University of Pennsylvania
Executive Master's in Technology Management
EMTM500: Computer Visualization
Professor Norman Badler

Why Does Michael Jordan have Square Elbows?

The challenges of realistic human figures in video games

by Steven Birmingham

February, 2002

photo: Allen Iverson defends Michael Jordan in EA Sports's NBA Live 2002

Contents

Introduction

I grew up playing video games. From Pong to Pac-Man to Final Fantasy, I have witnessed firsthand the evolution of this now immense industry. In 2002, I had the pleasure of learning from Dr. Norman Badler, the professor for EMTM500: Computer Visualization at the University of Pennsylvania. Dr. Badler's area of expertise is representation of human beings through computer graphics. His perspectives on this topic, his book, his many published articles, and my interest in video games led to the inspiration for this paper.

Background

Early video games were two-dimensional. I remember my excitement when the Atari 2600 platform arrived with 2D games such as Missile Command and Asteroids. My parents paid about $100 for the console and about $30 for the games. The graphics were necessarily Spartan since the console had only 128 bytes of RAM, 4K of ROM, and a 1.19 Mhz processor (Gallagher, p. 71). The first game I saw with human characters was Karate Champ for Commodore-64. Like games for the Atari, that game was based on sprites, small bitmapped characters that could be moved around the screen. The then-high-tech fighters were drawn using a 320 by 200 pixel 16-color display (Falstein, p. 53).

Today, my platform of choice is a Sony Playstation 2, with a 294 Mhz processor and 5.6Ghz of ROM. In contrast to the Atari, the Playstation can display up to 1,280 by 1,024 pixels in 32-bit color, and render 20 million polygons per second with Z-buffering, textures, lighting, and transparency (Sony). I paid about $300 for the console and I pay about $50 for the games. The games offer stunning near-photo-realistic graphics, video, artificial intelligence, and stereo surround sound.

Game production now rivals movie production in scope and cost. Professional voice actors, artists, and musicians are listed in credits that scroll on the screen like at a movie theater. With these resources and powerful platforms at their disposal, game programmers should be able to produce believable representations of human beings for their top-selling video games. Yet they can not.

Challenges

The challenges of creating virtual humans go beyond mere video board power. According to Gratch (p. 54), "building a virtual human is a multidisciplinary effort, joining traditional artificial intelligence problems with a range of issues from computer graphics to social science." Early game characters like those in Karate Champ moved around the screen without changing shape, or with minor changes in shape corresponding to various poses (arms out, lying down, falling down, etc.).

For modern games, players are required to move as if they existed in the real world, according to their body characteristics, rules of the game, and physics. They must be able to assume any pose demanded of them by their controller, from bouncing a ball to jumping over an obstacle. They must do it in a realistic fashion. They must do it in real time. Unlimited processing power and a few more years of research in the field might make this problem trivial; however, in 2003, many challenges remain. Figure 1 lists some of these key challenges.

Figure 1: Some Challenges in Creating Realistic Humans

Motion	Use motion capture (realistic, limited flexibility) or develop animator-created motion (less realistic, flexible)
Computing	Game platforms lack power to do complex real-time animation
Physics	The more complete the physics model, the more computations needed
Skeleton	The more realistic the joints and degrees of freedom, the more computations needed
Muscles	Real muscles change shape when used. Game character muscles do not.
Skin/Sweat	Texture mapping may produce passable skin, but athletes sweat.
Clothes/Hair	Animating moving cloth and hair is a very complex graphics problem that has not yet been solved.
Individuality	Very memory-intensive to store each player's profile individually.
Teamwork	Tough enough to coordinate the motion of full teams of players, let alone model individual players' propensities for teamwork.

According to Badler (2001, p. 33) "ours may be the last generation that sees and readily knows the difference between real and virtual things." Today's virtual athletes are a great improvement over their predecessors, but a far cry from fooling even children into thinking that they are real. For that to happen, game programmers must focus their attention on improving three major areas: representing motion, controlling motion, and recreating motion.

Representing Motion

Motion Capture

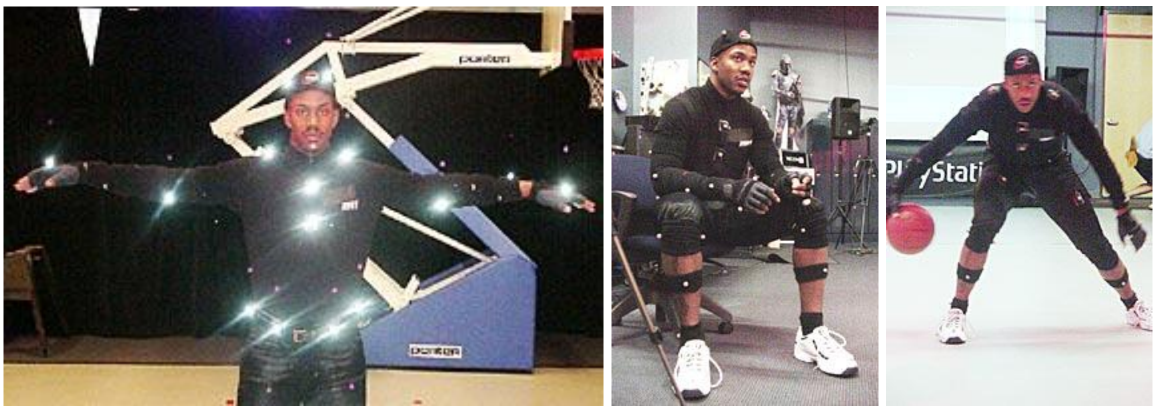
Recently, as shown in Figure 2, Phoenix Suns guard Stephon Marbury donned "black football pants, a long-sleeve black T-shirt, black gloves, elbow pads and a baseball cap, minus the brim...and...several dozen 'retro reflective markers' attached to his body," to model for a motion capture session for 989 Sports's NBA Shootout 2002 video game. (McPeek). Fifteen different cameras recorded him dribbling, shooting, dunking, and performing layups. The results were fed into a computer, processed, and then given to a computer graphic artist.

Although motion capture is a popular way to represent motion for a sports video game, it has its limitations. For one, game programmers are forced to use a small number of motion capture sequences to conserve memory and reduce processing time. For NBA Shootout 2002, this means that all the game's guards will have the moves of Stephon

Marbury, even, McPeek says, Kobe Bryant and Allen Iverson, two players with the most distinctive moves in the NBA. Conversely, in Sega's NBA 2K1, Marbury would use Iverson or Brian Grant's moves, since they provided the motion capture data for that game (Bhong).

If a game uses pure motion capture for complex moves, such as a signature dunk, it may temporarily suspend the user's ability to control the player while it recreates the motion. For example, in some early games, such as Konami's NBA Jam, once the user selected a dunk, the player would perform the move regardless of player input for the next few seconds.

Figure 2: NBA Player Stephon Marbury Suited up for Motion Capture



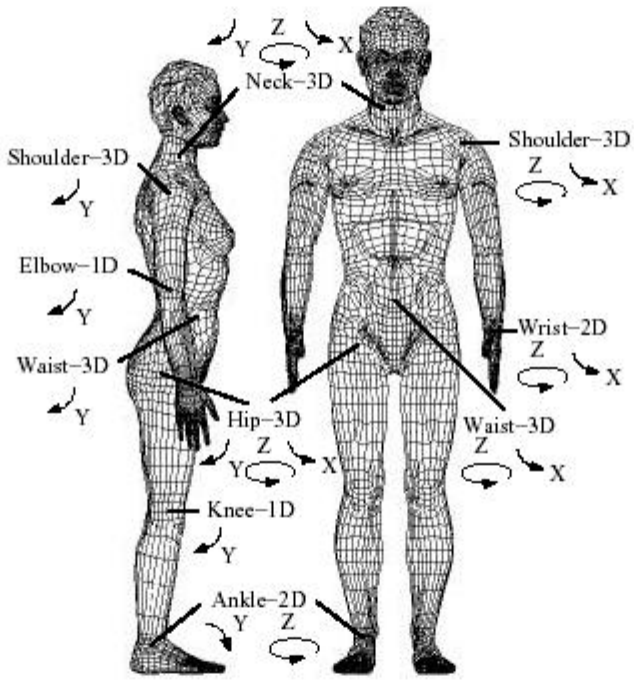
Another limitation is the range of motions that can be represented with video capture. For example, if Marbury recorded a left-handed layup only, all guards in the game would perform only left-handed layups. Park and Shin (p.106) describe a system to apply a blending algorithm to derive motions that were not recorded in the motion capture library. Through this system, a player could perform moves not specifically recorded by live athletes if similar moves had been recorded. According to Zordan (pg. 1), motion capture "must meet two, sometimes conflicting, goals: remaining close enough to the motion capture data to keep the important characteristics of the original motion while deviating sufficiently to accomplish the given task." In his paper, he describes using inverse kinematics to create a library of 550 table tennis moves from an original motion capture library of 12 moves.

Simulation

An alternative to motion capture is simulation. In a video game simulation, player models would be created based on human skeletons. Each joint would have a certain number of degrees of freedom (Hodgins, p. 72). For example, the shoulder might be able to move in 3 different directions while the elbow might be able to move in only one direction. All the model's bones and muscles would be reduced to mathematical formulas to describe how they can move. As shown in Figure 3, a typical human model might have 15-17 body segments and 22 or so degrees of freedom.

By combining user input with the model's control algorithms, the model can be made to perform various actions desired by the user. For example, the user can enter a command to make the model extend its elbow, and the model will comply by animating the requested motion on the screen. More complex motions can be effected by combining various simple motions, such as extending the elbow while opening the hand to simulate a player reaching for the ball or throwing a pass.

Figure 3: Human Models Showing Joints and Degrees of Freedom

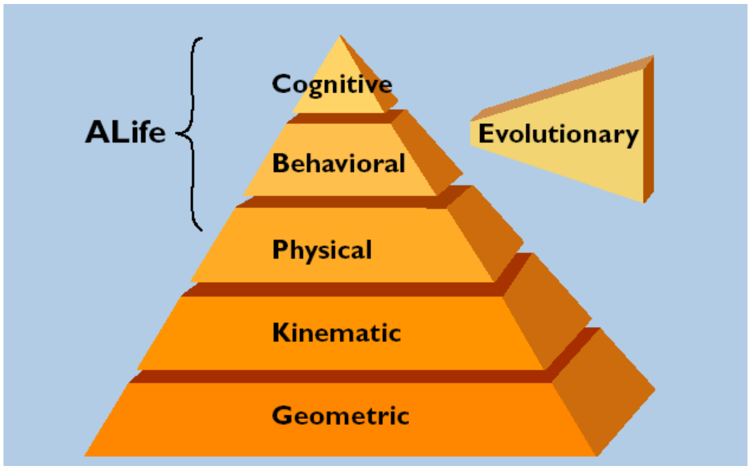


Nakata (p. 207) describes the implementation of even more complex sets of motions by employing the biological control theories of Barteneiff and Kestenberg. These theories detail the relationship between mechanics, muscles, nerves, and motion. If frogs control many of their motions by flexing either upper- or lower-body muscle groups together, then, too, should virtual beings be able to create motion through activating groups of muscles rather than individual ones.

With sufficient processing power and programming time, it could be possible to create models that not only react appropriately to user commands, but also to stimuli from other virtual beings and the environment (Terzopoulus, p. 38). If a model can incorporate not only the skeletal, muscular, physics, and environmental definitions, but also definitions of what it can see and hear, it can react to those stimuli to create an even more realistic user experience. In a sports game, players could reach out when the ball is nearby (without user intervention) much like a real player would reach for a pass as it came to them. This implementation begins to show behavioral elements that are part of the definition of artificial life as presented by Terzopoulus (see Figure 4).

The processing power of modern platforms allows games to exploit extremely detailed and anatomically correct models in real time. Viewpoint Datalabs (www.viewpoint.com) sells human models made up of 30,000 polygons. Two basketball teams of 5 players would require 300,000 polygons. The video card in Sony's Playstation2 can draw 20 million polygons per second under normal loads, allowing it to easily represent two teams plus the ball, baskets, and other elements of the environment, with no optimization.

Figure 4: The Building Blocks of Artificial Life



To add the physics of a true model to the mix would overload any platform, however. Says Zoran (p. 94), "these simulations are too slow to be used in games today, running about 10 times slower than real-time, without graphics, on a 400 Mhz R12000 SGI."

Controlling Motion

Given the selection of motion capture, models, or a combination of both for the representation of motion in a game, the programmers must decide how the user will control the available motions with the limited input capabilities of the platform. Most game consoles feature a controller with up to 10 buttons plus one or two analog sticks that move in two axes.

Laszlo (p. 201) showed that it is possible to control a character with 2 joints and 5 total degrees of freedom with a basic computer mouse. Moving the mouse up and down controlled one joint, while moving it left and right controlled the other. For more complex characters needed in games, this type of control is often supplemented by the use of buttons. Each button can represent a discrete action (jump, throw, duck, run), or can be used as a modifier for some pre-defined action (jump *higher*, throw *left).* Current games allow dozens of moves to be controlled through combinations of stick moves up/down/left/right and buttons.

Often, games require characters to perform a sequence of moves to accomplish the task requested by the user. For example, the user may enter a command to throw the ball *to* a specific player, or to follow a specific player. In these instances, much of the work done on the *Jack* system at the University of Pennsylvania can be applied (Badler 1992, Ch. 3, Granieri, p. 225). If the game stores the character's current state, and the user inputs the desired state, the game could draw from transition tables to determine the optimal path through the various required actions to achieve the desired state. Thus, by knowing the starting state of the character and giving it a goal, the game programming could follow the most logical path to that goal, implementing the necessary motion along the way. To throw the ball to player 2, a character must face that player, lean forward, extend both arms, extend all fingers, then retract all fingers, then both arms, then lean back.

Recreating Motion

Perhaps the most significant challenge of all is recreating motion featuring realistic virtual humans. The physics must look believable. The motion must look natural. The environment must look appropriate. To accomplish these goals, researchers look to robotics, biomechanics, and computer graphics.

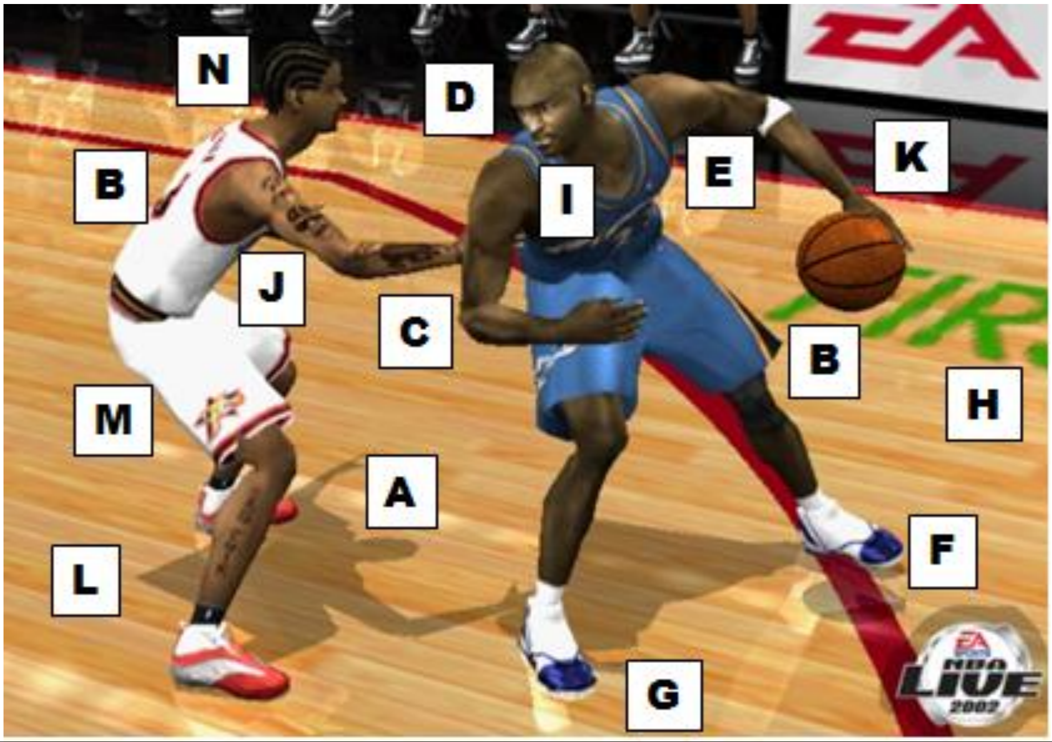
Says Badler (2001, p. 34), “the best-looking virtual humans today are painstakingly modeled and animated for movie special effects. Such off-line animation techniques may be contrasted with real-time animations for interactive simulations, training, and games. But the time lag between off-line and online animation techniques is shrinking.” Computer game creators must trade off realism for speed, constrained by the graphics capabilities of the hardware and software.

Many of today’s top games feature reasonably realistic human characters animated in real time. The virtual Allen Iverson (star of the Philadelphia 76ers NBA team) in Figure 5 is from Sega’s NBA 2K3 basketball game. Although the virtual Iverson captures some of the player’s key attributes such as tattoos, basic facial structure, and hair style, it falls significantly short of fooling the viewer into thinking it is a photo of the real Allen Iverson. The photo of Iverson shows much more complex skin tone variation, sweat, specular highlights, wrinkles, flexed muscles, textured cloth uniform, shadows, and smoother shapes.

Figure 5: Virtual vs. Real Allen Iverson

Graphics techniques such as texture maps and bump maps can help add some realism. Complex algorithms that require more processing power than gaming platforms offer are needed to create flowing cloth, realistic sweat, and flexing muscles. Figure 6, a screen capture from EA Sports’s NBA Live 2002, reveals just how far mainstream games are from producing perfect virtual humans.

Figure 6: Virtual Iverson Guards Virtual Michael Jordan



A	Extremely sharp shadows	Graphic lighting system (ray tracing?) causing shadows to be crisp
B	Pointy pants and tight jerseys	Program not sophisticated enough to model flowing cloth. One of the hardest problems in computer graphics.
C	Elbow out of joint	Skeletal model with skin/muscles attached only at certain points
D	White nose	Graphic lighting system causing specular highlights where they are not needed
E	Angular muscles	Object built from polygons
F	Non-bending ankle	Limited degrees of freedom in model
G	Perfect reflection in wood floor	Incorrect reflective characteristics or oversimplified texture for floor object.
H	Grainy letters on floor	Small-sized texture map used for large object
I	No sweat	Not included in programming due to memory or processor constraints
J	Muscles always flexed	Changes in shape due to exertion not included due to memory or processor constraints
K	No thumb	Inappropriate culling or ordering of objects?
L	Square shadow	????
M	Washed out color on shorts	Too much ambient light or too little detail in texture map
N	Small head on large neck	Inappropriate mixing and matching of model body parts

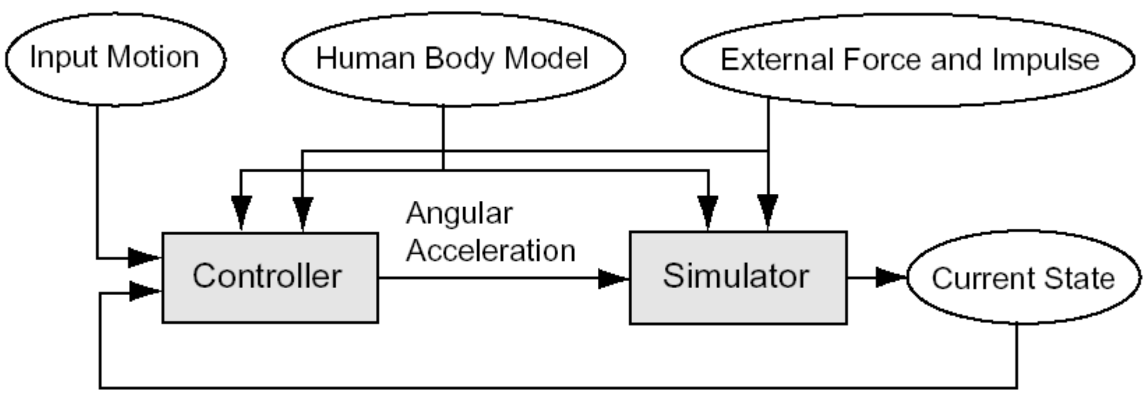
In addition, while games may allow players to create custom avatars, they typically force players "to select from a fixed set of models, with little support for avatar customization beyond texture selection." (Lewis, pg. 1). Says Hodgins (pg. 71), "people are skilled at perceiving the subtle details of human motion. We can, for example, often identify friends by the style of their walk when they are still too far away to be recognizable otherwise." Differentiating characters in video games, especially those who represent real-life athletes, requires far more memory and computer power than is available today.

Real humans affect their environment and are affected by it. Contact with objects causes skin to stretch and soft tissue to change shape. According to Gratch (pg. 58), "modeling and animating the local, muscle-based, deformation of body surfaces in real time is possible through shape morphing techniques, but providing appropriate shape changes in response to external forces is a challenging problem."

To achieve basic interaction with the environment and realistic motion, game programmers can adopt a control method like the one shown in Figure 7 (Oshita, pg. 4). It involves accepting real-time input from the user and processing it to calculate joint angles, joint velocity, and angular acceleration for all joints in a human model. The system incorporates rules to satisfy comfort (making sure the model performs within the limits of the human body) and balance considerations.

Figure 7: Dynamic Motion Control Technique for Human-like Articulated Figures

© The Eurographics Association and Blackwell Publishers 2001



Humans employ various motions of the ankle and hips to maintain balance, in addition to using the arms and legs. Video game characters often look like they are floating, because the games do not contain realistic balance algorithms. Faloustos et. al. present numerous balance algorithms, however they produce interactive model animation at only 6 frames per second (p. 949). When balance fails, people fall. The way they fall depends on their physique, age, and training (Faloustos, et. al, p. 939). Games seldom contain realist falling sequences, and rarely contain realistic rising sequences after a fall.

Other, more subtle motions allow us to tell real humans apart from virtual ones. Nostrils flare and chests heave under heavy breathing. Eyes move to follow objects or to reflect moods, and they blink frequently. Involuntary twitches sometimes appear in the face. Skin stretches as joints rotate, especially in the neck and hands. Computer games have not yet advanced to the point where these subtleties can be incorporated.

Conclusion

Games have come a long way from their early days of few colors and stick figures. Today's platforms and programs provide a rich, interactive virtual environment for players to explore. Avatars and other characters add realism to the environment and make play exciting.

To create perfect virtual humans, researchers must overcome key challenges in the areas of representing motion, controlling motion, and recreating motion. They will do this through increased processing power and continued advances in robotics, biomechanics, and computer graphics. When these tasks are complete, ours will be the last generation to be able to tell virtual humans from real humans.

Bibliography

Badler, N., Phillips, C., et. al. "Simulating Humans: Computer Graphics, Animation, And Control." Oxford University Press, 1992.
Provided background for the paper, although some of the information has been superceded by subsequent research by the authors and others. Gave detailed explanations of the basic challenges and solutions to representing humans..

Badler, N., Palmer, M., et. al. "Animation Control for Real-time Virtual Humans." August 1999/Vol. 42, No. 8 Communications Of The ACM. pp 65-73.
Covers some of the important qualities required for a realistic animated human. Touches upon the differences in requirements for various applications, including games. Mentions the challenges of true animation versus simple playback.

Badler, N. "Animation 2000++". IEEE Computer Graphics and Applications. January/February 2000. pp 28-29.
Provided a key high-level overview of the issues of representing human motion. Touched on motion capture, possible ways in which the use of animation might evolve. Also gave a brief mention of the possible future use of natural language as a system to control motion.

Badler, N. "Virtual Beings." Communications Of The ACM March 2001/Vol. 44, No. 3, pp 33-35
Gives a brief look at the future of animated humans. Poses some moral and ethical questions that will need to be addressed. Predicts that virtual humans will become indistinguishable from real humans within one generation.

Bhong, C., Lim, D., et. al. "Shootout: Get Serious (website)" www.bhong.com/shootout
Compares two basketball video games: Sega's NBA 2K1 and NBA Live 2001. Gives a player's perspective on the non-technical issues of the games.

Egan, K. "Introduction to Computer Graphics: Video Games." Brown University Computer Science 123 lecture notes, December 5, 2002.
Provides an interesting overview of how video games work, including the physics, computer graphics, and artificial intelligence. Includes references to specific games and some of the programming behind them.

Faloutsos,P, van de Panne, M. et. al. "The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills." Computers & Graphics 25 (2001) 933–953.
Covers specific controllers needed to create a variety of natural human motions such as rising, falling, and crouching. Presents detailed algorithms to produce these motions. Also provides a framework for creating a general-purpose computer graphics human to act as a virtual stuntman.

Falstein, N. "Portrait of the Artists in a Young Industry." Computer Graphics, May 1998. pp 52-54.
Gives a brief but interesting look into the early history of computer game graphics. Touches on some early hardware limitations. Covers historical costs of the early attempts at games and special effects.

Gallagher, S. and Park, S. H.. "Innovation and Competition in Standard-Based Industries: A Historical Analysis of the U.S. Home Video Game Market" IEEE Transactions On Engineering Management, Vol. 49, No. 1, February 2002. pp 67-82.
Although the article covers the historical events in the video game console market, I used it only for the technical descriptions of various platforms.

Granieri, J., Crabtree, J. et al. "Production and Playback of Human Figure Motion for Visual Simulations" ACM Transactions on Modeling and Computer Simulation, Vol. 5, No. 3, July 1995, pp 222-241
Presents a system to store motion in transition tables created offline and then used to recreate motion in real time. Used as one possible option in my paper.

Gratch, J., Rickel, J. et al. “Creating Interactive Virtual Humans: Some Assembly Required.” IEEE Intelligent Systems. July/August 2002. pp 54-63.
This article provided the framework for my entire paper. It clearly outlined the key challenges of animating humans in real time. It also provided several important quotes for the paper.

Haruyuki, T., Hashimoto, K., et al. “CPU for Playstation 2.” Sony Computer Entertainment. Tokyo, Japan. 2001.
Only the abstract was available for this paper. It provided some technical details on the hardware that drives this key gaming platform.

Hodgins, J., and Wooten, W. “Animating Human Athletics.” College of Computing, Georgia Institute of Technology, 1995.
Provides a general framework for understanding the setup of basic virtual human bodies. Introduces kinematics, motion capture, skeletons, degrees of freedom, and other basic concepts. Especially useful for my paper since it relates to athletics.

Kushner, D. “The Wizardry of Id.” IEEE Spectrum. August, 2002. pp 42-47.
Gives a great account of the evolution and influence of Id Software’s contributions to gaming. Covers in some detail the technical advances that allowed real-time navigation of ever-more-complex virtual environments.

Laszlo, J., van de Panne, M., et.al. “Interactive Control For Physically-Based Animation” SIGGRAPH 2000, New Orleans, LA USA
Covers the topic of how to control characters with many degrees of freedom using a limited-capability input device such as a mouse or keyboard. Provides examples of grouping various actions together to be controlled as one. Goal is to allow input to control a character in real time. Very useful for my paper due to the exploration of control of bipedal figures.

Lewis, M. “Evolving Human Figure Geometry.” Advanced Computing Center for the Arts and Design, The Ohio State University. Technical Report OSU-ACCAD-5/00-TR1, May 2000.
Presents the concept of varying specific body parts to provide variety and realism in computer graphics figures. Contemplates using Maya software to simulate genetic mutation and evolution to generate populations of human figures. Used in the paper to highlight the need to allow for non-perfect, non-symmetric figures.

McPeek, J. “Suns guard to grace video game cover: Marbury in Motion.” Phoenix Suns Website, www.nba.com/suns/news/marbury_motion_010727.html. July 27, 2001.
Gives an account of an actual motion capture session with an NBA star. Data is used for a top-selling basketball video game. Covers some of the practical aspects of motion capture as well as some problems. Used in the paper as an example.

Nakata, T. “Algorithmic Choreography of Animal-like Body Movement Based on Bartenieff’s and Kestenberg’s Theories.” IEEE International Workshop on Robot and Human Interactive Communication. 2001. pp 207-212.
Describes the use of two specific theories to control virtual muscles and skeletons to create virtual motion. Proposes flexing virtual muscle groups to create motion similar to a horse trotting and other motions.

Oshita, M., and Makinouchi, A. “A Dynamic Motion Control Technique for Human-like Articulated Figures.” Eurographics Volume 20, Number 3, 2001.
The paper presents a system that controls motion using real-time user input optimized for video games and other online applications. It considers angles, angular rotation, velocity, gravity, balance, comfort, and contact with the ground. Used for figure 7 and general ideas on controlling figures.

Park, S., Hwang, Y. et al. “Human Computer Competition in Game Situation: Motion Planning for Boxing.” Sandia National Laboratories, Albuquerque, New Mexico, 1996.
This paper presents a system that simulates a boxing match between a human player and a computer-controlled player. Most interesting for my paper was the explanation of the input system used to control the human boxer: motion sensors attached to the human’s body. It also describes motion planning procedures based on complex formulae.

Park, S., and Shin, S. “On-line Locomotion Generation Based on Motion Blending.” © 2002 ACM 1-58113-573-4/02/0007, pp 105-111.
Talks about adapting motion-capture data to duplicate curved movements and other actions not in the motion capture library. Addresses the issue of blending algorithms that are efficient enough to work in real-time.

“Sony Computer Entertainment Announces The Development Of The World’s Fastest Graphics Rendering Processor Using Embedded Dram Technology.” (Press release) Tokyo, March 2, 1999 -- Sony Computer Entertainment
Gives a detailed description of the capabilities and technical specifications of the Sony Playstation 2 video board. Used in the paper to contrast today’s processing power with that of early systems.

Terzopoulus, D., “Artificial Life for Computer Graphics.” Communications of the ACM. Vol. 42, No. 8. August, 1999.
Takes a biological approach to modeling humans and other species. Touches upon the possibility of virtual evolution. Covers the importance of giving virtual humans perception to guide their actions.

Zordan, V., and Hodgins, J. “Motion capture-driven simulations that hit and react.” Copyright © 2002 by the Association for Computing Machinery, Inc.
Describes a specific experiment where limited motion capture data is blended with equations to create more flexible motion. Especially applicable to this paper since it addresses a sports application.