University of Alberta

Augmenting Animated Quadrupedal Gaits Via Motion Warping

by

Jean-Paul C. Samson

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Computing Science

Edmonton, Alberta

Spring 1999

0-612-40104-9

University of Alberta

Library Release Form

Name of Author: Jean-Paul C. Samson

Title of Thesis: Augmenting Animated Quadrupedal Gaits Via Motion Warping

Degree: Master of Science

Year this Degree Granted: 1999

Jean-Paul Samson

Jean-Paul C. Samson

1 Gresham Boulevard
St. Albert, Alberta
Canada T8N 1A8

Date: April 6, 1999

It cannot be claimed that the study of animal propulsion has greatly contributed towards the general welfare of mankind or to the amenities of modern life. These things depend—or so it would seem—on the invention of larger aeroplanes, more powerful motor cars and faster ships.... No animal can compete in size or speed with mechanical machines so why bother to study animals except as an intellectual respite from the pressures of daily life?

— James Gray, *Animal Locomotion* [23]

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Augmenting Animated Quadrupedal Gaits Via Motion Warping** submitted by Jean-Paul C. Samson in partial fulfillment of the requirements for the degree of Master of Science.

John Buchanan
Supervisor

Benjamin Watson
Examiner

Curtis Hrischuk
Examiner (External)

Date: March 11, 1999

Dedication

For Mopsy and Benji—

Man's Best Friends

Abstract

It is a long-standing objective of computer animation to imitate animal movements. This task is complicated by the diversity of variations of an individual action. For example, a walk is influenced by the creature's path, gait, and speed. We propose an editing system specifically tailored to reshaping animated quadrupedal locomotions. After careful consideration of published research, it was decided to employ a low-level motion warping technique devised by Witkin and Popović. A series of tests will highlight the benefits and weaknesses of this approach.

Given the motion of one stride of a gait, the animal is made to traverse a user-defined course. Some simple constraints, such as affixing stance feet on the ground, are ensured. The durations of the leg phases are adjusted according to the animal's speed. Transitions between gaits, for instance from walking to running, are accomplished by temporally aligning the leg phases and applying progressive blending.

Acknowledgments

Assembling a thesis is a massive undertaking. Even though it stands as one individual's work, there are a great many others that contribute to its creation.

The inception of my thesis journey took place during one session of my graduate-level graphics course. As I took my seat in the classroom, the instructor, John Buchanan, informed me that he wished to speak to me after his lecture. I am not known to be a troublemaker, but I was nevertheless concerned. Was I about to be reprimanded for some act I was not aware of having committed? After class, John asked if I would like to work on an upcoming endeavour involving the animation of small, fur-bearing creatures. (Actually, I think at the time he was more interested in figuring out if I was related to my father. No, I'm not going to explain.) Thank you, John, for the opportunity to work on this project and for your gentle guidance as my overseer.

After this conversation, we had a meeting with our collaborator, L. Zack Florence, to discuss the goals of the project. Zack headed the team at the Alberta Research Council (ARC) in Vegreville. He ensured that I was provided with computer equipment and always pushed to get the project completed in a timely manner. Duane Evenson, a summer student at ARC, dedicated himself to generating animated animal movements. I would not have had the time (nor, dare I say, the skill) to devise such high-quality, visually-convincing motion data which was essential for testing my algorithms. Marion Herbut was the resident LightWave 3D™ expert at ARC[1]. Marion crafted the marten model and provided general advice regarding the LightWave™ animation tools. Michelle Hiltz, a statistician at ARC, acted as an intermediary in discussions. She aided in clarifying my computer-oriented explanations to the rest of the group. I am grateful to Zack, Duane, Marion, and Michelle for all their help.

1. LightWave 3D is a trademark of NewTek, Inc.

I am thankful of my parents, John and Liz, for their support over the years. My father dispensed some occasionally useful comments regarding my writing. My mother ensured that I had a never-ending supply of macaroni and cheese dinners. (Even though I lived at home, she still thought I ought to eat as a student. I've sampled so many varieties of pasta that I am now an expert in the field of noodles as well as computer animation.) In addition, when computer resources at the university became scarce, I used my parent's home computer to generate my animation results.

Without the companionship of my fellow students in the graphics lab at the University of Alberta, life during the past couple of years would have been very dull. In particular, the members of John Buchanan's Rendering Group require special mention. Lisa "Aye-ca-rumba" Streit was the Canadian spirit, with her easy laugh and innate skill at organizing the chaotic computer crowd. Paul "There's Something Wrong with Your Setup" Ferry provided a dose of Irish skepticism. (I recall the first night I met Paul. As we were walking home from the bar, he bemoaned the fact that John had offered a forthcoming animation project to a student other than him. He didn't realize at the time who that other student was.) Mario "Scratch-and-Win" Costa Sousa contributed a Brazilian coolness with his tales of the Carnival and his convincing rap musician impersonations. Oleg "Be Careful, Guys" Veryovka was the argumentative Ukrainian, consistently offering alternate (and usually better) solutions to problems, as well as hosting some very good barbecues.

A final note of recognition goes to my longtime friend David Robinson, who supplied me with some handy software utilities.

I wrote this thesis using FrameMaker® document authoring software on NeXTstations® and Intel®-based PCs running the OPENSTEP® for Mach operating system[2]. The project involved the creation and management of some 350 images and over 100 equations and symbols. I am grateful for the rich working environment provided by OPENSTEP®, as I was not prepared to ignore the past twenty years of user-interface research and program my thesis in LaTeX.

Funding for this research was provided by the Alberta Agricultural Research Institute and the Fur Institute of Canada.

2. FrameMaker is a registered trademark of Adobe Systems Incorporated. NeXTstation and OPENSTEP are registered trademarks of NeXT Software, Inc. and Apple Computer, Inc. Intel is a registered trademark of Intel Corporation.

Table of Contents

List of Figures

Chapter 1

Introduction

Animation is the process of creating the appearance of motion by the rapid display of a series of still images. In regards to traditional animation, this definition is specialized to the photographing of hand-illustrated pictures or posed sculptures onto frames of film. For computer-based animation, it is more appropriately described as the rendering of instantaneous states of a dynamic scene to a display screen or recording device [47].

There are many applications for animations. The most publicly visible are in the entertainment sector. Animation technologies have advanced from the days of classic, hand-drawn Disney films such as *Bambi* [6] into state-of-the-art computer rendered effects seen in movies like *Jurassic Park* [30] and video games such as *Quake* [15]. Animated diagrams may be employed for education purposes. For instance, an animation could depict the chemical reactions that generate a current within an electric cell. When examining large amounts of data that change over time, animations provide a powerful form of visualization.

Much effort has been devoted to developing methods for transcribing the motions of real-life objects onto computer model counterparts. For any given action, there are many possible variations. For example, an animal that is walking slowly will have a different stride than when it walks quickly. Having gone through the time consuming process of creating an animated movement, editing facilities are beneficial for generating customized variants.

Of particular interest is the animation of legged figures such as humans or animals. Much of the research has concentrated on bipedal forms. As the aforementioned films *Bambi* and *Jurassic Park* demonstrate, there is also a need for quadrupedal animation techniques.

This chapter provides a brief overview of traditional artistic and computer-oriented approaches to animation. The need for methods for altering existing quadrupedal motions is emphasized. An editing system is proposed that satisfies this goal, and the criteria for evaluating this solution are presented.

1.1 Approaches to Animation

1.1.1 Traditional Animation Fundamentals

Just as film technology extended still photography to create motion pictures, individual drawings in a series can be used to form animations. This new art form was exemplified by the work of Winsor McCay during the 1910's [48]. A cartoonist by trade, McCay single-handedly pioneered the animated cartoon using sequences of simple sketches. One of his most noteworthy animations was *Gertie the Dinosaur* in 1914 (see Figure 1.1). The decision to feature a dinosaur was to prove that his results were not achieved by merely tracing photographs.

Gertie was a unique creation. To be suitably convincing to the audience, her movements needed to be consistent with a large, lumbering four-legged animal. In addition, Gertie was bestowed with human traits such as timidness and dejection. Although a success with the viewing public, commercial prosperity eluded McCay. He eventually returned to drawing conventional cartoons for newspapers, his animation techniques and works forgotten.

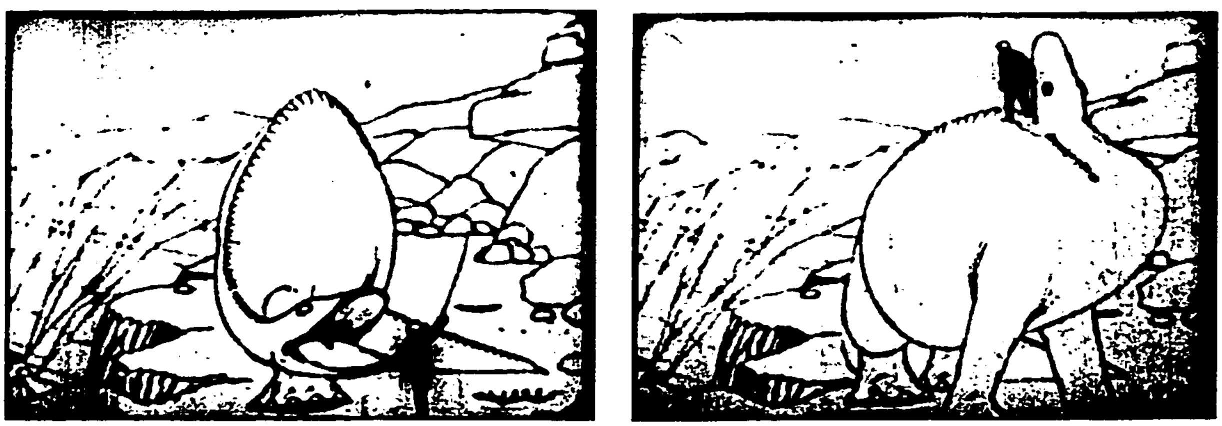


Figure 1.1: **Selected frames from the early animated cartoon, *Gertie the Dinosaur*, drawn by Winsor McCay in 1914 [19].**

Walt and Roy Disney formed an animation studio in 1923. The company would realize many remarkable technical achievements, culminating in some of the grandest animated movies ever produced. Research played an important role at the studio with an aim to improve the quality of their animations. Over time, McCay's animation techniques were independently rediscovered, explored, and further refined.

A number of animation principles were formulated at Disney's studio, as revealed by Thomas and Johnston in their book *Disney Animation: The Illusion of Life* [48]. A selection of these techniques include pose-to-pose, repeated actions, arcs, timing, and exaggeration:

1. **Pose-to-Pose**: To derive the structure of an animation sequence, animators typically employ a pose-to-pose approach. A few key drawings of the sequence are produced; these images depict the essential actions at their "extreme" positions. Afterwards, the missing in-between frames are drawn.

2. **Reuse of Actions**: Many actions are repeated, either continuously or occasionally. For example, a walking movement is cyclic and can be strung to itself to create a longer sequence. A common action such as walking can also be used in multiple scenes. However, modification of the beginning and end of an action may be required for transitional purposes.

3. **Arcs**: Objects, especially the limbs of an animal, tend to move in arcs rather than along straight lines. For instance, when moving an arm to point somewhere, the hand inscribes an arc centred at the shoulder.

4. **Timing**: The timing of an action implies its urgency. If a person slowly lifts a mug of coffee to his lips and takes an elongated drink, one might infer the person is relaxed or tired. The mug is quickly lifted and the beverage hurriedly swallowed, this might suggest the person is angered or rushed.

5. **Exaggeration**: By exaggerating an action, it is made more convincing or appealing to the audience, despite lacking realism in a physical sense. In reality, a person running around a corner would lean into the turn. An animated character's cornering might be further emphasized by leaning more and having his feet slide upon the floor.

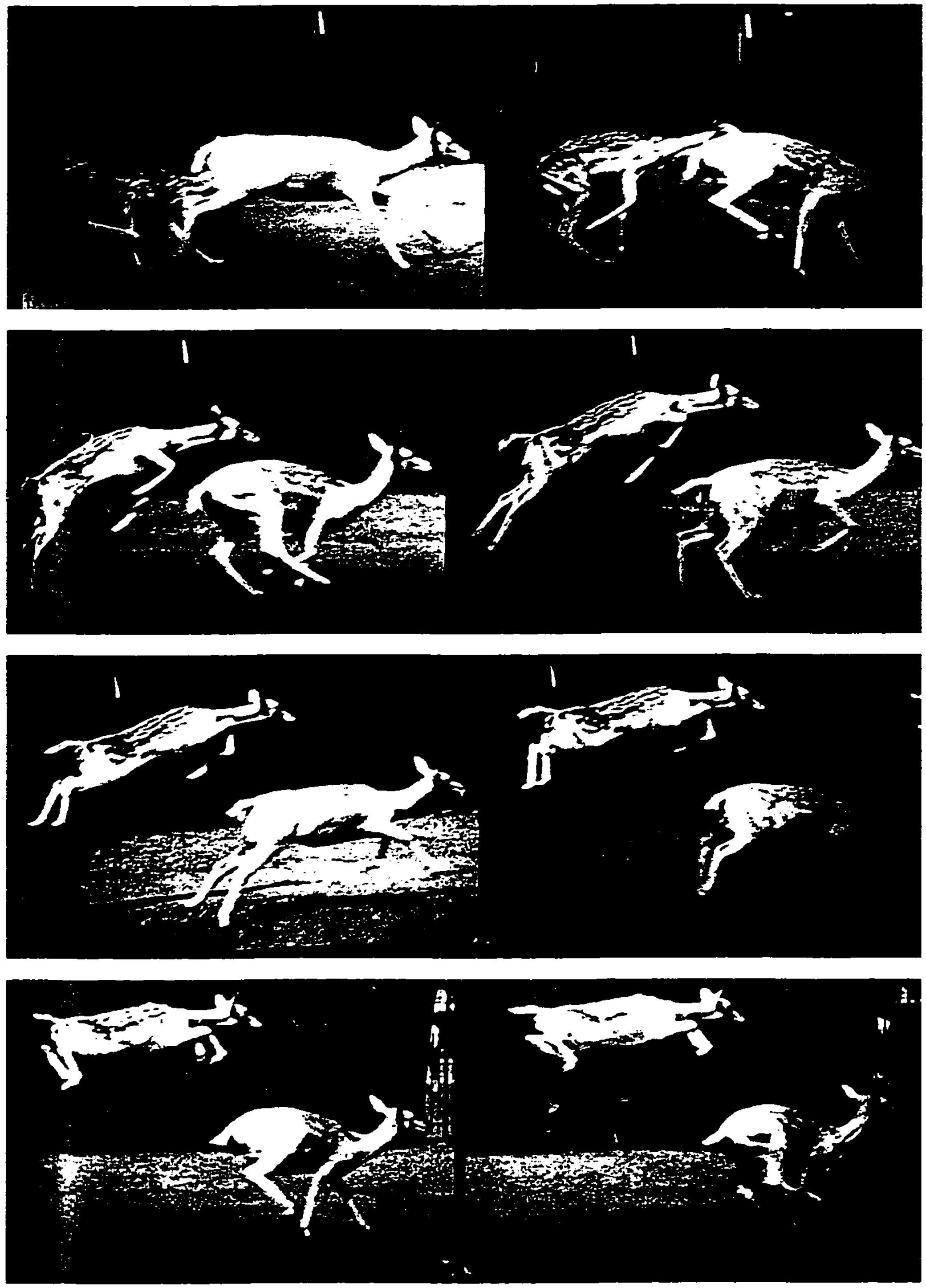


Figure 1.2: Photographs of two deer, one galloping and the other jumping, taken from Plate 691 of Muybridge's *Animal Locomotion* [39].

When preparations for the feature-length film *Bambi* [6] commenced in 1938, Disney had already completed its first animated movie, *Snow White* [45]. Two other films, *Pinocchio* [41] and *Fantasia* [17], were still in the process of being animated. Despite all this experience, it was felt that the animators did not have sufficient skills to draw the deer that would feature in *Bambi*. This was the first movie where quadrupeds would play the main characters, rather than bipeds such as humans or personified animals. An experimentation phase commenced to learn how to draw deer.

By studying a variety of resources, geometric models of deer were constructed and the animals' movements identified. Some live deer were brought to the studio, providing an immediate subject for examination. A carcass of a fawn was dissected to determine the muscular and skeletal structures. Muybridge's seminal photographic work capturing the movements of humans and animals was examined (see Figure 1.2) [39]. Nature films were scrutinized on a frame-by-frame basis.

From this research, animators learned a number of important facts. There are limitations to the maneuverability of the joints of an animal; a junction may only rotate between fixed tolerances in one, two, or three dimensions (see Figure 1.3). In motion, quadrupeds employ a variety of gaits; which gait is active depends on how fast the animal is moving. Finally, the animation concepts of arcs, timing, and exaggeration were recognizable in the movements of real animals.

With the necessary information regarding deer compiled, the animation of *Bambi* commenced in 1939. The film was finally completed three years later in 1942.

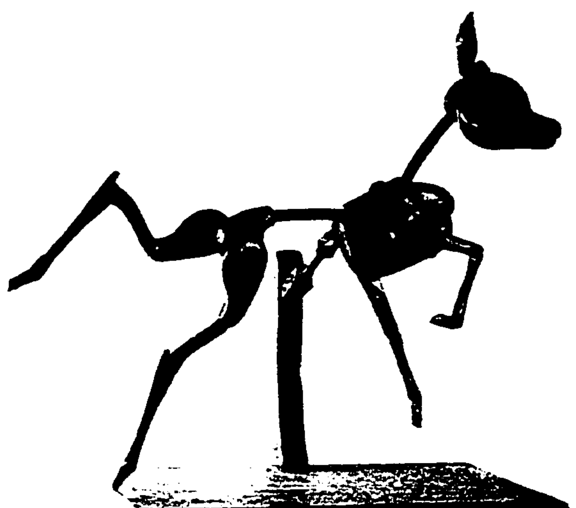


Figure 1.3: **An articulated sculpture of a deer, built as a study aid for animators of the film *Bambi* [48].**

With the introduction of computers, much of the animation process could be automated and performed with greater precision. It is crucial that computer-based systems support the animation of legged figures such as quadrupeds and incorporate the basic principles of reuse of actions, timing, exaggeration, *etc.*

1.1.2 Computer-Based Animation

Many tradition artistic animation techniques have direct counterparts in computer animation systems. In addition, many new approaches to animation have been spawned that were not previously possible because of the computational complexities involved.

The essential problem of animation is to specify the motion of objects within a scene. These objects are typically constructed from three-dimensional geometric shapes (although they are usually viewed as a projection onto a two-dimensional surface). To support posing, the rigid segments of an articulated figure are connected by flexible joints. Junctures are limited to have one to three degrees-of-freedom (*i.e.* axis of rotation); further limitations in the range of rotation for each degree-of-freedom may be imposed. The articulated figure has been universally adopted by the traditional and computer animation community alike for object representation.

One of the earliest computer-based animation endeavors was accomplished by Burtnyk and Wein, culminating in the National Film Board short, *Metadata*, in 1971 [47]. A pose-to-pose approach was adopted. Line drawings of selected frames were rendered by artists and the vertices in these images identified. The computer then automatically generated in-between frames using linear or quadratic interpolation [14].

This process, known as keyframing with interpolation, continues to be a common approach in commercial animation systems. The figures in the scene are posed in sparse key frames and the intermediate frames are produced by interpolation curves. The use of curves, rather than simple linear segments, is crucial for making motions appear smooth. The movements may be edited in the same manner by which they were originally constructed—by manipulating the control parameters in the keyframes. The motions may be stored and then reused in future sequences. Keyframe-based animation systems have been a success largely because they support many of the basic principles of traditional animation.

Approaches other than keyframing that are unique to computer animation have been introduced. These techniques include dynamic simulation, motion capture, and inverse kinematics:

- Dynamic simulation uses principles from physics to model the motion of an object. Control algorithms are devised that encode the rules of an action. These algorithms accept high-level parameters as input. For instance, information guiding a walking action would include speed and step frequency. From these constraints the forces and torques required to propel the body segments to their desired positions are approximated. Optimization sessions are necessary to tune the constants of the control algorithms to arrive at a stable motion. Bruderlin and Calvert developed KLAW, a dynamics-based animation system of a human walk [13]. Hodgins *et al.* have concentrated on human athletic activities, devising simulations of running, bicycling, and vaulting [29].

- For motion capture, devices are attached to the surface of a real person or, less commonly, an animal. These are used to measure the movement of body segments [38, 16]. The devices may be potentiometers, magnetic field sensors, or light reflectors. Potentiometers mechanically measure the rotations of joints. Magnetic sensors measure the field emitted by a nearby source to derive their location. A video camera captures the illuminated light reflectors and a computer extracts their movements. The captured movements are then used to control a geometric model.

- Inverse kinematics finds the orientations of the joints of a multi-segment limb such that the tip—termed an end effector—touches or reaches towards a defined goal point [10]. Because of the redundancy of the articulated figure, there are many possible solution poses. Secondary objectives can be used to restrict the inverse kinematics problem to yield a more desirable positioning. Bruderlin and Calvert have devised a successor to their dynamic simulation of bipedal walking, GAITOR, that employs inverse kinematics techniques [12].

Progressive advances in computer animation and rendering techniques have led to results that are nearly indistinguishable from their photographic equivalents. It is fitting that one of the first successful deployments of computer-animated creatures took the form of the dinosaurs of the film *Jurassic Park*, just as McCay had used *Gertie the Dinosaur* to impress the audiences of his early animated cartoons.

1.2 Motivation

Despite all of the advances offered by computer-oriented animation, in practice few of these new methods are employed by animators. The reasons for this are partially due to the difficulties applying and disadvantages inherent in the techniques. An animation system needs to facilitate three tasks:

1. Support the definition of an original movement of an articulated model.

2. Provide tools for the modification of defined motions, thus promoting reuse and variety.

3. Ensure that transitions between concatenated motions appear smooth.

In addition, it is desirable to support the traditional animation principles such as timing and exaggeration. To achieve these requirements, animation systems typically use a combination of computer-based techniques.

Computer animation research has concentrated on the issue of generating original motions. Studies of automated methods for deriving new motions from existing ones have been sparse. Nor has there been much consideration of the problem of transition from one motion into the next for concatenation purposes. Each of the major computer-based animation methods have benefits and weaknesses in these regards:

- Keyframing with interpolation is a very simple approach easily understood by animators, providing a pose-to-pose paradigm. By using splines for interpolation, the motions are guaranteed to be smooth, with no discontinuities in velocity. This technique supports all three of the tasks required of an animation system. The animator manually sets the positions and joint angles of the articulated figure; this offers good, low-level control, but is time consuming.

- By simulating the physics of motion, dynamic animations are plausible and realistic in appearance. Because of the complexity of the dynamics involved, such systems make simplifying assumptions and frequently incorporate some alternative animation methods such as inverse kinematics. Dynamic simulation supports the definition and modification of movements as required by a robust animation system. Building the control system requires expert knowledge and involves time intensive tuning sessions; these are tasks that an animator may be incapable of performing. High-level parameters provide a moderate degree of control for modifying an action. However, many animators dislike the limitations of the controls imposed by the tight constraints of the physical model.

- Motion capture is excellent for acquiring an original motion. Despite the large volume of data involved, there is little intrinsic information regarding the constituent movements of an action. This makes altering and concatenating motions difficult. Also, the specialized equipment required can be expensive.

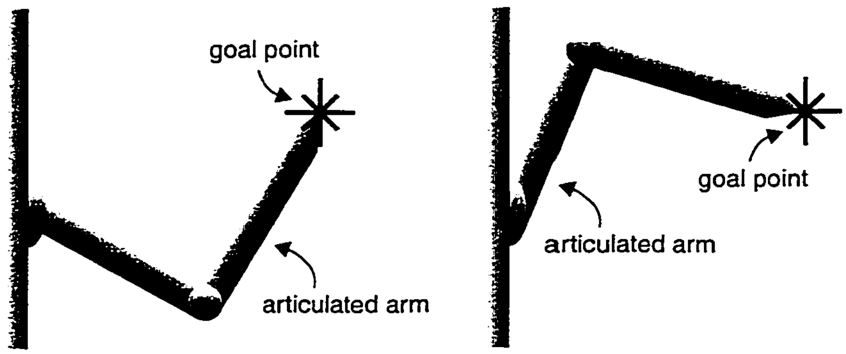


Figure 1.4: **The inverse kinematics solutions for a two-dimensional, two segment object reaching for a goal point.**

- Inverse kinematics reduces the number of parameters an animator must manually control; flexibility is restricted by the positioning of end effectors. The technique is often used in conjunction with keyframing with interpolation. The major deficiency is that there are many possible orientations of body segments that satisfy the goal. For a two segment object within a two-dimensional scene and having freely rotatable joints, there are two poses that permit the tip touch the goal point (see Figure 1.4). For objects in a three-dimensional world or with greater than two junctures, the number of solutions may be infinite. This makes it difficult to obtain the particular pose that an animator desires, whether creating an initial motion or editing an existing one.

There are four goals to strive for when designing a motion editing scheme:

1. The controls should be intuitive and easy-to-use for animators. Where possible, paradigms that the animator is already familiar with should be employed.

2. The augmentation method should preserve the qualities of the source motion. There are subtle timings and emphases inherent in the original action that should be retained.

3. There should be support for tradition, artistic animation principles such as timing and exaggeration.

4. The editing method should be able to work with motions defined by any one of a variety of techniques such as keyframing or motion capture.

Many of the investigations into legged animation have concentrated on the bipedal form. This is understandable because two appendages is the minimum number required for legged locomotion, and is thus the simplest model for which to develop algorithms of motion[1]. There are more animation applications for human figures than for four-legged animals.

Quadrupeds are an interesting subject because of the variety of creatures (having differing biomechanical characteristics) and locomotion patterns. More study needs to be devoted to animating the motions of these animals.

1.3 Objectives of Thesis

This thesis investigates the plausibility of motion warping as an editing method for modifying quadrupedal motion. Motion warping, as proposed by Witkin and Popović, is the controlled deformation of existing motion curves, yielding derived movements that retain the qualities of the original [57]. The system is comprised of the following components:

1. Motion capture is used to acquire the original actions.
2. Motion warping provides low-level editing controls for perturbing these actions.
3. Custom-tailored motion warping controls are defined for modifying quadrupedal locomotions.

The four-legged animal is represented by a hierarchical geometric model composed of rigid body segments connected by joints. Any one of a variety of methods may be used to create animated actions for this creature. The recommended approach is motion capture data obtained by tracing frames of video footage. These movements will be stored in the animation scene format of NewTek's LightWave 3D™, a commercial animation package. The problems of defining the model and its motion templates is not a central concern of this thesis.

The core method for editing these actions is motion warping. The original action is represented by a set of motion curves defined with respect to time; there is one such curve for each parameter of the model. The user can deform a curve by establishing and applying timing, scaling, and offset functions.

1. This is not entirely true. It is feasible to have one-legged locomotion, but this does not occur in nature. Human ingenuity, in its agelong conquest against nature, has prevailed with the invention of the pogo stick. However, this unpopular form of transport is barely worth mentioning.

Motion warping supports all four goals of an editing scheme:

1. The warping functions are specified using keyframing, a pose-to-pose style of approach with which animators are familiar.

2. Because the functions affect the motion curve on a global scale, the qualities inherent in the source movement are retained.

3. The animation principles of timing and exaggeration have a counterpart in the timing and scaling warps respectively.

4. Most techniques for defining animated actions yield motion curves as their final output. Thus, warping can be applied to actions created by keyframing, motion capture, inverse kinematics, or dynamic simulation.

In addition, by warping two motions into alignment and progressively blending, a smooth transition from one action into the next is achieved.

Specification of the low-level motion warping functions is time consuming. Higher level controls can be devised for specialized cases based on the characteristics and limitations of an action. For this thesis, we will consider the problem of quadrupedal locomotion.

The creature traverses a user-defined path and can employ any of a variety of gaits. The path dictates the location and alignment of the animal at any given time. Given an animated rendition of a single period of a locomotion, warping is used to position the animal along the route and set its heading and pitch. Periodic oscillations in the gait are taken into account.

A single gait is applicable over a range of velocities. However, the cycle's frequency and stride length change with speed. These modifications can be made by applying timing or offset warps. When it becomes necessary to change gaits, transitions are also realized using warping. The phase timings of the two gaits are temporally aligned and the leg motions blended progressively over the course of the transition. The test case for this editing system will involve a marten (*Martes americana*)—a small, weasel-like mammal.

Before an approach to editing quadrupedal motion could be decided, an examination of current animation methods was undertaken. This thesis presents a review of the literature of articulated figure animation and animal locomotion. Animation techniques range from traditional, artistic approaches to physics-based simulations to behavioural and emotional systems. Animal locomotion studies provide gait analyses of walking motions. This material furnishes the necessary background for understanding the challenges of animating articulated figures and quadrupeds in particular.

1.4 Evaluation Criteria

In order to effectively evaluate the ideas proposed in this thesis, three criteria will be considered:

1. **Implementation**: The general motion warping algorithm and miscellaneous supporting libraries are implemented and interfaced with LightWave 3D™[2]. High-level parameters specific to controlling quadruped locomotion are defined. Automated means of creating warps from these parameters are derived.

2. **Demonstration of Results**: Motion warping is applied to a couple of test cases. As a preliminary demonstration of the method, a simple bouncing ball motion is augmented. A robotic arm is used to judge the implications of different forms of spatial warps. The quadruped locomotion controls are manipulated to modify sample motions of a marten walking. The creature is made to follow a path over sloping terrain and gradually change its active gait.

3. **Review of the Technique**: Having completed an implementation and conducted experiments, the benefits and shortcomings of motion warping are enumerated. The effectiveness of warping for altering quadrupedal movements is evaluated.

1.5 Contents of Thesis

This chapter has provided a brief overview of the history of animation and the motivations and goals of this thesis. The main objective of this work is to design a system for editing the motions of four-legged creatures via motion warping. This system should support some of the basic principles of animation.

The next chapter provides a comprehensive look at the animation and locomotion research pertaining to legged figures. Some of the motivations behind articulated figure design and gaits are taken from the biomechanics literature. The computer animation research is categorized by the main components of an animation system—motion creation and editing. In chapter three, the structure of the quadruped editing system is discussed in detail. The process of defining and applying low-level motion warps and the design of specialized editing controls for quadrupedal locomo-

2. Lightwave 3D™ is a three-dimensional modelling and animation layout package from NewTek, Inc. The software lets an animator design geometric objects, place them in a scene, specify the motions of these objects, add textures and lighting effects, and finally produce a high quality rendering of the sequence.

tion are addressed. Chapter four presents a series of test cases demonstrating the system in action. For the first example, some simple motion warps are demonstrated on a bouncing ball. The next experiment involves spatial adjustments to the motion of a swivelling arm. The subsequent case involves a marten walking along a user-specified path. The final example consists of a transition from an amble to a bound. Accompanying these tests is an evaluation of the advantages and deficiencies of motion warping, particularly as it applies to quadrupedal motion. The final chapter summarizes the effectiveness of the motion warping editing system. Based on the research results, some alternate directions for addressing the problems of motion editing are suggested.

Chapter 2

Overview of Existing Animation Research

The study of animal locomotion can be divided into two components: describing the movement of animal subjects by observations and measurements, and explaining these motions by combining biology with physics-based mechanics (*i.e.* biomechanics) [46]. In the former case, examinations involve cataloguing the physical structure and variety of gaits of the creature. For the latter, a locomotory model of the animal is developed. Muscles provide the power, propelling lever-like bones which, in turn, apply forces against a surface or swing through the air [2, 23]. These two research approaches are applicable to more than just animal biology—they are also pertinent to the study of animating such creatures. Kinematics-oriented animation methods are comparable to empirical studies in animal locomotion; kinematics is concerned with the creation or reproduction of movements without regard to the physics involved [24]. In contrast, motions created by dynamic simulation adhere to physics-based models inherited from biomechanics.

Before an attempt can be made to animate computer-based animals, it is worthwhile examining some prior work in animal locomotion. This process will bring to light some of the challenges and provide information concerning legged animation that would otherwise need to be discovered independently. To begin, this chapter will consider animal locomotion from a biological perspective with an emphasis on land-based mammals. Studies in this field frequently consider animals' locomotive gaits, which vary depending on the creature's skeletal-muscular structure and desired speed of travel. Next, a review of animation literature with emphasis on bipedal and quadrupedal motion is presented. This overview is divided roughly into two broad categories: motion specification and motion editing. Specification of animated movements can be realized using kinematics or dynamics methods, or a mixture of the two. Editing techniques covered include motion warping, emotion extraction and application, and behavioural control.

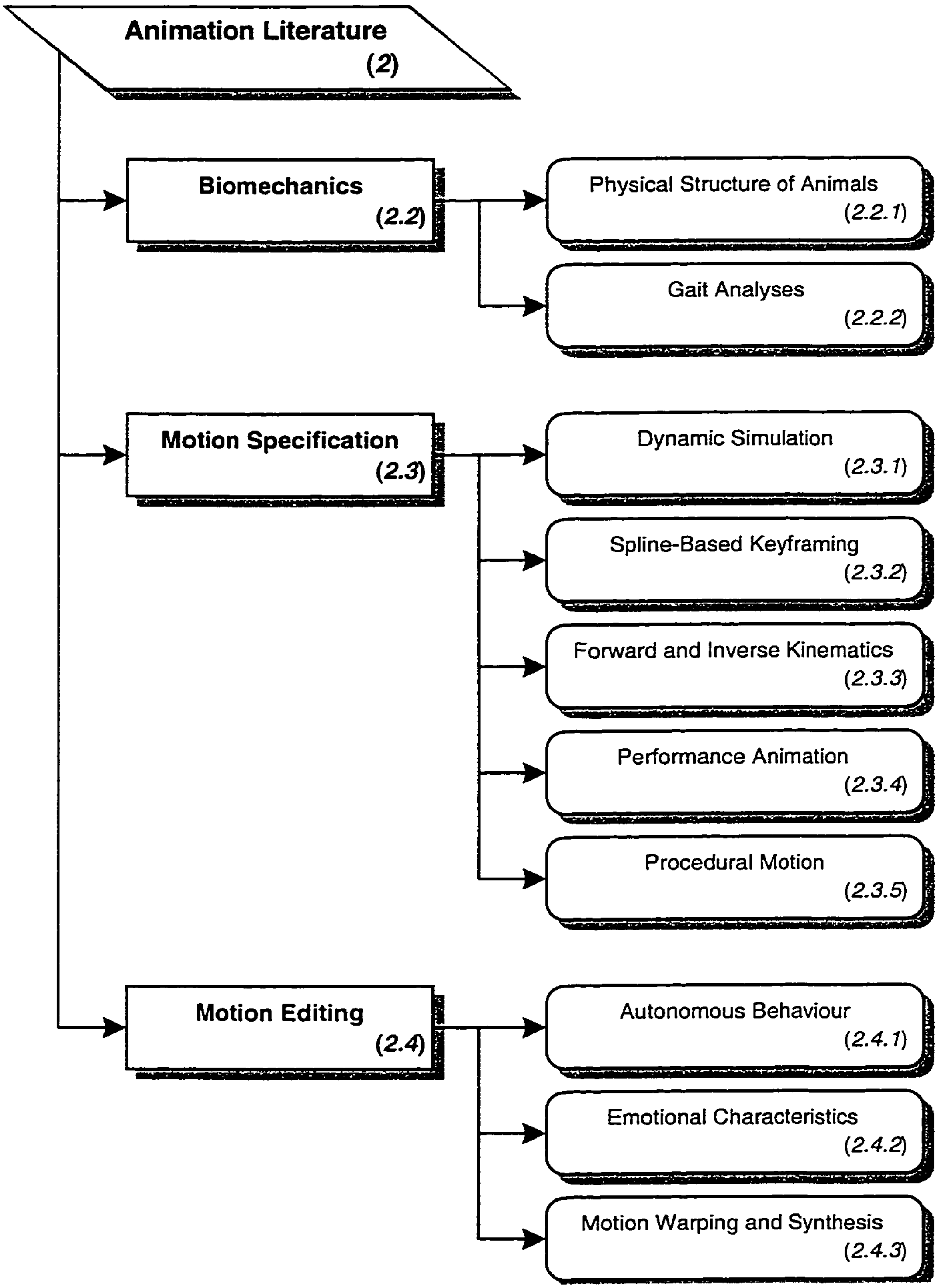


Figure 2.1: **A road map of the animation literature. The material is divided into three main classifications: biomechanics, motion specification, and motion editing. These are subdivided further, and the section numbers for each category given.**

2.1 A Road Map of the Animation Literature

This chapter provides extended coverage of many of the leading articles on the subject of animating the locomotions of legged figures. Each document is placed in one of three categories:

1. **Biomechanics**: These texts deal with the scientific examination of the mechanical aspects of animal biology. They make no reference to animation. (See *Section 2.2—Animal Locomotion.*)

2. **Motion Specification**: These techniques are used to create original movements for articulated models. They may also include some facilities for altering these motions. (See *Section 2.3—Specification of Animated Movements.*)

3. **Motion Editing**: These methods are exclusively for modifying or combining existing animated motions; they provide little or no support for constructing the initial movements. (See *Section 2.4—Editing Existing Animated Motions.*)

These primary classifications are further subdivided. Within the field of biomechanics are studies of the physical structure of animals (*Section 2.2.1*) and gait analyses (*2.2.2*). Motion specification techniques are based on dynamic simulation (*Section 2.3.1*), keyframing with spline interpolation (*2.3.2*), forward and inverse kinematics (*2.3.3*), performance animation involving motion capture (*2.3.4*), and procedural motion with scripting (*2.3.5*). Editing of movements is supported by systems incorporating autonomous behaviour (*Section 2.4.1*), emotional characteristics (*2.4.2*), and motion warping and synthesis (*2.4.3*). The chart in Figure 2.1 graphically shows these subdivisions, listing their section numbers.

2.2 Animal Locomotion

2.2.1 The Biomechanical Structure of Animals

Most of the following discussion on the biomechanics of animals is taken from Gray's *Animal Locomotion* [23]. A legged terrestrial animal is described by Gray as "a loaded flexible beam supported and propelled by four extensible limbs whose range of movement and posture is largely determined by the overriding effect of the weight of the body." The beam in this definition refers to the creature's back, and the limbs are the legs. In the case of mammals, the joints and bones comprising a leg all fall along a vertical plane aligned lengthwise along the body and passing through the hip or shoulder joints. When in motion, the legs tend to remain in this plane. Joints have little rotational flexibility around the axis running along the length of the limb's bones (see Figure 2.2).

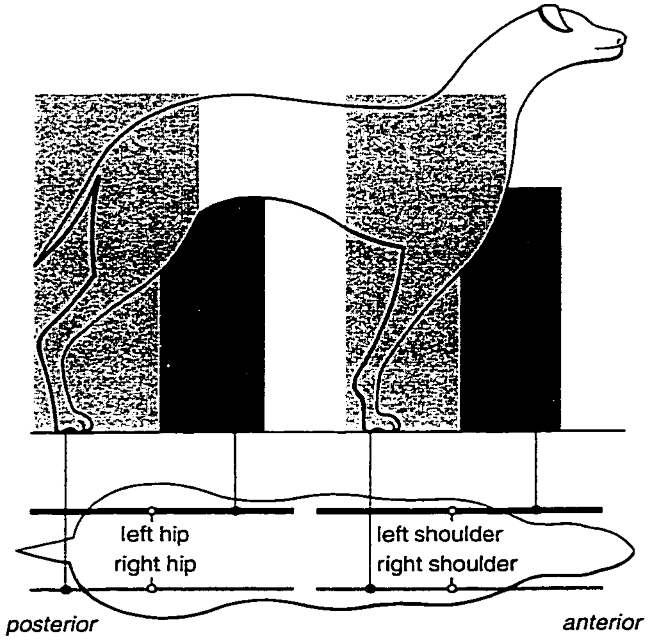


Figure 2.2: **Restrictions in the leg motions of mammals. A leg may only move in its vertical plane and does not twist. The leg planes on the right side of the animal are marked in light grey, on the left side in dark grey. (Diagram adapted from Gray's *Animal Locomotion* [23].)**

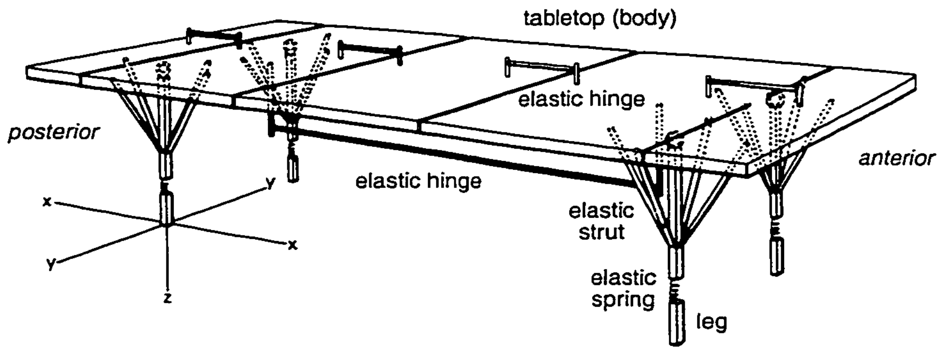


Figure 2.3: **A collapsible table as an analogy of the skeletal-muscular structure of an animal. Elastic springs, braces, and hinges connect the table segments to each other, just as muscles connect the bones of an animal. (Image from Gray's *Animal Locomotion* [23].)**

A good analogy of the skeletal-muscular structure of an animal is a collapsible table (see Figure 2.3). The table, like an animal, is symmetric about its longitudinal axis running from the anterior to the posterior. The surface of the table is made of rigid segments connected by elastic hinges; this represents the back of the animal with muscles for bending the spine. Each of the table's legs are connected by a joint to the tabletop and held in place by a number of elastic braces. This is comparable to the role of the hip or shoulder joints, with extrinsic muscles connecting a creature's leg to its torso. The table's legs have built-in springs that may be extended or contracted. The limbs of an animal act as telescopic struts running from the hips and shoulders to the feet; intrinsic muscles power the joints within each leg. This table model is comparable to the articulated figures used in computer animation.

An animal's legs support the mass of the creature, keep it balanced, and propel the animal during travel. Each supporting leg bears a fraction of the total body mass; this proportion depends on the location of the torso's centre of gravity and the hip and shoulder joints. The muscles controlling each leg joint must provide sufficient tensional forces to counterbalance the gravitational force on the contributing body mass. Since leg segments and joints operate as levers, the required joint torque is directly proportional to the gravitational force; the coefficient is the ratio of length of the gravitational lever arm to that of the muscle's lever arm (see Figure 2.4) [20, 23, 43].

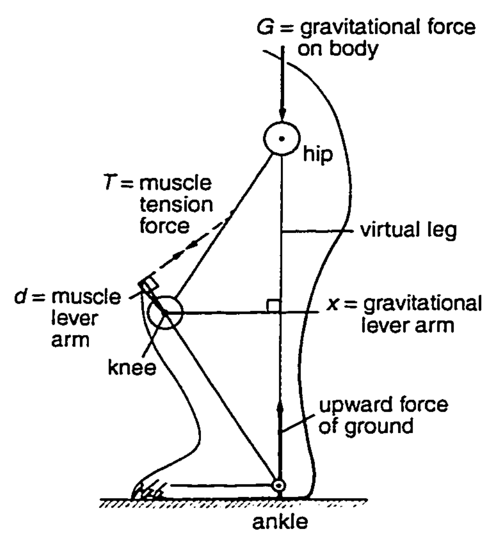


Figure 2.4: **Limbs of the leg act as levers, supporting the body mass. The torque in the knee generated by a muscle, $\tau_{muscle}=T\bullet d$, must be equivalent to the torque due to gravity, $\tau_{gravity}=G\bullet x$ [20, 43]. (Diagram adapted from Gray's *Animal Locomotion* [23].)**

To maintain a stable balance, the centre of gravity of an animal must fall within the triangle or rectangle formed by the three or four legs planted on the ground (see Figure 2.5) [2, 23, 46]. Should the centre of gravity lie outside this polygon, the creature will tip over. In addition, to prevent rolling about the longitudinal axis, the left and right legs must support half the body mass apiece. Similarly, to prevent pitching, the anterior and posterior legs must bear a fraction of the body mass based on their distance from the transverse axis of the centre of gravity. The last requirement is that a line between the centres of pressure of the left and right limbs must intersect the centre of gravity. For example, if only the two left legs and the front right leg are supporting the animal, then the centre of pressure for the left limbs is at the gray dot shown in Figure 2.5.

As an animal walks around, its legs apply horizontal forces that drive the animal onwards. The stance legs, being fixed to the ground, rotate about the hips or shoulders as the body moves forwards. In turn, each leg must be lifted, quickly swung forwards, and placed on the ground. Strict stability can only be maintained when at least three legs are in contact with the ground. However, during rapid motions such as running, there may be fewer than three stance legs at a given time. Dynamic equilibrium is maintained by compensating for imbalances over the entire running cycle.

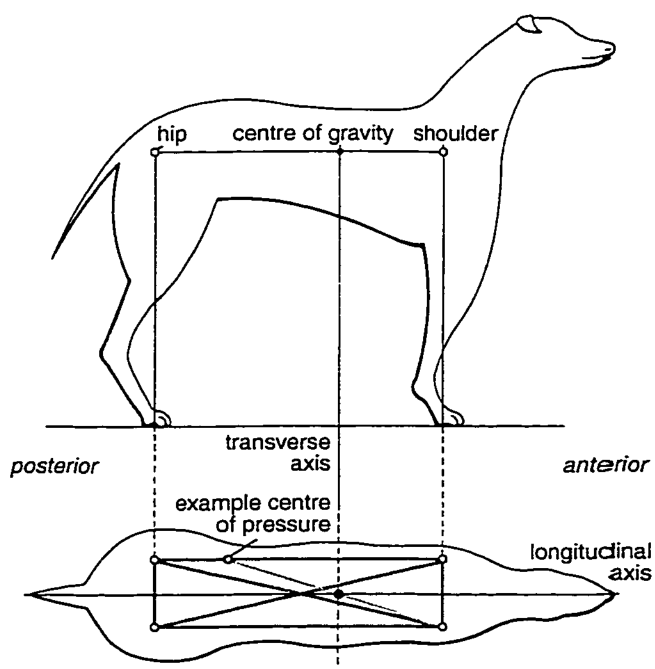


Figure 2.5: **To retain balance, an animal's centre of gravity must fall within the convex hull formed by the hip and shoulder joints of the supporting legs. The possible convex hulls are marked by dark grey lines. (Image adapted from Gray's *Animal Locomotion* [23]).**

In mammals, there are two distinct skeletal-muscular builds: graviportal and cursorial. Graviportal animals such as elephants are heavy and have thick bones. Cursorial mammals such as horses have long, light limbs that are better adapted to running—the speed at which an animal can run is dependent upon the length of its legs. By standing on the front of the foot (*i.e.* a digitigrade posture) rather than the heel (*i.e.* a plantigrade habit), the effective length of the leg is increased.

2.2.2 Gait Analysis

To locomote, animals move their legs in a periodic fashion. At times, some of the legs push against the ground, propelling the creature onwards; these are designated the stance or supporting legs. Others are lifted, swung through the air, and positioned in preparation to become stance legs. A gait is defined as the coordinated, rhythmic pattern of movements of an animal for the purposes of locomotion [46]. A gait is deemed symmetric if the left legs move through the same pattern as the right, albeit out of phase.

A gait may be characterized by the timings of the animal's constituent movements, the postures of the limbs, and the forces and torques generated by the legs. Concentrating on the former, the order in which the feet make ground contact and the duration of each leg's stance phase need to be enumerated. To better convey these timings, researchers such as Gray [23], Sukhanov [46], and Alexander [1, 2] have adopted diagrammatic notations. An example of a typical quadruped walking gait is depicted in Figure 2.6.

During a single period of the locomotory cycle, an animal completes one stride. The number of strides achieved per unit time is called the stride frequency. The stride length is the distance spanned from the tip of one footprint to the associated footprint in the subsequent stride [2]. In the diagram on the right in Figure 2.6, the stride has been subdivided into eight phases; each phase represents a transition in the pattern of the supporting legs. The columns of bars denote when a leg is in its stance stage; an absence of a block indicates that the leg is in its transport state—that is, the foot is not in contact with the ground. The terms relative phase and duty factor refer to the time a foot touches down and the duration of the leg's support stage, respectively, expressed as a fraction of the cycle. The central graphics shows the layout of supporting legs for each progressive phase. The arrow, representing the body, points towards the anterior and the dots denote the current stance legs.

A more compact representation of the gait timings is presented in the left diagram in Figure 2.6. The circular nodes represent the legs. The numbers within are the order in which the limbs become supporting legs; arrows connect the nodes in this sequence. The fractions beside each node denote the relative phases of each leg.

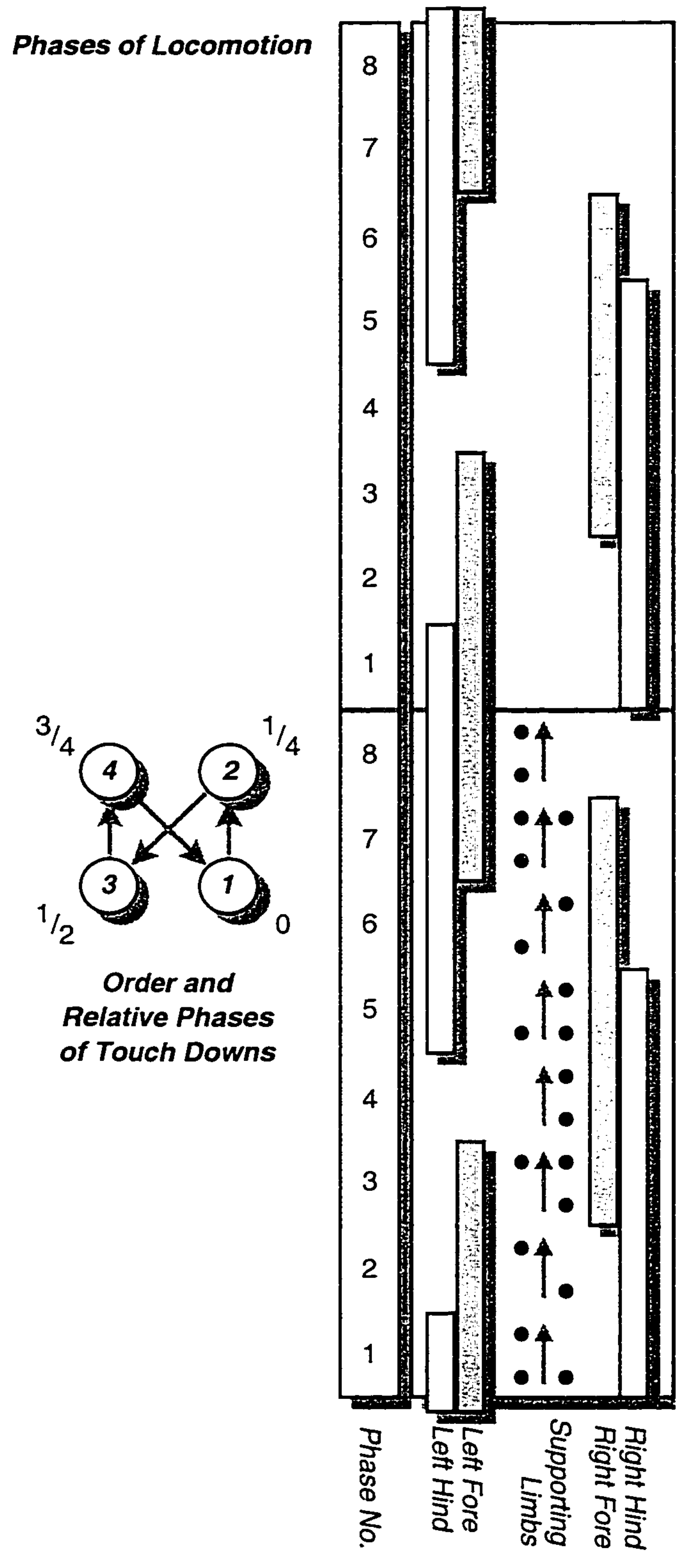


Figure 2.6: **The phases of locomotion for a typical quadruped walking gait. The left diagram shows the order and relative timings of the feet as they make ground contact. The right diagram illustrates the support and flight phases of each leg. (Graph based on a diagram from Gray's *Animal Locomotion* [23].)**

Figure 2.7: **A photographic sequence of a cat in transition from a trotting to a galloping gait, taken from Plate 717 of Muybridge's *Animal Locomotion* [39].**

Photographic studies, such as those completed by Muybridge and presented in his *Animal Locomotion* books, have provided the visual evidence to compile these gait diagrams [39]. During the 1870's, Muybridge captured nearly 800 photographic sequences, comprised of some 20 000 images, of the movements of human and animal subjects. Some examples of these progressions appear in Figures 1.2 and 2.7. The images were photographed by a series of cameras, oriented in a line. Electric circuits activated the shutter of each camera in the sequence at regularly-spaced intervals of time. These photographic collections are important from an animation perspective because they capture the essence of an animal's motion. An animator can transcribe the poses to his animated animal, resulting in movements that appear realistic.

An animal may use one of a variety of locomotory gaits. The choice of gait is dependent upon several factors:

- **Skeletal-Muscular Structure**: As was mentioned in the previous section, each species of animal has a unique build which limits the kinds of gaits it can employ. For example, elephants, due to their large size, are limited to an amble-style gait (see Figure 2.8). Camels, having long legs and large feet that can become entangled, use a pace pattern rather than the more common trot or canter [2].

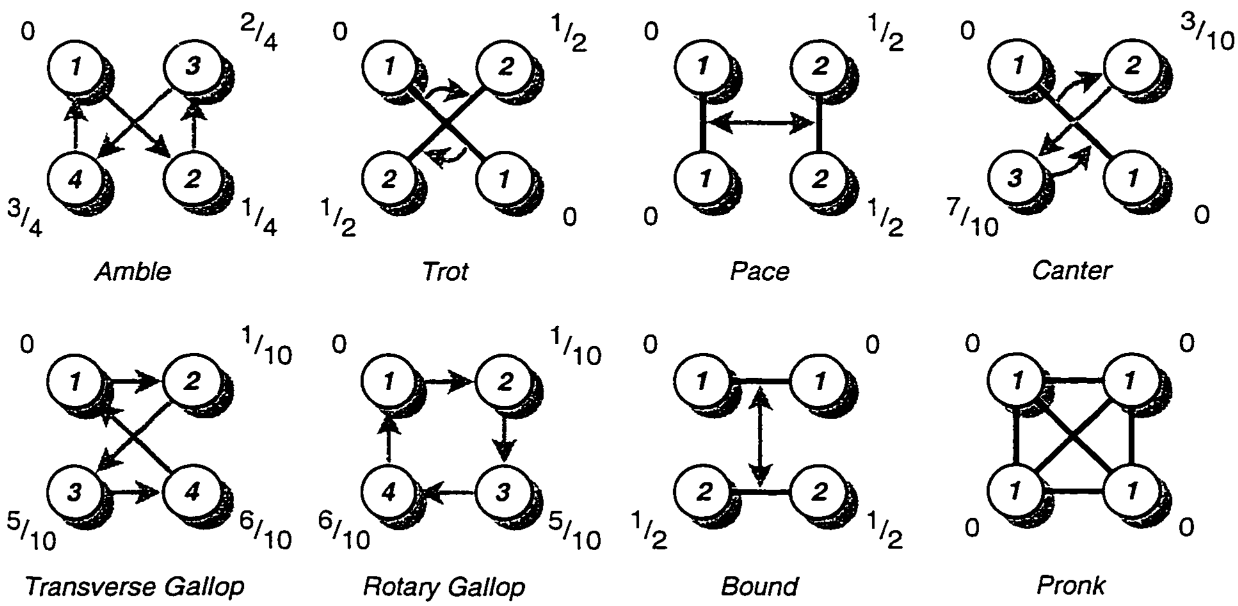


Figure 2.8: **A variety of running gaits and their relative phases. The choice of gait employed by an animal is dependent upon its skeletal-muscular structure, psychological state, and the desired speed of the animal. (Diagrams adapted from Alexander's *Locomotion of Animals* [1, 2].)**

- **Psychological or Emotional State**: The choice of gait may also be influenced by psychological or emotional aspects. For instance, when startled, an antelope pronks (see Figure 2.8) [1, 2]. A hunting lioness slinks to stealthily approach its prey.

- **Speed**: Some gaits are better adapted to high speed travel. In some cases, it is necessary to switch gaits due to limitations of physics. At increased speeds, an animal's gait becomes uneconomical from an energy cost perspective; an alternate gait can reduce the energy requirements [2].

As a simple demonstration of the need for different gaits for various speeds of locomotion, consider a walking biped. A walking gait requires there to always be a support leg throughout the stride. The stance leg inscribes a circular arc centred at the foot and having its circumference at the hip (see Figure 2.9). Assume the person is walking forward at a speed v and the length of his leg is l. Then to maintain its circular motion at the apex, the body must accelerate towards the centre of the arc with magnitude v^2/l [20]. Gravity is the only force that can provide this acceleration[1]. Thus, the walking speed is limited to

$$v \leq \sqrt{g \cdot l} \tag{2.1}$$

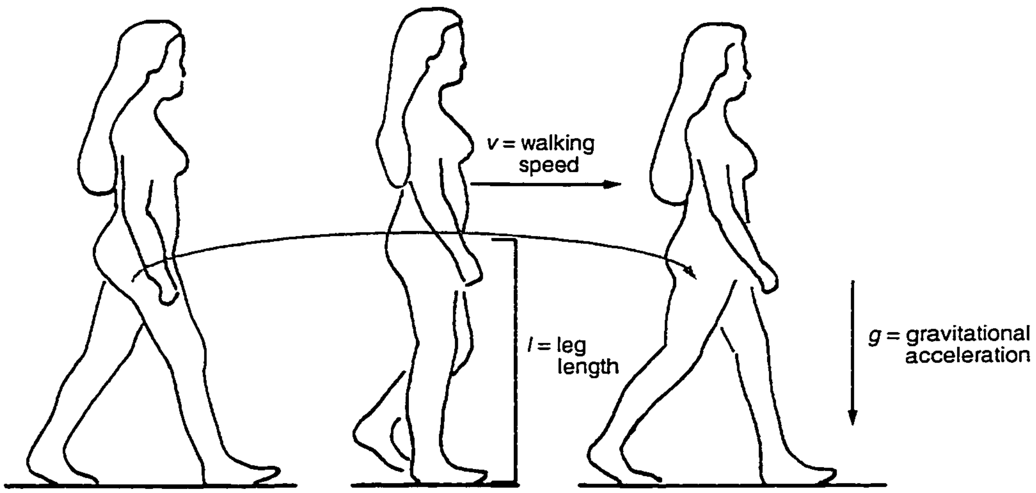


Figure 2.9: **The maximum speed of walking is bounded by gravitational limitations. If the speed is too great, then gravity cannot provide the necessary acceleration to ensure the body maintains its arc of motion. (Diagram adapted from Alexander's *Locomotion of Animals* [2].)**

1. Since the foot is not fastened to the ground, the leg cannot pull the body towards the centre of the arc.

where g is the acceleration due to gravity. A typical male, for instance, can walk at a maximum rate of 3.0 m/s. To move any faster requires switching to an alternate gait—running. At times during a running stride, there are no stance legs.

As a quadruped's speed of locomotion increases, the timings of its leg movements change in three ways [23]:

1. The period of a stride is shortened. More specifically, the duration of the stance and flight phases are both reduced.
2. The time span of the stance phase decreases proportionally more than that of the flight state.
3. The spacing between the stance phases of the fore and hind limb on the same side of the body increases. The spacing between stance states of the fore and diagonally opposite hind limb decreases.

Some of these transformations are evident when comparing the walking gait from Figure 2.6 with that of a transitional gait between a walk and a trot, shown in Figure 2.10.

Alexander has noted that despite the wide range of animal builds, there is an invariant relationship between stride length and locomotion speed [3]. Let λ be the stride length and v the speed of locomotion. Expressing these in a non-dimensional form inversely proportional to the hip height, h

$$\hat{\lambda} = \frac{\lambda}{h} \qquad \hat{v} = \frac{v}{\sqrt{g \cdot h}} \tag{2.2}$$

the association between stride length and speed of motion is then given by

$$\hat{\lambda} \cong 2.3\, \hat{v}^{0.6} \tag{2.3}$$

2.3 Specification of Animated Movements

A major goal when animating animals is to reproduce the appearance of the actual creature, both in terms of the model and the motion. To realize this objective, the obvious approach, in light of the biomechanics literature, is to strive for realism by way of physical approximations. Thus, a large amount of the computer animation research has concentrated on dynamic simulation to impart movements on a model. However, physics-based techniques are hampered by complexities computational and experiential in nature. In addition, there is a demand for ways to instill artistic qualities into animations. This leaves open the possibility for alternate forms of motion specification.

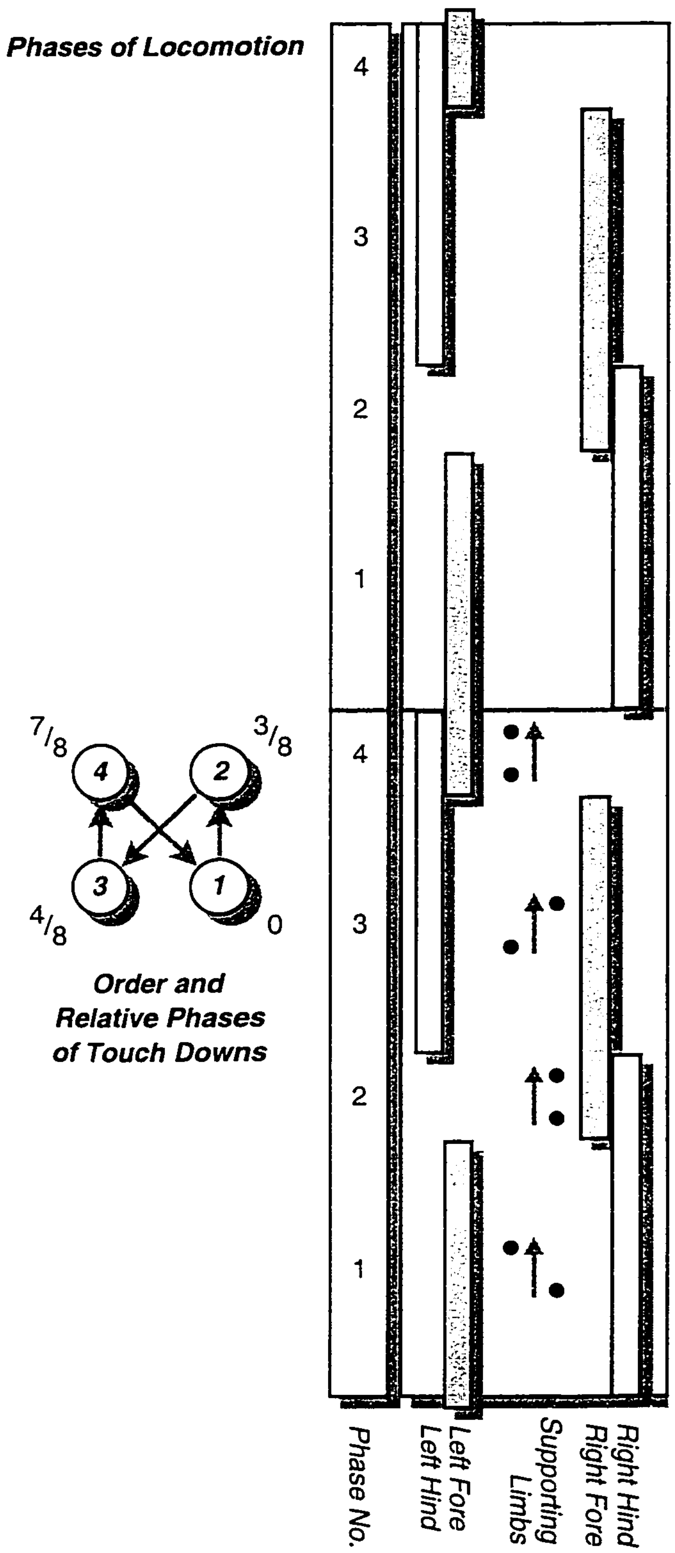


Figure 2.10: **The phases of a transitional quadruped gait, changing from walking to trotting (based on a diagram from Gray's *Animal Locomotion*) [23]. Note the reduced duration of the stance phases and the closing proximity of diagonally opposite limbs' support states.**

Before computers could provide extensive computational power, movements were frequently constructed using conceptually simple kinematics techniques. Spline-based keyframing and inverse kinematics may be grouped under this banner. Motion capture is another approach that has become increasingly commonplace, especially as the reliability of equipment has improved. Using motion capture, spatiotemporal data describing a movement is acquired directly from a real person or animal and is overlaid onto its computer model counterpart. Procedural techniques describe primitive actions using mathematical functions. Motions are formed by combining these actions in synchrony.

2.3.1 Dynamic Simulation

By embracing concepts from biomechanics and adhering to principles of physics, dynamic simulation can be used to generate visually realistic motions. As with all of the animation methods that will be described, the object to be animated is an articulated model. The figure is represented by a tree hierarchy containing rigid segments connected by rotatable joints. The model is frequently simplified due to the mathematical complexities of dynamic simulation [4]. The task of creating a movement for the figure is decomposed into determining the placement and orientation of the torso (at the root of the hierarchy) and the angles of the joints over time. Alternatively, this problem can be defined in terms of the forces and torques required to drive the figure into goal poses. To find a solution, dynamic simulation requires some additional information concerning the model: the lengths, masses, and positions of the centres of mass for each of the segments [4, 13].

Equations of motion express the torques and motions of model segments in terms of their accelerations and velocities, linear and rotational, and external influences such as gravity [4]. The equations are recursive because the motions of the distal elements of a limb (*e.g.* the foot) affect the torques and forces on segments proximal to the torso (*e.g.* the hips). These equations form the low-level core of a control system that guides the motion of the figure.

Before these equations can be put to use, a decision needs to be made concerning what the input parameters should be. The values assigned to these parameters specify and constrain the motions to be generated. Possible high-level controls fall into two categories:

1. forces and torques, and

2. accelerations, velocities, and positions.

More complex constraints may be automatically calculated from these parameters and from the properties of the articulated figure.

To interject additional guidance, state machines are often used as intermediate level controllers [13, 29]. The states shift as the articulated figure achieves specified goals; each phase defines a new set of constraints that guide the motion.

When converted to matrix form, the equations of motion for the system have as many rows and columns as degrees-of-freedom. These systems involve nonlinear differential equations and usually do not have a readily obtainable solution. Simplification and approximation techniques are employed to solve for the variables. One possibility is to minimize the number of degrees-of-freedom by using a simplified articulated model. The results of the dynamic simulation are then combined with kinematic techniques to obtain the movements of a more complex figure. To converge on a solution, the system may be worked out for each time step in turn or in an iterative, global manner over the duration of the motion.

One of the most difficult tasks of dynamic simulation is devising a control algorithm that yields a stable system and produces the desired motion. Many authors have hand-crafted their algorithms for walking, running, and other activities [13, 29]. However, some progress has been made in automating this challenging task [52].

2.3.1.1 Bipedal Locomotion

One of the earliest attempts at applying dynamic simulation to create articulated figure movements was made by Armstrong and Green in 1985 [4]. Their goal was, given a minimal amount of accurate input data, to produce realistic motions that reflected the model's environment.

To this end, the authors formulated three equations of motion. The first relates the torque of a model segment—the product of its moment of inertia and angular acceleration—to the torques and forces contributed by its child segments, gravity, its relationship to the inertial frame, and the user. The second equation expresses the force exerted on a component by these same contributors. The final formula describes how a segment's motion (*i.e.* acceleration) propagates to child components.

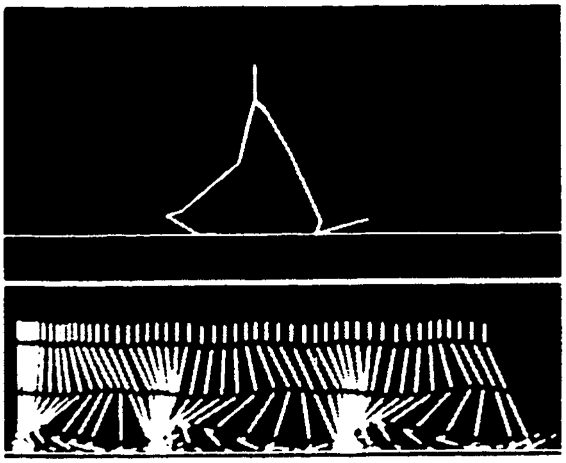
Armstrong and Green decided that the input parameters to their system should be torques and forces applied to the articulated figure's segments. Thus, the equations of motions were solved to obtain two values, namely the angular acceleration of and the force on these segments. To simplify the derivation, a linear relationship was assumed between the linear acceleration and each of these two variables.

The simulation is executed for each time step in turn. Each segment's angular acceleration and force are calculated, proceeding breadth-first from the distal segments to the elements proximal to the torso. Then, the motions of proximal model segments are transferred onto their children. Care must be taken to choose a sufficiently small time step to avoid high-frequency aliasing effects that accompany unstable motions.

As a demonstration of the technique, an animation of a finger tapping on a surface was generated. This was achieved by periodically applying an external, upward force to the fingertip for a short duration. Joint rotational limits were enforced by introducing resistive torques at extreme orientations.

In an effort to harness the best of the major animation approaches, Bruderlin and Calvert generated human motion using a hybrid of dynamics and kinematics in their KLAW (Keyframe-Less Animation of Walking) system [13]. Its multilayered architecture features, at the uppermost tier, simple specification of motion via a few intuitive parameters (*e.g.* walking speed and step length) and attributes (*e.g.* distance between feet). The walk is then decomposed into the characteristic support states and leg gait phases.

A simplified model of the human form featuring a straight, telescopic leg is used in the dynamic simulations. The equations to be solved and their constraints vary, depending on the current phase of the model. The dynamic simulation computes the torques affecting key joints of the model; this is sufficient for determining the gross walking movements. At the same time, kinematics is applied to position the legs of a complete articulated figure, whilst also providing feedback to the dynamic simulation. An example of a natural walking gait generated by the system is shown in Figure 2.11.



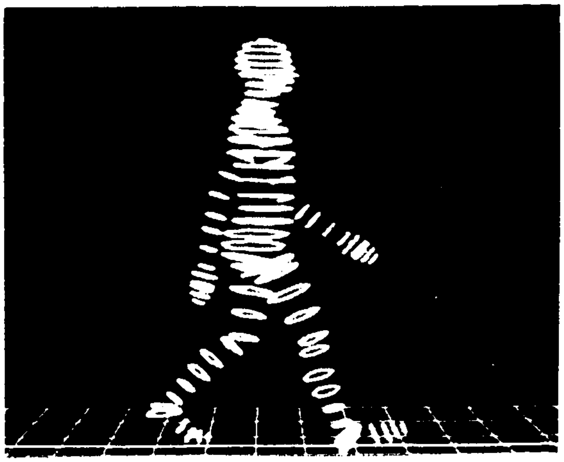


Figure 2.11: **A natural walking motion at a speed of 5 km/h generated using KLAW. (Images from Bruderlin and Calvert's "Goal-Directed, Dynamic Animation of Human Walking" [13].)**

In more detail, at the top level of the system, the input parameters are used to determine the state-phase timings. The states cycle from double support—both feet touching the ground—to single support—one foot is in contact with the surface. The phases oscillate from stance to swing for each leg. Empirically-derived equations are employed to find the duration of each phase. Equations for the length and angle of the legs at a phase's endpoints, based on the user-specified step length, are derived geometrically and trigonometrically. These equations embrace the principle of step symmetry—the angle of both legs with respect to the vertical are equal. The feet are introduced using kinematical methods.

The above values are computed and used as constraints in the dynamic simulation. Bruderlin and Calvert apply separate Lagrangian equations of motion for each leg. The system of equations used depends on the current phase of the leg. A solution is converged upon using iterative approximation techniques applied over the duration of the phase. This yields the torques necessary to drive the articulated figure. In the stance phase, the leg is treated as a spring and damper model, and the connecting hip is made to follow a sinusoidal curve. The foot is added using inverse kinematics. The swing phase is broken down into three components. First, the ankle follows a curve until the toe is under the knee. The knee is then rapidly straightened. Finally, a small moment at the hip pushes the heel of the foot onto the ground.

Hodgins *et al.* have concentrated on developing control algorithms for dynamic techniques to render physically-realistic, natural-looking animations of athletes [26, 29]. The authors highlight some of advantages of dynamic simulation as featured in their systems:

- The same control algorithms can be used to derive similar movements (*e.g.* running at different speeds) whilst maintaining physical accuracy.
- On-line interaction allows the animator to guide the characteristics of the action.
- Secondary motions—supplementary movements not required to achieve the primary motion goal, but rather affected by the primary motion—may be introduced into the simulation.

Their articulated figures feature joints with up to three degrees-of-freedom and powered by torque-supplying muscles. The masses and moments of inertia of these components, taken from biomechanics literature, are computed using medical data and assuming uniform density.

A typical control algorithm contains a number of components. A state machine represents the phases of the action and the set of applicable control laws. The control laws specify the positions of key limb segments or angles of joints based on the difference between current and desired values. Inverse kinematics is used to find the desired poses for the rest of the segments. Proportional-derivative servos with user-defined gains calculate the torques required to propel the joints towards these positions. Inactive limbs are used to stabilize the system (*i.e.* for balance).

As a demonstration, Hodgins *et al.* consider a running motion. Each leg of this cyclic movement passes through several states. As a leg enters its flight phase, the foot rolls onto its toes (toe contact), the knee pushes (unloading), and the leg is lifted and swung forwards (flight). As the heel contacts the ground (heal contact) at the start of the stance phase, the knee bends (loading), and the foot rolls until it is once again flush on the ground (heel and toe contact).

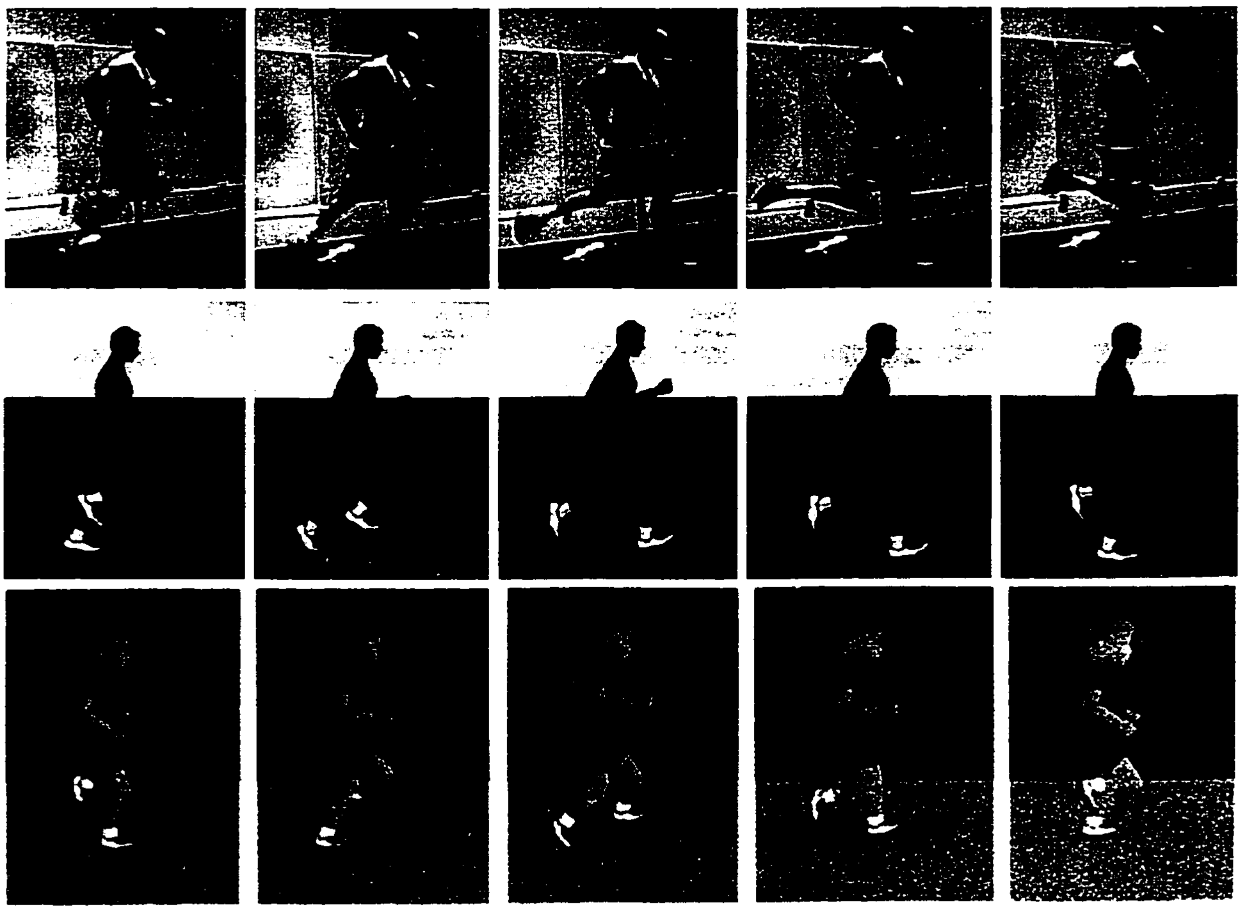


Figure 2.12: **A real and virtual athlete running, the latter created using Hodgin *et al.*'s running control algorithm. (Images taken from Hodgins' article "Animating Human Motion" that appeared in *Scientific American* [26].) The last series depicts the same system adapted for a child. (From Hodgins and Pollard's "Adapting Simulated Behaviors For New Characters" [28].)**

Given the speed and direction as input, state-specific control laws are applied to obtain the appropriate joint torques. One control system is used to position the foot as it contacts the ground at the completion of the leg's flight stage. A kinematic expression is used to determine the position of the heel strike based on the current and desired speed of the runner. Ground speed matching is employed so that the relative speed of the foot to the ground is negligible at the time of contact. Inverse kinematics is applied to find the leg pose.

During the stance stage, the control system represents the knee as a spring, converting the kinetic energy of the landing into its potential equivalent. The potential energy is later used to propel the body forwards. Meanwhile, the in-flight leg is kinematically shortened and its associated hip twisted to reduce the torque on the upper body. Shoulder joints are oscillated in synchrony with the legs.

Some sample frames from a running gait generated using this control algorithm, together with equivalent frames from a real runner for comparison purposes, are shown in Figure 2.12. The authors also describe control processes for bicycling and vaulting.

Hodgins and Pollard expanded the aforementioned research to find automated means of adapting their dynamics-based, athletic behaviours to new characters [28]. A control system for one figure becomes unstable when applied another character with differing physical proportions. Scaling of the geometry and mass is used to make a preliminary estimate of a new control system.

The problem of reusing the same motions on articulated figures of various builds has long plagued the animation industry. Disney animators discovered that tracing the recorded movements of a real person could not be directly transferred onto an animated character. The quality of the motion visibly degraded [38]. Likewise, a controller tuned for one character will fail to work for another because the servo gains are inappropriate for the second character's differing geometric and mass characteristics. The motion of the new figure rapidly collapses.

To solve this deficiency, first the model's physical parameters are scaled exponentially according to a constant factor—the ratio of the leg lengths of the two characters. These parameters include the state (*e.g.* joint angles), servo gains, user-specified values (*e.g.* forward speed), constants, and the time step. Secondly, mass scaling is applied on a per body part basis. The servo gains for a segment are scaled by the mass or moment of inertia ratio. This is only an approximation, but serves as a good initial guess for the new control system.

Finally, the newly derived control system must be fine tuned to ensure a stable motion. Only a restricted space needs to be searched. In the case of running, the varying parameters include ground speed matching, body pitch, flight duration, and forward speed. Simulated annealing was found to be the most successful method for deriving the new control system. The process was progressively applied to intermediate characters and ended with the final goal figure. Figure 2.12 shows a running motion adapted for a child.

The advantage of this technique is that it takes into account knowledge about the behaviour to limit the search space. Alternatively, motion capture data can be scaled physically, but not by mass. The authors believe that both length and mass scaling are required to obtain reasonable motions. However, later research by Gleicher demonstrates otherwise [22].

2.3.1.2 Multi-Legged Locomotion

The work presented in the preceding section concentrated on the simulation of two-legged human actions. Research into the four-legged locomotions employed by the majority of the animal kingdom has also been undertaken.

Kokkevis, Metaxas, and Badler developed a framework whereby a four-legged animal could apply variable gaits to locomote over uneven terrain [32]. To add a degree of primitive autonomy, the animal can be given goals (such as to chase a ball) which the system tries to satisfy. Dynamics are used to calculate the body force required to position the animal. This force is then distributed to the stance legs. A controller enforces the gait sequence using a state machine.

The first operation is to find the gross body force and torque. Six integral plant controllers are used to regulate the body's orientation, velocity, and its distance above the ground. This force and torque is then allotted to the stance legs; the appendages push against the ground to provide the requisite forces. Because the legs are inclined and can only provide a limited amount of force and torque, it may not be possible to produce the required forces. Linear programming involving a cost function to be minimized is used to find the best distribution. In addition, inadequate friction limits the magnitude of the forces parallel to the ground. Meanwhile, kinematics is employed to pose the in-flight legs.

A state machine is used to define the constituent leg movements, and a gait controller enforces the sequencing of the legs. The authors use an ambling gait for their locomotions; a stable balance is ensured by having the centre of gravity of the animal always fall within the triangle of support.

Some representative frames of a dog walking over a ramp are depicted in Figure 2.13. The authors found the motions appeared stiff, possibly due to the rigid body model used.

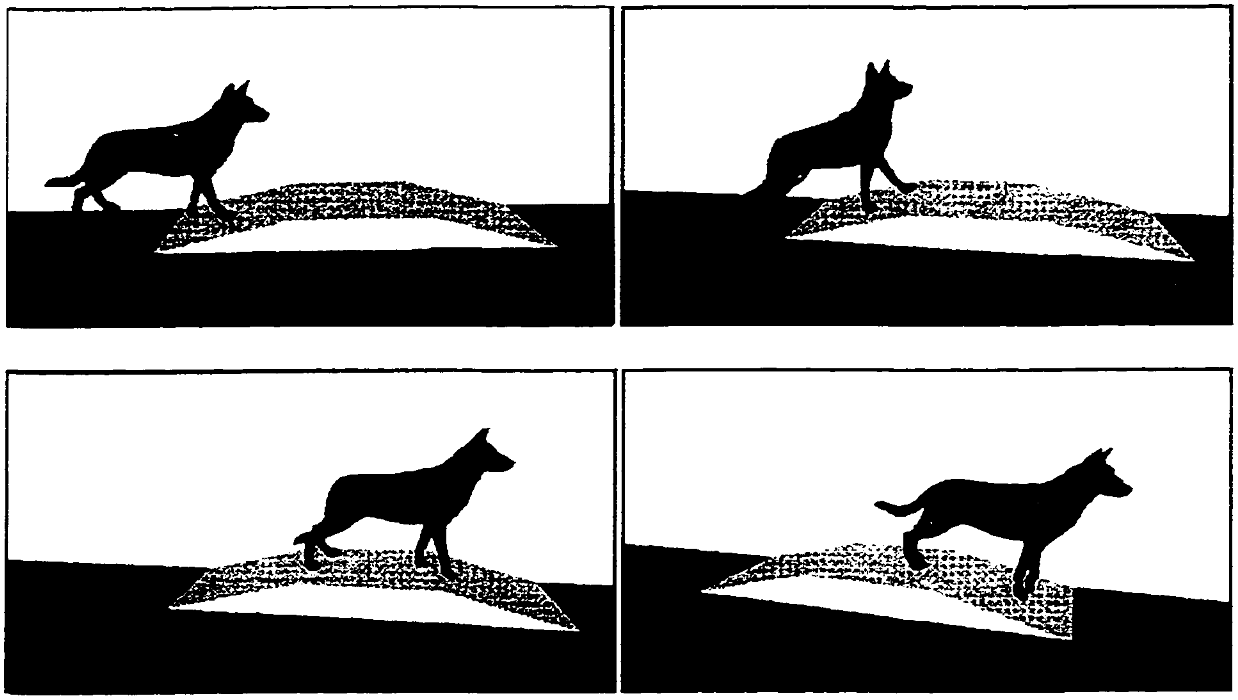


Figure 2.13: **Some frames from an animation of a dog walking over a ramp as generated by Kokkevis, Metaxas, and Badler's system. (Images from "Autonomous Animation and Control of Four-Legged Animals" [32].)**

Torkos and van de Panne developed a quadruped animation system that generates legged locomotions that follow user-defined foot prints and timings [49, 50]. The torso is treated as two mass points —one at the hips and the other at the shoulder—connected by a spring. The optimal trajectory of the body along the path of footsteps is found by dynamic simulation. The optimization equation tries to minimize any leg forces that are deemed implausible or uncomfortable. After the path of the hip and shoulder joints are found, the legs can be added, connecting the hip and shoulder points to the footprints. The authors position the legs using an example-based inverse kinematics technique; sample motions chosen from a database limit the possible leg poses.

The total force exerted by a hip or shoulder point is simply its acceleration, based on the current estimated trajectory, scaled by its mass. Contributing to this force is the spline spring, gravity, and the legs. The leg force is solved for, as the forces contributed by the spline spring and gravity are already known or easily calculated. The infeasible components of the leg force are then determined. Forces not in the plane formed by the representative leg lines passing through the mass point are deemed implausible. An additional penalty is charged should the body collide with the ground or the legs deviate too far from their neutral stance. To improve the motion path of the hip and shoulder points, small changes are made to the trajectory spline. These alterations are retained should they reduce the implausible leg forces.

Figure 2.14: **Torkos and van de Panne's system for creating quadruped locomotions according to user-defined footprints. (Image taken from "Footprint-based Quadruped Motion Synthesis" [49, 50].)**

Figure 2.14 shows a quadruped traversing variable terrain, placing its feet in the marked footprints.

Sims uses a novel scheme of evolutionary survival and adaptation to create articulated creatures fit to carry out particular actions [44]. A genetic language is used to define the "building blocks" that make up the creature's physical characteristics (*i.e.* DNA genotypes). The language uses directed graphs with each node defining a rigid body part, the type of joint connecting it to its parent, and a list of connections attaching siblings to the part.

A creature has both a central brain as well as distributed neural constructs in its elemental parts. These neural components take input from sensors that communicate joint angles, contact detection, and photosensitivity, and respond with output in the form of torques and forces to drive a dynamics-based model. The brain uses a neural network with a variety of neuron types (*e.g.* sum, maximum, *etc.*). The effectors in the body are also connected to neurons; each neuron drives one degree-of-freedom of a joint.

Initial genotypes are created by the user, randomly, or from a previous population of creatures and then instantiated. After a period of time has elapsed, the creatures' fitness is evaluated. For example, the measure of fitness for walking is the fastest movement along a straight line. The highest ranking creatures are "mated" and their siblings form the next generation to be evaluated. In mating, the genotypes are combined by grafting or crossover operations. Further mutations are introduced by randomly altering internal node parameters, adding new nodes, changing neural connection parameters, and inserting and removing connections.

Examples of walking machines evolved using this method are depicted in Figure 2.15.

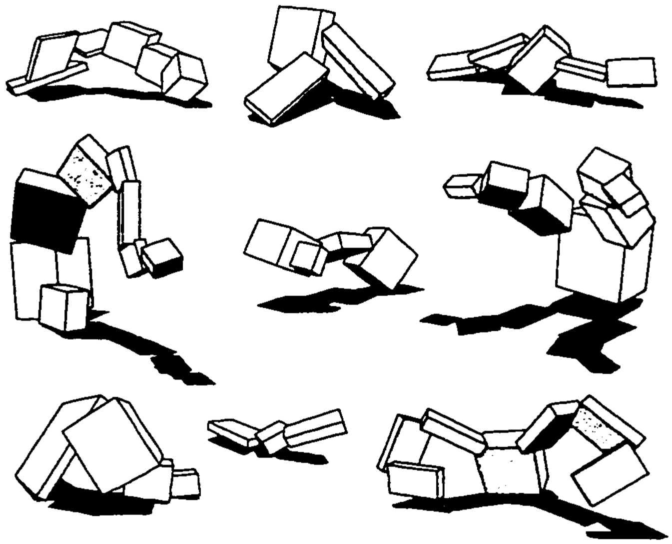


Figure 2.15: **Walking creatures developed using Sims's evolutionary approach. (Images acquired from Sims's "Evolving Virtual Creatures" [44].)**

2.3.1.3 Automating the Design of Dynamic Controllers by Optimization

Space-time constraints is a technique introduced by Witkin and Kass for automatically finding physically-accurate motions satisfying user-defined constraints [56]. These constraints range from what the object needs to accomplish (*i.e.* where it needs to move), how it should achieve this (*i.e.* its mannerism), the model's physical structure, and its resources (*e.g.* internal muscles). A solution motion that adheres to Newton's laws of motion is found using a global optimization process.

To enumerate the motion problem, the user is provided with a tool kit of black boxes. These boxes have inputs, outputs, and contain symbolic expressions for converting the former to the latter. These expressions represent equations commonly used in describing the dynamics of a system. The boxes also provide symbolic solutions to the derivatives of their expressions, as will be required for the motion solver. The user instantiates and connects the boxes together to describe the motion problem.

The user chooses constraints and an optimization equation, both expressed in terms of the unknown, independent variables to be solved for. A typical optimization equation might be to minimize the force, energy, or power required by the object to complete the motion. Constraint expressions, which must be equal to zero, impose rigid requirements on the motion. For example, one common constraint is that an object must assume some position at a particular time. Constraints are also used to express the relationships between the independent variables.

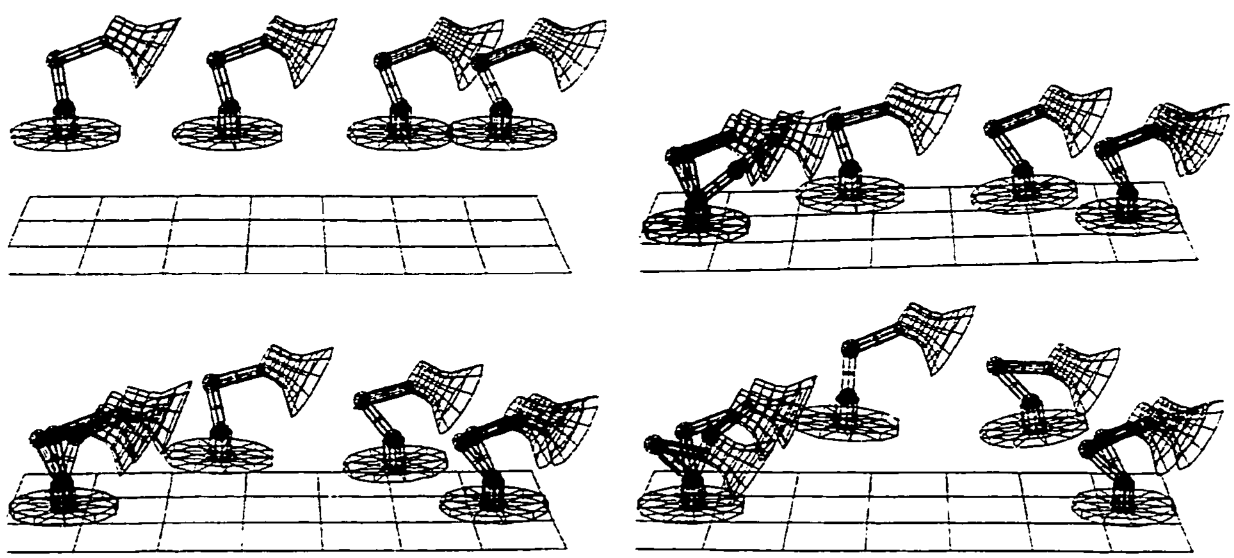


Figure 2.16: **Some iterations of Witkin and Kass's method showing the progression towards satisfying the space-time constraints of a hopping lamp. (Images taken from "Spacetime Constraints" [56].)**

The system is solved in an iterative, global fashion. The procedure finds the second-order derivative of the optimization equation and first-order derivative of the constraints with respect to the independent variables. The former is projected onto the null space of the latter to find the values for the these variables. In essence, first a solution is found that satisfies the hard constraints. Then, the optimization equation is minimized whilst ensuring the constraints are held.

As a demonstration of the system in action, Figure 2.16 shows the results of several iterations of the optimization process to find a hopping motion for a lamp.

Van de Panne, Kim, and Fiume propose a means to automatically find the controllers for periodic motions [52]. Indeed, most locomotions are cyclic in nature, so non-periodic controllers may be decisively eliminated in the search for an effective control system. Motions tend to proceed through a series of states characterized by the articulated figure's poses. The authors' system randomly generates these state pose graphs, tests them according to some optimization criteria, and then continues to refine and retest the most promising graphs until a suitably efficient controller is found.

In more detail, the user specifies the masses of the articulated figure's components, its joint limits, and the gains for proportional-derivative joint servos. Other inputs include the period and number of pose states for a motion. The system then randomly generates pose control graphs with states of equal duration. Poses may remain static over the duration of a stage, or may linearly change between states; there is no benefit to either approach in the simulation. The dynamic simulation is then run and the resulting motion evaluated by optimization criteria (*e.g.* fastest straight-line walk). The best-performing pose control graphs are then modified by making minute adjustments to a single joint orientation of one of the state poses. The modified graph is tested again, and the refinements continue until a suitably affective controller is derived. These processes are known as generate-and-test and modify-and-test.

Van de Panne, Kim, and Fiume's experiments showed that switching from a modify-and-test approach to simulated annealing can lead to improved results. Also, the design of the articulated model was found to be important for ensuring visually-convincing motions are generated. The resulting controllers are susceptible to period-doubling—the motion alternates between two paths that satisfy the key state poses—and even chaotic behavior.

2.3.2 Spline-Based Keyframing

Lasseter advocates that computer animators to embrace the principles of traditional, hand-drawn animation [34]. Computer animation has brought forth a wealth of new techniques, most notably a transition from two-dimensional characters to full three-dimensional models. However, the automation and precision afforded by a computer by no means guarantees appealing results. By adhering to traditional animation concepts, computer-produced animations are more likely to succeed.

Many of these animation principles have been enumerated in the first chapter of the thesis. Axioms in addition to pose-to-pose, reuse of actions, arcs, timing, and exaggeration include:

- **Anticipation**: Anticipation involves attracting a viewer's attention, preparing him for an upcoming action.
- **Staging**: Staging encompasses the idea of planning and presenting an idea with clarity.
- **Slow-In, Slow-Out**: Motions near important keyframes should proceed slower than the less critical transitions in-between.
- **Squash and Stretch**: Objects exhibit elastic geometric deformations during motion, particularly at times of rapid acceleration.

Historically, animations have been constructed using a pose-to-pose approach. Using this technique, the motions are first defined in terms of extreme poses, termed keyframes, with the intermediate poses added later. Thus, keyframing is the foundation for implementing all other animation principles.

Not surprisingly, some of the earliest computer-based animation systems also used keyframing as the basis for creating animated scenes. Burtnyk and Wein created a system that let animators sketch and store skeletal line drawings in three dimensions [14]. These cel drawings are composited to form the keyframes of an animation sequence. In-between frames are automatically computed by interpolation.

By manipulating input devices such as mice, light pens, and thumbwheels, original drawings are created using a freehand pencil tool and simple geometric shapes. In addition, the geometry can be deformed, either by pushing shapes into the edges of the drawing or by applying distortion functions. Completed drawings are archived for later use.

Keyframes of an animation sequence are created by accessing and compositing stored drawings. Affine transformations may be applied to the sketches as needed to position them in the frame. In-between frames are computed by linear or quadratic interpolation of associated edges in the keyframe drawings. In the case where the number of drawing strokes between successive keyframes does not match, strokes are split or combined.

Quadratic interpolation does not provide sufficient smoothness or control for visually high-quality motion. Cubic interpolating splines have become the standard in modern animation software. Kochanek and Bartels created an intuitive cubic interpolating spline useful in keyframe animation systems [31]. Their CTB spline has control parameters that affect:

- **tension**—how acutely the curve bends around key points,
- **continuity**—how continuous the curve is as it passes through key points, and
- **bias**—the direction the curve takes through the key points.

The CTB spline is defined by four coefficients: the starting position, ending position, starting tangent, and ending tangent. The values used for the starting and ending tangent are governed by the tension, continuity, and bias parameters. Four cubic polynomial functions are used to form the spline, each having an initial and final value and slope of zero with one exception apiece, where it is one (see Figure 2.17). These are the Hermite interpolation basis functions.

Tension scales the influence of the overall tangent vector passing through a key point. A long tangent line makes the curve appear slack; a short tangent line makes the curve wrap tightly around the point. Continuity scales the incoming and outgoing tangent contributions at a key point. This creates discontinuities useful for instantaneous changes in the direction and speed of motion. Bias influences the direction of the curve through a key point. In essence, it allows the curve to undershoot or overshoot the point as desired.

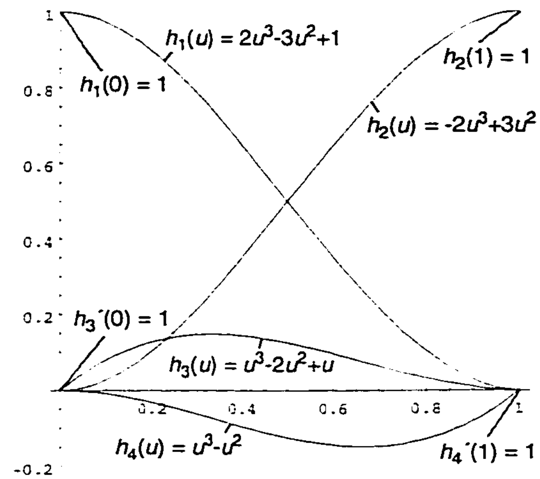


Figure 2.17: **The Hermite interpolation basis functions. Note that each function has a single non-zero value or slope located at one of the endpoints of the interval [0, 1]. This feature is exclusive to the particular function.**

One of the major problems with splines is a lack of specific control over the velocity of the parametric curve. To amend this shortcoming, Bartels and Hardtke add a control spline called a speed-profile curve [7]. The spacing of values sampled from the original trajectory spline are controlled by this speed-profile curve. When the speed-profile curve is at a minima, the density of sampling of the motion spline is high, and thus velocity along the trajectory is reduced. Oppositely, maxima in the curve translate to sparse sampling and an increased trajectory velocity. The speed-profile curve is not meant to give absolute control of velocity, but rather a relative notion of how much faster or slower one part of the motion should be in comparison with another.

The speed-profile is represented as a B-spline. Via a differential equation, the trajectory arc length relates this curve to the change in sampling rate of the motion spline. Solving this equation yields a function that maps the parametric input of the speed-profile spline to the associated parametric input of the motion spline. The authors use a Runge-Kutta integration process to converge on a numerical solution.

Normally in keyframe animation, the animator not only specifies the poses in the sequence, but also the timings. Inexperienced animators find the task of determining the best timings of keyframes (and consequently, the velocity of objects) to be a challenge. Liu and Cohen introduce the concept of keyframe motion optimization with constraints to find these timings automatically [36]. The keyframe optimization problem is specified in terms of constraints such as articulated model poses and high-level kinematic values (*i.e.* velocity or timing restrictions). The solution motion takes the form of Hermite interpolation splines between each pair of keyframes. The unknown timing and velocities required to define the splines are found by the optimization process.

The optimization algorithm consists of an iterative gradient computation and optimization made by searching along the line of the negative gradient. The objective function to be minimized involves energy expenditure. More precisely, the objective function is the square of the work, summed over the duration of the sequence. Each limb contributes kinetic energy at its centre of mass due to linear and rotational motion, and has potential energy due to gravity. The resulting motions are not guaranteed to be realistic because the constraints may be unreasonable.

2.3.3 Forward and Inverse Kinematics

Kinematics deals with motion apart from the considerations of forces and torques involved in dynamics. Application of kinematics to articulated figures falls into two categories [53]:

1. **Forward or Direct Kinematics**: The position of each joint in the figure is individually specified by the animator. The effects of a joint motion are propagated to child segments. For example, bending a knee also changes the position and orientation of the foot (with respect to the global frame of reference).

2. **Inverse Kinematics**: Only the distal ends of limbs are positioned. The movement of the other, more proximal segments are automatically computed.

Kinematics is usually combined with spline-based keyframing. The forward or inverse kinematic poses are specified in keyframes by the animator. Interpolation is applied to derive the intermediate poses.

Robust articulated figures potentially have hundreds of degrees of freedom, making them highly redundant structures for performing actions. Manipulating this plethora of variables to achieve visually convincing motion is a difficult process. By reducing the number of joint parameters involved to just the end points, inverse kinematics simplifies the motion specification problem. However, there are still many possible postures that satisfy the end effector goals. The solution will be a local minima found near the preliminary guess of a satisfactory pose. The solution space can be limited by identifying a primary goal restricted by some secondary objective, as Boulic and Mas have documented [10].

A Jacobian matrix is formed from the current state of the model. The matrix contains the translations and rotations (in the form of instantaneous linear and rotational velocities, respectively) of each joint in relation to their associated end effector. Computing the minimum norm solution of this matrix (*i.e.* the pseudo-inverse) and multiplying this by the desired change in end effector positions yields the angle changes for all of the joints. A secondary task can be inserted into the equation as an additional term. Alas, the solution is only valid for subtle movements. This is because the equation involves instantaneous, first-order variations that are inaccurate predictors for large scale motions.

Additional concerns that must be addressed are joint limits and singularities. Most joints have bounds to their range of rotation. These limits may be realized by:

- removing the joint from the equation when it reaches its limit,
- truncating the joint's final value so that it lies within bounds, or
- defining a secondary task that avoids the limits from being exceeded.

Singularities occur when a segment aligns in such a way that there is a loss of freedom in the movement of the effector.

Some uses for secondary goals include:

- **Interactive Posture Guiding**: A figure can be posed by using a cost function or attracting an end effector to a user-specified location.
- **Automated Posture Adaptation for Balance**: Balance can be achieved by applying a cost function designed to minimize downward torques.
- **Motion Deformation**: Deformation is accomplished by tracking a reference motion as a secondary task whilst ensuring Cartesian restrictions are met via the primary goal.

2.3.3.1 Kinematics of Legged Locomotion

Because of its conceptual simplicity—only the location and orientation of the figure and its joints are manipulated—kinematics is a good approach for animators. In addition, kinematics works in conjunction with the familiar spline-based keyframing for specifying motion. Dynamics requires a knowledge of physics and mathematics, a rare background for an artist. Inverse kinematics, in particular, is ideally suited to posing articulated legs for animated locomotion.

Girard outlines an approach for designing legged animation in the context of the PODA computer animation system [21]. Limb movements are created using a combination of inverse and direct kinematics with speed control. Locomotion is realized by associating these limb motions with gaits. Gaits are defined by user-specified phase timings; transitions are realized by automated means. Based on the active gait's leg movements and Newtonian equations of motion, the position and orientation of the body along a user-defined path is computed.

When creating a leg motion, the endpoints of the limb are fixed and inverse kinematics invoked to automatically pose the proximal joints. Because inverse kinematics rarely yields the desired pose, the animator then modifies the joint orientations. At the same time, the end effector remains locked in position. The speed of the leg motion is controlled by manipulating a time versus distance graph. This graph reparameterizes the original motion spline on the basis of arc length.

Locomotory gaits are specified by the phase timings of the legs and the speed of the animal's body. Gait transitions are achieved by gradually shifting the legs' flight phases to align with those of the concluding gait. However, there are a number of possible permutations of leg phase shifts. The chosen permutation minimizes the total magnitude of these shifts.

The locomotion path and speed the creature is given by a user-specified spline. PODA attempts to have the animal follow this path as precisely as possible. However, the number of supporting legs and their capacity for acceleration limit changes in velocity. The vertical motion of the body is based on the support profile of the legs as defined by the gait. Turning and banking are found automatically by employing Newton's equations of motion. Banking must balance the centrifugal force of turning against the gravitational force downward.

Bruderlin and Calvert continued their research into human walking, developing an entirely kinematical system entitled GAITOR [12]. Their system yields improved performance by allowing the parameters and attributes to be manipulated in real time; the sacrifice in visual quality is negligible.

Three sets of constraints are computed: the length of the leg phases, the angles of the legs after each step, and four key points characterizing the hip movement during each step. The first two criteria are computed as in their previous work [13]. Observation-derived equations for the horizontal and vertical hip maxima are calculated from the phase timings and some other attributes. CTB splines interpolate the hip points. The stance leg is fitted in-between these hip positions and the stationary foot. A pelvis is added—its rotation, list, and displacement are given by user-defined gait attributes. The opposite hip for the swinging leg is appended to the pelvis by interpolation. The swing leg, which may be in one of three phases, is positioned using interpolation.

2.3.4 Performance Animation and Motion Capture

Maiocchi defines motion capture as "the measurement and recording of the direct actions of an actor for immediate or delayed analysis and playback" [38]. Performance animation describes the manipulation of virtual models using the captured data. There are three classifications of motion capture equipment, each having strengths and weaknesses:

1. **Mechanical**: Mechanical acquisition involves the manipulation of joysticks or potentiometers. Such systems are low cost, but the mechanisms are cumbersome, limiting freedom of movement.

2. **Optical**: Optical systems use an array of cameras to record reflective markers strategically placed on an actor; a computer later deciphers the three-dimensional movements from the videos. This affords good freedom of movement. However, camera equipment is expensive, the light reflectors tend to become obstructed, and post-processing is required to extract the animation data.

3. **Magnetic**: Magnetic systems use sensors, the position of which can be measured with respect to a central emitter. The actor must remain in close proximity to the transmitter and metal objects in the vicinity can affect results. Also, the actor must be outfitted with an obtrusive harness and cables, restricting movement. However, the acquired animation data can be accessed instantaneously. Polhemus has extended the range of their magnetic systems to a space of fifty by twenty-five feet with support for up to thirty-two sensors [11].

Da Silva *et al.* propose a system for extracting angles and processing motion capture data for use with articulated figures [16]. Motion capture data frequently documents only the positions of points on a real actor. In this form, the data is not easily transferred to a digital figure nor convenient to edit. The system converts this data to a neutral form: orientation data encoded as Euler angles. This massaged motion data can then be processed by filtering, concatenation, and blending. The author's framework provides a graphical user interface for visualizing these processes.

Positional motion data is organized according to the graph of the articulated figure. Points are considered in adjacent triplets. The first and second points form a line segment, as do the second and third. The angle, expressed clockwise, between these two line segments is the relative angle of the joint at the middle point.

Motion operations are unary (*e.g.* filtering), binary (*e.g.* concatenation), or n-ary (*e.g.* blending). Filtering can be used to extract frequency bands, useful for extracting emotional qualities from movements [51]. Concatenation is realized by interpolating the end of the first motion into beginning of the second. Blending is achieved by identifying synchronization points between two or more motions, reparameterizing, and then mixing.

2.3.5 Procedural Motion with Scripting

Specification of motion by procedural means involves writing scripts of movements, using function-like procedures as the building blocks. Procedural motion is frequently used to describe any form of parameter-controlled locomotion (including dynamic simulation and kinematics). We specialize the definition to purely functional and language-based motion specification.

Perlin describes a procedural language-based animation system that uses a noise function to add subtle, random details to repetitive actions [40]. Scripting simplifies the generation of prototypes of new hierarchical models and actions. The model specification language uses a postfix notation featuring procedures that draw the limbs, identify the joints, and perform affine transformations. An action, for instance a rumba dance, is defined in terms of each joint's motion and rotational limits.

A joint movement is expressed in terms of rotation and noise functions. The orientation functions change over time according to combinations of four normalized trigonometric functions of varying frequencies and phases. This formulation is ideal for creating periodic motions such as walking. The noise function is defined as follows:

- Zero for integer values.
- For non-zero inputs, it takes the value of a Hermite spline interpolating two random gradients defined at the surrounding integers.

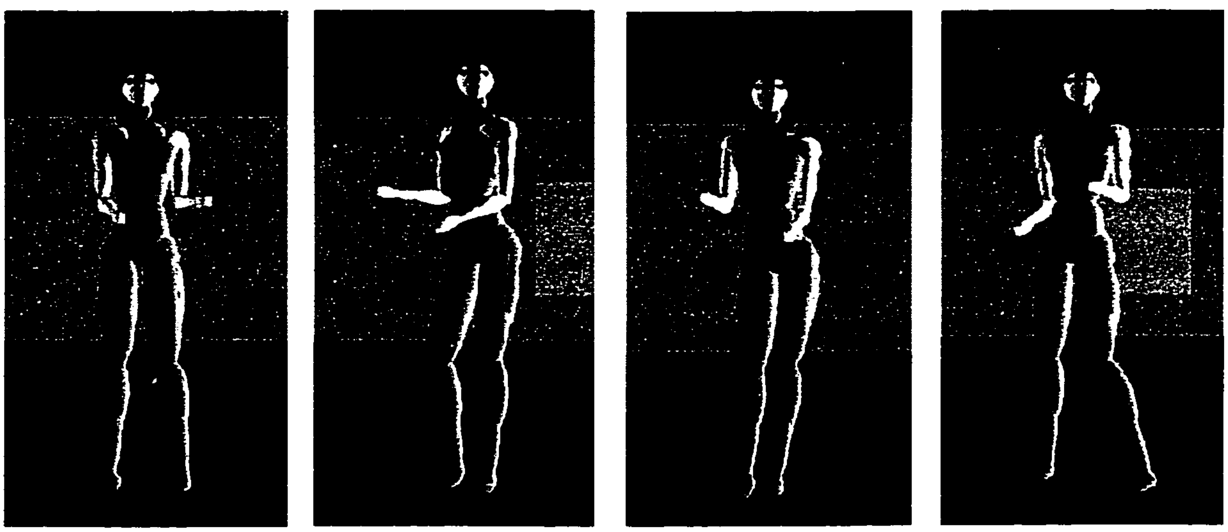


Figure 2.18: **A few frames from a rumba dancing action, programmed using Perlin's procedural motion with scripting system. (Images taken from "Real Time Responsive Animation with Personality" [40].)**

Actions may be combined by using a weighted sum of each joint angle from each contributing action. Transitions from one motion to another are achieved by cross-dissolving. An S-shaped ramp is used to blend from one motion into the next over a user-defined duration.

Some clips from a rumba dancing action, programmed using Perlin's framework, are shown in Figure 2.18.

2.4 Editing Existing Animated Motions

By using any of the previously specified methods, original motions can be created. The next task is to assemble the actions into animated sequences. The problem of editing these motions may be divided into two tasks: the modification of individual movements and the chaining of these motions into a series.

There are a number of motivations for altering motions. Subtle variations in the timing and scale of a gesture convey emotional qualities. For instance, a person who walks slowly with his head tilted forward might be considered depressed or tired. A fast walking person with his head held high appears to be proud or confident. Frequently, there are some constraints to be satisfied during an action. An object might need to be in a particular position at a precise point in time. For example, a bat must connect with a pitched baseball as the ball passes over the home plate. In these sample situations, one fixed version of a motion will not suffice.

Having prepared a set of customized movements, the second task is to concatenate these motions into an animated sequence. Of major concern are discontinuities formed by adjoining motion curves. The visual effect is an unnatural jump in the action. Thus, when chaining two movements together, it is desirable to ensure a degree of smoothness. In traditional animation, artists address this concern by adhering to the principle that objects move in arc-like motions.

Many of the aforementioned motion specification methods do facilitate some customizing. For instance, the speed and direction of a dynamic simulation of running may be adjusted [29]. However, customizing movements may be difficult to realize in cases where support is not specifically built into the system. The dynamic systems for biped walks previously described operate only on flat surfaces. The animator could not make a character climb up a flight of steps using only the high-level input parameters. To support stairs, the controllers of the system require modification—a difficult undertaking. All this assumes that the original system and input parameters are accessible. If only the resulting raw motion data is available, then there is a need for alternative editing techniques.

In the upcoming sections, several computer-based means of editing animated motions will be covered. A novel way to compose sequences is to let the computer decide, in whole or in part, what actions an animated character is to perform. This is the goal of semi-autonomous behavioural systems. Another method involves isolating emotional characteristics of motions. Emotional qualities embedded in the timings, exaggerations, and frequency information of a motion may be extracted and applied to another. A third editing technique involves influencing a motion, either by applying a set of user-specified functions or by combining it with a collection of similar movements. These processes are termed motion warping and motion synthesis, respectively.

2.4.1 Autonomous Behaviour

Autonomous behaviour refers to the capacity for an animated creature to choose its own actions. The virtual creature makes decisions on what movements to carry out based on its internal goals. Take, for example, an autonomous dog character. At some particular instance in time, the animal may be feeling thirsty or playful. In the dog's view is a puddle of water and a rubber ball. Depending on its priorities, the creature may decide to get a drink of water or go play with the ball. The dog makes a decision that satisfying his thirst is more important. The animal will then apply known locomotory motions to move to the puddle, and then proceed to swallow the water using its built-in drinking action. This means of creating animation sequences is ideally suited to interactive virtual environments.

Blumberg and Galyean created a robust system with multiple tiers of control for autonomous animals operating within a virtual habitat [9]. The system features:

- a mechanism to allow an animator to "direct" a character at several levels,
- a behavioural model so that the characters can sense their surroundings using synthetic vision and choose a reasonable course of action, and
- an extensible architecture.

Behaviours instruct the motor system to perform mechanical skills, which in turn drives the geometry. The animator can provide motivational and task-oriented instructions to the behaviour layer or issue direct instructions to the motor system of the character.

The motor system has, at its top level, a controller to translate behaviours into motor skills. The motor skills are basic motions that control the degrees of freedom of the joints. Semaphores prevent conflicting skills from using the same degrees of freedom concurrently. An inverse kinematics-based animation system is used to pose the creatures.

Behaviours compete for control of the motor system. The choice is dependent on the character's internal state—its needs and wants—and the environmental conditions. Inhibition blocks other behaviours to reduce flip-flopping. A decreasing level of interest aids in preventing impossible goals from being attempted repetitively.

2.4.2 Bestowing Emotional Characteristics

A powerful way to make animation appealing is to endow the characters with emotion-like qualities. Subtle changes in a neutral motion in terms of timing and position may be construed as an emotional response. Researchers have sought automated means to extract and apply emotion. However, there is no consensus as to which features of motion are emotional in nature. These traits may vary by culture amongst humans and by species amongst animals.

Unuma, Anjyo, and Takeuchi suggest that by representing joint rotations in the form of a Fourier series, emotional details can be extracted and applied to other motion curves [51]. The authors believe that the frequency information of a motion function encodes its emotion-like traits. Thus, a wide variety of motions can be derived by substituting the emotional essences from a small library of core movements. Interpolation and extrapolation of the Fourier terms results in mixing, overstating, or underemphasizing the characteristics of two motions. Superposing applies emotional qualities extracted from one movement onto another action. This process of instilling emotion-like characteristics involves less intervention on the part of the animator than does customizing the movements by dynamic or kinematic means.

Motions to be combined must first be synchronized such that their equivalent keyframes occur at the same time. To interpolate between motions, each of the two actions is assigned a weight between zero and one such that the sum of these weights equals one. Their Fourier terms are combined using these weights to yield the blended motion. The resulting motion inherits emotional characteristics from the two source movements, albeit at a reduced level of significance. Extrapolation uses weights outside the range of zero to one. The emotion from the higher-weighted motion is exaggerated whilst the emotion from the lower-weighted movement is understated. Transitions from one movement into the next is achieved by cross-dissolving—gradually shifting the weights from zero to one over time.

The difference in the Fourier coefficients of two motions can be used to construct a characteristic equation that reflects the emotion-like qualities of one of the actions. For instance, by taking the difference between a fatigued run and normal run, a "tired" characteristic is extracted. This quality can then be superimposed with, say, a walking action to create the appearance of a tired walk. To create further diversity, blue noise can be applied to the Fourier series, making the character appear to shiver.

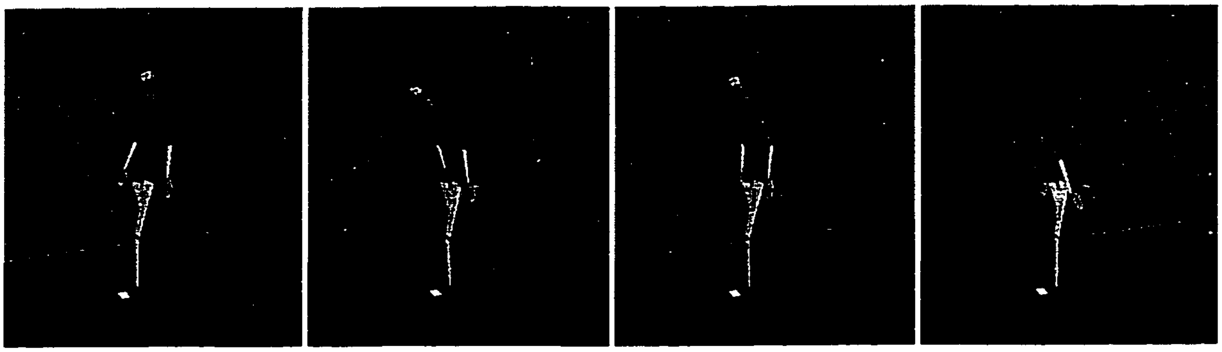


***Figure 2.19:* Fourier-based emotion interpolation and extrapolation of tired walks. (Frames taken from Unuma, Anjyo, and Takeuchi's "Fourier Principles for Emotion-based Human Figure Animation" [51].)**

Finally, the authors add some control specific to customizing locomotions. The step length is implemented as a scaling of a motion's Fourier series and speed as a scaling of the independent variable time. Other controls include the figure's bounce and the type of gait.

Some examples of interpolation and extrapolation of a normal and tired walk appear in Figure 2.19. The first image depicts a normal walk, the second a tired walk. The third image is a result of an interpolation of the normal and tired strides, resulting in a somewhat tired walk. The final frame shows extrapolation with an emphasis placed on the tired walk, resulting in a heavily-fatigued gait.

Amaya, Bruderlin, and Calvert's found that frequency information had little relation to the emotional content of motions. Instead they argue, based on empirical examinations of motion capture data, that the timing and spatial amplitude better describe the feeling of an action [5]. Their technique for extracting emotion-like qualities employs signal processing to derive emotional transforms. These transforms encompass the difference between neutral and emotion-laden actions. The data can then be applied to other, neutral actions.

To compute an emotional transform requires two motions, a neutral action and an equivalent, emotion-influenced action. The user must first identify the phases in the movements, typically bounded by characteristic poses (*i.e.* keyframes). A temporal description of the motion is found for each of these phases and for each joint. This is defined as the distribution of frames with respect to the normalized distance covered along the object's trajectory. The temporal emotional characteristics are grafted into a new movement by substituting in this distribution.

Figure 2.20: **Instilling the emotional characteristics of sadness and anger into a kicking action. (Frames taken from Amaya, Bruderlin, and Calvert's "Emotion from Motion" [5].) The emotional transforms were originally extracted from the act of lifting and drinking from a glass.**

To isolate the spatial amplitude of an action, the joints of the articulated figure are classified according to their proximity to the torso. A distance value is calculated for each of these categories and for each phase. A line is drawn between the endpoints of the phases. The maximum distance between this line and the motion trajectory is used for the amplitude metric. The ratio between the emotional and neutral source motions is used to scale the amplitude of a second, neutral movement.

Figure 2.20 shows some frames of a person kicking his foot. The first sequence represents the starting neutral motion. In the second sequence, this action has been modified by a sad emotional transform. The third series shows the motion transformed into an angry kick. The source movements for these emotional transforms was a person lifting and drinking from a glass.

2.4.3 Motion Warping and Synthesis

Image warping was developed as a method to metamorphose, in a flowing manner, one two-dimensional image into another according to shared features. Morphing is the term coined for the animated version of this process. Historically, filmmakers have employed stop-motion photography to achieve similar effects [8]. In computer graphics, superior results can be attained by interpolating between corresponding pairs of vertices of two three-dimensional models. Early attempts to metamorphose two-dimensional pictures involved either simple cross-dissolving or mapping pixels using a particle system. Morphing was later extended to the deformation of the motion curves defining three-dimensional animations [57].

2.4.3.1 Image Warping and Morphing

Image warping refers to the controlled deformation of images, yielding results akin to the reflection from a bent mirror. The distortion is achieved by altering a mesh, line segments, or points aligned to important features in the source pictures and mapping the pixels accordingly. Morphing animations can be created by progressively cross-dissolving corresponding pixels from the original images.

Beier and Neely propose metamorphosing images by warping equivalent attributes, specified by line segments, from one keyframe to the next and applying cross-dissolving [8]. The transformation process is as follows:

1. The animator indicates related features in the starting and ending images via directed line segments. To metamorphose between two moving objects, line drawings are created in multiple keyframes. The line segments for intermediate frames are found by interpolating between the keyframe drawings.

2. Each pixel of a transition image is set by finding its related pixel in the two source images and applying cross-dissolving. First, the in-between frame point's position and orientation with respect to each of the defined line segments is determined. The associated points in a keyframe image will have same relative position and orientation to these lines. The displacement between these pairs of points is weighted by the transitional frame point's proximity to the line segment. After considering the intermediate frame's point against all the line segments and averaging the weighted displacements, the result is the ideal keyframe image pixel to use in the mapping. This process, known as Shepard's interpolation, is applied on both original images. Cross-dissolving is applied to get the final pixel value for the transitional image.

Figure 2.21: **Morphing between two portraits using Beier and Neely's feature-based image warping technique. (Images taken from "Feature-Based Image Metamorphosis" [8].)**

Because line segments are in effect globally and interact with each other, there are sometimes undesirable side-effects. There is also a processing penalty to this algorithm as the number of line segments increase.

The three images in Figure 2.21 show the results of morphing by this method. The left and right images are the originals and the middle image a mixture of these two. Directed line segments show the correspondence between features.

A more robust metamorphosis method is suggested by Litwinowicz and Williams [37]. The user identifies common features using drawings composed of point, line, and curve primitives. Thin-plate splines—useful for the interpolation of scattered (*i.e.* nonuniformly-spaced) data points—interpolate the displacements, horizontal and vertical, between two drawings. Unlike Shepard's interpolation, thin-plate splines do not suffer from the cusps. Another positive aspect is that the drawing primitives have a global effect with the strongest influence being on the primitives nearest the point of interest. Also, the complexity of the morph is not dependent upon the number of attributes.

The solution spline is formed on a discrete grid. Discontinuities can be specified by a separate map, in essence breaking the grid into isolated, independent areas. New features (*i.e.* drawing points) may be inserted without affecting the current mapping. Because successive frames of a morphing animation are usually only subtly different, the previous spline can be used as an initial estimate for the spline of the subsequent frame, reducing computational costs.

2.4.3.2 Motion Warping

Witkin and Popović extend the concept of warping to the motion curves used in animation [57]. An animator specifies the modified poses in sparsely-spaced keyframes. Using spline-based interpolation, these changes are then applied over the entire motion curve. There are numerous benefits to this form of motion editing:

- The overall quality, including crucial fine details in the original movement, are retained.
- Motions similar to a single prototype can be specified with relative ease.
- Frequently, it is more efficient to apply warping than to modify the individual control points of a motion curve. This is especially true of motion capture data where keyframes are spaced closely.

There are also some weaknesses to motion warping:

- Specification of warps is a low-level activity. Further research to automate the process is required.
- There are no means to enforce geometric restrictions beyond adding extra keyframes.

The motion curve for each model parameter is considered separately. First, the animator specifies how time frames of the original motion are to be extended or diminished in the derived movement. This takes the form of a set of keyframed time mappings from warped to normal time. The points are interpolated by Cardinal splines.

The value of the deformed motion curve is defined in terms of the original multiplied by a scaling function and transposed by an offset function. These two functions take normal time as their single input variable. The animator specifies the warped poses by keyframing, and chooses the extent of the scaling and shifting influences on the source movement. The scaling and translating functions are formed from splines interpolating the keyframed spatial warping points.

Transitions between two different actions are achieved by motion blending using Perlin's approach [40]. The two movements are warped into alignment such that their constituent phases overlap. Cross-dissolving the motion curves yields the complete transitory action.

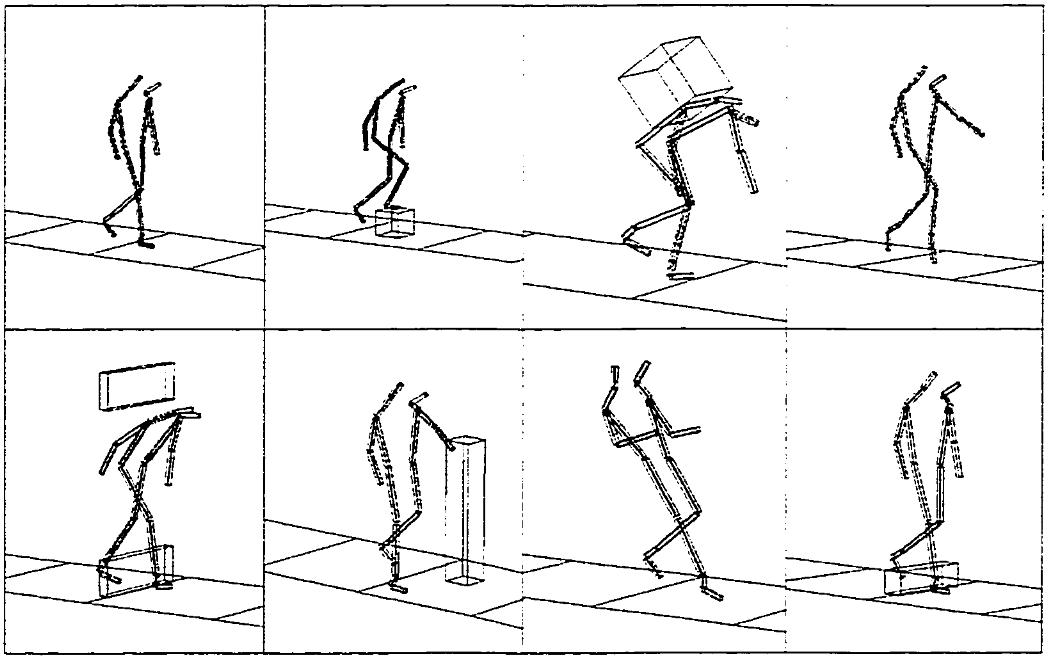


Figure 2.22: **Customizing an existing walking action using Witkin and Popović's motion warping. (Diagrams taken from "Motion Warping" [57].) The figure is made to bend, duck, or step over obstacles.**

Several variations of a walking action created by motion warping are illustrated in Figure 2.22. Motion warping was used to make the figure duck under, step over, or move around obstacles. Each clip used from one to five motion warping keyframes.

2.4.3.3 Motion Synthesis

One of the main weaknesses of motion warping is that specification of deformations is low-level and tedious. By integrating geometric constraints into the technique, the task of customizing motions can be partially automated. These restrictions may be acquired from the environment, input directly by the user, or be a resultant of the original motion specification process. The process of automatically adapting movements is termed motion synthesis.

Wiley and Hahn generate precise movements for a figure in real-time from a set of similar sample motion [54]. Starting from a set of like actions, say a hand reaching for a point, the motions are resampled onto a uniform three-dimensional grid using trilinear interpolation. This step, in essence, sorts the motions. Finding the closest candidate movements, used to generate a specific new motion, is now trivial. To have the hand reach for a specific location, a triple interpolation of the movements associated with the eight closest points in the grid is performed. Because the eligible movements usually have differing timings, they are first temporally rescaled into alignment. The timing of the final synthesized motion is found by interpolating the timings of the contributing movements. The authors demonstrate their technique with a pointing action that uses linear interpolation and a cycling action that involves rotational interpolation.

This motion synthesis system has the advantage of being fast and very stable. It is a simple way of achieving an example-constrained form of inverse kinematics. There is a trade-off between storage space versus density of motion data. An increase in the number of sample movements improves the quality of the synthesized motions at the cost of increased memory requirements.

Lamouret and van de Panne propose a more elaborate form of motion synthesis that catalogues the example movements by the state of the character, the environment, and the style of motion [33]. A function takes the state, environment, and style and accesses a motion database, extracting the closest matches. The retrieved actions are then tailored to precisely fit the circumstance of the animated character. This process of adaptation involves translating the end locations of the motion to the required positions.

The database of actions contains sets of motion primitives. One such primitive might be a single cycle of a periodic action. To prevent the database from becoming unnecessarily large, only non-redundant actions may be added to the collection. The best-fit criteria used by the extraction function minimizes the sum of the discrepancies in state, environment, and style. The state is described by the position of the articulated model's joints weighted by their mass. The environment is represented by the location of the figure's end effectors.

Having selected the most appropriate sample motion from the database, the movement must be fine-tuned. The adaption process first translates the starting location of the motion into the correct position. The required end position of the motion is achieved by linearly introducing the necessary translation over the course of the movement. This approach is simplistic and would benefit from an improved weighting scheme and inverse kinematics.

As an example, the authors created a database of movements for a leaping lamp. The motions were originally created by dynamic simulation. The lamp was made to hop over variable terrain by synthesizing new motions from the database.

Gleicher designed a method for adapting the motions of an articulated model to another figure with differing geometric proportions [22]. One of the major problems of simply rescaling a motion is that spatial constraints are broken. For example, when the action of a tall character walking along the ground is transferred directly to a shorter character, the character's feet no longer touch the surface and the stride length appears exaggerated. Gleicher retargets a motion by letting the animator reinforce these constraints.

A symbolic space-time solver, à la Witkin and Kass [56], adapts the motion globally. The solver attempts to minimize the degree of alterations made to the movement and restrict the introduction of undesirable frequency content. For the sake of simplicity, the number of constraints is kept to a minimum, although this policy can lead to a sacrifice of quality. For instance, there is no consideration of gravitational effects or balance. The author notes that an inverse kinematics solver is insufficient for motion adaptation; it finds local solutions and thus cannot suppress objectionable high-frequency artifacts.

The constraints restrict the possible posture of the character, prevent impossible interactions with the environment, or enforce specific spatiotemporal goals to be attained. Taking these restrictions and the original motion as input, the solver finds the necessary changes to the movement that satisfy the given constraints. As an initial guess at a solution, the input motion is scaled and translated according to some of the positional constraints. An objective function minimizes the square of the magnitude of these alterations. To prevent high frequency components, the changes are represented by a smooth B-spline having well-spaced control points.

A couple of frames of two characters performing a swing dance are shown in Figure 2.23. The size of one of the dancers is reduced, necessitating a retargeting of the dance motion for one or both of the characters.

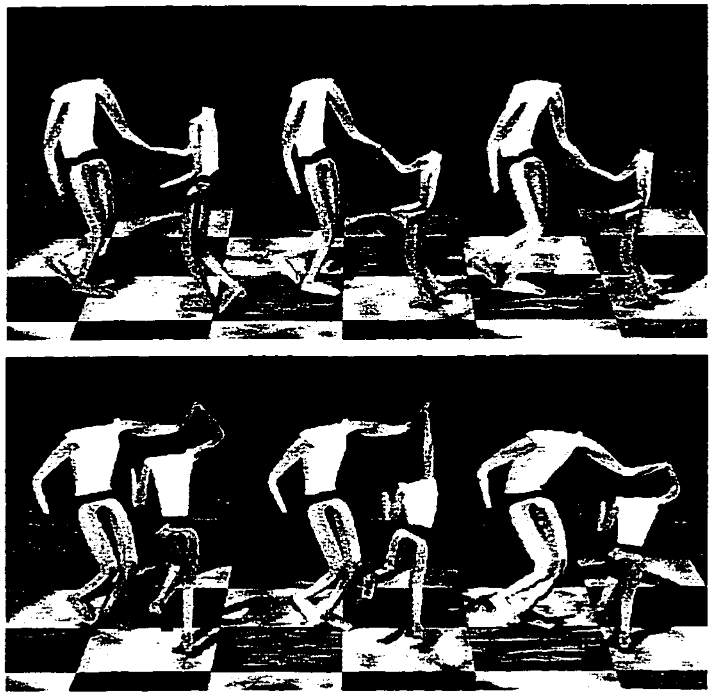


Figure 2.23: **Retargeting a swing dance for a smaller dance partner. (Clips taken from Gleicher's "Retargetting Motion to New Characters" [22].) The left pairs show the original action. The middle frames show the motion adapted for the shrunken dancer. The action of both characters has been adjusted in the right frames.**

2.5 Epitome

This thesis examines Witkin and Popović's motion warping technique in depth. In the next chapter, the motion warping equations and their usage are discussed. This forms the basis of a system for modifying multi-legged gaits. The system automates the specification of the warping equations given input regarding the gait phases, model attributes, and the external environment.

Chapter 3

Architecture of the Quadruped Motion Warping System

After careful appraisal of the published research, it was decided that motion warping would form a good basis for a quadruped animation editing system. Motion deformation has several desirable characteristics that benefit the animation of animal locomotions. A predefined motion, in this case a gait, can be modified based on the prevailing environment. Our system automatically adjusts, via motion warping, the gait to fit a three-dimensional, user-defined path. Despite these adaptations, the derived locomotion retains the visual qualities of the original locomotion. Motion warping with blending also aids in producing transitions from one gait to another.

In this chapter, the design of the quadruped motion warping system is covered in detail. The structure of the geometric animal model is one important consideration. The model must be flexible enough to support a range of motions, and its shape and surface should be visually convincing.

Another topic is the acquisition of the motions to be warped. Although not the emphasis of this thesis, a few suggestions are made regarding the creation of quadrupedal actions. Techniques for defining animal motions are limited; for instance, performance animation is rarely feasible. Instead, transcribing movements from photographic studies or adapting existing motions from other animals must be undertaken.

Given a basic set of motion curves, warps can be specified and applied. These take the form of user-defined curves that influence the temporal and spatial characteristics of the original movement. Witkin and Popović's motion warping formulas are enumerated herein [57]. In addition, the equations for blending two aligned motions are discussed.

These general motion warping concepts are then tailored to altering quadrupedal locomotions. Automation is feasible because of the limited domain of interest, namely four-legged gaits. A gait is characterized by the timings of the phases of each of the legs. In the animation system, temporal warping is used to adjust these phases based on changes in speed. Given a user-defined path in three dimensions, spatial warps are automatically generated to make the animal move along the route. This involves positioning and revolving the body to run parallel to the path. During its stance stage, a foot must remain stationary upon the ground, no matter what the terrain. The ankles and hips are rotated to help ensure this constraint.

To achieve a transition, the leg phase timings are gradually shifted and resized from those of the starting into those of the ending gait using a temporal warp. At the same time, progressive blending of the positions of the legs in the two strides is applied.

3.1 The Geometric Model

The investigation of biomechanics in the previous chapter has revealed two important points relevant to the design of a graphical animal model:

1. Legs operate in vertical planes parallel to the length of the torso (Figure 2.2).
2. A collapsible table with hinged legs is analogous to the skeletal structure of an animal (Figure 2.3).

These criteria justify the adoption of a hierarchical model constructed from rigid segments connected by rotatable joints.

There are a number of possible representations the model structure and its joint states. First, the hierarchical relationship between model segments is considered. Each model section is connected to one or more segments, one proximal and the rest distal with respect to the torso. One method for maintaining these associations is for each segment to store a reference to its single adjacent segment located closer to the body. Another option, albeit requiring a more complicated structure, is for the proximal segments to remember each of their connected distal components. Our system uses the former approach for simplicity and compatibility with LightWave 3D™. Figure 3.1 shows the tree-like structure relating the parts of a typical animal.

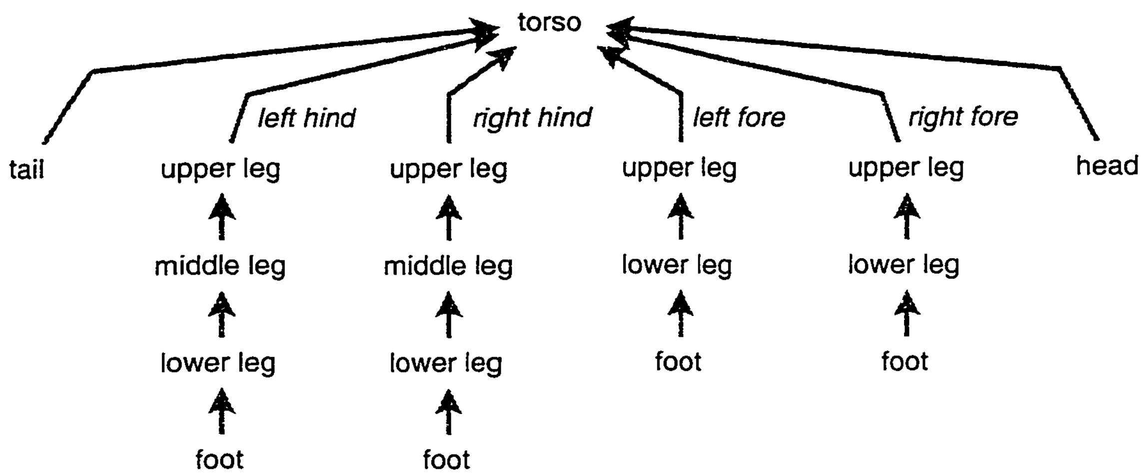


Figure 3.1: **A typical hierarchy of model segments for a quadruped. Associations between connected segments are stored inwards from the extremities towards the torso. That is, a given segment maintains a reference to its adjacent, proximal model part. For example, a foot knows it is connected to the lower leg.**

Another consideration is how the articulated joints are bonded to the geometric model. Our system uses multiple rigid geometric segments connected by rotatable junctures. The joint for each component is located at the origin of its local coordinate system. The position of the segment within the model is specified in the coordinate system of its connected proximal part. An offset from the proximal component's origin marks the origin for this segment.

Rotations are specified in terms of Euler angles. A juncture may have from one to three degrees-of-freedom (*i.e.* axes of rotation). A segment is rotated at the origin of its coordinate system; the effects of the revolution are propagated to the distal segments. Figure 3.2 shows the joints of an animal—a marten—and their degrees-of-freedom. Note that the legs have only one axes of rotation, ensuring that their movements are constrained within a vertical plane.

Another way of manipulating the geometric model is via "bones". Bones, as the name suggests, act as a skeleton to which the geometric model is affixed. By moving the bones, the attached "skin" of the model is deformed. This yields a realistic surface motion for applications involving a continuous skin such as animals and humans. A model composed of rigid articulated segments lacks this continuity at the geometric intersections of its parts. On the downside, the region of influence of a bone can be difficult to bound. Articulated joints and bones can be combined to form hybrid models with bones inserted in the individual model segments. This lets the limbs bend and flex with a muscular effect. For reasons of simplicity, our prototype animation system does not use bones.

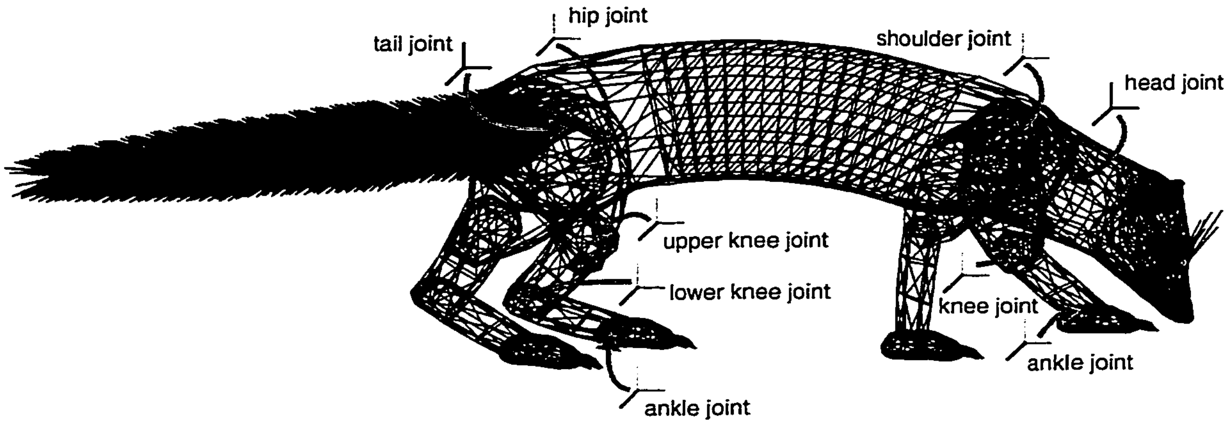


Figure 3.2: **The articulated model of a marten. The animal is composed of rigid segments connected by joints having one, two, or three degrees of freedom. The front legs are made from three segments (including the foot) and the back legs from four. In general, leg joints have only one degree of rotational freedom (about the axis pointing into the page).**

Depending on the animation technique employed, the model components may store additional state information. For instance, dynamic simulation requires information regarding each segment's mass, moment of inertia, etc. Our system requires no further data beyond the model hierarchy and joint locations.

3.2 The Generic Motion Warping Technique

3.2.1 Acquisition of Motions

At a fundamental level, animation systems represent motions using curves. These curves, defined with respect to time, control one parameter of the model apiece. The parameter reflects a single joint angle about an axis or a position in one dimension. Even actions created using sophisticated techniques such as dynamic simulation can be expressed in this form. Thus, there is a wide range of possible methods that can be used to develop a set of quadruped movements.

Although not the main topic of this work, the problem of constructing original motions for animals has some interesting challenges. Each of the major animation specification techniques—dynamic simulation, kinematics, and motion capture—have limitations in this regard.

One of the continuing problems of dynamic simulation is the design of control algorithms. Frequently, significant simplifications of the underlying mechanics are made to ensure that the system is sufficiently fast and manageable. Because animals involve the simulation of not two, but four legs, the system becomes unavoidably more difficult to control. Additional simplifying assumptions could be made in the control mechanism, but this would sacrifice realism. Another concern is the gathering of measurements such as the masses and moments of inertia of the creature's constituent parts. Whilst these statistics are readily obtainable for humans, there may be no existing studies that provide this data for the animal of concern. Finally, even if a sound control system is available for a biologically similar animal, adapting the algorithm for another creature is complicated by scaling issues (see Hodgins [28]).

Motion capture requires placing sensors on a live actor, a requirement that is not possible with many animals. In particular, wild animals or smaller creatures would find such devices intrusive. The attached sensors might interfere with joint freedom, yielding abnormal motions. Instead of carrying out the desired action, the subject would likely spend its time trying to remove the devices. It is not possible to direct an untamed animal, so it may not perform the desired movement. Finally, some actions are not possible within the confined space and carefully controlled environment of many motion capture systems. The most plausible setup is an optical system with the key points on the animal marked with reflective paint.

Keyframe-based kinematics is a very basic, low-level technique that has no particular disadvantages tied to the subject to be animated. Specification of an animal movement, or any other kind of motion, remains a time consuming process of trial-and-error.

For this thesis, a traditional keyframing was employed to create animated gaits for a marten. This was the only feasible choice given the limited resources available. Two sources were used as references. A graph of a cat's walk, taken from Gray's *Animal Locomotion*, was used as the basis of an ambling gait [23]. Video footage of a marten bounding and ambling provided a benchmark for comparisons.

3.2.2 Representation of Motion and Deformation Curves

Splines are a natural complement to keyframing, providing automatic interpolation of in-between values. They also provide continuity through the key points so that the motions appear smooth with no abrupt changes in direction. Animation packages such as LightWave 3D™ use Kochanek and Bartel's Continuity-Tension-Bias (CTB) splines, a form of cubic polynomial spline [31]. CTB splines have the advantage of additional controls for governing the motion curve's shape. Our system makes use of CTB splines, both for representing original motions and for the warping curves that influence these motions.

A good overview of CTB splines and their controls was provided in the previous chapter (*Section 2.3.2—Spline-Based Keyframing*). As they form the core of our motion warping system, details regarding the implementation of the equations warrant mentioning. The following discussion is based on Kochanek and Bartel's paper, "Interpolating Splines with Local Tension, Continuity, and Bias Control" [31].

CTB splines are expressed in terms of the Hermite basis functions.

$$
\begin{aligned}
h_1(u) &= 2u^3 - 3u^2 + 1 \\
h_2(u) &= -2u^3 + 3u^2 \\
h_3(u) &= u^3 - 2u^2 + u \\
h_4(u) &= u^3 - u^2
\end{aligned}
\tag{3.1}
$$

A graph of this system of equations was given in Figure 2.17. A motion curve is made up of a series of spline segments.

A segment is defined as the curve joining two adjacent user-defined key points, P_i and P_{i+1}. These points are used to weight the contribution of the first two Hermite functions, $h_1(u)$ and $h_2(u)$. A tangent vector, D_i, at the key point P_i is defined in terms of the two chords formed with the two surrounding points, P_{i-1} and P_{i+1}. Likewise, the tangent D_{i+1} at P_{i+1} is given in terms of the chords subtending P_i and P_{i+2}.

$$
\begin{aligned}
D_i &= s_1(P_{i+1} - P_i) + s_2(P_i - P_{i-1}) \\
D_{i+1} &= d_1(P_{i+2} - P_{i+1}) + d_2(P_{i+1} - P_i)
\end{aligned}
\tag{3.2}
$$

The coefficients s_1, s_2, d_1, and d_2 scale the contributions of each chord and will be discussed shortly. The tangent vectors are combined with the final two Hermite functions, $h_3(u)$ and $h_4(u)$. The spline segment is then defined as

$$
P(u) = P_i \cdot h_1(u) + P_{i+1} \cdot h_2(u) + D_i \cdot h_3(u) + D_{i+1} \cdot h_4(u) \tag{3.3}
$$

where $0 \le u \le 1$. Evaluating the function $P(u)$ gives an interpolated point a fraction u of the way between the two key points P_i and P_{i+1}. Figure 3.3 graphically illustrates the points and tangent vectors that define a CTB spline segment.

The coefficients in Equation 3.2 are expressed in terms of the user-defined continuity ($-1 \le c_i \le 1$), tension ($-1 \le t_i \le 1$), and bias ($-1 \le b_i \le 1$).

$$
\begin{aligned}
s_1 &= \frac{(1-t_i)(1-c_i)(1+b_i)}{2} & s_2 &= \frac{(1-t_i)(1+c_i)(1-b_i)}{2} \\
d_1 &= \frac{(1-t_{i+1})(1+c_{i+1})(1+b_{i+1})}{2} & d_2 &= \frac{(1-t_{i+1})(1-c_{i+1})(1-b_{i+1})}{2}
\end{aligned}
\tag{3.4}
$$

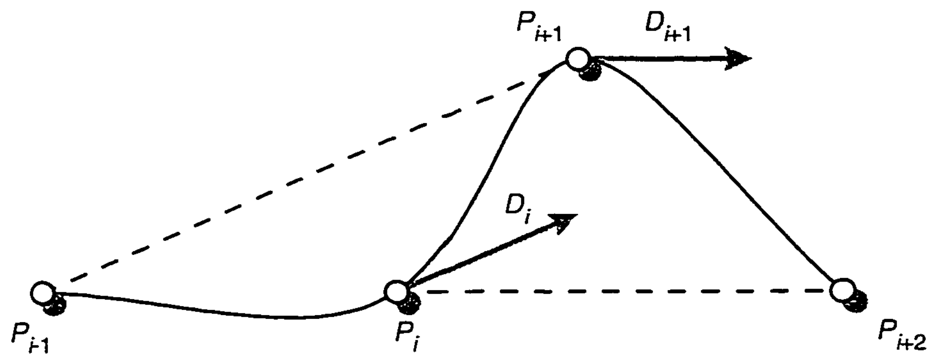


Figure 3.3: **The coefficients of the four Hermite basis functions that make up a CTB spline segment.**

Values along a motion curve need to be accessed by a frame number, not by a normalized parameter u. Because the time intervals between key points varies, so too does the fractional step size corresponding to successive frames in each segment. For this reason, the magnitudes of the tangent vectors, D_i and D_{i+1}, need to be adjusted to avoid discontinuities in speed through the key-frames.

$$\begin{aligned} \text{adjusted } D_i &= D_i \cdot \frac{2 \cdot N_i}{N_{i-1} + N_i} \\ \text{adjusted } D_{i+1} &= D_{i+1} \cdot \frac{2 \cdot N_i}{N_i + N_{i+1}} \end{aligned} \tag{3.5}$$

N_i is the number of frames between P_i and P_{i+1}.

Representation of a spline requires storing a list of points and their associated continuity, tension, and bias. The points are sorted according to their associated frame number. To compute the interpolated point for a given frame number t, the pertinent section of the spline must be found in the list. This segment is identified by finding the P_i (at frame t_i) and P_{i+1} (at frame t_{i+1}) such that $t_i < t < t_{i+1}$. The frame must be expressed in terms of a fractional parametric value u, found using the formula:

$$u = \frac{t - t_i}{t_{i+1} - t_i} \tag{3.6}$$

Now Equation 3.3 may be applied to yield the interpolated point at the desired frame.

The tangent vectors D_i remain constant most of the time; exceptions include when points are added into or removed from the spline, or a point's control parameters are altered. For efficiency, the tangents may be precomputed. Then, Equation 3.3 is reorganized with the coefficients for each degree of the polynomial collected and calculated in advance; these coefficients are expressed in terms of P_i, P_{i+1}, D_i, and D_{i+1}.

There is a concern common to all splines that needs addressing. How should the curve behave near its ends where there are insufficient points to fully specify the segment? If the motion is cyclic, then points on the opposite end of the spline can be reused to fully define the tangent vectors for the segment. For nonperiodic motions, the best estimate for the tangent vector is made based on the last pair of points. Finally, if there are insufficient points to define even a single segment, then a constant or linear interpolation function may be substituted.

3.2.3 The Motion Warping Equations

3.2.3.1 Temporal Warping

Motion curves in animation are defined with respect to time in units called frames. Let one such motion curve be $\theta(t)$ with t, the time, running over the interval of the animation sequence. The movement can be modified both temporally and spatially. Parts of the sequence can be speeded up, slowed down, or even reversed by warping the timings of the motion. For example, to transform a speedy walk into a leisurely stroll requires the duration of each step to be extended. The time deformation is specified as a mapping from warped time back to the original time of the source motion.

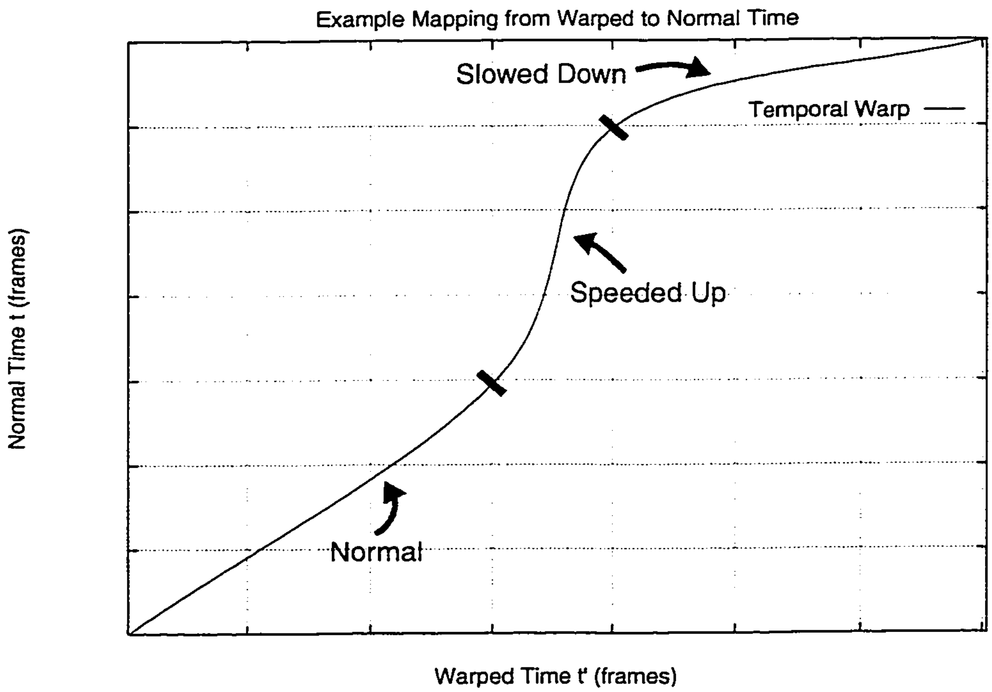


Figure 3.4: **An example of a time warp demonstrating speeding up and slowing down effects. Time is mapped from warped time back into normal time. Steep slopes (greater than one) indicate a speed up whilst shallow slopes (less than one) denote a slow down.**

$$t = f(t') \tag{3.7}$$

Figure 3.4 shows a typical graph of a time warp mapping function. A traditional keyframing approach to specifying the time warp is adopted. The function is represented as a CTB spline passing through i user-specified temporal points (t_i', t_i).

Expressing the temporal warp in terms of t rather than t' might be a more intuitive formulation. Since the animator is familiar with the timings of the original motion, the association should be from t to t'. However, since the motion curves are indexed in terms of t, it is more convenient, from an implementation perspective, to map from warped time back to normal time. It is unlikely that this less intuitive formulation leads to a loss of understanding on the part of the animator.

3.2.3.2 Spatial Deformation

A motion curve can be augmented spatially by applying scaling or translation warps. Again, specification by keyframing with interpolation by CTB splines is used. The deformed curve $\theta'(t)$ is calculated by multiplying the original $\theta(t)$ by the scaling spline $s(t)$ and then adding in the offset function $o(t)$.

$$\theta'(t) = s(t) \cdot \theta(t) + o(t) \tag{3.8}$$

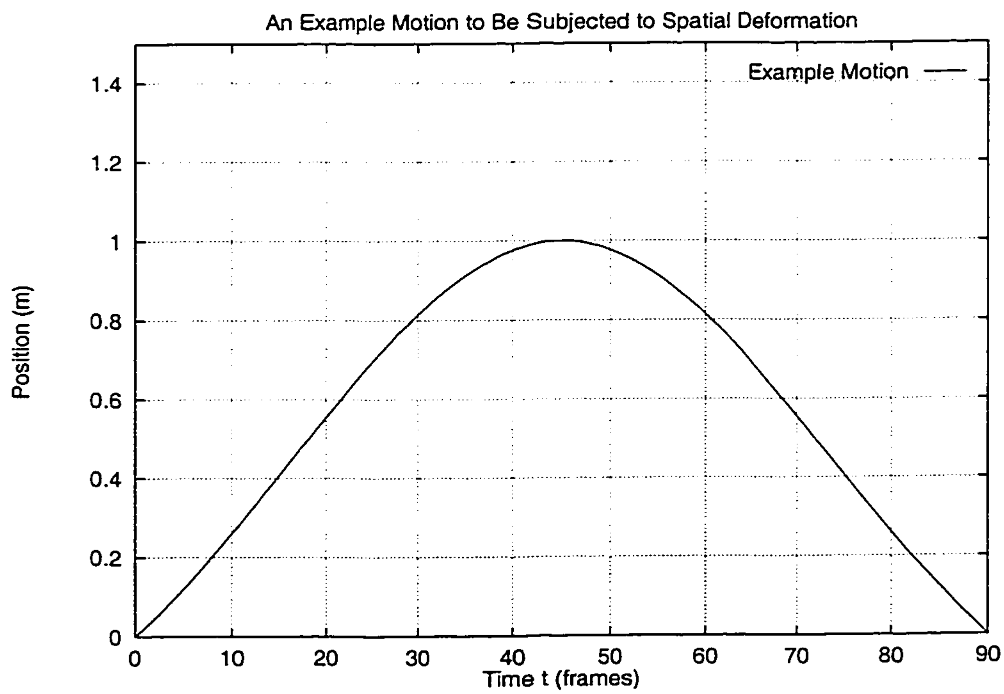


Figure 3.5: **A simple sample motion to be subjected to spatial warps. Figure 3.6 shows the effects of a scaling warp; Figure 3.7 illustrates an offset warp.**

The scaling spline passes through i user-defined points (t_i, s_i) whilst the offset spline intersects j points (t_j, o_j). Incorporating the time warp into the spatial deformation equation yields the complete motion warping function.

$$\theta'(t') = (s \circ f)(t') \cdot (\theta \circ f)(t') + (o \circ f)(t') \tag{3.9}$$

The open circle operator represents function composition.

$$(f \circ g)(x) = f(g(x)) \tag{3.10}$$

A simple motion, shown in Figure 3.5, is used to demonstrate the application of spatial warps. Two separate deformations, one involving a scaling function, the other an offset warp, are applied. The warping functions are similar in shape—for instance, the scaling curve's maximum peak value of two is equivalent in effect to the offset curve's maximum value of 0.638—yet produce different outcomes. The results are shown in Figures 3.6 and 3.7.

Because there are two forms of spatial warps, there are an infinite number of combinations of scaling and displacement that lead to equivalent results at the keyframes of the warping functions. That is, there are many values for (t_i, s_i) and (t_i, o_i) that result in the same $\theta'(t_i)$. Scaling can be used to exaggerate or understate the action. Displacement facilitates the introduction of new characteristics into the motion. The animator must choose what combination of scaling and offset creates the envisaged result.

3.2.4 Combining Motions

Motions can be blended together using weighted averaging. This technique can be employed to derive a new motion that combines the qualities of its two parents or to create smooth transitions between concatenated movements. Normalized s-shaped curves, $w_1(t)$ and $w_2(t)$, that vary between zero and one over the duration d are used as the weighting functions. This ensures that the endpoints of the changeover are continuous.

$$\begin{aligned} w_1(t) &= \frac{\cos(\frac{t}{d} \cdot \pi) + 1}{2} \\ w_2(t) &= 1 - w_1(t) \end{aligned} \tag{3.11}$$

Figure 3.8 graphically shows the slow-in, slow-out blending curves. The combined motion, $\theta_{blended}(t)$, is found by applying the weighting functions to the source motions.

$$\theta_{blended}(t) = w_1(t) \cdot \theta_1(t) + w_2(t) \cdot \theta_2(t) \tag{3.12}$$

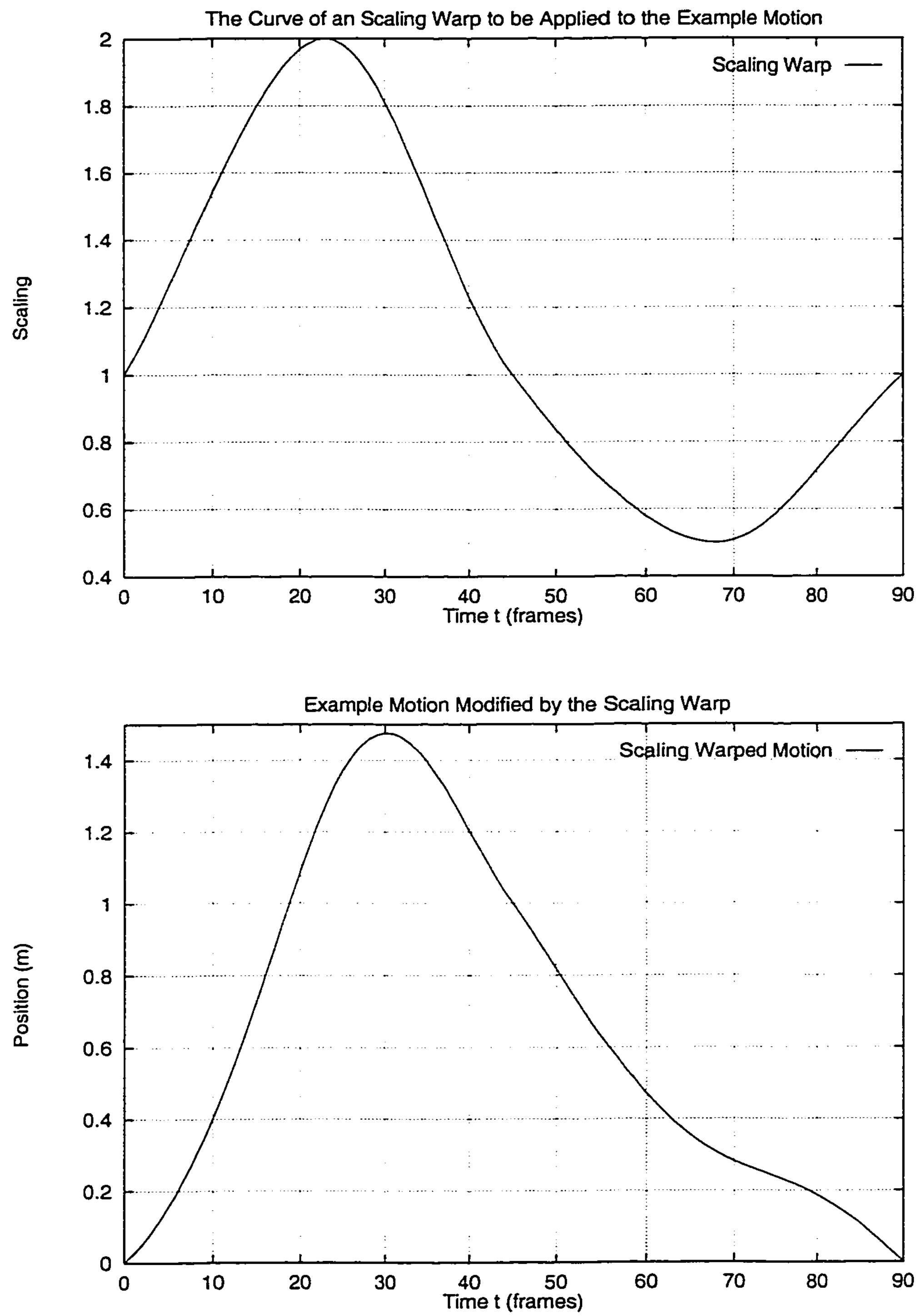


Figure 3.6: **A scaling warp applied to the example motion from Figure 3.5. The first half of the motion is exaggerated by scaling with a magnitude greater than one. The second half has been underemphasized by scaling with a fraction less than one.**

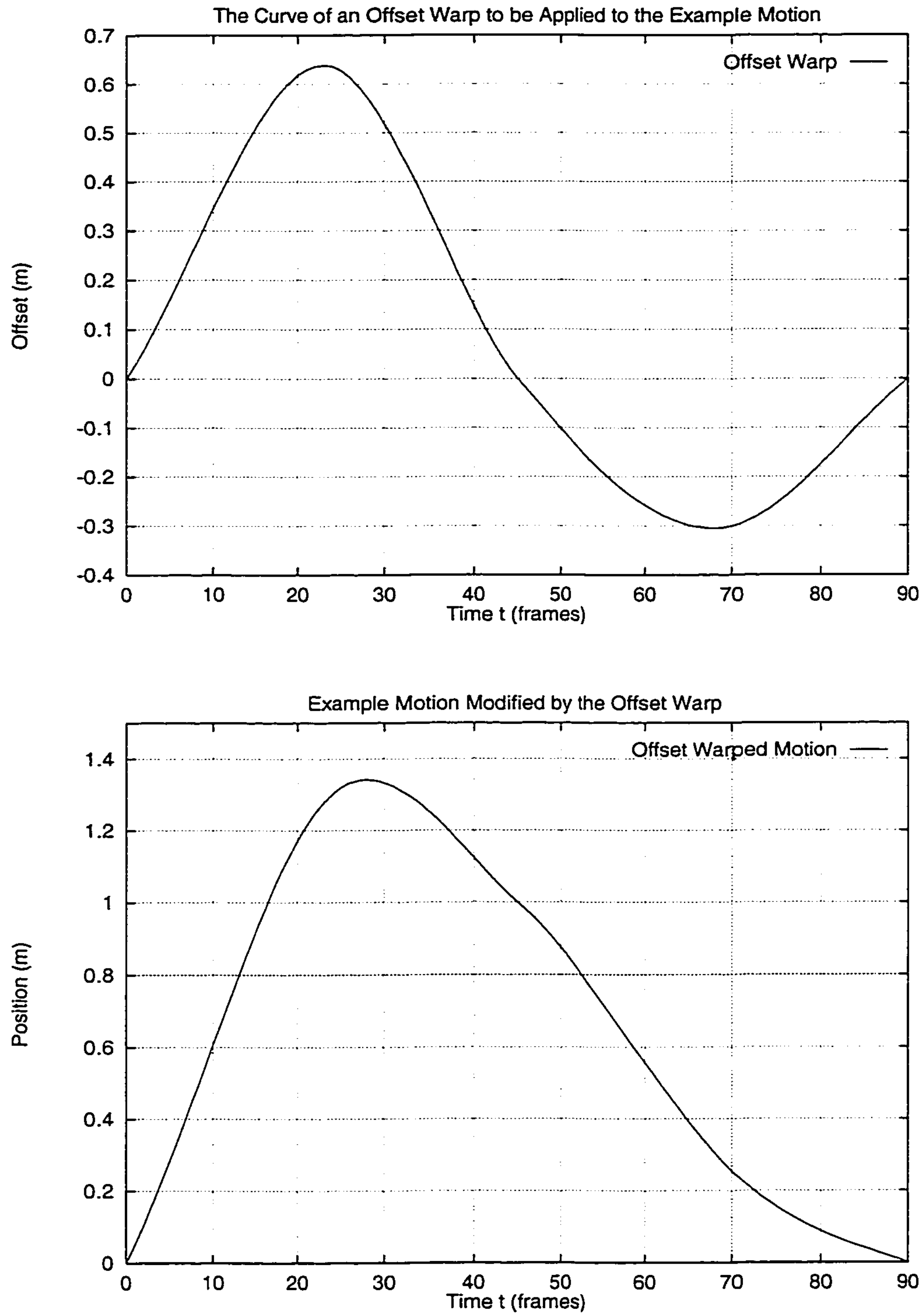


Figure 3.7: **An offset warp applied to the sample motion from Figure 3.5. The initial part of the motion has been extended by adding a positive offset, whereas the final portion has been decreased by a negative offset.**

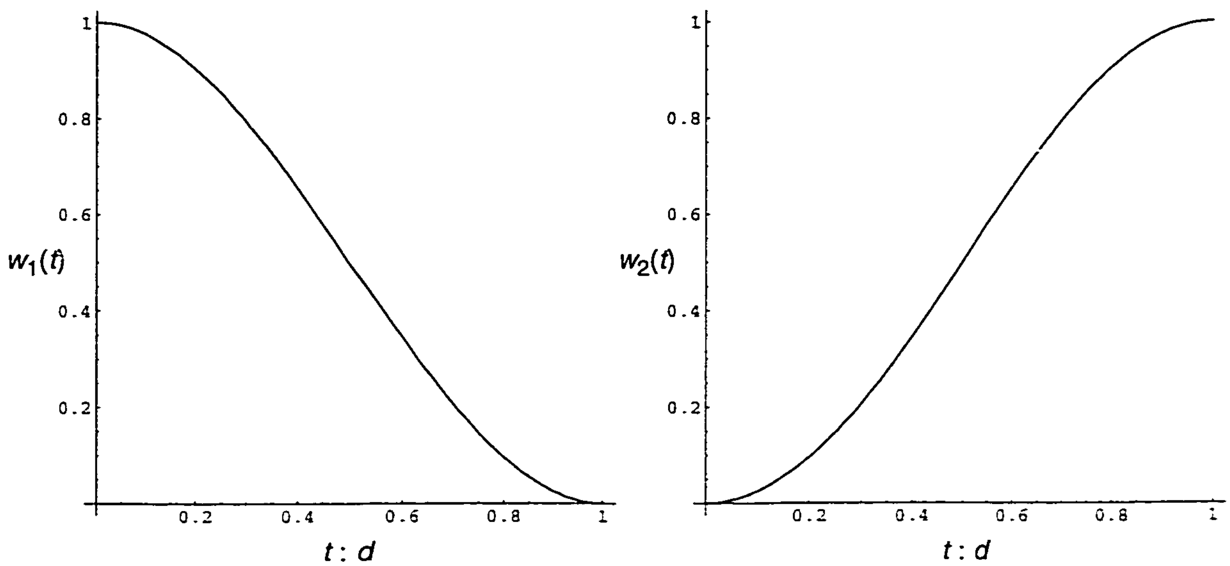


Figure 3.8: **The s-shaped weighting curves used in transitional blending. The left function, $w_1(t)$, is used to weight the contribution of the starting motion with values going from one to zero. The right function, $w_2(t)$, weights the ending motion with magnitudes from zero to one. The horizontal axis is the ratio of time into the transition, t, to the duration of the passage, d.**

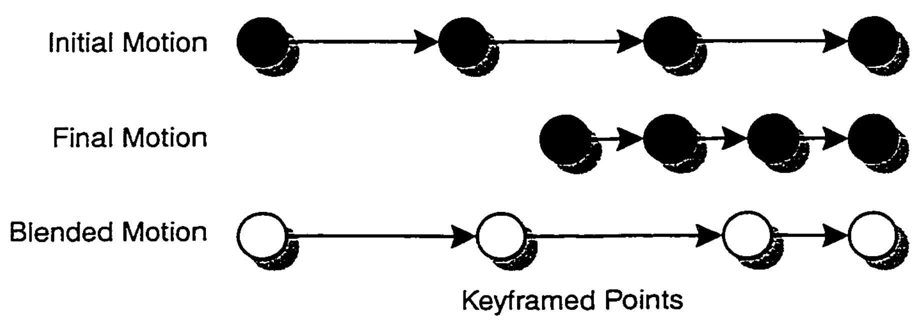


Figure 3.9: **A possible side-effect of blending two motions. The nodes in the original starting motion are distanced further apart than those in the ending motion. After applying transitional blending, the resulting mixed motion has an irregularly large initial speed not mirrored by its source motions.**

If the contributing motions are not suitably aligned, undesirable side effects are introduced into the combined movement. A simple demonstration of a misalignment artifact is depicted in Figure 3.9. The circular nodes represent positions or rotations; adjacent points are spaced equally apart in time. The nodes in the starting motion are located further apart than those in the concluding motion, suggesting a higher velocity. The transitional blending algorithm described above is then applied to the two source motions. The blended movement has an abnormally high initial speed, even greater than the original starting motion's. This effect was probably not the animator's intention. As this example illustrates, motion warping must first be applied to the constituent actions to bring them into alignment before blending takes place.

3.3 Motion Warping of Quadrupeds

A major objective of this work is to evaluate the applicability of motion warping to customizing quadrupedal locomotions. With this purpose in mind, several means of automatically tailoring animal gaits are developed. Adjusting the locomotion to make the animal negotiate a course along variable terrain is one useful capability. This requires positioning and rotating the torso of the creature to align with the path. In addition, due to curvature of the path and surface, the ankles and hips are adjusted to try to ensure stance legs stay affixed to the ground. These constraints are realized by computing appropriate offset warps.

The route is specified in terms of positions to be reached at chosen times. Therefore, variations in the speed of traversal of the path must be handled. The animal's speed affects the length of its step and, consequentially, the duration of its stride (see *Section 2.2.2—Gait Analysis*). Specifically, the time span of the stance and flight phases of the creature's legs change. To accomplish these effects, Alexander's general empirical equation relating stride length to velocity of motion is employed [3]. The leg state durations are modified using temporal warps.

An animal employs many styles of locomotion; the gait that is active is partly dependent upon the creature's speed. Transitions between locomotory styles are rarely instantaneous. Rather, the motions characterizing the succeeding gait are gradually introduced, eventually supplanting the movements of the initial gait. Temporal warps are used to line up and gradually adjust the leg phases of the gaits involved. Because equivalent states involve similar leg motions, no spatial reshaping is necessary. Blending of the temporally-aligned locomotions yields the complete transitory animation.

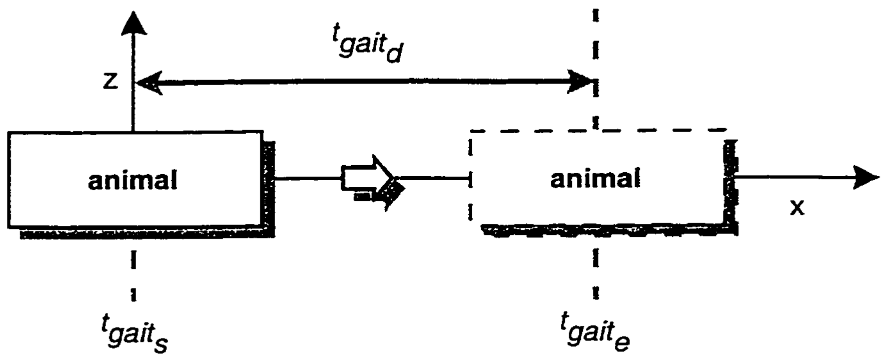


Figure 3.10: **The form of the sample stride of a gait. The motion is used as input into the motion warping system. At the beginning of the stride at time t_{gait_s}, the torso is centred at the origin. The animal proceeds to move along the x-axis until it reaches the end of the stride at time t_{gait_e}.**

The major input to the system consists of animation data for one stride of each gait. The animal, specifically its torso, moves along a single axis of the world coordinate system. The speed of the animal need not be constant. Fluctuations along the other two axes are also supported. The stride begins with the body centred at the origin at time t_{gait_s}, assumed to be at frame zero. The animal continues to move along the x-axis until the stride finally recycles at time t_{gait_e}. Thus, the animal's pose at t_{gait_s} is the same as at t_{gait_e}. This is shown diagrammatically in Figure 3.10.

The duration of a single stride of the gait, t_{gait_d}, is calculated as

$$t_{gait_d} = t_{gait_e} - t_{gait_s} \tag{3.13}$$

The average speed, $\bar{v}_{gait}$, of the animal during this stride is

$$\bar{v}_{gait} = \frac{x_{gait_e} - x_{gait_s}}{t_{gait_d}} \tag{3.14}$$

The positions x_{gait_s} and x_{gait_e} are the location of the torso along the x-axis at the start and end of the stride.

In addition to the raw motion data describing the gait, the limbs of the creature need to be identified to the system. Because the model links distal objects to their proximal counterparts, only the four foot objects need to be specified. Then, the links can be followed back to the torso to determine all of the segments of a leg. A text file containing this and other information accompanies the model and motion data file. The contents of this file will be discussed in a later section.

The system's design is purposefully simple. It is not meant to be highly accurate in a physical sense. Greater emphasis is placed on gross geometric aspects such as the placement of legs. The motive for creating the system is to test the viability of motion warping as an editing technique and its eligibility for automation.

3.3.1 Locomotion Along a Path

The first automated motion warping feature to be explicated is the application of a gait along a user-defined path. The route is expressed using three-dimensional CTB splines. This is a convenient representation—the course can be drawn interactively within an existing animation package. The terrain for the scene is first loaded into the animation software. The user then draws the path, ensuring that it lies flush against the ground[1].

Given a frame number i, the overall position of the animal along the route, P_i, is interpolated using the path spline (see Figure 3.11). The speed along the path varies, becoming greater as the distance between successive points lengthens. If the sample gait is applied to the path with a one-to-one correspondence between frames, any differences in speed between the two would cause the stance legs to slide. As a simple means to cope with this effect, a temporal scaling is used to speed up and slow down the gait according to the speed along the path.

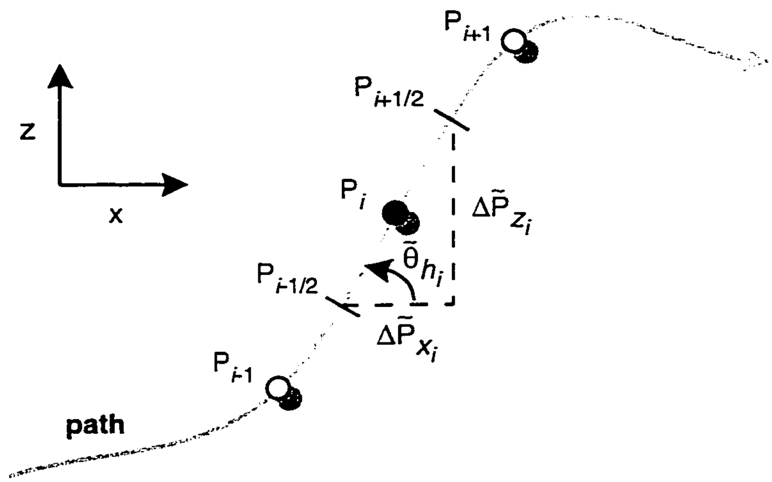


Figure 3.11: **Estimating the speed and direction of the animal along the path at point P_i. The direction is taken as the difference in position along each axis at points $P_{i-\frac{1}{2}}$ and $P_{i+\frac{1}{2}}$, each a half frame away from P_i. The velocity is also readily obtained from this difference.**

1. A plug-in for the animation application could be devised that automatically projects the path onto the ground. The animator then need only specify the route on a two-dimensional plane.

The distance covered along each axis at frame i is estimated by taking the difference between the point half a frame before, $P_{i-\frac{1}{2}}$, and the point half a frame ahead, $P_{i+\frac{1}{2}}$ (see Figure 3.11).

$$\begin{aligned} \Delta\tilde{P}_{x_i} &= P_{x_{i+\frac{1}{2}}} - P_{x_{i-\frac{1}{2}}} \\ \Delta\tilde{P}_{y_i} &= P_{y_{i+\frac{1}{2}}} - P_{y_{i-\frac{1}{2}}} \\ \Delta\tilde{P}_{z_i} &= P_{z_{i+\frac{1}{2}}} - P_{z_{i-\frac{1}{2}}} \end{aligned} \tag{3.15}$$

The total distance covered over the frame is

$$\tilde{d}_{path_i} = \sqrt{\Delta\tilde{P}_{x_i}^{\ 2} + \Delta\tilde{P}_{x_i}^{\ 2} + \Delta\tilde{P}_{x_i}^{\ 2}} \tag{3.16}$$

The speed, $\tilde{v}_{path_i}$, in metres per frame at point P_i is then approximated by

$$\tilde{v}_{path_i} = \frac{\tilde{d}_{path_i}}{\Delta t_{path_i}} \tag{3.17}$$

where Δt_{path_i}, the duration of a frame along the path, is always one. The adjusted time interval for the frame from the gait, $\Delta t_{gait_i}{}'$, at point P_i is

$$\Delta t_{gait_i}{}' = \Delta t_{gait_i} \cdot \frac{\tilde{v}_{path_i}}{\tilde{v}_{gait}} \tag{3.18}$$

where Δt_{gait_i}, the normal length of a frame from the gait, is always one, and $\tilde{v}_{gait}$ is the average speed given by Equation 3.14.

The problem of placing the animal on its route is broken down into two parts: positioning the torso such that it is aligned with the path and posing the legs to compensate for body rotations.

3.3.1.1 Positioning the Body

As the animal moves along a curved path, its torso turns. Lengthwise, the animal's body must lie parallel with the route, requiring a rotation. The direction of the course at point P_i is used to derive the required revolution.

$$\tilde{\mathbf{d}}_{path_i} = \frac{\Delta\tilde{P}_{x_i}}{\tilde{d}_{path_i}}\mathbf{i} + \frac{\Delta\tilde{P}_{y_i}}{\tilde{d}_{path_i}}\mathbf{j} + \frac{\Delta\tilde{P}_{z_i}}{\tilde{d}_{path_i}}\mathbf{k} \tag{3.19}$$

$\Delta\tilde{P}_i$ is found using Equation 3.15 and the total distance $\tilde{d}_{path_i}$ from Equation 3.16. The heading, $\tilde{\theta}_{h_i}$, is the angle between the x and z-axis components of the direction (see Figure 3.11).

$$\tilde{\theta}_{h_i} = \begin{cases} \operatorname{atan}\left(\frac{\Delta\tilde{P}_{z_i}}{\Delta\tilde{P}_{x_i}}\right), & \text{if } \Delta\tilde{P}_{x_i} > 0 \\ \operatorname{atan}\left(\frac{\Delta\tilde{P}_{z_i}}{\Delta\tilde{P}_{x_i}}\right) + 180°, & \text{if } \Delta\tilde{P}_{x_i} < 0, \Delta\tilde{P}_{z_i} \geq 0 \\ \operatorname{atan}\left(\frac{\Delta\tilde{P}_{z_i}}{\Delta\tilde{P}_{x_i}}\right) - 180°, & \text{if } \Delta\tilde{P}_{x_i} < 0, \Delta\tilde{P}_{z_i} < 0 \\ 0, & \text{if } \Delta\tilde{P}_{x_i} = 0 \end{cases} \tag{3.20}$$

There is a flip in direction as the heading passes through ±180°, although there are no visible repercussions of this inconsistency in the final animation. The pitch of the animal, $\tilde{\theta}_{p_i}$, as it climbs or descends is

$$\tilde{\theta}_{p_i} = \operatorname{atan}\left(\frac{\Delta\tilde{P}_{y_i}}{\sqrt{\Delta\tilde{P}_{x_i}^{\ 2} + \Delta\tilde{P}_{z_i}^{\ 2}}}\right) \tag{3.21}$$

The next task is to relocate the body around the point P_i on the path. First, the torso is positioned on the x-z plane. The animal cannot simply be centred at the point, as this ignores deviations in the lateral and longitudinal position in a gait that does not have a constant velocity. These fluctuations necessitate a displacement of the animal from the centred point P_i.

First, the frame of the gait, t_{gait_i}, associated with the path point P_i is calculated.

$$t_{gait_i} = \left(\sum_{n=0}^{i} \Delta t_{gait_n}{}'\right) \text{modulus } t_{gait_d} \tag{3.22}$$

The warped time of a frame, $\Delta t_{gait_i}{}'$, is from Equation 3.18 and t_{gait_d}, the duration of the gait, is from Equation 3.13. At time t_{gait_i} into the gait, the average longitudinal distance covered is

$$\bar{x}_{gait_i} = \bar{v}_{gait} \cdot t_{gait_i} \tag{3.23}$$

using $\bar{v}_{gait}$, the average gait speed, from Equation 3.14. The longitudinal displacement from this average is

$$\Delta x_{gait_i} = x_{gait_i} - \bar{x}_{gait_i} \tag{3.24}$$

where x_{gait_i} is the longitudinal position of the body during the gait at time t_{gait_i} (see Figure 3.12).

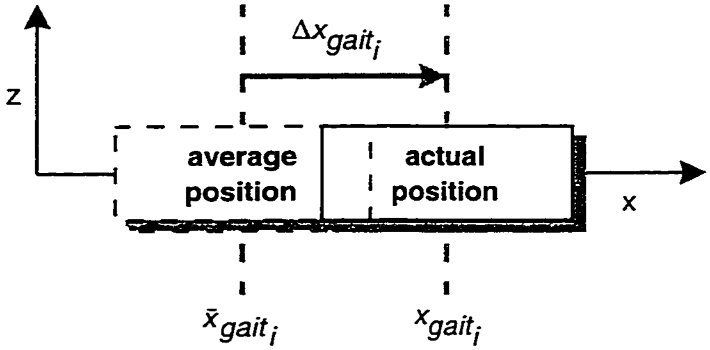


Figure 3.12: **Finding the longitudinal deviation of the animal from its average to actual position at time t_{gait_i} into its gait. Its average position is the location the creature would be if it had moved constantly at its average speed.**

Finally, the displaced position of the animal along the path, taking into account its heading, is determined.

$$\tilde{P}_{x_i}' = \tilde{P}_{x_i} + \cos(\tilde{\theta}_{h_i}) \cdot \Delta x_{gait_i} - \sin(\tilde{\theta}_{h_i}) \cdot z_{gait_i}$$
$$\tilde{P}_{z_i}' = \tilde{P}_{z_i} + \sin(\tilde{\theta}_{h_i}) \cdot \Delta x_{gait_i} + \cos(\tilde{\theta}_{h_i}) \cdot z_{gait_i} \quad (3.25)$$

The value z_{gait_i} is the latitudinal placement of the animal at time t_{gait_i} into the gait.

The last step to positioning the creature on the path is figuring its vertical displacement. In the sample gait, the height of the body, y_{gait_i}, at time t_{gait_i} guarantees clearance of the ground. To ensure clearance on an incline, the height should be applied perpendicularly to the terrain surface. As an alternative, the height can be rescaled and applied vertically from the path point $\tilde{P}_{y_i}$ (see Figure 3.13). The vertical offset of the animal's body is given by

$$\tilde{P}_{y_i}' = \tilde{P}_{y_i} + \frac{y_{gait_i}}{\cos(\tilde{\theta}_{p_i})} \quad (3.26)$$

For steep inclines, the pitch approaches ±90° and the cosine scaling factor increases towards infinity. To avoid such large vertical offsets, the pitch is capped at some lesser angle, say ±45°. There remains the potential for the feet to intersect or float above the terrain. This is due to the approximation of the ground used and the method for vertically displacing the animal's torso.

In the context of motion warping, Equation 3.22 acts equivalently to a temporal warp; it speeds up or slows down the playback of the gait depending on the path speed. Equations 3.20, 3.21, 3.26, and 3.26 set the location and direction of the animal using a combination of offset and scaling. These are automated forms of spatial warps.

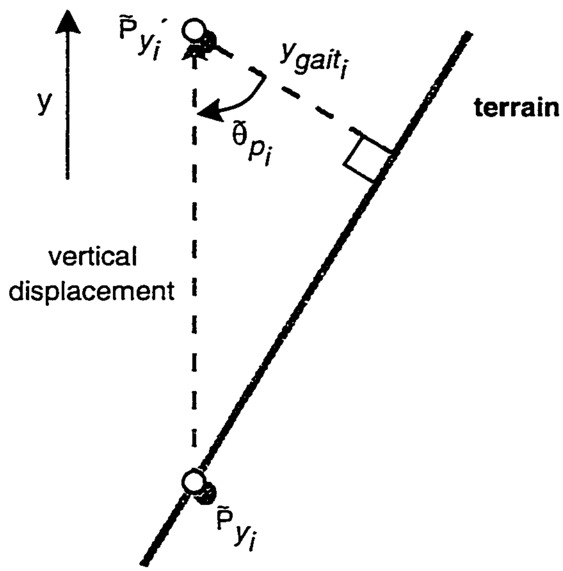


Figure 3.13: **Scaling the height of the animal above the ground based on the steepness of the terrain. Rather than displacing the animal perpendicularly away from the terrain, instead the creature is moved vertically. This helps prevent its feet from intersecting the ground.**

3.3.1.2 Arranging the Limbs

The sample gait assumes that the animal is moving straight across level ground. As such, the leg planes and the flats of the stance feet run parallel to the longitudinal axis of the animal. When negotiating uneven terrain, the bottoms of the stance feet are no longer parallel to this axis but are flush with the ground. When turning a corner, the shoulder or hips of the supporting legs must swivel whilst the feet remain affixed to the surface. These qualities must be added to the animal's gait.

When a leg begins its flight stage, preparations commence to bring the limb into position for its approaching stance phase. The ankle rotates to bring the flat of the foot roughly parallel to the upcoming ground. Meanwhile, the hips and shoulders revolve so that when the foot is planted on the ground, it will not twist about its vertical axis.

To implement these conditions, the ankle, hip, and shoulder joints of the model must be identified. A text file accompanies the sample gait that names the animal's torso object, number of limbs, titles of the foot objects, and the timings of each leg's support phase (see Figure 3.14). By moving up the model hierarchy from a foot to the body object, all the segments of a particular leg can be isolated. Of particular interest to the problem at hand are the ankle joints—associated with the feet—and the hip or shoulder joints connecting the upper leg to the torso.

When the flight stage of a leg commences, a look ahead is performed to find the position along the path at the middle of the following stance phase, taking into account the temporal warp due to the varying speed along the path. This midpoint location is assumed to be the terrain characteristics as the foot touches down. With regards to time, the middle of the stance phase of a leg, t_{stance_m}, occurs at

$$t_{stance_m} = \begin{cases} \frac{t_{stance_s} + t_{stance_e}}{2}, & t_{stance_s} \le t_{stance_e} \\ \frac{t_{stance_s} + t_{stance_e} + t_{gait_d}}{2} \text{ modulus } t_{gait_d}, & t_{stance_s} > t_{stance_e} \end{cases} \quad (3.27)$$

```
***
*** Gait Information for the Marten Ambling Motion Template
***

body    *** The name of the object representing the main body of the
        *** animal.

4       *** The number of limbs the animal has.

foot.frontRight 25 11   *** The names of the foot objects, followed
foot.frontLeft   6 31   *** by the ground touchdown and push off
foot.backRight   9 33   *** timings.
foot.backLeft   28 14

0.132   *** The average height (in metres) of the hip/shoulder joints
        *** above the ground when the animal is in a standing
        *** posture.
```

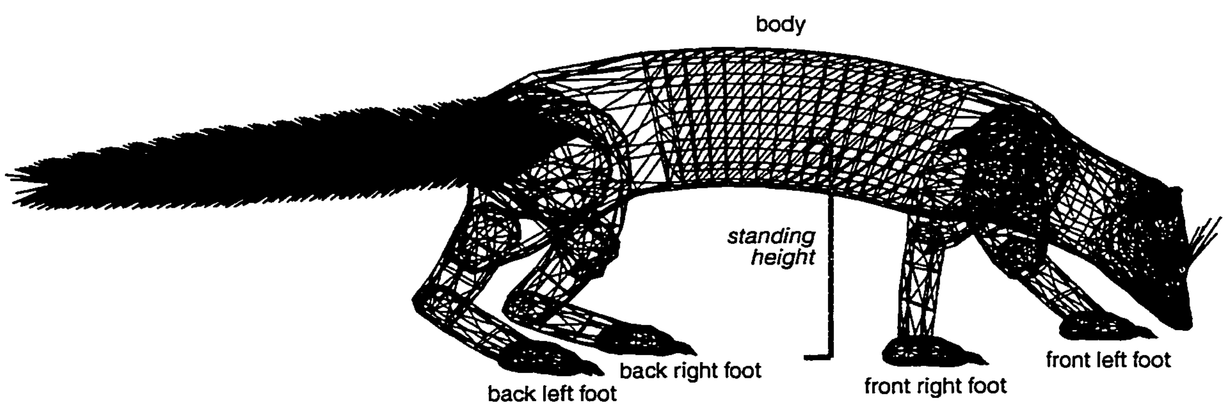


Figure 3.14: **Identification of the segments of an animal model. A text file names the body object, number of limbs, the foot objects, and the timings of the leg phases. By working up the model hierarchy from the foot to the torso, all the segments of a leg can be identified.**

The start and end times of the leg's stance stage are denoted by t_{stance_s} and t_{stance_e}, respectively. To find the time along the route when the next stance midpoint occurs, Equation 3.22 is repetitively applied until a frame i is found such that t_{gait_i} approximates t_{stance_m}. That is, the next minima of $|t_{gait_i} - t_{stance_m}|$ is sought. The position P_i along the path associated with $t_{gait_i} \cong t_{stance_m}$ is now known. For future reference, this midpoint shall be identified as $\tilde{P}_{stance_m}$. A similar algorithm is applied find the path position at the start of the next stance phase, $\tilde{P}_{stance_s}$.

The difference in the pitch and heading between the stance start and middle points, $\Delta\theta_{p_{ankle}}$ and $\Delta\theta_{h_{hip}}$ respectively, gives a rotational offset for the ankle and hip of the leg at touchdown.

$$\begin{aligned} \Delta\theta_{p_{ankle}} &= \tilde{\theta}_{p\,stance_m} - \tilde{\theta}_{p\,stance_s} \\ \Delta\theta_{h_{hip}} &= \tilde{\theta}_{h\,stance_m} - \tilde{\theta}_{h\,stance_s} \end{aligned} \tag{3.28}$$

Equations 3.20 and 3.21 are used to calculate the headings and pitches from the touchdown and mid stance path points $\tilde{P}_{stance_s}$ and $\tilde{P}_{stance_m}$. The pitch and heading adjustments adapt the leg to the sloped terrain and curved path at touchdown. The rotational displacements are linearly introduced over the course of the leg's flight phase. A more realistic action might be achieved by introducing the ankle revolution near the end of the flight stage.

When the leg enters its stance phase, the foot must stay flat on the ground and not pivot about its vertical axis. Thus, continual correction based on the path arc and ground slope must be applied to the supporting leg.

$$\begin{aligned} \theta_{p_{ankle_{i+1}}} &= \theta_{p_{ankle_i}} + (\tilde{\theta}_{p_i} - \tilde{\theta}_{p_{i+1}}) \\ \theta_{h_{hip_{i+1}}} &= \theta_{h_{hip_i}} + (\tilde{\theta}_{h_i} - \tilde{\theta}_{h_{i+1}}) \end{aligned} \tag{3.29}$$

The compensation is based on the change in pitch and heading between subsequent points along the route.

This approach for immobilizing the stance leg's foot is imperfect. There is still some lateral sliding of the foot due to curvature of the path. This is particularly visible through corners having a high curvature.

3.3.2 Varying the Speed of Locomotion

The sample gait only reflects locomotion at a specific fixed speed. However, a gait is generally valid over a range of speeds. In the previous section, a naïve speeding up or slowing down of the gait was applied as the animal's speed increased or decreased. This simplistic approach is inaccurate in nature and can be improved.

To this end, some observations concerning the relationship between gait speed, phase timings, and stride length need to be recognized.

- As the speed increases, the durations of both the stance and flight stages are diminished. This results in an overall reduction in the stride duration [23].
- The duration of the stance phase decreases faster than that of the flight phase [23].
- There is an invariant relationship, irrespective of the species of animal, between the speed of locomotion and the stride length [3].

$$\hat{\lambda} \cong 2.3\,\hat{v}^{0.6} \tag{3.30}$$

The stride length and speed in the above equation have been normalized by the height of the hips above the ground.

$$\hat{\lambda} = \frac{\lambda}{h} \qquad \hat{v} = \frac{v}{\sqrt{g \cdot h}} \tag{3.31}$$

In these equations, λ represents the stride length, v the speed, h the hip height, and g the gravitational constant[2]. The measurement is taken when the creature is in a standing posture.

These principles are integrated into the system to more accurately mirror the effects of velocity variations on a gait. The visual results of increased speed include not only timing changes. In flight, there is increased extension of the limbs and lifting of the feet. At touchdown, the joints are flexed further to cushion the impact. The quadruped motion warping system will address only the temporal effects of speed on a gait.

The sample gait, if designed correctly, should abide by the relationship between speed and stride length given in Equation 3.30. However, when this is not the case, the equation needs to be adapted to the gait. This is achieved by altering the scaling coefficient—normally having a value of 2.3—according to the average speed and stride length of the sample locomotion[3]. The modified function relating speed, v, to stride length, λ, becomes:

---

2. The acceleration due to gravity, g, is approximately 9.81 m/s.
3. Had two samples of the same gait at different speeds been available, then both the coefficient and exponent could be found.

$$\lambda = \lambda_{gait} \cdot \left(\frac{v}{\bar{v}_{gait}}\right)^{0.6} \tag{3.32}$$

where $\bar{v}_{gait}$ is the average speed (Equation 3.14) and λ_{gait} the stride length of the example gait.

$$\lambda_{gait} = x_{gait_e} - x_{gait_s} \tag{3.33}$$

The speed along the path varies, even as a single stride of the gait is being executed. Thus, the stride length is changing over the interval of the step. To find the stride length and duration when the speed is varying, a look ahead along the route is necessary. Eventually, the distance along the path approximately converges with the average stride length over that same section. This yields the correct stride length for the upcoming portion of the path.

$$\sum_{i=t_{stride_s}}^{t_{stride_e}} \tilde{d}_{path_i} \cong \frac{\sum_{i=t_{stride_s}}^{t_{stride_e}} \lambda_i}{t_{stride_e} - t_{stride_s} + 1} \tag{3.34}$$

In the above equation, $\tilde{d}_{path_i}$ represents the path distance subtended over frame i (Equation 3.16) and λ_i is the stride length for the path speed at frame i (Equation 3.32). The duration of the stride is calculated using the determined step start and end times, t_{stride_s} and t_{stride_e}.

$$t_{stride_d} = t_{stride_e} - t_{stride_s} + 1 \tag{3.35}$$

During its stance phase, a foot is locked in place on the ground and thus does not cover any distance of the stride length. Rather, the foot covers the entire stride length during its leg's flight stage. Any variation in speed between the sample gait and path results in slippage of the stance foot. By temporally speeding up or slowing down the playback of the leg motion during its support phase, this problem is corrected. The appropriate temporal warping function was given in Equation 3.18.

The time remaining in the stride, t_{flight_d}, is devoted to the flight phase.

$$t_{flight_d} = t_{stride_e} - t_{stance_d} \tag{3.36}$$

The duration of the stance stage, t_{stance_d}, is known at its completion. Thus, frames from the flight stage of the gait are sampled at an adjusted interval.

$$\Delta t_{gait_i}' = \Delta t_{gait_i} \cdot \frac{t_{flight_d}}{t_{gaitflight_d}} \tag{3.37}$$

where $t_{gait flight_d}$ is the duration of the flight stage of the unmodified sample gait. Δt_{gait_i}, the normal duration of a frame, is always set to one.

The major deficiency of this approach is that the phases for the legs are considered independently. This results in a cumulative loss of synchronization of the limbs. That is, the temporal relationships between the phases of each leg change. This side effect limits the usefulness of this approach for applying speed effects to a gait. However, for constant velocities, synchrony is guaranteed.

3.3.3 Transitions Between Gaits

A gait is only valid over a limited range of speeds. Once this extent is exceeded, a different gait must be activated. However, the transition from one gait into the next is rarely instantaneous. Rather, the shift occurs over a period of time.

To generate a transition between gaits, the quadruped motion warping system performs two steps. First, the two contributing gaits are temporally aligned. This involves creating two intermediate animations, one for each gait. Within these animations, the phase timings are linearly shifted and scaled from those of the first gait into those of the second. In each case, the limb motions of the source gait are retained. The second step is to blend these two intermediate animations together. The process is as described in *Section 3.2.4—Combining Motions.*

```
***
*** Gait Information for the Marten Bounding Motion Template
***

body     *** The name of the object representing the main body of the
         *** animal.

4        *** The number of limbs the animal has.

foot.frontRight 11 13   *** The names of the foot objects, followed
foot.frontLeft  11 13   *** by the ground touchdown and push off
foot.backRight   1  5   *** timings.
foot.backLeft    1  5

0.132    *** The average height (in metres) of the hip/shoulder joints
         *** above the ground when the animal is in a standing
         *** posture.
```

Figure 3.15: **Phase timing information for the marten's bounding gait. The bounding and ambling gait from Figure 3.14 share the same objects, including names. However, the two examples of locomotion differ in the state timings of the limbs.**

As input, the animator provides the movements of one stride of each of the two gaits involved. In addition, an information file identifying the limbs and state timings is included with each gait (see Figure 3.14 and 3.15). Since the two sample locomotions represent the same animal model, they contain identically named objects. Of course, the phase timings of the two gaits differ. Finally, the animator gives the duration of the transition, $t_{transition_d}$.

In creating the intermediate aligned animations, consideration is given to modifying the leg state timings as well as the position of the body. Over the time span of the transition, the stance and flight stages must be temporally shifted and resized from those of the first gait to those of the second. Figure 3.16 graphically shows the phase modifications that occur as an amble transforms into a bound.

Because there is a continual change of the phase timings throughout the transition, a neutral means of representing the current point in the stride sequence for each leg is used. The first state element is the current phase of the limb, be it the support or flight stage. The other element is the fraction into the phase at some intermediate frame i, denoted u_i, which has a real value between zero and one.

The gait phases of an in-between frame i are delineated by the time span of the stride, t_{gait_d}, and the starting time and duration of the stance stage for each limb, t_{stance_s} and t_{stance_d}, respectively. The choice of these three values is somewhat arbitrary, but is sufficient to fully define the phase timings for the transitory locomotion.

The duration of the stride is found with

$$t_{transition_d} = (1 - r_{transition}) \cdot t_{starting\,gait_d} + r_{transition} \cdot t_{ending\,gait_d} \tag{3.38}$$

where $t_{starting\,gait_d}$ is the duration of the starting gait, $t_{ending\,gait_d}$ the time span of the concluding gait, and $r_{transition}$ is the fraction into the transition as calculated using

$$r_{transition} = \frac{i}{t_{transition_d}} \tag{3.39}$$

Similarly, the time span of the support phase of a leg is

$$t_{transition\,stance_d} = (1 - r_{transition}) \cdot t_{starting\,stance_d} + r_{transition} \cdot t_{ending\,stance_d} \tag{3.40}$$

with $t_{starting\,stance_d}$ being the duration of the stance phase of the beginning gait and $t_{ending\,stance_d}$ that of the ending gait. The support stage for the limb commences at time

$$t_{transition\,stance_s} = (1 - r_{transition}) \cdot t_{starting\,stance_s} + r_{transition} \cdot t_{ending\,stance_s} \tag{3.41}$$

where $t_{starting\,stance_s}$ is the start time of the initial gait's support phase, and $t_{ending\,stance_s}$ is the start time of the concluding gait's stance stage.

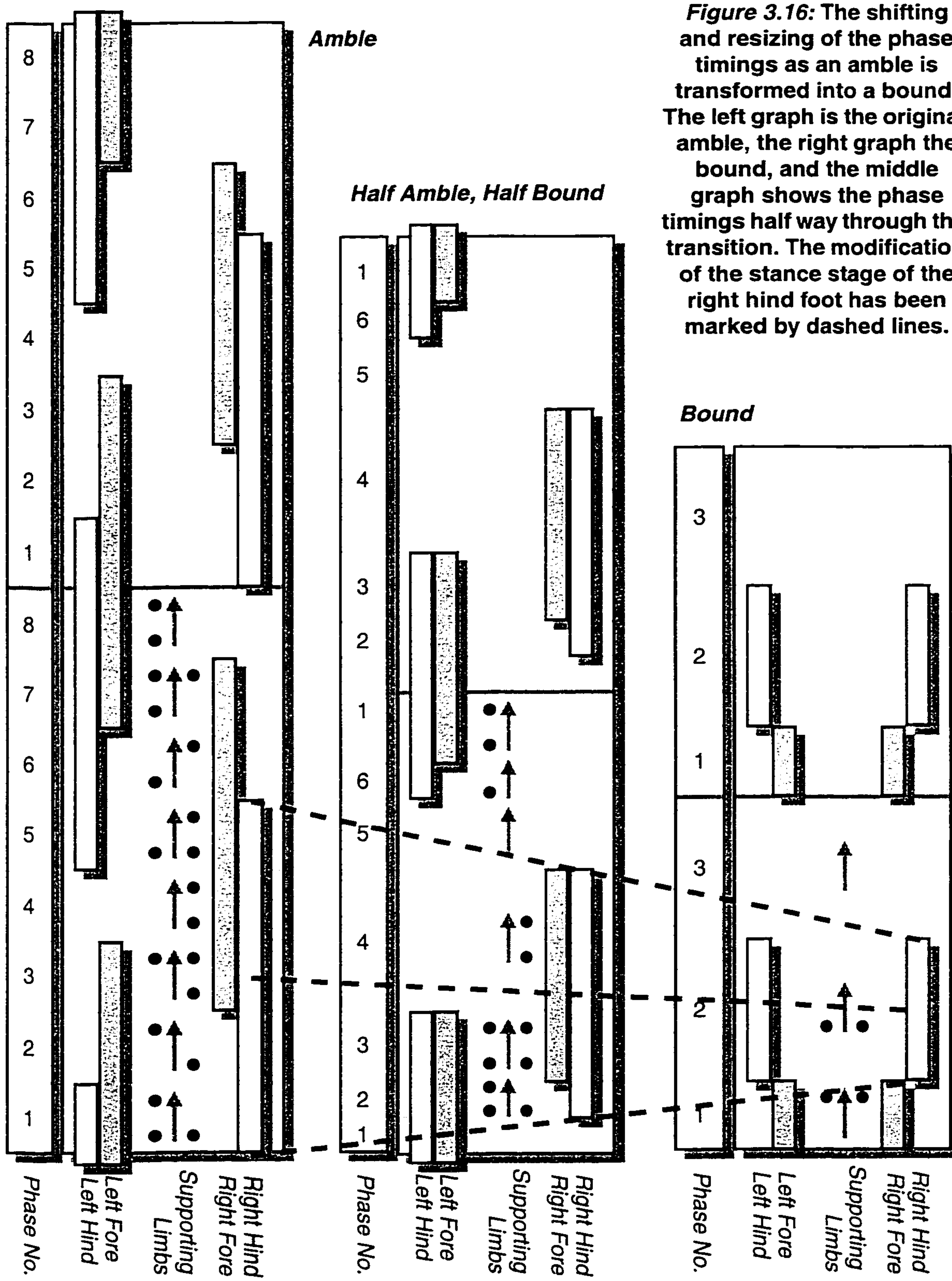


Figure 3.16: **The shifting and resizing of the phase timings as an amble is transformed into a bound. The left graph is the original amble, the right graph the bound, and the middle graph shows the phase timings half way through the transition. The modification of the stance stage of the right hind foot has been marked by dashed lines.**

The relative location in the stride sequence from the previous frame is $u_{i\text{-}1}$. This can be converted into a concrete frame number, t_i, within the current transitory gait.

$$t_i = \begin{cases} \left(t_{transition\,stance_s} + u_{i-1} \cdot t_{transition\,stance_d} + 1\right) \text{ modulus } t_{transition_d}, & \text{in stance phase} \\ \left[t_{transition\,stance_s} - (1 - u_{i-1}) \cdot \left(t_{transition_d} - t_{transition\,stance_d}\right) + 1\right] \text{ modulus } t_{transition_d}, & \text{in flight phase} \end{cases} \tag{3.42}$$

The frame number must be converted back to its relative form, which is then used to access the leg motion from the original gait. Care must be taken to update the current phase beforehand, as this may have changed between frames. The fractional position into the support stage is given by

$$u_i = \begin{cases} \dfrac{t_i - t_{transition\,stance_s}}{t_{transition\,stance_d}}, & t_i \geq t_{transition\,stance_s} \\ \dfrac{t_i + t_{transition_d} - t_{transition\,stance_s}}{t_{transition\,stance_d}}, & t_i < t_{transition\,stance_s} \end{cases} \tag{3.43}$$

and similarly for the flight phase by

$$u_i = \begin{cases} \dfrac{t_i - t_{transition\,flight_s}}{t_{transition\,flight_d}}, & t_i \geq t_{transition\,flight_s} \\ \dfrac{t_i + t_{transition_d} - t_{transition\,flight_s}}{t_{transition\,flight_d}}, & t_i < t_{transition\,flight_s} \end{cases} \tag{3.44}$$

Finally, the associated frame from the original motion is found using

$$t_{gait_i} = \begin{cases} \left(t_{gait\,stance_s} + u_{i-1} \cdot t_{gait\,stance_d} + 1\right) \text{ modulus } t_{gait_d}, & \text{in stance phase} \\ \left[t_{gait\,stance_s} - (1 - u_{i-1}) \cdot \left(t_{gait_d} - t_{gait\,stance_d}\right) + 1\right] \text{ modulus } t_{gait_d}, & \text{in flight phase} \end{cases} \tag{3.45}$$

This process repeats for each leg over the entire interval of the transition (*i.e.* until $i = t_{transition_d}$).

When generating the intermediate animations, the second problem requiring attention is the position of the body. Compensation is required for latitudinal and longitudinal oscillations of the torso as discussed in *Section 3.3.1.1—Positioning the Body*. Each leg may use the motion from a different frame of the original gait. This makes it difficult to choose the motion of the remaining animal segments, particularly the body. A simple choice is to use the average of the torso positions, $P_{body_{i,\,l}}$, associated with each leg l.

$$P_{body_i} = \frac{P_{body_{i,\,l}}}{4} \tag{3.46}$$

This approach is not truly sufficient. The position of the body is propagated to the legs, which may result in unwanted intersections with the ground or, during the support stage, the feet not contacting the ground. Another artifact is that the torso movement may appear "jumpy" rather than smooth.

After the two in-between animations for each gait are created, the results can be progressively blended. This is a straightforward procedure as described in *Section 3.2.4—Combining Motions.*

3.4 Summary

In this chapter, the particulars of an animation system based on Witkin and Popović's motion warping technique were presented. First, consideration was given to the structure of the articulated geometric model. Next, the means for obtaining sample locomotions and the representation of these movements were addressed. The generic motion warping equations, both temporal and spatial, were revealed. One notable application of motion warping involved the blending of two movements.

An architecture for automatically adapting the locomotions of quadrupeds via motion warping was disclosed. One feature of the prototype system involved the application of a single sample stride of a gait to a three-dimensional, user-specified path. Another capability was the temporal alteration of the limb phase timings based on the speed of locomotion. Lastly, a facility for generating transitions via blending was discussed.

In the next chapter, some case studies involving motion warping in general and the quadruped system in particular will be shown. These examples highlight some of the capabilities and deficients of the motion warping technique. In addition, results from of the quadruped animation system will be presented. These tests will be used to judge the viability of motion deformation for animating animal locomotions.

Chapter 4

Evaluation of Results

In the previous chapter, the specifics of motion deformation and the workings of a quadruped animation system utilizing warping were detailed. However, there was little in the way of critical evaluation presented. The goals of this chapter are twofold. One priority is to demonstrate, via examples, the motion warping method and the quadruped locomotion system. Based on these experiences, the second objective is to evaluate the proposed animation approaches, identifying their strengths and deficiencies.

The results are broken down into three case studies. A bouncing beach ball serves as a preliminary example of the various types of deformation. The source action features the sphere springing in place four times. This action is modified into a decaying bounce via a scaling warp, a hop up a some stairs using an offset warp, and a "running jump" by blending. By studying the process of defining warps, the effectiveness of deforming as opposed to directly modifying motions can be evaluated.

The next tests involve customizing a swivelling arm movement. The goal is to make the robotic arm achieve some new intermediate posture whilst retaining the characteristic movements of the original swing. First, the swivel action is adjusted by hand to yield a subject for comparisons. Then, various weights of scaling and offset warps will be tried that satisfy the goal posture. These trials will give a better understanding of the effects of the two forms of spatial deformation.

The final examples involve the animation of a marten using the quadruped motion warping system. These demonstrations involve two gaits—an amble and a bound. A winding path is drawn over some rolling terrain and the bounding locomotion applied to this route. A second experiment involves creating a transition from the amble to the bounding gait. Although the simple system for reshaping the animal's locomotions works as expected, deficiencies in the resulting animations highlight the need for more complex control schemes.

4.1 A Simple Demonstration of Motion Warping—The Bouncing Beach Ball

A ball is one of the simplest objects to animate because it does not contain articulated parts. To illustrate the basics of augmenting motions via warping, a bouncing ball is used as a test subject. Its bouncing movement will be reshaped into a decaying bounce and a hop up some stairs, and blended with a rolling motion to create a "running jump" effect. The purpose of these examples is to demonstrate the steps involved in creating and applying deformations. The defining of the warping functions is explained with the assistance of graphs. The efficiency of this specification process and the problems of representing the deformed motions is addressed. Much of this material was original presented in the paper "Augmenting Animations Via Motion Warping" [42].

4.1.1 The Original Bouncing Ball Movement

The prototype action features the ball bouncing in place four times, always reaching the height. The motion spline is constrained with keyframes at the point of ground contact and at the apex of the bounce. There is an instantaneous change in velocity as the ball strikes the ground, implemented by setting the CTB spline continuity parameter to a minimum at these points. Figure 4.1 is a graph of the vertical motion curve, and Figure 4.2 shows some selected frames from the bouncing action.

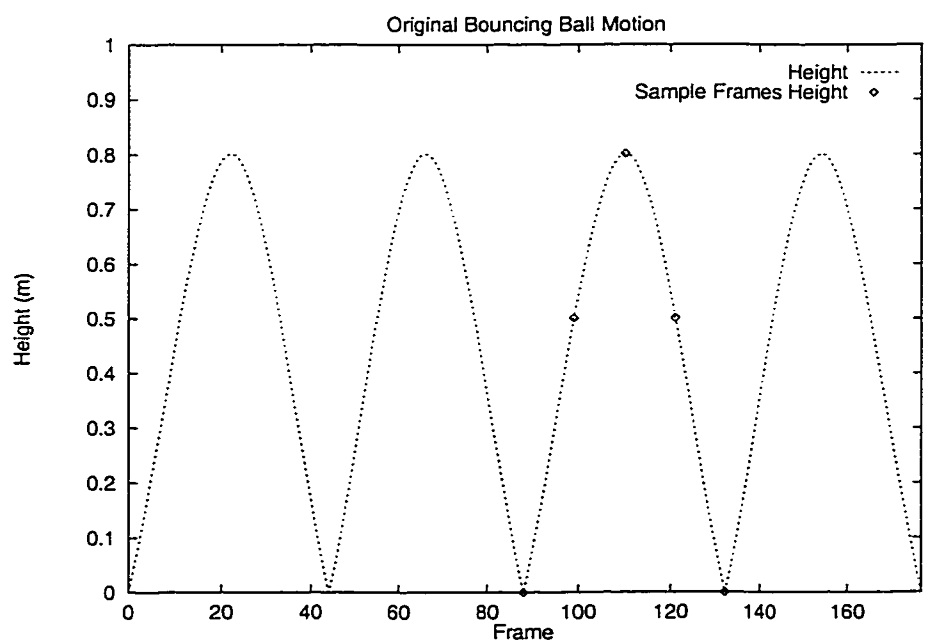


Figure 4.1: **A graph of the original motion of the bouncing beach ball. The ball bounces on the ground four times, each time reaching the same apogee. The marked frames correspond to those of the sample images presented in Figure 4.2.**

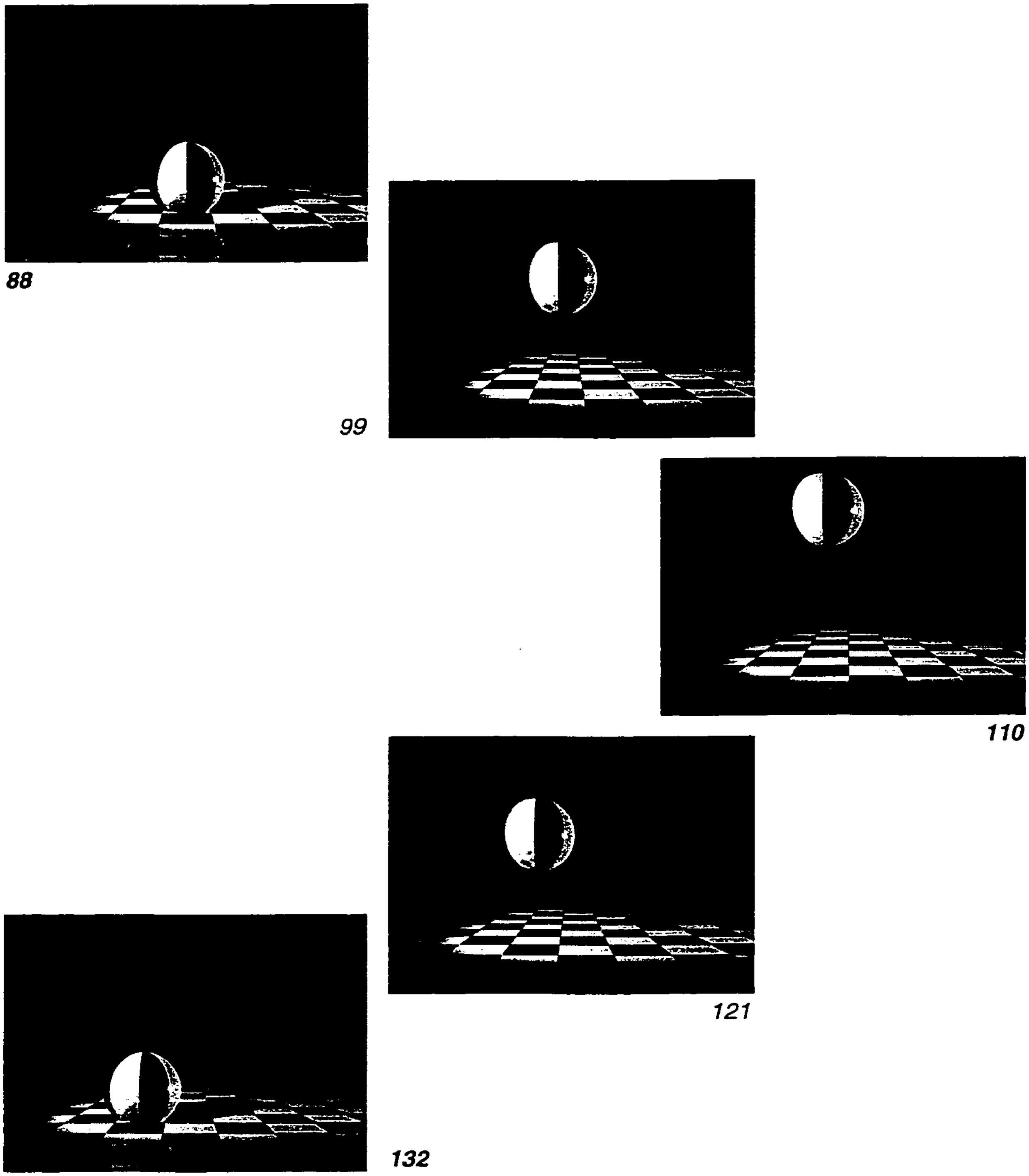
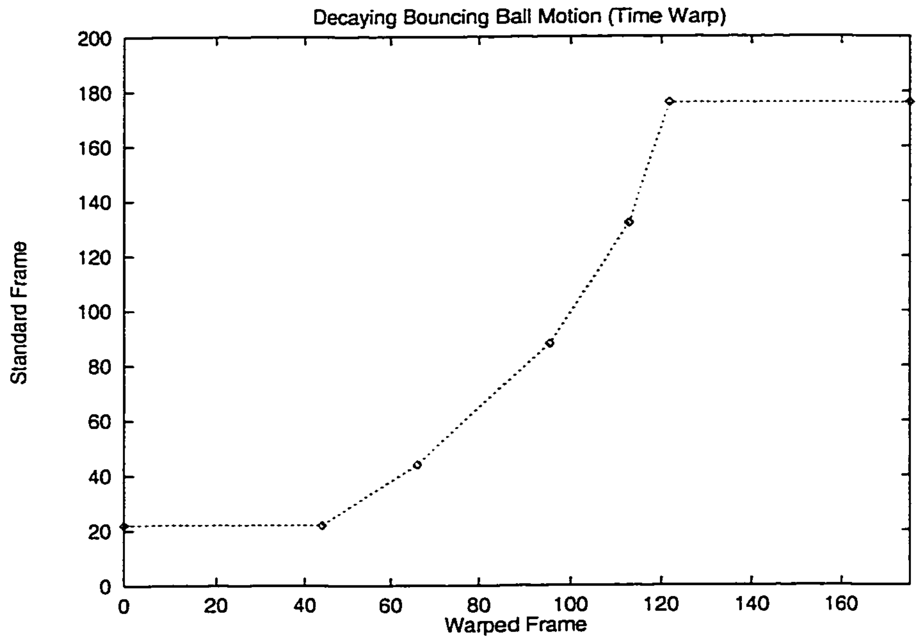


Figure 4.2: **Rendered frames from the original bouncing beach ball animation. The values beside each of the images indicate the frame number; the bold variants identify the keyframes. The frame numbers are marked on the motion graph in Figure 4.1.**



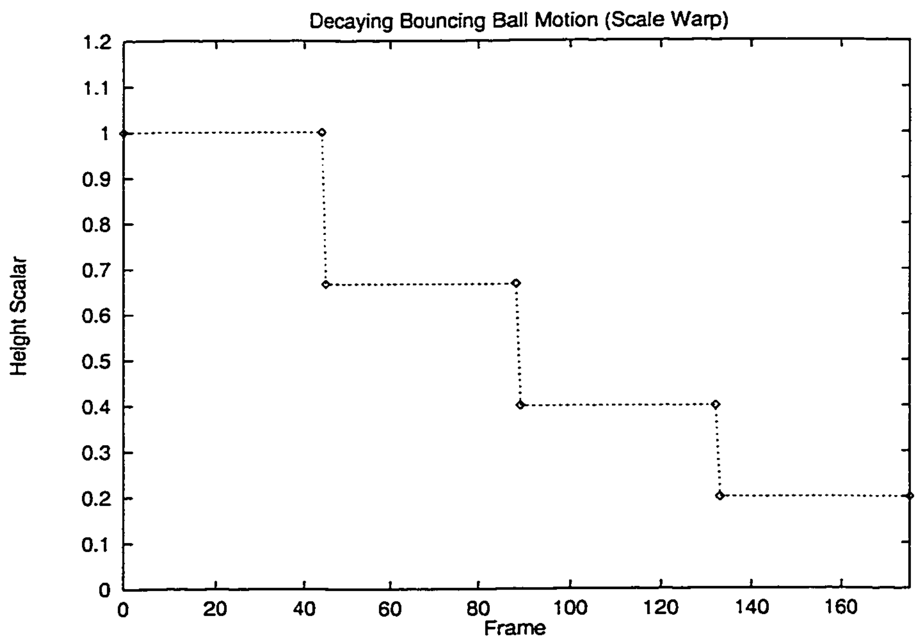


Figure 4.3: **The motion warping curves that yield a decaying bounce when applied to the original (given in Figure 4.1). Each hop is reduced, both temporally and in height, using piecewise linear segments. The keyframes for these segments are marked by points in the above graphs. Each bounce is of a shorter duration, so the temporal warp speeds up the playback of each successive bound (see top graph). The altitude is diminished via scaling by increasingly smaller fractions (see bottom graph). Figure 4.5 shows a few frames from the resulting animation.**

4.1.2 A Decaying Bounce Created Using Temporal and Scaling Warps

The first experiment with motion deformation involves the creation of a decaying bounce. This requires the height and time span for each consecutive bounce to decrease. These effects are achieved by specifying splines that warp the time and scale of the motion. In a physics sense, the total energy of the ball remains constant when the sphere is in the air, but sustains an instantaneous loss when contacting the ground. This suggests that modifications to the timing and height of the bounce can be expressed by piecewise linear functions with one segment per spring. The slope of each successive segment of the temporal warp is steeper, meaning that time is speeding up. Constant-valued fragments are used to scale the height of each jump. Figure 4.3 shows the graphs for these two deformation functions. Frames from the resulting animation are depicted in Figure 4.5.

This example illustrates that motion warping is not necessarily a more efficient means of editing over modifying the original curves directly. Customizing the source motion necessitates changing the height of four points and the time of six. In contrast, the deformation splines require the placement of fifteen points. In this case, it requires less manipulation to edit the source motion than to create the warping curves. Of course, the bouncing ball is a simplistic action, requiring few keyframes to specify its motion. Thus, altering the motion directly is simple. A more complex animation, especially one consisting of motion capture data, would be a better candidate for warping.

4.1.3 An Offset Warp for Generating a Bounce up a Series of Steps

The next sample involves altering the bounce so that the ball climbs a series of stairs. Only an offset warping spline was used to insert new characteristics into the bounce. Linear segments specify the horizontal, vertical, and depth displacements as the sphere moves amongst the steps. To add variety, the ball is made to spin in its direction of motion. The individual parameters of the offset warping spline are graphed in Figure 4.4. Frames from the resulting animation are shown in Figure 4.5. As this example demonstrates, an object may be repositioned and new qualities added, such as spinning, without editing the original data.

There is a notable difference in the way a warped motion is represented as compared to a movement that was edited directly. Assuming that the original action takes the form of a keyframed spline, then the source spline may be adjusted by manipulating existing control points and, if necessary, a few new ones added. Application of deformation curves involves the composition, addition, and multiplication of splines representing the time, offset, and scaling warps, respectively. The resulting movement is difficult to represent as a spline. For one thing, the motion spline needs to be reparameterized according to the time warp. Also, addition and multiplication by the spatial deformation splines result in a high-order polynomial. In particular, since CTB splines involve cubic polynomials, the resulting motion spline is of sixth-degree.

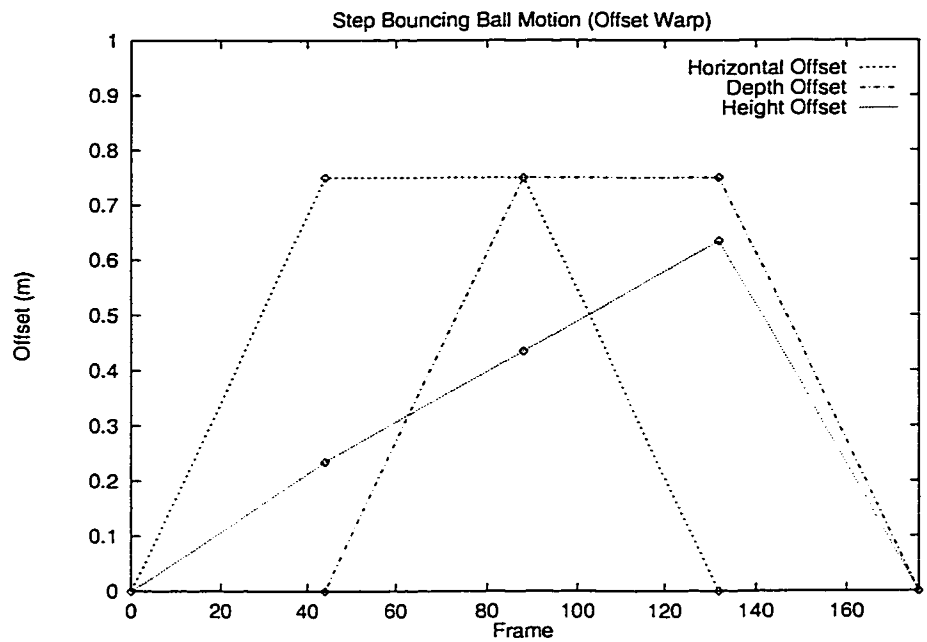
There are a number of reasons why high-order polynomial splines are a poor representation:

- One problem is that existing animation packages only support traditional cubic forms of splines. It is difficult to provide meaningful controls to an animator for editing such splines. An approximation of lesser degree could be estimated, but this negates one of the main benefits of warping, sacrificing the details inherent in the original action.
- The source motion may not be in the form of sparely-keyframed splines. For instance, motion capture data is composed of points captured at regular, closely-spaced intervals. In this case, there is little need to represent the reshaped motion as a polynomial spline[1].

The compromise adopted by our motion warping architecture is to represent the new motion by splines containing points sampled at each frame in the warped time frame. This representation is effective for any form of source motion, be it spline-based or from motion capture data. Because the deformed motion is comprised only of keyframed points and splines, it is compatible with existing animation applications.

There is the potential to lose data through successive deformations if a record of all operations is not maintained. In the preliminary warp, if the time is sped up, then a sequence of data points might be compressed to lie between frames. Because the granularity is limited to whole-numbered frames, these points are removed from the resulting motion. If, in the subsequent warp, time is expanded around this interval, then the information encoded in the now missing points becomes important. If the original motion data becomes unavailable, then this information is not replaceable. Instead, interpolation must suffice to represent this absent data.

1. An exception would be a time warp that slows down the playback of the motion, necessitating interpolation between captured points.



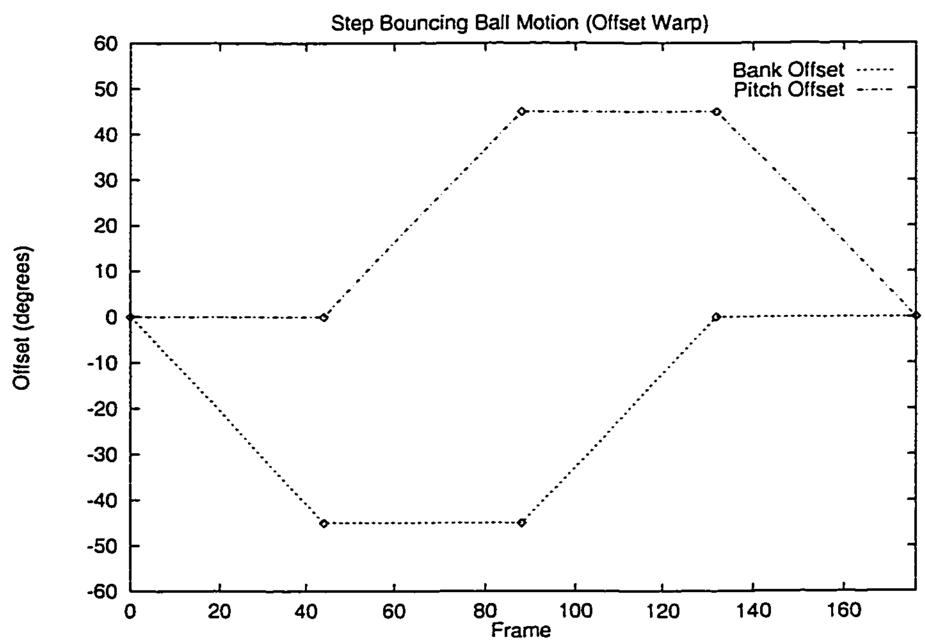


Figure 4.4: **To reshape the bouncing motion so that the ball climbs up a series of steps, an offset warp is applied. The position and rotation of the sphere are modified using linear segments as defined by the keyframed points in the above graphs. Application of the warp makes the ball bounce atop each stair in turn. Figure 4.5 shows some frames from the final animation.**

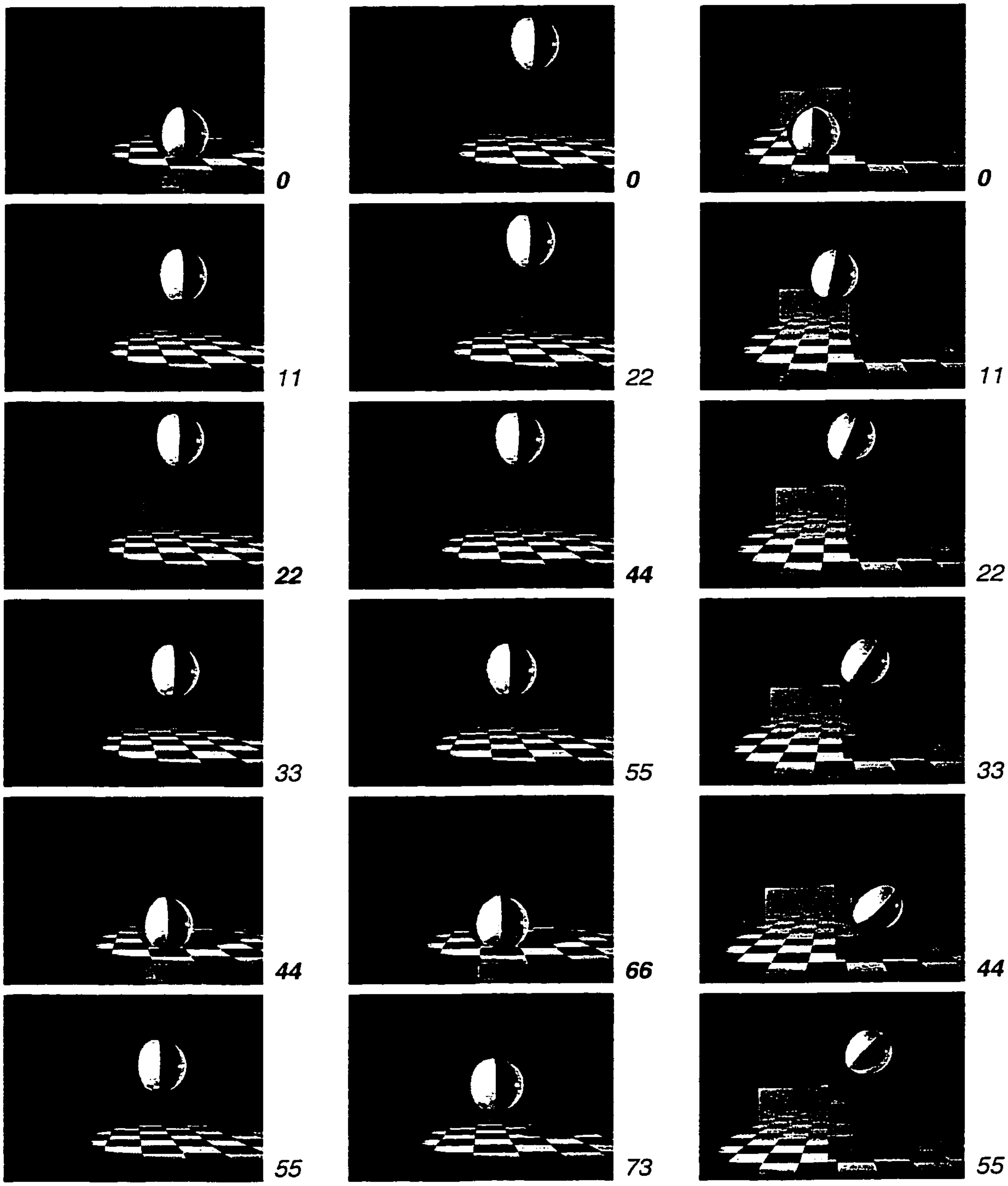


Figure 4.5: **Two examples of motion warping. The first column depicts frames of the original bouncing ball movement. The second features images of the decaying bounce, created using temporal and scaling warps (see Figure 4.3). The third column shows the ball hopping up a series of steps; this effect was accomplished via offset warps (see Figure 4.4).**

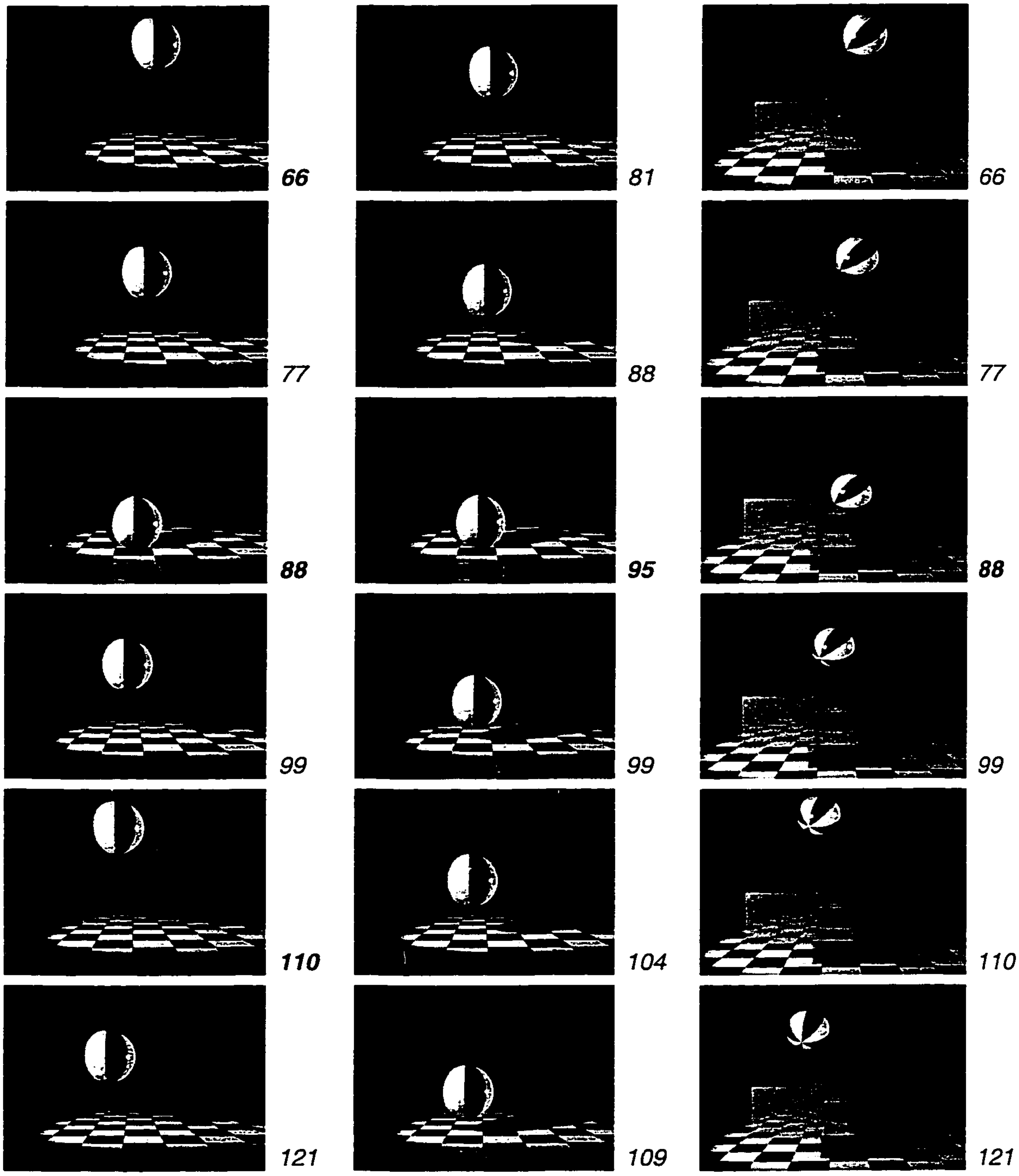


Figure 4.5 (Cont.): **Two examples of motion warping. The first column depicts frames of the original bouncing ball movement. The second features images of the decaying bounce, created using temporal and scaling warps (see Figure 4.3). The third column shows the ball hopping up a series of steps; this effect was accomplished via offset warps (see Figure 4.4).**

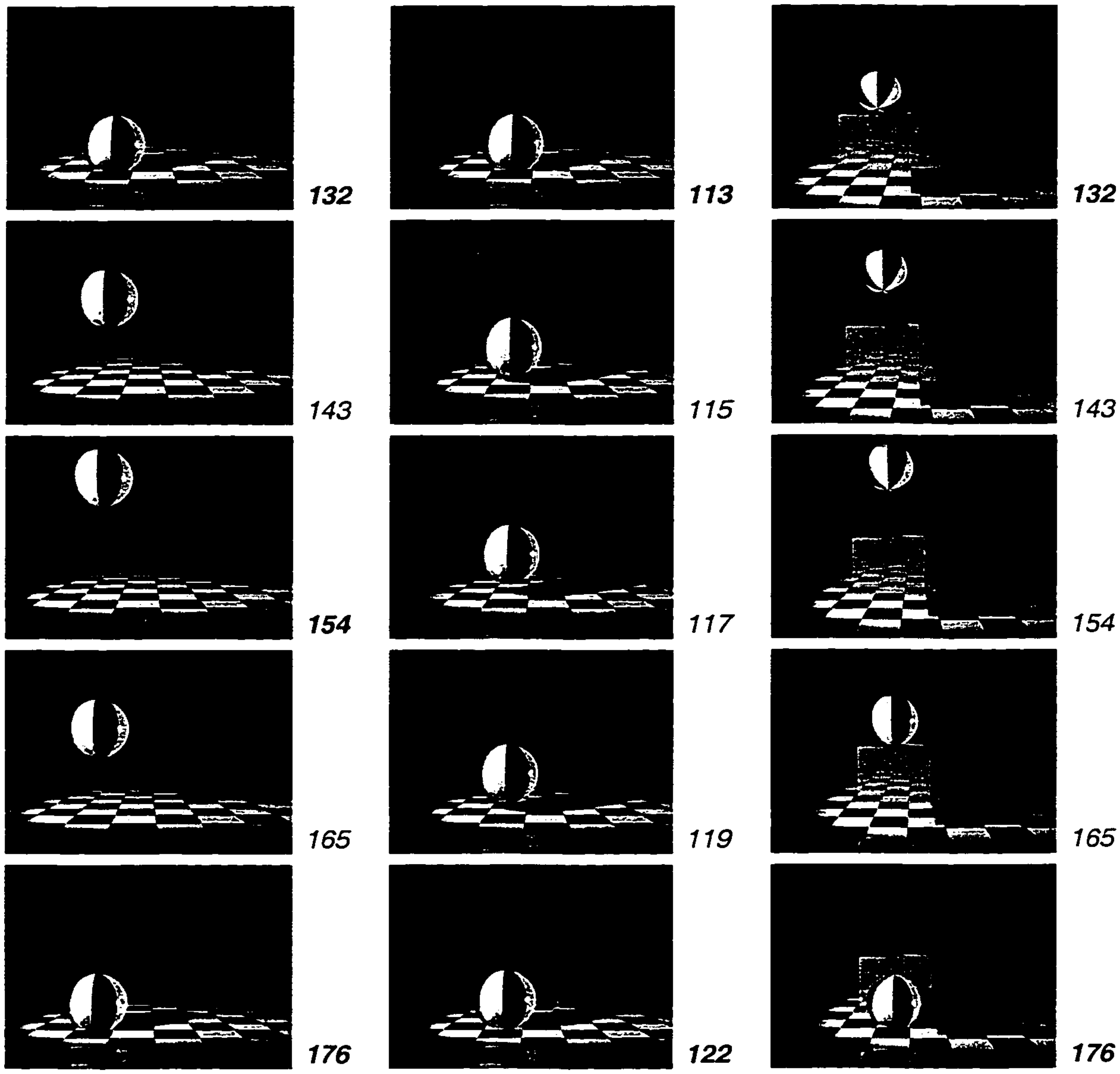


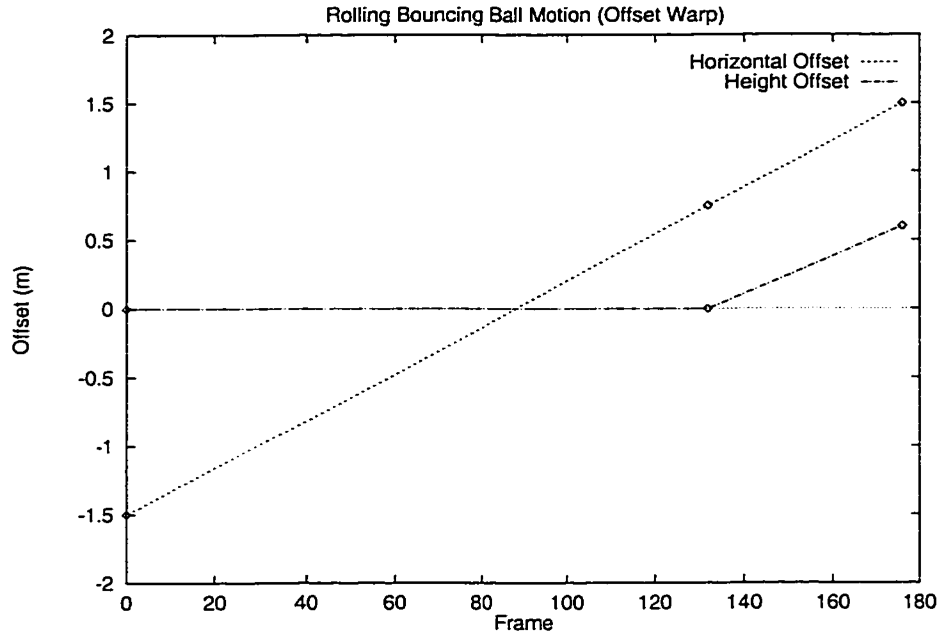
Figure 4.5 (Cont.): **Two examples of motion warping. The first column depicts frames of the original bouncing ball movement. The second features images of the decaying bounce, created using temporal and scaling warps (see Figure 4.3). The third column shows the ball hopping up a series of steps; this effect was accomplished via offset warps (see Figure 4.4).**

4.1.4 Generating a "Running Jump" by Blending

The final experiment of the set involves blending. The goal is to create an animation with the ball rolling along, springing higher and higher, and eventually landing on top of a block. This action is termed a "running jump".

The action begins solely with a rolling motion constructed using an existing animation application. Some rendered frames of this movement are shown in Figure 4.7. To add the springing action, the bouncing motion is gradually introduced into the roll via blending. Before this can occur, however, the bounce must be properly aligned with the roll. This requires displacing the springing ball horizontally to coincide with the position of the rolling ball. In addition, the orientation must be made to match that of the rolling sphere. The offset warps graphed in Figure 4.6 meet these constraints. At the conclusion of the transition, the ball's motion will be entirely defined by the aligned bounce. Frames from this adapted bouncing action are given in Figure 4.7.

The blending formulae (Equations 3.11 and 3.12) are applied with the rolling motion given by $\theta_1(t)$ and the aligned bouncing action by $\theta_2(t)$. Frames from the transitory animation are presented in Figure 4.7. In this example, blending yields an effective and convincing result.



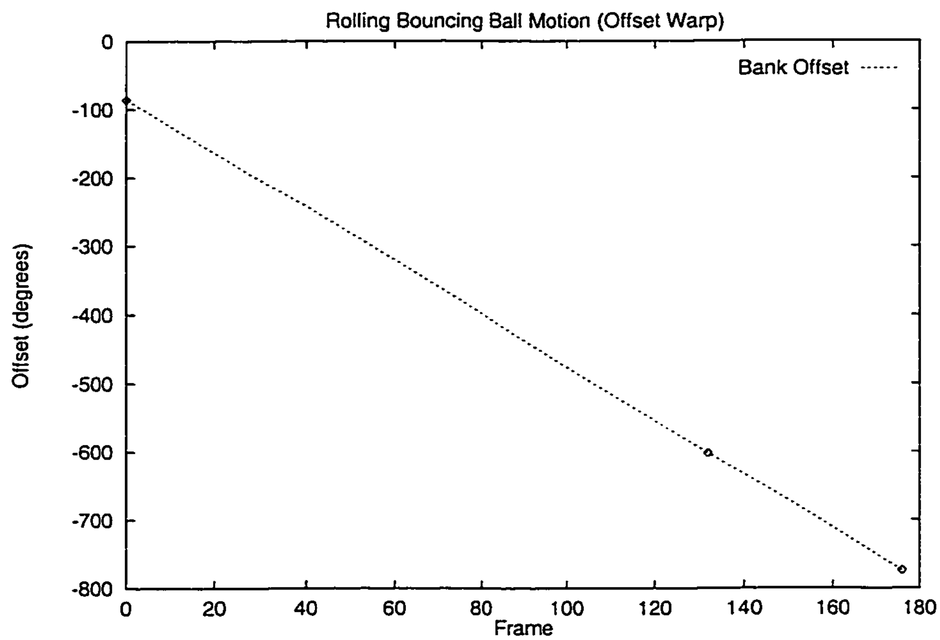


Figure 4.6: **Before a transition can be created through progressive blending, the two constituent movements must first be brought into alignment. To generate a "running jump" onto a block, the original bounce must be aligned with the rolling motion. This entails positioning the bouncing ball over its rolling counterpart and matching the latter's rotation. The above graphs show the offset warping parameters that achieve this goal. Selected frames from the deformed bounce appear in Figure 4.7.**

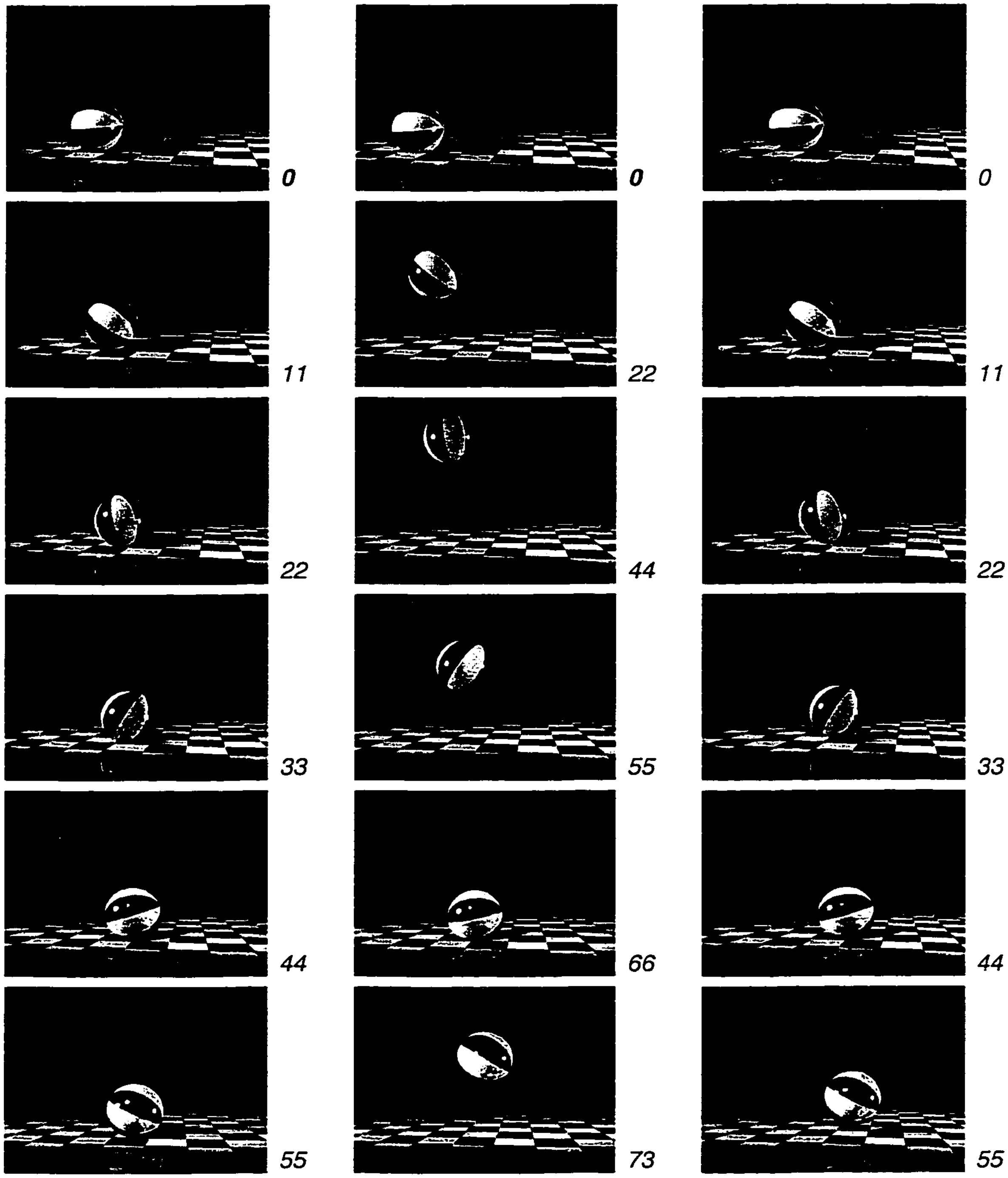


Figure 4.7: **An instance of motion blending to create a "running jump". The left column shows frames from a rolling action. The middle column displays an aligned bouncing motion, derived by warping the original bouncing action. The ball's horizontal position and rotation are matched to that of the rolling ball using the warps illustrated in Figure 4.6. The right column depicts the results of blending from the roll into the aligned bounce, culminating in a jump onto a block.**

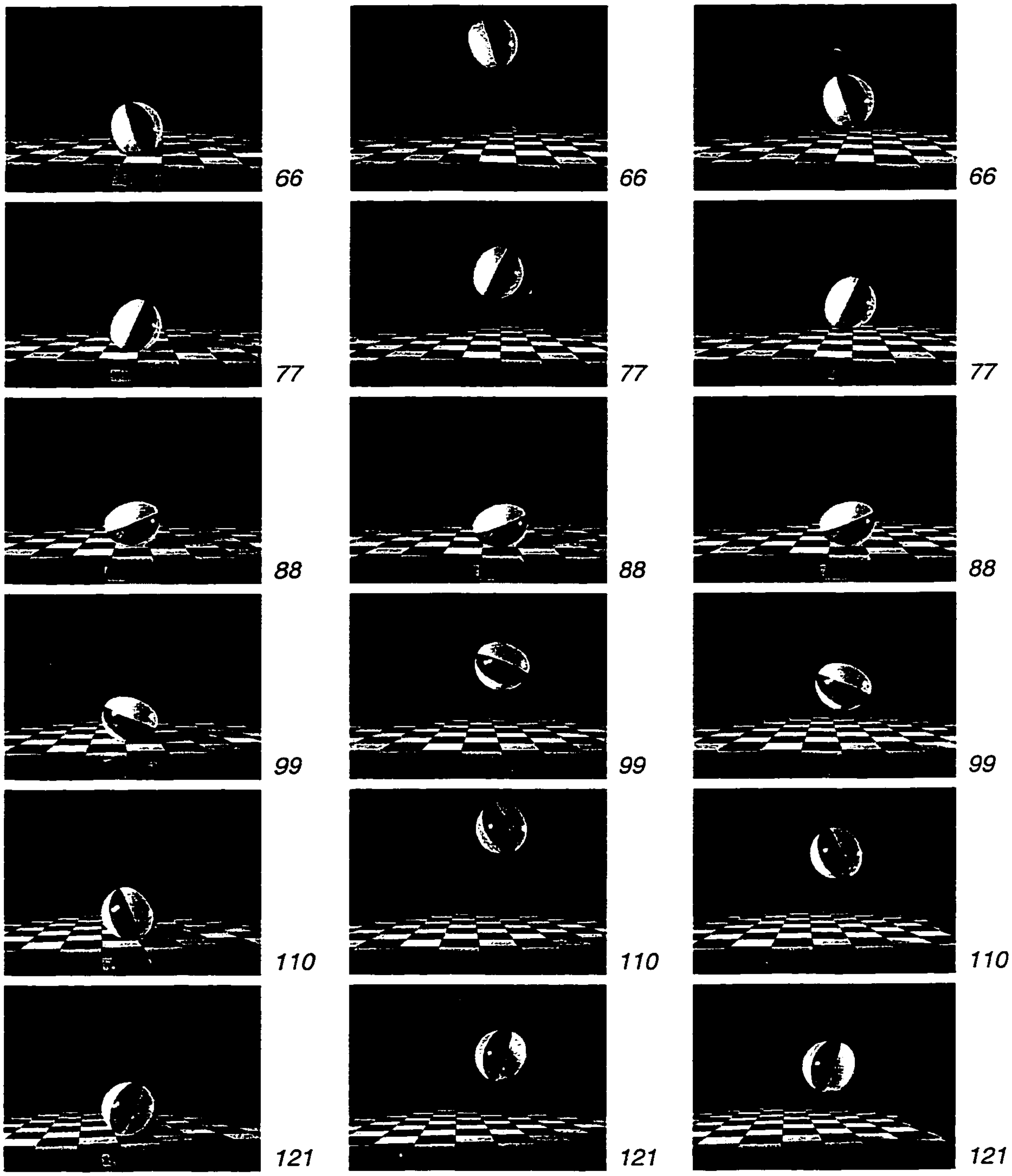


Figure 4.7 (Cont.): **An instance of motion blending to create a "running jump". The left column shows frames from a rolling action. The middle column displays an aligned bouncing motion, derived by warping the original bouncing action. The ball's horizontal position and rotation are matched to that of the rolling ball using the warps illustrated in Figure 4.6. The right column depicts the results of blending from the roll into the aligned bounce, culminating in a jump onto a block.**

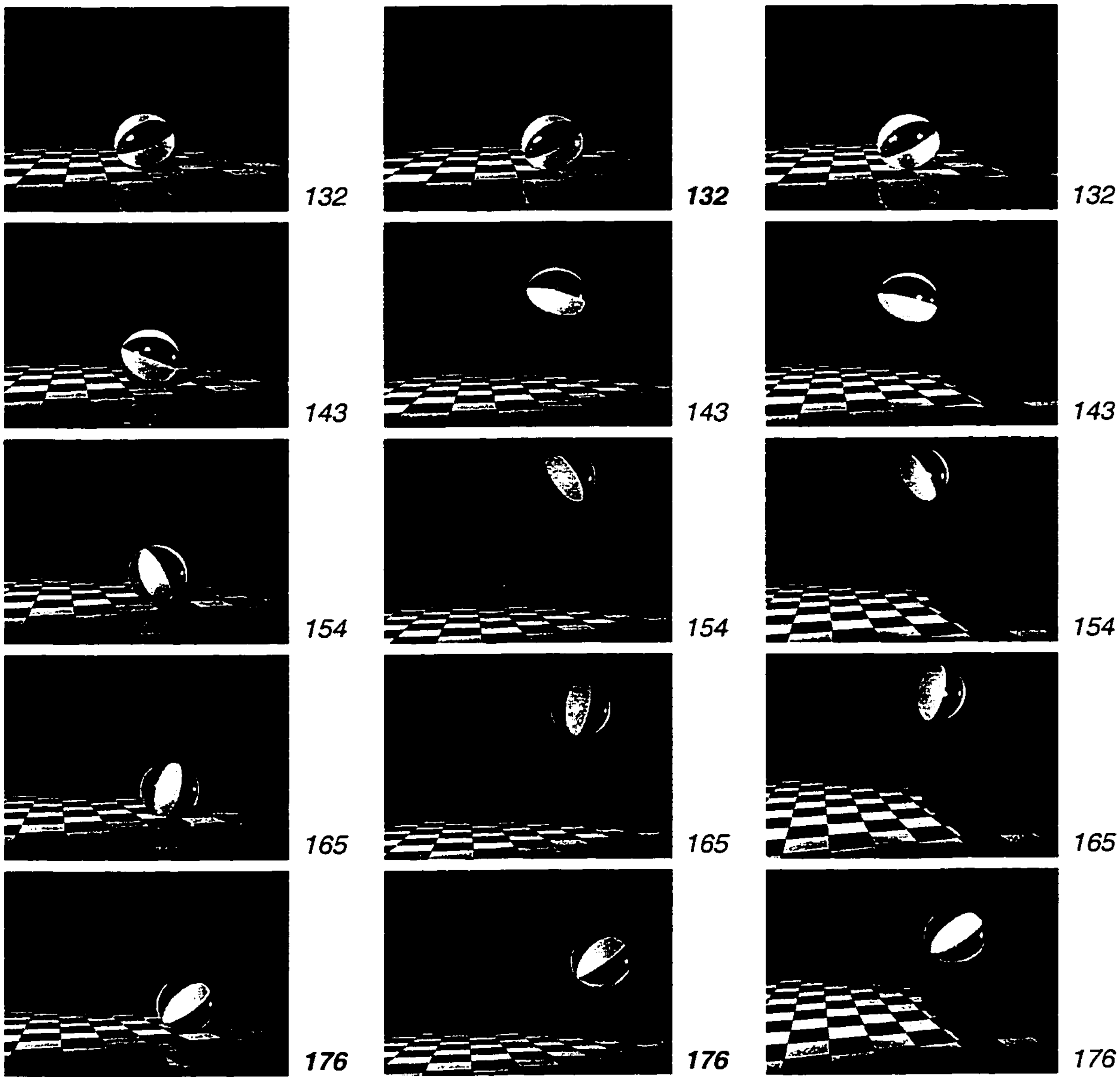


Figure 4.7 (Cont.): **An instance of motion blending to create a "running jump". The left column shows frames from a rolling action. The middle column displays an aligned bouncing motion, derived by warping the original bouncing action. The ball's horizontal position and rotation are matched to that of the rolling ball using the warps illustrated in Figure 4.6. The right column depicts the results of blending from the roll into the aligned bounce, culminating in a jump onto a block.**

4.2 Comparing Offset and Scaling Spatial Warps—The Swivelling Robotic Arm

In their motion warping paper, Witkin and Popović pay little attention to the impact of offset versus scaling warps. The authors, describing the application of these spatial warps, state that "scaling of joint angles is useful for exaggeration, in which case the offset function can be used to set the zero-point around which scaling takes place" [57]. It is not clear, however, what effect scaling will have as opposed to displacement in attaining some augmented goal posture.

The purpose of the following collection of experiments is to study the effects of varying strengths of spatial deformations. The subject of these tests is a three segment, articulated arm.

4.2.1 The Swivelling Arm Motion Template

The swivelling arm is composed of three identical cylindrical segments connected end-to-end. The joints are constrained to revolve around a single axis in these experiments, although their design is sufficiently robust to support rotational freedom in three dimensions. The base of the arm is affixed to the ground.

An original motion for the arm was constructed using LightWave 3D™ animation software. The action is composed of five keyframe poses and has a duration of 150 frames (five seconds). These defining frames are illustrated in Figure 4.8. At the beginning of its action, the arm lies horizontally flat upon the ground. Then, it commences to swing leftward and oscillate. Further frames from the swinging animation are provided in Figure 4.10.

4.2.2 Goals for Reshaping the Swivelling Arm Movement

The initial swivelling movement forms the basis of a new motion differentiated by an altered pose partway through its sequence. The new action is defined as having the same start and ending postures. Three seconds into the action, however, the arm takes a different pose. Figure 4.8 shows these keyframe constraints.

To provide a subject for comparison and a goal for which to strive, the swivelling action is altered by hand to meet these constraints. Concurrently, the animator strives to retain as much of the general appearance of the source movement as possible. This process is time-consuming for the user, involving much trial-and-error when adjusting the keyframes. The animation is viewed to obtain of global perspective of the effects and the keyframe poses refined as necessary. Figure 4.10 shows clips of the hand-customized animation that meets the new constraints.

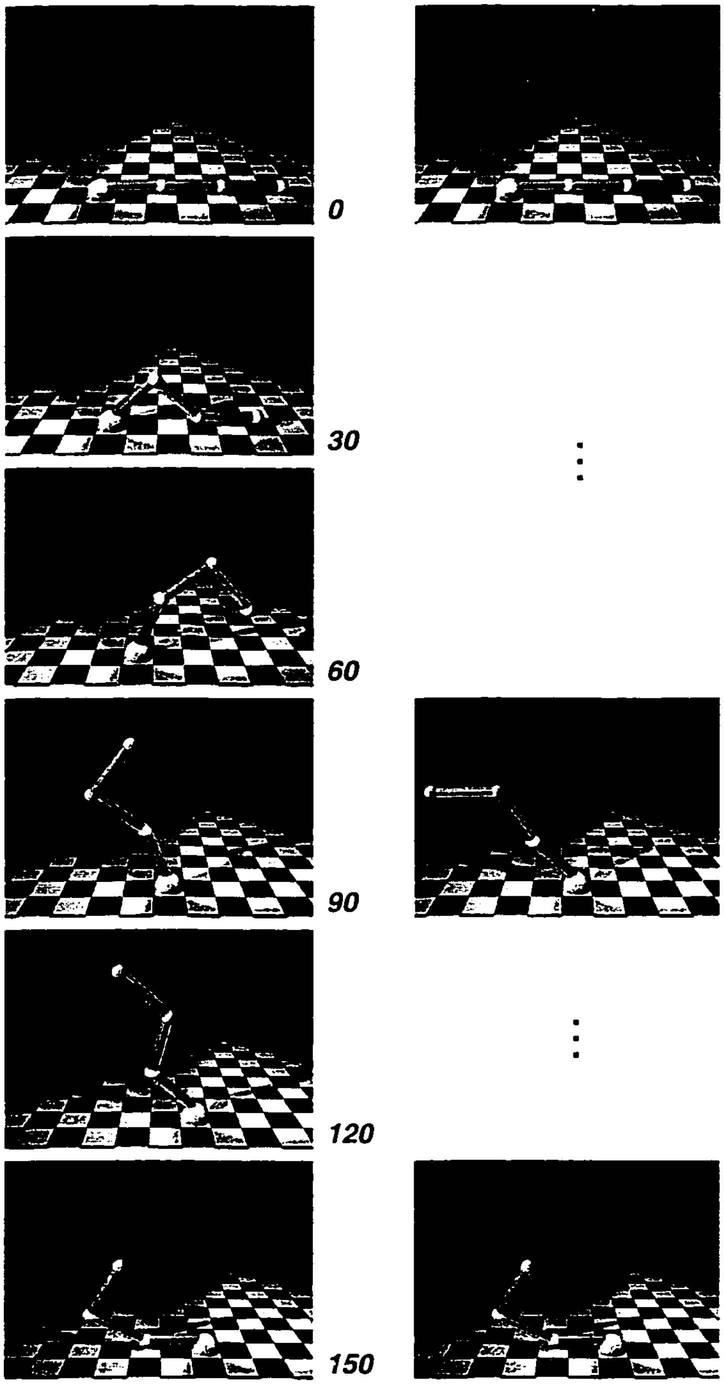


Figure 4.8: **The keyframe poses defining the original and deformed movements of an articulated arm. The left column shows the keyframes of the initial motion of the arm. CTB splines are used to interpolate the intermediate frame postures. The right column shows the required poses of the modified movement.**

This sample, created by direct manipulation of the original swivelling motion, is by no means the sole correct result. Rather, it serves as a specimen to try to duplicate using the motion deformation technique. Several tests are conducted involving various weights of scaling and offset warps; the goal is to try to find a combination that yields a similar result to the hand-modified version.

4.2.3 Various Mixtures of Scaling and Offset Deformations that Realize the Goals

The first experiment to reshape the swivelling action involves solely a scaling warp. There is a noteworthy limitation of scaling warps that does not effect displacement warps. At times when the motion parameter is zero, no amount of scaling can affect its value. In contrast, since offsets are expressed by addition rather than multiplication, the displacement function can influence the parameter's value even when at zero. Fortunately, in this example the source motion does not contain any zeroed joint angles at the critical goal pose at frame ninety.

In defining the spatial warp, a philosophy of minimizing the number of restrictions in the deformation function is adopted. Embracing this intent, the scaling warp is defined with just three keyframes, one for each of the goal constraints. At the start and end times, the arm's joints are scaled with a neutral value of one. At frame ninety, the scaling values are chosen to ensure that the modified goal pose is achieved. Figure 4.9 features the graph of the scaling splines.

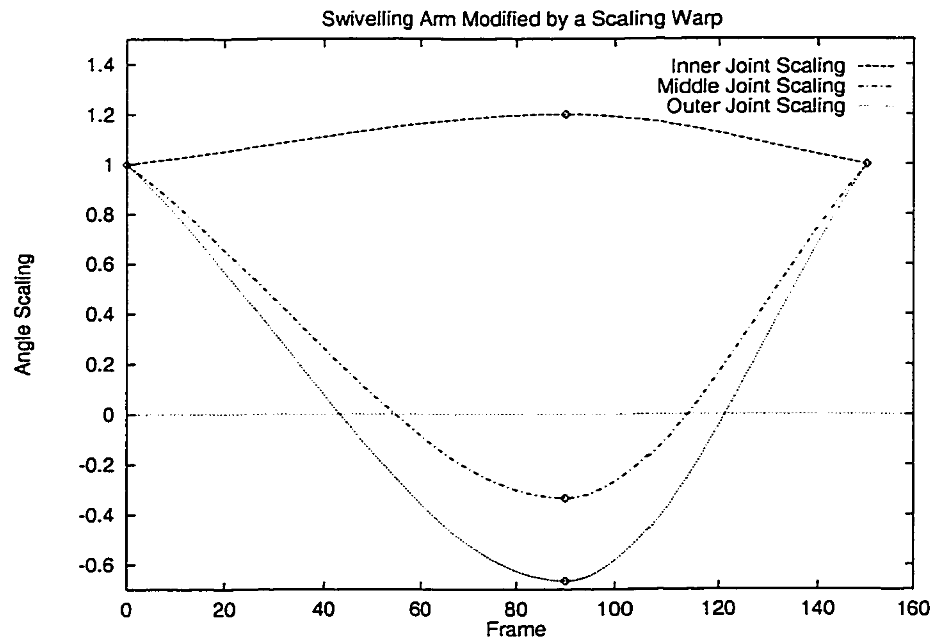


Figure 4.9: **Graph of the scaling warp for reshaping a swivelling arm motion to meet the new keyframe goal poses. The beginning and concluding poses remain the same, whilst at frame ninety the joint angle's of the arm are scaled to achieve a modified posture. Images from the animation resulting from the application of this warp are shown in Figure 4.10.**

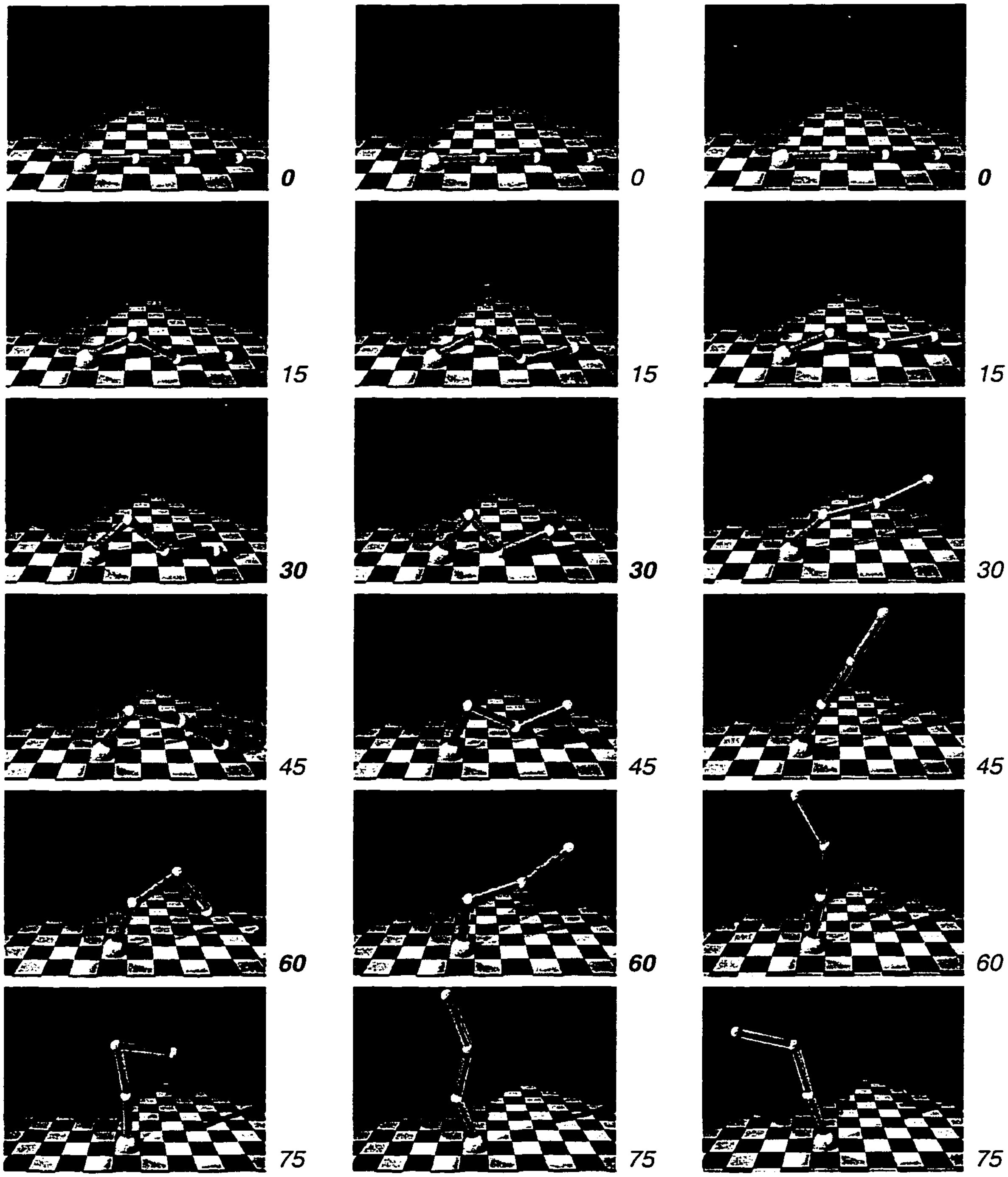


Figure 4.10: **Images from the motion of an articulated arm altered to meet some specific keyframe goal postures. The first column depicts frames from the original movement. The corresponding frame number is listed beside each image; bold values identify the keyframe poses. The second column is a hand-customized version of the action. An attempt was made to retain the basic qualities of the original swivelling action whilst achieving the required goal poses. The third column has been adjusted in an automated fashion using a scaling spatial warp (graphed in Figure 4.9).**

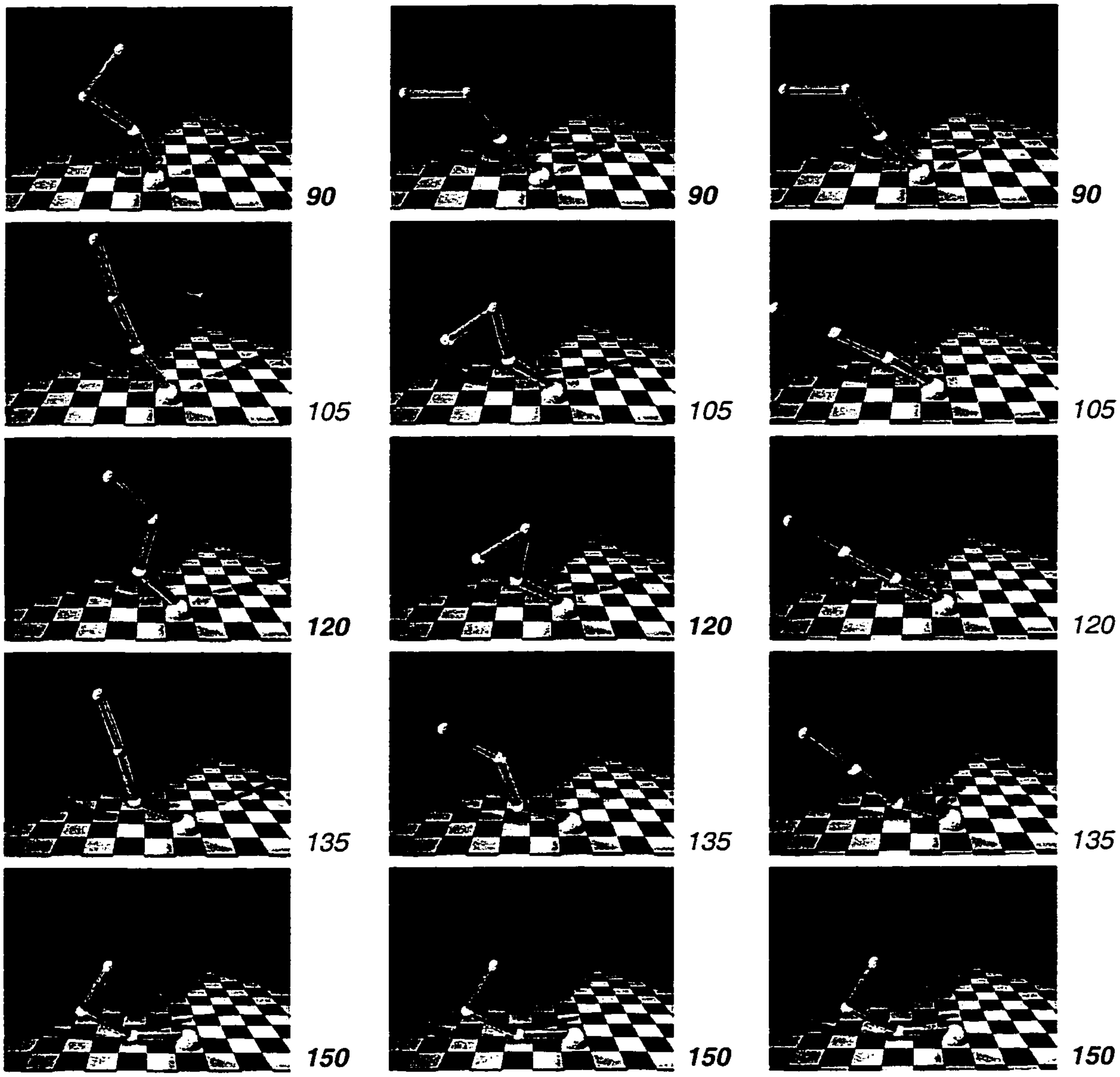


Figure 4.10 (Cont.): **Images from the motion of an articulated arm altered to meet some specific keyframe goal postures. The first column depicts frames from the original movement. The corresponding frame number is listed beside each image; bold values identify the keyframe poses. The second column is a hand-customized version of the action. An attempt was made to retain the basic qualities of the original swivelling action whilst achieving the required goal poses. The third column has been adjusted in an automated fashion using a scaling spatial warp (graphed in Figure 4.9).**

Clips from the scaled action are shown in Figure 4.10. Comparing these to the equivalent frames from the hand-altered action, the visual discrepancies are significant. The movement of the middle and outermost joints when the scaling factor is less than one appear stiff; with fractional weights, joints tend towards their default orientation. The arm becomes straight when its joints are at their neutral angle of zero. This explains the overall rigidity of the two outermost parts of the arm, particularly near frames fifty and one hundred fifteen where the scaling magnitude is closest to zero. Meanwhile, the magnitude of the joint angle at the base remains close to one. Thus, there is little perceptible change in the action of the innermost segment.

The antithesis to the previous test employs only an offset warp to meet the modified goal pose constraints. Again, the only keys are at the starting and concluding frames, and at the critical ninetieth frame. The offset is zero at the end times, ensuring no change in arm posture. The displacements at frame ninety ensure that the required intermediate pose is attained. A graph of the displacement warps for the three joints is given in Figure 4.11.

Because the offset warps are not affected by singularities, the arm does not suffer from the "stiffness" of the scaled version. The motion comes very close to that of the hand-altered, benchmark sample, as the clips given in Figure 4.14 demonstrate. This evidence suggests that when large-scale modifications are required, as is the case at frame ninety in this example, offset warps produce better results than scaling warps.

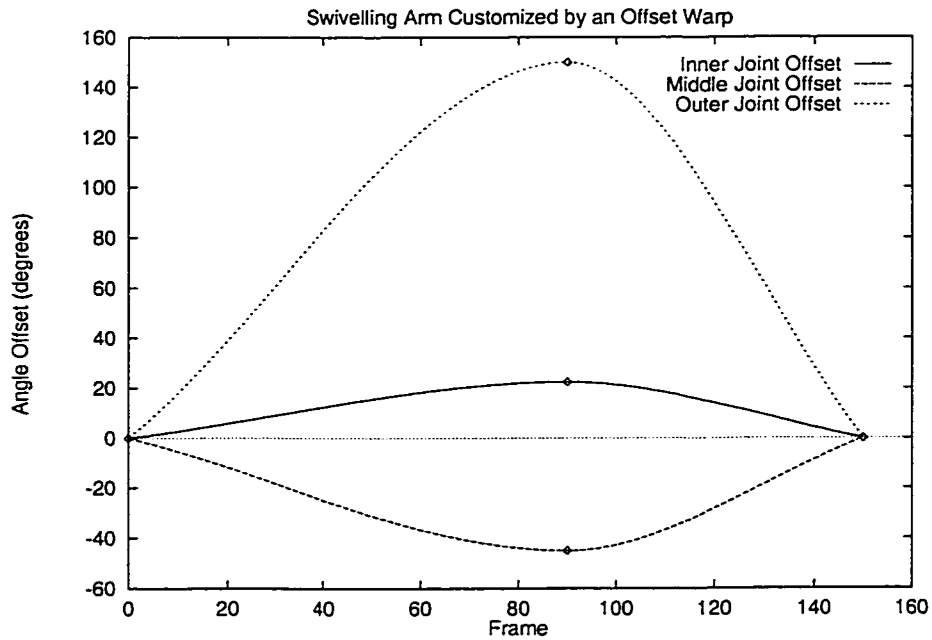


Figure 4.11: **Graph for reshaping the motion of a robotic arm solely by an offset warp. At the start and end points, the arm's poses are unaltered. At frame ninety, an offset is introduced to the joints so that the desired new posture is attained. Figure 4.14 shows frames from the resulting animation.**

Although this example almost mirrors the movements of the hand-customized swing, at times the arm appears to flex too much. Such is the case at frames thirty and one hundred twenty. By combining both forms of spatial warps, this weakness may be corrected.

The third experiment in this set involves scaling and offset functions that each contribute half of the joint angle changes. As usual, the first and last keyframes of the warping functions take their neutral values of zero for the displacement and one for the scaling. At frame ninety, the difference between the original and desired joint angles is found. Half this difference is realized via the scaling warp, the other half by the offset function. Figure 4.12 presents the graphs of the two spatial warps.

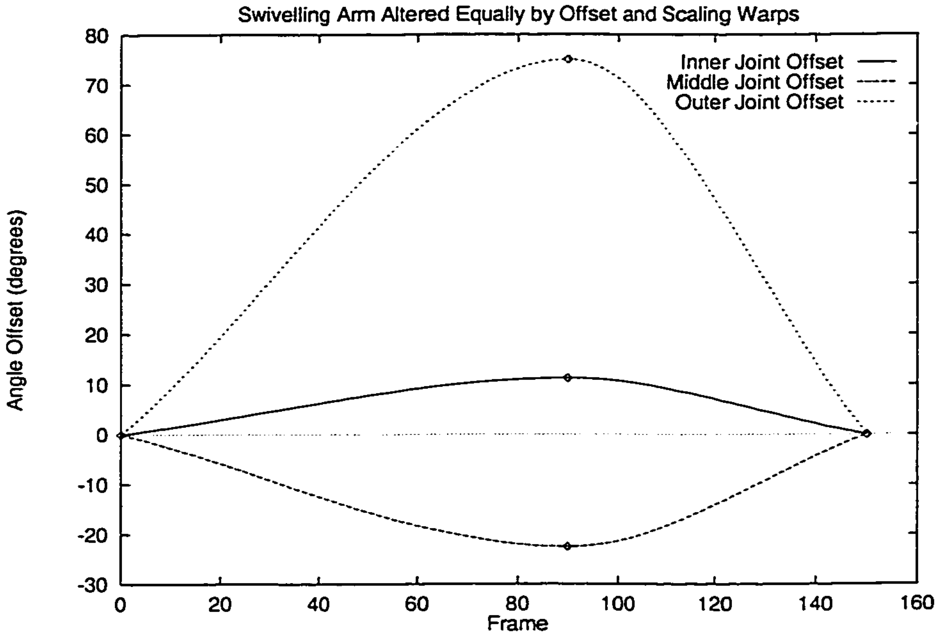
Not surprisingly, the resultant action appears as a fifty-fifty cross between the earlier exclusive scaling and offset tests. Selected frames of this motion are given in Figure 4.14. Although this particular example yielded an inferior outcome as compared to the sole offset warped version, it does demonstrate the potential of using a mixture of spatial warps. Combining both scaling and displacing functions affords an additional degree of flexibility, giving the animator a better opportunity to create his envisioned motion.

The final experiment in this series used some drastic deformations. The arm is moved into a neutral position—lying horizontally straight with its tip pointing towards the right—at frame ninety using the offset function. Then the scaling warp compensates to move the arm into its required pose. These spatial warps are graphed in Figure 4.13.

Clips from the animation produced from these warps are shown in Figure 4.14. Comparing these to both the source and hand-modified animations, it is apparent that this warp has adverse effects. The new motion bares little resemblance to its original or desired action. This examples serves to show that, although the key constraint poses are satisfied, an extreme warp can distort a motion beyond recognition.

As the above series of experiments has demonstrated, customizing a motion by direct means is a time-consuming process of trial and error. As modifications are made to its keyframes, the animation is played back to see the visual consequences. This task requires much thought and the skills of an experienced animator.

Motion warping helps to automate the process of customizing a motion. An animator with little experience can now quickly produce quality results. In the robotic arm experiments, the hand-tailored swivelling action can be duplicated by using an offset warp with some additional scaling modifications. The process involves specifying the constraint poses and then testing different weights of offset and scaling functions to reshape the movement.



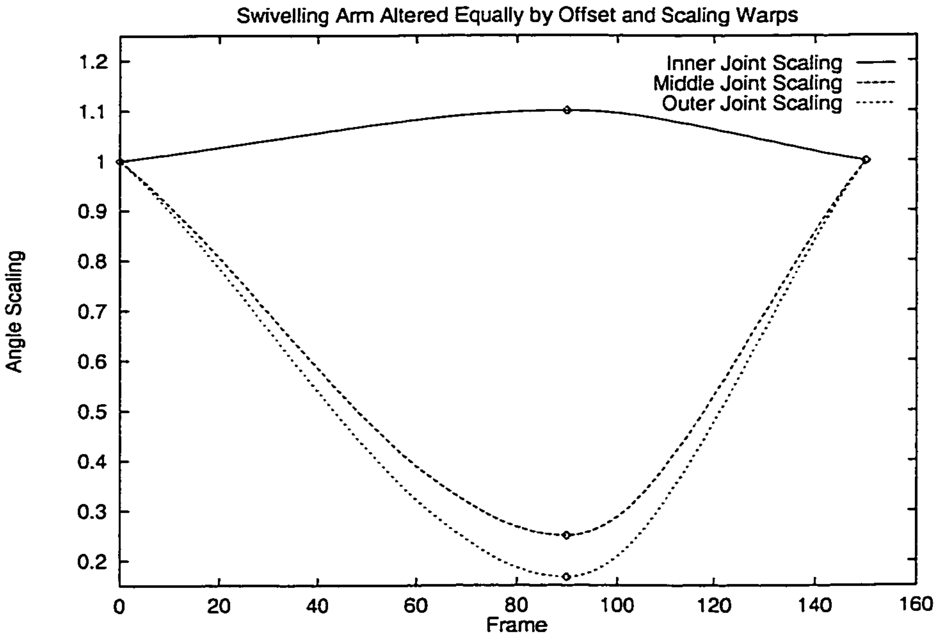


Figure 4.12: **Graphs for adjusting the motion of a swivelling arm equally by offset and scaling warps. To reach the specified goal posture at frame ninety, half the angular change of a joint is contributed by the offset warp (top graph) and half by the scaling warp (bottom graph). The start and end poses remain unchanged. Figure 4.14 depicts images from the resulting animation.**

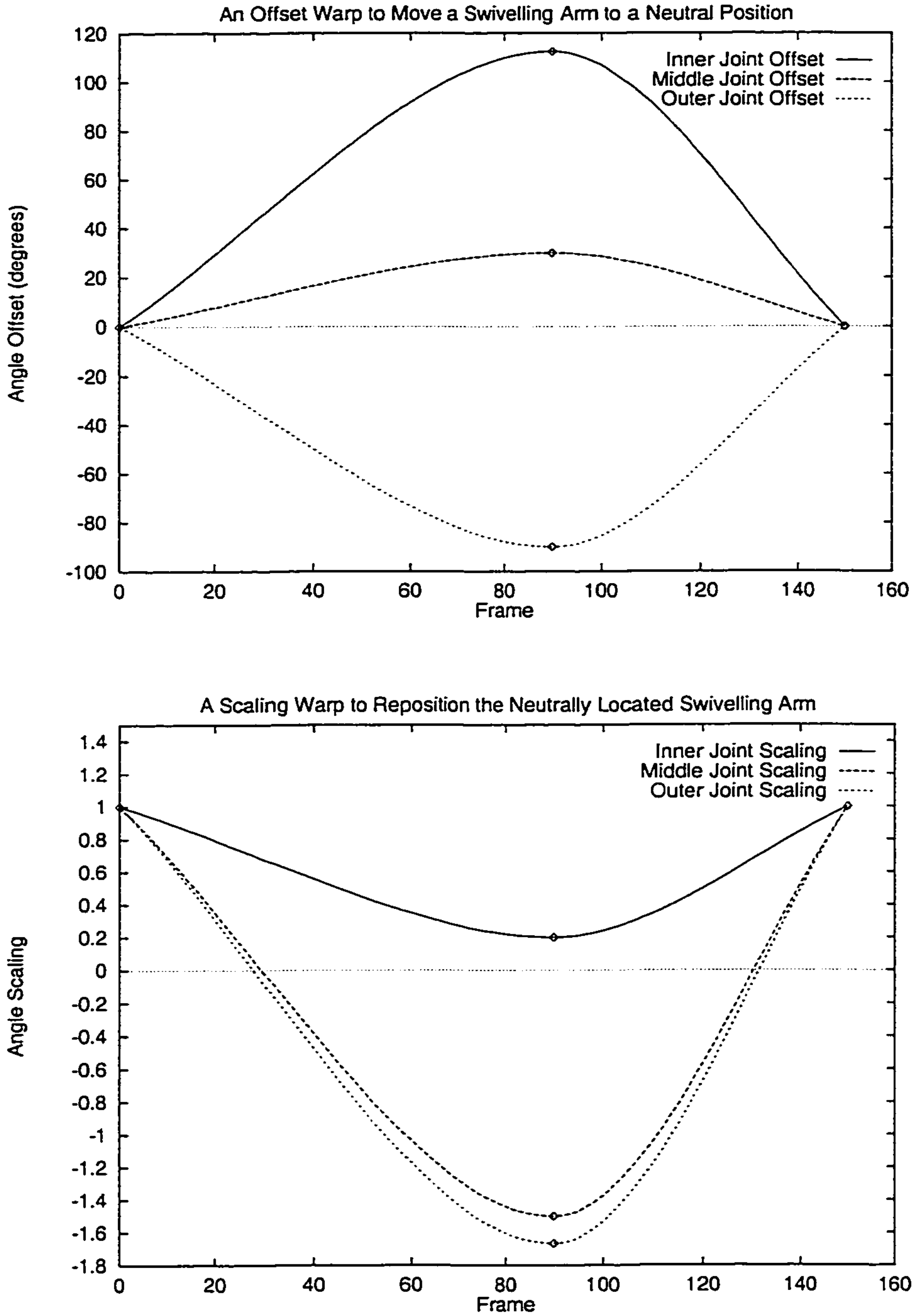


Figure 4.13: **Graphs of spatial warps to neutralize and reposition a robotic arm via offset and scaling warps, respectively. The arm must achieve a specified goal posture at frame ninety, but the starting and end pose must remain unchanged. At this middle frame, the arm's position is nullified to lie horizontally on the ground by an offset warp (top graph). A scaling warp then repositions the arm in its final pose (bottom graph). Figure 4.14 contains frames from the generated animation.**

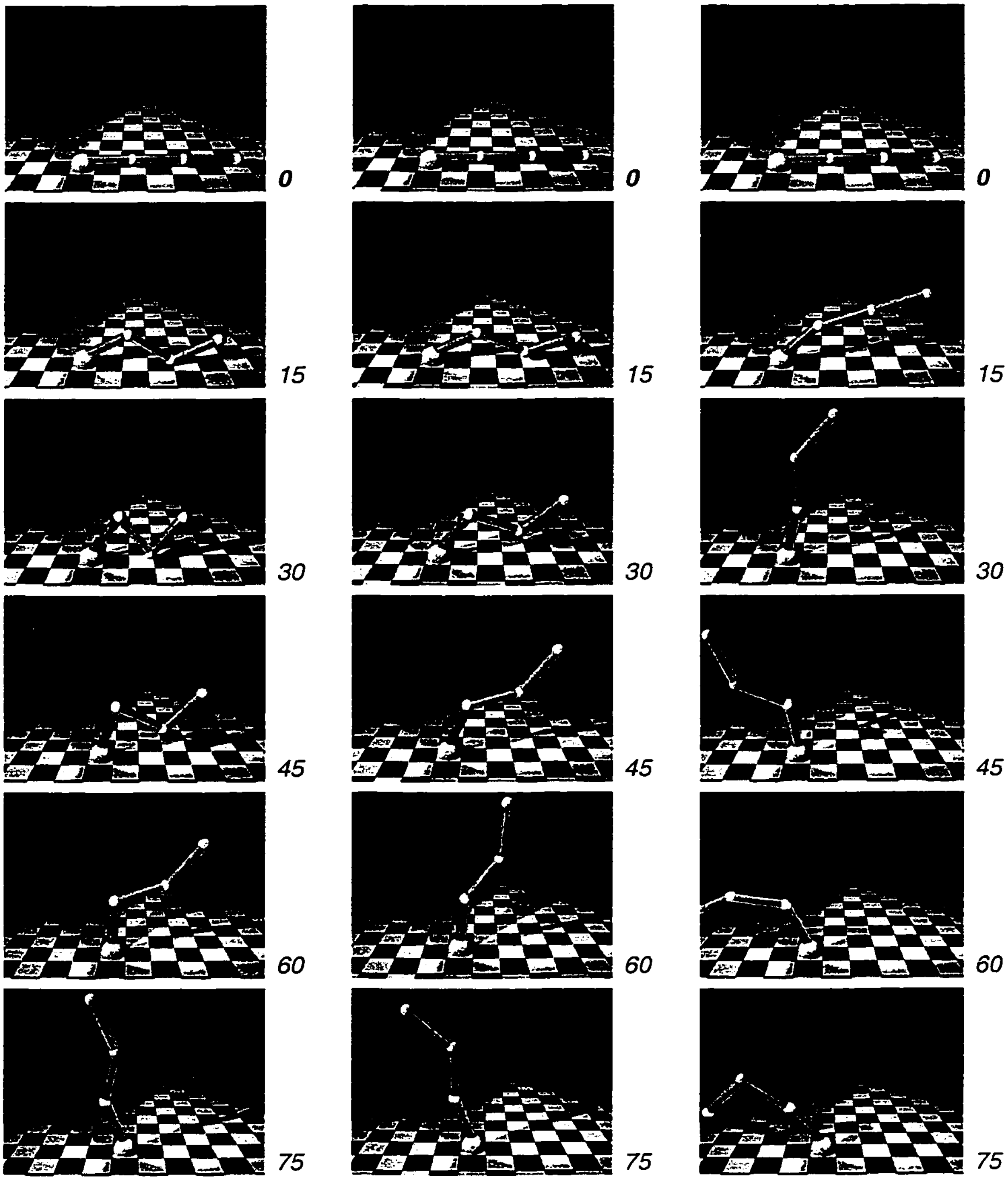


Figure 4.14: **Images from three possible spatial warps of an articulated arm that satisfy some goal postures. The left column shows the resulting frames when only an offset warp (see Figure 4.11) is applied to the swivelling movement. The middle column is the result of a balanced mixture scaling and offset warps (see Figure 4.12). In the right column, the motion was nullified by moving the arm to a neutral position—lying horizontally on the ground—via an offset warp. Then, a scaling warp was applied to reposition the arm in its goal pose (see Figure 4.13).**

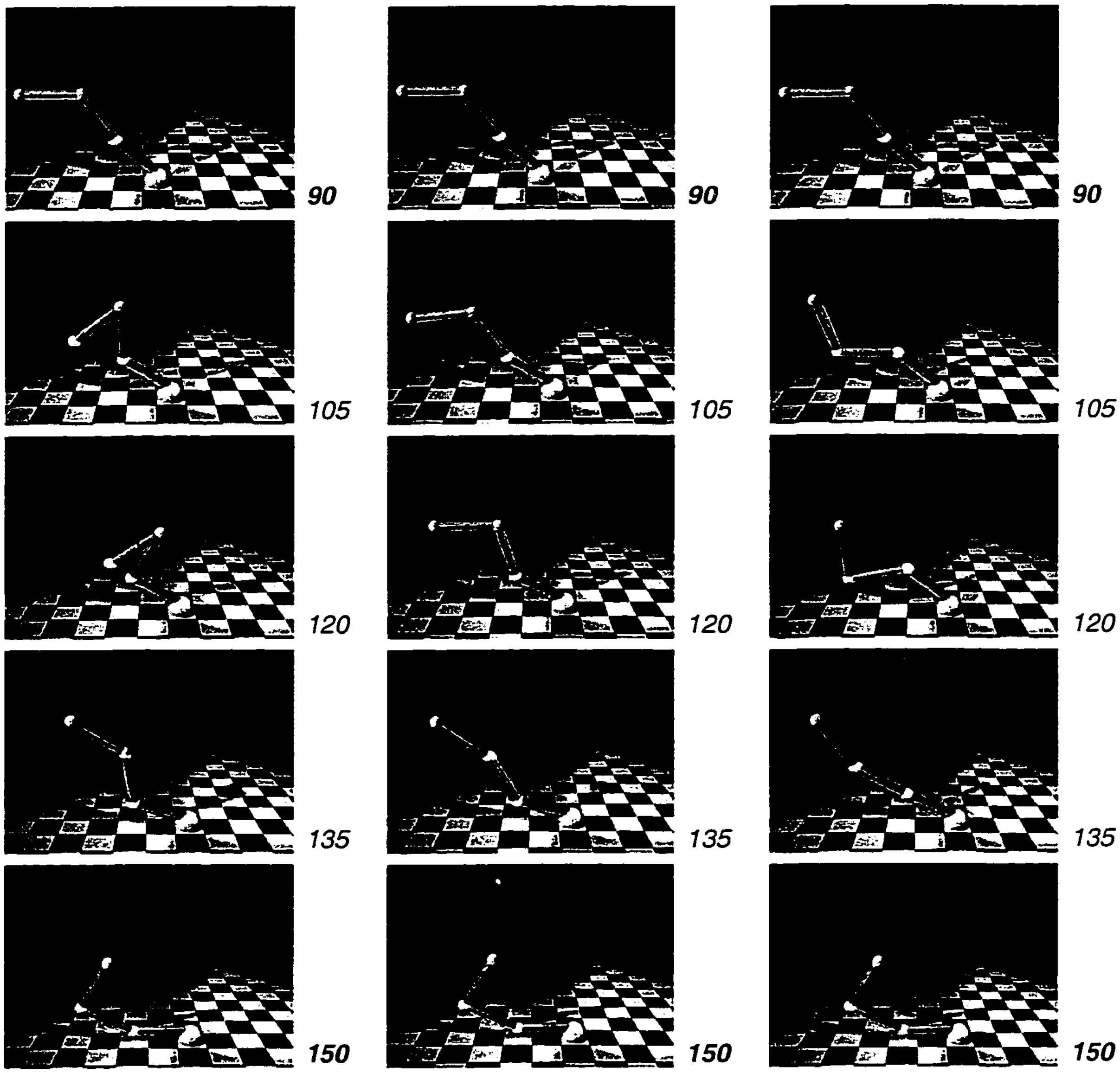


Figure 4.14 (Cont.): **Images from three possible spatial warps of an articulated arm that satisfy some goal postures. The left column shows the resulting frames when only an offset warp (see Figure 4.11) is applied to the swivelling movement. The middle column is the result of a balanced mixture scaling and offset warps (see Figure 4.12). In the right column, the motion was nullified by moving the arm to a neutral position—lying horizontally on the ground—via an offset warp. Then, a scaling warp was applied to reposition the arm in its goal pose (see Figure 4.13).**

Scaling functions are afflicted by a notably limitation—whenever a motion parameter is zero, no amount of scaling can modify its value. Significant deviations between a source pose and the desired goal pose may be better realized by emphasizing the displacement more than the scaling warp.

4.3 The Quadruped Motion Warping System in Action—The Ambling and Bounding Marten

A final series of experiments demonstrate the effectiveness of the automations embedded into the quadruped-specific motion warping system. In preparation, some motions encompassing a single stride of a gait of a marten are designed. The timings of the leg phases of these locomotions are identified. These form the core input parameters to the system.

The two major functions of the system are then tested: locomotion along path and transitions between gaits. For the former, a path over some undulating terrain is constructed. The system is then invoked with the marten's bounding action to make the creature navigate this route. For the transition, a duration is specified along with the starting and concluding gaits, in this case an amble and a bound. The system then constructs an animation such that the marten's movements gradually change between the two source locomotions.

The resulting animations help to illustrate the capabilities and limitations of the quadruped warping architecture. Although the system works as intended, its simplistic nature renders some undesirable effects in the generated animations. Problems include object intersections and unconvincing movements.

4.3.1 The Source Ambling and Bounding Locomotions

A prerequisite to the system is the motion of the stride of a gait. Two kinds of gaits were designed for the marten model: an amble and a bound. Because of the unavailability of specialized equipment and software, these locomotory motions were replicated using traditional keyframing with spline interpolation.

The ambling gait was based on a motion study of a walking cat borrowed from Gray's *Animation Locomotion* [23]. The diagram in Figure 4.15 shows side view of the thirty-eight frame long stride; the hip, shoulder, knee, ankle, and toe positions are marked and connected by line segments. Of course, because the skeletal structure of the marten differs from the cat's, the leg movements need to be adapted to the marten model. Care is taken that the stance feet remain stationary on the ground. Only the legs on one side need defining; these are copied, albeit half a phase out of sequence, to the other side. The creature begins its stride at the origin, moving along the x-axis. Half of the thirty-eight frames of the marten's ambling stride are rendered in Figure 4.16.

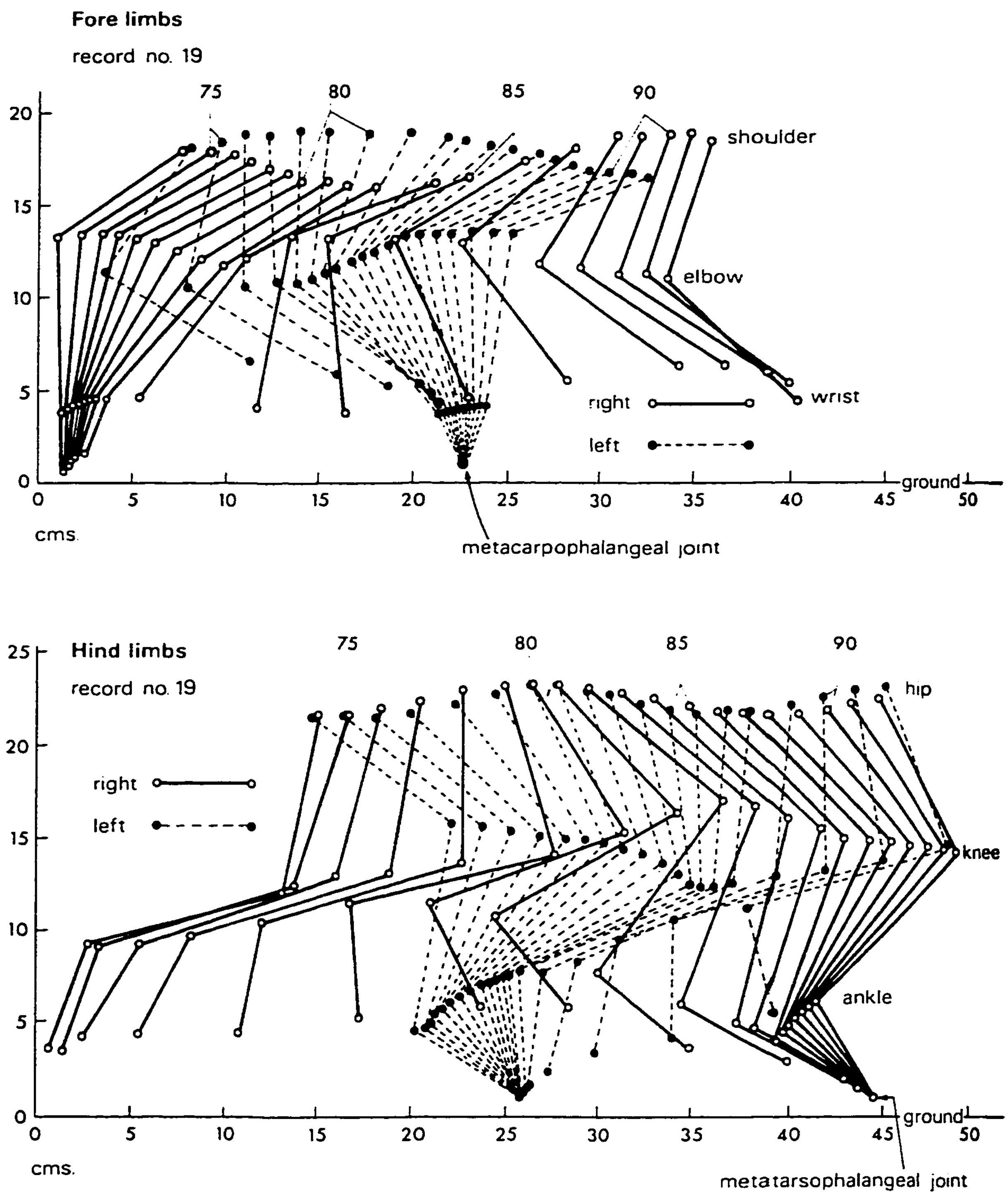


Figure 4.15: **The motions of the limbs of a walking cat (taken from Gray's *Animal Locomotion* [23]). The joints, denoted by circular nodes, are connected by line segments. Samples are spaced $^1/_{38}{}^{th}$ of a second apart. These measurements were used as a basis for the marten's ambling gait shown in Figure 4.16.**

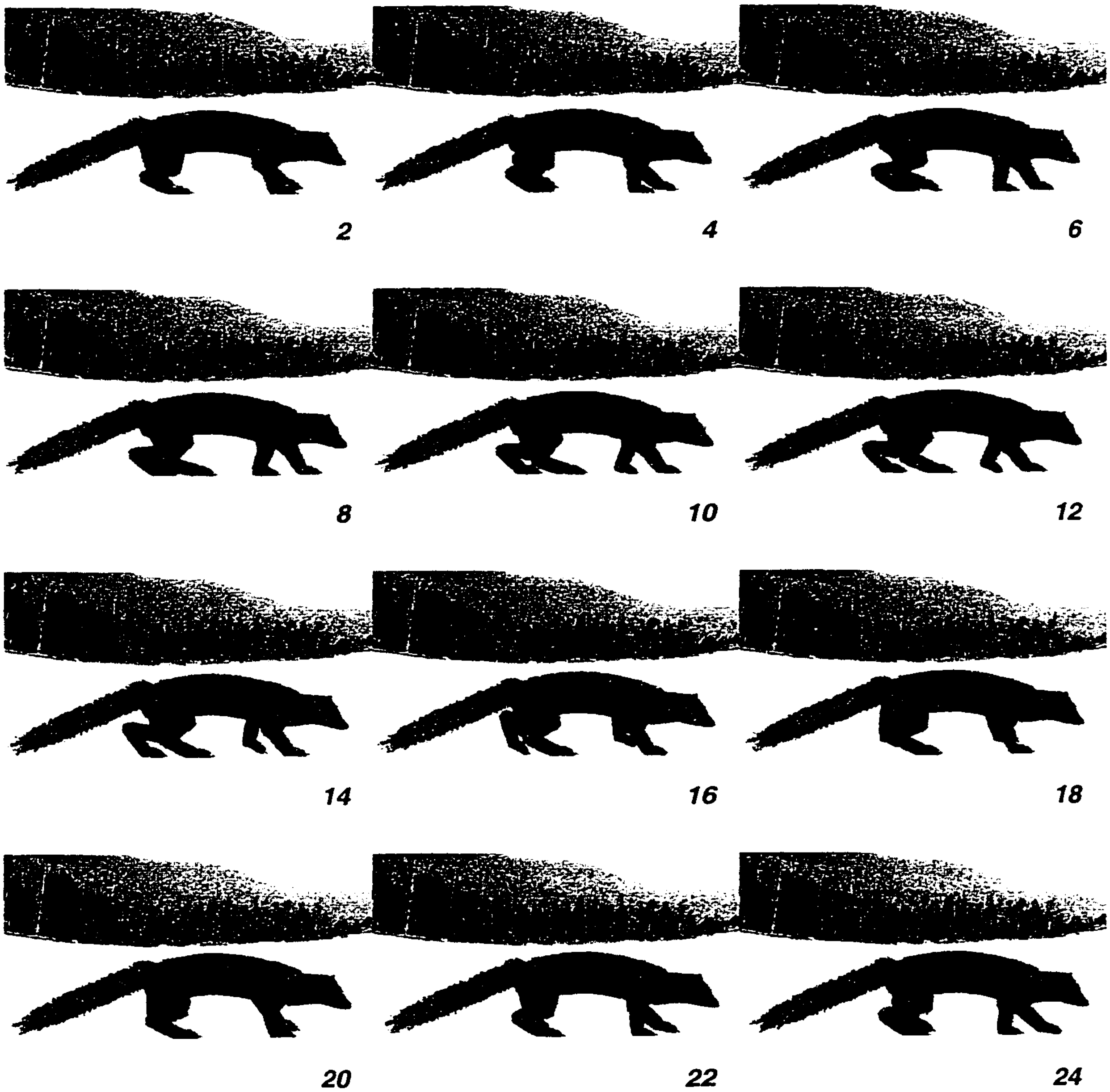


Figure 4.16: **Half of the thirty-eight frames that comprise the marten's ambling action. Keys were created for every frame of the movement, as indicated by the bold frame numbers. These keyframes strictly specify the angles of the body, head, tail, and leg joints.**

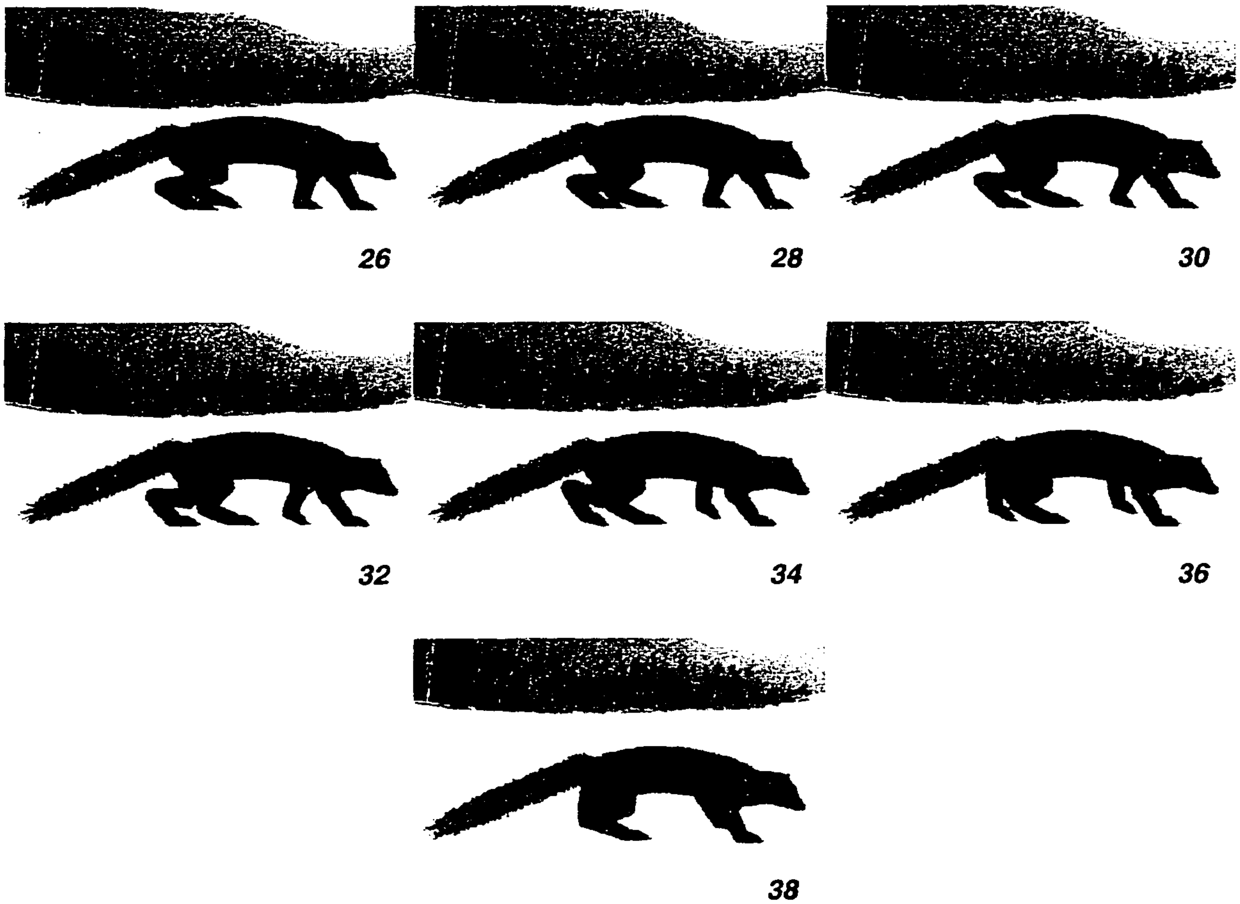


Figure 4.16 (Cont.): **Half of the thirty-eight frames that comprise the marten's ambling action. Keys were created for every frame of the movement, as indicated by the bold frame numbers. These keyframes strictly specify the angles of the body, head, tail, and leg joints.**

Unlike for the amble, quality reference data was unavailable for the bound. Instead, videos of small mammals were studied—including a marten—and an animator copied these motions as best as possible. The thirteen frames composing the bounding action are illustrated in Figure 4.17. Because of the method chosen to create the source gaits, none of these motions are accurate from a physics standpoint. Nevertheless, they look reasonable visually.

After the gait animations are finalized, the state timing information is tabulated. A side view is examined to decide the most probable frames for each foot's touchdown and takeoff. These times identify the stance stages of the legs. The state timings for the amble and bound were given in Figures 3.14 and 3.15 respectively.

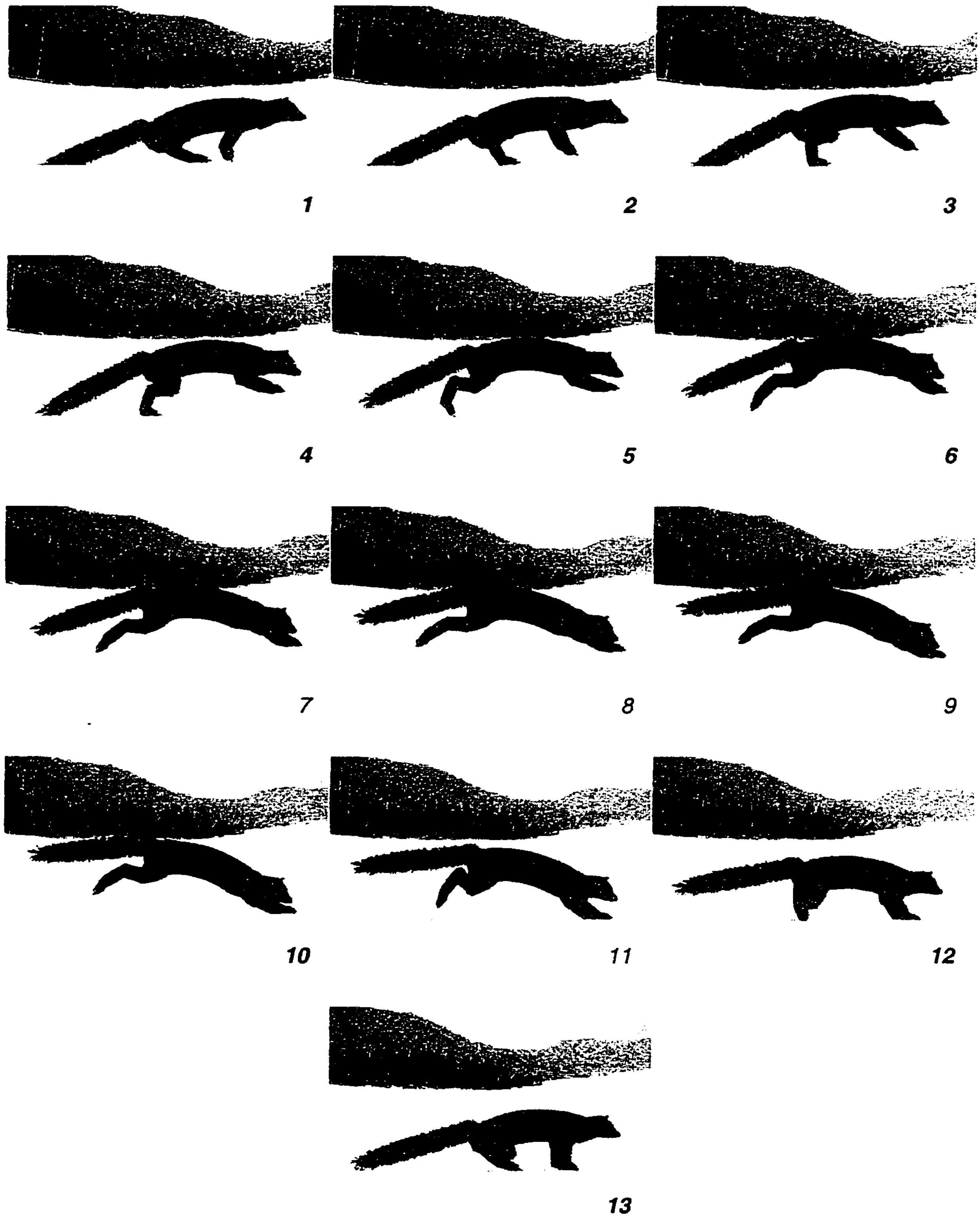


Figure 4.17: **The thirteen frames making up the marten's bounding action. Of these, nine are keyframes; these critical frames, identified by the bold frame numbers, specify the angles of the body, head, tail, and leg joints.**

4.3.2 Application of the Bound Along a Path

Before activating the system to create a locomotion along a path, the environment must first be set up. This involves constructing a geometric model of the terrain and a course running over the surface. In this case, the former is composed of snowy mounds created by vertically distorting a plane.

A path was carefully drawn along the surface of the ground. This route is shown in Figure 4.18. For variety and to fully test the attributes of the system, the path proceeds up and down hills and curves from side-to-side. The path points were distributed in time such that the path speed was roughly equivalent to the bounding gait's average speed. In addition, the spacing of points is fairly tight so that the path traces the hills' surface with reasonable accuracy. This was a time-consuming endeavor that could be made more efficient by automation.

Ideally, the route could be drawn in the two-dimensional x-z plane. Then, it could be automatically projected down onto the ground. One potential drawback is a loss of speed control. A path of constant velocity in two-dimensions, when projected onto the terrain, would undergo speedup as the slope of the hill steepened. Thus, the path would require reparameterization to ensure that the speed remains consistent. Another difficulty lies in interpreting the terrain geometry from a model created in an animation package.

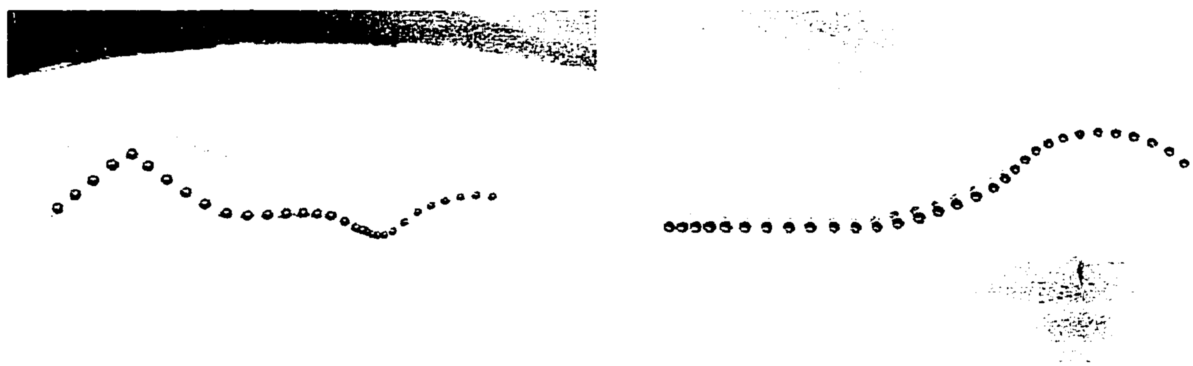


Figure 4.18: **The user-specified path for the marten to follow as it bounds across the hills. The animal will move from the left to the right. The course is designed using splines; the keyframe points are placed at ground level. Figure 4.20 shows some frames from the resulting animation.**

The bounding action is then applied to the path using the system. Selected frames from the resulting animation are featured in Figure 4.20. As expected, the marten effectively employs its bounding action over the span of the designated course. The body follows the contours of the hills, and the feet tend to remain in contact with the ground during their stance stage. Closer examination reveals several deficiencies, most attributable to the simple forms of warping used.

The first notable problem is with the feet intersecting or "floating" above with the ground, although the latter is not so readily apparent. This effect can be seen in frame forty of Figure 4.20. A contributing factor to this shortcoming are the approximations and simplification assumptions used to decide the location and slope of the ground.

The ground's position and slope is assumed to be constant, with the sole measurement taken below the centre of the body. The angle of the surface is represented by the chord passing through two surrounding points on the path curve. If the ground is not consistent, then the supporting legs will intersect it if the surface is concave up, or float above it should the surface be convex up. This effect will also occur if there is any misalignment of the path's curve with the surface topography. In the aforementioned example, the terrain is concave up and the path is likely not perfectly aligned with the surface, thus explaining the leg collision.

The next problem deals with unlikely or impossible directional changes made by the marten. These are visible when there are abrupt changes in the terrain's slope. An illustration of an unconvincing action occurs around frame twenty-one as the marten leaps over the crest of the hill. In this case, the animal leaps upwards and then quickly pitches downwards. A more appropriate behaviour would be for the creature to ease over the summit.

One possible way to rectify this is to smooth the curve of the path around this point. This can be done automatically using a low-pass filter. However, this act reintroduces the previous problem of ground intersections and floating; the route is no longer guaranteed to lie flush atop the topography. Ideally, more robust adjustments need to be made to the motion of the quadruped.

Some effort has been made to compensate for slippage of the stance foot as the heading and pitch of the animal changes. These corrections are, at times, insufficient. Take, for example, the animal turning. The hip of a stance leg rotates in a direction opposite to the body; this action negates the torso's orientation change which would otherwise be propagated through the leg to the foot. However, the leg's position slides with respect to the ground. This problem is illustrated in Figure 4.19. In this illustration, the location of the leg is shown to change as the body is rotated. In the animation from Figure 4.20, slippage can be seen near the end of the sequence, although the effect is not visible in the provided images.

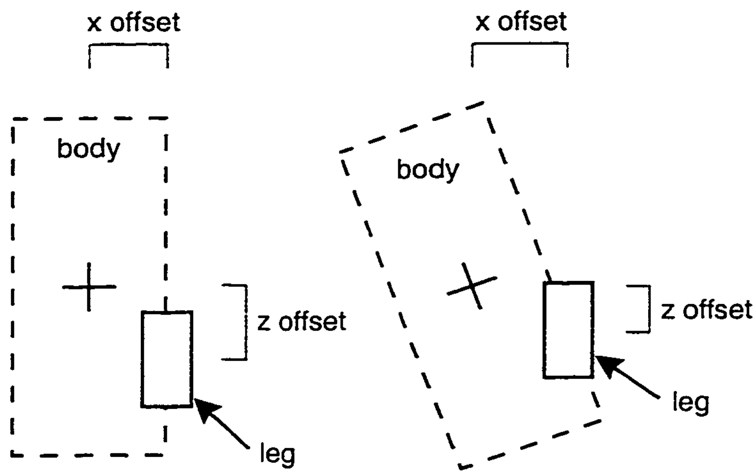


Figure 4.19: **Slippage of the legs occurs as the body rotates during a turn. Whilst the relative offset of the leg with respect to the body remains unchanged, the location of the leg in world coordinates shifts.**

The algorithm for changing the stride length on the basis of the speed considers each leg independently. The synchronization of the phases of the legs is gradually lost as the speed of the animal fluctuates. In the variable terrain animation, care was taken to maintain a path speed equal to the bounding gait's. Nevertheless, some loss of synchronization is apparent by the end of the animation. To amend this deficiency, we need to keep track of cumulative stride lengths for all of the legs. Stride lengths of a leg are biased by this average.

There are several optimizations that can be employed to alleviate some of the aforementioned problems. The first is a better knowledge of ground topology. Rather than relying on the path to delineate the topography, the terrain geometry should be accessed directly. Further, The true positions of feet can be found by applying forward kinematics. With this knowledge, the feet can be placed squarely on the ground, removing the intersection or floating foot problem. Also, sliding can be eliminated by applying additional rotations to the legs to counter the body turning.

There are many more aspects that need to be considered to procure a realistic locomotion. The motion of the entire leg needs to be treated as a whole, whereas motion warping is restricted to individual parameters. For instance, an animal moving down a steep slope will have its hind legs in a crouched position with the forelimbs outstretched to lift the body towards the horizontal. Our system does not consider these forms of complex limb adjustments. To do so would require some form of multiparameter warping, perhaps involving inverse kinematics. However, such a technique might be subject to high frequency artifacts [22].

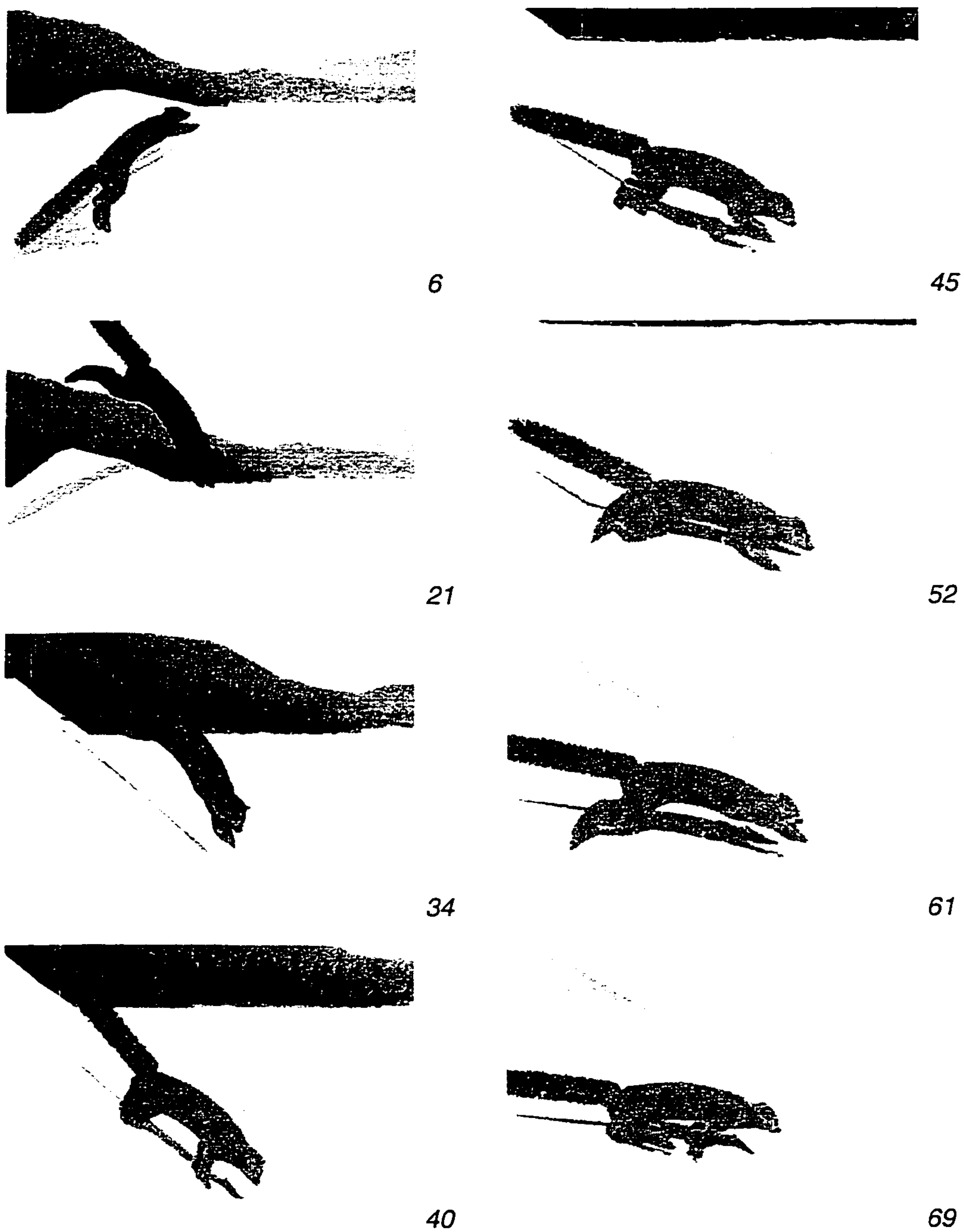


Figure 4.20: **Some images taken from the animation of the marten bounding atop some hills along a user-defined path. This path is shown in Figure 4.18. The centre of the torso is positioned directly above the animal's point along the path. The body is rotated to be aligned with the path slope. Adjustments to the hip and ankle joints are made to ensure the stance feet remain affixed to the ground.**

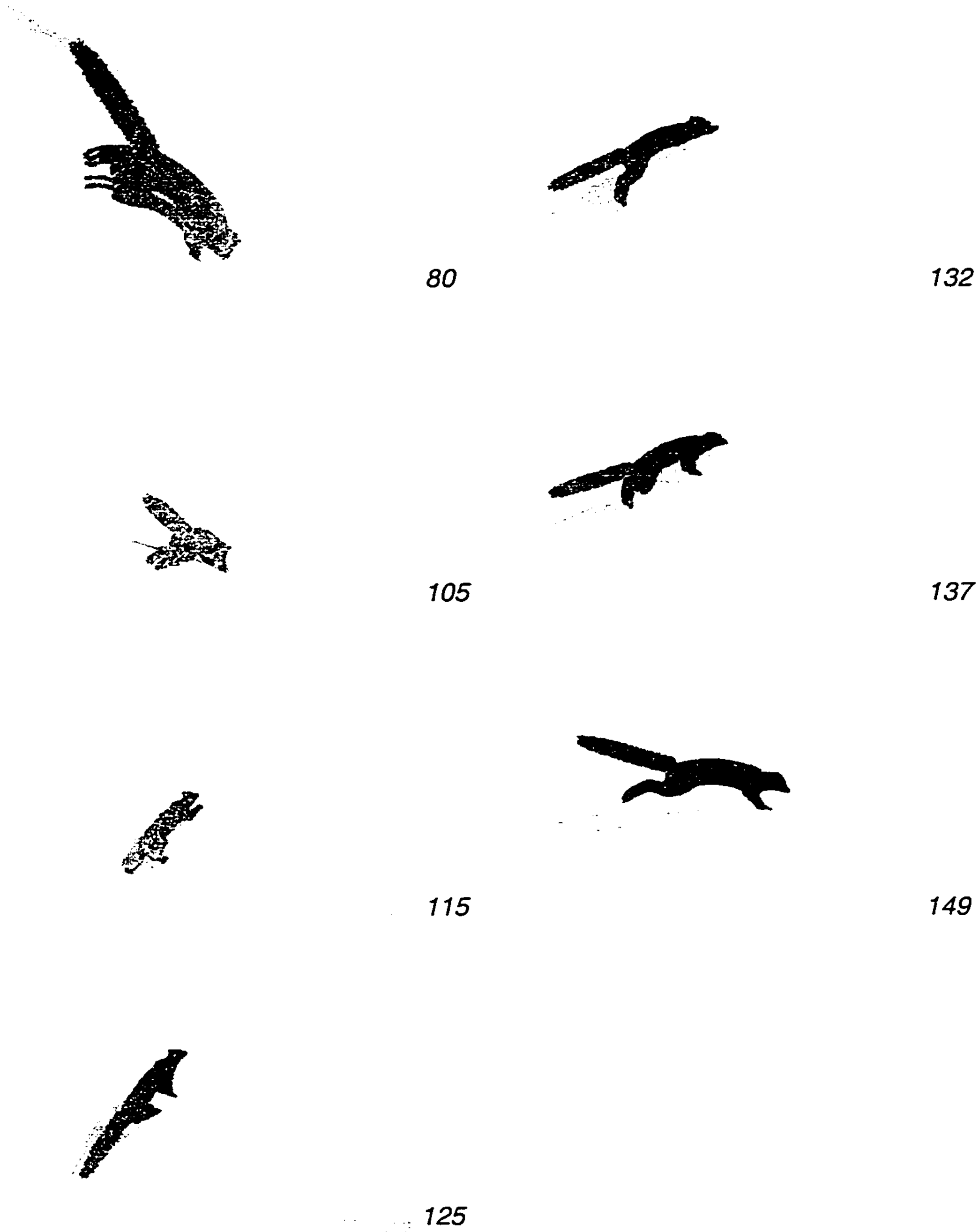


Figure 4.20 (Cont.): **Some images taken from the animation of the marten bounding atop some hills along a user-defined path. This path is shown in Figure 4.18. The centre of the torso is positioned directly above the animal's point along the path. The body is rotated to be aligned with the path slope. Adjustments to the hip and ankle joints are made to ensure the stance feet remain affixed to the ground.**

4.3.3 Transition from the Amble to the Bound

The final test involves the creation of a transition between two gaits—from an amble to a bound. This process involves two steps:

1. Two intermediate sequences are generated, one for each of the contributing gaits. The leg timings of the two sequences are equivalent. There is a progressive shift from the timings of the ambling gait to those of the bounding gait. This involves scaling and shifting the leg phases over the duration of the transition. The actual motions, in a spatial sense, remain untouched. The animal begins its transition moving at the speed of the amble, and accelerates constantly until the bounding speed is attained. Figure 4.21 shows frames from these intermediary animations.

2. The two intermediate sequences are merged, starting with the ambling version and progressively blending into the bounding version. Images from the complete transition are given in Figure 4.21.

One of the difficulties of building a transition is to decide on the orientation of the body for each frame. In the intermediate animations, the legs are out of phase with respect to each other. Thus, the torso orientation associated with the pose of each leg differs. In the system, an average of the body orientations of all four legs is used. This is not an ideal solution because it contributes to foot collisions with the ground. This effect can be seen in some of the intermediate sequence frames shown in Figure 4.21. For instance, in frame thirty of the bound, the foot of one of the rear legs intersects the ground.

Upon consideration, several unconvincing aspects can be noted in the transitory animation. One characteristic is that the rear legs appear gimpy, literally dragging behind the animal during push off. Again, foot intersections with the terrain manifest themselves; this is due to the weighted averaging employed in selecting the body orientation.

The overall transition is unsatisfactory, showing that a passage from one gait to another cannot be effectively created by naïve blending. The gait changes in more complex ways during a transition. It is likely that the changes in the phase timings do not follow the simple linear progression that is employed. Nor do the legs take interpolated poses between the two gaits. The topic of transitions requires further study to uncover the issues involved.

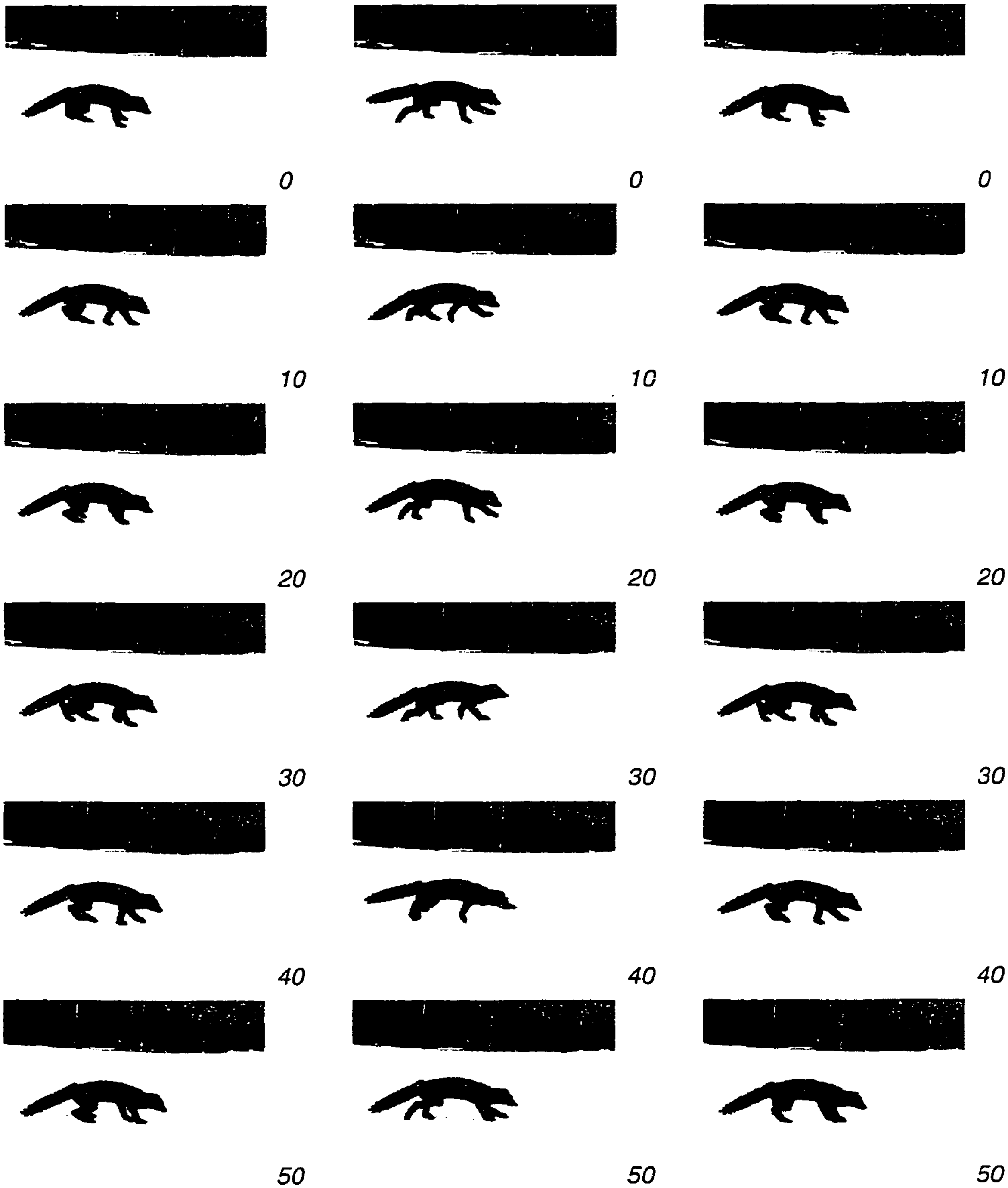


Figure 4.21: **Generating a transition between two gaits. First, the two contributing gaits are temporally aligned so that the leg phases linearly change from those of the amble to those of the bound over the transition interval. The left column shows the temporally adjusted amble and the middle column the bound. These two modified gaits are then spatially blended from the first into the second to achieve the final transition in the right column.**

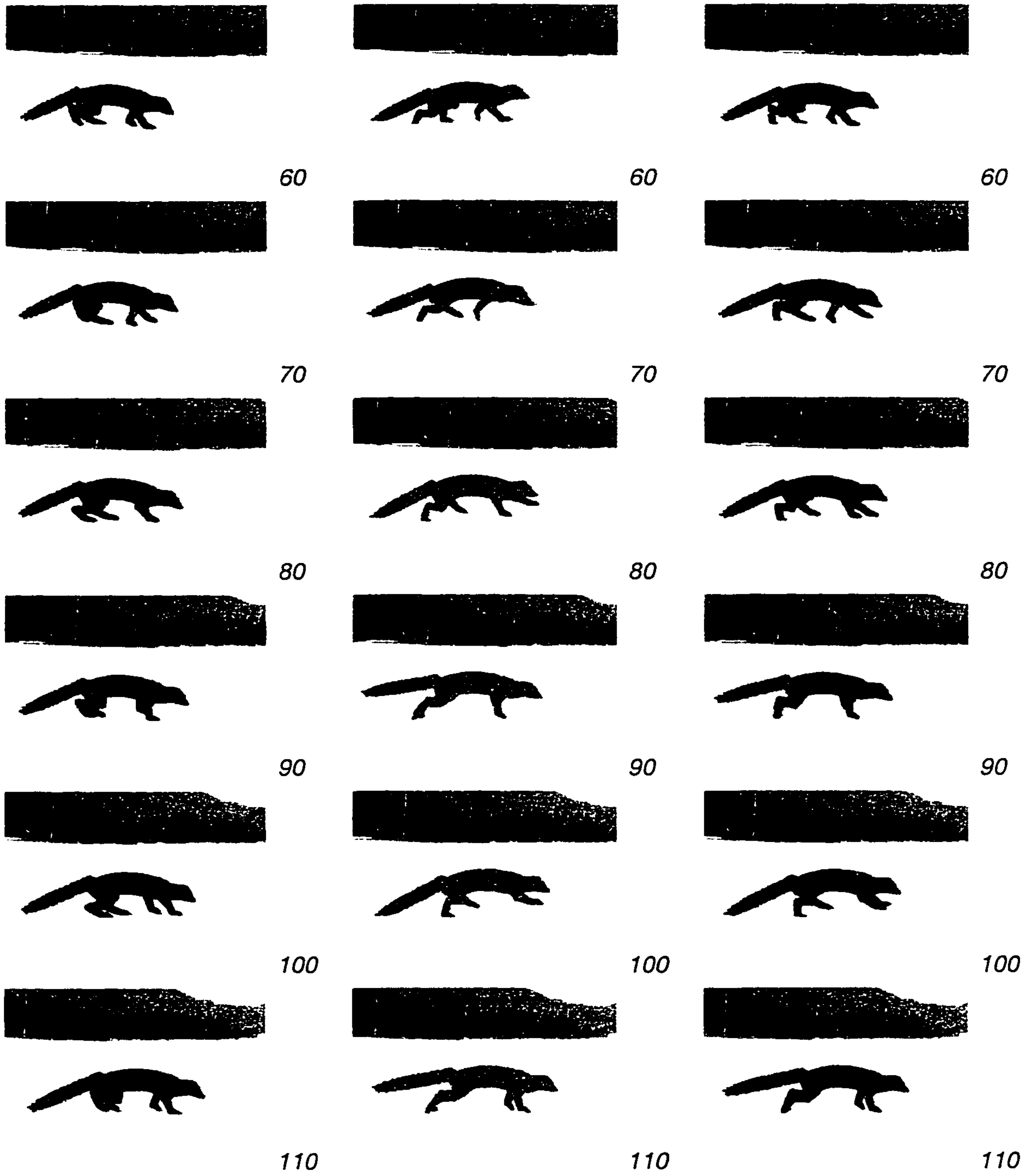


Figure 4.21 (Cont.): **Generating a transition between two gaits. First, the two contributing gaits are temporally aligned so that the leg phases linearly change from those of the amble to those of the bound over the transition interval. The left column shows the temporally adjusted amble and the middle column the bound. These two modified gaits are then spatially blended from the first into the second to achieve the final transition in the right column.**

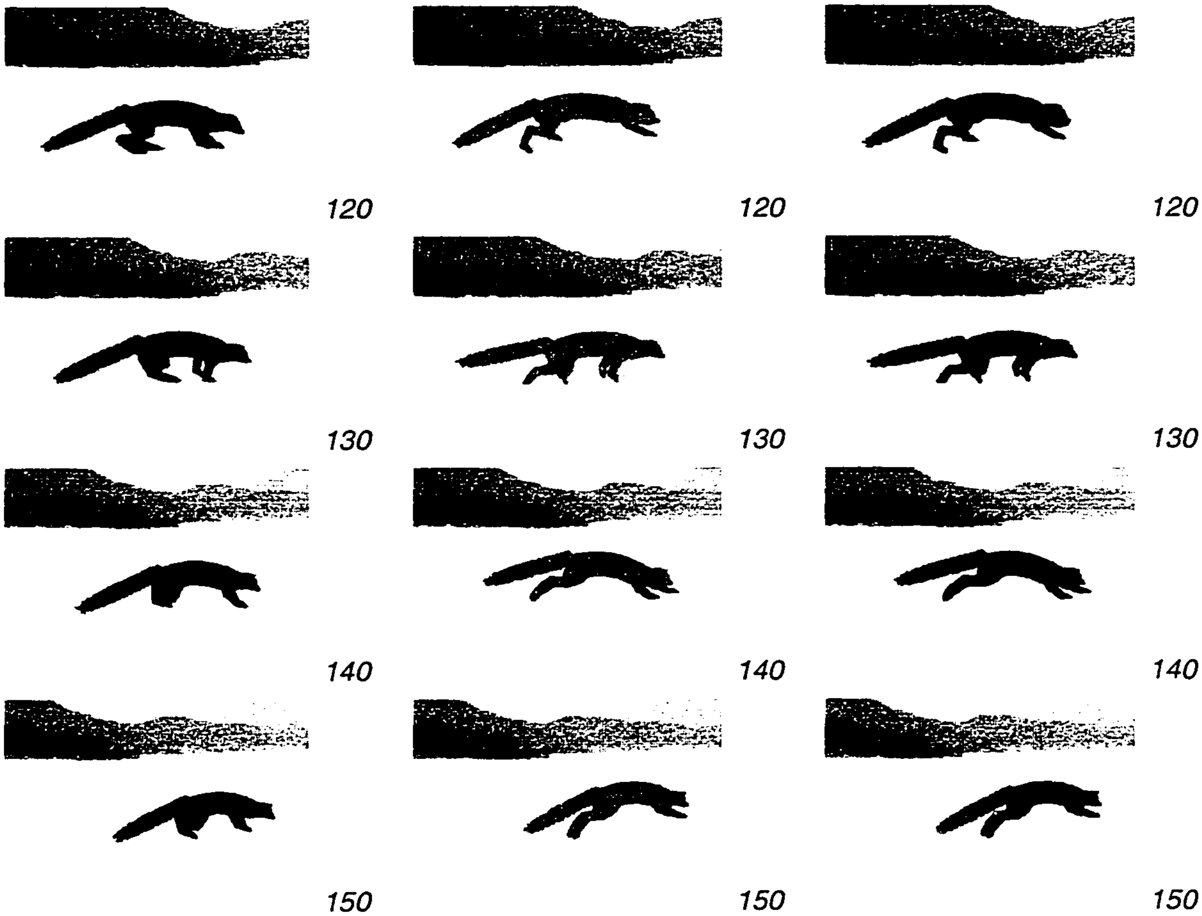


Figure 4.21 (Cont.): **Generating a transition between two gaits. First, the two contributing gaits are temporally aligned so that the leg phases linearly change from those of the amble to those of the bound over the transition interval. The left column shows the temporally adjusted amble and the middle column the bound. These two modified gaits are then spatially blended from the first into the second to achieve the final transition in the right column.**

4.4 Summary

This chapter documented the results of several collections of experiments involving the reshaping of motions by warping. Many of the benefits and problems underlying the technique were highlighted.

The first two sets of tests involved motion warping in general. A bouncing beach ball provided a simple demonstration of each of the forms of temporal and spatial deformation. It was found that, in cases such as this, the process of specifying warps is not necessarily more time-efficient than editing the motion curves directly. Also, there are difficulties inherent in representing a reshaped motion. The next experiments involved the modification of a swinging robotic arm. The purpose was to evaluate the effects of varying weights of offset and scaling deformations. It was unclear how to choose between these two varieties of spatial warps; a trial-and-error approach must be employed to find the best compromise.

Adapting quadruped-specific locomotions formed the second series of experiments. A bounding gait was automatically tailored to make a marten traverse over some hilly terrain. Specifying the course and speed over the topography was a time-consuming task. This suggests a need for automatic extraction of the ground's topography and reparameterization of the path to give precise speed control. Other problems such as foot-ground intersections and slippage occur due to the approximations of the surface and foot positions used. Another test involved generating a transition from one gait to another; the results proved to be unsatisfactory. The linear change in phase timings and the blended movements of the two source gaits did not appear visually convincing.

The upcoming chapter presents the final conclusions of the thesis. The contents of the work are summarized and the key outcomes enumerated. Several of the salient features and experimental results of motion warping and the quadruped locomotion system are discussed. The objectives and the overall design of the system are reiterated. Finally, some noteworthy ideas meriting future research are suggested.

Chapter 5

Conclusions

This thesis has explored, via implementation, test cases, and a review, the plausibility of using motion warping as an editing technique in an animation system. In particular, the problem of augmenting four-legged locomotions has been considered. In this section, a final summary of the contents of the thesis and its results is made. The merits and weaknesses of the motion editing technique, revealed during the process of implementation and testing, are indicated. The objectives of the thesis, namely the goals of an automated system for adjusting animated quadruped gaits, are reiterated. The procedures for adapting gaits to variable terrain and for creating transitions are revisited. Lastly, some proposals for future work in motion warping, especially with regard to quadruped animation, are suggested.

5.1 Summary

5.1.1 Contents of Thesis

This thesis has examined the potential of motion warping as an editing technique, especially as it applies to the domain of quadruped locomotion. A comprehensive review of the background literature pertaining to articulated figure animation was conducted. This provided the insight necessary to choose a design approach for the quadruped editing system. The architecture incorporates a motion warping technique to deform existing movements according to defined constraints.

Generic warping facilities were first developed, and some simple animations were created to demonstrate the technique. This process clarified many of the benefits and weaknesses of the method. Most seriously, motion warping was found to be a low-level technique, its influence localized to individual parameters of an articulated figure. Further refinements and high-level controls were introduced specific for directing the movements of a quadruped. These are used to make the animal move along a defined path at variable speeds and to build transitions between gaits.

5.1.2 Significant Results

There are a number of noteworthy concerns with respect to the motion warping technique. Difficulties include the representation of the warping parameters and the resulting reshaped motions. In the case where the motion is defined by little data, warping is less effective (*i.e.* more time-consuming) than editing the original data directly. In addition, specifying an excessive deformation at some point in the motion may induce undesirable perturbations in the surrounding frames. Likewise, poor results are produced if the warp spans several extreme poses within the motion and an insufficient number of warping keyframes are defined.

There are several positive aspects of motion warping, too. The technique incorporates many features desirable in an editing system:

1. Warping can be applied to movements obtained from a variety of sources, such as keyframe and motion capture data.

2. Traditional animation principles are preserved; the scaling and time warps are equivalent to the axioms of exaggeration and timing, respectively.

3. Motion deformation aids in building smooth transitions from one movement to another during an animation sequence.

In addition, warping is applied globally over the entire time span of an action; thus, the movement retains the details and qualities inherent in the original. Finally, the technique can be integrated into a more complex editing system, as the quadrupedal locomotion facilities demonstrate.

Given the animated motion of a quadruped gait and some simple knowledge regarding its leg phases, it is possible to automatically adapt the locomotion for new environments and purposes. For instance, the gait can be applied along a three-dimensional, user-defined path—the outcome is that the animal locomotes along the route. Transitions can be realized by gradually adjusting the phase timings and progressively blending between the leg positions of two gaits.

Over much of the sequence, the motion of a marten bounding along a path atop some hills appears reasonable and smooth. The body remains parallel to the terrain, whilst the ankles and hips are rotated, ensuring that the stance feet remain firmly planted on ground. However, because of the approximations used and some limitations of motion warping, there are some problems with the system.

1. At times, the support legs intersect or hover above the ground and the feet slip. These defects are attributable to the approximation used for the terrain location and slope.

2. The animal makes unlikely directional changes when passing over summits, dips, or other high frequency topography. There is a need to compensate for these extrema.

3. Leg motions become unsynchronized when the stride length is adjusted to compensate for speed variations. This is due to each leg being considered independently.

4. Simple linear changes in the phase timings and blending of leg positions is insufficient for creating visually believable transitions.

5. Customizing gaits is limited to the motion warping of individual leg parameters. There is a need to express these warps at a more global scale, influencing a set of leg joints.

6. Continuous constraints imposed by motion warps result in all contributions from the source motion being ignored. This makes deformation a poor choice for enforcing continuous restrictions.

5.2 Review of Results

From implementing the technique, the problems and benefits of motion warping were identified. Of concern are how to express the temporal and spatial deformation parameters, and how to represent the motion resulting from a warp. The test cases provided insight into the circumstances under which motion deformation is an ineffective form of editing. This highlighted the need for more sophisticated controls catered towards specific problem domains.

- **Representation of the warping parameters**: Witkin and Popovič suggest a keyframing approach for specifying motion warps, using Cardinal splines for interpolation [57]. We use Kochanek and Bartel's continuity-tension-bias (CTB) splines [31]. Spline-based keyframing has the benefit of being a familiar specification method for animators, and CTB splines in particular offer additional control over the interpolation. A disadvantage is that the deformation splines are localized to one parameter of the articulated model. In addition, CTB splines offer a trade-off between speed versus directional control through key points. A final consideration is whether to express the spatial warp in terms of the motion's original or warped time, although there is no practical benefit in choosing one form over the other.

- **Representation of the deformed motion**: Warping was designed to work with motions obtained from any one of a multitude of sources. Application of the warping equations to spline-based motions yield high-order polynomials. This is an overly complex representation not supported by commercial animation packages. Approximation by a lower-degree polynomial would sacrifice the details inherent in the original movement. Finally, deforming motion captured data, described as a multitude of discrete points, yields an equal number of modified points.

  The solution is to express the resultant action as sampled parameter values, one per frame, interpolated by a CTB spline. Unfortunately, there is the potential errors to accumulate when reshaping a previously deformed movement; data may have been discarded during the initial warp that was needed in the description of the final motion.

- **Blending for transitional purposes**: As has been indicated, the third required feature of an animation editing system is the capability to concatenate motions into a consistent sequence. Motion reshaping can be employed to bring two motions into a reasonable temporal alignment. Then, cross-dissolving is applied to the models' parameters to yield a smooth transition. For instance, a switch from one gait to another can be achieved by temporally warping the phase timings of the starting gait into those of the concluding gait. Then, the leg positions are blended to complete the transition.

- **Test cases**: A bouncing ball and a robotic arm were used to test the generic motion warping technique. The motion of a bouncing ball is simple, requiring keyframes only at ground contact and at the apogee. Several modifications were made by warping, such as a decaying bounce and jumping up a series of steps. For these examples, the specification of warps was concluded to be more involved than directly editing the original motion data. However, for more complex motions involving a large number of keyframes (*e.g.* motion capture data), warping would be the more efficient approach to editing.

  The swivelling arm experiments were used to compare the effects of scaling and offset warps. In many cases, the deformations adversely distorted the original movement. These results are attributable to a number of factors: an insufficient number of warping keyframes were defined, the goal pose of the reshaped arm was significantly different from the associated pose in the source motion, and the weights chosen for the scaling and offset functions were poor.

- **Controls for quadruped locomotion**: Several high-level controls were introduced, providing the information necessary to automatically create motion warps for a walking quadruped. Along with the animated motion of one stride of a gait, the user provides details regarding the name of each foot object in the model and the phase timings of each leg[1].

  To create a locomotion over varying topography, the user must specify a path in three-dimensions. This is a time-consuming task. The animator must place the route precisely on the surface of the terrain. The spacing of the keyframes controls the speed of the animal. This task could be better automated by projecting a two-dimensional version of the path on the terrain and reparameterizing based on desired speed.

  To generate a transition, the user supplies the two contributing gaits together with their phase timing data. Also, the duration of the passage must be indicated.

- **Outcomes of the quadruped locomotion system**: One test involved using the quadruped system to create an animation of a marten bounding along a chosen path. The resulting locomotion appeared, for the most part, to be reasonable. The creature moved smoothly along the course. Adjustments made to the torso's orientation and position mirrored the terrain's slope and altitude. The ankles rotated to ensure that the supporting feet remained flush on the ground.

  However, there are several problems with this approach, some easily amendable, others more complicated. For a given frame along the route, the height of the terrain is assumed to be the altitude of the path under the centre of the body. The slope is also measured at this point and taken to be constant. These approximations are used to decide the orientation and location of various parts of the animal.

  These assumptions lead to instances when a stance foot either intersects or floats above the ground. Also, slippage of supporting feet occurs, particularly when performing turns. These can be corrected by integrating knowledge of the terrain and forwards kinematics to exactly position the feet. There is also a tendency for unlikely body movements. For instance, acute changes in the path, such as occur at summits or regions with high frequency variations, result in the body orientation changing too rapidly. This problem is difficult to address; perhaps the application of a low-pass filter to the path would help reduce the disturbance.

---

1. This data could be derived by detecting when a foot resides in a low, stationary position. This duration forms the stance phase; for the rest of the stride, the leg is in its flight phase.

Adaptations were made to the phase timings of the legs to adjust the stride length based on the quadruped's speed. Because these modifications were only considered individually for each leg, the phase timings rapidly become out of synchronization as the creature accelerates. A possible improvement would be to keep a record of the timings of all the legs. Then, adjustments could be made that consider the cumulative phase timings of the limbs as a whole.

- **Gait transition results**: A five second transition from an amble to a bound was generated. The motion of the animal is smooth over the interval, with the leg timings and motions progressively changing between the two gaits. However, the results do not appear convincing. For instance, the hind legs literally appear to be gimpy, dragging behind the animal. One possible reason for this effect is that the leg phase timings do not change in a simple linear fashion. Nor, perhaps, can the transitory leg motions be represented as a weighted mix of the two source gaits.

  The transitional animation suffers from leg intersections with the ground. This is partially attributable to the difficulty in choosing a body position and orientation given the different poses of each leg. Forward kinematics would help in this matter. The body or limb postures could then be customized based on the position of each foot, preventing foot collisions with the terrain. The ability to apply a motion warp globally on an entire limb, rather than on individual parameters, would be useful for repositioning limbs that drag on the ground.

5.3 Overview of Objectives

The process of constructing an animation sequence via a computer-based editing system may be divided into three tasks: specifying the constituent motions, customizing these movements, and concatenating motions to form a sequence. Many techniques have evolved that satisfy the first problem of creating animated movements, as a review of the literature has shown. These include keyframing, motion capture, inverse kinematics, and dynamic simulation. Each of these approaches exhibit trade-offs with respect to the user's degree of control and the generality of the solution[2].

2. Generality is characterized by whether the method can be applied to a variety of articulated figures or used to create different kinds of motion.

The issue of editing existing movements merits further investigation. There is some support for customizing motions inherent in several of the aforementioned specification techniques. However, an independent editing system is required should only the final motion data be available (*e.g.* motion capture data). An editing system should embody four objectives:

1. The controls should be easy and intuitive for an animator to use.
2. An altered motion should retain the subtleties inherent in the original movement.
3. Traditional animation principles such as arcs, timing, and exaggeration should be supported.
4. The system should work with motion data obtained from any source.

Animation research has focused on the study of bipedal motion. The humanoid form is of natural interest to people and has a diverse range of applications. Nevertheless, a large variety of animal species use four legs for locomotion. Quadrupedal motion interjects some additional difficulties because of the added complexity of multiple legs and the variety of gaits.

This thesis has explored motion warping as a technique for editing animated movements. The benefits and shortcomings of the technique with regard to the editing system criteria have been enumerated. Implementation and application difficulties have also been highlighted. The problem of modifying quadruped walks using motion warping has been the specific domain of interest. The quadruped locomotion editing system has the following qualities:

- The primary task is to automatically make the animal move along a user-defined path at varying speeds. A secondary motive is to facilitate the creation of transitions between gaits.
- The approach is sufficiently general to work with a variety of four-legged, articulated models and sample walking motions.

5.4 Outline of the Design of the Quadruped Motion Warping System

Motion warping, proposed by Witkin and Popović, is a method for deforming animated movements [57]. The reshaped variants retain the characteristics inherent in the original motions. Articulated models, a common representation of objects in computer-animated scenes, are animated by changing parameter values, such as joint rotations and segment positions, over time. Using the motion deformation method, the user specifies time, scale, and offset warping functions that are applied to individual parameters. The temporal warp speeds up, slows down, or even reverses intervals of the motion. The scaling warp exaggerates and understates sections of the movement. Portions of the movement can be displaced and new features inserted via the offset warp.

Although a warp may have an influence over an entire motion, its use is limited to a single parameter[3]. To facilitate the warping of quadruped walks, some high-level controls have been devised. The animator provides the following data to the system:

1. A four-legged articulated model of the creature.

2. A sample of one cycle of a walking motion of the animal.

3. The identity of the model components such as the torso, hips, and feet.

4. The state timings of each leg, namely when a foot makes contact with and when it pushes off from the ground.

5. The path the quadruped is to walk along, if relevant; the speed of locomotion is inherent in the description of the path.

6. The duration of the transition, if applicable.

Given this information, motion warps are automatically computed to make the animal move along the given path. Because the creature's speed and direction may not match that of the template walking movement, several constraints must be considered. The altitude and slope of the terrain is estimated at a point directly under the torso using the path curve. These values are used to position the body of the animal on the route and set its heading and pitch. Ankle and hip rotations ensure that the stance feet remain stationary on the ground even as the terrain changes and the path turns. The effects of speed on the stride are realized by adjusting the leg phase timings.

To create a passage from one movement to another for concatenation purposes requires the inputs described above, albeit with a set of two gait motions. In addition, the duration of the transition must be indicated. The process commences with the creation of temporally aligned versions of each of the locomotions, repeating the strides as necessary so that the action spans the specified duration. The aligned sequences start with the phase timings of the first motion and linearly proceed into the phase timings of the final motion. The joint parameters of the two augmented gaits are blended over the interval using a slow-in, slow-out curve. This yields the final transitional animation.

---

3. The exception is a time warp, which may be applicable over multiple parameters.

5.5 Future Work

This work has uncovered some of the problems of motion warping as an editing technique. One significant shortcoming is that separate warps must be specified for each parameter of an articulated model. This instigated the creation of high level controls for automatically warping quadrupedal walking motions. However, there is a potential for further research on motion warping in general and, in particular, its application to quadruped motion.

There is a need to expand the general motion warping method so that it may be applied over groups of parameters. There are two aspects to this problem to consider:

1. **How may parameters be grouped?** Given that many animations involve a tree hierarchy of joined segments (*i.e.* an articulated figure), one possibility is to limit the selection to connected components. Another choice is to select parameters based on close spatial proximity. An evaluation of the effectiveness of these groupings needs to be conducted.

2. **How should the warp be specified and applied to this collection of parameters?** Articulated model parameters can be preassigned weights and then a single spatial warp specified and distributed over the parameters within the group. Alternatively, attractors can be used to pull or repulse the parameters within a defined region. An inverse kinematics approach could be employed that yields a solution that minimally perturbs the original motion.

This thesis focused on editing quadruped walks using motion warping. There are several other motions of which animals are capable. High level controls for directing actions such as jumping and climbing need to be developed. For instance, variations in the degree of crouch before and horizontal extension of the legs during a leap may be expressed as a function of the jump's height and distance.

It is important for the animation system to have available the most accurate information regarding the positions of objects, particularly the terrain over which the articulated figure is moving. The approximations used by the quadruped system were insufficient. To improve the performance, forward kinematics could be used to determine the positions of the distal segments of the figure. A means to extract surface information from terrain objects is also vital. Together, these features can be used to ensure good contact of stance legs with the ground.

Because the motion editing facility devised in this thesis is a prototype, its user interface is rudimentary. There is a need for graphical interfaces to enter the gait information and a requirements for better integration with existing animation software. For example, walking phases could be specified using state diagrams as depicted in Gray's *Animal Locomotion* [23]. Other parameters, particularly those that vary depending on the animal's speed, could be interactively adjusted and visualized.

The quadruped locomotion technique incorporates empirical information as encoded by its high level controls. More visually realistic results may be attainable by including additional knowledge such as descriptions of leg postures and dynamics. Animals have a variety of leg structures. For instance, digitigrades walk on their toes whereas plantigrades walk on the soles of their feet. A robust quadruped animation system needs to support these various leg postures and take into consideration their effect on walking movements. By adhering to physics-based limitations, the number of high level controls required may be reduced, their values replaced by the results of dynamic simulations.

Bibliography

[1] Alexander, R. McNeill. *How Animals Move*. Ed. Alexander Kuznetzov. CD-ROM. San Rafael: Maris Multimedia, 1995.

[2] Alexander, R. McNeill. *Locomotion of Animals*. Glasgow, Scot.: Blackie & Son Limited, 1982.

[3] Alexander, R. McNeill. "Mechanics and Scaling of Terrestrial Locomotion." *Scale Effects in Animal Locomotion*. Ed. T.J. Pedley. New York: Academic Press Inc., 1977. 93-110.

[4] Armstrong, W.W. and M. Green. "The Dynamics of Articulated Rigid Bodies for Purposes of Animation." *Proceedings Graphics Interface '85*. Ed. M. Wein and E.M. Kidd. Graphics Interface '85, 27-31 May 1985, Montréal, PQ. Toronto: The Canadian Information Processing Society, 1985. 407-415.

[5] Amaya, Kenji, Armin Bruderlin, and Tom Calvert. "Emotion from Motion." *Proceedings Graphics Interface '96*. Ed. Wayne A. Davis and Richard Bartels. Graphics Interface '96, 22-24 May 1996, Toronto, ON. Toronto: The Canadian Information Processing Society, 1996. 222-229.

[6] *Bambi*. Dir. David Hand. RKO Radio Pictures Inc., 1942. 69 min.

[7] Bartels, Richard H. and Ines Hardtke. "Speed Adjustment for Key-Frame Interpolation." *Proceedings Graphics Interface '89*. Ed. Marceli Wein and Evelyn M. Kidd. Graphics Interface '89, 19-23 Jun. 1989, London, ON. Toronto: The Canadian Information Processing Society, 1989. 14-19.

[8] Beier, Thaddeus and Shawn Neely. "Feature-Based Image Metamorphosis." *Computer Graphics (SIGGRAPH '92 Conference Proceedings)*. Ed. Steve Cunningham. SIGGRAPH '92, 26-31 Jul. 1992, Chicago, IL. 26.2 New York: ACM SIGGRAPH, 1992. 35-42.

[9] Blumberg, Bruce M. and Tinsley A. Galyean. "Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments." *Computer Graphics Proceedings, Annual Conference Series, 1995*. Ed. Robert Cook. SIGGRAPH 95, 6-11 Aug. 1995, Los Angeles, CA. New York: ACM SIGGRAPH, 1995. 47-54.

[10] Boulic, Ronon and Ramon Mas. "Hierarchical Kinematic Behaviors for Complex Articulated Figures." *Interactive Computer Animation*. Ed. Nadia Magnenat Thalmann and Daniel Thalmann. Hemel Hempstead, Eng.: Prentice Hall Europe, 1996. 40-70.

[11] "Break-Through In Motion Capture." *Next Generation Online* 6 Apr. 1998. 21 Sep. 1998 <http://www.next-generation.com/jsmid/news/2743.html>.

[12] Bruderlin, Armin and Tom Calvert. "Interactive Animation of Personalized Human Locomotion." *Proceedings Graphics Interface '93*. Ed. Norman Jaffe, Alain Fournier, and Pierre Poulin. Graphics Interface '93, 19-21 May 1993, Toronto, ON. Toronto: The Canadian Information Processing Society, 1993. 17-23.

[13] Bruderlin, Armin and Thomas W. Calvert. "Goal-Directed, Dynamic Animation of Human Walking." *Computer Graphics Proceedings, Annual Conference Series, 1989*. Ed. Richard J. Beach. SIGGRAPH '89, 31 Jul.-4 Aug. 1989, Boston, MA. 23.3 New York: ACM SIGGRAPH, 1989. 233-242.

[14] Burtnyk, N. and M. Wein. "Computer-Generated Key-Frame Animation" *Journal of the SMPTE*. 80 (1971): 149-153.

[15] Carmack, John, John Romero, *et al. Quake*. CD-ROM. Vers. 1.06. Computer software. Mesquite: id Software, Inc., 1996.

[16] da Silva, Fernando Wagner, Luiz Velho, Paulo Roma Cavalcanti, and Jonas Gomes. "An Architecture for Motion Capture Based Animation." *Proceedings of SIBGRAPI'97*. X Brazilian Symposium of Computer Graphics and Image Processing, Oct 1997. 49-56.

[17] *Fantasia*. Dir. Ben Sharpsteen et al. RKO Radio Pictures Inc., 1940. 120 min.

[18] Florence, Zack L., *et al.* "Development of Computer-based 3-D, Biomechanic Models of Selected Animals, Including Problem Wildlife Species." Unpublished report, Alberta Research Council, Vegreville, AB, 1998.

[19] *Gertie the Dinosaur*. Dir. Winsor McCay. Box Office Attractions Company, 1914.

[20] Giancoli, Douglas C. *Physics for Scientists and Engineers*. 2nd ed. Englewood Cliffs: Prentice-Hall, 1988. Rpt. of *General Physics*.

[21] Girard, Michael. "Interactive Design of 3D Computer-Animated Legged Animal Motion." *IEEE Computer Graphics and Applications*. 7.6 Los Alamitos: IEEE Computer Society, 1987. 39-51.

[22] Gleicher, Michael. "Retargeting Motion to New Characters." *Computer Graphics Proceedings, Annual Conference Series, 1998*. SIGGRAPH 98, 19-24 Jul. 1998, Orlando, FL. New York: ACM SIGGRAPH, 1998. 33-42.

[23] Gray, James. *Animal Locomotion*. The World Naturalist. Ed. Richard Carrington. London, Eng.: Weidenfeld and Nicolson, 1968.

[24] Hébert, James M., John F. Gross, Bruce Branit, Brad Peebler, and David Hibsher. *LightWave 3D Reference Manual*. Ed. James M. Hébert. Topeka: NewTek, Inc., 1995.

[25] Hébert, James M., John F. Gross, Bruce Branit, Brad Peebler, and David Hibsher. *LightWave 3D User Guide*. Ed. James M. Hébert. Topeka: NewTek, Inc., 1995.

[26] Hodgins, Jessica K. "Animating Human Motion." *Scientific American*. Mar. 1998: 64-69.

[27] Hodgins, Jessica K. *GVU: Animation Lab*. Sep. 22 1998 <http://www.cc.gatech.edu/gvu/animation/>.

[28] Hodgins, Jessica K. and Nancy S. Pollard. "Adapting Simulated Behaviors For New Characters." *Computer Graphics Proceedings, Annual Conference Series, 1997*. SIGGRAPH 97, 3-8 Aug. 1997, Los Angeles, CA. New York: ACM SIGGRAPH, 1997. 153-162.

[29] Hodgins, Jessica K., Wayne L. Wooten, David C. Brogan, and James F. O'Brien. "Animating Human Athletics." *Computer Graphics Proceedings, Annual Conference Series, 1995*. Ed. Robert Cook. SIGGRAPH 95, 6-11 Aug. 1995, Los Angeles, CA. New York: ACM SIGGRAPH, 1995. 71-78.

[30] *Jurassic Park*. Dir. Steven Spielberg. Universal Pictures, 1993. 127 min.

[31] Kochanek, Doris H.U. and Richard H. Bartels. "Interpolating Splines with Local Tension, Continuity, and Bias Control." *Computer Graphics (SIGGRAPH '84 Conference Proceedings)*. Ed. Hank Christiansen. SIGGRAPH '84, 23-27 Jul. 1984, Minneapolis, MN. 18.3 New York: ACM SIGGRAPH, 1984. 33-41.

[32] Kokkevis, Evangelos, Dimitri Metaxas, and Norman I. Badler. "Autonomous Animation and Control of Four-Legged Animals." *Proceedings Graphics Interface '95*. Ed. Wayne A. Davis and Przemyslaw Prusinkiewicz. Graphics Interface '95, 17-19 May 1995, Québec, PQ. Toronto: The Canadian Information Processing Society, 1995. 10-17.

[33] Lamouret, Alexis and Michiel van de Panne. "Motion Synthesis By Example." *Computer Animation and Simulation '96*. Ed. R. Boulic and G. Hegron. Seventh International Workshop on Computer Animation and Simulation, 31 Aug-1 Sep. 1996, Poitiers, Fr. Wien, Aus.: Springer-Verlag, 1996. 199-212.

[34] Lasseter, John. "Principles of Traditional Animation Applied to 3D Computer Animation." *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*. Ed. Maureen C. Stone. SIGGRAPH '87, 27-31 Jul. 1987, Anaheim, CA. 21.4 New York: ACM SIGGRAPH, 1987. 35-44.

[35] "LightWave 3D Scene File Format." *SDK Information*. Vers. 4.0. Jun. 1995. NewTek, Inc. 22 Jul. 1998 <http://www.newtek.com/developer/sdk.html>.

[36] Liu, Zicheng and Michael F. Cohen. "Keyframe Motion Optimization By Relaxing Speed and Timing." *Computer Animation and Simulation '95*. Ed. D. Terzopoulos and D. Thalmann. Sixth Workshop on Computer Animation and Simulation, 2-3 Sep. 1995, Maastricht, Neth. Wien, Aus.: Springer-Verlag, 1995. 144-153.

[37] Litwinowicz, Peter and Lance Williams. "Animating Images with Drawings." *Computer Graphics Proceedings, Annual Conference Series, 1994*. Ed. Steve Cunningham. SIGGRAPH 94, 24-29 Jul. 1994, Orlando, FL. New York: ACM SIGGRAPH, 1994. 409-412.

[38] Maiocchi, Roberto. "3-D Character Animation Using Motion Capture." *Interactive Computer Animation*. Ed. Nadia Magnenat Thalmann and Daniel Thalmann. Hemel Hempstead, Eng.: Prentice Hall Europe, 1996. 10-39.

[39] Muybridge, Eadweard. *Animal Locomotion*. 3 vols. New York: Dover Publications, Inc., 1979. Rpt. of *Muybridge's Complete Human and Animal Locomotion*. 1887.

[40] Perlin, Ken. "Real Time Responsive Animation with Personality." *IEEE Transactions on Visualization and Computer Graphics*. 1.1 (1995): 5-15.

[41] *Pinocchio*. Dir. Hamilton Luske and Ben Sharpsteen. RKO Radio Pictures Inc., 1940. 88 min.

[42] Samson, Jean-Paul C. "Augmenting Animations Via Motion Warping." *Western Computer Graphics Symposium 1998*. Western Computer Graphics Symposium, 23-26 Apr. 1998, Whistler, BC. Vancouver: 1998. 71-81.

[43] Serway, Raymond A. *Physics for Scientists & Engineers*. 3rd ed. Philadelphia: Saunders College Publishing, 1990.

[44] Sims, Karl. "Evolving Virtual Creatures." *Computer Graphics Proceedings, Annual Conference Series, 1994*. Ed. Steve Cunningham. SIGGRAPH 94, 24-29 Jul. 1994, Orlando, FL. New York: ACM SIGGRAPH, 1994. 15-22.

[45] *Snow White and the Seven Dwarfs*. Dir. David Hand. RKO Radio Pictures Inc., 1937. 83 min.

[46] Sukhanov, V.B. *General System of Symmetrical Locomotion of Terrestrial Vertebrates and Some Features of Movement of Lower Tetrapods*. Trans. Dr. M.M. Haque. Ed. Dr. V.S. Kothekar. New Delhi, India: Amerind Publishing Co. Pvt. Ltd., 1974.

[47] Thalmann, Nadia Magnenat and Daniel Thalmann. "Computer Animation in Future Technologies." *Interactive Computer Animation*. Ed. Nadia Magnenat Thalmann and Daniel Thalmann. Hemel Hempstead, Eng.: Prentice Hall Europe, 1996. 1-8.

[48] Thomas, Frank and Ollie Johnston. *Disney Animation: The Illusion of Life*. Ed. Walton Rawls. New York: Abbeville Press, 1981.

[49] Torkos, Nick. "Footprint-based Quadruped Motion Synthesis." Diss. U of Toronto, 1997. 26 Sep. 1998 <http://www.dgp.toronto.edu/people/torkos/thesis.html>.

[50] Torkos, Nick and Michiel van de Panne. "Footprint-based Quadruped Motion Synthesis." *Proceedings Graphics Interface '98*. Ed. Wayne Davis, Kellogg Booth, and Alain Fournier. Graphics Interface '98, 18-20 Jun. 1998, Vancouver, BC. Toronto: The Canadian Information Processing Society, 1998. 151-160.

[51] Unuma, Munetoshi, Ken Anjyo, and Ryozo Takeuchi. "Fourier Principles for Emotion-based Human Figure Animation." *Computer Graphics Proceedings, Annual Conference Series, 1995*. Ed. Robert Cook. SIGGRAPH 95, 6-11 Aug. 1995, Los Angeles, CA. New York: ACM SIGGRAPH, 1995. 91-96.

[52] van de Panne, Michiel, Ryan Kim, and Eugene Fiume. "Virtual Wind-up Toys for Animation." *Proceedings Graphics Interface '94*. Ed. Wayne A. Davis and Barry Joe. Graphics Interface '94, 18-20 May 1994, Banff, AB. Toronto: The Canadian Information Processing Society, 1994. 208-215.

[53] Watt, Alan and Mark Watt. *Advanced Animation and Rendering Techniques—Theory and Practice*. New York: ACM Press, 1992.

[54] Wiley, Douglas J. and James K. Hahn. "Interpolation Synthesis of Articulated Figure Motion." *IEEE Computer Graphics and Applications*. 17.6 Los Alamitos: IEEE Computer Society, 1997. 39-45.

[55] Witkin, Andrew. *Andy Witkin's Gallery*. 21 Sep. 1998 <http://www.cs.cmu.edu/afs/cs/user/aw/www/gallery.html>

[56] Witkin, Andrew and Michael Kass. "Spacetime Constraints." *Computer Graphics (SIGGRAPH '88 Conference Proceedings)*. Ed. Richard J. Beach. SIGGRAPH '88, 1-5 Aug. 1988, Atlanta, GA. 22.4 New York: ACM SIGGRAPH, 1988. 159-168.

[57] Witkin, Andrew and Zoran Popović. "Motion Warping." *Computer Graphics Proceedings, Annual Conference Series, 1995*. Ed. Robert Cook. SIGGRAPH 95, 6-11 Aug. 1995, Los Angeles, CA. New York: ACM SIGGRAPH, 1995. 105-108.

Glossary

action

A collection of coordinated motions of articulated model segments which, considered as a whole, form a recognizable, high-level movement. For example, a walking action is composed of the motions of leg segments as they swing through the air and push against the ground.

animation

The process of creating the appearance of motion by the rapid display of a series of still images.

animation, computer

The rendering of instantaneous states of a dynamic scene to a display screen or recording device [47].

animation sequence

A series of actions applied to an object within a scene.

animation, traditional

A specialized form of animation where hand-illustrated pictures or posed sculptures are photographed onto frames of film.

anticipation

The traditional animation principle of attracting a viewer's attention, preparing him for an upcoming action.

arcs

A principle of animation is that objects tend to move along smooth, curved paths or arcs rather than straight lines [34, 48].

articulated figure or *model*

A two or three-dimensional geometric model constructed of rigid segments connected by joints.

autonomous behaviour

Automation of the animation of characters using artificial intelligence techniques to bestow behaviours [9]. These behaviours are a result of the desires and goals internalized by the character.

biomechanics

The study of animal skeletal-muscular structure and motion from biological and mechanical perspectives.

bones

A type of control object attached to a model that influence the surrounding geometric surface. As bones are manipulated, the attached surfaces are deformed, conveying the effect of flexing.

degrees-of-freedom (d.o.f.)

The dimensionality of a parameter describing the placement of an articulated figure segment. A segment has from one to six degrees-of-freedom: its location in space (requiring up to three dimensions) and its rotation (necessitating up to three angles, one for each of the axes of rotation).

duty factor or *support time*

The fractional duration of the locomotory period during which a leg supports the mass of the body [2].

dynamics

The study of the mechanics of motion in terms of forces and its effects on motion and equilibrium.

dynamic simulation

Given control algorithms governed by high level parameters, an action is generated by approximating the physical forces required to propel body segment masses to their desired positions.

end effector

A distal point on an articulated model that is positioned to suggest an overall pose for a limb. Inverse kinematics is employed to find a limb posture that satisfies this goal.

exaggeration

A traditional animation principle of emphasizing an action to make it more convincing or appealing [34, 48].

gait

The coordinated, rhythmic pattern of movements of an animal for the purposes of locomotion [46].

image warping

The controlled deformation of images, yielding results akin to the reflection from a bent mirror. The distortion is achieved by altering a mesh, line segments, or points aligned to important features in the source images and mapping the pixels accordingly.

in-between frames

Those frames between the keyframes of an animation sequence.

inverse kinematics

A technique for finding the joint angles of an articulated model such that the endpoint of the model reaches for or touches a desired goal point.

keyframing with interpolation

The process of creating keyframes of an animation sequence. Interpolation between the keyframes is used to automatically generate the intermediate frames.

keyframes

Isolated frames of an animation sequence depicting an object in a extreme pose that characterizes its current action.

kinematics

The study of motion without regard to the physics (*i.e.* masses and forces) involved.

mechanics

A study of the physics of energy and forces and their effects on bodies of mass.

morphing

An animated version of image warping, depicting the progressive deformation of images.

motion capture

Acquisition of motions directly from a live subject using devices attached to the subject. A computer automatically tracks these devices to obtain data describing the movement.

motion warping

The controlled deformation of an animated motion by modifying the timing, scale, and offsets of the movement [57].

motion synthesis

The creation of customized movements by mixing representative example motions taken from a database [33, 50, 54].

performance animation

Animation using motions obtained from a motion capture system.

pose-to-pose action

An approach to animating a sequence where by the essential extreme poses are drawn first and the in-between frames filled in later [34, 48].

procedural motion

Motion defined by combining and concatenating function-based equations [40].

relative phase

The fractional time into a locomotory cycle when a foot touches down and the associated leg becomes a stance leg [2].

secondary motion

Supplementary movements not required to achieve some primary motion goal, but rather affected by the primary motion.

slow-in, slow-out

An axiom of traditional animation is that motions near the critical keyframes should proceed slower than the less important transition motions of the in-between keyframes.

squash and stretch

A principle of traditional animation where objects exhibit elastic geometric deformations during motion, particularly at times of rapid acceleration [34, 48]. For example, a cartoon-like bouncing ball will compress when hitting the ground and elongate when flying through the air.

staging

This traditional animation fundamental encompasses the idea of planning and presenting an idea with clarity.

stance stage or *state*

The section of a stride in which a leg is a supporting leg.

stance or *supporting legs*

During a locomotory cycle, a leg that is planted on the ground, providing support for the body and propelling the figure onwards.

stride

One period of motion of a locomotory cycle.

stride frequency

The number of strides completed per unit time [2].

stride length

The distance covered from the centre of one footstep to the corresponding footstep of the same leg in the subsequent stride [2].

symmetrical gait

The corresponding left and right legs of a quadruped move through the same phases with the same relative timings, albeit some fraction of a cycle apart [2, 46]. Consequentially, the sequence of foot touchdowns alternate between the fore and hind legs.

timing

The traditional animation principle of timing is essential for conveying the urgency or scale of the motion of an object. [34, 48].

transport stage or *state*

The part of a stride in which a leg is swinging through the air.

visualization

Presentation of specific subsets of data in a form that is more easily analyzed or understood by a person.