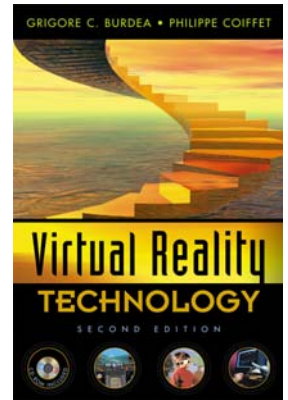


G. Burdea and P. Coiffet
***Virtual Reality Technology* – 2nd Edition,**
Wiley, New York, 2003
© Wiley, 2003



Textbook Table of Contents

Foreword

Preface

TOC

1. Introduction

- 1.1 The Three *I*'s of Virtual Reality
- 1.2 A Short History of Early Virtual Reality
- 1.3 Early Commercial VR Technology
- 1.4 VR Becomes an Industry
- 1.5 The five Classic Components of a VR System
- 1.6 Review Questions
- References

2. Input Devices: Trackers, Navigation, and Gesture Interfaces

- 2.1 Three-Dimensional Position Trackers
 - 2.1.1 Tracker Performance Parameters
 - 2.1.2 Mechanical Trackers
 - 2.1.3 Magnetic Trackers
 - 2.1.4 Ultrasonic Trackers
 - 2.1.5 Optical Trackers
 - 2.1.6 Hybrid Inertial Trackers
- 2.2 Navigation and Manipulation Interfaces
 - 2.2.1 Tracker-Based Navigation/Manipulation Interfaces
 - 2.2.2 Trackballs
 - 2.2.3 Three-Dimensional Probes
- 2.3 Gesture Interfaces
 - 2.3.1 The Pinch Glove
 - 2.3.2 The 5DT Data Glove
 - 2.3.3 The DidjiGlove
 - 2.3.4 The CyberGlove
- 2.4 Conclusion
- 2.5 *Review Questions*
- References

3. Output Devices: Graphics, Three-Dimensional Sound, and Haptic Displays

- 3.1 Graphics Displays

- 3.1.1 The Human Visual System
- 3.1.2 Personal Graphics Displays
- 3.1.3 Large-Volume Displays
- 3.2 Sound Displays
 - 3.2.1 The Human Auditory System
 - 3.2.2 The Convolvotron
 - 3.2.3 Speaker-Based Three-Dimensional Sound
- 3.3 Haptic Feedback
 - 3.3.1 The Human Haptic System
 - 3.3.2 Tactile Feedback Interfaces
 - 3.3.3 Force Feedback Interfaces
- 3.4 Conclusion
- 3.5 *Review Questions*
- References

4. Computing Architectures for VR

- 4.1 The Rendering Pipeline
 - 4.1.1 The Graphics Rendering Pipeline
 - 4.1.2 The Haptics Rendering Pipeline
- 4.2 PC Graphics Architecture
 - 4.2.1 PC Graphics Accelerators
 - 4.2.2 Graphics Benchmarks
- 4.3 Workstation-Based Architectures
 - 4.3.1 The Sun Blade 1000 Architecture
 - 4.3.2 The SGI InfiniteReality Architecture
- 4.4 Distributed VR Architectures
 - 4.4.1 Multi-pipeline Synchronization
 - 4.4.2 Co-located Rendering Pipelines
 - 4.4.3 Distributed Virtual Environments
- 4.5 Conclusion
- 4.6 *Review Questions*
- References

5. Modeling

- 5.1 Geometric Modeling
 - 5.1.1 Virtual Object Shape
 - 5.1.2 Virtual Object Appearance
 - 5.2 Kinematics Modeling
 - 5.2.1 Homogeneous Transformation Matrices
 - 5.2.2 Object Position
 - 5.2.3 Transformation Invariants
 - 5.2.4 Object Hierarchies
 - 5.2.5 Viewing the Three-Dimensional World
 - 5.3 Physical Modeling
 - 5.3.1 Collision Detection
 - 5.3.2 Surface Deformation
-

- 5.3.3 Force computation
- 5.3.4 Force Smoothing and Mapping
- 5.3.5 Haptic Texturing
- 5.4 Behavior Modeling
- 5.5 Model Management
 - 5.5.1 Level-d-Detail Management
 - 5.5.2 Cell Management
- 5.6 Conclusion
- 5.7 *Review Questions*
- References

6. VR Programming

- 6.1 Toolkits and Scene Graphs
- 6.2 WorldToolKit
 - 6.2.1 Model Geometry and Appearance
 - 6.2.2 The WTK Scene Graph
 - 6.2.3 Sensors and Action Functions
 - 6.2.4 WTK Networking
- 6.3 Java 3D
 - 6.3.1 Model Geometry and Appearance
 - 6.3.2 The Java 3D Scene Graph
 - 6.3.3 Sensors and Behaviors
 - 6.3.4 Java 3D Networking
 - 6.3.5 WTK and Java 3D Performance Comparison
- 6.4 General Haptics Open Software Toolkit
 - 6.4.1 GHOST Integration with the Graphics Pipeline
 - 6.4.2 The GHOST Haptics Scene Graph
 - 6.4.3 Collision Detection and Response
 - 6.4.4 Graphics and PHANToM Calibration
- 6.5 PeopleShop
 - 6.5.1 DI-Guy Geometry and Path
 - 6.5.2 Sensors and Behaviors
 - 6.5.3 PeopleShop Networking
- 6.6 Conclusion
- 6.7 *Review Questions*
- References

7. Human Factors in VR

- 7.1 Methodology and Terminology
 - 7.1.1 Data Collection and Analysis
 - 7.1.2 Usability Engineering Methodology
 - 7.2 User Performance Studies
 - 7.2.1 Testbed Evaluation of Universal VR Tasks
 - 7.2.2 Influence of System Responsiveness on User Performance
 - 7.2.3 Influence of Feedback Multimodality
 - 7.3 VR Health and Safety Issues
-

- 7.3.1 Direct Effects of VR Simulations on Users
- 7.3.2 Cybersickness
- 7.3.3 Adaptation and Aftereffects
- 7.3.4 Guidelines for Proper VR Usage
- 7.4 VR and the Society
 - 7.4.1 Impact on Professional Life
 - 7.4.2 Impact on Private Life
 - 7.4.3 Impact on Public Life
- 7.5 Conclusion
- 7.6 *Review Questions*
- References

8. Traditional VR Applications

- 8.1 Medical Applications of VR
 - 8.1.1 Virtual Anatomy
 - 8.1.2 Triage and Diagnostics
 - 8.1.3 Surgery
 - 8.1.4 Rehabilitation
- 8.2 Education, Arts, and Entertainment
 - 8.2.1 VR in Education
 - 8.2.2 VR and the Arts
 - 8.2.3 Entertainment applications of VR
- 8.3 Military VR Applications
 - 8.3.1 Army Use of VR
 - 8.3.2 VR Applications in the Navy
 - 8.3.3 Air Force Use of VR
- 8.4 Conclusion
- 8.5 *Review Questions*
- References

9. Emerging Applications of VR

- 9.1 VR Applications in Manufacturing
 - 9.1.1 Virtual Prototyping
 - 9.1.2 Other VR Applications in Manufacturing
- 9.2 Applications of VR in Robotics
 - 9.2.1 Robot Programming
 - 9.2.2 Robot Teleoperation
- 9.3 Information Visualization
 - 9.3.1 Oil Exploration and Well Management
 - 9.3.2 Volumetric Data Visualization
- 9.4 Conclusion
- 9.5 Review Questions
- References

CD ROM Table of Contents

Videos and Laboratory Manual

1. Introduction to VRML and Java 3D

Objectives

- 1.1 Overview of the VRML language
- 1.2 The VRML Browser
- 1.3 Examples of VRML Worlds
- 1.4 The Basic VRML Syntax
- 1.5 Objects Creation in VRML
- 1.6 Introduction to Java 3D [Advanced]
- 1.7 VRML & Java 3D [Advanced]

Homework

Project 1-0 Install a VRML Browser

Project 1-1 Create a Simple VRML World

Project 1-2 Load VRML files in Java 3D [Advanced]

2. Sensor and Event Processing

Objectives

- 2.1 Route and Event Processing
- 2.2 Sensor Nodes
- 2.3 Interpolators in VRML
- 2.4 Creating Objects in Java 3D [Advanced]
- 2.5 Event Scheduling in Java 3D [Advanced]
- 2.6 Interpolators in Java 3D [Advanced]
- 2.7 Sensors in Java 3D [Advanced]
- 2.8 Hardware Device Interface in Java [Advanced]

Homework

Project 2.1 Interaction using Sensor Nodes

Project 2.2 Simple Interaction in Java3D [Advanced]

Project 2.3 Behavior in Java3D [Advanced]

Project 2.4 Interaction using a 3D Tracker [Advanced]

3. VRML and Java Script

Objectives

- 3.1 Programming in VRML
- 3.2 Script Node in VRML
- 3.3 Event Processing in a VRML file for scripting
- 3.4 A Scripting Example using JavaScript
- 3.5 A Scripting Example using Java [Advanced]
- 3.6 Stereoscopic Viewing using StereoEyes™ Glasses

Homework

Project 3-1 Trajectory of a Bouncing Ball in JavaScript
Project 3-2 Test Stereoscopic View with different parameters
Project 3-3 VRML Loader with Stereoscopic view [Advanced]

4. Scene Hierarchy, Geometry and Texture

Objectives

- 4.1 Scene Hierarchy in VRML
- 4.2 Constructing a Hierarchical Object: The Snowman
- 4.3 Geometry nodes in VRML
- 4.4 Extended geometry node details
- 4.5 Textures in VRML
- 4.6 Geometry in Java 3D [Advanced]
- 4.7 Texture Mapping in Java 3D [Advanced]

Homework

- Project 4.1* Create a Hierarchical hand model
- Project 4.2* Creating a Garden in VRML
- Project 4.3* Human-like Robot in Java 3D [Advanced]

5. VRML PROTO and Glove Devices

Objectives

- 5.1 Creating a New Node in VRML
- 5.2 An Example of Prototyping in VRML
- 5.3 The New Node for Device Interface in VRML
- 5.4 Data acquisition and calibration of the 5DTglove™ [Advanced]

Homework

- Project 5.1* Glove Calibration
- Project 5.2* Human-like Robot
- Project 5.3* Glove Calibration and Hand Animation [Advanced]

6. Viewpoint Control, Sound and Haptic Effects

Objectives

- 6.1 Navigation and Its Control
- 6.2 Using 3D Sound in VRML
- 6.3 Creating Force Feedback Joystick interface [Advanced]

Homework

- Project 6.1* Viewpoint Control using Glove Data
- Project 6.2* Force Feedback Joystick Interaction in Java 3D [Advanced]

Resources

References

Appendix

- A.1 Available Java 3D Loaders
 - A.2 A JNI Example Program for Polhemus
 - A.3 Combining VRML world in HTML Documents
-

- A.4 Configuration of the system to see Stereoscopic view using StereoEyes
- A.5 Example Grading Policy for Project 3-1 Bouncing Ball
- A.6 An Example of Final Project Assignment (Requirement)
- A.7 A Sample Sheet for VR Final Project Grading