Vision Enabled Computer Games: Implementation of a Virtual Badminton Trainer

THESIS PRESENTED
BY
I. J. P. WIJESENA

SUPERVISED BY

Dr. C. R. De SILVA

This thesis was submitted to the
Department of Computer Science and Engineering
of the
University of Moratuwa Sri Lanka
In partial fulfilment of the requirements for the
Degree of Master of Science

Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
December, 2005

Declaration

The work submitted in this thesis is the result of my own investigations, except where it is otherwise stated.

It has not already been accepted in substance for any degree, and also is not being concurrently submitted for any other degree.

.....

I. J. P. Wijesena

(Candidate)

.....

Dr. C. R. De Silva

(Supervisor)

Abstract

Vision is a critical component in human-to-human communication. The richness of face-to-face communication is far greater than that of voice or text. But if we consider way humans and computers are interacting visually, the use of artificial or mechanical means is evident. Though, the human predominantly reacts visually, the computer hardly does so, except for producing its output. This research aims at bridging this gap by imparting visual human computer interaction. Even though the material described in this thesis concentrate mainly on computer games, the proposed methods are applicable to many other application areas, too.

A camera mounted in front of a player will capture the motion of the player. Information extracted from this video stream can then be used to bring the human player into the virtual playing field. The virtual shuttlecock will be generated virtually according to the parameters defined in the computer. The virtual playing field will be projected on a large screen in front of the human player.

The racket detection algorithm is based on colour and shape descriptors. After extracting the racket it will be analyzed to identify its orientation. Then it will be combined with previous badminton racket positions, to find out its velocity in order to detect its collision and its outcome with the virtual shuttlecock.

Dedicated to
game developers

Acknowledgement

I take pleasure in thanking Dr. C. R. De Silva, for supervising this project and using his valuable time to instruct and guide me throughout my M.Sc. at the University of Moratuwa. I would also like to thank Dr. D. P. Thrishantha Nanayakkara, my external supervisor for his valuable comments.

I would like to express my sincere thanks to Dr. Ashok Peiris and Dr. Sanath Jayasena, Heads of the Department of Computer Science and Engineering, who gave me the permission to use departmental resources for this work. I am also grateful to Prof. (Mrs.) N. Rathnayake and Dr. N. Munasinghe, Directors of postgraduate studies for helping me in many ways. Thanks also to Dr. Gihan Dias, Research Coordinator of the Department of Computer Science and Engineering for his valuable contribution to my work and also encouraging me to do an M. Sc. degree.

Thanks are also due to my fellow research students at the research lab – Mr. Narada Wickramage, Mr. Kutila Gunasekera, Mr. Sumith Bandula, Mrs. Pradeepa Gurusinghe, and Mr. D.A.B.C. Disanayake. I am very much grateful for their encouragement, assistance and the friendly environment they maintained in the lab. Special thanks are definitely due to Mr. Yoo Taekyoon the Korean volunteer, working in the department of Computer Science and Engineering under the Korea International Cooperation Agency for this assistance in video taping and photographing. I would also wish to thank Mr. G. H. A. Perera who edited this thesis.

I would also wish to thank all members of the staff of the Department of Computer Science and Engineering and Network Administrators of LEARN who helped me in numerous ways.

Finally I wish to acknowledge the financial support received under the Science and Technology Personnel Development Project SRI (SF) 1535 by the Asian Development Bank to carry out the work described in this thesis.

Table of content

Declaration

Abstract

Acknowledgement

Table of content

List of figures

List of tables

Symbols and abbreviations

Chapter 01

1.Introduction 1

1.1 Making computers to 'see' the external world 1

1.2 Development to date 2

1.3 Motivation 2

1.4 Major contribution of this thesis 3

1.5 Outline of the thesis 4

Chapter 02

2.Review of the state of the art 5

2.1 Early developments 5

2.2 Windows, icons, menus and pointing devices (WIMP) 5

2.2.1 Advantages and disadvantages of WIMP interfaces 6

2.2.2 Post WIMP technologies 7

2.3 Vision based human computer interaction 7

2.3.1 Head tracking 8

2.3.2 Face and facial expression recognition 9

2.3.3 Eye tracking 10

2.3.4 Gesture Recognition 10

2.4 Computer games 11

2.5 Smart games 12

2.5.1 Face tracking and games 12

2.5.2 Gesture recognition and video games 14

2.5.3 Fitness games 14

2.5.4 Sports training 15

2.5.5 Webcam based games 16

2.5.6 EyeToy® 17

Chapter 03

3.Vision based input for computer games: use of colour and shape for object detection 19

3.1 Use of colour and shape for object detection 20

3.2 Colour based object detection 22

3.3 Use of shape descriptors 28

3.3.1 Simple shape descriptors 28

3.3.2 Rapid calculation of perimeter 30

3.3 Combined algorithm to detect objects 32

3.4 Real-time extraction of image frames from a video stream 34

3.5 Microsoft® DirectShow™ 34

3.5.1 Filters 35

3.5.2 Filter graph 35

3.5.3 Filter graph manager 35

3.5.4 Basic filter graph 36

3.5.5 Filter graph to capture frames 36

3.5.6 Sample grabber filter 37

3.6 Design and implementation 37

3.6.1 Frame capturing procedure 39

3.7 Labelling algorithm 40

3.7.1 First pass 40

3.7.2 Second pass 41

3.8 Optimized labelling algorithm 41

3.8.1 The effect of skipping the second pass 43

3.9 Implementation of the algorithm 44

3.9.1 The duplicate buffer 44

3.9.2 First pass 44

3.9.3 Second pass 45

Chapter 04

4 Creating the virtual playing field 49

4.1 The projections 49

4.1.1 Picking a random location 50

4.1.2 The projection velocity 51

4.2 The shuttlecock 53
4.2.1 The shuttlecock construction 53
4.2.2 Colour change of the shuttlecock based on its location 55
4.3 Tracking of the badminton racket 56
4.4 Effects of lighting 58
4.5 Orientation of the badminton racket and collision of the shuttlecock with the badminton racket 59
4.5.1 Game initialization and calibration 59
4.5.2 Rotation of the badminton racket 62
4.6 Badminton racket velocity 63
4.6.1 X and Y direction velocity 63
4.6.2 Z direction velocity 63
4.7 Problems in detecting the Bender Angle Direction 66
4.7.1 Eliminating multiple possibilities 66
4.7.2 Changes in bender angle 68
4.7.3 Calculating the bender angle 68
4.8 Problems in detecting the twister angle direction 69
4.8.1 Left of the base gravity 69
4.8.2 Right of the base gravity 70
4.8.3 Changes in the twister angle 70
4.8.4 Calculating the twister angle 71
4.9 Collision with the virtual shuttlecock 72
4.10 Shuttlecock velocity after the collision 73
5.**Discussion** 75
6.**Conclusion and future work** 79
7.**References** 82

List of figures

Figure 1-1: Basic diagram of the system 3
Figure 2-1: CAMSHIFT-based face tracker used to play Quake 2 13
Figure 2-2: Using gesture recognition to play video games 14
Figure 2-3: A participant playing the Shadow Boxer game 15
Figure 2-4: Test setup, showing camera (1), camera view projected on a screen (2), user (3) and test instructor (4) 16
Figure 2-5: Webcam pong by Eyekanoid Game 17
Figure 2-6: Player playing EyeToy game 17
Figure 3-1: Overall setup of the system 19
Figure 3-2: Basic algorithm for colour base object detection applied to test image 22
Figure 3-3: Flow chart of basic algorithm for colour based object detection 23
Figure 3-4: New algorithm applied to red badminton racket shape (a) Original image (b) red colour detected (c) image filled (d) border drawn around racket (e) combined image 24
Figure 3-5: Algorithm applied to an image with a red badminton racket without editing (a) original image with a red badminton racket (b) red colour detected 24
Figure 3-6: Test image with a blue badminton racket (a) original image with blue badminton racket (b) blue colour detected 25
Figure 3-7: Images taken by a digital video camera 25
Figure 3-8: White sheet of paper pasted on the face of the badminton racket 26
Figure 3-9: Images taken by a digital video camera with white paper pasted on the face of the badminton racket 26
Figure 3-10: LED bulbs used in the test 27
Figure 3-11: Test images with lit LED bulbs (Taken by a digital video camera) 27
Figure 3-12: Test image of small 9V bulbs (Taken by a digital video camera) 27
Figure 3-13: The blue light created using blue cellophane paper (Taken by a digital video camera) 28
Figure 3-14: Finding perimeter - Ideal method 29
Figure 3-15: Finding perimeter - approximation 30
Figure 3-16: Combined algorithm 32

Figure 3-17: Outcome of the algorithm, when applied to test images (Taken by a digital video camera) (a) Original image (b) intermediate image (c) badminton racket detected and painted with different colour 33
Figure 3-18: A filter graph created using a “Filter Graph Editor” to play a movie file 35
Figure 3-19: Filter graph used to capture frames 36
Figure 3-20: UML diagram for extracting frames 38
Figure 3-21: Steps involved in capturing a frame 39
Figure 3-22: 4-connectivity 40
Figure 3-23: Time taken by the same application removing label collisions and without removing label collisions 42
Figure 3-24: The adverse effect of skipping the second pass (Image taken by digital video camera) 43
Figure 3-25: Duplicate buffer 44
Figure 3-26: First pass settings 45
Figure 3-27: Overall methodology for second pass 45
Figure 3-28: Stage selection algorithm 46
Figure 3-29: Stage - I label setting 46
Figure 3-30: Stage - II label setting 46
Figure 3-31: Stage - III label setting 47
Figure 3-32: Stage - IV label setting 48
Figure 4-1: Hit window 49
Figure 4-2: 3D view of the coordinate system 50
Figure 4-3: Travelling distance 51
Figure 4-4: The projection 52
Figure 4-5: Parts of the shuttlecock 54
Figure 4-6: Interacting with the shuttlecock 55
Figure 4-7: Colour of the shuttlecock based on its location 56
Figure 4-8: Tracking successful after doubling the prediction area 57
Figure 4-9: Complete miss of the badminton racket even with doubled prediction area 57
Figure 4-10: Lights used by the system 58
Figure 4-11: Base Gravity 60
Figure 4-12: Identifying the major axis 60

Figure 4-13: Difficult to detect the major axis..... 61
Figure 4-14: Circular game play..... 61
Figure 4-15: Rotation of the badminton racket..... 62
Figure 4-16: Rotation of the badminton racket..... 62
Figure 4-17: X and Y direction speed..... 63
Figure 4-18: Badminton racket moved forward without twisted or bent..... 64
Figure 4-19: Badminton racket moved forward with bent but not twisted..... 64
Figure 4-20: Badminton racket moved forward without bent but twisted..... 65
Figure 4-21: Twisted and bent movement..... 65
Figure 4-22: Badminton racket length based on its location..... 66
Figure 4-23: Problems in detecting the bender angle..... 67
Figure 4-24: Forcing shuttlecock upward..... 67
Figure 4-25: Bender angle change during last consecutive frames..... 68
Figure 4-26: The bender angle..... 68
Figure 4-27: Problems in detecting the twister angle..... 69
Figure 4-28: Left of the base gravity..... 69
Figure 4-29: Right of the base gravity..... 70
Figure 4-30: Changes in the twister angle..... 70
Figure 4-31: Calculating the twister angle..... 71
Figure 4-32: Rotation of the badminton racket-1..... 72
Figure 4-33: Rotation of the badminton racket-2..... 72

List of tables

Table 3-1: Time taken by the same application removing label collisions and without removing label collisions 42
Table 4-1: Projection system values 50
Table 4-2: 3D view variables 50
Table 4-3: Shuttlecock parameters 53
Table 4-4: Variables used in shuttlecock creation 54
Table 4-5: Colour change of the shuttlecock based on its location 55
Table 4-6: Values used in finding Z-axis speed 66
Table 4-7: The bender angle calculation parameters 68
Table 4-8: The twister angle calculation parameters 71

Symbols and abbreviations

WIMP	Windows, Icons, Menus and pointing devices
FSA	Finite-state automata
GUI	Graphical user interface
SDK	Software development kit
AVI	Audio-video interleaved
MPEG	Moving Picture Experts Group
WDM	Windows Driver Model
COM	Component Object Model