

Development of 3D Image Manipulation Software Utilizing the Microsoft Kinect

A report submitted to the School of Engineering and Energy, Murdoch University in partial
fulfilment of the requirements for the degree of Bachelor of Engineering

Prepared By: Aston Ladzinski

Bachelor of Engineering

Department of Engineering and Energy

Supervisor: A/Prof Graeme Cole

Associate Supervisor: Dr. Gareth Lee



MURDOCH
UNIVERSITY

PERTH, WESTERN AUSTRALIA

Abstract

Studying Engineering at Murdoch University enables students to experience many facets of engineering. In the Electrical Engineering based courses there is a vital need to have an understanding of various programming languages and methods and then explore ways in which this knowledge can be utilised. This document describes how the Microsoft Kinect can be utilized to control 3D images, specifically medical images, through the use of programming skills and software development kits.

This project involves learning how the Microsoft Kinect sensor actually works and requires the development of two programs that utilised this sensor and can be easily implemented. The first program is designed to display information that the Kinect sensor is able to detect. By developing this program, it enables a user to quickly gain an understanding of what data is available for manipulation. The second program is designed to manipulate a generic 3D image through the use of a set of gestures initiated by the user. This enables the user to see how the information retrieved from the sensor in the first program can be manipulated for useful purposes, by the creation of algorithms.

The development of these two programs required the learning of a new language. The language that was used is called C#, and it is a complex object orientated language. Through the use of the language in the Microsoft Visual Studios 2010 programming environment, it was possible to create the programs through many iterations of development. The thesis documents how this approach was made, both the method of learning C# and also incorporating and utilising the benefits of C# into the programs.

From the research conducted it was found that creating effective gestures was quite difficult as it required some form of predictive logic, and that incorporating medical 3D imagery was at a completely different level of programming skills. Despite these difficulties an effective, easy to use and modifiable program was developed that will allow future research to continue in this field. Therefore possible future projects and developments are also discussed in order to give the reader an idea of what the sensor is capable of if time and knowledge are readily available.

Acknowledgements

For their guidance, assistance and patience in completing this project, acknowledgements are required for the following people:

Project Supervisor: Associate Professor Graeme Cole
Lecturer, Murdoch University

Associate Project Supervisor: Dr. Gareth Lee
Lecturer, Murdoch University

Mr Will Stirling
Technical Officer, Murdoch University

Crystal Ladzinski and Evie Joy Ladzinski
My wife and daughter

Table of Contents

Abstract.....	1
Acknowledgements.....	2
Table of Figures.....	5
List of Tables	6
List of Appendices	7
Terminology and Acronyms	8
Chapter 1: Introduction	10
1.1 Project Background and History	10
1.2 Project Scope	10
1.3 Project Objectives	11
1.4 Project Revisions	13
1.5 Thesis Structure	14
Chapter 2: Technical Introduction to the Microsoft Kinect.....	15
2.1 Overview of the Microsoft Kinect.....	16
2.1.1 The RGB Camera	17
2.1.2 The Depth Sensor.....	18
2.1.3 The Microphone Array.....	20
2.1.4 The Tilt Motor	22
2.2 Microsoft Kinect Processor	23
2.2.1 Microsoft Kinect Communication	25
2.3 Microsoft Kinect Limitations.....	26
Chapter 3: Introduction to C# and Microsoft Visual Studios 2010.....	27
3.1 Overview of C#.....	27
3.1.1 Learning of C#	28
3.1.2 Utilisation of C# for Microsoft Kinect	29

3.2 Overview of Microsoft Visual Studios 2010.....	31
3.2.1 Learning of Microsoft Visual Studios 2010	32
3.2.2 Incorporating C# in Microsoft Visual Studios 2010	32
Chapter 4: Development of the Microsoft Kinect Interface Program	34
4.1 Original Idea	34
4.2 The Development Stages	35
4.3 The Project Outcome	38
Chapter 5: Development of the Microsoft Kinect Image Manipulation Program	40
5.1 Original Idea	40
5.2 Specific Requirements	42
5.3 The Development Stages	46
5.4 The Project Progress	49
5.5 Program Improvements	50
Chapter 6: Outcomes of the Thesis	52
6.1 Overall Project Outcome.....	52
6.2 Personal Outcomes	53
6.3 Project Difficulties.....	55
Chapter 7: Concluding Remarks.....	57
7.1 Conclusion.....	57
7.2 Future Projects.....	58
Bibliography	61
Appendices.....	65

Table of Figures

Figure 1: Microsoft Kinect Sensor [18]	15
Figure 2: Kinect Component Identification with Non-Structural casing removed [1].....	16
Figure 3: RGB Image produced by the Microsoft Kinect RGB Camera [22].....	17
Figure 4: Image of the Infra-red light projected by the Kinect sensor [23].....	18
Figure 5: Depth Image produced by the calculations completed by the Microsoft Kinect Depth Sensor [23]	19
Figure 6: Microsoft Kinect diagram with Microphone array displayed [20]	20
Figure 7: Diagram representing the ability of the Microsoft Kinect to localize sound [20]	21
Figure 8: Kinect Tilt Motor removed from Kinect [18]	22
Figure 9: Kinect Processor Schematic Diagram [18]	23
Figure 10: Kinect Skeleton Data and Joint Information [3]	24
Figure 11: Kinect SDK Architecture [3].....	25
Figure 12: Kinect Sensor ideal positioning and range diagram [2]	26
Figure 13: Layout of the depth bits received from the Microsoft Kinect [1].....	36
Figure 14: Kinect Data and Display Manipulation Program.....	38
Figure 15: Kinect Image Manipulation Program Interaction Panel	49

List of Tables

Table 1: Required Gesture Types for Kinect Image Manipulation Program.....	42
Table 2: Final Gesture Types for Kinect Image Manipulation Program.....	44

List of Appendices

- A1: Code and Comments for Microsoft Kinect Interface Program
- A2: Code and Comments for Microsoft Kinect Image Manipulation Program
- A3: DICOM MRI Brain Image

Terminology and Acronyms

API: Application programming interfaces are interfaces that allow different methods of software to effectively communicate. An API is required in order for the Kinect to be able to function in Windows.

C#: An object orientated programming language developed by Microsoft that was used to develop the programs within this project.

C++: An object orientated programming language that was intended to be used initially for the programs that were developed within this project.

CMOS: Complementary metal oxide semiconductor is the most popular method of capturing image information digitally. It uses MOSFET type transistors to be able to build logic circuits that can sense information in the form of a camera, such as is used in the Microsoft Kinect.

DICOM: A world wide convention for storing medical images in a specific format, it stands for Digital Imaging and Communications in Medicine.

HMI: A Human Machine Interface is software or hardware that allows a user to interact with a machine. This can be as simple as a button or as complex as gesture or voice recognition.

ICSE: An engineering major available at Murdoch University referred to as Industrial Computer Systems Engineering.

IDE: An integrated development environment that allows a user to develop various programs or applications with various tools, such as was used in this project in the form of Microsoft Visual Studio 2010.

LabView: A graphical based programming language developed by National Instruments specifically for the use of engineers and scientists for the initial purpose of data acquisition.

Matlab: A text structured programming language specialising in matrices and mathematics and widely used for mathematical modelling in academia and industry.

MKIMP: Microsoft Kinect Image Manipulation Program developed in this project for the purpose of manipulating 3D images.

MKIP: Microsoft Kinect Interaction Program developed for the use of understanding data produced by the Microsoft Kinect.

MRI: Magnetic Resonance Imaging is a popular form of medical imaging as it can distinguish between tissue types without the need for exposing the patient to unnecessary dangerous radiation.

NUI: Natural User Interface is an old library developed by Microsoft for the Microsoft Kinect, however recently been superseded by the more direct Microsoft.Kinect library.

Prime Sense: The Company who designed the technology behind the Kinect and created the processor that can analyse the position of a body and its comprising parts.

RGB: Refers to the Red Green Blue type camera sensor that is part of the Microsoft Kinect sensor system.

SDK: Software Development Kit used to make programming easier by allowing a programmer to utilise already available libraries, definitions or functions as is the case with the Microsoft Kinect SDK.

VGA: Video Graphics Array refers to the resolution of 640 x 480 pixels that is available with the Microsoft Kinect Cameras. It also refers to the 8-bit colour palette of 256 different distinguishable colours per pixel.

WPF: Windows Presentation Foundation is part of the .NET framework and enables user to build software by acting as a link between the user and the control item, such as what is required when programming the RGB display of the Kinect.

XAML: Extensible Application Mark-up Language is a programming language required for WPF, which was used to create a graphical user interface. It is declarative, meaning it is not procedural.

.NET: .NET or 'dot-net' is a framework developed by Microsoft to aid in creating object orientated programming abilities to internet applications, and is used extensively in Microsoft Visual Studios 2010.