/ /g'|awk '{print $1,$2}'|sort -b -k1n,1| uniq >
       intersection.dat
   ```

These commands are used to find the number of horizontal/vertical borders:

```
Number-of-Horizontal-Borders=$(cat horizontal.dat|wc -l)
Number-of-Vertical-Borders=$(cat vertical.dat|wc -l)
```

In terms of cell segmentation, the tables are divided into two categories, namely, regular and irregular tables.

5.7.1 Tables Categories

5.7.1.1 Regular table

In regular tables, all vertical borders meet all horizontal borders so all intersections are the coordinate points of cell bounding boxes. It means obtaining intersections leads to access to cells bounding boxes. In regular tables of size MxN, where M is number of rows and N is number of columns, each row has N number of cells.The below pseudo code defines a regular table.

```
Total Number of cells MxN
M=Number of rows
N=Number of columns
for (i =1 to M): ROW(i) has N cells
```

If X_i,Y_j is the top left coordinate point then:
X_{i+1},Y_j is top right coordinate point.
X_i,$Y_{j+1}$is bottom left coordinate point.
X_{i+1},$Y_{j+1}$is bottom right coordinate point.
These commands provide dimensions of original table image:

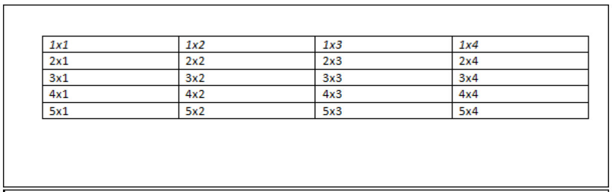
```
Height=$(identify -format %h Binary-Table.png )
Width=$(identify -format %w Binary-Table.ng )
```

The following snippet obtains cell bounding boxes in a regular table and extracts cells using convert tool and option -crop

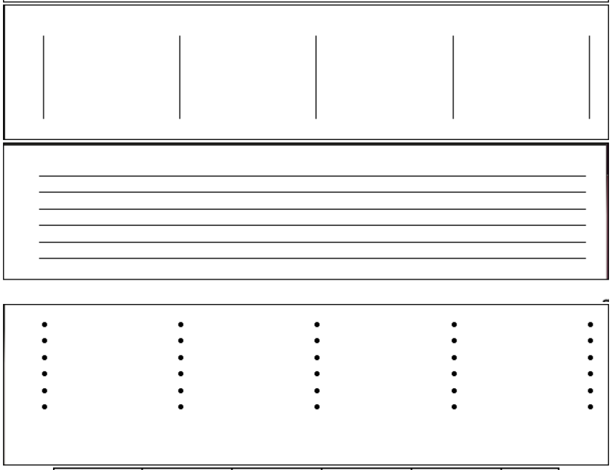
```
for (( j=1 ; j<=$(($Number-of-Horizontal-Borders));j++ )) do
for (( i=1 ; i<=$(($Number-of-Vertical-Borders));i++ )) do
Xstart=$(cat vertical.dat|awk 'NR=='$i'')
Xend=$(cat vertical.dat|awk 'NR=='$i'+1')
```

```
Yend =$(cat horizontal.dat|awk 'NR=='$j'+1')
Ystart=$(cat horizontal..dat|awk 'NR=='$j'')
Cell-Width=$[$Xend-$Xstart]
Cell-Height=$[$Yend-$Ystart]
convert Binary-Table.png -shave 0x0 -repage $Width"x"$Height+0+0
   png:-| convert png:-  -crop     $Cell-Width "x"$Cell-Height"+"
   $Xstart"+"$Ystart      colum'$i'row'$j.png
done
done
```

Figure 5.26 illustrates an image of a regular table, its vertical borders, horizontal borders, intersection points and the table of cell bounding boxes.



1x1	1x2	1x3	1x4
2x1	2x2	2x3	2x4
3x1	3x2	3x3	3x4
4x1	4x2	4x3	4x4
5x1	5x2	5x3	5x4



Width	Height	Xstart	Ystart	Xend	Yend
134	16	176	31	310	47
134	16	176	64	310	80
134	16	176	97	310	113
134	17	176	47	310	64
134	17	176	80	310	97
135	16	312	31	447	47
135	16	312	64	447	80
135	16	312	97	447	113
135	17	312	47	447	64
135	17	312	80	447	97
136	14	175	32	311	46
136	14	175	65	311	79
136	14	175	98	311	112
136	15	175	48	311	63
136	15	175	81	311	96
137	14	311	32	448	46
137	14	311	65	448	79
137	14	311	98	448	112
137	15	311	48	448	63
137	15	311	81	448	96

Figure 5.26: From top to bottom: Illustration of a regular table (BinaryTable.png), vertical borders (vertical.png), horizontal borders (horizontal.png), vertical/horizontal borders intersection (intersection.png) and the values of cells bounding boxes

5.7.1.2 Irregular table

There is no particular relationship between number of rows and number of columns in irregular tables. Therefore, there is no guarantee that each vertical border meets all horizontal borders. Thus all intersection points must be checked to ensure whether they are coordinate points of a cell bounding box or not.

Let (X_i,Y_i) be considered as the coordinate of the top left of a cell bounding box if and only if there are two intersection points such that (X_j,Y_j) is the top right coordinate value and (X_k,Y_k) is the bottom left coordinate value which meet the following conditions:

$X_j > X_i$ & $Y_i = Y_j$

$X_K = X_i$& $Y_k > Y_i$

As a result (X_j,Y_k) becomes the bottom right coordinate value of the cell which is extracted and labeled as Cell(i) using the top left and bottom right coordinate values of the cell bounding box.

The following snippet for all existing coordinate values in intersection.dat (explained in Section 5.7) checks mentioned conditions to extracts cell bounding boxes from an irregular table.

Supposing (Xstart , Ystart) is top-left coordinate value in bounding box

(Xend,Ystart) is top-right coordinate value in bounding box which Xend is greater than Xstart.The following snippet is used to obtain Xend.

```
Number-of-Intersections=$(cat intersecrtion.dat |wc -l)
for (( i=1 ; i<=$Number-of-Intersections;i++ ))
do
Xstart=$(cat intersection.dat|awk 'NR=='$i''|awk '{print $1}')
Ystart=$(cat intersection.dat|awk 'NR=='$i''|awk '{print $2}')
Xend=$(cat intersection.dat|awk '$1>'$Xstart  && $2=='$Ystart''|
   sort -b  -k1n,1 |awk 'NR==1{print $1}')
```

(Xend ,Yend) is bottom right coordinate value in bounding box which Yend is greater than Ystart. The following snippet is used to obtain Yend.

```
if [[ -n $xe ]];then
Yend=$(cat intersection.dat|awk '$2>'$Ystart' && $1=='$Xend''|
   sort -b  -k2n,2 |awk 'NR==1{print $2}')
fi
```

Obtaining Xend ,Yend supports to calculate width and height of the cell and extract the cell as the foll wing snippet indicates.

```
Cell-Width=$[$Xend-$Xstart]
Cell-Height=$[$Yend-$Ystart]
convert Binary-Table.png -shave 0x0 -repage $Width"x"$Height+0+0
   png:-|convert png:- -crop   $Cell-Width"x"$Cell-Height"+"
   $Xstart"+"$Ystart  cell'$i.png
fi
done
```

Figure 5.27 presents an irregular table and its table of cell bounding boxes.

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td colspan="2">5</td><td>6</td><td rowspan="3">7</td></tr>
<tr><td>8</td><td>9</td><td>10</td></tr>
<tr><td>11</td><td>12</td><td>13</td></tr>
<tr><td>14</td><td colspan="3">15</td></tr>
</table>

cell	Width	Height	Xstart	Ystart	Xend	Yend
1	136	16	27	30	163	46
5	272	17	27	46	163	63
8	136	16	27	63	163	79
11	136	17	27	79	163	96
14	136	16	27	96	163	112
2	136	16	163	30	299	46
9	136	16	163	63	299	79
12	136	17	163	79	299	96
3	137	16	299	30	436	46
6	137	17	299	46	436	63
10	137	16	299	63	436	79
13	137	17	299	79	436	96
4	136	16	436	30	572	46
7	136	50	436	46	572	96
15	436	16	163	96	572	112

Figure 5.27: Sample of an irregular table and its cells bounding box values

The Complete code for TableReader is available in Appendix B.

5.8 High Volume Transactional Output (HVTO) Segmentation

HVTOs (explained in Section 2.10.8) are provided to deliver to the customer. In terms of accessibility HVTO are divided to two categories in this research:

1. Structured, but not necessarily tagged. Consequently, they are not navigable. Since a HVTO contains several separated items, the HVTO reading process by a user is very different from a normal document reading process, which is done sequentially line by line, from top left to bottom right. Thus, navigation ability is a very important capability during a HVTO reading session. Although this category is not an image, in some cases they are not accessible through screen readers due to PDF properties such as restrictions adopted during creation or Digital Rights Management (DRM) technology that publishers use to control or restrict digital media content on electronic devices. If the PDF is structured, by converting it to XML, each separable item will be converted to an individual XML element and then becomes accessible through screen readers. These elements do not guarantee accurate navigation and usability. By further investigation, modification based on XML, and parsing this category will be accessible, usable and navigable.

2. Scanned PDF HVTO documents are inaccessible and require OCR to extract the text from it. However, doing OCR before running segmentation may destroy information. HVTO segmentation must be performed before sending it to OCR. This uses the block segmentation technique to segment HVTO vertically and horizontally, then each block can be sent to OCR individually without creating conflict with another block or destroying the conceptual meaning of HVTO.

Due to the not-in-order nature of HVTO, in reading it by vision-impaired users besides accessibility through screen readers, usability and keep appropriate ordering must be considered. HVTO segmentation supports usability in reading HVTO.

Block Segmentation method explained in Section 5.4.5 is used for HVTO segmentation.

Figure 5.28 illustrates a sample of bill image, and its results of horizontal segmentation and vertical segmentation. The blue lines are used to determine bounding box of each block segment.

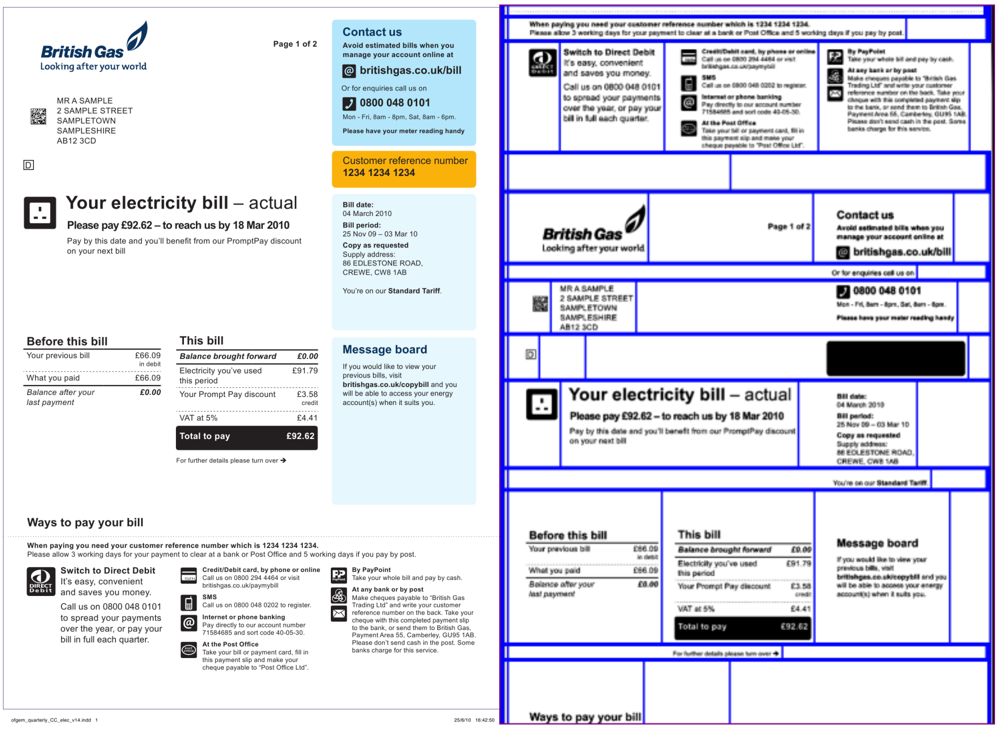


Figure 5.28: HVTO Segmentation: The sample of a bill (British Gas, 2013) (left) and result of horizontal/vertical segmentation (right)

5.9 Non-Textual (Graphical) Components Accessibility

5.9.1 ChartRecognition

Text/non-text segmentation of document layout analysis classifies blocks of the page to text and non-text areas. Images of non-text areas are sent to ChartRecognition routine for chart classification such as pie, bar or line chart and presents the extracted data as comprehensive description to vision-impaired users in audio format. The goal is to provide simple to use, efficient, and available presentation schemes for non-textual information in comprehensible form without the need for additional devices or equipment. The output is a textual summary of the graphic including the core content of the hypothesized intended message of the graphic designer. The textual summary of the graphic is then conveyed to the user in audio format by TTS software. The benefit of this approach is automatic provision of information to vision-impaired. This method combines both passive and active accessibility approaches. In other words, this method presents a complete description of illustration (passive) and provides navigation ability through various fields of data which helps users to explore and build a mental map of the visual components (active). Since large quantities of data are presented in a graphic, efficient access strategies are required according to the tasks that users are trying to accomplish during navigation through information.

In general terms, each chart structure contains two parts:

1. Text: including: the number and amount of each axis ticks, axis labels, legends and title.

2. Image including: graph body or chart body (Watanabe et al., 2012)

Therefore, the ChartRecognition module includes two sub-modules which are responsible for extraction of text data and image data.

1. Text data extraction by running OCR retrieves some conceptual data from charts such as axes tick values, chart title, axes labels, minimum and maximum value for each axis. The precision of retrieved information in this sub-module is highly dependent on precision of OCR.

2. Image data extraction contains graph digitization to provide data table by processing it. Data table is a table with two columns to shows amount of each dependent variable related to its independent variable. The used image processing techniques in this research support the accuracy of the digitization data table.

It was assumed in this research that all pie and bar chart images contain two major components: Chart and Legend. ChartRecognition module uses two hypotheses regarding legend: 1) Data in pie chart legend is in descending order; and 2) Data in bar chart legend represents bars from left to right. Chart component in bar chart essentially has two perpendicular axes and in pie chart has a sectored circle. Legend in most cases consists of two columns: colour samples and labels. Legend is actually a map for understanding meaning of chart by matching two columns.

Chart/Legend Segmentation is performed using following steps:

1. Finding largest connected area in image of chart;

2. Labelling largest connected area as chart-only image; and

3. Removing chart-only image from original image to obtain legend-only image.

The code for Chart-Legend Segmentation is available in Appendix B.

Figure 5.29 shows Chart-Legend Segmentation Results for two samples.[9] Figure 5.29-Left, from top to bottom illustrates a sample of Pie Chart, Pie-Chart-Only and Legend-Only images. Figure 5.29-right, from top to bottom illustrates a sample of Bar Chart, Bar-Chart-Only and Legend-Only images.

[9]Note: The quality of image is poor(blur) as image was extracted from real scanned document

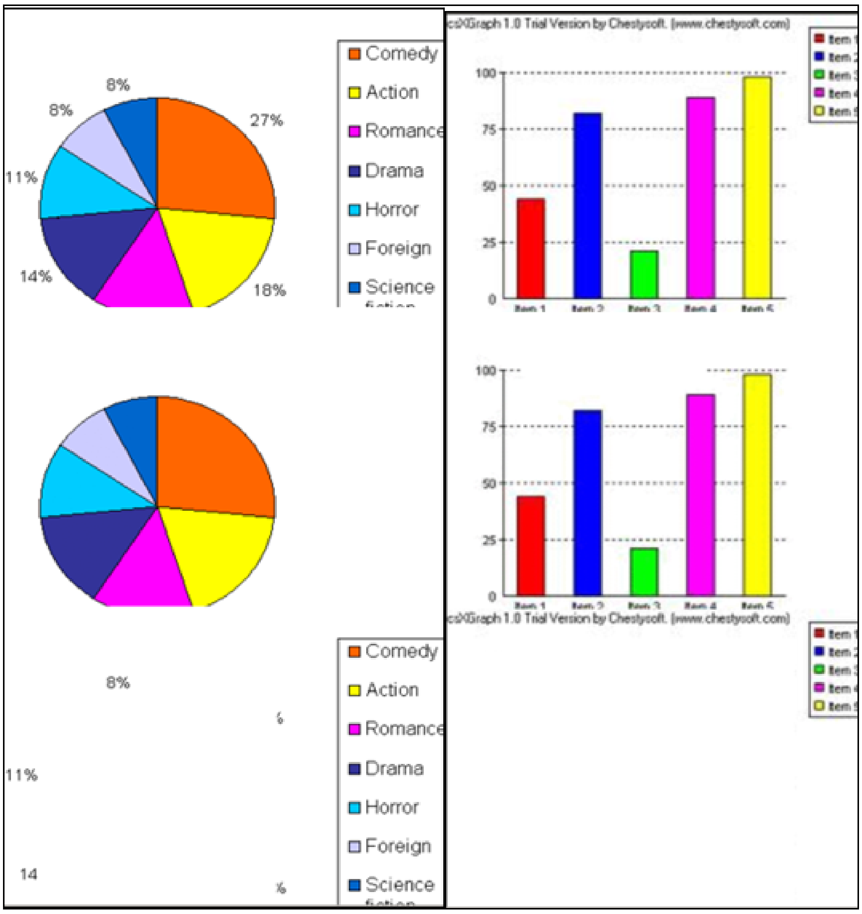


Figure 5.29: Chart-Legend Segmentation Results: Left: from top to bottom a sample of Pie Chart, Pie-Chart-Only and Legend-Only images. Right: from top to bottom a sample of Bar Chart, Bar-Chart-Only and Legend-Only images

Chart-Only image is applied to Horizontal/Vertical Lines Removal which removes horizontal lines by "-morphology dilate rectangle:1x10" as the following command indicates:

```
convert Chart-Only.png  -morphology dilate rectangle:1x10
    Removed-Horizonatl-Lines.mpc
```

Result of horizontal lines removal is sent to vertical lines removal to remove vertical lines by "morphology dilate rectangle:10x1" as the following command indicates:

```
convert Removed-Horizonatl-Lines.mpc -morphology dilate rectangle
    :10x1   Removed-Horizonntal-Vertical-Lines.mpc
```

The result of Horizontal/Vertical Lines Removal is converted into binary and the remaining horizontal lines from binary image is removed to acquire shape-only image. Shape-only image supports to classify charts using feature extraction. This command generates shape-only image.

```
convert Removed-Horizonntal-Vertical-Lines.mpc -threshold 70% mpc
    :-|convert  mpc:-  -morphology dilate rectangle:1x20  -negate
    -trim -negate shape.txt
```

These features can be extracted from the shape-only image and The following snippets are used to extract these features.

Number of remaining unconnected pieces = P

```
P=$(cat  shape.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |sed
    's/,/ /g'|sort -b -k1n,1|awk '{print $1}|| awk 'p{print $1,$1-
    p}{p=$1}{if (NR==1) print $1,$1}'|awk '$2>1'|wc -l)
```

Number-of-Remaining Black Pixels = RBP

```
RBP=$RBP =$( convert original.mpc -threshold 70% -morphology
    dilate :30 diamond png :- | convert png :- -rotate 90 png :-|
    convert png :- - morphology dilate :30 diamond txt :-| sed '1d
     '| grep -Ev '# FFFFFF '| wc -l))
```

Distance Diversity = Number of horizontal distance between pieces = DD

```
DD=$(cat shape.txt| awk 'p{print $1,$1-p}{p=$1}{if (NR==1) print
    $1,$1}'|awk '$2>1'|sed '1d'|awk '{print $2}'|sort|uniq|wc -l)
```

Deviation-of-Distance Diversity = Amount of variation of distances = DDD

```
DDD=$(cat shape.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |sed
     's/,/ /g'|sort -b -k1n,1|awk '{print $1}| awk 'p{print $1,$1-
    p}{p=$1}{if (NR==1) print $1,$1}'|awk '$2>1'|sed '1d'|awk '{
    print $2}'|sort|uniq|awk 'p{print $1-p}{p=$1}'|awk '$1>2'|wc -
    l)
```

Deviation-of-Height Diversity= Amount of variation of height of parallel pieces = DHD

```
DHD=$(cat shape.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |sed
     's/,/ /g'|sort -b -k1n,1|awk '{print $1}'| awk 'p{print $1,$1
    -p}{p=$1}{if (NR==1) print $1,$1}'|awk '$2!=1'|awk '{print $1
    }'|awk '{  a[$1]++}END { for(i in a)  print i,a[i]}' |sort -b
    -k1n,1| awk 'p{print $1,$1-p,$2}{p=$1}{if (NR==1) print $1,$1,
    $2}'|  awk '$2>1'|awk '{print $3}'|sort|uniq|awk 'p{print $1-p
    }{p=$1}'|awk '$1>2'|wc -l)
```

The feature extracted are used to chart classification as follows:

- If RBP!=0 and P<2 then chart is Pie
- If RBP!=0 , P>2 , DDD=0 and DHD !=0 then the chart is Bar
- If RBP!=0 , P>2 , DDD!=0 or DHD =0 then the chart is Line
- If RBP=0 , P<=1 then the chart is Line

- If RBP=0 , P>1, DDD=0 and DHD !=0 then the chart is Bar
- If RBP=0 , P>1, DDD!=0 or DHD =0 then the chart is Line

Figure 5.30 illustrates the flowchart for chart classification using feature extraction.

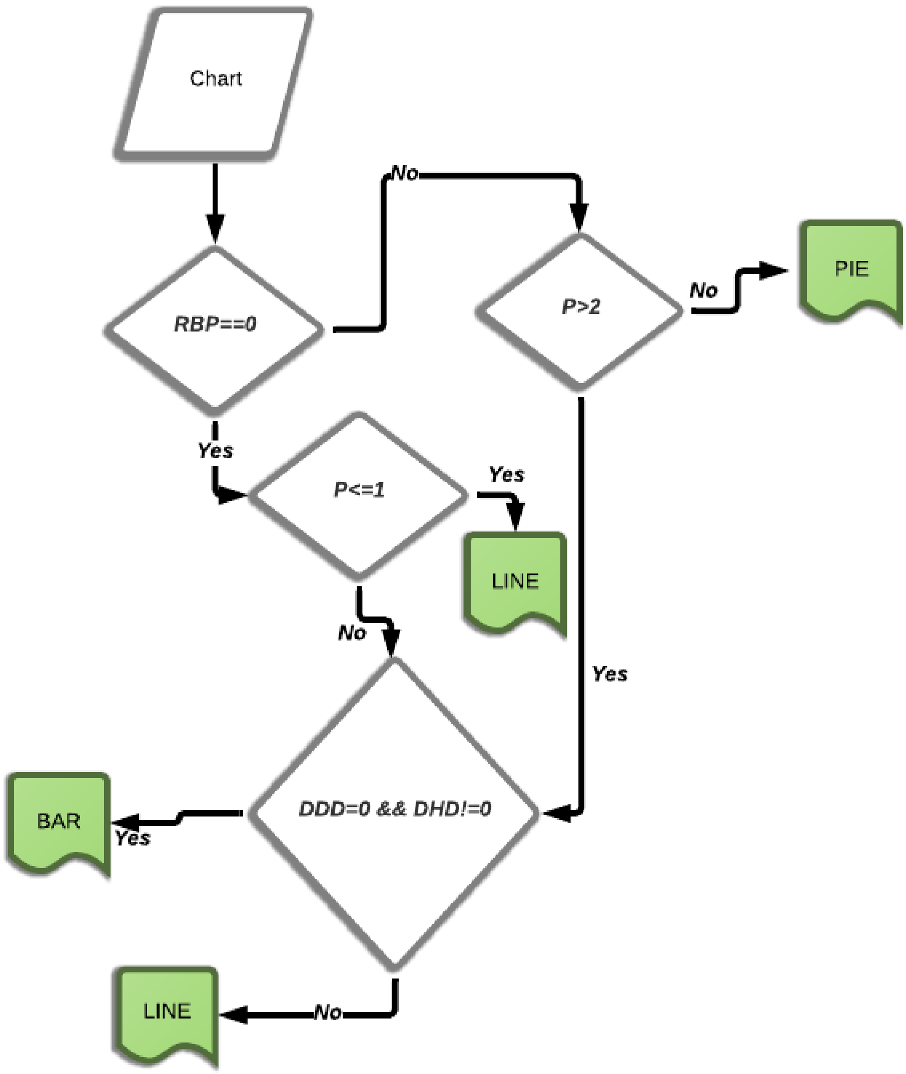


Figure 5.30: Chart classification

Figure 5.31 shows horizontal/vertical lines removal and binary conversion results for the samples of pie and bar chart in Figure 5.29.

Figure 5.31 Left-Top illustrates a Pie-Chart-Only. image, Figure 5.31 Left-Bottom result of horizontal/vertical lines removal and binarizing for the Pie-Chart-Only.

Figure 5.31 Right-Top illustrates a Bar-Chart-Only image, Figure 5.31 Right-Bottom result of horizontal/vertical lines removal and binarizing for the Bar-Chart-Only.

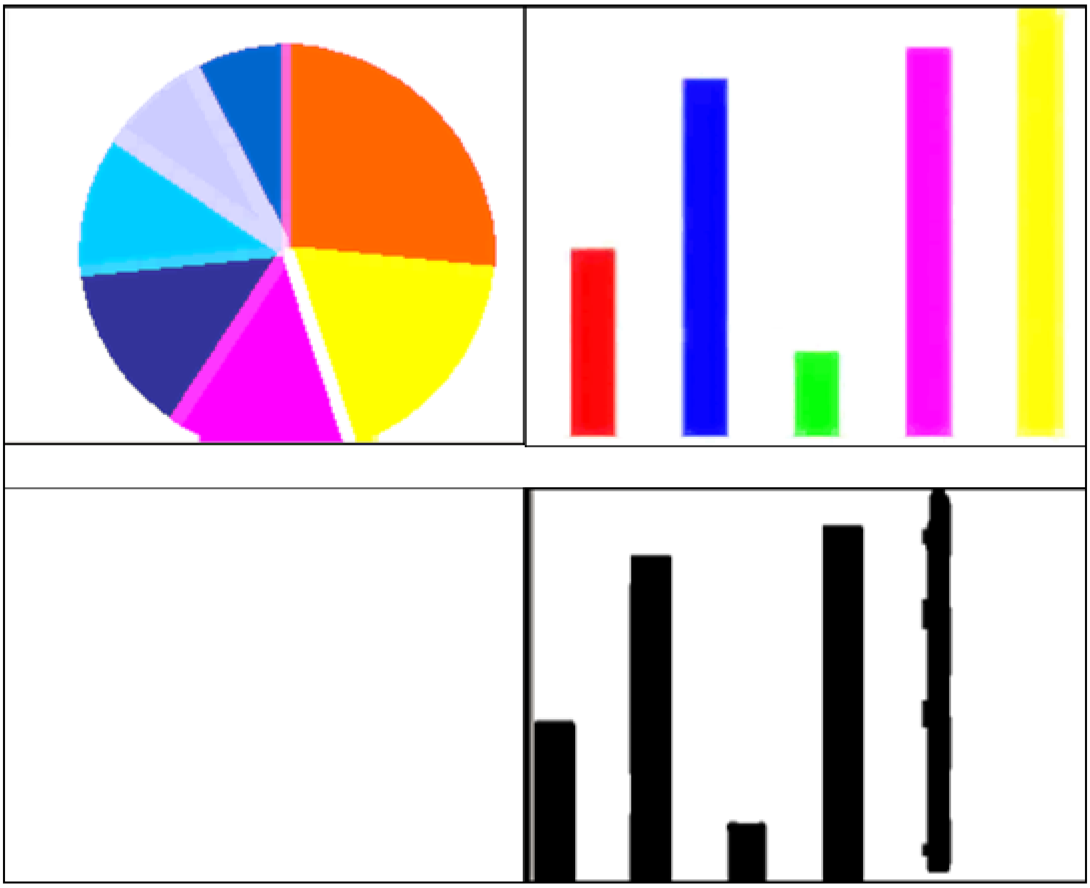


Figure 5.31: Horizontal/Vertical lines removal and binarizing result for samples of pie chart and bar chart. Left: top to bottom, Pie-Chart-Only and its Binarized-Removed-Horizontal-Vertical-Lines. Right: top to bottom Bar-Chart-Only and its Binarized-Removed-Horizontal-Vertical-Lines

5.10 Examined Non-textual Components in this Research

5.10.1 Bar Chart and BarChartReader

The bar chart is commonly used visualization techniques often used in newspapers, journals and magazines. Bar chart includes rectangular bars with lengths proportional to the values that they represent. Bar charts provide a visual presentation of categorical data (Kelley & Donnelly, 2009). Categorical data is a grouping of data into discrete groups, such as months of the year. The bars can be plotted vertically or horizontally to show comparisons among categories. One axis of the chart shows the specific categories being compared, and the other axis represents a discrete value. A vertical bar chart is sometimes called a column bar chart. These charts contain rectangles in which the height of them indicate the quantities of the corresponding variables. Bar charts are usually scaled so that all the data can be fitted onto the chart and may be arranged in any order. In a column bar chart, the categories appear along the horizontal axis; the height of the bar corresponds to the value of each category. A bar chart is very useful for recording discrete data. This research focused on Column Bar Chart which is presented with a legend.

The main conceptual information which are required to provide bar chart accessibility are:

- The chart title;
- The labels of axes;
- Number of categories;
- Bars labels or title of categories;
- The height of each bar; and
- Range of discrete values

BarChartReader as part of ChartRecognition in this research is responsible for:

- Segmenting the image of bar chart to chart-only and legend-only;
- Segmenting chart-only to horizontal axis, vertical axis and bars-only image; and
- Applying bars-only image to second sub-module of ChartRecognition to extract number of bars and their height.

After recognising bar chart using the mentioned features, bar-chart-only will be sent to Horizontal/Vertical Axes Segmentation which uses following snippet.

To extract X-Axis:

```
convert Binary-Bar-Chart-Only.png -morphology dilate rectangle:15
    x1 temp.png
convert Binary-Bar-Chart-Only.png temp.png -compose Lighten -
    composite horizontal-lines.txt
```

The greatest value of y in horizontal-lines image is Y-Position-of-X-Axis.

```
Y-Position-of-X-Axis=$[$(cat horizontal-lines.txt|grep -Ev white
     | sed '1d; s/:.*//g;s/,/ /g'|sort -b -k2n,2|tail -1|awk '{
    print $2}')-5]
```

X-Axis is obtained by cropping Binary-Bar-Chart-Only image from coordinate point (0,Y-Position-of-X-Axis) with dimensions Width x (Height -(Y-Position-of-X-Axis))

```
convert Binary-Bar-Chart-Only.png -crop "0x"$[$Height-$Y-Position
    -of-X-Axis]"+0+"$Y-Position-of-X-Axis X-Axis.png
```

To extract Y-axis:

```
convert Binary-Bar-Chart-Only.png -morphology dilate rectangle:1
    x15 temp.png
convert Binary-Bar-Chart-Only.png temp.png -compose Lighten -
    composite vertical-lines.txt
```

The minimum value of x in vertical-lines image is X-position-Y-axis.

```
X-Position-of-Y-Axis=$(cat vertical-lines.txt |grep -Ev white | sed '1d; s
/:.*//g;s/,/ /g'|sort -b -k1n,1|head -1|awk '{print $1}' )
```

Y-axis is obtained by cropping bar-chart-only image from coordinate point (0,0) with dimensions (X-Position-of-Y-Axis) x Height.

```
convert  Binary-Bar-Chart-Only.png -crop $X-Position-of-Y-Axis "
    x0+0+0" Y-axis.png
```

Dimensions of Bar-Only image is (Width-(X-Position-of-Y-Axis)) x Y-Position-of-X-Axis

Bars-Only image starts from coordinate value point (X-positon-Y-axis,0), as the following command indicates:

```
convert  Binary-Bar-Chart-Only.png -crop $[$Width-($X-Position-of
    -Y-Axis)]"x"$Y-Position-of-X-Axis"+"$X-Position-of-Y-Axis"+0"
    Bars-Only.png
```

Figure 5.32 illustrates Horizontal/Vertical Axes Segmentation results for the bar chart in Figure 5.29.

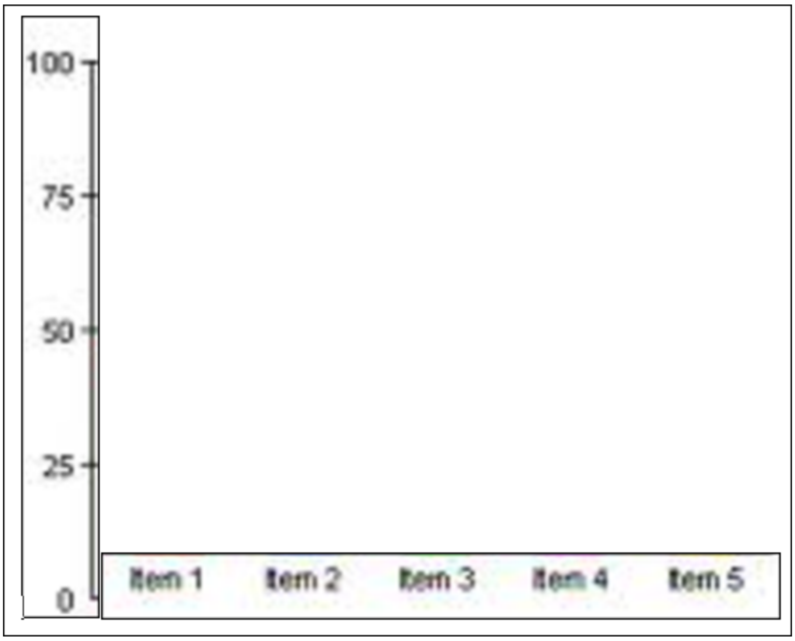


Figure 5.32: Horizontal/Vertical Axes Segmentation

The tasks which are related to first ChartRecognition sub-module are:

- Apply images of horizontal and vertical axes to the OCR to obtain the bar labels, horizontal title, vertical range, vertical title and vertical tick value ;
- Apply images of legend-only to obtain the bar labels; and
- Match the labels and bars.

Successful performance of the first ChartRecognition sub-module allows to calculate conversion rate to convert bars height in pixel scaling into real height value and make an alternative descriptive text for bar chart.

The second ChartRecognition sub-module extracts all unique x-values of foreground pixels in Bars-Only and removes x-values with a difference of 1. In the result each individual x and its repeating amount represent a bar and its height respectively. The following command is used to obtain a data file containing the number and height of bars.

```
convert Bars-Only.png txt:- |sed 's/:.*#/ /g;1d;s/,/ /g'|grep
   black |sed  's/,/ /g'|sort -b -k1n,1|awk '{print $1}'|awk '{ a
   [$1]++}END { for(i in a) print i,a[i]}'|sort -b -k1n,1|awk 'p{
   print $1,$1-p,$2}{p=$1}{if (NR==1)print $1,$1,$2}'|awk '$2>1 |
    NR==1 {print $1,$3}'>Bars.dat
```

The lowest/highest bars on the chart and two bars which have the closest values are emphasized information implicit in bar chart. The second sub-module of ChartRecognition related to BarChartReader is responsible to extract this information. Assuming Bars.dat contains the height of bars, the emphasized information can be provided by these commands:

```
Lowest Bar: min=$(cat bars.dat |sort b k2n,2|awk {print $1}'|head
    -1)
Highest Bar :max=$(cat bars.dat|sort b k2n,2|awk '{print $1}'|
   tail -1)
Number of bars=$(cat bars.dat|wc l)
Bars-Height=($(cat bars.dat|awk {print $2}'))
```

The code for ChartRecognition sub-module 2 related to BarChartReader is available in Appendix B.

Figure 5.33 illustrates: a) A sample of Bar Chart; b) Chart-only; c)Legend-only; d) Vertical axis; e) Horizontal axis; and F) Binary-Bars-Only images by BarChartReader.

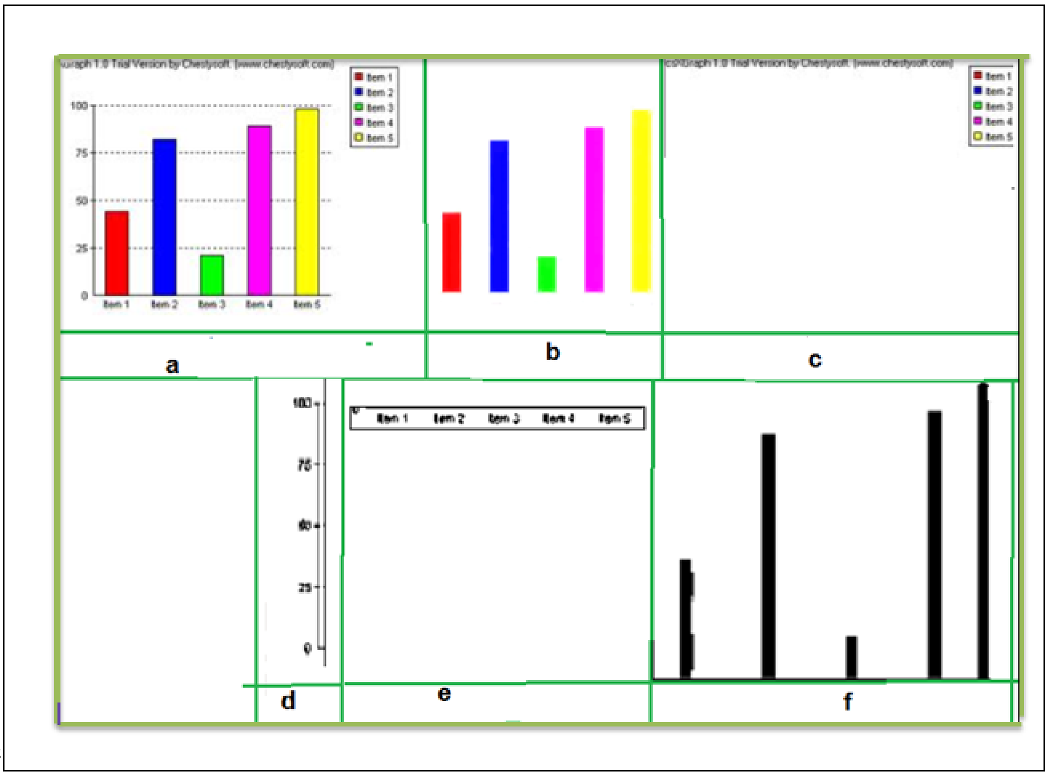


Figure 5.33: a) Bar chart; b) Chart-only; c) Legend-only; d) Vertical axis; e) Horizontal axis; and f) Binary-Bars-Only images by BarChart Reader.

5.10.2 Line Chart and LineChartReader

A line chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments (Andreas, 1965). It is similar to a scatter plot except that the measurement points are ordered (typically by their x-axis value) and joined with straight line segments. Line charts show how particular data changes at equal intervals over time or some other control.

A line chart is often used to visualize a data trend over time intervals (time series), thus the horizontal line axis is often drawn chronologically (Salkind, 2006). A line chart is typically drawn bordered by two perpendicular lines, called axes. The horizontal axis is called the x-axis and the vertical axis is called the y-axis. Typically, the y-axis represents the dependent variable and the x-axis represents the independent variable. The individual axes represent numbers with small marks called ticks, indicating significant values on the line. The ticks may be annotated with the value they represent. A short description of the axis is often used to annotate each axis as labels.

The chart may contain an overall description called a title and if the chart contains more than one line, it may contain a list describing each line, called a key or a legend. Finally, the presented data is plotted at the intersection of the perpendicular lines extending from the axes, and straight-line segments are drawn between those intersection points.

In experimental sciences, data collected from experiments are often visualized by a graph or line chart. The table "visualization" is an accepted way of displaying exact values, but can be a poor way to understand the underlying patterns of represented values. Understanding the process described by the data in the table is aided by producing a graph or line charts. Thus, for line chart accessibility, graph digitization is a process which involves converting the pixels in the image of line chart to original (x, y) data values. Extracted original (x, y) data values from the image helps to generate a digitized data table (Ryan, 1986). Therefore, graph digitization has two responsibilities, first finding the position of pixels of the main graph in the image and second running a mathematical function for all found pixels to assign a coordinate system to the pixels in the graph based upon the extracted scaling values, line chart image size and line chart's origin point pixel position.

Although the basic concept of converting the image pixel values to scaled values is straightforward, practical consideration such as specifying an accurate vertical and horizontal ratio must be addressed. The emphasized information that must be extracted from the image for data extraction that contains the graph digitization are:

- Counting the number of lines;
- Counting the number of bends on the line; and
- Counting the number of intersections.

The LineChartReader module in this research focused on a single line chart. It is responsible for:

- Segmenting original image of line chart to horizontal and vertical axes and graph-only image;
- Removing grid lines from original image to discriminate grid lines from the axes and obtain markers-only image; and
- Applying markers-only image to second sub-module of ChartRecognition to obtain exact position of markers in image.

The factors must be considered and described in a line chart to provide accessible line charts contain: chart title, axes range, axes label and the number of vertical and horizontal ticks. This information is extracted from vertical and horizontal axes by sending them to the first sub-module OCR related of ChartRecognition. Digitization accuracy firmly depends on the output results of ChartRecognition first sub-module. These results assist in obtaining X-Ratio and Y-Ratio for digitization as it is shown in formulae given below:

X-Ratio=(Difference of two consecutive Xticks)(Number of Xticks)/W

Y-Ratio=(Difference of two consecutive Yticks)(Number of Yticks)/H

$X - Ratio = MaxXtick/W$

$Y - Ratio = MaxYtick/H$

Assuming (Xp,Yp) is a pixel position coordinate value, and (X,Y) is original value in chart, conversion of pixel to data was carried out using the following formulae:

$X = X - Ratio * Xp$

$Y = Y - Ratio * Yp$

After performing these processes, digitized data table of the different values for X and Y are generated. The X,Y slope calculation of the individual line between two adjacent points is also calculated and added to data table using this formula:

$slope_i = (Y_i - Y_{i-1}) / (X_i - X_{i-1})$

Calculating the slope between two points helps provide better summary description for line chart trends using the following rules:

- If a line segment slopes upward from left to right, the slope is positive, the graph is increasing;
- If a line segment slopes downward from left to right, the slope is negative, the graph is decreasing;
- If a line segment is horizontal, the slope is 0 (zero), the graph is fixed; and
- If a line segment is vertical, the slope is undefined.

Figure 5.34 illustrates the sample of line chart, Horizontal/Vertical segmentation results, Horizontal grid lines removal result; and Marker-Only images extracted by LinechartReader.

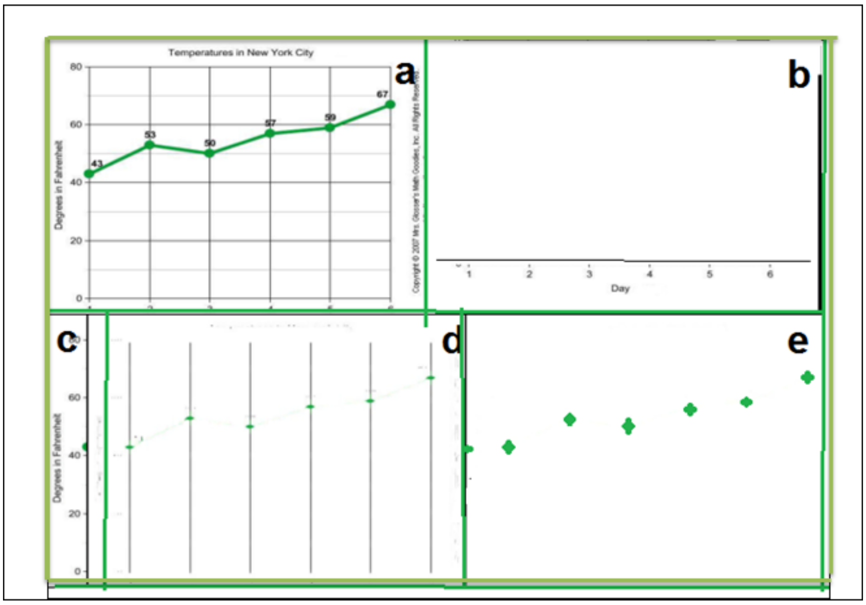


Figure 5.34: a) A line chart sample; b) Horizontal axis; c) Vertical axis; d) Horizontal grid lines removal result; and e) Marker-Only images by LineChartReader

5.10.3 Pie Chart and PieChartReader

A pie chart is a circular chart divided into sectors, illustrating numerical proportions. In a pie chart, the arc length of each sector, its central angle and area are represented in proportional quantity (Cleveland, 1985; Friendly, 2008). Pie charts are widely used in the business world and the mass media (Good & Hardin, 2003). However, many experts have criticized pie charts and recommend avoiding them. Statisticians generally regard pie charts as a poor method of displaying information and they are uncommon in scientific literature. Pie charts can be replaced in most cases by other plots such as the bar chart. Pie charts take up larger amount of space on the page compared to the more flexible alternative of bar charts, which can display other values such as averages or targets at the same time. Representing large amounts of data with pie charts make too small slices which have to rely on colours, textures or arrows. In addition representing several values with the same colour make interpretation difficult.

Most subjects have difficulty ordering the slices in the pie chart by size. When the bar chart is used, the comparison is much easier (Cleveland, 1985). Similarly, comparisons between data sets are easier using the bar chart. However, if the goal is to compare a given category (a slice of the pie) with the total (the whole pie) in a single chart, then a pie chart can often be more effective than a bar graph (Simkin & Hastie, 1987; Spence & Lewandowsky, 1991).

The main concepts in pie chart, which are required for pie chart accessibility are pie chart title, number of sectors, and title of sectors. In most cases, this information is provided by caption or legend.

PieChartReader in this research is responsible to

- Segment the image of pie chart to Pie-Only and Legend-Only images;
- Apply Pie-Only image to colour reduction and colour quantization;
- Obtain colour histogram; and
- Obtain colour percentage.

PieChartReader may not recognise pie chart with slices in close colour range. For pie chart slice percentage calculation, the following steps were used:

1. Identification of dimensions of pie-chart-only image

   ```
   w=$(identify -format "%w" Pie-Chart-Only.png)
   h=$(identify -format "%h" Pie-Chart-Only.png)
   ```

2. Conversion the background of Pie-Chart-Only to transparent. Transparent-Background-Pie-Only was generated to avoid ambiguity of sectors with colours are close to background colour. The following command is used to convert background colour to transparent.

   ```
   convert Pie-Chart-Only.png  -fill none -draw "matte 0,0
      floodfill"  mpc:-|convert mpc:-  -fill none -draw "matte
      0,$h floodfill"  mpc:-|convert mpc:-  -fill none -draw "
      matte $w,0 floodfill" mpc:-|convert mpc:-  -fill none -
      draw "matte $w,$h floodfill" mpc:-|convert mpc:-
      Transparent-Background-Pie-Only.png
   ```

3. Colour reduction for Transparent-Background-Pie-Only image. The following command is used to reduce colours.

   ```
   convert Transparent-Background -Pie -Only -separate  -
      threshold  70%  -combine  Color-Rduced-Transparent-
      Background-Pie-Only.png
   ```

4. Colour histogram generation for Color-Reduced-Transparent-Background-Pie-Only image. The following snippet first calculates the total number of pixels in histogram, then computes the percentage of each colour in histogram.

```
SUM=$(convert Color-Reduced-Transparent-Background-Pie-Only.png -format %c
 histogram:info:-|awk '{print $1,$NF}'|grep -Ev black |awk '{sum+=$1}END{
print sum}')convert Color-Reduced-Transparent-Background-Pie-Only -format %c
histogram:info:-|awk '{print $1,$NF}'|grep -Ev black |awk '{print 100*$1/'$SUM
',$2}'
```

The tasks related to first ChartRecognition sub-module provide label of slices by sending legend only image to the OCR and make an alternative descriptive text for the pie chart matching label and percentage of slices. Each sector of pie chart has to be matched with the variable via the legends. The matching algorithm can interpret the underlying data.

Figure 5.35 shows from left to right a sample of Pie Chart, its Pie_Only, Legend-Only, Transparent-Background-Pie-Only and Color-Reduced-Transparent-Background-Pie-Only images extracted by PieChartReader.

```
SUM=$(convert Color-Reduced-Transparent-Background-Pie-Only.png -format
%c histogram:info:-|awk '{print $1,$NF}'|grep -Ev black |awk '{sum+=$1}END{
print sum}')convert Color-Reduced-Transparent-Background-Pie-Only -format %c
histogram:info:-|awk '{print $1,$NF}'|grep -Ev black |awk '{print 100*$1/'$SUM
',$2}'
```

The tasks related to first ChartRecognition sub-module provide label of slices by sending legend only image to the OCR and make an alternative descriptive text for the pie chart matching label and percentage of slices. Each sector of pie chart has to be matched with the variable via the legends. The matching algorithm can interpret the underlying data.

Figure 5.35 shows from left to right a sample of Pie Chart, its Pie_Only, Legend-Only, Transparent-Background-Pie-Only and Color-Reduced-Transparent-Background-Pie-Only images extracted by PieChartReader.

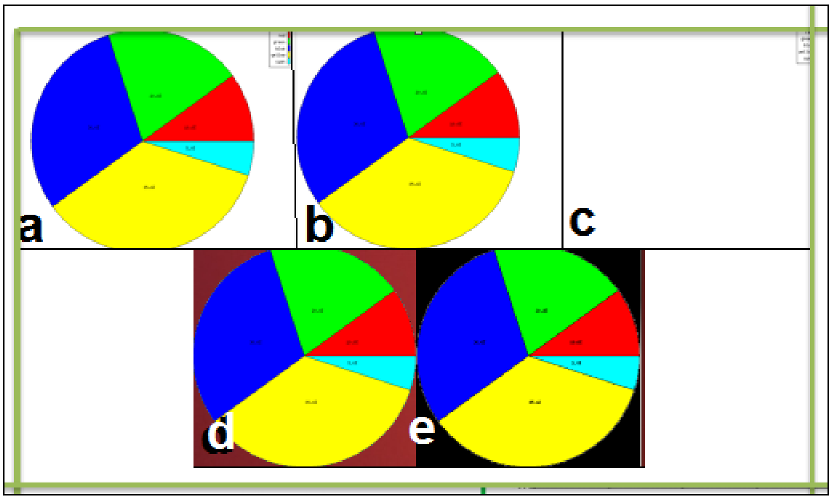


Figure 5.35: From left to right: a) A sample of Pie Chart; b) Pie-Only; c) Legend-Only; d) Transparent-Background-Pie-Only; and e) Color-Reduced-Ttransparent-Pie-Only

5.10.4 GNUPLOT Evaluation Tool

GNUPLOT is an open source command-line program that can generate two or three-dimensional plots of functions, data and data fits. It is frequently used for publication-quality graphics as well as education. One major accessibility benefit derived from GNUPLOT is encoded as plain text. Authors can create and edit it with a text-processor authoring tool. Plain text encoding leads to simplicity in use and rendering. GNUPLOT presents prepared, simple, text based and easy to use patterns to generate charts and graphs (Williams & Keklly, 2004). Data extracted from table digitization of LineChartReader or BarChartReader is sent to GNUPLOT as .gnu file to regenerate visual presentation. GNUPLOT is used for mathematical graph digitization to reproduce graph applying mathematical function. GNUPLOT supports error detection, correction and evaluate accuracy of chart reader modules.

This is a sample .gnu file, which is used by GNUPLOT as a configuration file to plot line chart. According to this configuration file, GNUPLOT plots a line chart, X-axis range is between 0 to 380 Y-axis range is between 0 to 360, input data table is line.dat, the output file is linechart.png and line is plotted in red colour.

```
#line.gnu
set term png truecolor
set output linechart.png
set xrange [0:380]
set yrange [0:360]
plot line.dat with lines lc rgb red notitle
```

line.dat contains digitization table or coordinate values extracted by LineChartReader.

Figure 5.36 illustrates an original line chart (left) and generated line chart by GNUPLOT based on data extracted by LineChartReader(right)

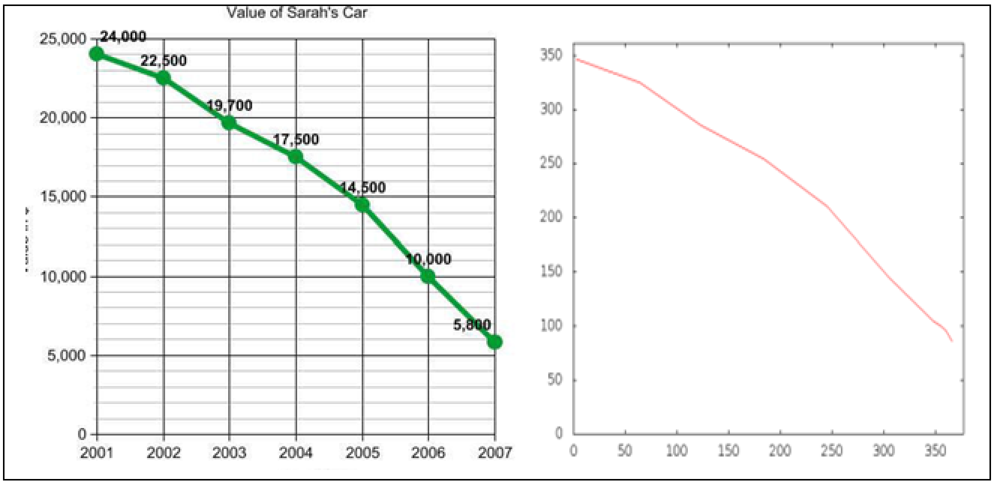


Figure 5.36: A sample line chart (left) and generated line chart by GNUPLOT based on data extracted by LineChartReader (right)

The following is the sample of .gnu file that is used by GNUPLOT to generate bar chart. According to this configuration file, GNUPLOT plots a bar chart, bar width is 0.5, input data table is bar.dat, the output file is barchart.png and bars are plotted by red colour.

```
#Barchart.gnuset term png truecolorset output bar.pngset boxwidth 0.5set style fill solidplot bar.dat using 1:2 withboxes rgb red
```

bar.dat is the result of ChartRecognition second sub-module from graph digitization. bar.dat file contains two columns.First column indicates the bar number, and second column is the height of each bar.

```
#bar.dat
1 79.1667
2 75.4167
3 55.5833
4 31.875
5 15.7917
```

Figure 5.37 illustrates an original bar chart (left) and generated bar chart by GNUPLOT based on data extracted by BarChartReader(right).

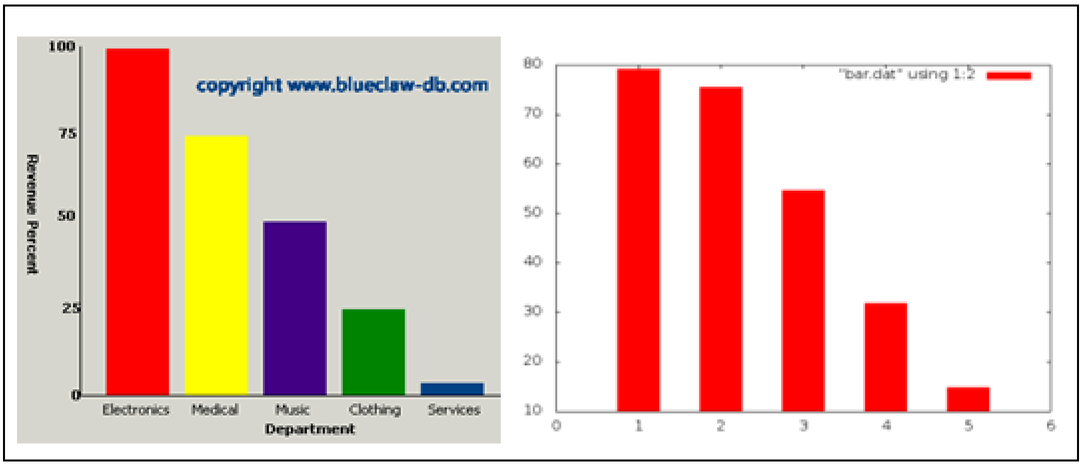


Figure 5.37: A sample bar chart (left) and generated bar chart by GNUPLOT based on data extracted by BarChartReader(right)

5.11 A Method to Present Chemical Equations to Vision-Impaired Students

5.11.1 Representing Chemical Equation in Markup Format

"Students with vision impairments will learn chemistry in the classroom and laboratory, test and enjoy the most productive careers when they have access to the proper combinations of computer hardware and software and other assistive technology (Scadden,1991)."

Chemists who are vision-impaired use an assorted range of assistive technologies to work productively and safely in academia and industry (Woods, 1996). As stated previously (section 2.11.3), both tactile and audio representation methods have disadvantages for presenting chemical equations to vision-impaired students. Chemical equation representation in mark-up format is proposed in this research as a solution to chemical equation accessibility and balancing. To achieve this purpose, an application was developed which includes two sub-modules:

1. Classification of the implicit information in a chemical equation, tagging classified information and representing them in a mark-up format; and

2. Providing algebraic equations related to chemical equation to balance it.

To develop the application to convert the chemical equations to a mark-up format, a Linux Bash Script was used. The Bash script has several text manipulation tools in command mode which are suitable for vision-impaired users. Vision-impaired users can run the application in text mode as a command using TTS without using any visual aspects.

5.11.2 Species Classification and Reactants/Products Extraction

Species in a chemical reaction is a general term used to mean atoms, molecules or ions. A species can contain more than one chemical element. The general format for species is:

$Species = Symbol_{AtomicNumber(PhysicalState)}{}^{Charge}$

The physical state of each reactant or product is represented by:

(l)=liquid

(g)=gas

(s)=solid.

Coefficients are useful for keeping the same number of atoms. Chemical equations in most cases, come as a text line, therefore to convert it to a mark-up format, text processing techniques are used. The application reads a chemical equation as a parameter such as this:

```
Equation=$1
```

To classify an equation the script needs to remove all space character from the chemical equation and convert yield sign to equal sign (—> to =)

```
Equation=$(echo $1 |sed 's/ //g;s/ --->/=/g;s/-//g')
```

Find "=" sign position

```
EqualSign=$(echo $Equation|grep bo "="|sed 's//:.*$//')
```

Separate left and right side using "=" sign position:

```
Reactants=${Eauation:0:$[$EqualSign+1])
Products=${Equation:$EqualSign)
```

Extract Reactants and product using “+” sign position as the separator:

```
Reactant=($(echo Reactants|sed 's/+/ /g')
Product=($(echo Products|sed 's/+/ / g')
NoR=number of Reactants=${#Reactant[@]}
NoP=number of Products=${#Product[@]}
```

In example, $AgNO_3 + Cu = Cu(NO_3)2 + Ag$:
NoR=2
NoP=2
Reactant=AgNO3 and Cu
Products=Cu(NO3)2 and Ag

5.11.3 Chemical Elements Extraction and Symbol Replacement

In terms of text processing of chemical element symbols, they are divided into two categories:

1. Single uppercase character alphabet as it is shown in Table 5.8-Category 1.
2. Double character elements containing one uppercase character followed by a lowercase character as shown in Table 5.9-Category 2.

To extract symbols and replace them with element names, the length of each species in terms of text processing is calculated, for example:H_2 + O = H_2O, the length of the H_2O is 3. The following command is for obtaining the species length.

```
SpeciesLength=$(echo $Species| wc -c )
```

After calculation of length of species, it must be parsed character by character as below pseudo code indicates.

No of elements= 0
for i=1 to SpeciesLength.
If $character_i$ is uppercase & $character_{i+1}$ is lowercase then :
Consider $character_i character_{i+1}$ as an extracted symbol from Category 2
i++2; No of elements++1
Else
If $character_{i+1}$ is uppercase then:
Consider $character_i$ as an extracted symbol from Category 1
i++1; No of elements++1
End If
End If

End For

In example, $AgNO_3 + Cu = Cu(NO_3)2 + Ag$, N and O are from category 1 and Ag and Cu are from category 2.

Consequently extracted symbols can be replaced by element names using information from what is seen in Tables 5.8 and 5.9.

Table 5.8: Category 1 of symbols for chemical elements

	Category 1
SYMBOL	ELEMENT
H	Hydrogen
B	Boron
C	Carbon
N	Nitrogen
O	Oxygen
F	Fluorine
P	Phosphorus
S	Sulphur
K	Potassium
V	Vanadium
Y	Yttrium
I	Iodine
W	Tungsten
U	Uranium

For better understanding of the conceptual meaning of reaction and remembering the elements involved in the reaction, the symbol of the extracted element is replaced with a full name using the List of Periodic Table Elements. The following snippet shows the code for replacing the symbols by the full name from the elements.

```
for((k=0;k<$NumberElements;k++));do sym=$symbol[$k]ElementName[$k]= $(cat
chemicalelements.lib |awk '$1 /'$sym'/print $2'|head -1)done
```

In example, $AgNO_3+Cu = Cu(NO_3)2+Ag$, Ag with Silver, N with Nitrogen, O with Oxygen and Cu with Copper are replaced.

5.11.4 Calculation of Total Quantity of Element at Left and Right Side and Comparison

The quantity of each element appears in two forms:

1. Single quantity N: the number after an element indicates that there are "N" atoms of that particular element in each molecule. For example, the water molecule H_2O has two hydrogen atoms. The quantity comes immediately after the symbol name. This kind of quantity is considered as Q1 in this

Table 5.9: Category 2 of symbols for chemical elements

					Category 2
SYMBOL	ELEMENT	SYMBOL	ELEMENT	SYMBOL	ELEMENT
He	Helium	Cs	Caesium	Es	Einsteinium
Li	Lithium	Ba	Barium	Fm	Fermium
Be	Beryllium	La	Lanthanum	Md	Mendelevium
Ne	Neon	Ce	Cerium	No	Nobelium
Na	Sodium	Pr	Praseodymium	Lr	Lawrencium
Mg	Magnesium	Nd	Neodymium	Rf	Rutherfordium
Al	Aluminium	Pm	Promethium	Db	Dubnium
Si	Silicon	Sm	Samarium	Sg	Seaborgium
Cl	Chlorine	Eu	Europium	Bh	Bohrium
Ar	Argon	Gd	Gadolinium	Hs	Hassium
Ca	Calcium	Tb	Terbium	Mt	Meitnerium
Sc	Scandium	Dy	Dysprosium	Ds	Darmstadtium
Ti	Titanium	Ho	Holmium	Rg	Roentgenium
Cr	Chromium	Er	Erbium	Cn	Copernicium
Mn	Manganese	Tm	Thulium		
Fe	Iron	Yb	Ytterbium		
Co	Cobalt	Lu	Lutetium		
Ni	Nickel	Hf	Hafnium		
Cu	Copper	Ta	Tantalum		
Zn	Zinc	Re	Rhenium		
Ga	Gallium	Os	Osmium		
Ge	Germanium	Ir	Iridium		
As	Arsenic	Pt	Platinum		
Se	Selenium	Au	Gold		
Br	Bromine	Hg	Mercury		
Kr	Krypton	Tl	Thallium		
Rb	Rubidium	Pb	Lead		
Sr	Strontium	Bi	Bismuth		
Zr	Zirconium	Po	Polonium		
Nb	Niobium	At	Astatine		
Mo	Molybdenum	Rn	Radon		
Tc	Technetium	Fr	Francium		
Ru	Ruthenium	Ra	Radium		
Rh	Rhodium	Ac	Actinium		
Pd	Palladium	Th	Thorium		
Ag	Silver	Pa	Protactinium		
Cd	Cadmium	Np	Neptunium		
In	Indium	Pu	Plutonium		
Sn	Tin	Am	Americium		
Sb	Antimony	Cm	Curium		
Te	Tellurium	Bk	Berkelium		
Xe	Xenon	Cf	Californium		

research. Q1 only belongs to the element came before it. It is worth noting that if there is only one atom of an element in a molecule it is not written. For example: $H_2 + O = H_2O$ the quantity of Oxygen is 1.

2. Grouped quantity. Some formulae have parentheses and place a group of elements within parentheses '()'. The number comes immediately after the ')' parenthesis is a grouped quantity. This quantity is called Q2 by this research. This equation, $(NH_4)3PO_4 + Pb(NO_3)4 = Pb_3(PO_4)4 + NH_4NO_3$includes grouped quantity. Number 3 (in the first reactant) which comes after the closed bracket, belongs to all the elements that are enclosed within the '()'. It means the quantity of N is 3x1=3 and the quantity of H is $3x4 = 12$. In this case the total quantity of each element within the parentheses '()' is a product of $Q1$ x $Q2$. This quantity after the parentheses '()' must be considered for all elements surrounded by the '()'. Thus to find the grouped quantity it must:

- Parse forward through the species from left to right until reaching) at position j;
  - Parse back from) position toward start of equation until reaching (at i;
  - Use sub-string to extract part of species inside the bracket from position i and length j–i.;

    ```
    BracketSurrounded=$species:$i:$[$j-$i]
    ```

  - Use previous part to find out elements and their quantity inside the bracket (Q1);
  - Obtain Q2 and calculate Q1xQ2 for all elements inside the parentheses.As it is shown in the following snippet Q2 is the first number which comes after);

    ```
    Q2 or BracketRelatedQuantity= $(echo $species:$[$j+1]|grep -o
    '[0-9]* '|head -1)
    ```

The following example shows the quantity of each element inside the parentheses after finding Q2.
$(Element1Q11Element2Element3Q13)Q2$
$Elements1 : Q11$ x $Q2$
$Elements2 : Q2$
$Elements3 : Q13$ x $Q2$

5.11.5 Insert Unknown Coefficients Before Species Including Reactants and Products

Coefficients are the numbers in front of each species. They have a very important meaning. A unique unknown coefficient for each reactant or product is considered as shown below:

$a(NH_4)3PO_4 + bPb(NO_3)4 = cPb_3(PO_4)4 + dNH_4NO_3$

The following snippet shows the coefficients inserting process to species.

- Finding reactants in equation before equal sign

```
alphabet="abcdefghiklmnopqrstuwxyz"
reactant=($(echo $equation|sed 's/=.*//g'|sed 's/+/ /g'))
```

- Finding products in equation after equal sign

```
product =($(echo $1|sed 's/.*=//g|sed 's/+/ /g'))
```

- Finding the number of reactants and the number of products

```
No_Reactant=${#reactant[@]}
No_Products=${#product[@]}
```

- Assign alphabet letters to reactants one by one:

```
for ((r=0 ;r<$No_Reactant;r++));do
CoeffReactant[$r]=${alphabet:$r:0}${reacatnt[$r]}
done
```

- Assign the alphabet letters to products one by one:

```
for ((p=0 ;p<$No_Product;p++));do
CoeffProduct[$p]=${alphabet :$[$p+No_Reactant ]:2}${product[$p]}
done
```

5.11.6 Defining Algebraic Equations to Obtain Coefficients and Balance Chemical Equation

If the number of participating elements in the reaction is considered "n" that means there are "n" algebraic equations related to the chemical equation. These equations are extracted by performing following steps for all participated elements in reaction:

1. Consider a specific element;
2. Remove all species which do not contain the specific element at the left and right;
3. Keep all species containing the specified element at the left and right;
4. For remaining species obtain the mathematical production of the quantity of elements in species by the coefficient and replace species with obtained mathematical production

Table 5.8 indicates the algebraic equation for the example from the previous section $a(NH_4)3PO_4 + bPb(NO_3)4 = cPb_3(PO_4)4 + dNH_4NO_3$. As this equation has five elements, five algebraic equations were generated with 4 unknown coefficients.

Table 5.10: Algebraic equations

Element	Related algebraic equation
N	3a+4b=2d
H	12=4d
O	4a+12b=16c+3d
P	a=4c
Pb	b=3c

The values for all coefficients are obtained by solving extracted mathematical equations.

5.11.7 Tagging Classified Information and Markup Format Generation

Table 5.9 illustrates chemical equation classified information and their associated meaningful tags in mark-up format.

Table 5.11: Mark-up Format Tags

Level	Information	Tags
2	Species	<Species ></Species >
1	Products	<Products ></Products >
1	Reactants	<Reactants></Reactants>
3	Symbol of Elements	<Elements></Elements>
3	Full name of Elements	<name></name>
3	Quantity at left	<Q_Left></Q_Left>
3	Quantity at right	<Q_Right></Q_right>
2	Coefficients	<Coefficients></Coefficients

The following snippet shows using recommended tags by Table 5-9 to generate mark-up format

```
<!DOCTYPE html>
<html>
<body> Unbalanced chemical equation:
AgNO3 + Cu = Cu(NO3)2 + Ag
<h1><reactant>reactants</reactant>
<h2><species>AgNO3</species>
<h3><element>Ag</element>
<name>Silver</name>
<left>1</left>
<element>N</element>
<name>Nitrogen</name>
<left>1</left>
<element>O< /element>
<name>Oxygen</name>
<left>3</left></h3></h2>
<h2><species>Cu</species>
<h3><element>Cu</element>
</name>Copper</name>
<left>1</left>
</h3></h2> </h1>
<h1><product>product </product>
<h2><species>Cu(NO3)2</species>
<h3> <element>Cu</element>
<name><copper</name>
<right>1</right>
<element>N</element>
<name>Nitrogen</name>
<right>2</right>
<element>O</element>
<name>Oxygen</name>
```

```
<right>6</right></h3></h2>
<h2><species>Ag</species>
<h3><element>Ag</element>
<name>Silver</name>
<right>1</right>
</h3> </h2> </h1>
</body>
</html>
```

Over 100 simple, intermediate and complicated unbalanced chemical equation were applied to the developed application in this research and equivalent description for these equations automatically were provided. Three simple, intermediate and complicated samples and their results are shown in Figure 5.39.

Equation with Coefficients	Reactants	Products	Elements	L	R	Algebraic Equation
a Fe + b Cl2 = c FeCl3	Fe Cl2	FeCl3	Cl= Chlorine Fe= Iron	2 1	3 1	2b=3c a=c
a KMnO4 + b HCl = c KCl + d MnCl2 + e H2O + f Cl2	KMnO4	KCl MnCl2 H2O	Cl=Chlorine H=Hydrogen K=Potassium Mn=Manganese O=Oxygen	1 1 1 1 4	5 2 1 1 1	b=c+2f+2d b=2e a=c a=d 4a=e
a K4Fe(CN)6 + b H2SO4 + c H2O = d K2SO4 + e FeSO4 + f (NH4)2SO4 + g CO	K4Fe(CN)6 H2SO4 H2O	K2SO4 FeSO4 (NH4)2 SO4	C=Carbon Fe=Iron H=Hydrogen K=Potassium N=Nitrogen O=Oxygen S=Sulphur	6 1 4 4 6 5 1	1 1 8 2 2 17 4	6a=g a=e 2b+2c=8f 4a=2d 6a=2f 4b+c=4d+4e+4f+g b=d+e+f

Figure 5.38: Algebraic finding to balance three samples

The codes to convert chemical equation to alternative text descriptive are in section Appendix B. Six samples of unbalanced chemical equation are available in Section 7.9.

5.12 Summary

This chapter described the methods to develop various modules of the CRS. PDF layout analysis based on segmentation layers is method used in this research to keep reading order and PDF components classification. The PDF classified components by this method are text, image, table and mathematical expressions.

Block segmentation preserved correct reading order. Text/non-Text segmentation extracted image from segmented block. Line segmentation and word segmentation supported global line labelling and mathematical expressions extraction. RCE segmented mathematical expressions to primitive single components horizontally and vertically. Cell segmentation represented table cells by rows, columns or both. The non-alphanumeric mathematical symbols were recognised by kNN and binary vectors. Mathematical symbols layout analysis was considered to identify symbols relationships in neighborhood and find the symbol role in expressions. MATHSPEAK is a developed module to prevent or reduce ambiguity caused by Amsmath mathematical expressions to text description conversion. MathGraphReader extracted all prominent information and data about a mathematical graph, collected them as a test description to convey to vision-impaired student using TTS. Conversion of chemical equation to mark-up format was an approach to assist vision-impaired students for chemical equation balancing.

Chapter 6

Implementation of Hardware Platform

6.1 Introduction

This chapter describes the research and development to implement the initial prototype for DAISY Player using FPGA and design, implement and test different hardware platforms for the CRS. Investigation of different Input/Output devices to find variously accessible user interactions to cope with variety of disabilities interface and provide user communication with system, are explained.

6.2 FPGA Prototype

6.2.1 FPGA Prototype Requirements

The initial DAISY Player was implemented by using the Spartan3E Starter kit board by Digilent hardware platform (Nazemi et al., 2011). The basic requirements for this implementation were:

- A Linux host computer that enables FTP and trivial file transfer protocol (TFTP) services;
- Petalinux as tool chain to port uClinux OS to Microblaze; and
- Serial communication program or terminal emulator such as minicom or Kermit to communicate with the MicroBlaze.

The following steps were taken to build the DTB player:

- The application was implemented using C++ and ported into the kernel image as user-application

- The hardware platform was built using Xilinx's EDK environment (Xilinx, 2006)
- The Petalinux requirements (copy-autoconfig)
- The uClinux kernel

The player application was built using C++ to take advantage of existing Linux libraries and portability across various possible platforms.

6.2.2 FPGA DTB Books Player Functionality

A computer supplies the media files to the DTB player. Alternatively, a Media Access Controller (MAC) coupled to an on-board physical layer network chip supplies a standard Ethernet connection to receive DAISY books over an FTP connection. The DTB player imports book files over the FTP channel recognising the relevant standard or format. The DAISY standard supports three types of books:

1. Full-text /Full-audio;
2. Audio only; and
3. Text only.

The first two types, contain a collection of MP3 files which are organised according to different navigation levels of the books. These files are then played using a conventional MP3 player program. For text only Daisy books and ePUB books, text to speech (TTS) is needed. In this project, hardware TTS was used to simplify the processing and reduce the computational load on the Microblaze controller. The DTB player has the following playing functions: pause, stop and navigate by chapter, section, subsection or paragraph. The completed program was compiled in uClinux with the microblaze-uclinux-g++ compiler. The DAISY software program was executed in the uClinux operating system.

The Most critical issue encountered in creating a software application under uClinux with MicroBlaze was the lack of a Memory Management Unit (MMU). Without a MMU, MicroBlaze is limited to operating systems with a simplified protection and a virtual memory-model. Hence segmentation faults or bus errors occur when the CPU attempts to access a location in memory that does not exist (Hollabaugh, 2002).

6.2.3 FPGA DTB Player Hardware Components

The Minimum requirements to build a uClinux hardware platform within this project were:

- A MicroBlaze soft-core processor;

- A Timer Interrupt controller;
- 32 MB of external DDRAM for loading and running the DTB Player application;
- 64 MB of flash memory for storing the kernel image of the OS, DTB Player and device drivers;
- An Ethernet MAC - A Media Access Controller coupled to an on-board physical layer network chip to supply a standard Ethernet connection to receive DAISY or EPUB books as input data over the FTP protocol;
- RC8660 DOUBLETALK module is connected to the board via a secondary RS232 serial port. This module receives DAISY text files and "reads" them out loud by means of sophisticated text to speech algorithms (RC SYSTEMS, 2005);
- Two Universal Asynchronous Receiver Transmitter (UART) channels DCE-DTB Player for personal computers, DTE: connection between TTS module and MicroBlaze;
- Digilent's PMODs DA2 and AMP1 (Digilint, 2008) to play audio files, PmodDA2 digital to analogue converter (DAC) and PmodAMP1 (audio amplifier); and
- Push button user interface to communicate user with the DTB player. Push buttons and DIP Switches provide the user interface. These devices are accessed using the XIo-In32(Button-Base-Address) function. Table 6.1 shows the return values of this function and how they are used by the DTB software application.

Table 6.1: Push button values for different functions

XIo-In32 (Button-Base-Address)	Button	Navigation function
0x8	Left	Last same level
0x4	Right	Next same level
0x2	Up	Upper level
0x6	Down	Lower level
0xC	RIGHT+LEFT	Play
0xA	LEFT+UP	Exit

6.2.4 FPGA DTB Player

Two external modules to the FPGA, PmodDA2 and PmodAMP1 by Digilent generate the audio. The DAC was incorporated into the hardware platform using EDK's Create or Import Peripheral Wizard. The PLB bus was used to connect

DAC to MicroBlaze. The wizard generates a generic VHDL skeleton to incorporate external peripherals into the MicroBlaze-based embedded systems. VHDL codes to control the DAC are user_logic.vhd and dac.vhd.

These two VHDL files had to be modified to complete the interface between MicroBlaze and the DAC. The DAC IP core output signals were declared as External Ports and the User's Constraints File (UCF) was modified to establish the connection between FPGA pins and DAC ports (Xilinx, 2006). MicroBlaze controls the DAC with two signals, START and DONE. START initiates a digital-to-analogue conversion. When the conversion is completed, the DAC asserts DONE to signal MicroBlaze that a new conversion may be started. The finite state machine coordinates this signal exchange. The description of this controller was added to the dac.vhd file. Details of the connection between the DAC and the PLB bus were added to the user_logic.vhd file. Figure 6.1 shows finite state machine for PmodDA2.

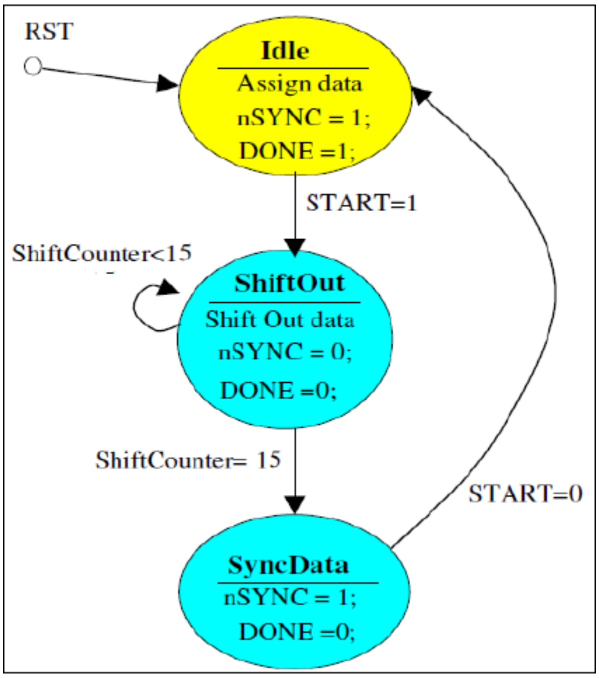


Figure 6.1: Finite state machine for PmodDA2

The PmodAMP1 is a speaker/headphone amplifier that amplifies low power audio signals to drive either stereo headphones or a monophonic speaker. The PmodAMP1 directly interfaces with the PmodDA2 module.

In the process of creating the embedded system using EDK's wizard, the Instruction Cache and Data Cache have to be set to at least 2KB. Once the hardware platform has been created using EDK's wizard, the system.mss file must be modified to set the parameter OS_NAME to Petalinux. Then the option Add Software Application Project must be set to create FS-Boot application. In this application the compiler's optimization level must be set to Size Optimized.

The FS-Boot code, including source files and header files, is part of the PETALINUX package. FS-Boot is a simple boot loader developed by PetaLogix, intended to serve as the primary bootstrap mechanism when the MicroBlaze CPU first boots (Pfefferl, 2008). FS-Boot's primary purpose is to bootstrap the main system boot loader from flash memory. FS-Boot has features such as small code footprint to minimize on-chip BRAM resource usage, serial download for main boot loader (U-Boot) SREC images and auto-boot capability. Once the create FS-Boot application option has been selected, the bit stream and libraries can be generated. The resulting configuration file (.bit) must be downloaded to the FPGA to test the system.

When EDK builds a hardware project, the PetaLinux BSP kconfig.in file for 2.6 kernel is also created and must be copied across to the selected PetaLinux platform. The kernel automatically detects the software project installation and installs the corresponding hardware AutoConfig files. PetaLinux AutoConfig framework allows the hardware configurations to be propagated to the boot loader and Linux kernel configurations. The uClinux OS was installed on flash memory and configured to the following settings:

- Kernel: 2.6;
- File system: CRAMFS;
- Vendor: Xilinx;
- Platform: Spartan3E; and
- Network Protocols: FTP,TFTP.

After downloading the download.bit file from the EDK via the JTAG cable, the SPARTAN3E board was connected to the Linux machine using one of the RS232 channels. Before the connection is established, a Kermit terminal emulator must be running on the Linux machine. This step is necessary for monitoring OS loading on MicroBlaze. When the FS-Boot application runs, the FS-Boot prompt must be interrupted with the character 's' to select the SREC image download option. This option loads the u-boot.srec file to RAM via the RS232 cable. After this, a second stage of bootloader starts running.

When the loading process is completed, the prompt u-boot> will appear on the Kermit terminal. The file u-boot-s.bin must be saved in the flash base address and image.bin in the base address + 0x8000.

According to the device utilisation report generated by EDK, the complete design occupies less than 53% of the FPGA programmable logic. Implementation and testing were carried out using Xilinx's ISE suite, version 10.1. To test the audio sub-system, a PCM file was transferred to uClinux over FTP. Samples were sent to the DAC at the sample rate stored in the file. The sample rate was extracted from the PCM file using the function XIo_Out32 (XPAR_DAC_BASEADDR, samples). Several music MP3 files were reproduced using this method.

To test the TTS synthesis (RC8660 module), DAISY and EPUB files were transferred from the Linux machine to the DTB player using FTP over Ethernet. The player stored the file's text navigation block into an array and then sent it character by character to the TTS synthesiser. The transfer took place over the second RS232 channel.

The function used to transmit characters is XUartLite_SendByte(DCE_UART_BASE_ADDRESS, buffer[i]). Text files were correctly reproduced in spoken English. Once all the DTB player's hardware sub-systems were tested, the FPGA's configuration file was converted to the .mcs format and permanently stored in the SPI flash memory. In the result, the DTB player gets loaded into the FPGA after power-up and the personal computer is no longer necessary to download the design, and it is only used as an FTP server for book transfers.

Figure 6.2 illustrates an overview of components of DTB Player using FPGA.

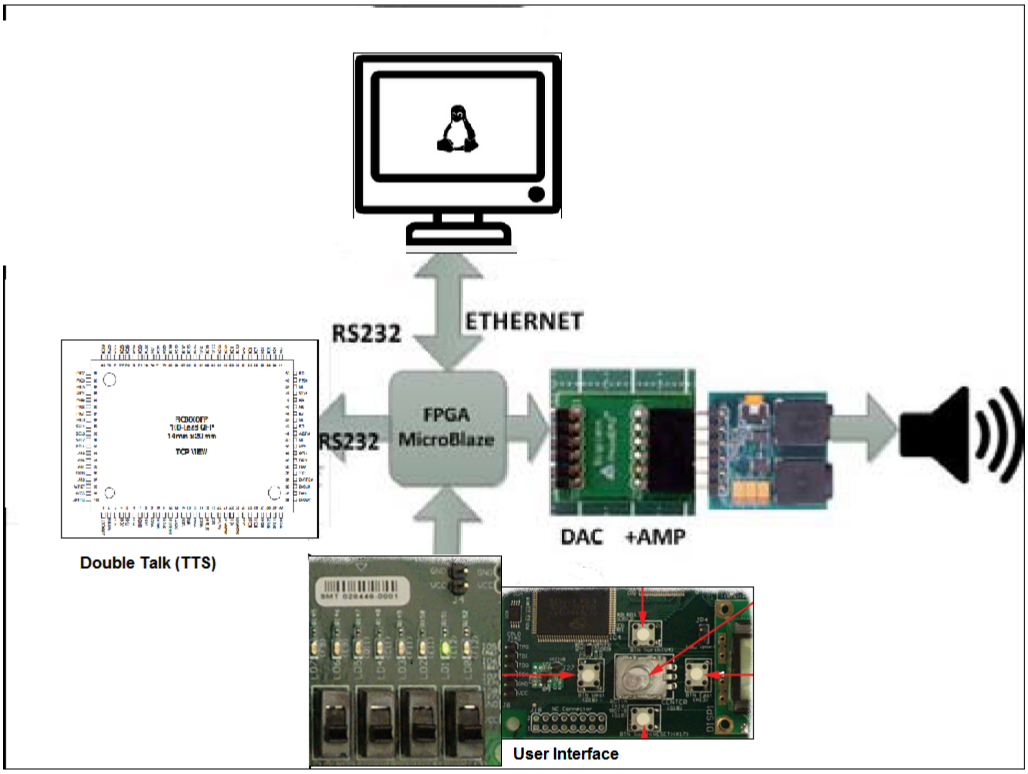


Figure 6.2: FPGA-based embedded system for DTB player

6.3 Embedded Platform for Reading System

The initial prototype for the embedded CRS was implemented using the Beagle Board as the hardware platform. The Beagle Board is an open source board and utilises the OMAP3530 (ARM-CORTEX-A8). By using the Beagle Board the redevelopment of the layout has been simplified. There are several accessories within the Beagle Board development environment which were deemed unnecessary for the CRS. Hence, they can be removed to cut cost to the user. Many of the required hardware components for Reading System in Beagle Board contain unused capabilities that increase cost and can be replaced with simpler hardware to decrease power consumption and keep costs to minimum. To developed CRS software modules machine learning and image processing techniques were utilized and these techniques need powerful processor to be run efficiently

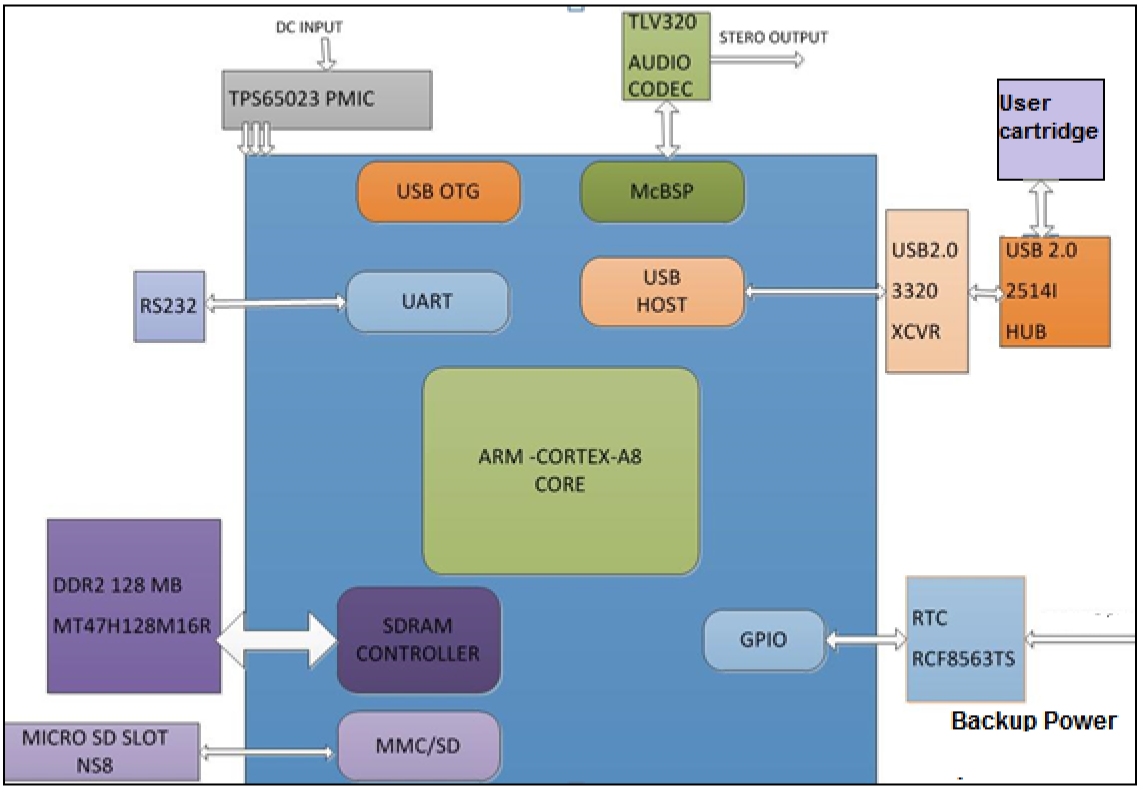


Figure 6.3: Block diagram of customized board for reading systems

As it is shown in Figure 6.3 main components are:

- Microprocessor ARM-CORTEX-A8 core. AM3517 or ARM-CORTEX -A9 Quad core processor is suitable for designing the CRS.
- Multi-port USB hosts to connect USB storage for file transferring.
- Multichannel transmit/receive buffer to support TLV320 input/output audio codec
- Removable media interface to boot from mini SDRAM.
- 512 MB up to 2GB DDR2 data rate Memory and controller
- Standard 3.5mm headphone jack and microphone jack and Audio Codec controller

- Storage Card Slot and controller
- 5V/2A Power
- USB 2.0 Host (High speed standard A type) and USB Host for controller
- 10/100Mbps Ethernet with RJ-45 Jack+ controller

6.4 User Interaction Methods

6.4.1 Introduction

The user Interaction section explains about how user communicates with the CRS. User interaction contains Input/output peripherals which are connected to the CRS to provide user communication opportunity. The input interfaces allow users to send a request for specific function and control operations . The output interfaces are designed to effectively present processing results of user selected operations.This section presents speech recognition, key pad and joystick as input and Text-to-Speech and Braille terminal as output interface devices.

6.4.2 Speech Recognition

Speech Recognition was implemented as a method for user interaction with the CRS to address the issue of other disabilities communication with system. It is based on:

- Sending user commands to the CRS through audio input port using microphone.
- Processing speech user commands by system utilizing speech recogniser software package

Julius is a high-performance, two-pass large vocabulary continuous speech recognition (LVCSR) decoder software (Lee, 2010) that was considered as an interaction method for the CRS. Julius is based on word N-gram and context-dependent Hidden Markov Model (HMM). Julius is modularized to be independent from model structures and various HMM types are supported. It can be used to build a kind of voice command system of small vocabulary, or to perform various spoken dialog system tasks.

Julius voice recognition engine uses hand-designed Deterministic Finite Automaton (DFA) grammar as a language model instead of an acoustic model. In the language model a grammar consists of two files: one is a 'grammar' file that describes sentence structures in a BNF (Backup\ Normal Form) style, the another is a 'voca' file that defines words with its pronunciations for each category such as a CRS commands (Heine, 2011). The mkdfa.pl script compiles the Julius format grammar (.voca) to a deterministic finite automaton file (.dfa) and a dictionary file (.dict) respectively (Lee, 2010).

The CRS basic functions applied and tested by Julius are shown in Table 6.2.

Table 6.2: Basic Functions in VOCA file in Voice Recognition System for CRS

Command	Definition
%ID	DO d uw
% COMMAND_PLAY	PLAY p l ey
%COMMAND_NEXT	NEXTn eh k s t
% COMMAND_BACK	BACKb ae k
% COMMAND_UP	UPah p
% COMMAND_DOWN	DOWNd aw n
%COMMAND_CONTINUE	CONTINUE k ax n t ih n y uw
% COMMAND_PAUSE	PAUSE p ao z
% COMMAND_FINISH	FINISHf ih n ih sh

The following command runs Julius and sends recognition results to stdout

```
padsp ./julius -nostrip -quiet -input mic -C julian.jconf
```

julian.jconf configuration file contains the following information:

```
-dfa CRS.dfa
-v CRS.dict
-h /usr/share/julius-voxforge/acoustic/hmmdefs
-hlist /usr/share/julius-voxforge/acoustic/tiedlist
-penalty1 5.0          # first pass
-penalty2 20.0         # second pass
-iwcd1 max      # assign maximum likelihood of the same context
-gprune safe           # safe pruning, accurate but slow
-b2 200                # beam width on 2nd pass (#words)
-sb 200.0              # score beam envelope threshold
-spmodel "sp"          # HMM model name
-iwsp                  # append a skippable short pause  model
   at all word ends
-iwsppenalty -70.0     # transition penalty for the short pause
   models
-smpFreq 16000         # sampling rate (Hz)
```

CRS.dict is:

```
0       [<s>]   sil
1       [</s>]  sil
2       [COMPUTER]      k ax m p y uw t ax
3       [PLAY]  p l ey
3       [NEXT]  n eh k s t
```

```
3       [PREV]  p r iy v
3       [SHOW]  sh ow
3       [PAUSE] p ao z  [CRS ]  c r s
3       [DO]    d uw
3       [PLAY]  p l ey
3       [NEXT]  n eh k s t
3       [BACK]  b ae k
3       [UP]    ah p
3       [DOWN]  d aw n
3       [CONTINUE]      k ax n t ih n y uw
3       [PAUSE] p ao z
3       [FINISH]        f ih n ih sh
```

6.4.3 Customised Tactile Keypad

Another approach for user interaction for the CRS was designing a tactile keypad. To design a custom keypad, two different methods were considered:

1. Utilizing GPIO and scanning for key events directly from the application processor; and
2. Using a USB interface with a small micro-controller to communicate and monitor keys.

The first approach has the advantage of minimal cost at the expense of programming complexity (having to regularly monitor for keystrokes). Compared with using GPIO, USB keypad has greater cost and simpler integration. Hence, a customized keypad was developed using the second approach with the minimum number of keys.

A USB Keypad is an input user interface attached to an Arduino UNO rev3 daughter board. The Arduino UNO rev3 is used to implement the USB connectivity of keypad. For this purpose, the Arduino needed to be programmed and atmega8u2 must be put into Device Firmware Update (DFU) mode, then the firmware is flashed by running 'make dfu' in the arduino-usb_serial directory (Levi, 2012). Figure 6.4 illustrates PCB schematic of USB keypad.

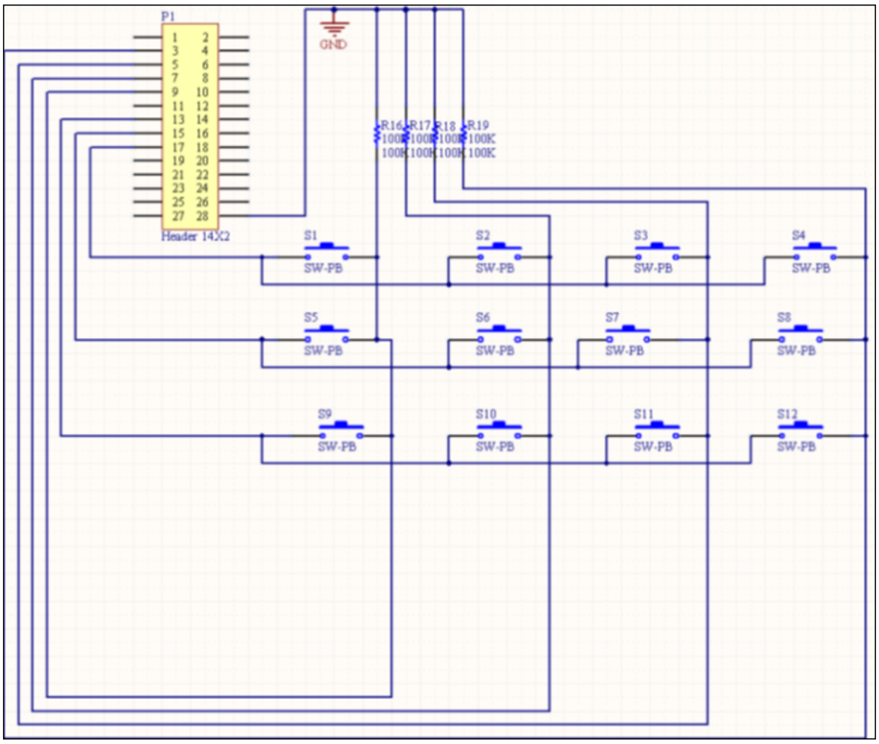
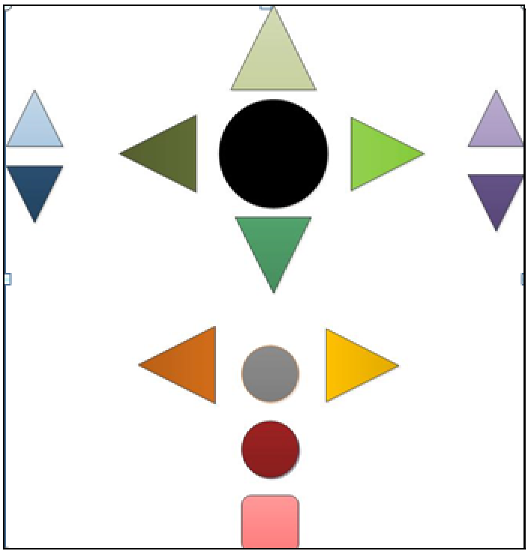


Figure 6.4: Arduino Hex Keypad Schematic

The CRS keypad has several keys to control and interact with the system. Tactile dots allow key identification for users with vision impairments to perform different functions. These functions include: stop, play, pause, fast forward, rewind, increase and decrease volume, increase and decrease speed, bookmark and navigation (4 arrow keys). Some keys in this keypad have been designed to be multifunctional depending on the format of the material being read. For example the two left and right navigation keys are used in mp3 player to cycle through a play list of mp3 files and in DAISY player are used to move the reading position to the previous and next item at the current level (e.g. move from chapter 1 to chapter 2). The keypad contains three pairs of keys to support variable speed playback, accelerated fast forward/rewind and volume levels. When performing some functions, certain keys may be disabled, for example, while an audio file is being played (via audio player not DAISY player), bookmark and navigation level up and down keys are disabled.

Figure 6.5 shows key arrangement in customized keypad for the CRS.



Key	Function
	Play/Pause
	Navigation
	Volume down/Volume up
	Speed down/Speed up
	Forward /Rewind
	Bookmark
	Information
	Exit

Figure 6.5: CRS customised keypad

6.4.4 Users Feedback about USB keypad

Initial testing to evaluate USB keypad usability was undertaken with two users who were interviewed and fell into different categories. User A was a male above the age of 50, who was a common Digital Talking Book (DTB) user due to his severe dyslexia. User B was a female above the age of 50, who was a regular DTB due to blindness. User A is considered a standard user, while user B is considered an advanced user due to the difference in their technological skills and experience. They (Users A, B) made the following recommendations when they were asked for their feedback regarding the customized keypad:

- Buttons shaping should be based on the National Library Service (NLS) player.
- Buttons need to be bigger and should have more space between them.
- Braille labels under buttons should be used, rather than symbols.
- Power switch on the side with a tactile indicator is helpful.
- The device should be able to function even without a cartridge.
- Button arrangement in mobile devices is suitable for designing a custom keypad particularly for navigation purposes (Levi, 2012) as it is shown in Figure 6.6.

1 forward	2 navigation	3 volume up
4 navigation	5 play/pause /resume	6 navigation
7 rewind	8 navigation	9 volume down

Figure 6.6: Mobile keypad arrangement

Figure 6.7 shows NLS Digital Talking Book Player

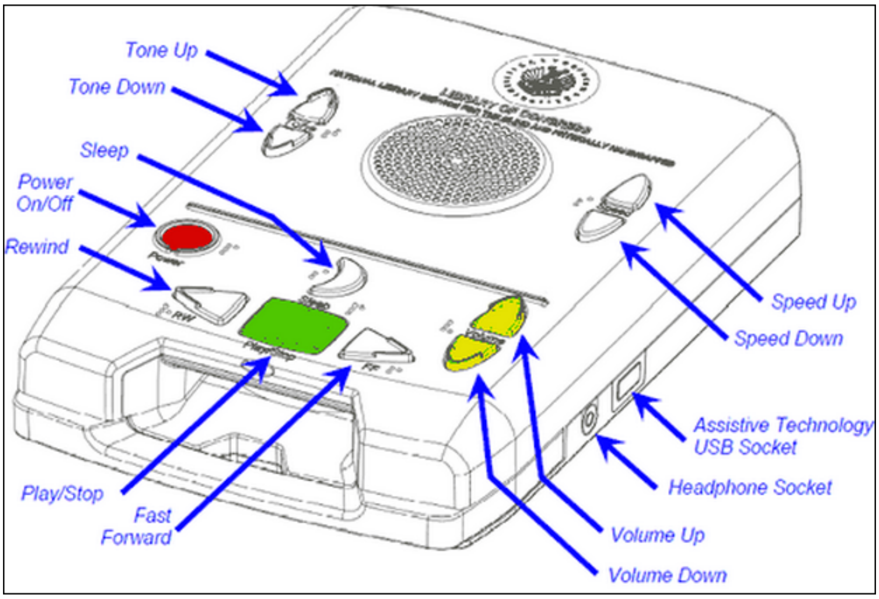


Figure 6.7: NLS Digital Talking Book Player (Maine State Library, 2013)

6.4.5 Joysticks

Using a joystick as an interface can help individuals with upper extremity or fine motor disabilities who have difficulty using a keypad. It enables the user to navigate through DAISY books simply just like keypad users. The joystick system connection detection can be done by checking /dev/input/js0 availability in Linux system. Three axes and four buttons make different combination for basic functions implementation in the CRS as shown in Table 6.3.

These commands are used for joystick installation:

```
apt-get install libsdl-dev
apt-get source joystick
jstest -nonblock  /dev/input/js0
jscal -c /dev/input/js0
```

The system checks joystick availability by :

```
if [ -f  /dev/input/js0 ];then
echo"Joystick is connected"
./joystick      #C cocdes to identify  which button presses or
    which axes moved
else
echo"Joystick is  not  connected, try later  "
fi
```

Table 6.3 indicates the CRS functions and their assigned axis movement or button when joystick is used.x

Table 6.3: Using joystick as a user interface

Command	Axis/Button Value
NEXT	Axis[0]=32767
LAST	Axis[0]=-32767
DOWN	Axis[1]=32767
UP	Axis[1]=-32767
EXIT	Button[0]=1
PAUSE	Button[1]=1
REWIND	Button[2]=1
FORWARD	Button[3]=1

The script joystick.c by accessing the buttons or axes values allows the system to recognise user request or user commands and run an appropriate function based on Table 6.3. Axes movement is checked for navigation purpose and button pressed is monitored for pause, resume, forward, rewind and stop functions. The code for joystick driving and monitoring is available in Appendix B.

6.4.6 Text to Speech (TTS)

Users achieve Braille reading rates around 100 words per minute. The average visual reading rate is around 250 words per minute and around 200 words per minute of synthetic speech. Experienced synthetic speech users often may prefer higher rates (Murray, 2008). Most applications in the CRS generate plain text in their final steps. TTS was implemented and utilised to present the CRS output result to user as an output interface noting its high rate conveying data.

TTS can be implemented by two methods:

- Software TTS such as espeak or Flite (Festival)
- Hardware TTS such as RC8660 Doubletalk

Most of the multilingual TTS systems are software applications that allow people with visual impairments or reading disabilities to listen to written material using a computer (Carlson et al.,1982). Several hardware TTS products are commercially available, such as Doubletalk, DECtalk and Dolphin, but are either not affordable or not multilingual. In comparing these two methods the hardware TTS produces higher quality speech and playback control commands.

In this research, the Doubletalk chipset was selected because it could be easily integrated into the system. It contains voice control parameters such as speed, volume, tone, pitch and expression. The Doubletalk receives text over an RS232 serial port and sends the output audio via its audio out port. The Doubletalk chipset implements text-to-speech synthesis with full dynamic control of the voice characteristics. The Doubletalk uses as an integrated TTS processor that incorporates technology based on a unique voice concatenation technique using 'real human' voice samples (RC SYSTEMS, 2005).

The Doubletalk supports both Code Page 437 (covers United States and Western Europe) and ISO 8859-1/ANSI character sets (covers Americas, Western Europe, Oceania and much of Africa Standard Romanisation of East-Asian languages). Also both of these character sets are suitable for representing Latin scripts. Enabling the Doubletalk to speak in non-English languages requires a pronunciation guide for each non Latin/Romanised language to transcribe the pronunciation rules into exception forms. Since alphabet sets other than Latin could not be recognised by Doubletalk, it is necessary to generate text using Latin characters. Therefore, an accurate Romanization or transliteration system must be provided to generate an unambiguous one-to-one mapping between Latin characters in the UNICODE range and non-Latin characters. The Romanization system must be able to preserve both the pronunciation and the written forms of the text. Non-Latin text transliteration requires a reliable system to preserve the orthographic as well as the phonological features of the language (Sagot & Walther, 2009).

Doubletalk with RC86L 60F1L chipset, accepting ASCII input, a concatenated chipset with two minutes of recording memory is available with real-time control over volume, pitch and speed. This real-time functionality is essential for the product's success as a helpful interaction device (Plumpe & Meredit, 1998). RC8660 contains two commands to stop and start generating speech from text. These commands have been used to implement control playback function such as pause, resume, forward and rewind, calculating the total length of the text and spent time before the forward or rewind command help to simulate forward/rewind using Stop function. The stop and start commands are:

```
Stop command:echo -en '\x01\x10' > /dev/ttyUSB0
Start command:echo -en '\x01\x12' > /dev/ttyUSB0
```

Speech rate calculation of RC8660 has been done by sampling and averaging as the following formulae indicate:

```
Speech-Rate=number of letters/time
Elapsed-Letters=Speech- Rate * Elapsed-time
Elapsed-time=Current time -Starting time
```

Control playback functions can be implemented as follows:

- Pause: by sending stop command when pause key is pressed.:
- Elapsed letters when pause key is pressed = speech rate * elapsed time
- Resume: by starting TTS task from elapsed letters when pause key is pressed again
- Forward: by sending stop command when forward key is pressed and starting TTS from elapsed letters when forward key is pressed+(10*speech rate) (it means 10 seconds later)
- Rewind: by sending stop command when rewind key is pressed and starting TTS from: elapsed letters when rewind key is pressed-(10*speech rate)(it means 10 seconds before).

Table 6.4 includes player functions and related commands for RC8660 in Linux.

Table 6.4: TTS control commands

Command	Function
echo -en '\x01+1s\x00' > /dev/ttyUSB0	Speed up
echo -en '\x01+1v\x00' > /dev/ttyUSB0	Volume up
echo -en '\x01+40p\x00' > /dev/ttyUSB0	Pitch up

6.4.7 Braille Terminal

A refreshable Braille display or Braille terminal is an electormechanical device for displaying Braille characters, usually by means of round-tipped pins raised through holes in a flat surface. Vision impaired users, who cannot use a computer monitor, use it to read text output. A tactual computer monitor or Braille display includes rows and columns of rectangular cells. Each cell includes four rows and two columns of movable pins which are felt and read by a vision impaired user. The pins are driven by electromechanical impact drivers and are held in position by resilient elastodynamics (Becker et al., 2004). Speech synthesizers are also commonly used for the same task and a blind user may switch between the two systems or use both at the same time depending on reading circumstances.

In order to represent text file in Braille, an option was implemented for the CRS to communicate with Braille display as an auxiliary output. The user can send the request to the CRS by pressing a certain key for reading a specific part of plain text in Braille format. Upon receiving this request, the system will interrupt normal reading session, send the requested part of the text to the Braille display and wait to receive an acknowledgment from user to send the next part to Braille display or go back to normal reading. A six-dot Braille cell allows 64 possible combinations of dots. Since in many languages, 64 combinations are not enough to represent letters, numbers and punctuation marks, there are multiple Braille codes. Accuracy and compatibility translation with multiple Braille codes and compatibility with vast majority of Braille displays are critical issues regarding transformation of text to Braille. Mathematical content is detected, segmented, semantic analysed, merged, rendered and finally converted to text description which can be sent to Braille termina. CRS contains a software application that was developed to allow the interaction of the different modules of CRS with Braille displayslThis application is responsible for:

- Accurately translating an input string to Braille;
- Compatible translation with multiple Braille codes; and
- Sending the translation to connected Braille displays.

The following open source packages were used to develop this part of the CRS:

- libbrlapi-dev;
- brltty, and
- liblouis-dev.

BRLTTY is a daemon which provides access to the Linux console (text mode). It drives the Braille terminal and provides complete screen review functionality.

The following bash script snippet shows how the CRS sends text to Braille terminal:

System keeps starting time of reading session.

```
braille=0
Starting_Time=$(date +"%s")
```

TTS (espeak) reads text file

```
espeak -f text.txt
```

User sends a request for using Braille-Terminal by pressing a key.

TTS stops working.

```
braille=1
killall -v espeak
```

As system receives request, elapsed time is calculated.

```
Request_Time=$(date +"%s")
Elapsed_ Time=$(($Request_Time-$Starting _Time))
```

The number of elapsed characters is calculated assuming TTS transfer rate is equal to 15 character/second.

```
Elapsed_characters=$(($Elapsed_ Time*15))
```

One hundred characters cuts using sub-string from requesting position, and sends for preprocessing to generate input.

```
echo  ${Sring:$num:100}|tr -d '\n' > str.txt
cat str srt.txt|sed 's/\//\\/g'|sed  's/x0020/   /g' | sed
's/\\/ /g' >input.txt    #generate plain text data
```

Plain text input file is sent to ./braille executable file which is compiled with:

```
gcc brailler.c - llouis -lbrlapi -w -o braille
```

Figure 6.8 shows Beagle Board as the CRS initial prototype connected to USB cartridge as source of data, keypad as user interface, Braille terminal as tactile output device and speaker as audio output device.

Figure 6.8: Reading system connected to Braille terminal

6.5 Summary

This chapter described the suitable hardware platform to implement CRS. Using a FPGA for this purpose generated "segmentation fault" error in some cases. The Beagle Board as an open source single board computer needed customization and modification to cut extra cost in final device price.

The CRS user interfaces, the input and output devices that assist user to communicate and interact with the system were also discussed.

A USB Keyboard was considered with minimum number of buttons as an input device. Additionally, joystick compatibility with the CRS was tested. It was shown that joystick can be used as an input interface for people with motor disability. Since most of the CRS modules generate text, to convey final result to user, TTS was used to convert text format to audio format. The TTS result was sent to audio output port. Alternatively, Braille Terminal could be used as an output interface to convert text to Braille to represent to user.

Chapter 7

Testing for Evaluation and Usability

7.1 Introduction

The Modules explained in Chapter 5 were applied to the various samples and the results are presented in this chapter. These results clearly demonstrate that the ChartRecognition module for text extraction needs an improved OCR. The other modules were tested and satisfactory evaluated. All data extracted by various modules of CRS, can be collected in the text description, converted to audio or Braille and represented to user utilizing audio or tactile method.

.

7.2 Block Segmentation

Block segmentation module is used for keeping reading order in multiple columns documents, It can also support usability of HVTO and engineering drawing documents.

Figure 7.1- top illustrates an engineering drawing sample and Figure 7-bottom is the result of block segmentation for the sample engineering drawing. Block are identified with blue lines. Each block is sent to OCR individually to retain reading order and meaning.

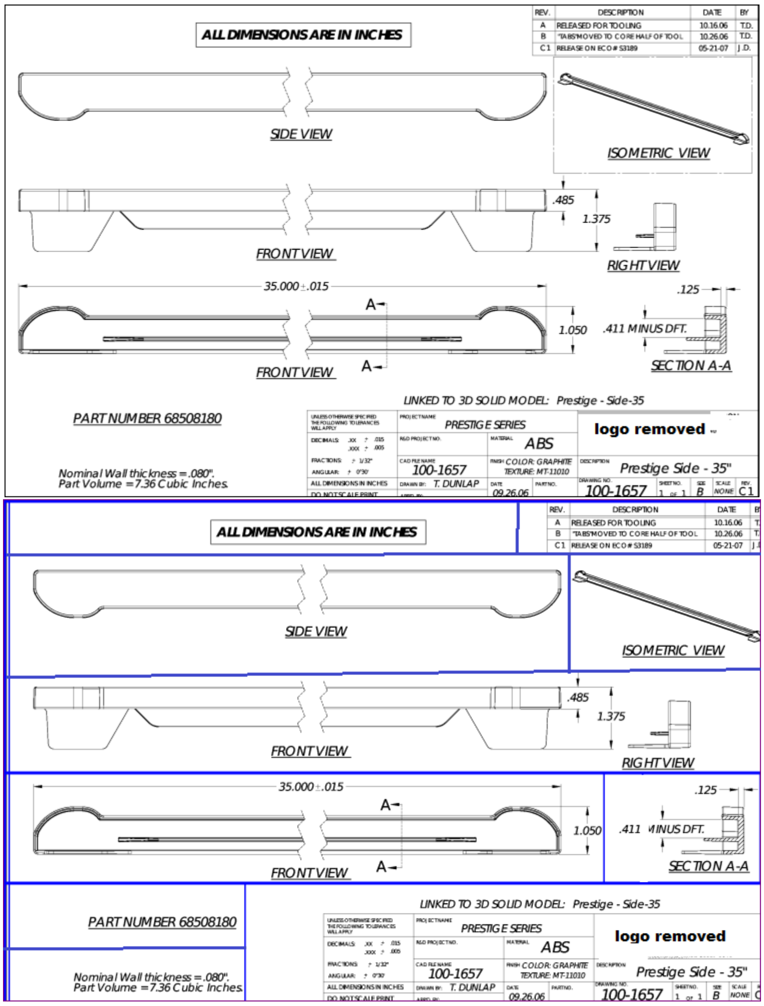


Figure 7.1: Engineering drawing

7.3 Chart Recognition

ChartRecognition results for 23 different images are shown in Figures 7.2, 7.3 and 7.4. These results indicate 91% accuracy for this module.

Figure 7.2 includes 11 pie charts and 2 bar charts which all were recognized as pie chart.

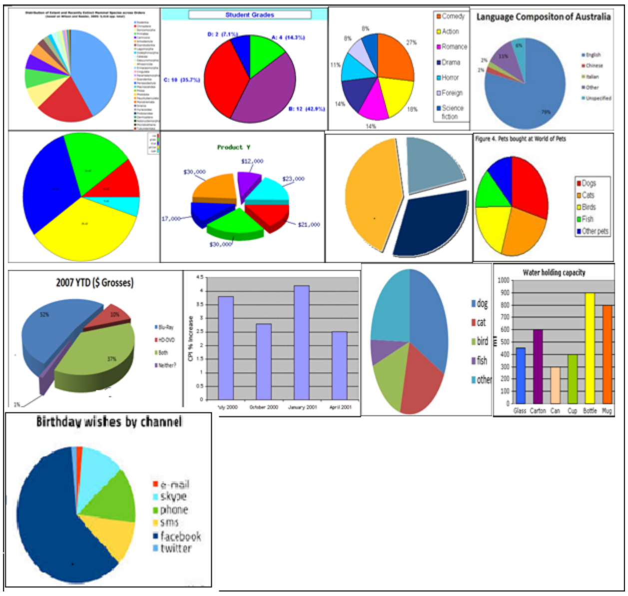


Figure 7.2: Recognised as pie chart

Figure 7.3 shows six samples of bar charts which the ChartRecognition module was applied to and correctly recognised.

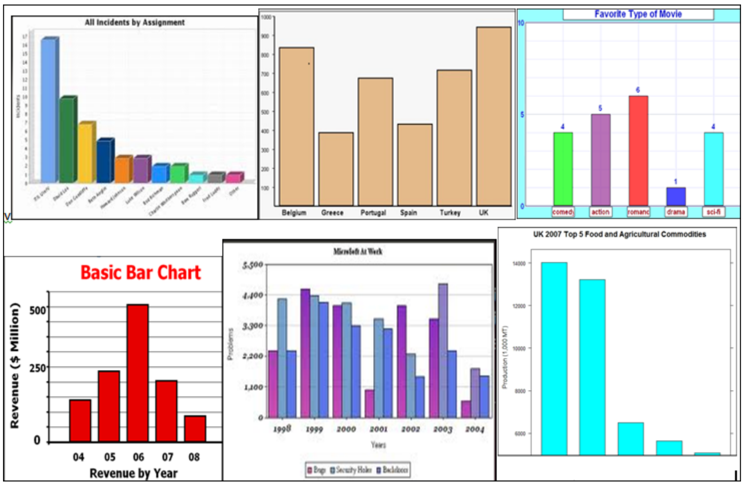


Figure 7.3: Recognised as bar chart

Figure 7.4 shows four samples of line charts which the ChartRecognition module was applied to and correctly recognised.

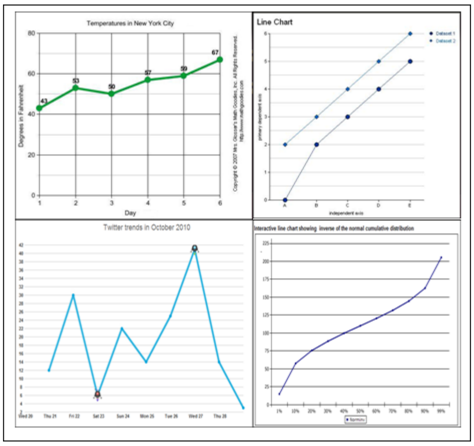


Figure 7.4: Recognised as line chart

Table 7.1 contains values of features for chart recognition of eighteen pie charts (left side) and five bar charts (right side).These table supports figure 5.30.

As stated in Chapter 5 thre features can be extracted from the shape-only image in charts are:

Number of remaining unconnected pieces after process = P

Number-of-Remaining Black Pixels after process= RBP

Deviation-of-Height Diversity= Amount of variation of height of parallel pieces = DHD

As Figure 5.30 illustrates:

If P>0 e&& DHD>0 && RBP=0 then the chart is bar chart.

If P<=1 && RBP>0 then the chart is pie chart.

The Table 7.1 information confirmed Figure 5.30 classification method.

Table 7.1: Chart recognition features

P<=1	RBP>0
1	66727
0	105864
1	27501
0	91140
1	81350
0	91245
0	6448
1	125065
0	31674
1	3158
1	27501
1	55270
1	57331
1	3158
0	62988
0	115157
0	116812
0	66300

P>=2	DHD>0	RBP=0
3	2	0
9	6	0
6	5	0
6	5	0
3	2	0

7.4 BarChartReader

Most errors occurring in this section are based on OCR dependency and lack of appropriate ratio to convert pixels to coordinate values. In the most cases BarChartReader recognised number of bars, two close bars, maximum and minimum bars properly. Figure 7.5 illustrates four different bar charts which BarChartReader extracted maximum/minimum, two close bars and number of bars from image of charts.

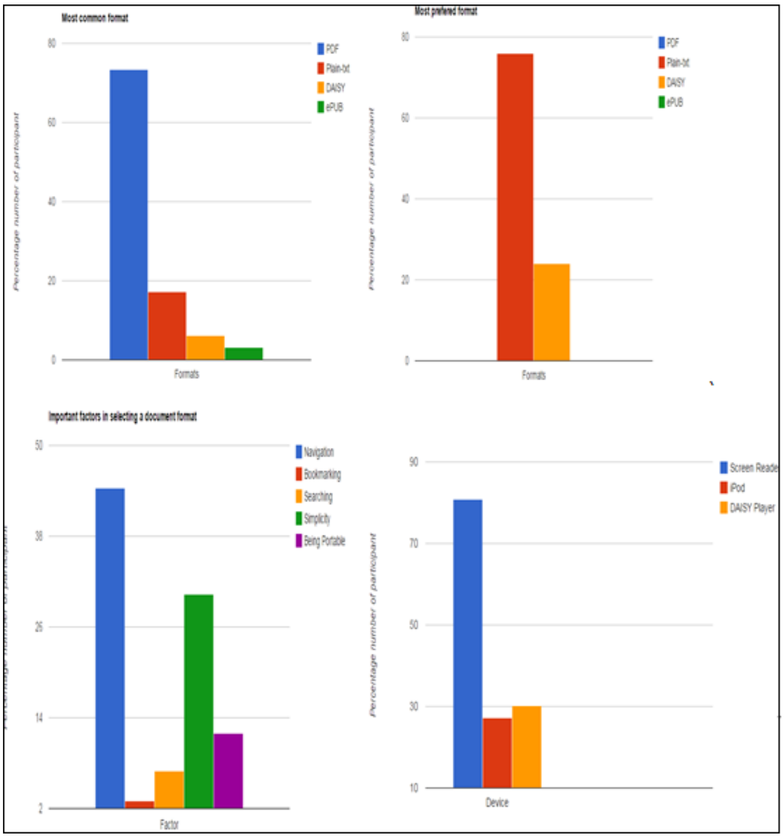


Figure 7.5: Examined bar charts by BarChartReader. Maximum / Minimum Bars, total number of bars were propery recognised.

Figure 7.6 shows two bar charts which BarChartReader was applied to. The reason of spelling or punctuation errors is OCR dependency. BarChartReader extracted following information from the images, collected in the alternative text description and presented to users via TTS or Braille terminal based on user request. Users can receive essential concepts and information about bar chart from this presentation.

The data extracted by BarChartReader processing from left hand side of Figure 7.6 is as follows:

```
This is bar chart contains 5 bars
It is supposed bars height is from 0 to 8 based on ?.
The bar labelled romance is equal 4.
The bar labelled comedy is equal 2.
The bar labelled drama is equal 3.
The bar labelled action is equal 6.6 this bar has maximum value
The bar labelled fiction is equal 0.6. this bar has minimum value
```

The data extracted by BarChartReader from right hand side of Figure 7.6 is as follows:

```
This is bar chart contains 5 bars
```

```
It is supposed bars height is from 0 to 14000 based on Pmaucnan
    (1 â€coo MT)
The bar labelled Cow is equal 12574 this bar has maximum value
The bar labelled wheat is equal 11480.
The bar labelled sugar is equal 2317.
The bar labelled potato is equal 1145.
The bar labelled barely is equal 507. this bar has minimum value
```

.

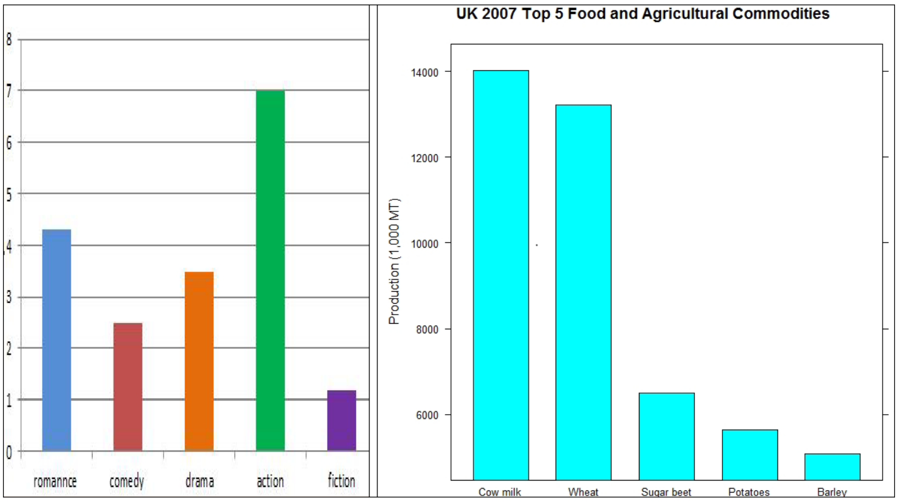


Figure 7.6: Bar chart samples which were examined by BarChartReader

7.5 PieChartReader

As stated in Chapter 5 pie charts can be replaced in most cases by other plots such as the bar chart. The following snippet was used for converting pie chart to bar chart for better processing and presentation.

Pie chart histogram file is obtained and simplified to keep only information about colours and pixels numbers using following command:

```
convert temp.gif -format '%c' histogram:info:|grep -v none|sort -
    rn|sed 's/:.*#/ #/g'|awk '$1>0 {print $1,$2}'>histogram
```

Then total number of pixels in pie is calculated:

```
sum=$(cat histogram|awk '{print $1}'|awk '{sum+=$1}END{print sum
    }')
```

The number of slices is calculated with counting the number of different colors:

```
color_no=$(cat histogram|wc -l)
```

Percentage of each colour is obtained using following command:

```
cat histogram|awk '{printf "%d %s\n", $1*100/'$sum',$2}'>
    new_histogram
```

For each individual colour such as $Color_i$ with $Count_i$ as percentage of $Color_i$ and i is from 1 to number of colours in histogram, a bar is generated with Width=20 and Height=$Count_i$. The following snippet is used for this purpose:

```
cat new_histogram|
for ((i=1;<=$color_no;i++));do
count=$(cat new_histogram|awk 'NR=='$i''{print $1}'
color=$(cat new_histogram|awk 'NR=='$i''{print $2}'
while read count color ; do
convert -size 20x$[$count] xc:$color miff:-
done |
convert - -alpha set -gravity south +append Bar_chart.gif
done
```

Figure 7.7 illustrates four pie charts which PieChartReader was applied to. The total number of slices, percentage of each slice and an alternative equivalent bar chart for each pie chart are represented in this figure.

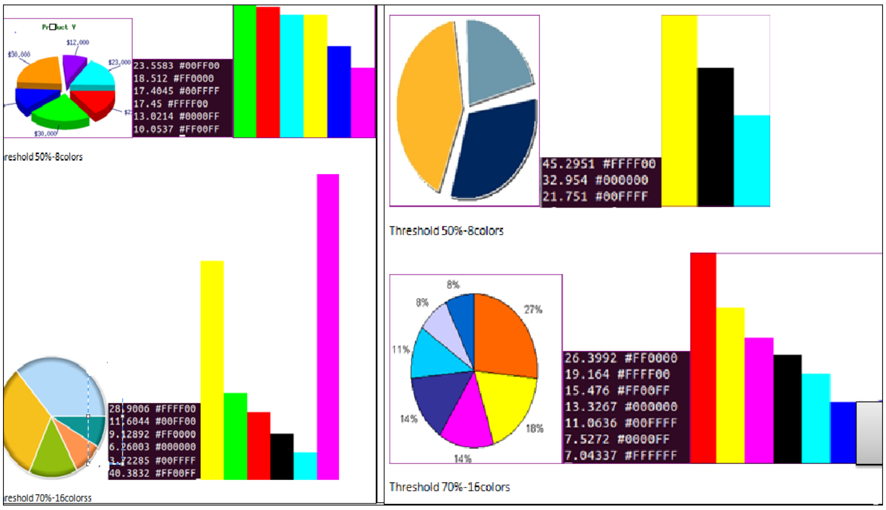


Figure 7.7: Sample of Pie charts,their slices percentage and equivalent bar charts generated by PiechartReader

7.6 LineChartReader

Accuracy of the data extracted by LineChartReader is dependent on OCR functionality. OCR is responsible for extracting essential information from axes such as range of axes and x-tick /y-tick

Figure 7.8 illustrates three line charts which the LineChartReader was applied to. The information extracted by LineChartReader includes horizontal /vertical axes range, number of critical points and the value of dependent variable (y) for each particular independent variable(x) in critical points.

The data extracted by Line Chart Reader from Figure 7.8 (left) is as follows:

```
This line chart contains 7 critical points
It is supposed that horizontal axis is from 2006 to 2013
Vertical axis is from 0 to 4
At Point :
2006 the value is 2. 3
2007 the value is 1.8
2008 tthe value is 0. this is minimum value
2009 the value is 3
2010 the value is 1.6
2011 the value is 3.9 this is maximum value
2012 the value is 2.7
```

The data extracted by LineChartReader from Figure 7.8 (middle) is as follows:

```
This chart contains 6 critical points
It is supposed that horizontal axis is from 0 2001 to 2007
Vertical axis is from zero to 25000
At point
2001 the value is 23137
2002 the value is 21021
2003 the value is 18484
2004 the value is 15577
2005 the value is 11333
2006 the value is3 8313
```

The data extracted by Line Chart Reader from Figure 7.8 (right) is as follows:

```
This chart contains 14 critical points
It is supposed that horizontal axis is from zero to 24
Vertical axis is from zero to 16.
0 the value is 5.31629. this is minimum value 5.31629
5.29496 the value is 9.04792.
9.03597 the value is 11.246.
13.0072 the value is 8.23003.
16.9209 the value is 13.5463.
17.3237 the value is 13.7508
20.8345 the value is14.9776. this is maximum value equal is
   14.9776
21.2374 the value is14.4153.
```

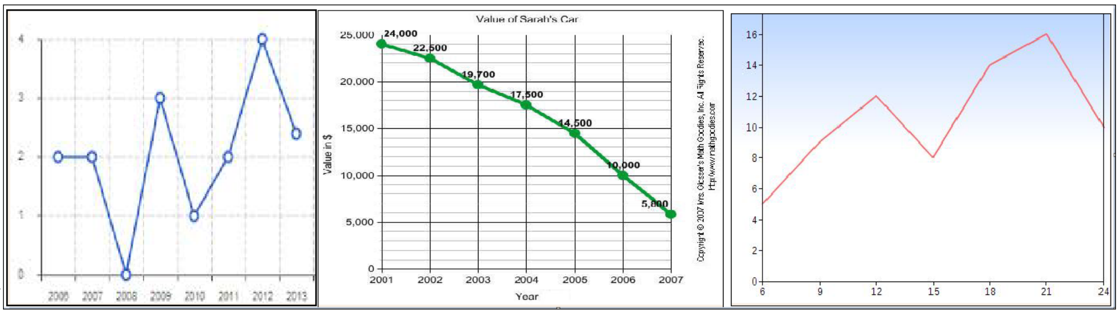


Figure 7.8: Line Chart Samples processed by LineChartReader

7.7 MathGraphReader

Figures 7.9,7.10,7.11 and 7.12 illustrate four mathematical function graphs which MathGraphReader was applied to (left) and their equivalent generated by GNUPLOT (right). This section presents the information that MathGraphReader extracts from mathematical function graphs using digitization and image processing techniques. This information makes text description to convert audio or tactile formats.

Sighted user can extract the following information from the left hand side of Figure 7.9:

```
The minimum is located at (-2.25 ,-4.54)
```

Date extracted by MathGraphReader using right hand side of Figure 7.9 and the function f(x)=x**4+3*x**3+4 is as follows:

```
The degree of this graph is 4
It is polynomial
There are 3 possible maximum and minimum
The y-intercept is (0,4)
The graph has x-intercept are (-3,0) and (-1,0)
The graph is located at 1,2,3 quadrants
Absolute minimum is (-2.14815 -4.40191)
```

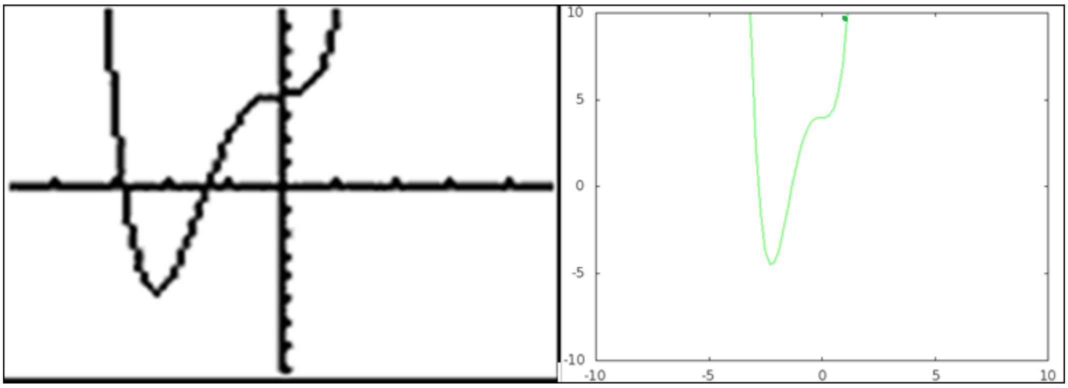


Figure 7.9: Sample1 Mathematical graph

Sighted user can extract the following information from the left hand side of Figure 7.10:

```
There are two maximum points at (-1.11, 2.12) and (0.33, 1.22).
There is a minimum at (-0.34, 0.78).
Absolute maximum is (-1.11, 2.12)
The maximum value ( 0.33,1.22) is a relative maximum, being the
   largest value relative to points close to this on the graph.
The minimum value (-0.34,0.78) is a relative minimum, being the
   smallest value relative to points close to this on the graph.
```

Date extracted by MathGraphReader using right hand side of Figure 7.10 and function f(x)=-1*x**6-3*x**3+x+1 is as follows:

```
The degree of this graph is 6
It is polynomial
There are 5 possible maximum and minimum
The y-intercept is (0,1)
The x-intercept values are (-1,0)(1,0)
The graph is located at  all quadrants
Absolute maximum is (-1, 2.10526)
Relative minimum is (-0.222222 0.813397)
Relative maximum is (0.296296 1.19617)
```

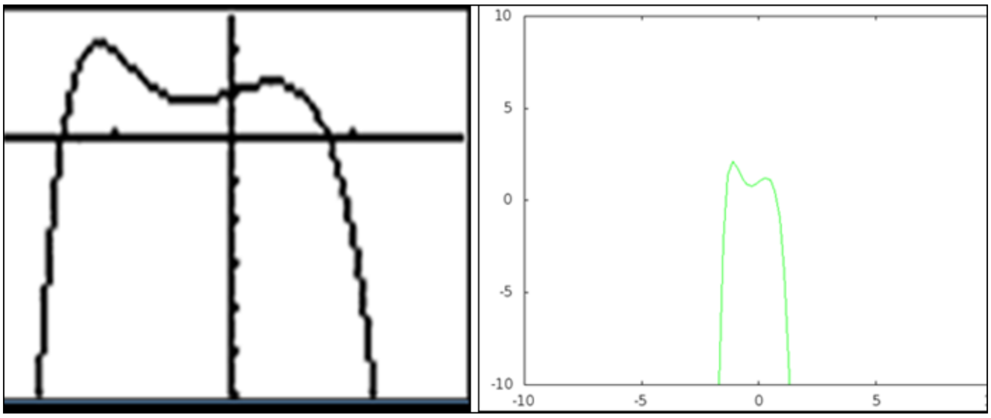


Figure 7.10: Sample2 Mathematical graph

Sighted user can extract the following information from the left hand side of Figure 7.11:

```
There are two minimum points on the graph at (0.70, -0.65) and
   (-1.07, -2.04).
There is a minimum at (0.70, -0.65)is called a relative minimum
   because it is not the minimum or absolute, smallest value of
   the function. It is a minimum value relative to the points
   that are close to it on the graph.
The minimum value at (-1.07, -2.04 )is called an absolute minimum
    because it is the smallest value of of the function.
There is a maximum at (0, 0). This maximum is called a relative
   maximum because it is not the maximum or absolute, largest
   value of the function. It is a maximum value relative to the
   points that are close to it on the graph.
```

Date extracted by MathGraphReader using right hand side of Figure 7.11 and function f(x)=2*x**4+x**3-3*x**2 is as follows:

```
The degree of this graph is 4.
It is polynomial
There are 3 possible maximum and minimum
The y-intercept is (0,0)
The x-intercept are (-1 ,0)(0,0)(-1,0)
The graph is located at  all quadrants
Absolute minimum is ( -1.03704 , -1.96172 )
Relative maximum is ( 0, 0 )
Relative minimum is (0.740741 -0.574163)
```

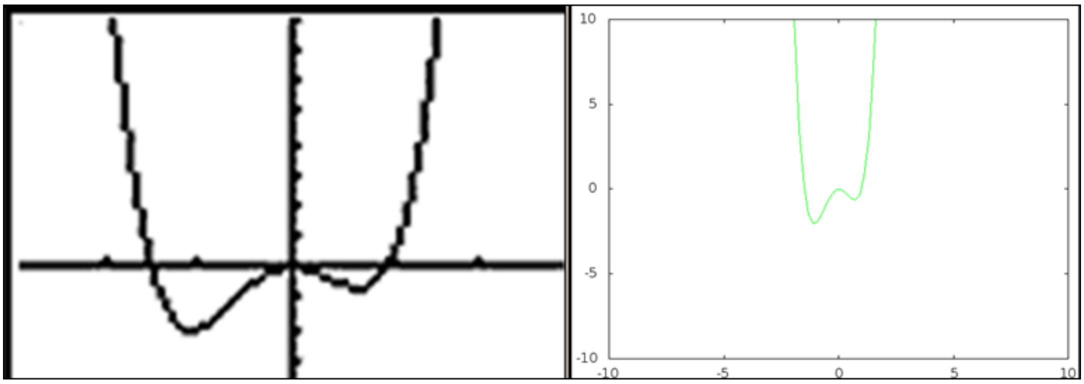


Figure 7.11: Sample3 Mathematical graph

Sighted user can extract the following information from the left hand side of Figure 7.12:

```
There is a relative minimum (0.24 , -0.34)
There is a relative maximum value (2.09, 6.05 )
```

Date extracted by MathGraphReader using right hand side of Figure 7.12 and function f(x)=-2*x**3+7*x**2-3*x is as follows:

```
It is polynomial
There are 2 possible maximum and minimum
The y-intercept is ( 0,0)
The x-intercept are (0,0), (1,0),(3,0)
The graph is located at 1,2,4 quadrants
Relative minimum is (0.37037, -0.287081)
Relative maximum is (2.22222,6.07656 )
```

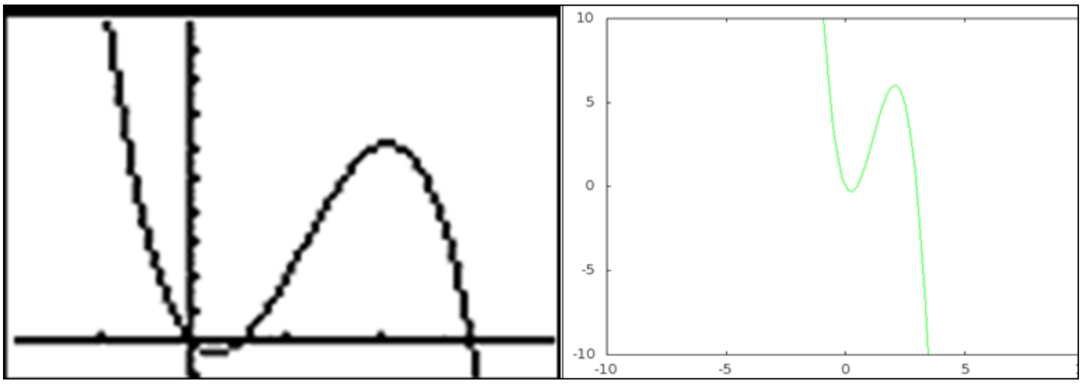


Figure 7.12: Sample4 Mathematical graph

7.8 MATHSPEAK

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \tag{7.1}$$

In this section 16 different Amsmath mathematical expressions were applied to MATHSPEAK to be rendered and make alternative description for these expressions without ambiguity.

$$a^3 + b^3 = (a + b)(a^2 - ab + b^2) \tag{7.2}$$

```
        amsmath: \begin{equation}a^3+b^3=(a+b)(a^2-ab+b^2) \end{
            equation}
MATHSPEAK Alternative Text Description:
<a /> < Power 3/> <  plus /> < b /> < Power 3/> <  equal/> < open
    bracket/><a  /><plus /> <  b /> < close bracket/> < open bracket
    /> < a/> < Power  2/> <  minus/><a/><times/> < b/> <  plus/> <  b/> <
     Power 2/> <  close bracket/>
```

$$\sqrt[n]{1 + x + x^2 + x^3 + \ldots} \tag{7.3}$$

```
amsmath: \begin{equation}\sqrt[n]{1+x+x^2+x^3+\ldots}\end{
    equation}
 MATHSPEAK Alternative Text Description:
<root  n  of/> <  open bracket/> <   1/> <  plus  x/> <   plus/> <   x
     /> <  Power 2/> <   plus/> <  x/> <  Power 3/> <   plus/> <
    continue  until /> < close bracket/>
```

$$\sum_{i=1}^{10} t_i \tag{7.4}$$

```
amsmath: \begin{equation} \sum_{i=1}^{10} t_i\end{equation}
MATHSPEAK Alternative Text Description:
<Sum />
< from i    equal    1/>
<to  10 />
< for t/> <  index i />
```

$$\frac{\frac{1}{x} + \frac{1}{y}}{y - z} \tag{7.5}$$

```
amsmath: \begin{equation}\frac{\frac{1}{x}+\frac{1}{y}}{y-z}\end{
    equation}
 MATHSPEAK Alternative Text Description:
<open bracket/>
< 1  over x/> <   plus/> <   1  over y/>
<   close bracket/>
< over/>
```

```
< open bracket/>
< y  minus z/>
<   close bracket/>
```

$$\lim_{x \to \infty} \exp(-x) = 0 \tag{7.6}$$

```
amsmath: \begin{equation} \lim_{x \to \infty} \exp(-x) = 0\end{
   equation}
 MATHSPEAK Alternative Text Description:
< Limit />
< as x approches  to Infinity />
< of/>< Exponential of/><  open bracket/><  minus  x/>< close
   bracket/><  equal,     0 />
```

$$k_{n+1} = n^2 + k_n^2 - k_{n-1} \tag{7.7}$$

```
amsmath: \begin{equation}k_{n+1}=n^2+k_n^2-k_{n-1}\end{equation}
MATHSPEAK Alternative Text Description:
<k/><  index /><   open bracket/>< n /><  plus/><  1 /><  close
   bracket/><    equal/><  n/><  Power 2 />< plus/><   k/><
   index  n/><   Power 2/><   minus/><   k />< index /><   open
   bracket,/>< n  />< minus />< 1/><   close bracket,/>
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k} \tag{7.8}$$

```
amsmath: \begin{equation} \frac{n!}{k!(n-k)!} = \binom{n}{k}\end{
   equation}
MATHSPEAK Alternative Text Description:
< n factorial/><   over/><  open bracket/><  k factorial/><  open
    bracket/><  n  minus k/><   close bracket/>< factorial/><
   close bracket/>< equal/><  Binomial of/><  n choose k/><
   closed Binomial/>
```

$$2^k - \binom{k}{1}2^{k-1} + \binom{k}{2}2^{k-2} \tag{7.9}$$

```
amsmath: \begin{equation} 2^k-\binom{k}{1}2^{k-1}+\binom{k}{2}2^{
   k-2}\end{equation}
MATHSPEAK Alternative Text Description:
< 2/><  Power k/><   minus/><  Binomial of/><  k choose 1/><
   closed Binomial/>< times/><  2/><  Power/><  open bracket/><
   k  minus 1/><   close bracket /><   plus/><  Binomial of/><  k
```

```
 choose 2/>< closed Binomial/>< times/>< 2/>< Power /><
open bracket/>< k minus 2/>< close bracket/>
```

$$\frac{1}{k}\log_2 f(c)\sqrt{\frac{1}{k}\log_2 f(c)} \tag{7.10}$$

```
amsmath: \begin{equation}\frac{1}{k}\log_2 f(c)\sqrt{\frac{1}{k}\
   log_2 f(c)}\end{equation}
MATHSPEAK Alternative Text Description:
< 1 over k/><times/>< Logarithm based 2/><for/>< function of
   c/>< times/>< Square root of />< 1 over k/>< Logarithm
   based 2 />< for/><function of c/><close square root/>
```

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}} \tag{7.11}$$

```
amsmath: \begin{equation}
  x = a_0 + \frac{1}{a_1          + \frac{1}{a_2           + \frac
     {1}{a_3 + a_4}}}\end{equation}
MATHSPEAK Alternative Text Description:
<x/> < equal/> < a index 0 /> < plus/> < 1/> <over/> < open
   bracket/> < a index 1/> < plus/> < 1/> < over/> < open
   bracket/> < a index 2 /> < plus/> < 1 /> <over/> < open
   bracket/> < a index 3/> < plus/> < a index 4/>
< close bracket/> < close bracket/> < close bracket/>
```

$$M = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{pmatrix} \tag{7.12}$$

```
amsmath: \begin{equation}
 M =\begin{pmatrix}
      \frac{5}{6} & \frac{1}{6} & 0     \\        ]
      \frac{5}{6} & 0           & \frac{1}{6} \\]
      0           & \frac{5}{6} & \frac{1}{6}
    \end{pmatrix}
\end{equation}}
MATHSPEAK Alternative Text Description:
< M /> <equal/>< matrix of 3 rows and 3 columns/>
< 5 over 6 />< next element/>< 1 over 6 /> < next element 0
    /> < next row/>
< 5 over 6 /> < next element/> < 0 /> < next element/> <
   1 over 6 /> < next row/>
```

```
<  0 /> < next element /> < 5 over 6 /> < next element/> <
    1, over, 6/>
```

$$\sqrt{\frac{\sum_{i=1}^{n}(x_i-y_i)}{\prod_{i=1}^{n}(x_i+y_i)}} \tag{7.13}$$

```
amsmath: \begin{equation}
 \sqrt{\frac{\sum_{i=1}^{n}(x_i-y_i)}{\prod_{i=1}^{n}(x_i+y_i)}}
\end{equation}
 MATHSPEAK Alternative Text Description:
<Square root of/>< Sum /><from i  equal  1 />< to n
   /><for/>< open bracket />< x index i />< minus /> < y
   index i />< close bracket/> < over/>< Production /><from
    i  equal  1 />< to n /><for/>< open bracket /> < x
   index i />< plus /> < y index i />< close bracket
   /><close square root/>
```

$$\lim_{n \to \infty}(1+\frac{1}{n})^n \tag{7.14}$$

```
amsmath: \begin{equation}
\lim_{n \to \infty}\(1+\frac{1}{n})^n
\end{equation}
MATHSPEAK Alternative Text Description: ccccccc
<Limit /><as n approches to Infinity />< of /> < open bracket
   /> < 1/> < plus /> < 1 over n /> < close bracket /><
   Power n />
```

$$(x+a)^n=\sum_{k=0}^{n}\binom{n}{k} x^k a^{(n-k)} \tag{7.15}$$

```
amsmath: \begin{equation}
(x+a)^n=\sum_{k=0}^{n}\binom{n}{k} x^k a^(n-k)
\end{equation}
MATHSPEAK Alternative Text Description:
<open bracket/>
<1 plus a/>< Power n/>< equal/>< Sum /><from k  equal  0
    /><to n/><for/>< Binomial of /><n choose k /><closed
   Binomial /><times/>< x />< Power k /><times/>< a />< Power
   />< open bracket />< n minus k /><close bracket />
```

$$(1+x)^n = 1+\frac{nx}{1!}+\frac{n(n-1)(x)^2}{2!}+cdots \tag{7.16}$$

```
amsmath: \begin{equation}
(1+x)^n=1+\frac{nx}{1!}+\frac{n(n-1)(x)^2}{2!}+cdots
\end{equation}

MATHSPEAK Alternative Text Description:
<open bracket/>< 1 plus x />< close bracket/>< Power n /><
    equal />< 1/>< plus/>< nx over 1 factorial/><
   plus />< open bracket />< n/><times/>< open bracket />< n
      minus 1 />< close bracket/>< x />< Power 2 />< close
   bracket />< over/>< 2! />< plus />< continue until />
```

7.9 Alternative Text Description for Chemical Equation

This section represents extracted information from six samples of chemical equations. This information contains name of participants elements in chemical reaction, quantity of each elements at the left and right side of equation and corresponding algebraic equations to find out coefficients.

a$Fe + bCl_2 = cFeCl_3$

```
Fe plus Cl index 2 yields Fe Cl index 3
For balancing it needs 3 coefficients  this equation includes 2
   elements, these elements are:
Cl=Chlorine left=2 right=3
Fe=Iron left=1 right=1
 use following equations to obtain  coefficients
2*b=3*c
1*a=1*c
```

a$KMnO_4 + bHCl = cKCl + dMnCl2 + eH_2O + fCl_2$

```
K Mn O index 4 plus H Cl yields K Cl plus Mn Cl index 2 plus H
   index 2 O plus Cl index 2
For balancing it needs 6 coefficients, this equation includes 5
   elements, these elements are:
Cl=Chlorine  left=1 right=5
H=Hydrogen left=1  right=2
K=Potassium left=1 right=1
Mn=Manganese left=1 right=1
O=Oxygen left=4 right=1
use following equations to obtain  coefficients
1*b=1*c+2*d+2*f
1*b=2*e
1*a=1*c
```

```
1*a=1*d
4*a=1*e
```

a K4Fe(CN)6 + b H2SO4 + c H2O = d K2SO4 + e FeSO4 + f (NH4)2SO4 + g CO

```
K index 4 Fe open bracket C N closed bracket index 6 plus H index
     2 S O index 4 plus H index 2 O yields K index 2 S O index 4
    plus Fe S O index 4 plus open bracket N H index 4 closed
    bracket index 2 S O index 4 plus C O
For balancing it needs 7 coefficients, this equation  includes 7
    elements, these elements are:
C=Carbon  left =6 right= 1
Fe=Iron  left = 1 right= 1
H=Hydrogen  left = 4 right= 8
K=Potassium  left = 4  right= 2
N=Nitrogen  left = 6 right= 2
O= Oxygen  left = 5 right= 17
S=Sulphur  left = 1 right= 3
 use following equations to obtain  coefficients
6*a=1*g
1*a=1*e
2*b+2*c=8*f
4*a=2*d
6*a=2*f
4*b+1*c=4*d+4*e+4*f+1*g
1*b=1*d+1*e+1*f
```

a C6H5COOH + b O2 = c CO2 + d H2O

```
C index 6 H index 5 C O O H plus O index 2 yields C O index 2
    plus H index 2 O
For balancing it needs 4 coefficients, this equation includes 3
    elements, these elements are:
C=Carbon  left = 7 right= 1
H=Hydrogen  left = 6 right= 2
O=Oxygen  left = 4 right= 3
 use following equations to obtain  coefficients
7*a=1*c
6*a=2*d
2*a+2*b=2*c+1*d
```

a K4Fe(CN)6 + b KMnO4 + c H2SO4 = d KHSO4 + e Fe2(SO4)3 + f MnSO4 + g HNO3 + h CO2 + i H2O

```
K index 4 Fe open bracket C N closed bracket index 6 plus K Mn O
    index 4 plus H index 2 S O index 4 yields K H S O index 4 plus
     Fe index 2 open bracket S O index 4 closed bracket index 3
```

```
   plus Mn S O index 4 plus H N O index 3 plus C O index 2 plus H
    index 2 O
For balancing it needs 9 coefficients,this equation includes 8
   elements, these elements are:
C=Carbon  left = 6 right= 1
Fe=Iron  left = 1 right= 2
H=Hydrogen  left = 2 right= 4
K=Potassium  left = 5 right= 1
Mn=Manganese  left = 1 right= 1
N=Nitrogen  left = 6 right= 1
O=Oxygen  left = 8 right= 26
S=Sulphur  left = 1 right= 5
 use following equations to obtain  coefficients
6*a=1*h
1*a=2*e
2*c=1*d+1*g+2*i
4*a+1*b=1*d
1*b=1*f
6*a=1*g
4*b+4*c=4*d+12*e+4*f+3*g+2*h+1*i
1*c=1*d+3*e+1*f
```

a PhCH3 + b KMnO4 + c H2SO4 = d PhCOOH + e K2SO4 + f MnSO4 + g H2O

```
Ph C H index 3 plus K Mn O index 4 plus H index 2 S O index 4
   yields Ph C O O H plus K index 2 S O index 4 plus Mn S O index
    4 plus H index 2 O
For balancing it needs 7 coefficients, this equation includes 7
   elements these elements are:
C=Carbon  left = 1right= 1
H=Hydrogen  left = 5 right= 3
K=Potassium  left = 1 right= 2
Mn=Manganese  left = 1 right= 1
O=Oxygen  left = 8 right= 11
Ph  left = 1 right= 1
S=Sulphur  left = 1 right= 2
 use following equations to obtain  coefficients
1*a=1*d
3*a+2*c=1*d+2*g
1*b=2*e
1*b=1*f
4*b+4*c=2*d+4*e+4*f+1*g
1*a=1*d
1*c=1*e+1*f
```

7.10 Mathematical Handwritten Documents Segmentation Results

Accommodating handwritten information retrieval in this system is reserved for further development. Thus line segmentation and symbol segmentation modules of this research were used for some handwritten materials. This section provides results of line segmentation and character segmentation applied to a handwritten sample. As it was shown in Figure 7.13 the handwritten sample document was segmented to lines without any over-segmentation or broken symbols. However under-segmentation occurs for lines 3 and 4 due to skew of handwriting. In symbol segmentation area , lines 5, 6 are segmented to symbols. Although due to have connected components $x^2, 3x$ in line 5 and $x^2, 2x$ in line 6 were recognised as primitive components.

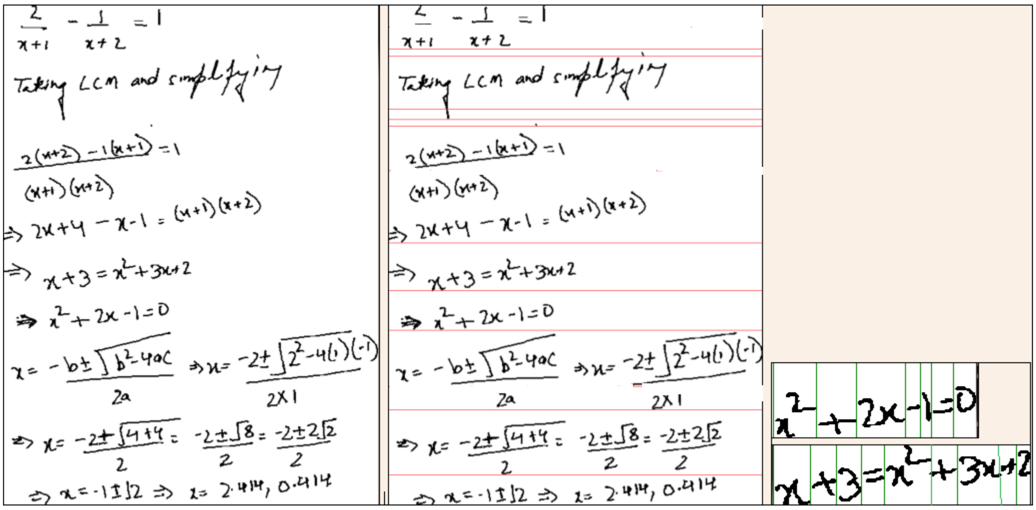


Figure 7.13: Handwritten mathematical sample (left), line segmentation result (middle) and character segmentation of two lines (right).

7.11 Summary

Besides presented samples for each module in Chapter 5, more samples applied to various modules in CRS application to investigate how accurate they are. Evaluated modules in this chapter include Block Segmentation, Chart Recognition, BarChartReader, PieChartReader, LineChartReader, MathGraphReader, MATHSPEAK , Chemical Equation and Mathematical Handwritten Documents Segmentation. Average of accuracy rate for segmentation modules is more than 95%. For chart related modules due to firm dependency to OCR accuracy is approximately 73% . Applied samples for MATHSPEAK and Chemical Equation include vast range of simple, moderate and difficult samples. Since MATHS-

PEAK and Chemical Equations are based on text processing instead of image processing the accuracy is near 100%.

Chapter 8

Conclusion

This chapter recapitulates the research that has been carried out and described in this thesis. Through the conducted literature review, it was identified that existing traditional methods to represent documents in alternative format to print disabled people suffer many disadvantages. These methods are based on tactile and audio learning. Most significant disadvantages of tactile representation are:

- Obligation to learn how to use some extra (expensive) equipment and devices or codes such as Haptic devices and Braille codes.
- Dependency on tactile sensitivity which not all vision-impaired may have.
- Braille documents are bulky, heavy and not available for all materials such as user manuals, references and scientific books. Braille is appropriate to represent linear and ordered material. In order to address Braille inability to represent multidimensional or spatially arranged formulae such as algebra, multiple codes for mathematics and science were developed including the Nemeth Braille Code in the United States, Unified English Braille (UEB) in Great Britain and the Woluwe Code in Dutch-speaking parts of Belgium(Karshmer & Bledsoe,2002) Mathematics are represented braille using a dedicated braille code containing braille-specific symbols, or using a linear translation of a Mark-up format into braille. Results of a reasech by (Bitter, 2013) conclude that dedicated braille codes are significantly better at assisting the reader than linear translation of a Mark-up format into braille. Vision-impaired learners use tactile sensitivity that do not allow an overview of mathematical expression and they have to entirely remember mathematical expression in order to apply it which leads increasing the cognitive load. Multidimensional mathematical formulae representation involves higher cognitive complexity.

Disadvantages identified for audio representation are:

- Anlogue representation for print publication which usually human read and recorded on cassette tape or record easily get damaged and wear off over time; distribution is costly and offer only sequential access to information with extremely limited navigation.

- Sonification or conveying data via sound pitch and 2-dimensional acoustics is difficult to convey data accurately with the acoustic method (non-speech sound), and moreover, since acoustics are volatile, information can easily be misinterpreted

As a result there is a real need to explore avenues to improve electronic document accessibility to print disabled people. This research set out to address this need by designing and implementing a stand-alone, simple to use and affordable Complete Reading System (CRS) for print disabled users. The main goals were to:

1. Develop CRS modules (Chapter 5);
2. Integrate and implement the developed modules using a Single Board Computer (SBC) as a hardware platform; and
3. Find appropriate compatible input/output interface for user-system interaction (Chapter 6).

Developed modules support the accessibility of a large variety of electronic documents including multiple types of PDF (tagged, structured and scanned PDF), DOC, DOCX, ODT, Plain Text, Audio Formats and DAISY.

Empirical results of two surveys conducted within this research have shown that:

1. PDF is the most commonly used and most inaccessible through assisitive technology; and
2. Navigation ability plays significant role during reading session and provide active and dynamic reading experience.

Moreover, experimental results in this study have confirmed that:

1. All types of PDF have accessibility issues through assisitive technology:
   - DRM in structured PDF prevents screen reader from reading text; and
   - Screen readers are unable to convert scanned PDFs to speech due to inability to deal with images.
2. The solution regarding PDF accessibility issues is not limited to perform OCR or removing DRM, but it contains:
   - PDF layout analysis to retain reading order; and
   - Information retrieval from different components in PDF such as not-in-order component (tables), non-textual component (charts and graphs), and multidimensional components (mathematical expressions and chemical equations).

- Non-textual information retrieval has two main sub-modules: text data extraction; and graphics data extraction.

Table 8.1, 8.2 and 8.3 indicate identified detailed objectives of this study regarding accessibility issues, proposed corresponding solutions and utilised techniques, packages or tools. Performing these methods produced successful outcomes.

Table 8.1: Research objectives, proposed methods and used techniques -1

Objective	Methods implemented	Packages/tools/technologies
Structured PDF accessibility issues caused by DRM	PDF to HTML Conversion HTML to text Conversion	poppler-utils (pdftohtml) html2text
Scanned PDF accessibility issues	OCR	OCRopus
Preprocessing	Binarization Black border removal Margin removal Skew detection and correction	Image processing: Threshold Image processing: Negate Image processing: Trim Trigonometry Image processing: rotate
Retain reading order	PDF Layout analysis Block segmentation	Remove non interested white space Morphology eroded Horizontal white space=Width(Page) Horizontal page segmentation Vertical white space=Height(Horizontal segment) Vertical segmentation $\Rightarrow$Block Bounding Box analysis
Detection Extraction isolated mathematical expression	Line segmentation Page global line labelling	Horizontal white space=Width(Block) Horizontal segmentation $\Rightarrow$Line Feature extraction, Machine Learning SVM
Detection Extraction embedded mathematical expression	Word segmentation	Vertical white space=Height(Line) Vertical segmentation$\Rightarrow$Word Feature extraction Machine Learning SVM
PDF navigation ability	hOCR generation	PDF layout analysis Segmentation Bounding Box analysis

Table 8.2: Research objectives, proposed methods and used techniques-2

Objective	Methods implemented	Packages/tools/technologies
Conversion of multidimensional mathematical expression to linear form	Recursive Components Extraction (RCE) VCE and HCE Bracket Rule	Vertical White Space=Height(Mathematical block) Vertical segmentation Horizontal White Space=Width(Vertical segment) Horizontal segmentation ⇒Symbol
Recognition of order, role and relation of mathematical symbol	Structural symbol analysis Symbol layout analysis	Feature Extraction: Slope Height-Ratio Middle-line-Ratio
Mathematical symbol recognition	Binary Vector Machine Learning kNN	Euclidean Distance allknn (mlpack)
Meaningful representation & concepts preservation of Mathematical expressions	Merging symbol Rendering	Symbol Layout Analysis Parsing Symbols Analysis Bounding Box Neighbourhood
Table (not-in-order) information retrieval & effective and practical table representation	Cell segmentation Sort cells Analysis Bounding Box	Image Processing: Morphology, Crop Bounding Box Finding
Extraction non-textual components such as charts	Text/non-Text Segmentation	Block ⇒Morphology dilated⇒Morphology eroded clip-mask Block ⇒non-Textual. ⇒-negate ⇒-compose plus -composite Block⇒Text-Only

Table 8.3: Research objectives and proposed methods to achieve them-3

Objective	Methods implemented	Packages/tools/technologies
Chart Classification Charts information retrieval	Chart Recognition Chart/Legend segmentation Shape/Axes segmentation	Feature extraction Chart-Only ⇒Morphology dilate rectangle:1x10 Removed-Horizontal-Lines Y-axis is the most left vertical line Chart-Only ⇒Morphology dilate rectangle:10x1 rectangle:10x1 Removed-Vertical-Line X-axis is the lowest vertical line OCR: legend & axes Image processing: Shape-only Bars-only, Pie-only, Markers-only
Mathematical graph information retrieval	Regenerate the graph Finding graphs critical points: Maxima/Minima Intercept points Flexes Alternative text description	Applying graph function to GNUPLOT Digitization Mathematical operations
Chemical equation presentation	Component extraction Component classification Design tagging system for: species, reactant, product, quantity and element Mark-up representation	Text processing Bash script
Mathematical representation without ambiguity	Rendering Parsing Text processing	c++ Bash script
DAISY player	Mark-up format processing	Bash script

During this research, development and integration of series of modules produced a Complete Reading System. The prototype was implemented using a SBC hardware platform, Linux as the Operating System and the developed software application within this research providing functionality. After system is turned on and start up process is passed, Spoken Main Menu requests the user to connect USB storage containing files (to read) to the system. Upon detecting USB device Spoken Menu provides an opportunity for the user to find and select desired file to read (section 5.2: Overview of CRS Module). A main concern for CRS is dealing with Scanned PDF. In such a case, CRS performs PDF layout analysis and block segmentation. Page document is divided to table block and text/non-text block. Table block is applied to TableReader which is responsible for cell segmentation. Each individual cell is sent to OCR and the results are represented to user by row or by column or both according to user request. Text/non-text block is segmented, then text block is segmented to line followed by MIR to extract and retrieve possible mathematical expression from text block. Non-Text segment could be chart or mathematical function graph. ChartRecognition runs Chart/legend segmentation. Chart without legend is applied to axes segmentation. Axes and legend which contain text information are sent to OCR to support digitization and provide textual description. This procedure is illustrated in Figure 8.1.

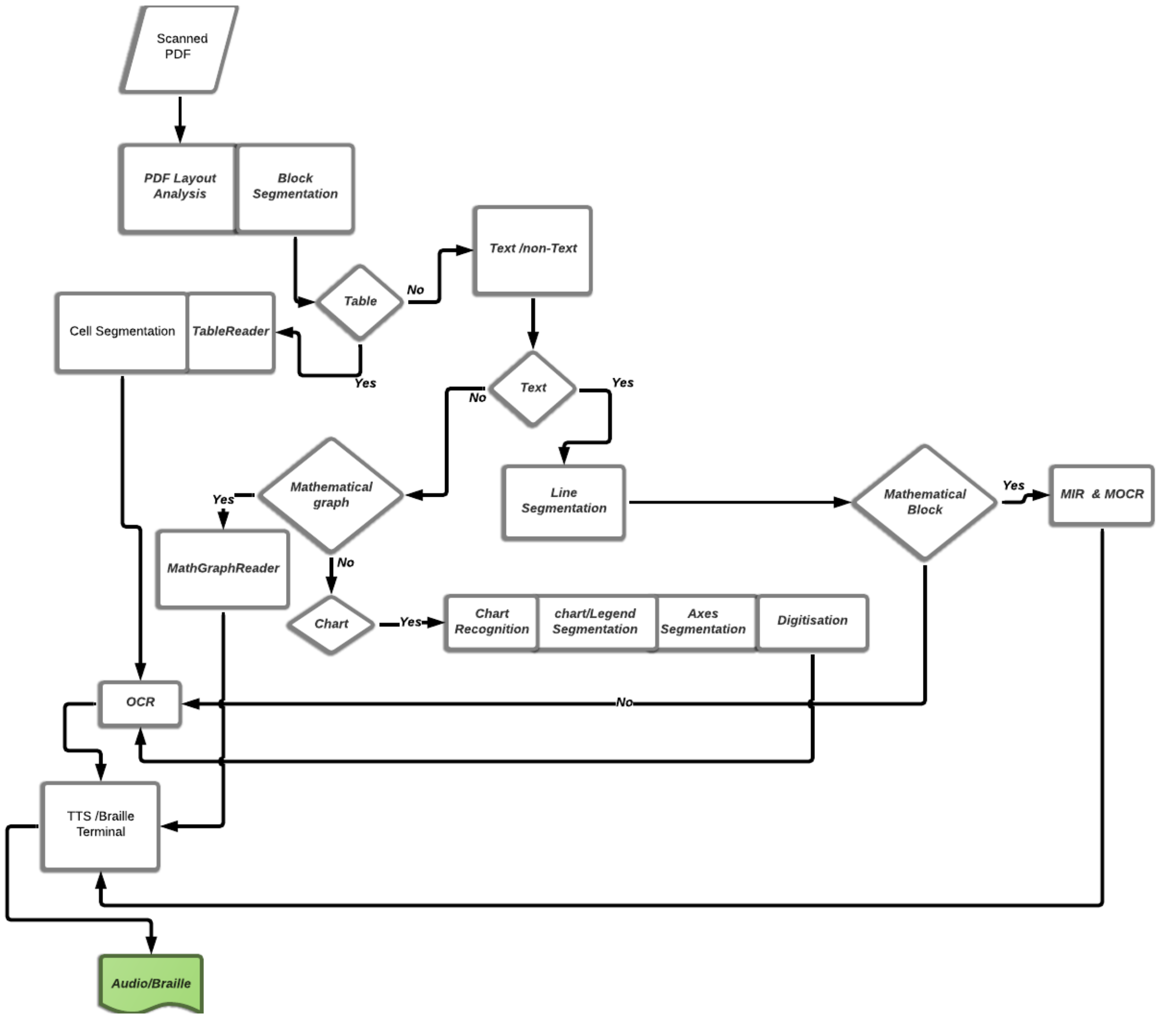


Figure 8.1: Scanned PDF processing with CRS

Segmentation layers used in this research play significant roles in document layout analysis and can potentially be adapted for multiple categories of document images such as scanned PDF, unstructured text document images (HVTOs, forms, and envelop) and mostly graphics document images (maps and engineering drawings).

Several tasks related to scanned PDF such as retaining reading order, information retrieval from tables and mathematical function graphs, representing chemical equations and mathematical expression in comprehensive manner and charts recognition and segmentation were successfully performed in this research.

The compatible DAISY player in this system allows users to access, navigate, bookmark, and search complex material such as encyclopedia, references books and user manuals.

Various human computer interaction methods were studied in order to facilitate communication for people with multiple disabilities. The results determined that joystick or customised keypad with minimum number of tactile buttons utilizing USB controller can be suitable devices to transfer user commands to system. The use of Braille terminal or audio out port in conjunction with the TTS engine renders alternative text description which was generated with CRS modules. The system developed is fully compatible with speech input, joystick control and standard refreshable Braille display terminal.

8.1 Future Work

Implementing an initial prototype for the Digital Talking Book Player using FPGA illustrated that CRS implementation and integration of software application modules require a powerful processor and sufficient memory. Thus a Single Board Computer (SBC) with an ARM processor as the hardware platform for CRS was investigated. There are several peripherals within the SBC development environment which were deemed are unnecessary for the CRS. In addition, many of the required hardware components for the Reading System in SBC contain unused capabilities that increase cost and should be replaced with simpler hardware to decrease power consumption and retain costs as low as possible. To achieve this purpose customizing an open source SBC was identified as a task to follow this research.

MathGraphReader, is a module developed in this research which converts mathematical function graph to accessible description text for print disabled users based on the availability of graph's function. This system may be extended to produce text description for graphs for which their mathematical function is unknown by using mathematical modeling.

Currently, the DAISY player compatible module of the CRS plays DAISY books from a connected USB cartridge. Further development using the DAISY Online Streaming Delivery Protocol will provide an opportunity for print disabled users to access vast range of DAISY materials available in online DAISY library (Nazemi et al., 2014).

In addition, the text extraction sub-module of information retrieval from graphs is highly dependent on OCR accuracy rate. The recognition rate of OCRs are affected by many factors beyond the scope of this research, consequently information retrieval of charts and graphs is limited to segmentation and extraction of components without recognition. As a result, digitization was not performed completely. To address this issue, improving OCR accuracy is an area requiring further investigation.

The segmentation modules have been already tested for some handwritten mathematical samples with reasonably accurate results.Therefore accommodating handwritten information retrieval in this system is reserved for further development.

Presently, the availability of print disabled students with required knowledge of mathematics and chemistry is limited. Further testing and evaluation in the modules related to mathematical expressions and chemical equation accessibility must be performed. The integration of all developed modules in Chapter 5 in a coherent complete functional system and commercialization a unique, stand alone simple-to-use system which allows users to simply access electronic documents is another concern for future work.

8.2 Final Reflection

The application of the results from this research, applied with the CRS now has the potential to significantly impact the vision impaired community. Through this research the world of both mathematics and chemistry has now been opened to vision impaired people in a way that it has never been. Previously, if a vision impaired person had wanted to pursue higher education within either of these fields it was possible but only with the support and assistance of someone transcribing into an appropriate format the research papers that the vision impaired person was needing to read. Through the CRS the area of mathematics and chemistry are now open to the vision impaired person to be able to take research papers, parse them through this system and have a format generated that can be played through an assistive technology device that will read the papers, including the formulae, graphs and other pictorial representations, out with the possibility to pause rewind and bookmark. The realm of higher education in mathematics and chemistry has now been opened to the vision impaired person to attempt on their own.

Bibliography

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Aly, W., Uchida, S., Fujiyoshi, A., & Suzuki, M. (2009). Statistical Classification of Spatial Relationships among Mathematical Symbols: Proceedings of the 10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain,(pp.1350-1354).

Aly, W., Uchida, S., & Suzuki, M. (2008). Identifying Subscripts and Superscripts in Mathematical Documents. Mathematics in Computer Science, 2(2), 195-209. doi: 10.1007/s11786-008-0051-9

American Mathematica Society. (1999). User's Guide for the amsmath Package(Version 2.0).

Assistive Technology and Accessibility Centers. (2007). Refreshable Braille Display.Indiana University. Retrieved from http://www.indiana.edu/~iuadapts/technology/hardwarfe/braille/alva.html

Australian Copyright Council. (2007). Print disability copyright guidelines. Copyright Agency website: Retrieved from http://www.copyright.com.au/licences/people-with-disabilities/print-disability-copyright-guidelines

Bailey, L. (2005). Creating Accessible PDF from Scanned Documents. Retrieved from http://wac.osu.edu/pdf/scan/PDFfromScanner.pdf

Barbe, W. B., Swassing, R. H., & Milone, M. N. (1979). Teaching Through Modality Strengths: Concepts and Practices. Columbus, Ohio: Zaner-Blosner.

Barrett, A. (2007). Can Quark Turn the Corner?. INC magazine.

Baxter, S. (2013). Three Ways to Process Information. Retrieved from http://elementary-science-software-review.toptenreviews.com/learning-styles-three-ways-to-process-information.html

Becker, J. V., Hinton, D. E., & Anderson, H. G. (2004). Braille computer monitor: Google Patents.US6700553 B2.

Ben-Zvi, R., Eylon, B., & Silberstein, J. (1987). Students visualization of a chemical reaction. Education in Chemistry. 117-120.

Berman, B. P., & Fateman, R. J. (1994). Optical Character Recognition for Typeset Mathematics: Proceeding of International Symposium on Symbolic and Algebraic Computation, (ACM Press) (ISSAC-94).

Bitter, M,(2013),Braille in Mathematics Education, Radboud University Nijmegen

Bodner, A. L., Micklos, L., & George M.(2013). Chemical Reactions: What Understanding Do Students with Blindness Develop? Chemistry education research and practice: 625-636.

Bohman, P. (2002). How to Create Accessible Acrobat PDF files. Retrieved from
http://wac.osu.edu/webaim/acrbaxs.html

Bohman, P. (2014). Why Don't Screen Readers Always Read What's on the Screen? Part 1: Punctuation and Typographic Symbols. Retrieved from http://www.deque.com/blog/dont-screen-readers-read-whats-screen-part-.1-punctuation-typographic-symbols/

Boyer, J. J. (2002). Liblouis:an open-source braille translator and back-translator. Abilitiessoft, Inc.

Breuel, T. M. (2008). The OCRopus open source OCR system.

Breuel, T. M., & Kaiserslautern, U. (2007). The hOCR Microformat for OCR Workflow and Results: Proceeding of the ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007.

British Gas. (2013). Understanding your bil. Retrieved from
http://www.britishgas.co.uk/youraccount/discover/your-bill.html

Brown, A. (2007). Non-Visual interaction with graphs', Manchester(PhD thesis).

Brown, A., Stevens, R., & Pettifer, S. (2013). Making Graph-Based Diagrams Work in Sound: The Role of Annotation,. Human-computer interaction, 193-221.

Brown, L. M., & Brewster, S. A. (2003). Drawing by ear: Interpreting sonified line graphs: Proceeding oft the International Conference on Auditory Display (ICAD2003).

Brown, L. M., Brewster, S. A., Ramloll, R., and, M. B., & Riedel, B. (2003). Design guidelines for audio presentation of graphs and tables. Proceeding of ICAD 2003.

Carberry, S., Elzer, S., & Demir, S. (2006). Information graphics: an untapped resource for digital libraries. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA.

Carlson, R., Granstrom, B., & Hunnicutt, S. (1982). A multi-language text-to-speech module: Proceeding IEEE International Conference at the Acoustics, Speech, and Signal (ICASSP '82).

Chang, S.K. (1970). A method for the structural analysis of two-dimensional mathematical expressions. Inf. Sci., 2(3), 253-272.

doi: 10.1016/s0020-0255(70)80052-4

Cleaver, S. (2011). 12 Ways to Teach Your Visual Learner.Retrieved from http://www.education.com

Cleveland, W. S. (1985). The elements of graphing data. Pacific Grove, Calif: Wadsworth & Brooks/Cole Advanced Books & Software.

Cohen, R. F., Haven, V., Lanzoni, J. A., Meacham, A., Skaff, J., & Wissell, M. (2006). Using an audio interface to assist users Who are visually impaired with steering tasks. Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility, Portland, Oregon, USA.

Conrod, B. E. (1996). Active and Passive Perceptual Learning in the Visually Impaired. Journal of Visual Impairment and Blindness, v80 n1 p528-31 Jan 1986.

Reading Rights Coalition. (1989).Council for Exceptional Children, (2008), Higher Education Opportunity Act Reauthorization. Retrieved from https://www.aucd.org/docs/CEC%20Higher%20Education%20Analysis.pdf

Crawford Technologies Inc. (2011). Is PDF an appropriate choice for high volume transactional production printing?. Retrieved from www.crawfordtech.com

Curtin, R. R., James, R. C., Slagle, N.P., Amidon, M. L., Gray, A.G. (2011). MLPACK: A scalable C++ machine learning library. School of Computational Science and Engineering Georgia Institute of Technology.

Diagram2012. (2012). Accessible Graphics, Graphics for Vision Impaired People,. Proceeding of the 7th International Conference on the theory and application. Retrieved from http://www.diagrams-conference.org/2012/content/accessible-graphics-graphics-vision-impaired-people.html

Digilent. (2008). PmodDA2 Reference Component.

Dixon, J. (2007). Eight-dot Braille. A Position Statement of the Braille Authority of North America.

dotlessbraille.org. (2002). The Nemeth Braille Code for Mathematics. Retrieved from http://www.dotlessbraille.org/index.html

Drümmer, O.,Chang, B. (2012). PDF/UA in a Nutshell. Accessible documents with PDF

Duda, R. O., & Hart, P. E. (1973). Pattern Classification and Scene Analysis. Wiley-Interscience, John Wiley and Sons, New York, vii, 24-30.

Edman, P. (1992). Tactile Graphics; American Foundation for the Blind: New York, 1992.

eGovernment Resource Centre. (2007). Making a PDF accessible. Retrieved from http://www.egov.vic.gov.au/website-practice/usability/pdf-usability-and-accessibility/making-a-pdf-accessible.html

Elzer, S., Schwartz, E., Carberry, S., Chester, D., Demir, S., & Wu, P. (2007). Browser extension for providing visually impaired users access to the content of bar charts on the web. Millersville University, POB 1002, Millersville, PA 17551, USA.

European Blind Union. (2011). Making information accessible for all: Program For Employment And Social Solidarity the Community. Retrieved from
http://www.euroblind.org/resources/guidelines

Felder, R. M., & Brent, R. (2005). Understanding Student Differences. Journal of Engineering Education.

Fitzpatrick D.,& McMullen D. (2008) Distance Learning of Graphically Intensive Material for Visually Impaired Students, ICCHP 219-225

Flowers, J. H. (2005). Thirteen years of reflection on auditory graphing: Promises, pitfalls, and potential new directions: Proceedings of ICAD 05-Eleventh Meeting of the International Conference on Auditory Display, Limerick, Ireland, July 6-9, 2005. Ed. Eoin Brazil. International Community for Auditory Display.

Friedmann, K. (2008). More Americans Are All Ears To Audiobooks. Audio Publishers Association.

Friel, S. N., Curcio, F. R., & Bright, G. W. (2001). Making Sense of Graphs: Critical Factors Influencing Comprehension and Instructional Implications. Journal for Research in Mathematics Education, 32(2), 124-158, Published by: National Council of Teachers of Mathematics. Retrieved from
http://www.jstor.org/stable/749671

Friendly, M. (2008). The Golden Age of Statistical Graphics. Statistical Science, 23(4), 502-535. doi: citeulike-article-id:6334627 doi: 10.1214/08-sts268

Fujiyoshi, A., Suzuki, M., & Uchida, S. (2008). Verification of Mathematical Formulae Based on a Combination of Context-Free Grammar and Tree Grammar. Intelligent Computer Mathematics, 5144, 415-429, Springer Berlin Heidelberg.

Gabel, D. (1998). The complexity of chemistry and implications for teaching. International handbook of science education, 233 ' 248. Boston, MA: Kluwer Academic Publishers.

Garain, U. (2009). Identification of Mathematical Expressions in Document Images: Proceeding of the 10th International Conference on Document Analysis and Recognition, 2009. ICDAR '09.

Garain, U., & Chaudhuri, B. B. (2000). A syntactic approach for processing mathematical expressions in printed documents: Proceeding of the15th International Conference on the Pattern Recognition, 2000.

Gardier, J., Morse, M., Good, W., Jan, J., Roman, C., Appleby, K., & Steendam., M. (1997). Strategies for working with children with cortical visual impairment.

Geoffrey, C., Tabin , M. D., & John , A. (2010). Visual Impairement. The Global Health Education Consortium and collaborating partners.

Goncu, C., & Marriott, K. (2008). Tactile chart generation tool: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, Halifax, Nova Scotia, Canada.

Gildea, R., Millen, J., Gerhart, R., & Sullivan, J. (2014). Duxbury Systems, Inc. Retrieved from http://www.duxburysystems.com/dbt.asp

Good, P., & Hardin, J. W. (2003). Common Errors in Statistics (and How to Avoid Them): Wiley. ISBN 0-471-46068-0.

Greenbacker, C. F., Wu, P., Carberry, S., McCoy, K. F., Elzer, S., McDonald, D. D., & Demir, S. (2011). Improving the accessibility of line graphs in multimodal documents: Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies, Edinburgh, Scotland.

Haralick, R. M. (1994). Document image understanding: geometric and logical layout: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '94.

Harris, P. E. (2001). Finding Files in non-HTML Formats. Retrieved from http://www.oxford-knowledge.com/searching-situations-3-finding-files-in-non-html-formats

Heine, B., & Song, K. A. (2011). On the grammaticalization of personal pronouns. Journal of Linguistics, 47(3), 587-630

Henter, T. (1989). Job Access With Speech(JAWS). American Foundation for the Blind.

Herr, N. (2006). The Sourcebook for Teaching Science. California State University

Highsmith, J. (2001) History: The Agile Manifesto. Agile Alliance.

Hermann, T., Hunt, A., & Neuhoff, J. G. (2011). The Sonification Handbook.Logos Publishing House, Berlin 2011, 586 pages, 1. edition (11/2011) ISBN 978-3-8325-2819-5

Hollabaugh, C. (2002). Embedded Linux : Hardware, Software, and Interfacing (English) 1st Edition: Pearson Education.

Huang, W., & Tan, C. L. (2007). A system for understanding imaged infographics and its applications: Proceedings of the 2007 ACM symposium on Document engineering, Winnipeg, Manitoba, Canada.

Hudson, R. (2004). PDFs and Accessibility. Retrieved from http://usability.com.au/2004/05/pdfs-and-accessibility/

Ina, S. (1996). Computer graphics for the blind. SIGCAPH Comput. Phys. Handicap.(55), 16-23. doi: 10.1145/231674.231677

Ion, P.D.F. (2000). MathML: a Key to Math on the Web?. Mathematical Reviews P. O. Box 8604 Ann Arbor, MI 48107.

James, W., & Gardner, D. (1995). Learning styles: Implications for distance learnining. New directions for Adult and Continuing Education.

Jamin, S. (2013). Balancing Chemical Equations ' Easy Algebraic Method, Creative Commons Attribution 3.0 Unported (CC BY 3.0). Retrieved from http://jaminsantiago.files.wordpress.com/2013/04/balancing-chemical-equations-easy-algebraic-method.pdf

Jayant, C. (2006). A Survey of Math Accessibility For Blind Persons and An Investigation on Text/Math Separation.

Jin, J., Han, X., & Wang Q. (2003). Mathematical formulas extraction: Proceedings of the Seventh International Conference on Document Analysis and Recognition, 2003.

Johnson, D. (2004). Accessible PDF: A Strategic Review. Retrieved from http://www.planetpdf.com/enterprise/article.asp?ContentID= accessible_pdf_a_strategic_re&gid=6118

Kacem, A., Belaid, A., & Ahmed, M. B. (2000). Embedded formulas extraction: Proceedings of the 15th International Conference on Pattern Recognition, 2000.

Kacem, A., Bela'd, A., & Ben Ahmed, M. (2001). Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context. International Journal on Document Analysis and Recognition, 4(2), 97-108. doi: 10.1007/s100320100064

Karshmer, A., & Bledsoe, C. (2002). Access to Mathematics by Blind Students. In K. Miesenberger, J. Klaus & W. Zagler (Eds.), Computers Helping People with Special Needs, 2398,471-476: Springer Berlin Heidelberg.

Karshmer, A., Gupta , G., Geiger, S., & Weaver, C. (1999). Reading and Writing Mathematics. The MAVIS Project, BIT (Behaviour& Information Technology).

Karshmer, A., Guptab, G., & Pontellic, E. (2007). Mathematics and Accessibility: a Survey. University of San Francisco , USA. Retrieved from https://www.utdallas.edu/~gupta/mathaccsurvey.pdf

Karshmer, A. I., & Bledsoe, C. (2002). Access to Mathematics by Blind Students Introduction to the Special Thematic Session: Proceedings of the 8th International Conference on Computers Helping People with Special Needs.

Karshmer ,A ., & Bledsoe, C.(2002).Access to mathematics by blind students". Computers Helping People with Special Needs

Kasia , M. (2013). Agile Vs. Waterfall: Evaluating The Pros And Cons. Retrieved from
https://www.udemy.com/blog/agile-vs-waterfall

Kelley, W. M., & Donnelly, R. (2009). The Humongous Book of Statistics Problems. NewYork: Alpha Books.

Kendrick, D. (2001). George Kerscher: A Pioneer in Digital Talking Books Still Forging Ahead. AFB AccessWorld Magazine.

Kerscher, G. (2003). DAISY is . Retrieved from daisy.org/publications/docs/20040510214528/DAISY-in-Brief_final.htm

Keshari, B., & Watt, S. M. (2007). Hybrid Mathematical Symbol Recognition Using Support Vector Machines. Proceedings of the Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007.

Kise, K., Sato, A., & Iwata, M. (1998). Segmentation of Page Images Using the Area Voronoi Diagram Computer Vision and Image Understanding, 70(3), 370-382. doi:Retrieved from http://dx.doi.org/10.1006/cviu.1998.0684

Kramer, G. (1993). Auditory Display: Sonification, Audification, and Auditory Interfaces: Perseus Publishing.

Kumar, S. (2014). Docx2txt. Retrieved from http://docx2txt.sourceforge.net/

Labahn, G., Lank, E., MacLean, S., Marzouk, M., & Tausky, D. (2008). MathBrush: A System for Doing Math on Pen-Based Devices. Paper presented at the Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on.

Lachowicz, D. (2009). AbiWord. Retrieved from http://www.abisource.com

Larsson, F. (2008). Braille Representation and Mathematics Notation comparison different electronic formats for mathematics and their implications for Braille production. Retrieved from http://www.punktskriftsnamnden.se/filer/pn/Braille_Representation _and_MathematicNotation.pdf

Learning Ally. (2012). What-is-a-print-disability. Retrieved from https://www.learningally.org/support-topic/what-is-a-print-disability/

LEE, A. (2010). The Julius book.

Leith, L. (2006). Inside DAISY. Retrieved from http://data.daisy.org/publications/docs/20070315155100/intro-article 2.html

Levi, Z. (2012). DAISY Player Project. (Final Year Project Thesis), University of Western Australia.

Lin, X., Gao, L., Tang, Z., & Hu, X. (2012). Identification of embedded mathematical formulas in PDF documents using SVM.

Macular Degeneration Foundation. (2012). Low Vision Aids & Technology. Retrieved from http://www.mdfoundation.com.au/resources/1/MDF_LowVisionAids.pdf

Madsen, L., Robertson, W., & Wright, J. (2013). The math tools package. Retrieved from http://texdoc.net/texmf-dist/doc/latex/mh/mathtools.pdf

Digital player features.Retrieved from http://www.maine.gov/msl/outreach/lbph/talkingbooks/digfeatures.htm

Malon, C., Uchida, S., & Suzuki, M. (2006). Support Vector Machines for Mathematical Symbol Recognition. Structural, Syntactic, and Statistical Pattern Recognition. 4109, 136-144: Springer Berlin Heidelberg.

Manktelow, J., & Carlson, A. (2014). Types of Learning Styles. Discover the differences and understand how that influences learning . Retrieved from http://www.learningrx.com/types-of-learning-styles-faq.htm

Matheson, A. (2013). Auditory Interface Design to Support Rover Tele-Operation in the Presence of Background Speech: Evaluating the Effects of Sonification, Reference Level Sonification, and Sonification Transfer Function. Department of Mechanical and Industrial Engineering University of Toronto.

Mayo, P. (2004). Assessment of the impact chemistry text and figures have on visually impaired students' learning, Doctoral Dissertation, Purdue University, West Lafayette.

McCallum, D., Ahmed, K., Jehoel, S., Dinar, S., & Sheldon, D. (2005). The design and manufacture of tactile maps using an inkjet process. Journal of Engineering Design, 16(6), 525-544.
doi: 10.1080/09544820500273946

McCathieNevile, C., & Koivunen, M. R. (2000). Accessibility Features of SVG. Retrieved from
http://www.w3.org/TR/SVG-access/

McMeekin, D., (2010), A software inspection methodology for cognitive improvement in software engineering.Curtin University

Meijer, P. B. L. (1992). An experimental system for auditory image representations. Biomedical Engineering, IEEE Transactions on, 39(2), 112-121. doi: 10.1109/10.121642

Mielke, D. (1996). BRLTTY. Retrieved from
http://mielke.cc/brltty

Melonfire, C. (2006). Understanding the pros and cons of the Waterfall Model of software development. Retrieved from
http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development

Miller, E. G., & Viola, P. A. (1998). Ambiguity and Constraint in Mathematical Expression Recognition: Proceedings of the 15'th National Conference on Artificial Intelligence.

Miner, D. L., Nieman, R., Swanson, A. B., & Woods, M. (2001). Teaching Chemistry to Students with Disabilities: A Manual for High Schools, Colleges, and Graduate Programs 4th Edition. 10-96'.

Murray, I. (2008). Instructional eLearning technologies for the vision impaired (PhD thesis),Curtin University.

Narendra, P.M., & Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. IEEE Trans. Comput. 26, 9 (September 1977), 917-922. http://dx.doi.org/10.1109/TC.1977.1674939

National Center Accessible Information Technology Education University Washington. (2013). How does accessibility differ across operating systems?. Retrieved from
http://accessibletech.org/access_articles/os/osAccess.php

Nazemi, A., Ortega-Sanchez, C., & Murray, I. (2011). Digital Talking Book Player for the Visually Impaired Using FPGAs: Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs.

Nazemi, A., & Murray, I. (2012). A Novel Complete Reading Embedded System for the Vision Impaired: Proceeding of the 3rd Annual International Conference on Computer Science Education: Innovation and Technology CSEIT 2012. Singapore, 2012. (Best Student Paper Award Winner).

Nazemi, A., Murray, I., & Mohammadi, N. (2012). Mathspeak: An Audio Method for Presenting Mathematical Formulae to Blind Students: Proceeding of the Human System Interactions (HSI), 2012.

Nazemi, A., & Murray, I. (2013). Mathematical Formula Recognition and Transformation to a Linear Format Suitable for Vocalization. Journal on computer science and engineering, 5(9).

Nazemi, A., Murray, I., & McMeekin, D.A. (2014), A Method to Implement DAISY Online Delivery Protocol , Proceeding of the 8th International Conference on Telecommunication System, Services, and Application. Bali.

NCAM. (2009). Provide access to graphs for users who are blind or visually impaired.

Retrieved from

http://ncam.wgbh.org/invent_build/web_multimedia/accessible-digital-media-guide/guideline-f-graphs

Neal, K. (2011). Why is OCR at 300 dpi a Standard?. Retrieved from http://scansnapcommunity.com

Noble, S. (2008). The Universal Design for Math Learning: Bridging Technology and Policy Divide Retrieved from https://www.dessci.com/en/reference/articles/lwd07-noble.htm

O'Brein, S. (2012). Optical Character Recognition. (Degree of Bachelor of Science). Worcester Polytechnic Institute.

Pfefferl, J. (2008). Petalinux. Retrieved from http://www.xilinx.com/tools/petalinux-sdk.htm

Pham, N. A., Morrison, A., Schwock, J., Aviel-Ronen, S., Iakovlev, V., Tsao, M.S., & Hedley, D. (2007). Quantitative image analysis of immunohistochemical stains using a CMYK color model. Diagnostic Pathology, 2(1), 1-10. doi: 10.1186/1746-1596-2-8

Plumpe, M., & Meredith, S. (1998).Which Is More Important In A Concatenative Text to speech system - pitch, duration, or spectral discontinuity?: Proceeding of the Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis.

Raman, T. V. (1994). Audio System for Technical Readings. (Doctor of Philosophy), Cornell University.

Raman, T. V. (1995). Emacspeak –A Speech Interface. Cambridge Research Lab,Digital Equipment Corp.

Ramloll, R., Yu, W., Brewster, S., Riedel, B., Burton, M., & Dimigen, G. (2000). Constructing sonified haptic line graphs for the blind student: first step: .Proceedings of the fourth international ACM conference on Assistive technologies, Arlington, Virginia, USA.

Ramus, F. (2003). Developmental dyslexia: specific phonological deficit or general sensorimotor dysfunction? Current Opinion in Neurobiology 13 (2): 212'8. doi:10.1016/S0959-4388(03)00035-7. PMID 12744976.

RC SYSTEMS. (2005).Text to Speech and Voice Synthesis Products. Retrieved from
http://www.rcsys.com/

Reading Rights Coalition. (1989). The definition of print disabled. Retrieved from
http://www.readingrights.org/definition-print-disabled

Rigamonti, M., Bloechle, J. L., Hadjar, K., Lalanne, D., & Ingold, R. (2005). Towards a canonical and structured representation of PDF documents through reverse engineering: Proceedings of the Eighth International Conference on Document Analysis and Recognition, 2005.

Roth, P., Richoz, D., Petrucci, L., & Pun, T. (2001). An audio-haptic tool for non-visual image representation: Proceeding of the Sixth International Symposium on the Signal Processing and its Applications.

Russo, J. (2010). DAISY is the Ticket. Retrieved from
http://www.learningally.org/Blog/AccessandAchievement/1s44/vobId830/

Ryan, D. L. (1986). Modern graphic communications : a CAD approach. Englewood Cliffs, N.J.: Prentice-Hall.

Sagot , B., & Walther, G. (2009). A Morphological Lexicon for the Persian Language.

Sahin, M., & Yorek, N. (2009). Teaching Science to Visually Impaired Students: A Small- Scale Qualitative Study. 6, 19-25.

Sahyun, S., Gardner, J., Gardner, C., & Benton, L. (1998). Audio and Haptic Access to Math and Science - Audio graphs, Triangle, the MathPlus Toolbox, and the Tiger printer. Proceedings of the IFIP:ICCHP '98.

Salkind, N. J. (2006). Statistics for People who (think They) Hate Statistics. University of Kansas: The Excel Edition.

Scadden, L. A. (1991). An overview of technology and visual impairment. Technology and Disability, 11(62).

Schwarzenegger, R. (1991). Making the GUI Talk. BYTE.

Shafait, F. (2009). Document image analysis with OCRopus. Proceeding of the IEEE 13th International Conference of the Multitopic . INMIC 2009.

Shafait, F., Keysers, D., & Breuel, T. (2006). Performance Comparison of Six Algorithms for Page Segmentation. Document Analysis Systems VII , 3872, 368-379, Springer Berlin Heidelberg.

Shafait, F. (2008). Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6), 941-954.

Sedgewick, R. (2010). The Role of the Scientific Method in Software Development, Princeton University.

Silverman, L. (2000). The two-edged sword of compensation: How the gifted cope with learning disabilities. Identifying and meeting the needs of twice exceptional learners, Avocus Publishing Inc., ISBN 189076504X, pp. 153-159.

Simkin, D., & Hastie, R. (1987). An Information-Processing Analysis of Graph Perception. Journal of the American Statistical Association, 82(398), 454-465. doi: 10.1080/01621459.1987.10478448

Spence, I., & Lewandowsky, S. (1991). Displaying proportions and percentages. Applied Cognitive Psychology, 5(1), 61-77. doi: 10.1002/acp.2350050106

Still, M. (2005). The Definitive Guide to ImageMagick (Definitive Guides). Retrieved from http://www.imagemagick.org/script/architecture.php

Stosberg, D. (2006). Odt2txt.py. Retrieved from https://wiki.openoffice.org/wiki/Odt2txt.py

Sullivan, J. (2002). Louis Braille and the Braille System. Retrieved from http://www.duxburysystems.com/braille.asp

Tank, E., & Kerscher, G. (2007). DAISY a beteer way to read a better way to publish: Proceeding of the the DAISY General Meeting.

Theofanos, M. F., & Redish, J. (2003). Bridging the gap: between accessibility and usability. interactions, 10(6), 36-51.doi: 10.1145/947226.947227

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. 2, 45-66. doi: 10.1162/153244302760185243

United Nations. (2006). Convention on the Rights of Persons with Disabilities. Access to Information(Article 21). Retrieved from http://www.un.org/disabilities/convention/conventionfull.shtml.

University of Washington. (2013). How can publishers create accessible math textbooks?.Retrieved from http://www.washington.edu/doit/Faculty/articles?384

Vision Australia. (2012). ABC of Braille. Retrieved from http://www.visionaustralia.org/living-with-low-vision/learning-to-live-independently/about-braille/abc-of-braille

Vision Australia. (2012). What is print disability?. Retrieved from http://www.visionaustralia.org/business-and-professionals/print-accessibility -services/what-is-print-disability

Vystrcil, J., Mikovec, Z., Karsulin, M., Simek, K., Spacek, P., & Slavik, P. (2011). Accessibilty Reaching Every where (How To Create Real Accessible Aria): Proceeding of the AEGIS, Belgium.

Watanabe, T., Yamaguchi, T., & Nakagawa, M. (2012). Development of Software for Automatic Creation of Embossed Graphs. Computers Helping People with Special Needs.7382, 174-181: Springer Berlin Heidelberg.

Waters, K. (2007). What Is Agile? (10 Key Principles of Agile). Retrieved from http://www.allaboutagile.com/what-is-agile-10-key-principles

Wild, G. (2010). A few problems with the concept of accessible PDF. Retrieved from
http://www.gianwild.com/2010

Williams, T., & Kelly, C. (2004). gnuplot 4.4 An Interactive Plotting Program.

Winder, A. (2010). Extending the page segmentation algorithms of the OCRpus documentation layout analysis system. (Master of Science in Computer Science), Boise State University Graduate College.

Wong, K. Y., Casey, R. G., & Wahl, F. M. (1982). Document analysis system. IBM Journal of Research and Development, 26(6), 647-656.

Woods, M. (1996). Working Chemists with Disabilities, Expanding Opportunities in Science. American Chemical Society: Washington, DC, 1996, ISBN 0841235023.

World Health Organisation. (2013). Visual impairment and blindness. Fact Sheet N'282.
Retrieved from
http://www.who.int/mediacentre/factsheets/fs282/en/.

World Health Organization. (2011). Visual impairment and blindness. Fact Sheet N'282.

Wu, H., Krajcik, J.S., & Soloway, E. (2001). Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom. Journal of Research in Science Teaching 38(7), 821-842.

Xenos Group Inc. (2010). The Hidden World of PDF Accessibility.
Retrieved from
http://www.actuate.com/download/Accessibility-WP.pdf

Xiaoyan, L., Liangcai, G., Zhi, T., Baker, J., Alkalai, M., & Sorge, V. (2013). A Text Line Detection Method for Mathematical Formula Recognition: Proceeding of the International Conference on Document Analysis and Recognition (ICDAR) 2013.

Xilinx. (2006). Spartan-3E Starter Kit Board User Guide. 1 Retrieved from http://www.xilinx.com/support/documentation/boardsandkits/ug230.pdf

Yarroch, W. L. (1985). Student understanding of chemical equation balancing, J. Res. Sci. Teach., 22(5), 449'459.

Yonghua, L., Kejun, W., Liqun, T., & Junfei, W. (2007). A Two-dimensional Structure Analysis Method of Printing Mathematical Formula Based On Characteristic Characters: Proceeding of the International Conference on Mechatronics and Automation, 2007. ICMA 2007.

Yu, W., & Brewster, S. (2002). Multimodal virtual reality versus printed medium in visualization for blind people: Proceedings of the fifth international ACM conference on Assistive technologies, Edinburgh, Scotland.

Yu, W., Ramloll, R., & Brewster, S. A. (2001). Haptic Graphs for Blind Computer Users: Proceedings of the First International Workshop on Haptic Human-Computer Interaction.

Zanibbi, R., & Blostein, D. (2012). Recognition and retrieval of mathematical expressions. International Journal on Document Analysis and Recognition (IJDAR), 15(4), 331-357. doi: 10.1007/s10032-011-0174-4

Zhang, X., Huang ,H., Zhang., K. (2009). SKNN Text Categorization Algorithm Based on Semantic Centre: Proceedings of the 2009 International Conference on Information Technology and Computer Science - Volume 01 (ITCS '09), Vol. 1. IEEE Computer Society, Washington, DC, USA, 249-252.
DOI=10.1109/ITCS.2009.57 http://dx.doi.org/10.1109/ITCS.2009.57

Zielinski, K. (2006). Software Engineering: Evolution and Emerging Technologies. Amsterdam: IOS Press. ISBN 1-58603-559-2

Appendix A

The following questions were asked in the first survey to investigate the accessibility of electronic documents for the vision-impaired.

1. Please indicate your gender

o Male

o Female

o Prefer not to say

2. Which range includes your age?

o Under 18

o 18-65

o Over 65

3. How many times per week do you read electronic documents?

o 0-5

o 5-10

o More than 10

4. What do you read most?

o Reference Books

o Recreation Books

o Short Documents e.g. Personal correspondence, Bills, Brochures

5. Which format do you prefer?

o Plain Text

o PDF

o DAISY

o ePub

6. Which factor is most important in choosing a format for electronic documents?

o Navigation

o Bookmarking

o Searching

o Simplicity

o Being Portable

7. How do you access electronic documents? (you may select more than one method.)

o Screen Reader

o Mobile Device

o Kindle
o Computer
o DAISY Player
8. In your opinion, which format is more common?
o Plain Text
o PDF
o Daisy
o ePub
9. What do you believe is your greatest barrier to full access of electronic documents? Please provide a brief description.

Appendix B

Margin Removal

```
#!/bin/bash
in="$1"
ext=$(echo "$in"|sed 's/\./!/g'|sed 's/.*!//g' )
name=$(echo "$in"|sed 's/\./!/g'|sed 's/!.*//g' )
if [[ $ext == "pdf" ]];then
pdftoppm $in ppm convert ppm-1.ppm -threshold 80% $name.png
else
convert $in -threshold 80% $name.png
fi
convert $name.png txt:-|grep -Ev white|sed 's/:.*#/ /g;1d;s/,/ /g
   ' > pixel.txt
cat pixel.txt |awk '{print $1}'|sort -b -k1n,1 >x.txt
cat pixel.txt|awk '{print $2}'|sort -b -k1n,1>y.txt
#Finding page bounding box
xs=$(cat x.txt |awk 'NR==1')
xe=$(cat x.txt |awk 'END{print }')
ys=$(cat y.txt |awk 'NR==1')
ye=$(cat y.txt |awk 'END{print }')
x=$(($xe-$xs))
y=$(($ye-$ys))
convert $name.png -crop $x"x"$y"+"$xs"+"$ys  $name"
   _without_margin.png"
```

Skew Detection and Correction

```
#!/bin/bash
in="$1" ext=$(echo "$in"|sed 's/\./!/g'|sed 's/.*!//g' )
name=$(echo "$in"|sed 's/\./!/g'|sed 's/!.*//g' )
if [[ $ext == "pdf" ]];then
pdftoppm $in ppm
convert ppm-1.ppm -threshold 80% $name.png
```

```
else
convert $in -threshold 80% $name.png
fi
convert $name.png txt:-|grep -Ev white|sed 's/:.*
#/ /g;1d;s/,/ /g' > pixel.txt
cat pixel.txt |awk '{print $1}'|sort -b -k1n,1 >x.txt
cat pixel.txt |awk '{print $2}'|sort -b -k1n,1 >y.txt
xs=$(cat x.txt |awk 'NR==1')
xe=$(cat x.txt |awk 'END{print }')
ys=$(cat y.txt |awk 'NR==1')
ye=$(cat y.txt |awk 'END{print }')
x=$(($xe-$xs)) y=$(($ye-$ys))
convert $name.png -crop $x"x"$y"+"$xs"+"$ys wm.png
convert $name.png -crop $x"x"$y"+"$xs"+"$ys $name"_without_margin
    .mpc"
convert wm.png wm.mpc
convert wm.mpc wm.txt cat wm.txt|grep -Ev white|sed 's/:.*#/ /g;1
    d;s/,/ /g'|awk '{print $1, $2}' > wpixel.txt
height=$(identify -format "%h" $name"_without_margin.mpc")
xs=$(cat wpixel.txt|sort -b -k1n,1|awk 'NR==1'|awk '{print $1}')
ye=$(cat wpixel.txt|sort -b -k2n,2|awk '{print $2}'|awk 'END{
    print}')
yxs=$(cat wpixel.txt|awk '$1=='$xs|awk '{print $2}'|awk '{sum+=$1
    } END { print sum/NR}'|bc -l)
xye=$(cat wpixel.txt|awk '$2=='$ye|awk '{print $1}'|awk '{sum+=$1
    } END { print sum/NR}'|bc -l)
delx=`echo "$xye" | bc -l`
dely=`echo "$height-$yxs"| bc -l` angle=`echo "$dely/$delx" | bc
    -l`
roting=$(echo "scale=3;a($angle)/0.017453293" | bc -l)
#Finding rotation angle using arctang() function
rot=$(($(echo $roting |awk '{printf "%d" ,$1+1}')))
convert wm.png -rotate -$rot $name"_unrotate.png"
#Rotate image of page minus rotating- angle
```

Block Segmentation

```
_vertical()
{
n=$1
height=`convert $n"-hor.mpc" +gravity +repage -format '%[fx:u.h
    -1]' -identify original.mpc`
width=`convert $n"-hor.mpc" +gravity +repage -format '%[fx:u.w
    -1]' -identify original.mpc`
```

```
convert $n"-hor.mpc" txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/g
    ;s/black/0/g'|awk '{print $1,$2,$4}'|awk '{if ( $3!=1) $3=0}{
    print $1,$2,$3}'>erode.txt
max-white-space=$(cat erode.txt|sort -b -k2n,2|awk '$3==0'|awk'{
    a[$2]++}END { for(i in a) print i,a[i]}'|sort -b -k2n,2|awk '
    END{print $1}') )
x=($(cat erode.txt|sort -b -k2n,2|awk '$3==0 && $2=='$max-white-
    space'|sort -b -k1n,1|awk  '{print $1}'|awk 'p{print $1,$1-p,p
    }{p=$1}{if (NR==1)print $1,$1,0}'|awk '$2>1'|awk '{print $1}')
    )
noy=${#x[@]}
if [[ $noy -gt 0 ]];then
for (( e=0;e<$noy;e++ ))
do
l=$(($e+1))
if [[ $e -ne $(($noy-1)) ]] ;then
convert $n-org.mpc -crop $((${x[$l]}-${x[$e]}))"x0+"${x[$e]}"+0"
$n-ver$e.mpc
else
convert $n-org.mpc -crop $((${x[$e]}))"x0+0+0" $n-ver$e.mpc
convert $n-org.mpc -crop $(($width-${x[$e]}))"x0+"${x[$e]}"+0"
$n-ver$[$e+1].mpc
fi done
else
convert $n-org.mpc $n-ver1.mpc
fi
convert $n-ver*.mpc  -background blue-splice 1x3+0+0 +append
    result$n.mpc
}
in="$1"
height=`convert "$in" +gravity +repage -format '%[fx:u.h-1]' -
    identify original.mpc`
convert "$in" -threshold 60% -fuzz 1% -trim +repage -scale 50%
    bin.mpc
convert bin.mpc -morphology dilate:0 diamond dilate.mpc
convert dilate.mpc -morphology erode:6 diamond -clip-mask  bin.
    mpc erode.mpc
max-white-space=$(convert erode.mpc txt:-|sed 's/:.*#/ /g;1d;s/,/
     /g;s/white/1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -
    k2n,2|awk '$3==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i
    ]}'|sort -b -k2n,2|awk 'END{print $2}')
x=($(convert erode.mpc txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white
    /1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|awk '
    $3==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i]}'|sort -b
```

```
    -k2n,2|awk ' '$max-white-space'-$2<5'|sort -b -k1n,1|awk '{
   print $1}'|awk 'p{print $1,p}{p=$1}{if (NR==1) print $1,$1}'|
   awk '$1-$2!=1'|awk '{print $1}'))
nox=${#x[@]}
for (( e=0;e<$nox;e++ )) do
l=$(($e+1))
if [[ $e -ne $(($nox-1)) ]] ;then
convert erode.mpc -crop "0x"$((${x[$l]}-${x[$e]}))"+0+"${x[$e]}
   $e-hor.mpc
convert bin.mpc -crop "0x"$((${x[$l]}-${x[$e]}))"+0+"${x[$e]} $e-
   org.mpc
else
convert erode.mpc -crop "0x"$(($height-${x[$e]}))"+0+"${x[$e]} $e
   -hor.mpc
convert bin.mpc -crop "0x"$(($height-${x[$e]}))"+0+"${x[$e]} $e-
   org.mpc fi done
for (( v=0;v<$(($nox));v++ )) do
_vertical $v
done
convert *-hor.mpc -background red -splice 1x1+0+0 -append horizon
   .png
convert result*.mpc -background blue -splice 1x1+0+0 -append x:
```

Another approach using morophology and finding the largest connected componenet.This method is more accurate but time consuming.

```
#!/bin/bash
infile=$1
i=1
convert $1 -scale 50% original.png
while [[ true ]];do
convert original.png -morphology erode:4 diamond mpc:-|convert
   mpc:- -depth 16 -fill black +opaque white  -write mpr:mask -
   negate  -morphology Distance Euclidean:4 -auto-level  -
   threshold 65534  -mask mpr:mask -morphology Dilate:-1
   rectangle +mask  large.png
info=$(identify -format "%@" large.png)
coords=$(echo $info |sed 's/+/ /g;s/x/ /g')
xs=$(echo  $coords|awk '{print $3}')
ys=$(echo  $coords|awk '{print $4}')
xe=$(echo  $coords|awk '{print $1+$3}')
ye=$(echo  $coords|awk '{print $2+$4}')
convert original.png -crop $info segment$i.txt
if [[ $(cat segment$i.txt|grep -Ev white |wc -l ) -gt 15 ]];then
coords=$xs"␣,␣"$ys"␣"$xe"␣,␣"$ye
```

```
echo $coords  $i>>bbox.dat
convert original.png -draw "fill␣white␣rectangle␣␣$coords" mpc:-| convert mpc:- original.png
i=$[$i+1]
else
break
fi
done
cat bbox.dat|sort -b -k3n,3|sort -b -k1n,1
```

Line Segmentation

```
in="$1"
height=`convert "$in" +gravity +repage -format '%[fx:u.h-1]' -identify original.mpc`
convert "$in" -threshold 60% -fuzz 1% -trim +repage -scale 50% bin.mpc
max-white-space=$(convert bin.mpc txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|awk '$3==1'|awk '{ a[$2]++}END{ for(i in a) print i,a[i]}'|sort -b -k2n,2|awk 'END{print $2}')
x=($(convert erode.mpc txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|awk '$3==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i]}'|sort -b -k2n,2|awk ''$max-white-space'-$2<5'|sort -b -k1n,1|awk '{print $1}'|awk 'p{print $1,p}{p=$1}{if (NR==1) print $1,$1}'|awk '$1-$2!=1'|awk '{print $1}'))
nox=${#x[@]}
for (( e=0;e<$nox;e++ )) do
l=$(($e+1))
if [[ $e -ne $(($nox-1)) ]] ;then
convert bin.mpc -crop "0x"$((${x[$l]}-${x[$e]}))"+0+"${x[$e]} $e-org.mpc
else
convert bin.mpc -crop "0x"$(($height-${x[$e]}))"+0+"${x[$e]} $e-org.mpc
fi
done
convert *-org.mpc -background red -splice 1x1+0+0 -append horizon.png
```

Word Segmentation

```
word_seg(){
```

```
in="$1"
convert "$in" -fuzz 0% -trim +repage -threshold 75% bin.mpc
width=`identify -format "%w" bin.mpc`
height=`identify -format "%h" bin.mpc`
convert bin.mpc -morphology dilate:0 diamond dilate.mpc
convert dilate.mpc -morphology erode:3 diamond -clip-mask bin.mpc
    bbin.mpc
y=($(convert bbin.mpc txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/
   g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k1n,1|awk '$3
   ==1'|awk '{ a[$1]++}END{ for(i in a) print i,a[i]}'|sort -b -
   k1n,1 |awk ''$height'-$2<2'|sort -b -k1n,1|awk '{print $1}'|
   awk 'p{print $1,$1-p}{p=$1}{if (NR==1)print $1,$1}'|awk '$2
   !=1{print $1}'))
noxx=${#y[@]}
convert bin.mpc -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:- -crop $((${y[0]}+1))"x0+0+0" 00-ver.mpc
for (( e=0;e<$(($noxx));e++ )) do
l=$(($e+1))
if [[ $e -ne $(($noxx-1)) ]] ;then
convert bin.mpc -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:- -crop $((${y[$l]}-${y[$e]}+1))"x0+"${y[$e]}"+0
   $e-ver.mpc
ye=${y[$l]}
else
ye=$width convert bin.mpc -shave 0x0 -repage $width"x"$height+0+0
    png:-|convert png:- -crop $(($width-${y[$e]}+1))"x0"${y[$e]}"
   +0" $e-ver.mpc
fi
done
convert $(ls *ver.mpc|sort ) -rotate 90 -background blue
-splice 1x3+0+0 -append -rotate -90 x:
}
```

Character Segmentation

```
character-segmentation(){
in=$1
out=$2
i=1
convert "$1"  -trim +repage  -threshold 80% mpc:-|convert mpc:- -
   morphology dilate:0 diamond   work.mpc
height=`identify  -format "%h" work.mpc`
while true; do
```

```
  segment_position=`convert work.mpc txt:- |grep 000000| sed '1d;
      s/:.*//;s/,/ /g'|sort -b -k1n,1|awk 'NR==1'`
  [ -z "$segment_position" ] && break
echo  $segment_position
convert work.mpc  -fill none -draw "matte $segment_position
   floodfill"  work.mpc
width=`identify  -format "%w" work.mpc``
x1=c$(convert work.mpc txt:-|grep none| sed '1d; s/:.*//;s/,/ /g
   '|sort -b -k1n,1|awk 'NR==1{print $1}')
x2tconvert work.mpc txt:-|grep none| sed '1d; s/:.*//;s/,/ /g'|
   sort -b -k1n,1|awk 'END{print $1}')
x=$[$x2-$x1]
xx=$[$width-$x2]
convert work.mpc -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:- -crop  $x"x0+"$x1"+0" mpc:-|convert mpc:- -fill
   none -opaque black $i-$out
convert work.mpc -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:- -crop  $xx"x0+"$x2"+0" temp.mpc
convert temp.mpc -flatten -trim +repage work.mpc
i=$[$i+1]
done
convert $(ls *$out|sort -b -k1n,1)  -background blue -splice  2
   x2+0+0 +append x:
}
```

Recursive Component Extraction

```
!/bin/bash
#VCE
fin="$1"
name=$(echo "$fin"|sed 's/\./!/g'|sed 's/!.*//g'       )
fb="fbin"$name
convert "$fin" -threshold 60%  fbin$name.gif
width=`identify  -format "%w" fbin$name.gif`
height=`identify  -format "%h" fbin$name.gif`
if [[ $(convert fbin$name.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/
   white/1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k1n,1|
   awk '$3==1'|awk '{  a[$1]++}END { for(i in a)  print i,a[i]}'|
   sort -b -k1n,1|awk ''$height'-$2==0'|awk '{print $1}'|awk 'p{
   print $1,$1-p,p}{p=$1}'|awk '$2>2'|wc -l) -gt 2 ]];then
convert fbin$name.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/
   g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k1n,1|awk '$3
   ==1'|awk '{  a[$1]++}END { for(i in a)  print i,a[i]}'|sort -b
    -k1n,1|awk ''$height'-$2==0'|awk '{print $1}'|awk 'p{print $1
```

```
    ,$1-p,p}{p=$1}'|awk '$2>2'|sort -b -k3n,3| awk '{print $1,$3,
    NR"'$name'","'$fb'",'$width','$height'}'|awk '{print "convert
    "$4".gif -shave 0x0 -repage    " $5"x"$6"+0+0 gif:-|convert
    gif:-  -crop  " $1-$2"x0+"$2"+0  "$3".gif "}'>$fin.sh
convert fbin$name.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/
    g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k1n,1|awk '$3
    ==1'|awk '{  a[$1]++}END { for(i in a)  print i,a[i]}'|sort -b
     -k1n,1|awk ''$height'-$2==0'|awk '{print $1}'|awk 'p{print $1
    ,$1-p,p}{p=$1}'|awk '$2>2'|sort -b -k3n,3| awk '{print $1,$3,
    NR"'$name'"}'|awk '{print "./hce  "$3".gif  "}'>>$fin.sh
chmod +x $fin.sh
echo $fin.sh
./$fin.sh
fi


#HCE
#!/bin/bash
 in="$1"
nam=$(echo "$in"|sed 's/\./!/g'|sed 's/!.*//g'  )
convert "$in"  -threshold 80%    bin$nam.gif
fbi="bin"$nam
width=`identify  -format "%w" bin$nam.gif`
height=`identify  -format "%h" bin$nam.gif`
if [[ $(convert bin$nam.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/
    white/1/g;s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|
    awk '$3==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i]}'|
    sort -b -k2n,2|awk ''$width'==$2'|sort -b -k1n,1|awk '{print
    $1}'|awk 'p{print $1,$1-p,p}{p=$1}'|awk '$2>2'|awk '$1-$3>2'|
    wc -l) -gt 2  ]];then
convert bin$nam.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/g;
    s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|awk '$3
    ==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i]}'|sort -b -
    k2n,2|awk ''$width'==$2'|sort -b -k1n,1|awk '{print $1}'|awk '
    p{print $1,$1-p,p}{p=$1}'|awk '$2>2'|awk '$1-$3>2'|sort -b -
    k3n,3| awk '{print $1,$3,NR"'$nam'","'$fbi'"}'|awk '{print "
    convert "$4".gif  -crop  0x"$1-$2"+0+"$2"   "  $3".gif "}'>$in
    .sh
convert bin$nam.gif txt:-|sed 's/:.*#/ /g;1d;s/,/ /g;s/white/1/g;
    s/black/0/g'|awk '{print $1,$2,$4}'|sort -b -k2n,2|awk '$3
    ==1'|awk '{ a[$2]++}END { for(i in a) print i,a[i]}'|sort -b -
    k2n,2|awk ''$width'==$2'|sort -b -k1n,1|awk '{print $1}'|awk '
    p{print $1,$1-p,p}{p=$1}'|awk '$2>2'|awk '$1-$3>2'|sort -b -
```

```
    k3n,3| awk '{print $1,$3,NR"'$nam'"}'|awk '{print " ./vce  "$3
    ".gif "}'>>$in.sh
chmod +x  $in.sh
./$in.sh
fi
```

MATHSPEAK

```
//g++ mathspeak.c -o mathspeak
//./mathspeak mathml.txt
//MATH.cpp:mainprojectfile.
#include<string>
#include<iostream>
#include<fstream>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
int main(int argc,char*argv[])
{
string frac,nfrac,dfrac,tmp;
string to,from,sum,tmpp,tmppp;
int n1=0,n2=0,n3=0,n4=0;
char cnor[2],cnoc[2];
string p1="",p2="",slash="\\";
string index;
int sw=1;
int length=0,ihead,no_of_line=0;
char *buffer;
string fil,inp,map,title="",temp,text;
int i=0,j=0,l=0;
char com[200];
fil=argv[1];
ifstream is;
is.open(fil.c_str(),ifstream::in);
is.seekg(0,ios::end);
length=is.tellg();
is.seekg(0,ios::beg);
buffer=new char[length];
is.read(buffer,length);
is.close();
inp=buffer;
map=inp;
inp=inp.substr(inp.find("begin")+5,inp.find("end")-inp.find("
    begin")-5);
```

```
while(inp.find("=")!=-1){
no_of_line=no_of_line+1;
inp=inp.substr(inp.find('=')+1);
}
inp=map;
//equation//
if(inp.find("equation")!=-1){
inp=inp.substr(0,inp.find("{split}")-1)+inp.substr(inp.find("{
   spilit}")+6);
inp=inp.substr(inp.find("begin{split")+16,inp.find("end")-inp.
   find("begin{split}")-12);
p1="this␣is␣an";
p2="lines␣equation";}
//gather//
if(inp.find("{gather}")!=-1){
inp=inp.substr(inp.find("{gather}")+8,inp.find("end{gather}")-inp
   .find("begin{gather}")-13);
p1="this␣is␣agathering␣of";
p2="equation";}
//align//
if(inp.find("{align}")!=-1){
inp=inp.substr(inp.find("{align}")+8,inp.find("end{align}")-inp.
   find("begin{align")-13);
p1="there␣are";
p2="align␣equations";}
while(inp.find("leq")!=-1){
                         inp=inp.substr(0,inp.find("leq")-1)+",␣
                            less␣than␣or␣equal␣,␣␣"+inp.substr(
                            inp.find("leq")+3);}
while(inp.find("geq")!=-1){
                         inp=inp.substr(0,inp.find("geq")-1)+",␣
                            greater,␣than␣or,␣equal␣,␣␣"+inp.
                            substr(inp.find("geq")+3);}
while(inp.find("ln")!=-1){
                         inp=inp.substr(0,inp.find("ln")-1)+",␣␣
                            Natural␣logarithms␣,␣␣"+inp.substr(
                            inp.find("ln")+2);}
while (inp.find("CMA(")!=-1){
       n1=0;
          from="";
           sum=inp.substr(inp.find("CMA(")+4);
           n1=sum.find(")");
           from=sum.substr(0,n1);
                   if (inp.find("CMA(")>0){
```

```
                              inp=inp.substr(0,inp.find("CMA("))    +
                                 "␣Centred␣␣Moving␣AVerage␣of,␣"+from
                                 +inp.substr(inp.find("CMA(")+from.
                                 length ()+5);}
                         else{
                               inp="Centred␣␣Moving␣␣Average␣␣␣of,␣"+
                                  from+inp.substr(inp.find("CMA(")+5+
                                  from.length());}
  }
  while (inp.find("MA(")!=-1){
       n1=0;
          from="";
           sum=inp.substr(inp.find("MA(")+3);
           n1=sum.find(")");
           from=sum.substr(0,n1);
                   if (inp.find("MA(")>0){
                          inp=inp.substr(0,inp.find("MA("))     +
                             "␣␣Moving␣AVerage␣of,␣"+from+inp.
                             substr(inp.find("MA(")+from.length
                             ()+4);}
                   else{
                          inp="␣Moving␣␣Average␣␣␣of,␣"+from+inp
                             .substr(inp.find("MA(")+4 +from.
                             length());}
  }
while (inp.find("P(")!=-1){
       n1=0;
          from="";
           sum=inp.substr(inp.find("P(")+2);
           n1=sum.find(")");
           from=sum.substr(0,n1);
                   if (inp.find("P(")>0){
                       if (from.find("|")!=-1){
                                inp=inp.substr(0,inp.find("P("
                                   ))     +"␣␣Probability␣of,␣
                                   "+from.substr(0,from.find("
                                   |"))+",␣Conditional,␣"+from
                                   .substr(from.find("|")+1)+
                                   inp.substr(inp.find("P(")+
                                   from.length ()+3);}
                        else{
                                inp=inp.substr(0,inp.find("P("
                                   ))     +"␣␣Probability␣of,␣
```

```
                                    "+from+inp.substr(inp.find(
                                    "P(")+from.length ()+3);}
                    }
                            else{
                            if (from.find("|")!=-1){
                             inp="␣Probability␣of␣" +from.substr
                                (0,from.find("|"))+",␣Conditional,
                                ␣"+from.substr(from.find("|")+1)+
                                inp.substr(inp.find("P(")+2 +from.
                                length());}
                            else{
                            inp="␣Probability␣of␣" +from+inp.
                               substr(inp.find("P(")+3 +from.
                               length());}
                            }
}
        while (inp.find("={")!=-1){
              n1=inp.find("={");
                  n2=inp.find("}");
                  tmp=inp.substr(n1+2,n2-n1-2);
                  inp=inp.substr(0,inp.find("={")-1)    +",␣
                     Compelement␣of,␣␣" +tmp +inp.substr(inp.find
                     ("={")+2+tmp.length());
                  }
        //quantity
while (inp.find("(")!=-1){
              n1=inp.find("(");
                  n2=inp.find(")");
                  tmp=inp.substr(n1+1,n2-n1-1);
                  //if (tmp.find("+")!=-1 ||tmp.find("-")!=-1){
                          inp=inp.substr(0,inp.find("("))        +
                             ",␣open␣bracket,␣␣" +tmp +",␣␣close␣
                             bracket,␣"+inp.substr(inp.find(")")
                             +1);
            //}
                  //else {
                    //inp=inp.substr(0,inp.find("("))    +", " +
                       tmp +", "+inp.substr(inp.find(")")+1);}
                  }
//sum
        while (inp.find("sum")!=-1){
       n1=0;
           n2=0;
           to="";
```

```
            from="";
            sum=inp.substr(inp.find("sum")+3);
            n1=sum.find("_");
            n2=sum.find("{")+1;
            tmpp=sum.substr(n2);
            n3=0;
            while(tmpp.substr(n3,1)!="}"){
                          n3=n3+1;}
            from=sum.substr(n2,n3+n2-2);
             sum=inp.substr(inp.find("sum")+3);
            n3=0;
            if(inp.substr(inp.find("sum")+6+from.length(),1)=="^")
               {
             sum=sum.substr(sum.find("^"));
                 n1=sum.find("^");
             while(sum.substr(n3,1)!="}"){
                          n3=n3+1;}
 to=sum.substr(n1+2,n3+n1-2);
            inp=inp.substr(0,inp.find("sum")-1)+",␣Sum␣from,␣"+
               from+",␣to,␣"+to+inp.substr(inp.find("sum")+9+to.
               length()+from.length ());}
            else {
                       inp=inp.substr(0,inp.find("sum")-1)+",␣Sum␣
                          from,␣"+from+inp.substr(inp.find("sum")
                          +6+from.length ());}
         }
while (inp.find("prod")!=-1) {
      n1=0;
            n2=0;
            to="";
            from="";
            sum=inp.substr(inp.find("prod")+4);
            n1=sum.find("_");
            n2=sum.find("{")+1;
            tmpp=sum.substr(n2);
            n3=0;
            while(tmpp.substr(n3,1)!="}"){
                          n3=n3+1;}
            from=sum.substr(n2,n3+n2-2);
            sum=inp.substr(inp.find("prod")+4);
            n3=0;
            if(inp.substr(inp.find("prod")+7+from.length(),1)=="^"
               ){
             sum=sum.substr(sum.find("^"));
```

```
                n1=sum.find("^");
            while(sum.substr(n3,1)!="}"){
                        n3=n3+1;}
            to=sum.substr(n1+2,n3+n1-2);
            inp=inp.substr(0,inp.find("prod")-1)+",Production␣from,
               ␣"+from+",␣to,␣"+to+",␣of,␣"+ inp.substr(inp.find("
               prod")+10+to.length()+from.length ());}
            else {inp=inp.substr(0,inp.find("prod")-1)+",␣
                Production␣from,␣"+from+",␣of,"+inp.substr(inp.find
                ("prod")+7+from.length ());}
        }
        //product
while (inp.rfind("frac")!=-1){
                frac=inp.substr(inp.rfind("frac")+4);
         n1=inp.rfind("frac")+4;
                n2=frac.find("}{");
                inp=inp.substr(0,inp.rfind("frac")-1)+inp.substr
                    (n1,n2+1        )+",␣␣over,␣␣"+inp.substr (n1+
                    n2+1);
}
cout<<inp;
//root//
        while (inp.find("sqrt")!=-1){
       n1=0;
            n2=0;
            n3=0;
            n4=0;
            from="";
            sum=inp.substr(inp.find("sqrt")+3);
            n3=sum.find("{");
            n4=sum.find("}");
            tmp=sum.substr(n3+1,n4-n3-1);
        if (sum.find("[")!=-1){
            n1=sum.find("[");
            n2=sum.find("]");
            from=sum.substr(n1+1,n2-n1-1);
            if (tmp.find("+")!=-1 ||tmp.find("-")!=-1){
                    inp=inp.substr(0,inp.find("sqrt"))+",␣root,␣␣"
                        +from+",␣␣of␣open␣bracket,␣␣"+tmp+",␣close␣
                        bracket,␣"+inp.substr(inp.find("sqrt")+8+
                        from.length ()+tmp.length());}
            else {
```

```
            inp=inp.substr(0,inp.find("sqrt"))+", root,  "+from
               +", of "+tmp+inp.substr(inp.find("sqrt")+8+ from
               .length ()+tmp.length());
         }
         }
      else{
            if (tmp.find("+")!=-1 ||tmp.find("-")!=-1){
                  inp=inp.substr(0,inp.find("sqrt"))+", 
                     Square  root of  open bracket,  "+tmp+
                     ", close bracket,  "+inp.substr(inp.
                     find("sqrt")+6+tmp.length());   }
            else{
                  inp=inp.substr(0,inp.find("sqrt"))+", 
                     Square  root of,  "+tmp+inp.substr(inp
                     .find("sqrt")+6+tmp.length());      }
      }
      }
//binom
      while (inp.find("binom")!=-1){
               n1=0;
               n2=0;
               n3=0;
               frac=inp.substr(inp.find("binom")+6);
               n1=frac.find("}");
               nfrac=frac.substr(0,frac.find("}"));
               n2=frac.find("{");
               tmp=frac.substr(n2);
               n3=0;
               while(tmp.substr(n3,1)!="}"){
                        n3=n3+1;}
               if(n3==2)
                dfrac=frac.substr(n2+1,n3+n2-3);
               else
                          dfrac=frac.substr(n2+1,n3+n2-5);
      inp=inp.substr(0,inp.find("binom")-1)+", Binomial of, "+
         nfrac+", choose, "+dfrac+ ", closed Binomial, "+inp.
         substr(inp.find("binom")+9+nfrac.length()+dfrac.length
          ());
      }
// log
      while (inp.find("log")!=-1){
     n1=0;
         n2=0;
         from="";
```

```
            sum=inp.substr(inp.find("log")+3);
         if (sum.find("_")!=-1){
            n1=sum.find("_");
            n2=sum.find("␣")+1;
            from=sum.substr(n1+1,n2-n1-1);
                 inp=inp.substr(0,inp.find("log"))+",␣Logarithm␣
                    based␣,␣"+from+ ",␣for,␣"+inp.substr(inp.find(
                    "log")+4+ from.length ());
         }
         else{
                  inp=inp.substr(0,inp.find("log"))+",␣Logarithm␣
                     based␣ten␣for,␣"+inp.substr(inp.find("log")
                     +4+ from.length ());
         }
         }
         //integral
while (inp.find("int")!=-1){
 n1=0;
            n2=0;
            n3=0;
            to="";
            from="";
            sum=inp.substr(inp.find("int")+3);
            n1=sum.find("_");
            n2=sum.find("^");
            from=sum.substr(n1+1,n2-n1-1);
                    tmpp=sum.substr(sum.find("^")+1);
            if (tmpp.find("{")!=-1){
            while(tmpp.substr(n3,1)!="}"){
                          n3=n3+1;}
                 }
            else{
                    while( tmpp.substr(n3,1)!="␣"){
                            n3=n3+1;}
            }
         to=tmpp.substr(1,n3-1);
  inp=inp.substr(0,inp.find("int")-1)+",␣Integral␣␣from,␣"+from+"
     ,␣to,␣"+to+inp.substr(inp.find("int")+7+to.length()+from.
     length ());
}
//limit
while (inp.find("lim")!=-1){
       n1=0;
            n2=0;
```

```
            n3=0;
            to="";
            from="";
            sum=inp.substr(inp.find("lim")+3);
            n1=sum.find("_");
            n2=sum.find("to");
            n3=sum.find("}");
            from=sum.substr(n1+2,n2-n1-3);
            to=sum.substr(n2+2,n3-n2-2);
            inp=inp.substr(0,inp.find("lim")-1)+" Limit, "+from+" 
               to, "+to+" of,  "+inp.substr(inp.find("lim")+10   +
               to.length()+from.length ());
}
while (inp.find("int")!=-1){
 n1=0;
            n2=0;
            n3=0;
            to="";
            from="";
            sum=inp.substr(inp.find("int")+3);
            n1=sum.find("_");
            n2=sum.find("^");
            from=sum.substr(n1+1,n2-n1-1);
                    tmpp=sum.substr(sum.find("^")+1);
            if (tmpp.find("{")!=-1){
            while(tmpp.substr(n3,1)!="}"){
                          n3=n3+1;}
                 }
            else{
                    while( tmpp.substr(n3,1)!=" "){
                            n3=n3+1;}
            }
         to=tmpp.substr(1,n3-1);
  inp=inp.substr(0,inp.find("int")-1)+", Integral  from, "+from+"
      , to, "+to+inp.substr(inp.find("int")+7+to.length()+from.
      length ());
}
//limit
//matrix
while(inp.find("begin{pmatrix}")!=-1){
            n1=0;
            n2=0;
tmpp=inp.substr(inp.find("begin{pmatrix}")+15,inp.find("end{
   pmatrix}")-inp.find("begin{pmatrix}")-15);
```

```
            tmppp=tmpp;
             while(tmppp.find("dots&")!=-1){
                     tmppp=tmppp.substr(0,tmppp.find("dots&")-1)+
                        tmppp.substr(tmppp.find("dots&")+4);}
             tmp=tmppp.substr(0,tmppp.find(slash+slash));
           while(tmp.find("&")!=-1){
                     n1=n1+1;
                     tmp=tmp.substr(tmp.find("&")+1);}
           while(tmpp.find(slash+slash)!=-1){
                   n2=n2+1;
             tmpp=tmpp.substr(tmpp.find(slash+slash)+1);}
         p1="there␣is␣a␣matrix";
             sprintf(cnoc,"%d",n1+1);
sprintf(cnor,"%d",n2+1);
             tmpp=tmppp;
           tmpp=inp.substr(inp.find("begin{pmatrix}")+15,inp.find("
               end{pmatrix}")-inp.find("begin{pmatrix}")-15);
             while(tmpp.find("&")!=-1){
                         tmpp=tmpp.substr(0,tmpp.find("&"))+",␣␣␣
                            next␣element,␣"+tmpp.substr(tmpp.find(
                            "&")+1);}
          while(tmpp.find("{")!=-1){
                  tmpp=tmpp.substr(0,tmpp.find("{"))+tmpp.substr(
                     tmpp.find("{")+1);}
          while(tmpp.find("}")!=-1){
                  tmpp=tmpp.substr(0,tmpp.find("}"))+tmpp.substr(
                     tmpp.find("}")+1);}
             inp=inp.substr(0,inp.find("begin{pmatrix}")-1)+",␣␣
                matriX␣of,␣␣"+cnor +",␣rows,␣␣and,␣"+cnoc+",␣
                columns,␣"+tmpp;}
          while(inp.find("end{pmatrix}")!=-1){
                  inp=inp.substr(0,inp.find("end{pmatrix}")-1)+inp.
                     substr(inp.find("end{pmatrix}")+12);}
//        `
            while (inp.find("hdotsfor")!=-1){
             n3=0;
                 n4=0;
                     frac=inp.substr(inp.find("hdotsfor")+7);
                 n3=frac.find("{");
                     while(frac.substr(n4,1)!="}"){
                           n4=n4+1;}
                   from=frac.substr(n3,n4-n3-3);
```

```
            inp= inp.substr(0,inp.find("hdotsfor")-1)+",␣␣a␣row
               ␣of,␣"+from+",␣␣elements,␣␣"+inp.substr(inp.find
               ("hdotsfor")+10+from.length());
        }
        //function
   while (inp.find("$")!=-1){
             n1=0;
                n2=0;
            n1=inp.find("$");
                n2=inp.substr(n1+1).find("$");
                tmp=inp.substr(n1+1,n2);
                if (n1>0){
                        inp=inp.substr(0,n1-1)         +"␣
                           function␣of,␣␣" +tmp +inp.substr(n2
                           +2);}
                else{
                inp="␣function␣of␣␣" +tmp +inp.substr(n2+2);
                        }
                        }
        //fcout<<inp;
while (inp.find("|")!=-1){
             n1=0;
                n2=0;
            n1=inp.find("|");
                n2=inp.substr(n1+1).find("|");
                tmp=inp.substr(n1+1,n2);
                //cout<<tmp<<endl;
                if (n1>0){
                        inp=inp.substr(0,n1-1)         +"␣
                           absolute␣value␣of,␣␣" +tmp +inp.
                           substr(n1+2+tmp.length());}
                else{
                inp="␣absolute␣value␣of,␣␣" +tmp +inp.substr(n1
                   +       2+tmp.length());
                }
  }
                //cout<<inp;
while (inp.find("{")!=-1){
          n1=0;
          n2=0;;
          n1=inp.rfind("{");
          n2=inp.substr(n1+1).find("}");
        tmp=inp.substr(n1+1,n2);
```

```
            if((tmp.length()>1) && (tmp.find("+")!=-1 || tmp.find("
                -")!=-1 || tmp.find("_")!=-1)) {
          inp=inp.substr(0,inp.rfind("{"))         +",␣open␣bracket,
              ␣" +tmp +",␣␣close␣bracket,␣␣␣"+inp.substr(n1+n2+2);
                                  }
else {
          inp=inp.substr(0,inp.rfind("{"))         +tmp +inp.substr(
              n1+n2+2);
}
          }
while(inp.find("beta")!=-1){
                          inp=inp.substr(0,inp.find("beta"))+",␣
                              Beetaa␣␣"+inp.substr(inp.find("beta")
                              +4);}
while (inp.find("a_")!=-1 ){
            inp=inp.substr(0,inp.find("a_"))+",␣aee_,␣"+inp.substr(
                inp.find("a_")+2);}
while (inp.find("-a")!=-1 ){
            inp=inp.substr(0,inp.find("-a"))+"-,␣aee,␣"+inp.substr(
                inp.find("-a")+2);}
        while (inp.find("+a")!=-1 ){
                inp=inp.substr(0,inp.find("+a"))+"+,␣aee,␣"+inp.
                    substr(inp.find("+a")+2);}
        while (inp.find("=a")!=-1 ){
                inp=inp.substr(0,inp.find("=a"))+"=,␣aee,␣"+inp.
                    substr(inp.find("=a")+2);}
        while (inp.find("^a")!=-1 ){
                inp=inp.substr(0,inp.find( "^a"))+"^,␣aee,␣"+inp.
                    substr(inp.find("^a")+2);}
        while (inp.find("a+")!=-1 ){
            inp=inp.substr(0,inp.find("a+"))+",␣aee,+␣"+inp.substr(
                inp.find("a+")+2);}
        while (inp.find("a-")!=-1 ){
                inp=inp.substr(0,inp.find("a-"))+",␣aee-,␣"+inp.
                    substr(inp.find("a-")+2);}
        while (inp.find("a=")!=-1 ){
                inp=inp.substr(0,inp.find("a="))+",␣aee=,␣"+inp.
                    substr(inp.find("a=")+2);}
        while (inp.find("a^")!=-1 ){
                inp=inp.substr(0,inp.find("a^"))+",␣aee^,␣"+inp.
                    substr(inp.find("a^")+2);}
        while(inp.find("bigcup")!=-1){
            inp=inp.substr(0,inp.find("bigcup"))+",␣Or,␣␣"+inp.
                substr(inp.find("bigcup")+6);}
```

```
while(inp.find("bigcap")!=-1){
  inp=inp.substr(0,inp.find("bigcap"))+",␣And,␣␣"+inp.
     substr(inp.find("bigcap")+6);}
while(inp.find("neq")!=-1){
 inp=inp.substr(0,inp.find("neq")-1)+",␣not␣equal␣␣␣"+inp
    .substr(inp.find("neq")+3);}
while(inp.find("^")!=-1){
 inp=inp.substr(0,inp.find("^"))+",␣Power␣"+inp.substr(
    inp.find("^")+1);}
while(inp.find("sinh")!=-1){
                inp=inp.substr(0,inp.find("sinh"))+",␣
                   Shin␣␣"+inp.substr(inp.find("sinh")
                   +4);}
while(inp.find("cosh")!=-1){
                inp=inp.substr(0,inp.find("cosh"))+",␣
                   Cosh␣␣"+inp.substr(inp.find("cosh")
                   +4);}
while(inp.find("tanh")!=-1){
                inp=inp.substr(0,inp.find("tanh"))+",␣
                   Than␣␣"+inp.substr(inp.find("tanh")
                   +4);}
while(inp.find("cos")!=-1){
                inp=inp.substr(0,inp.find("cos"))+",␣
                   COSINUS␣␣"+inp.substr(inp.find("cos")
                   +3);}
while(inp.find("cis")!=-1){
                inp=inp.substr(0,inp.find("cis"))+",␣C␣I
                   ␣S,␣␣"+inp.substr(inp.find("cis")+3)
                   ;}
while(inp.find("sin")!=-1){
                inp=inp.substr(0,inp.find("sin"))+",␣
                   Sinus␣␣"+inp.substr(inp.find("sin")
                   +3);}
while(inp.find("pi")!=-1){
                inp=inp.substr(0,inp.find("pi"))+",␣Pi,␣
                   ␣␣"+inp.substr(inp.find("pi")+2);}
while(inp.find("infty")!=-1){
                inp=inp.substr(0,inp.find("infty"))+",␣
                   Infinity␣␣"+inp.substr(inp.find("
                   infty")+5);}
while(inp.find("cdots")!=-1){
                inp=inp.substr(0,inp.find("cdots"))+",␣␣
                   ␣continue,␣until␣␣␣"+inp.substr(inp.
                   find("cdots")+6);}
```

```
    while(inp.find("ldots")!=-1){
                    inp=inp.substr(0,inp.find("ldots"))+",␣␣
                        continue,␣until␣␣␣"+inp.substr(inp.
                        find("ldots")+6);}
    while(inp.find("exp")!=-1){
                    inp=inp.substr(0,inp.find("exp"))+",␣
                        Exponential␣of,␣␣␣"+inp.substr(inp.
                        find("exp")+3);}
    while(inp.find("factor")!=-1){
                    inp=inp.substr(0,inp.find("factor"))+",␣
                        factorial␣␣"+inp.substr(inp.find("
                        factor")+1);}
    while (inp.find("Bigr")!=-1 )    {
            inp=inp.substr(0,inp.find("Bigr")-1)+inp.substr(
                inp.find("Bigr")+4);}
    while (inp.find("Bigl")!=-1 )    {
    inp=inp.substr(0,inp.find("Bigl")-1)+inp.substr(inp.find(
        "Bigl")+4);}
    while (inp.find("quad")!=-1 )    {
            inp=inp.substr(0,inp.find("quad")-1)+inp.substr(
                inp.find("quad")+4);}
            //index
            while(inp.find("_")!=-1){
            if(inp.substr(inp.find("_")+1,1)!="{"){
                    inp=inp.substr(0,inp.find("_")+2)+"␣,␣"+
                        inp.substr(inp.find("_")+2);}
            if (inp.substr(inp.find("_")-1,1).find("a")!=-1){
                index="␣a␣index␣";}
                    else{   index="␣index␣";}
                                    inp=inp.substr(0,inp.find
                                        ("_"))+       index +
                                        inp.substr(inp.find("_
                                        ")+1)+"␣";}
            while(inp.find("&")!=-1){
        inp=inp.substr(0,inp.find("&"))+inp.substr(inp.find("
            &")+1);}
while(inp.find("[")!=-1){
        inp=inp.substr(0,inp.find("["))+inp.substr(inp.find("
            [")+1);}
while(inp.find("]")!=-1){
        inp=inp.substr(0,inp.find("]"))+inp.substr(inp.find("
            ]")+1);}
    while(inp.find("{")!=-1){
```

```
                inp=inp.substr(0,inp.find("{"))+inp.substr(inp.
                   find("{")+1);}
        while(inp.find("}")!=-1){
                inp=inp.substr(0,inp.find("}"))+inp.substr(inp.
                   find("}")+1);}
        while(inp.find(slash+slash)!=-1){
                 inp=inp.substr(0,inp.find(slash+slash))+",␣next␣
                    row,␣␣" +inp.substr(inp.find(slash+slash)+2)
                    ;}
     while(inp.find(slash)!=-1){
        inp=inp.substr(0,inp.find(slash.substr(0,1)))+inp.substr(
           inp.find(slash.substr(0,1))+1);}
        //minus
        while(inp.find("-")!=-1){
                         inp=inp.substr(0,inp.find("-"))+",␣␣
                            minus,␣␣"+inp.substr(inp.find("-")+1)
                            ;}
while(inp.find("+")!=-1){
                         inp=inp.substr(0,inp.find("+"))+",␣␣plus
                            ,␣␣"+inp.substr(inp.find("+")+1);}
        while(inp.find("=")!=-1){
                         inp=inp.substr(0,inp.find("="))+",␣␣␣␣
                            equal,␣␣␣␣"+inp.substr(inp.find("=")
                            +1);}
        system("rm␣text.txt"    );
ofstream os("text.txt");
os <<inp;
os.close();
 cout<<inp;
system ("espeak␣-f␣text.txt␣␣-w␣out.wav" );
        return 0;
}
```

MathGraphReader

```
asec ()
{    echo "scale=5;a(sqrt(($1^2)-1))" | bc -l}
acsc ()
{    echo "scale=5;a(1/(sqrt($1^2)-1))" | bc -l}
sin ()
{    echo "scale=5;s($1*0.017453293)" | bc -l}
cos ()
{    echo "scale=5;c($1*0.017453293)" | bc -l}
tan ()
```

```
{    echo "scale=5;s($1*0.017453293)/c($1*0.017453293)" | bc -l}
csc ()
{    echo "scale=5;1/s($1*0.017453293)" | bc -l}
sec ()
{    echo "scale=5;1/c($1*0.017453293)" | bc -l}
ctn ()
{    echo "scale=5;c($1*0.017453293)/s($1*0.017453293)" | bc -l}
asin (){
    if (( $(echo "$1 == 1" | bc -l) ));then
       echo "90"
    elif (( $(echo "$1 < 1" | bc -l) ));then
       echo "scale=3;a(sqrt((1/(1-($1^2)))-1))/0.017453293" | bc
          -l
    elif (( $(echo "$1 > 1" | bc -l) ));then
       echo "error"
    fi
}
acos ()
{
    if (( $(echo "$1 == 0" | bc -l) ));then
       echo "90"
    elif (( $(echo "$1 <= 1" | bc -l) ));then
       echo "scale=3;a(sqrt((1/($1^2))-1))/0.017453293" | bc -l
    elif (( $(echo "$1 > 1" | bc -l) ));then
       echo "error"
    fi
}
atan ()
{    echo "scale=3;a($1)/0.017453293" | bc -l}
acot ()
{    echo "scale=5;a(1/$1)/0.017453293" | bc -l}
asec ()
{    echo "scale=5;a(sqrt(($1^2)-1))/0.017453293" | bc -l}
acsc ()
{    echo "scale=5;a(1/(sqrt($1^2)-1))/0.017453293" | bc -l}
is_prime ()
{
  local factors
  factors=( $(factor $1) )
  if [ -z "${factors[2]}" ]
then
  return $PRIME
   else
  return $E_NOTPRIME
```

```
   fi
}
prime(){
PRIME=0
E_NOTPRIME=
input=$1

for n in $(seq $input)
do
  if is_prime $n
  then
  if [[ $(echo $input%$n) -eq 0 ]];then
    echo  $n
  fi
fi
done
}
minmax(){
pixel=$1
no=$(cat $pixel|wc -l)
ymax=$(cat $pixel|tail -1|awk '{print $2-208}')
ymin=$(cat $pixel|head -1|awk '{print $2-208}')
i=2
while [[ $i -lt $no ]]
do
min=$(cat $pixel|sort -b -k1n,1|awk 'NR=='$i''|awk '{print $2}')
minpluse=$(cat $pixel|sort -b -k1n,1|awk 'NR=='$i'+1'|awk '{print
    $2}')
minminus=$(cat $pixel|sort -b -k1n,1|awk 'NR=='$i'-1'|awk '{print
    $2}')
xmin=$(cat $pixel|sort -b -k1n,1|awk 'NR=='$i' {print $1-270}'|
   awk '{print ($1*'$x'/540)}')
yymin=$(cat $pixel|sort -b -k1n,1|awk 'NR=='$i' {print $2-208}')
if [[ ($min -le  $minpluse ) && ( $min -le  $minminus ) ]];then
   if [[ $yymin  -eq $ymin ]] ;then
    echo absolute minimum $xmin $yymin>>mm.dat
   else
        echo relative minimum $xmin $yymin>>mm.dat
   fi
else
     if [[ ($min -ge $minpluse ) && ( $min -ge $minminus ) ]];
         then
          if [[ $yymin  -eq $ymax ]] ;then
                echo absolute maximum $xmin $yymin>>mm.dat
```

```
        else
                echo relative maximum $xmin $yymin>>mm.dat
         fi
      fi
fi
i=$(($i+1))
done
cat mm.dat|awk '{print $1,$2,$3, ($4*'$y'/418)}'
}
in=$1
echo "f(x)="$in
#***************shape*********************
if [[ $(echo $in |grep -c "log") -ne 0 ]];then
echo "function" logarithmic functions For small x they are
   negative and at large x they are positive but stay small
else
if [[ $(echo $in |grep -c "sin") -ne 0 ]];then
echo the "function" is Sinusoidal the graph  has a wave shape
amplitude=$(echo $in|sed 's/sin/ sin/g'|sed 's/)/) /g'|awk '{
   print $1}' )
angular_velocity=$(echo $in|sed 's/sin/ sin/g'|sed 's/)/) /g'|awk
    '{hprint $2}'|sed 's/sin(//g'|sed 's/)//g'|sed  's/+/ + /g'|
   sed  's/-/ - /g'|awk '{print $1}'|sed  's/x//g')
if [[ -z $angular_velocity ]];then
angular_velocity=1
fi
 phase_angle=$(echo $in|sed 's/sin/ sin/g'|sed 's/)/) /g'|awk '{
    print $2}'|sed 's/sin(//g'|sed 's/)//g'|sed  's/+/ + /g'|sed
     's/-/ - /g'|awk '{print $NF}')
echo the amplitude affects the height of the wave is $amplitude ,
   the angular velocity affects the width of the wave is
   $angular_velocity and the phase angle shifts the wave left or
   right is $phase_angle
else
degree=$(echo $in |sed 's/\*/o/g'|sed 's/oo-/^m/g'|sed 's/oo/^/g
   '|sed 's/o/\*/g'|sed 's/\^/ ^  /g'|awk '{print $3}'|sed  's/+/
    + /g'|sed 's/-/ - /g'|awk '{ print $1}'|sed 's/m/-/g')
if [[ $(echo $degree |grep -c x) -ne 0 ]];then
echo it is rational
base=$(echo $in |sed 's/\*/o/g'|sed 's/oo-/^m/g'|sed 's/oo/^/g'|
   sed  's/+/ + /g'|sed 's/-/ - /g'|awk '{ print $1}'|sed 's/\^/
   ^  /g'|awk '{print $1}'|sed 's/m/-/g'|sed 's/o/ o /g'|awk '{
   print $NF}')
echo $base
```

```
if [[ ( $(echo  $base|grep -c "/") -eq 0 ) ]];then
if [[ $base -gt 1 ]]; then
 echo the graph  is exponential growth.
fi
else
echo the  is exponential decay.
fi
else
coefno=$(echo $in |sed 's/\*/o/g'|sed 's/oo-/^m/g'|sed 's/oo/^/g
   '|sed 's/o/\*/g'|sed  's/+/ + /g'|sed 's/-/ - /g'|awk '{ print
     NF}')
if [[ -z $degree ]];then
degree=1
echo  the shape is straight line
slope=$(echo $in |sed 's/\*/o/g'|sed 's/oo-/^m/g'|sed 's/oo/^/g'|
   sed  's/+/ + /g'|sed 's/-/ - /g'|awk '{ print $1}'|sed 's/ox/
   x  /g'|awk '{print $1}')
 echo  the slope is $slope
if [[ $slope -gt 0 ]];then
echo the line rises
else
echo  the line falls
fi
else
       if [[ ( $(echo  $degree|grep -c "/") -ne 0 ) ]];then
            echo   x = 0 these functions are all zero. The curves
                are vertical at the origin and as x increases
               they increase but curve toward the x axis.
        else
              if [[ $coefno -eq 1 ]];then
                      if [[ $degree -lt 0  ]];then
                               divzero=1
                               echo  When x = 0 the functions
                                  suffers a division by zero and
                                  therefore are all infinite.
                                  When x is big and positive they
                                   are small and positive.
                               if [[ $(echo $degree%2|bc ) -eq 0
                                  ]];then
                                    echo When x is big and
                                       negative then  y is small
                                       and positive
                                else
```

```
                    echo  When x is big and
                       negative then  y is
                       small  and negative.
                fi
        else
                echo When x = 0 these functions
                   are all zero. When x is big and
                    positive they are all big and
                   positive.
                if [[ $(echo $degree%2|bc ) -eq 0
                   ]];then
                      echo  When x is big and
                         negative then y  is  big
                          and positive
                else
                   echo When x is big and negative
                      then y big and negative.
                fi
        fi
    else
        if [[ $degree -gt  2  ]] ;then
           echo the degree of this graph is   $degree
               .it is polynomial there are   $((
              $degree-1)) possible  maximum and
              minimum
        fi
        if [[ $degree -eq 2  ]] ;then
            echo "the␣function␣is␣Quadratic␣functions
               .the␣␣graphs␣is␣parabola"
          slope=$(echo $in |sed 's/\*/o/g'|sed 's/oo
             -/^m/g'|sed 's/oo/^/g'|sed  's/+/ + /g'|
             sed 's/-/ - /g'|awk '{ print $1}'|sed 's
             /x/ ox  /g'|awk '{print $1}')
             if [[ $slope -gt 0 ]];then
                   echo the parabola opens upward
             else
                     echo  the parabola opens
                        downwards
              fi
        fi
     fi
  fi
fi
```

```
fi
fi
#***************y intercept*****************
if [[ $divzero -ne 1 ]];then
yxo=$(echo $in |sed 's/\*/o/g'|sed 's/oo/^/g'|sed 's/o/\*/g'|sed
   's/x/0/g'|bc)
echo the y-intercept is y equal $yxo
fi
#**************make gnu file***************
echo "set␣term␣png"  >graph.gnu
echo  'set output  "plane.png"'>>graph.gnu
echo 'set  xrang restore'>>graph.gnu
echo 'set  yrange restore'>>graph.gnu
echo 'show xrange'>>graph.gnu
echo 'show yrange'>>graph.gnu
echo plot $in  with lines lc  rgb '"green"' notitle>>graph.gnu
gnuplot  graph.gnu>&range.txt
x1=$(cat range.txt|grep xrange|sed 's/:.*//g'|sed 's/.*\[//g')
x2=$(cat range.txt|grep xrange|grep -o -P '(?<=:).*(?=])')
y1=$(cat range.txt|grep yrange|sed 's/:.*//g'|sed 's/.*\[//g')
y2=$(cat range.txt|grep yrange|grep -o -P '(?<=:).*(?=])')
convert plane.png   -crop 540x418+60+22 png:-|convert png:- out.
   txt
cat out.txt |grep -Ev white|grep -Ev black|grep -Ev grey|sed 's
   /:.*#/ /g;1d'|sed 's/,/ /g'|awk '{print $1,418-$2}'|sort -b -
   k2n,2 >pixel.dat
x=$(echo $x2-$x1|bc)
y=$(echo $y2-$y1|bc)
cat pixel.dat|sort -b -k2n,2|awk '$2==208'|awk '{print $1-270}'|
   awk 'p{print $1-p,$1}{p=$1}{if(NR==1)print $1,$1 }'|awk '$1!=1
    {if ($2>0)printf "%d\n",0.5+$2*'$x'/540;else printf "%d\n"
   ,-0.5+$2*'$x'/540}'|awk '{print "the␣graph␣has␣x-intercept␣at"
   ,$1}'
 if [[ $(cat pixel.dat|sort -b -k2n,2|awk '$1>270 && $2>208'|wc -
    l) -gt 0 ]];then
echo "the␣graph␣is␣located␣at␣quadrant␣1"
fi
if [[ $(cat pixel.dat|sort -b -k2n,2|awk '$1>270 && $2<208'|wc -l
   ) -gt 0 ]];then
echo  "the␣graph␣is␣located␣at␣quadrant␣4␣"
fi
if [[ $(cat pixel.dat|sort -b -k2n,2|awk '$1<270 && $2>208'|wc -l
   ) -gt 0 ]];then
echo "the␣graph␣is␣located␣at␣quadrant␣2"
```

```
fi
if [[ $(cat pixel.dat|sort -b -k2n,2|awk '$1<270 && $2<208'|wc -l
    ) -gt 0 ]];then
echo  "the␣graph␣is␣located␣at␣quadrant␣3"
fi
xx=($(cat pixel.dat|sort -b -k1n,1 |awk '{print $1}'|uniq))
nxx=${#xx[@]}
for ((j=1;j<$nxx;j++))
do
nn=$(cat pixel.dat|awk '$1=='${xx[$j]}''|wc -l)
cat pixel.dat|awk '$1=='${xx[$j]}''|sort -b -k2n,2|head -1>>temp.
    dat
done
yy=($(cat temp.dat|sort -b -k2n,2 |awk '{print $2}'|uniq))
nyy=${#yy[@]}
for ((j=1;j<$nyy;j++))
do
cat temp.dat|awk '$2=='${yy[$j]}''|sort -b -k1n,1|head -1>>ttemp.
    dat
done
minmax ttemp.dat
rm graph.gnu
```

TableReader

Regualar Table

```
#!/bin/bash
convert $1 -threshold 75% bin.png
height=$(identify -format "%h" bin.png)
width=$(identify -format "%w" bin.png)
convert bin.png -alpha off -channel A -transparent white -
    transparent black -negate +channel -write mpr:ORG +clone -
    negate -morphology Erode rectangle:60x1 -mask mpr:ORG -
    morphology Dilate rectangle:60x1 +mask -compose Lighten -
    composite -negate txt:-|grep -Ev '#FFFFFF'| sed '1d; s/:.*/ /g
    ;s/,/ /g'|awk '{print $2,$1}'|sort -b -k1n,1|awk 'p{print $1-p
    ,$1,$2}{p=$1}{if (NR==1)print $1,$1,$2}'|awk '$1>1'|awk '{
    print $2}'>hbox.dat
convert bin.png -rotate 90 -alpha off -channel A -transparent
    white -transparent black -negate +channel -write mpr:ORG +
    clone -negate -morphology Erode rectangle:60x1 -mask mpr:ORG -
    morphology Dilate rectangle:60x1 +mask -compose Lighten -
    composite -negate -rotate -90 txt:-|grep -Ev '#FFFFFF'| sed '1
```

```
   d; s/:.*/ /g;s/,/ /g'|awk '{print $1,$2}'|sort -b -k1n,1|awk '
   p{print $1-p,$1,$2}{p=$1}{if (NR==1)print $1,$1,$2}'|awk '$1
   >1'|awk '{print $2}'>vbox.dat
nv=$(cat vbox.dat|wc -l)
nh=$(cat hbox.dat|wc -l)
nv=$(($nv-1)) nh=$(($nh-1))
for (( j=1 ; j<=$(($nh));j++ )) do
for (( i=1 ; i<=$(($nv));i++ )) do
xs=$(cat vbox.dat|awk 'NR=='$i)
xe=$(cat vbox.dat|awk 'NR=='$i'+1')
ye=$(cat hbox.dat|awk 'NR=='$j'+1')
ys=$(cat hbox.dat|awk 'NR=='$j)
x=$(($xe-$xs))
y=$(($ye-$ys))
convert bin.png -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:-  -crop $x"x"$y"+"$xs"+"$ys $1'colum'$i'row'$j.
   png
done
done
title="this␣table␣contains␣$nh␣rows␣$nv␣colums"
convert $1*col*png  -background green -splice 3x3+0+0 -append  x:
convert $1*col*png  -background green -splice 3x3+0+0 -append
   table$1
```

Irregular Table

```
#!/bin/bash
convert $1  -threshold 70% bin.png
height=$(identify -format "%h" bin.png)
width=$(identify -format "%w" bin.png)
convert bin.png -alpha off -channel A  1.png
convert bin.png  -morphology dilate rectangle:10x1 2.png
convert bin.png  -morphology dilate rectangle:1x10       3.png
convert 1.png 3.png  -compose lighten -composite   ver.png
convert 1.png 2.png  -compose lighten -composite   hor.png
convert hor.png  -morphology erode rectangle:10x1 h.png
convert ver.png  -morphology erode rectangle:1x10 v.png
convert h.png v.png  -compose lighten -composite    txt:-|grep -
   Ev '#FFFFFF'| sed '1d; s/:.*/ /g;s/,/ /g'|awk '{print $1,$2}'|
   sort -b -k1n,1| uniq >point.dat
in=$1
len=$[$(echo  $1|        wc -c)-10]
nam=${in:0:$len}
np=$(cat point.dat|wc -l)
```

```
for (( i=1 ; i<=$[$np];i++ ))
do
xs=$(cat point.dat|awk 'NR=='$i''|awk '{print $1}')
ys=$(cat point.dat|awk 'NR=='$i''|awk '{print $2}')
echo $xs $ys

xe=$(cat point.dat|awk '$1>'$xs' && $2=='$ys''|sort -b  -k1n,1 |
   awk 'NR==1{print $1}')
echo $xe
if [[ -n $xe ]];then
ye=$(cat point.dat|awk '$2>'$ys' && $1=='$xe''|sort -b  -k2n,2 |
   awk 'NR==1{print $2}')
echo $ye
x=$(($xe-$xs))
y=$(($ye-$ys))
if [[ ( $x -ge 3 ) && ( $y -ge 3 ) ]];then
echo $x $y $xs $ys $xe $ye >>bbox.dat
convert bin.png -shave 0x0 -repage $width"x"$height+0+0 png:-|
   convert png:-  -crop   $x"x"$y"+"$xs"+"$ys $nam$i".png"
fi
fi
done
convert $nam"*png" -background blue -splice 1x1+0+0 -append whole
   .png
```

Chart-Legend Segmentation

```
_largest()
{
fil=$1
w=$(identify -format "%w" $fil )
h=$(identify -format "%h" $fil )
convert $fil text.txt
for (( i=3;i<=7;i ++));do
xo=$(echo $w*$i/10|bc)
yo=$(echo $h*$i/10|bc)
col=$(cat  text.txt|grep "$xo,$yo")
if  [[ $(echo $col|grep -c "(255,255,255)"      ) -eq 0 ]];then
xp=$xo
 yp=$yo
break
fi
done
point=$xp','$yp
```

```
convert $fil +repage \( -clone 0 -fill white -colorize 100 \)
\( -clone 0 -fuzz 1% -fill black +opaque white   -fill red -draw
   "color $point floodfill" -alpha off   -fill black -opaque
   white -fill white -opaque red -negate \)
-compose over -composite  +repage  chart-only.gif
info=$(identify -format  "%@" chart-only.png)
coords=$(echo $info |sed 's/+/ /g;s/x/ /g')
xs=$(echo  $coords|awk '{print $3}')
ys=$(echo  $coords|awk '{print $4}')
xe=$(echo  $coords|awk '{print $1+$3}')
ye=$(echo  $coords|awk '{print $2+$4}')
coords=$xs" , "$ys" "$xe" , "$ye
}
in=$1
convert  $in  -scale 100%  -resize 640x480! original.mpc
_blackborder original.mpc
_largest main.mpc
convert main.mpc -draw "fill white rectangle  $coords"  legend-
   only.png
```

ChartRecognition sub-module 2 related to Bar-Chart Reader

```
max=0
nobars=0
for (( i=9;i>=1;i-- ));do
convert horizonntal-vertical-lines-removal_result.mpc -quiet -
   threshold $[$i*10]"%"  bars-only.mpc
convert  bars-only.mpc -morphology dilate rectangle:1x20  -negate
    -trim -negate bars-only.txt
nobars=$(cat bars.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |
   awk '{print $1}'|awk '{ a[$1]++}END { for(i in a) print i,a[i
   ]}'|sort -b -k1n,1|awk 'p{print $1,$1-p,$2}{p=$1}{if (NR==1)
   print $1,$1,$2}'|awk '$2>1 || NR==1 '|wc -l)
if [[ $nobars -gt $max ]];then
max=$nobars
imax=$i
fi
done
convert horizonntal-vertical-lines-removal_result.mpc -threshold
   $[$imax*10]"%"   mpc :-|convert  mpc:- -morphology dilate
   rectangle:1x20  -negate -trim -negate bars.txt
```

```
NOB=$(cat bars.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |sed
    's/,/ /g'|sort -b -k1n,1|awk '{print $1}'|awk '{ a[$1]++}END {
     for(i in a) print i,a[i]}'|sort -b -k1n,1|awk 'p{print $1,$1-
    p,$2}{p=$1}{if (NR==1)print $1,$1,$2}'|awk '$2>1 || NR==1 '|wc
     -l)
cat bars.txt |sed 's/:.*#/ /g;1d;s/,/ /g'|grep black |sed  's/,/
    /g'|sort -b -k1n,1|awk '{print $1}'|awk '{ a[$1]++}END { for(i
     in a) print i,a[i]}'|sort -b -k1n,1|awk 'p{print $1,$1-p,$2}{
    p=$1}{if (NR==1)print $1,$1,$2}'|awk '$2>1 || NR==1 {print $1,
    $3}'>bars.dat
```

Balance Chemical Equation

```
#!/bin/bash
_count(){
nol=$(cat dat.dat|wc -l)
 for ((r=1 ;r<=$nol;r++));do
line=$(cat dat.dat|awk 'NR=='$r'')
no=$(echo $line|wc -w)
 for ((n=1 ;n<=$no;n++));do
m=$(echo $line|awk '{print $'$n'}')
q1=$(echo $line|awk '{print $('$n'+1)}'|grep -o '[0-9]*'
    )
if [[ -n $q1 ]];then
echo $q1 $m>>q.dat
    n=$[$n+1]
    else
echo 1 $m>>q.dat
fi
done
done
nol=$(cat q.dat|wc -l)
 for ((r=1 ;r<=$nol;r++));do
line=$(cat q.dat|awk 'NR=='$r' && $2==")" {print $1}')
if [[ -n $line ]];then
open=$(cat q.dat|awk 'NR<'$r' &&  $2=="("{print NR}'|
   tail -1)
cat q.dat|awk '{if (NR>'$open' && NR<'$r')print $1*'
   $line',$2;else print $1,$2}'>tmp.dat
mv tmp.dat q.dat
fi
done
cat q.dat |sed 's/(/bracket/g;s/)/bracket/g'|grep -Ev "
   bracket">qq.dat
```

```
awk '
  {
    A[$2]++
    B[$2]+=$1
    C[$2]=C[$2] (C[$2]==""?x:",") $1
  }
  END{
    for(i in A) print i, A[i], B[i], C[i]
  }
' OFS='\t' qq.dat
}
lower="abcdefghijklmnopqrst"
re=($(echo $1|sed 's/=.*//g'|sed 's/+/ /g'))
 pr=($(echo $1|sed 's/.*=//g'|sed 's/+/ /g'))
NOR=${#re[@]}
 NOP=${#pr[@]}
 for ((r=0 ;r<$NOR;r++));do
echo ${re[$r]}|sed 's/[a-z]*/&  /g'>>dat.dat
if  [[ $r -eq  $[$NOR-1] ]];then
echo ${lower:$r:1}${re[$r]} "␣=␣␣">>coef.dat
else
echo ${lower:$r:1}${re[$r]} "␣+␣">>coef.dat
fi
done
_count> left.dat
rm q.dat dat.dat qq.dat
 for ((r=0 ;r<$NOP;r++));do
echo ${pr[$r]}|sed 's/[a-z]*/&  /g'>>dat.dat
if  [[ $r -eq  $[$NOP-1] ]];then
echo ${lower:$[$r+$NOR]:1}${pr[$r]}>>coef.dat
else
echo ${lower:$[$r+$NOR]:1}${pr[$r]}"+">>coef.dat
fi
done
cat coef.dat|xargs
_count> right.dat
cat left.dat|sort -b -k1n,1>ll.dat
cat right.dat|sort -b -k1n,1>rr.dat
paste ll.dat rr.dat >eq.dat
cat left.dat|sort -b -k1n,1 |awk '{print $4}'>l.dat
cat right.dat|sort -b -k1n,1 |awk '{print $4}'>r.dat
paste l.dat r.dat|awk '{print $1,$2}'|sed 's/,/ /g'>
   algebra.dat
cat  ll.dat  |awk '{print $1}'> elem.dat
```

```
coefno=$[$NOR+$NOP]
elemno=$(cat elem.dat|wc -l)
eq=""
 for ((e=1 ;e<=$elemno;e++));do

 for ((c=1 ;c<=$coefno;c++));do
  cof=$(cat coef.dat|awk 'NR=='$c'')
  elm=$(cat elem.dat|awk 'NR=='$e'')
        if [[ -n $(echo $cof|grep  $elm)   ]];then
            line=$(echo  $cof|sed 's/+//g;s/=//g'|sed 's
               /[a-z]*/&  /g')
                echo $line|awk '{print $1}'>>eq$e
fi
done
done
 for ((e=1 ;e<=$elemno;e++));do
  eq=""
  left=$(cat eq.dat   |awk 'NR=='$e'{print $2}')
  right=$(cat eq.dat  |awk 'NR=='$e'{print $6}')
  line=$(cat algebra.dat |awk 'NR=='$e'')
  lno=$(cat eq$e|wc -l)
       for ((n=1 ;n<=$lno;n++));do
          eq=$eq$(cat  eq$e|awk 'NR=='$n'')"x"$( echo
             $line|awk '{print $'$n'}')"␣"
                 if [[ $n -eq $left ]];then
                         eq=$eq"␣=␣"
               else
                           if [[ $n -ne $lno ]];then
                                   eq=$eq"␣+␣"
              fi
           fi
done
echo $eq
done
for ((e=1 ;e<=$elemno;e++));do
    elm=$(cat elem.dat|awk 'NR=='$e'')
    echo  $(cat chemicalelements.lib|awk '$1 ~/'$elm'/{
       print $2}'|head -1)>>fullname.dat
done

 cat eq.dat|awk '{print $1,$3,$7}' >full.dat
paste fullname.dat full.dat
rm *dat
```

Joystick

```
#include <sys/ioctl.h>
#include <sys/time.h>
#include <sys/types.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <linux/joystick.h>
int main (int argc, char **argv){
     int quit=0;
        int fd;
       unsigned char axes=3;
       unsigned char buttons=4;
       fd = open(argv[argc - 1], O_RDWR);
        ioctl(fd, JSIOCGAXES, &axes);
    ioctl(fd, JSIOCGBUTTONS,&buttons);
        int *axis;
                int *button;
                int i;
        struct js_event js;
        axis = calloc(axes, sizeof(int));
        button = calloc(buttons, sizeof(char));
                while ( quit==0) {
                        if (read(fd, &js, sizeof(struct js_event)
                           ) != sizeof(struct js_event)) {
                                perror("\njstest: error reading")
                                   ;
                                exit (1);
                        }
                        switch(js.type & ~JS_EVENT_INIT) {
                        case JS_EVENT_BUTTON:
                                button[js.number] = js.value;
                                break;
                        case JS_EVENT_AXIS:
                                axis[js.number] = js.value;
                                break;
                        }
                if (axes) {
                for (i = 0; i < axes; i++)
                            if (axis[i]==32767  && i==0){
```

```
                                                            printf("\n
                                                               ");
                              quit=1;
                                                  printf("NEXT");}
                        if (axis[i]==-32767  && i==0){
                                                    printf("\n");
                              quit=1;
                                               printf("LAST");}
                   if (axis[i]==32767  && i==1){
                                                          printf("\n")
                                                             ;
                              quit=1;
                                                  printf("DOWN");}
                  if (axis[i]==-32767  && i==1){
                              quit=1;
                                                    printf("\n");
                                                  printf("UP");}
       }          }

                       if (buttons) {
                              for (i = 0; i < buttons; i++){
                              if (i==0 && button[i]==1){
                                   quit=1;
                                                             printf("\
                                                                n");
                                  printf("EXIT");}

                              if (i==1 && button[i]==1){
                                 quit=1;
                                                    printf("\n");
                              printf("PAUSE");}
                              if (i==2 && button[i]==1){
                                                    printf("\n");
                               quit=1;
                                                    printf("\n");
                              printf("REW");}
                              if (i==3 && button[i]==1){
                                      quit=1;
                                                    printf("\n");
                                    printf("FOR");}
                       }
                 }
}
                        return 0;
```

```
}
```