Chapter 1

Introduction

1.1 The Challenge of Model Complexity

Computer graphics involves the creation of images of things both real and imagined. *Computer animation* entails the generation and display of a series of such images. In either case, mathematical models are used to represent the *geometry*, *appearance*, and *motion* of the objects being rendered. Recent improvements in computer hardware and software have enabled mathematical models of increasing complexity to be used. Indeed, synthetic images of some objects have been created which bear remarkable similarity to actual photographs of real objects. As computing technology continues to improve, we can expect computer graphics and animation of increasing realism in the future. Ultimately, the dream of *virtual reality* may be fully realized, in which artificial scenes can be created that are indistinguishable from real life.

The fundamental challenges in computer graphics and animation lie primarily in having to deal with complex geometric, kinematic, and physical models. Researchers have developed 3D scanning, procedural modeling, and image-based rendering techniques in order to automate the modeling of the geometry and appearance of complex objects. For the purposes of animation, algorithms are needed to automate the generation of motions for such objects. Moreover, even objects whose geometry and appearance are very simple, may be difficult to animate due to the number of moving parts/joints which must be dealt with.

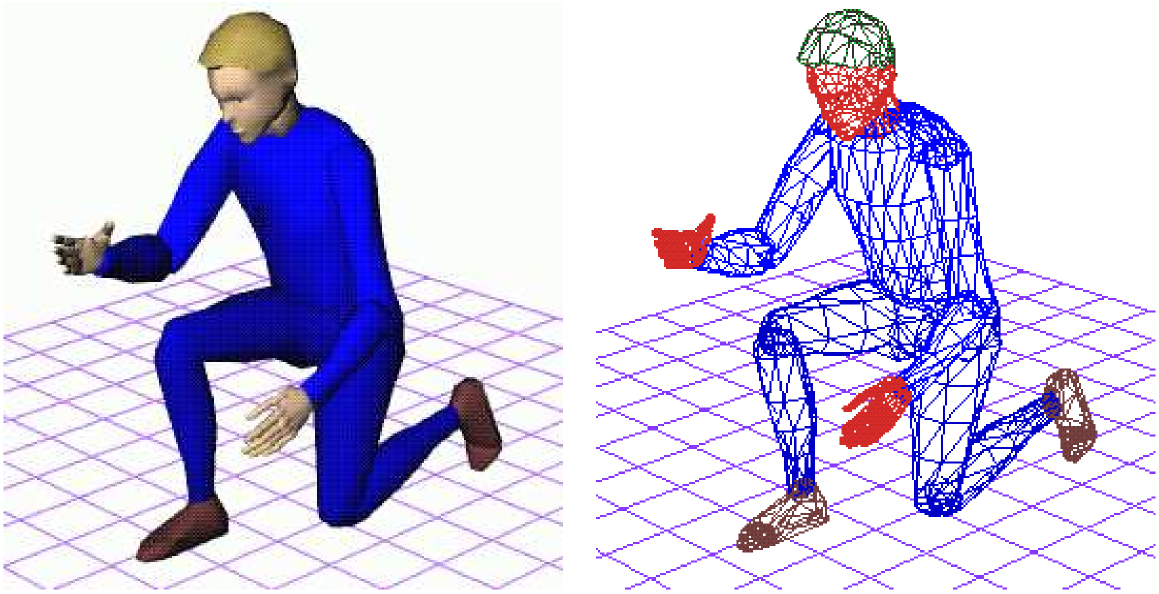

Figure 1.1: An animated human character composed of 52 moving parts with a total of 104 degrees of freedom

1.2 Task-Level Character Animation

This thesis concerns the automatic generation of motion for animated characters. Character animation poses particular challenges to motion generation due to the often complex, multi-jointed structure of the models (see Figure 1.1). In addition, the motion of each joint may be subject to multiple kinematic or dynamic constraints in order to produce body poses and motions that appear natural. Fundamentally, our goal is to create software that will enable a character to move naturally given a *task-level command* such as "walk over to the table and pick up the book". The underlying software must automatically generate the motion necessary to animate the character performing the given task.

We have concentrated on the animation of human-like character models, though the basic ideas contained in this thesis are applicable to other types of characters. We have focused our experimental efforts on human-like models for the following reasons:

1. Human characters (or characters with a human-like structure) are commonly portrayed in animated films or video games.

2. Human character models involve a complex, multi-jointed kinematic structure that is challenging, such that synthesizing human motion can be considered a benchmark for motion generation algorithms.

3. Real humans are very good at evaluating human motion and detecting subtle flaws, thus making it easier to compare the output of various motion generation strategies.

Researchers in the field have recently coined several new terms that refer specifically to animated human characters. Some of the more popular terminology includes "virtual humans", "virtual actors", "digital actors", "synthetic actors", "animated agents", and "avatars". Many of these terms have found their way into the mainstream media. Although these terms are often used synonymously, their origins can be traced back to disparate applications involving animated human characters that are very different in terms of scope and purpose. These differences stem primarily from the required characteristics of the animated characters, two of which are discussed in the next section. In order to avoid confusion due to pre-conceived notions, we will simply use the term *animated characters* as it applies in the broad sense.

1.3 Autonomy and Interactivity

Beyond geometry and kinematics, animated characters possess two fundamental characteristics which are primarily determined by the application domain in which they are being used:

The degree of *autonomy*: This generally refers to how much intervention the character requires from a user. At one extreme, are completely fine-grained, *user-controlled* characters that demand a human operator specify all joint motions by hand. At the other extreme, are *completely autonomous* characters whose motions are controlled entirely by a software program requiring no user intervention whatsoever.

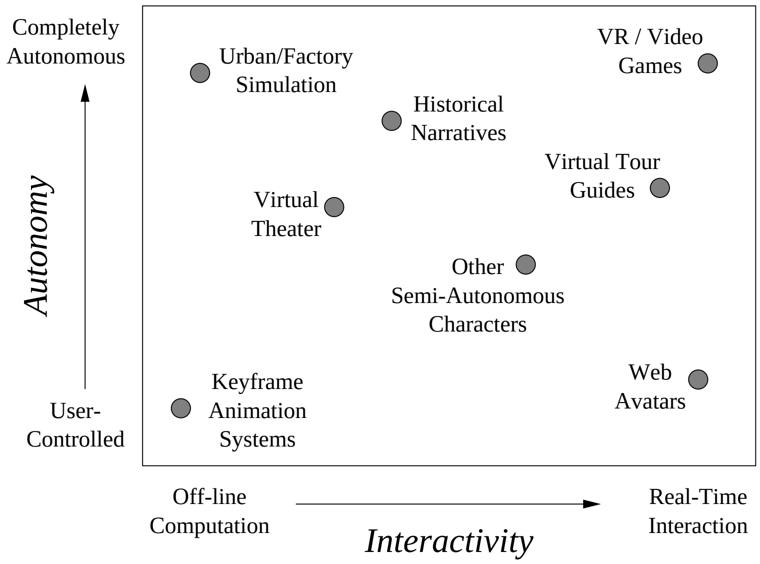

Figure 1.2: The characteristics of animated characters vary according to the application domain.

The degree of *interactivity*: This has to do with the time constraints of the application in relation to the user. At one extreme are completely *non-interactive* (off-line) software applications, such as simulations that have no strict execution time requirements. At the other extreme are highly-interactive applications that have strict execution time constraints. A common requirement is that of maintaining display update rates at 30 frames per second in order to allow *real-time interaction* with a user.

These two characteristics are orthogonal, and form a space of potential uses for character motion generation software that covers a wide range of applications. These include virtual reality, video games, web avatars, digital actors for desktop movie studios, and real-time virtual human simulations for urban, industrial, or military purposes. Some of these are plotted on the diagram in Figure 1.2.

As researchers, we are intrigued by the most challenging problems, namely fully-autonomous human character motion generation for applications that provide real-time interaction (such as virtual reality or video games). Clearly, any motion generation algorithm that is able to satisfy these criteria can also be used in applications with less demanding constraints. For example, generated motion for a character in an interactive application could be simply stored and used later to animate the same character off-line. However, generated motion that is acceptable for interactive video games may *not* be of an acceptable quality for off-line animation production (in terms of smoothness, naturalness, precision, or general artistic impression). Thus, tradeoffs exist between the quality, flexibility, and the time needed to compute motions for various applications.

1.4 Motivation and Previous Work

The primary motivation for task-level control in off-line animation systems stems from the time and labor required to specify motion trajectories for complex multi-jointed characters, such as human figures. Traditional keyframe animation techniques are extremely labor-intensive and require the artistic skill of highly-trained animators. For real-time interactive animation, keyframing "on-the-fly" is not an option. Rather, animation for both autonomous characters and characters under high-level control by the user must be generated via software.

The body of research relating to algorithms for generating animation is young but vast[TT90, BPW92, TDT96]. We only highlight here a few of the key ideas and briefly explain how they fit into the context of human figure animation. Like many things in computer graphics, some of these ideas were originally developed in other fields, such as robotics, computer vision, and artificial intelligence (AI), and have been slowly making their way into the field of computer animation.

The obvious first option to motion generation is to simply "play back" previously stored animation clips ("canned" motions). The animations may have be key-framed by hand, or obtained by *motion capture*. Motion capture techniques offer a fairly simple means of obtaining realistic-looking motion. The key idea is to record the motion of a real human performing the desired task via magnetic sensors, optical tracking, or other means[Mai96, BRRP97]. The motion is stored and used later to animate a character. With only minor editing, motion capture data can provide striking realism with little overhead. However, motion capture data alone (as well as all clip motions) are inflexible in the sense that the motion may often be only valid for a limited set of situations. Such motions must often be redesigned if the relative locations of other objects, characters, or starting conditions change even slightly.

Motion warping, blending, or signal processing algorithms[WP95, BW95, UAT95] attempt to address this problem by offering some added flexibility through interpolating or extrapolating joint trajectories in the time domain, frequency domain, or both. In practice however, they can usually only be applied to a fairly limited set of situations involving minor changes to the environment or starting conditions. Significant changes typically lead to unrealistic or invalid motions.

Alternatively, flexibility can be gained by adopting kinematic models that use exact, analytic equations to quickly generate motions in a parameterized fashion. Forward and inverse kinematic models have been designed for synthesizing walking motions for human figures[KB82, GM85, BC89, BTMT90]. Some kinematic models are based on the underlying physics of walking[AG85, OK94, KB96]. While most of these techniques assume level terrain, or a straight-line trajectory, some can handle curved trajectories and limited variations in the environment terrain[CH99]. Noise functions have also been proposed as a means of adding lifelike behavior to individual motions, or blends between motions[Per95, PG96]. These methods generate motions that exhibit varying degrees of realism.

Dynamic simulation and physically-based modeling techniques nicely handle the problems of physical validity and applicability to arbitrary situations. Given initial positions, velocities, forces, and dynamic properties, an object's motion is simulated according to natural physical laws[AG85, Bar89, MZ90, Mir96]. Dynamic simulation techniques are well-suited for generating *non-intentional motions* (such as animating falling objects, clothing, hair, smoke, wind, water, and other natural effects). While simulations are fairly straightforward to initialize and compute, specifying the desired outcome of the simulation is typically not possible. Thus, we are faced with a classic tradeoff between *physical realism* and *control*. If the animator wants full control (i.e. keyframing all motions), they will often sacrifice physical realism. Alternatively, if an animator desires physical realism, they sacrifice control. For animating *intentional motions* (such as walking, jumping, or lifting), the problem is reduced to designing a good controller that produces the necessary forces and torques to solve a particular task. Given the forces and torques, the simulation generates the resulting motions. Due to the complex dynamics of a multi-jointed system, designing a good controller is an extremely difficult task. Nonetheless, effective controllers have been successfully designed (by hand) for human running, diving, and gymnastic maneuvers[RH91, BH95, Hod96, HW98]. The advantage to designing a controller is that it is generally reusable for a given character and behavior. However, new controllers must be designed each time a new behavior or character morphology is desired. There has been some work on adapting existing controllers to other character morphologies[HP97]. Still other research focuses on automatically generating controllers via dynamic programming[dPFV90], genetic algorithms[NM93, Sim94], control abstraction[GT95], or through training a neural network[dPF93, GTH98]. These techniques have not yet been applied to complex human models.

Spacetime constraints provide a more general mathematical framework for addressing the problem of control [IC88, WM88, LGC94, GRBC96]. Complex animations with multiple kinematic[Gle98], and dynamic[PW99] constraints have been generated. Constraint equations imposed by the initial and final conditions, obstacle boundaries, and other desired properties of the motion are solved numerically. Related constraint-based and optimization techniques include [ZB94] and [BB98]. Other constraint-based techniques involving physically-based models are described in [LM99] and [AN99]. The potential downside to these methods is that the large number of constraints imposed by complex obstacle-cluttered environments can sometimes result in poor performance.

Various forms of motion planning have also been used to animate human figures. Motion planning techniques are designed to calculate collision-free trajectories in the presence of obstacles, and handle arbitrary environments. Motion planning algorithms have been successfully applied to the automatic animation of goal-directed navigation [LRDG90, HC98, BT98, Kuf98], object grasping and manipulation tasks [KKKL94a, BT97, Ban98], and body posture interpolation [BGW+94, JBN94]. These algorithms operate by searching the system configuration space ($\mathcal{C}$-space) for a collision-free path connecting a start configuration to a goal configuration. Though reasonable performance can be achieved for low degree of freedom problems (low dimensional $\mathcal{C}$-spaces), motion planning algorithms typically run slowly when faced with many degrees of freedom. Much of this thesis focuses on presenting techniques aimed at making motion planning practical for the real-time interactive animation of goal-directed navigation (Chapter 3) and object manipulation tasks (Chapter 5).

Nearly all of the motion generation techniques described in the previous paragraphs can be unified under a common framework by viewing the motion generation problem as a problem involving two fundamental tools: *model-building* and *search*. More specifically, solving a motion generation problem almost always involves *constructing a suitable model* and *searching an appropriate space* of possibilities.

Parallel to the research in solving motion generation problems for specific tasks, there has also been work on creating suitable architectures for designing autonomous animated characters in virtual worlds. Related research in robotics has been focused on designing control architectures for autonomous agents that operate in the physical world[Bro85, Ark92]. For animation, the ultimate goal is the realization of fully-autonomous, interactive, artificial human-like characters. Reactive behaviors applied to simple simulated creatures appeared in the graphics literature with Reynolds' BOIDS model[Rey87]. Tu and Terzopoulos implemented a strikingly realistic simulation of autonomous artificial fishes, complete with integrated simple behaviors, physically-based motion generation, and simulated perception[TT94]. This work was extended to include high-level scripting and cognitive models[Fun98]. Researchers at EPFL (Switzerland) have been working on realizing similar goals for human characters. Noser, *et al.* proposed a navigation system for animated characters using ideas for synthetic vision originally developed by Renault, *et al.*[RTT90]. Their implementation also included memory and learning models based on dynamic octrees[NRTT95]. These ideas were later expanded to include virtual tactility and audition[NT95, TNH96]. Researchers at the University of Pennsylvania have pioneered efforts at human simulation. Various algorithms for controlling their Jack human character model have been developed[JBN94, GBR+95], incorporating body dynamics[KMB95, LM99], and high-level scripting[BWB+95, Bad97]. Other systems include Perlin and Goldberg's Improv software for interactive agents[PG96, Per95], the ALIVE project at MIT[BG95, MTBP95, Blu96], Johnson's WavesWorld, the Oz project at CMU[BLR94], and the work of Strassman[Str91, Str94], and Ridsdale, *et al.*[RHC86]. Despite these achievements, creating autonomous animated human-like characters that respond intelligently to task-level commands remains an elusive goal, particularly in real-time applications.

1.5 The "Virtual Robot" Approach

Much research effort in robotics has been focused on designing control architectures for autonomous robots that operate in the real world[Lat91, Bro85, Ark92]. Researchers at U. Penn and EPFL have designed robotics-like architectures for animating human figures in virtual worlds [BWB+95, TDT96]. This thesis also approaches the problem of automatically generating motion for interactive character animation from a robotics perspective. Specifically, the problem of controlling an autonomous animated character in a virtual environment is viewed as that of controlling a *virtual robot* complete with virtual sensors. A virtual robot is essentially an *autonomous software agent* that senses, plans, and acts in a virtual world. User commands or high-level scripts provide task commands, while the software automatically generates the underlying motion to complete the task. This thesis focuses on developing practical methods for implementing a virtual robot approach to synthesizing motions for high-level navigation and manipulation tasks for animated human figures.

1.6 Overview of the Thesis

Chapter 2 describes a framework for automatically generating animation. Chapter 3 deals with the design of an efficient motion generation strategy for *goal-directed navigation.* Chapter 4 describes a method to efficiently simulate the visual perception and visual memory of a character, and explains how to combine this with the planning method of Chapter 3 to obtain a *perception-based navigation strategy.* Chapter 5 describes a new technique for single-query path planning used to quickly synthesize *object manipulation* motions. Chapter 6 describes how the algorithms from the previous chapters can be integrated using high-level scripting to create more interesting and complex behaviors. Finally, Chapter 7 summarizes the key concepts in this thesis along with a concluding discussion.