AUTONOMOUS AGENTS FOR REAL-TIME ANIMATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

James J. Kuffner, Jr.
December 1999

© Copyright 2000 by James J. Kuffner, Jr.
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jean-Claude Latombe
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Leonidas J. Guibas

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Christoph Bregler

Approved for the University Committee on Graduate Studies:

Abstract

Advances in computing hardware, software, and network technology have enabled a new class of interactive applications involving 3D animated characters to become increasingly feasible. Many such applications require algorithms that allow both autonomous and user-controlled animated human figures to move naturally and realistically in response to task-level commands.

This thesis presents a research framework aimed at facilitating the high-level control of animated characters in real-time virtual environments. The framework design is inspired by research in motion planning, control, and sensing for autonomous mobile robots. In particular, the problem of synthesizing motions for animated characters is approached from the standpoint of modelling and controlling a "virtual robot".

Two important classes of task-level motion control are investigated in detail. First, a technique for quickly synthesizing from navigation goals the collision-free motions for animated human figures in dynamic virtual environments is presented. The method combines a fast 2D path planner, a path-following controller, and cyclic motion capture data to generate the underlying animation. The rendering hardware is used to simulate the visual perception of a character, providing a feedback loop to the overall navigation strategy. Second, a method for automatically generating collision-free human arm motions to complete high-level object grasping and manipulation tasks is formulated. Given a target position and orientation in the workspace, a goal configuration for the arm is computed using an inverse kinematics algorithm that attempts to select a collision-free, natural posture. If successful, a randomized path planner is invoked to search the configuration space (C-space) of the arm, modeled as a kinematic chain with seven degrees of freedom (DOF), for a collision-free path

connecting the arm initial configuration to the goal configuration.

Results from experiments using these techniques in an interactive application with high-level scripting capabilities are presented and evaluated. Finally, how this research fits into the larger context of automatic motion synthesis for animated agents is discussed.

Acknowledgments

First, I would like to thank my advisor Professor Jean-Claude Latombe for his helpful guidance and support over the past six years. Through him, I have been introduced to a wealth of new ideas ranging from motion planning to mountaineering. I also gratefully acknowledge Professor Leo Guibas and Professor Christoph Bregler for serving on my reading committee, and for their mentorship during my time at Stanford.

While conducting this research, I had the opportunity to work with and learn from many people. I am grateful to Steve LaValle for allowing me to collaborate with him, and for offering his thoughtful advice and friendship. I thank Brian Mirtich of MERL, whom I enjoyed working with to write a software library based on his thesis research. I thank the team of Yotto Koga, Koichi Kondo, and Jonathan Norton, who made creating the ENDGAME computer animated short so much fun.

The students, staff, and visitors of the Stanford Computer Science Robotics Laboratory have enriched my educational experience immeasurably. In particular, I thank David Hsu, Lydia Kavraki, Craig Becker, Tsai-Yen Li, David Lin, Hector Gonzalez, Mark Ruzon, Diego Ruspini, KC Chang, Scott Cohen, Mehran Sahami, Stephen Sorkin, Joel Brown, Serkan Apaydin, Jing Xiao, Frank Hickman, Jutta McCormick, and everyone in the motion-algo group for their friendship and for the many interesting and insightful discussions.

I thank the Department of Computer Science for providing the facilities for conducting this research. I am very grateful to Professor Mark Levoy, Professor Pat Hanrahan, and the Stanford Computer Graphics Laboratory for allowing me to utilize their rendering hardware and video equipment.

Financial support for this research was provided in part by a National Science

Foundation Graduate Fellowship in Engineering, and MURI grant DAAH04-96-1-007, of which I am very grateful.

I thank Professor Hirochika Inoue and Professor Masayuki Inaba of the University of Tokyo for allowing me to join their robotics research laboratory in Japan upon completion of this dissertation.

I am indebted to the many friends who have helped me during my time at Stanford. In particular, I am grateful for the friendship of my former roommates, Apostolos Lerios, Daisuke Terasawa, and Iacovos Koumas, who survived living with me. Thanks also to Michael Svihura, Salli Tazuke, and Phil and Mariko Yang who have been such supportive friends. I owe a special debt of gratitude to Bill Behrman, who generously donated his time and energy to help me print and file the final copy of this dissertation.

I thank my family: Maureen, Andy, Teresa, and Joseph, for filling my childhood with wonderful memories. Special thanks to my parents, James and Kathleen Kuffner, to whom I dedicate this thesis, for their unconditional love and for teaching me the value of education.

Finally, I thank my wife Haruko, for her patience and encouragement throughout.

Contents

Abstract **v**

Acknowledgments **vii**

1 Introduction **1**

1.1 The Challenge of Model Complexity 1

1.2 Task-Level Character Animation 2

1.3 Autonomy and Interactivity 3

1.4 Motivation and Previous Work 5

1.5 The "Virtual Robot" Approach 9

1.6 Overview of the Thesis 9

2 Character Animation Framework **11**

2.1 Introduction 11

2.2 Autonomous Agents 11

2.3 Motion Synthesis 14

2.4 High-Level Tasks 17

2.5 A Virtual Robot Architecture 20

3 Goal-Directed Navigation **31**

3.1 Introduction 31

3.2 Navigation using Planning and Control 34

3.3 Algorithm Outline 38

3.4 Initialization 39

3.5 Planning 42
3.6 Path Following 49
3.7 Multiple Characters 55
3.8 Results and Discussion 57

4 Sensing the Environment **67**
4.1 Introduction 67
4.2 Simulated Visual Perception 68
4.3 Internal Representation and Memory 71
4.4 Perception-Based Navigation 74
4.5 Learning and Forgetting 79
4.6 Perception-Based Grasping and Manipulation 81
4.7 Other Forms of Perception 83
4.8 Results and Discussion 84

5 Manipulation Tasks **95**
5.1 Introduction 95
5.2 Manipulation Tasks for Humans 98
5.3 Inverse Kinematics 100
5.4 Path Planning 103
5.5 Task Animation 114
5.6 Results and Discussion 121

6 High-Level Behaviors **139**
6.1 Introduction 139
6.2 Programming Behaviors 139
6.3 Following and Pursuit Behaviors 141
6.4 Integrating Navigation and Manipulation 144

7 Conclusion **147**

References **151**

List of Figures

1.1 Animated human character composed of 52 moving parts 2
1.2 Applications for animated characters 4

2.1 The control loop of a robot (physical agent). 12
2.2 Multiple views of a human character model 15
2.3 Motions as curves in a multi-dimensional joint space 16
2.4 A composite task subdivided into a linear sequence of atomic tasks . 19
2.5 Resources available for motion synthesis. 21

3.1 A human character navigating in a virtual office. 32
3.2 Data flow for the navigation strategy 35
3.3 The kinematics and DOF for the major joints 36
3.4 Bounding the character's geometry 37
3.5 Computing the "treadmill" frame of reference. 41
3.6 Projecting the environment 45
3.7 Searching an embedded graph defined by a grid 46
3.8 Cutting corners in order to shorten paths 47
3.9 Reducing the number of neighbor node expansions. 48
3.10 State variables for the oriented disc 51
3.11 Base point tracking performance for path following. 54
3.12 Wandering and following behaviors using velocity-prediction 56
3.13 Multiple Characters navigating in a maze. 57
3.14 A human figure navigating in a maze environment. 59
3.15 A human figure navigating in an office environment. 60

3.16 Behavior oscillation due to dynamic obstacles. 65

4.1 The character view frustum 72
4.2 Another example view in a virtual office scene. 73
4.3 The basic sense-plan-control loop for static environments. 76
4.4 Updating $\mathcal{M}$ in static environments 76
4.5 Updating $\mathcal{M}$ in dynamic environments 78
4.6 Scenarios requiring $\mathcal{V}_M$ in order update $\mathcal{M}$ properly 78
4.7 The revised sense-plan-control loop for dynamic environments. 79
4.8 Grasping a coffee pot 82
4.9 Grasping a flashlight 82
4.10 A character views all objects in a room 85
4.11 A table is repositioned behind the character's back. 86
4.12 Planning using the spatial memory 86
4.13 Updating as the new object location becomes visible 87
4.14 Exploring an unknown maze environment 88
4.15 The character's view during exploration 89
4.16 The trajectory followed during exploration 90
4.17 Another example showing the trajectory followed 91

5.1 Manipulating the controls of a virtual car. 98
5.2 Simplified kinematic model of a human arm. 102
5.3 Path planning for a 7-DOF human arm 103
5.4 Growing a branch towards q_{target}. 108
5.5 Basic Planner state diagram 112
5.6 RRT-Connect Planner state diagram 113
5.7 The task frame and grasp frame. 116
5.8 Grasping a flashlight. 117
5.9 The eye-neck kinematic chain 118
5.10 Eye and head coordinated targeting movements 119
5.11 Frames of reference for the eye-neck chain. 120
5.12 Simple queries for a translating 2D rigid body. 123

5.13 More challenging queries for a translating 2D rigid body. 124
5.14 Planning for an L-shaped rigid body in 2D 126
5.15 Moving a piano in a room with walls and low obstacles. 127
5.16 A 6-DOF Puma robot moving a book. 128
5.17 A human character repositioning a bottle. 130
5.18 Playing a game of virtual chess. 131
5.19 A more difficult planning query. 132

6.1 Two characters following another character. 142
6.2 An example following (pursuit) script. 142
6.3 Wandering and following behaviors. 143
6.4 Pursuing a user-controlled character 144