Simulation of Human Diving

Wayne L. Wooten and Jessica K. Hodgins

College of Computing

Georgia Institute of Technology

Atlanta, GA 30332-0280

[wlw | jkh]@cc.gatech.edu

Abstract

In this paper we describe an animation of a human platform diver. We simulated the motion of the diver using a dynamic model and a control system. The dynamic model is a 32 degree-of-freedom rigid body model with dynamic parameters similar to those reported in the literature for humans. The control system uses algorithms for balance, jumping, and twisting to initiate the dive, proportional-derivative servos to perform the aerial portion of the dive, and a state machine to sequence the actions throughout the dive. The motion of the simulated diver closely resembles video footage of dives performed by human athletes. The combination of dynamic simulation and a control system allowed us to animate the diver using high level commands. The control and simulation techniques presented in this paper may be useful for analysis of sports performance and for providing realistic motion for synthetic actors in computer animation and virtual environments.

Keywords: Human Figure Animation, Simulation, Control Theory.

Introduction

Computer-generated animation of realistic human motion has been a goal of researchers in computer graphics for a number of years. In this paper, we explore dynamic simulation as a technique for generating animations of an Olympic sport, platform diving. The simulated diver performs three 10-meter platform dives: an inward $1\frac{1}{2}$ somersault pike, a reverse $3\frac{1}{2}$ somersault tuck, and a backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist. An image of the simulated diver is shown in figure 1. Direct comparison of the simulated diver with a human diver shows that
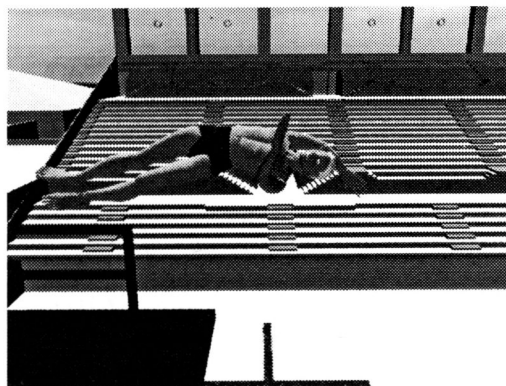

Figure 1: Graphical image of the diver in the flight phase of a backward $1\frac{1}{2}$ somersault pike with $\frac{1}{2}$ twist.

our system matches the details of human motion quite well. The system also allows interactive control of the parameters of the dive such as launch height, limb motion, and body orientation on the platform.

Animators have three techniques for generating human motion that appears realistic and natural-looking: keyframing, motion capture, and dynamic simulation. The techniques vary in the control given to the animator, in the realism of the generated motion, and in the ease of generalizing from one motion to another.

Keyframing allows the animator to control the motion of the animated figure at a low level by positioning body segments at certain key frames in the animation and interpolating the in-between frames automatically. Inverse kinematics can aid in this process by providing constraints for moving hierarchical linkages in an intuitive fashion. Keyframing gives the animator a fine level of control over the subtleties of motion; however, moving the joints and body segments so that the motion appears natural and physically realistic requires considerable skill on the part of the animator.

Digitization of human motion generates natural-looking and physically realistic motion because the animated motion is captured from a live actor. The animator no longer directly controls the motion, but instead modifies motion that has already been digitized. Errors in the captured data lead to flaws in the animated motion, particularly when the animated figure interacts with the environment. In addition, digitizing some motion, for example a 10-meter platform dive, is logistically difficult.

Unlike keyframing, which allows low-level control, and motion capture, which allows relatively little control, dynamic simulation enables the animator to generate motion using high-level control. The animator can request actions such as jump, squat, twist, or somersault and specify how the maneuver should be performed (the height of the jump, the depth of the squat, how far to twist). The simulation calculates the motion for the requested action. The details of the motion are computed automatically, and the animator is freed from the burden of specifying them. However, like motion capture, simulation prevents the animator from specifying the low-level details of the motion directly.

Dynamic simulation generates motion that is physically correct within the limits of the simulated model because the model takes into account the mass, inertia, and physical laws affecting the motion. Dynamic simulation of passive systems without an internal source of energy has been used successfully to animate natural phenomena such as colliding objects, splashing water, rising smoke, and blowing leaves (Barzel and Barr 1988; Hahn 1988; Terzopoulos and Fleischer 1988; Baraff 1989, 1991; Pentland and Williams 1989; Kass and Miller 1990; Wejchert and Haumann 1991; Stam and Fiume 1993; O'Brien and Hodgins 1995).

Dynamic simulation alone does not allow the computation of the motion of an active system with an internal source of energy, such as a human diver. For active systems, control algorithms must be used to compute the joint torques that will cause the simulated actor to perform the desired task. Genetic algorithms, dynamic programming, and other optimal control methods have been used to develop control algorithms for active systems (Witkin and Kass 1988; Brotmann and Netravali 1988; van de Panne, Fiume and Vranesic 1990; van de Panne and Fiume 1993; Ngo and Marks 1993; Sims 1994). To produce natural-looking motion, the control systems must mimic those used by humans, by avoiding excessive torques, extraneous motions, rapid accelerations, and other artifacts that make the motion appear unnatural (Bruderlin and Calvert 1989; Stewart and Cremer 1992; Hodgins, Sweeney, and Lawrence 1992; Hodgins 1994).

Background

Two areas outside of computer graphics provide material relevant to the simulation of diving: robotics and biomechanics. Although a robot with the degrees of freedom of a human body has not yet been built, the methods and techniques used in controlling robots in the laboratory are applicable to the simulation of more complex systems.

Raibert and his colleagues built a series of running machines that performed a number of dynamic tasks (Raibert 1986). Within this research, the control algorithms that allowed planar and three-dimensional two-legged robots to perform somersaults are the most relevant to diving (Hodgins and Raibert 1990; Playter and Raibert 1992). To initiate a somersault, the robot runs forward, thrusts with both legs, and pitches the body forward using the hip actuators. The robot shortens its legs during the somersault to increase angular velocity. The robot then lengthens its legs to land, lands on both feet, and continues running. The somersault is similar to a dive in that forces must be applied to the ground to generate the needed angular momentum for the ballistic part of the maneuver.

Murthy and Keerthi (1993) present an optimal control system for a two dimensional, four degree-of-freedom diver. They formulated a time-optimal control problem using state and control constraints and a numerical approach to compute the solution. The simulated diver performed both forward and backward somersaults. Solutions required about 10 minutes of computation on an Intel 486-based machine. Liu and Cohen (1994) also present a control method for diving that breaks the problem down into simpler and smaller subproblems. Their implementation required 0.7 seconds of computation on a Silicon Graphics R4000 workstation.

One area of biomechanics research is concerned with the techniques used by humans in performing aerial maneuvers. Results from this research provide insight into possible techniques for simulating the human diver. Yeadon and his colleagues analyzed human movement by recording the three-dimensional motion of aerial maneuvers with high-speed film and digitizing the resulting footage (Yeadon 1990; Yeadon, Athia, and Hales 1990). Yeadon also developed a dynamic model of the human body and a simulation system for aerial maneuvers. His system used inverse dynamics without a control system. In contrast, our diving simulation uses forward dynamics with a control system. Yeadon's data from film capture for three test subjects compared favorably with his simulation.

For many years, researchers have debated how cats land on their feet and how humans perform certain free-fall aerial maneuvers. Frohlich presents an enlightening discussion of the techniques used
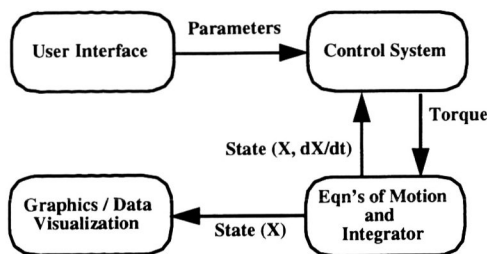
Figure 2: A diagram of the animation system used for the dynamic simulation of humans. The user controls the animation by providing parameters for the control system, such as the amount of twist during lift-off or the height of the jump. The control system uses this information to compute the torque that should be applied at the joints. The equations of motion are used to compute the acceleration for a given torque and state of the system. The velocity and position at the next time step are computed by integrating the acceleration. The system state is then used to draw the graphical image and provide feedback for the control system.

by humans for initiating somersaults and twists (Frohlich 1979). Through simulation and informal human experiments, he demonstrated that humans can perform somersaults and twists using torque generated by pushing on a platform. He also showed how they can perform similar maneuvers with no angular momentum from the platform.

Simulation of Human Diving

The system used to create the animation of a diving human consists of the equations of motion for a rigid-body model of a human, control algorithms for diving, a graphical display for viewing the motion, and a user interface for changing the parameters of the simulation (figure 2). The user directs the simulation by specifying desired characteristics for each phase of the dive. For example, the animator might specify when and how high the diver should jump from the platform. The details of the human model and control system are described below.

Model

The human diver is approximated by a rigid-body model consisting of 15 segments connected by 14 rotary joints with a total of 32 controlled degrees of freedom. Some joints, like the knee, are modeled as a single axis pin joint, others, like the shoulder and hip, are modeled by two and three axis gimbal joints. The volume, mass, center of mass, moments of inertia, and distance between the joints are calculated from a polygonal representation of the human body (figure 3, tables 1 and 2). The algorithm used to calculate the properties of the
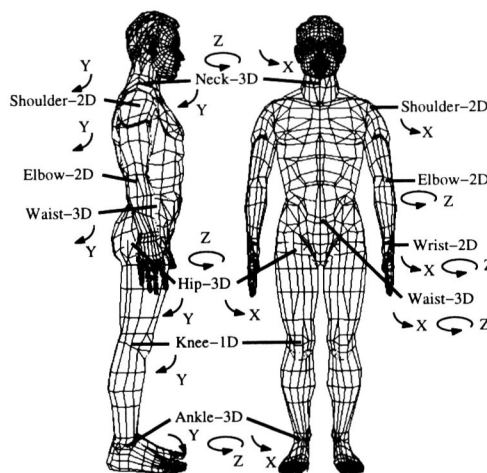

Figure 3: The controlled degrees of freedom of the human model. The number of degrees of freedom are shown at each joint. The direction of each arrow indicates the positive direction of rotation for each degree of freedom. The polygonal model was purchased from Viewpoint Datalabs.

polygonal model integrates over the set of tetrahedra formed by the triangular faces of the model and the origin (Lien and Kajiya 1984). Density data obtained from the anatomical literature were used in calculating the dynamic properties of the body segments (table 1). We assume that the density of each body part is uniform. The mass of the body parts, computed using the polygonal model, is similar to measurements from cadavers (Dempster and Gaughran 1965). We also tested the dynamic accuracy of the model by making it perform a backflip maneuver without the presence of angular momentum (Frohlich 1979). The back-flip caused our model to rotate 70°; Frohlich's model performed an 82° back-flip (figure 4).

The equations of motion for the model, were generated with a commercially available package that generates subroutines for the equations of motion employing a variant of Kane's method and a symbolic simplification phase (Rosenthal and Sherman 1986).

The dynamic interaction of the diver's feet with the platform are modeled using six constraints for each foot: two keep the metatarsus and heel above the surface of the platform, two prevent the foot from sliding left to right and front to back, and two prevent the foot from rolling or twisting about the point of contact. The linear and rotational acceleration of the contact point of the foot with respect to the platform is the constraint error. The penetration of the foot into the platform and the velocity of the foot relative to the platform are used to stabilize the constraint equations (Baumgarte 1972). The forces computed with the constraint matrix are

Link	Density (g/cm^3)	Mass (kg)	Moment of Inertia (x, y, z kgm^2)		
Trunk	1.008	29.27	0.73	0.63	0.32
Head	1.170	5.89	0.03	0.033	0.023
Pelvis	1.029	16.61	0.23	0.18	0.16
Thigh	1.040	8.35	0.15	0.16	0.025
Shank	1.078	4.16	0.055	0.056	0.007
Foot	1.066	1.34	0.0018	0.0075	0.007
Arm	1.067	2.79	0.025	0.025	0.005
Forearm	1.101	1.21	0.005	0.0054	0.0012
Hand	1.069	0.55	0.0016	0.002	0.0005

Table 1: Parameters of the rigid body model of a human. The moment of inertia is computed about the center of mass of each link. The densities are given in Dempster and Gaughran (1965).

Link	COM to Proximal (x, y, z m)			COM to Distal (x, y, z m)		
Trunk				to Head		
				.012	0	.32
Trunk				to Pelvis		
				.012	0	-.22
Trunk				to Arm		
				-.048	±.164	.12
Head	-.009	0	-.064			
Pelvis	.023	0	.103	.005	±.098	-.11
Thigh	.024	±.006	.120	-.052	±.019	-.21
Shank	.005	±.019	.165	-.002	±.009	-.25
Foot	-.046	±.009	.048			
Arm	-.0002	±.056	.120	-.005	±.036	-.17
Forearm	-.025	±.007	.090	.012	±.014	-.11
Hand	-.026	0	.085			

Table 2: The distance from the center of mass of each link to the proximal and distal joints in x, y, and z. The positive distance along the y axis refers to a location on the left side of the body; a negative distance refers to the right side. The z axis is vertical and the x axis is positive in the direction that the model is facing.

applied to the foot at the point of contact to prevent the foot from penetrating the ground or slipping on the ground. Torques are applied about the center of mass of the foot to prevent rotation. To allow the feet to leave the platform, the force is applied only when the foot has penetrated the platform a specified amount and its velocity is in the same direction as that of gravity. Friction is infinite in this simulation, so the diver cannot slip on the platform.

Constraints are also used to prevent the diver's joints from exceeding a user-specified minimum and maximum angle. The joint limits used in the diving simulation were taken from Heck, Hendryson, and Rowe (1964).

Control of Diving

To perform a platform dive, a human must take a number of control actions during the different phases of the dive. We reproduce this behavior in simulation by using a state machine to select the type of control needed for each phase of the dive: *compression*, *decompression*, *flight1*, *flight2*, and *entry*. Although the basic phases are common to all
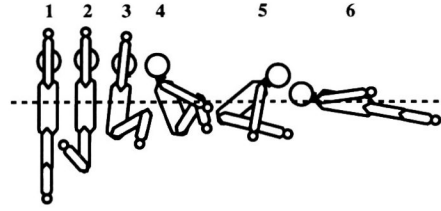

Figure 4: A back-flip can be performed by a human with no initial angular velocity by moving the limbs of the body in the order shown in this diagram.

dives, different dives require different control actions at each phase. For example, during *flight1*, a twisting diver activates control actions to twist at the waist, but a somersaulting diver bends at the waist.

The control laws for the three dives are presented in table 3. In the *compression* phase, the simulated diver bends his knees, hips, and ankles in preparation for the dive. In the *decompression* phase, the diver straightens his hips and knees, while pushing off the platform with his ankles. During *decompression* the arms swing to generate the appropriate angular velocity for a twisting or somersaulting dive. In *flight1* and *flight2*, the diver performs the diving maneuver. During the *entry* phase the diver straightens, puts his arms over his head, and enters the water.

During each dive, multiple levels of control are active. At the highest level, the state machine selects the phase of the dive to be performed. The transitions from one phase to the next are based on time. As each phase is selected, new desired values are set for lower level control systems. For example, desired values for the dives include the speed at which to raise the arms, how much to bend the knees in preparing for the jump, and the orientation of the head.

The middle level of control consists of a balance controller. This controller is active whenever the feet are in contact with the ground. Changes in the ankle and hip angles prevent the simulated diver from falling down even though the center of mass of the system is moving due to motion of the upper body and arms.

The center of mass of the diver, C, is projected onto the ground and compared with the desired center of mass, C_d. C_d is located in the center of a polygon defined by the contact points of the left and right feet. The equations used to derive desired positions for the ankle are

$$ankle_{x_d} = -(k_x(C_{x_d} - C_x) - kd_x\dot{C}_x) \quad (1)$$

$$ankle_{y_d} = k_y(C_{y_d} - C_y) - kd_y\dot{C}_y \quad (2)$$

where $ankle_{x_d}$ and $ankle_{y_d}$ are the desired angles for the x and y ankle joints and k_x and k_y are po-

State	Inward $1\frac{1}{2}$ somersault	Reverse $3\frac{1}{2}$ somersault	Backward $1\frac{1}{2}$ twist
Compression	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$ Swing arms behind back	Prepare for jump: Bend at knees $+y$ Bend at hips $-y$ Swing arms behind back
Decompression	Jump from platform: Bend at waist $-y$ Straighten hips Straighten knees Swing arms down Extend ankles	Jump from platform: Bend at waist $+y$ Straighten hips Straighten knees slightly Swing arms over head Extend ankles	Jump from platform: Bend at waist $+y$ Twist at waist $+z$ Straighten hips Straighten knees Swing arms forward Extend ankles
Flight1	Perform pike: Bend at hips $-y$	Perform tuck: Bend at hips $-y$ Bend at waist $-y$ Bend at knees $+y$ Bring arms down to knees	Perform twist: Untwist at waist $-z$ Twist at waist $+x$ Bring left arm over head Bring right arm across chest
Flight2		Perform tight tuck: Constrain hands to knees	Perform pike: Bring both arms to sides Untwist at waist $-x$ Bend at waist $-y$ Bend at hips $-y$
Entry	Prepare to enter water: Straighten hips Swing arms over head	Prepare to enter water: Straighten hips Straighten knees Straighten waist Swing arms over head	Prepare to enter water: Straighten hips Straighten waist Swing arms over head

Table 3: The state machine determines the control laws that are in effect at each phase of the dive.

sition gains for a controller that uses the error between the desired center of mass (C_{x_d}, C_{y_d}) and the actual center of mass (C_x, C_y) for feedback. This controller also reduces the velocity of the center of mass with the damping gains, kd_x and kd_y, and the velocity of the center of mass $(\dot{C}_x, \dot{C}_y)$. The equations are used for both the left and right feet.

Similar equations with different gains are used for calculating the desired position of the hips. Both the hips and ankles are used to control the left/right position of the center of mass. Using the ankles alone would cause the feet to lift off the ground if the diver moved too far from the center of support.

For some actions, such as jumping forward off the platform, the center of mass should be forward of the center of the polygon of support. In this case, a desired location for the center of mass is chosen and the hips are used to pitch the body and move the center of mass to the desired location. The ankles are not effective in performing this action because the metatarsus leaves the ground and the diver is unable to balance on his heels.

In preparation for jumping, the input to the balance controller is a desired height for the hips and a desired location for the center of mass. The knee is servoed to an angle that would produce the desired squatting height, the hips control the projected center of mass, and the ankles keep the feet on the ground and help in positioning the center of mass. If the center of mass leaves the polygon of support, the diver must either jump or fall down.

The lowest level of control takes the desired position of each joint, which has been calculated by the balance controller or provided by the user as part of the specification of the dive, and positions it using a proportional-derivative servo:

$$\tau = k_p(\phi_d - \phi) - k_v\dot{\phi} \qquad (3)$$

where τ is the torque computed for the joint, ϕ is the joint location, ϕ_d is the desired joint location, and $\dot{\phi}$ is the joint rotation rate. The gains of the proportional-derivative servo, k_p and k_v, were chosen empirically for each joint. A smooth trajectory is used to take the desired value from its current value to a newly computed desired value because eliminating step changes in the desired values reduces the jerk seen in the generated motion.

We simulated three dives using this control framework. Of the three, the inward $1\frac{1}{2}$ somersault pike (difficulty rating 2.0 (O'Brien 1992)) is the least difficult dive to control because the somersault does not require a large angular velocity and the diver has a large window of time to prepare for entry into the water. Graphs of the angular velocity for this dive are shown in figure 5. The angular velocity for the inward $1\frac{1}{2}$ somersault pike rises slowly, remains nearly constant for the duration of flight, and then drops when the diver untucks.

The reverse $3\frac{1}{2}$ somersault tuck (difficulty rating 3.4) is more difficult than the inward $1\frac{1}{2}$ somersault pike because the required angular momentum is larger and a very tight tuck is required during the flight phase to maintain the angular velocity. A graph of the angular velocity about the somersaulting axis for this dive is shown in figure 5. The rotation rate for this dive is much higher and is in
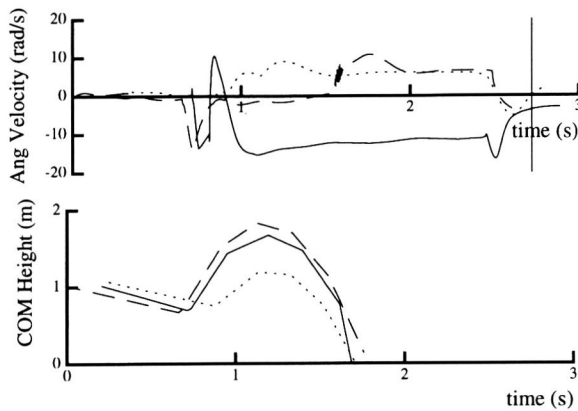
Figure 5: Graphs of the angular velocity about the somersaulting axis and the trajectory of the center of mass for three dives. The reverse $3\frac{1}{2}$ somersault tuck is represented by a solid line, the inward $1\frac{1}{2}$ somersault pike by a dotted line, and the backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist by a dashed line. The vertical line shows the point in time that the COM entered the water. The COM graph is truncated when the COM falls below the platform to allow the data to be scaled appropriately.

Figure 6: As a somersaulting diver brings one arm over the head and one arm to the side, a rotation about the y axis is induced. If the diver had non-zero angular momentum, a twist would be induced about the z axis in order to conserve angular momentum. Because of the rotation by θ, the body-local x axis is no longer aligned with the angular momentum axis and the body begins to twist.

Discussion

Images of the backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist performed by a simulated diver and a human diver are shown in the image sequence in figure 8. Although it is difficult to spot differences at a rate of 30 frames per second, single frame comparison makes clear that the Olympic athlete performs a better dive than the simulated human. The inward $1\frac{1}{2}$ somersault pike is performed in a similar fashion by the real and simulated divers, although the simulated diver pikes tighter at the top of the dive. Comparison of the reverse $3\frac{1}{2}$ somersault tuck reveals that the simulated diver somersaults more slowly than the Olympic athlete and is not vertical at entry. The backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist is performed differently by the simulated diver at the start of the dive. The simulated diver first twists in the opposite direction of that intended for the maneuver to gain enough twisting velocity to perform the dive. The Olympic diver has better form and does not need to turn in the wrong direction to initiate the twist.

the opposite direction from that seen in the inward and backward dive.

The backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist (difficulty rating 2.1) was the most difficult dive to control because it involves rotations about multiple axes. However, this dive is rated as just slightly more difficult than the inward $1\frac{1}{2}$ somersault pike for human divers. The backward $1\frac{1}{2}$ somersault with $\frac{1}{2}$ twist requires the generation of twisting torque at lift-off and the transfer of angular velocity about the twist axis to angular velocity about the somersaulting axis. The technique used to make the transition from a twist to a somersault is presented in Frohlich (1979). A diver having non-zero angular momentum and performing a somersault about the x axis can initiate a twist by throwing the left arm down and the right arm up. This motion results in a counterclockwise rotation of the diver about the y axis. This rotation causes a twisting motion about the z axis because the diver's principal axes of rotation are no longer aligned with their angular momentum axis, and the angular velocity vector now has components along both the x and z axes. Figure 6 illustrates this technique, and figure 7 shows the rates of rotation about the three axes and the location of the diver over time for the twisting dive. Angular velocity is highest about the twisting axis during the first half of the dive but decreases as the angular velocity about the somersaulting axis increases. Figure 7 also shows the movement of the right hip and shoulders.
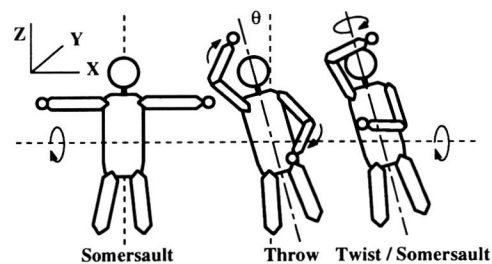
Although the motion produced by the simulation is physically realistic and natural-looking, many simplifying assumptions were made in the model. We assumed that air resistance was negligible and did not apply drag to the body as it fell through the air. This assumption does not affect the motion substantially because the diver falls at a maximum velocity of 15 m/s; at that velocity air resistance does not slow the diver significantly (Van Gheluwe 1981). We also assumed that the density of each body part of the dynamic model was uniform and that the joints were simple revolute joints. Despite the assumption of uniform density, the moments of inertia are similar to the data obtained from humans by Dempster and Gaughran (1965). The assumption of simple revolute joints influences the resulting motion, and a more accurate model of the shoulder
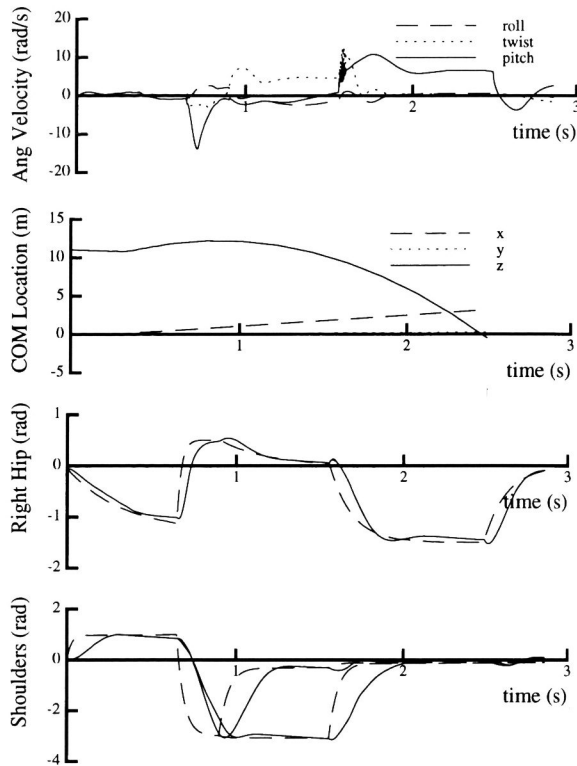
Figure 7: Graphs of the location of center of mass, angular velocity, and selected joint motion for the $\frac{1}{2}$ twist dive. The dashed lines in the lower three graphs represent the desired location of a joint, and the solid lines represent the actual location.

and spinal column, such as that used in the Jack system (Badler 1993), would make the motion look more natural. Our simulation is also missing the subtle secondary motion of such subsystems of the human body as clothing, hair, and skin that move in response to the motion of the human body.

Our model of the muscles is very simple—a torque source at each joint. The strength of individual joints was not taken into account, and this simplification could result in a simulated human that is stronger or faster than a real human. For example, a simulated diver might perform a dive that met the specifications, in that the diver tucked, opened, and entered the water vertically, but the performance might lack the grace and style of a human dive because the movement of the limbs was jerky. The absence of a strength model also means the simulated diver could perform dives using strategies that are impossible for humans.

After lift-off, our control system uses fixed positions for the desired values and fixed time intervals for the state changes. The control system does not use feedback to adapt those constants as the dive progresses. If the animator adjusts the gains or

desired positions of joints at the beginning of the dive the control system will not compensate for the change and the simulation may perform the dive incorrectly. A control system that took action to adjust the amount of rotation during flight would make the diving control system more robust in response to changes in the dynamic model and desired behavior. Playter and Raibert (1992) developed a control system that alters rotation during flight in somersaulting robots by servoing the length of the legs. This control strategy could be adapted to diving. However, the sensor requirements for such a control system might exceed the perceptual information available to a human performing diving maneuvers.

Physical simulation does not guarantee natural-looking motion, yet the motion of the divers appears realistic even when side-by-side comparisons are made with video footage. We believe that one reason why the motion looks natural is that diving is a very dynamic activity. Once the diver has left the board, his angular momentum is constant and he has only to move his limbs in the right sequence and with the right timing to enter the water vertically. The length of the animation is only a few seconds, and most of the motion occurs very quickly, making small errors in the motion hard to spot. The diving motion is by nature acyclic, and the entry phase of the dive requires only that the diver be nearly vertical. Small errors in the oscillations are much easier to see in cyclic behaviors, as they are repeated on each cycle. Finally, integration of natural elements such as splashing water (O'Brien and Hodgins 1995) as the diver enters the pool lends credibility to the diving motion.

Realistic simulation of human motion will be useful in entertainment and virtual environments. Using simulation in a virtual environment requires that the simulation run in real-time. Currently the simulations run 21 times slower than real-time on a Silicon Graphics Indigo2 150Mhz R4400 workstation. Significant speed improvements are needed before these algorithms can be used in interactive environments.

We believe that simulated motion of athletic endeavors has the potential to be useful in improving athletic performance and in human motion studies. When analyzing a particular motion, athletes and coaches could change parameters in a simulation of a human performing their sport. This ability would allow novice athletes, who might not understand how the motion of a particular body part affects their performance, to experiment and ask such questions as, "What would happen if I tucked tighter in this dive?" Interactive simulations could give both coaches and athletes better intuition about the physics involved in their sport and could lead to improved human performance.
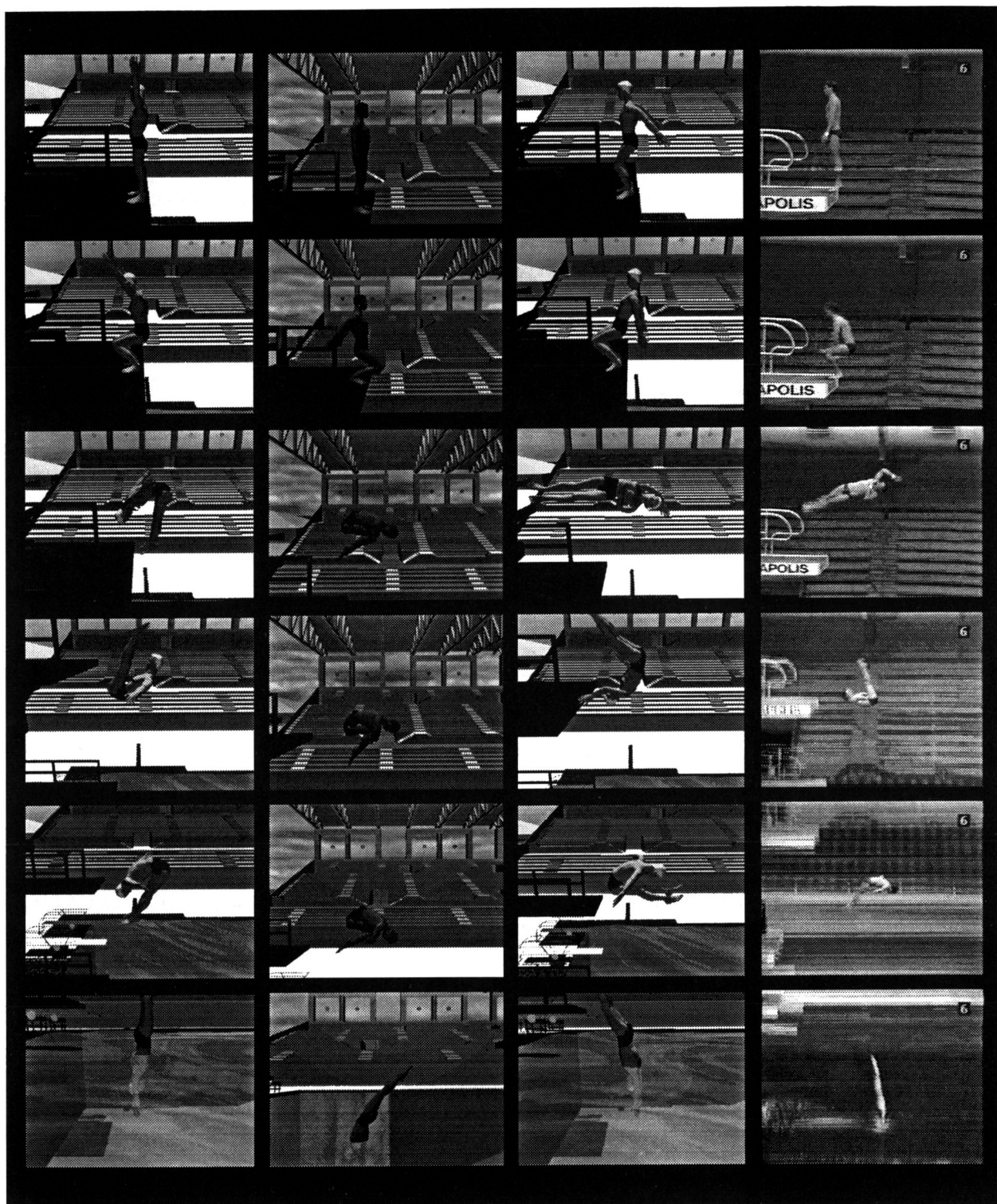
Figure 8: A sequence of images from the three simulated dives. The inward $1\frac{1}{2}$ somersault pike is on the left and the reverse $3\frac{1}{2}$ somersault tuck is shown in the second column. The third and fourth columns show the backward $1\frac{1}{2}$ somersault pike with $\frac{1}{2}$ twist as performed by the simulated diver and by an Olympic athlete. Each image in each sequence is separated by 0.5 seconds; time increases from the top to the bottom image in each column. In the reverse $3\frac{1}{2}$ somersault tuck, the diver is at the beginning of each somersault in frames 3, 4, and 5.

Acknowledgments

References

Badler, N. I., Phillips, C. B., Webber, B. L. 1993. *Simulating Humans*. Oxford: Oxford University Press.

Baraff, D. 1989. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. *Computer Graphics*, 23(3):223–232.

Baraff, D. 1991. Coping with Friction for Non-penetrating Rigid Body Simulation. *Computer Graphics*, 25(4):31–40.

Barzel, R., Barr, A. H. 1988. A Modeling System Based on Dynamic Constraints. *Computer Graphics*, 22(4):179–188.

Baumgarte, J. 1972. Stabilization of Constraints and Integrals of Motion in Dynamical Systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16.

Brotman, J. S., Netravali, A. N. 1988. Motion Interpolation by Optimal Control. *Computer Graphics*, 22(4):309–315.

Bruderlin, A., Calvert, T. W. 1989. Goal-Directed, Dynamic Animation of Human Walking. *Computer Graphics*, 23(3):233–242.

Dempster, W. T., Gaughran, G. R. L. 1965. Properties of Body Segments Based on Size and Weight. *American Journal of Anatomy*, 120:33–54.

Frohlich, C. 1979. Do Springboard Divers Violate Angular Momentum Conservation? *American Journal of Physics*, 47:583–592.

Hahn, J. 1988. Realistic Animation of Rigid Bodies. *Computer Graphics*, 22(4):299–308.

Heck, C. V., Hendryson, I. E., Rowe, C. R. 1964. Pamphlet for the Study of Joint Motion. Published by the Executive Committee of the American Academy of Orthopaedic Surgeons, 80–86.

Hodgins, J., Raibert, M. H. 1990. Biped Gymnastics. *International Journal of Robotics Research*, 9(2):115–132.

Hodgins, J. K. 1994. Simulation of Human Running. *IEEE International Conference on Robotics and Automation*, 1320–1325.

Hodgins, J. K., Sweeney, P. K, Lawrence, D. G. 1992. Generating Natural-looking Motion for Computer Animation. *Proceedings of Graphics Interface '92*, 265–272.

Kass, M., Miller, G. 1990. Rapid, Stable Fluid Dynamics for Computer Graphics. *Computer Graphics*, 24(4):49–57.

Lien, S., Kajiya, J. T. 1984. A Symbolic Method for Calculating the Integral Properties of Arbitrary Nonconvex Polyhedra. *IEEE Computer Graphics and Applications*, 4(5):35–41.

Liu, Z., Cohen, M. 1994. Decomposition of Linked Figure Motion: Diving. *5th Eurographics Workshop on Animation and Simulation*.

Murthy, N., Keerthi, S. 1993. Optimal Control of a Somersaulting Platform Diver. *IEEE International Conference on Robotics and Automation*, 1013–1018.

Ngo, J. T., Marks, J. 1993. Spacetime Constraints Revisited. *Computer Graphics*, 343–350.

O'Brien, R. 1992. *Ron O'Brien's Diving for Gold*. Champaign, Illinois: Leisure Press.

O'Brien, J. F., Hodgins, J. K. 1995. Dynamic Simulation of Splashing Fluids. *Computer Animation '95*, in press.

Pentland, A., Williams, J. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. *Computer Graphics*, 23(3):215–222.

Playter, R. R., Raibert, M. H. 1992. Control of a Biped Somersault in 3-D. *IEEE International Conference on Robotics and Automation*, 582–589.

Raibert, M. H. 1986. *Legged Robots That Balance*. Cambridge: MIT Press.

Rosenthal, D. E., Sherman, M. A. 1986. High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method. *Journal of Astronautical Sciences*, 34(3):223–239.

Sims, K. 1994. Evolving Virtual Creatures. *Computer Graphics*, 15–22.

Stam, J., Fiume, E. 1993. Turbulent Wind Fields for Gaseous Phenomena. *Computer Graphics*, 369–376.

Stewart, A. J., Cremer, J. F. 1992. Beyond Keyframing: An Algorithmic Approach to Animation. *Proceedings of Graphics Interface '92*, 273–281.

Terzopoulos, D., Fleischer, K. 1988. Modeling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. *Computer Graphics*, 22(4):269–278.

van de Panne, M., Fiume, E., Vranesic, Z. 1990. Reusable Motion Synthesis Using State-Space Controllers *Computer Graphics*, 24(4):225–234.

van de Panne, M., Fiume, E. 1993. Sensor-Actuator Networks. *Computer Graphics*, 335–342.

Van Gheluwe, B. 1981. A Biomechanical Simulation Model for Airborne Twist in Backward Somersaults. *Journal of Human Movement Studies*, 7:1–22.

Wejchert, J., Haumann, D. 1991. Animation Aerodynamics. *Computer Graphics*, 25(4):19–22.

Witkin, A., Kass, M. 1988. Spacetime Constraints. *Computer Graphics*, 22(4):159–168.

Yeadon, M. R. 1990. The Simulation of Aerial Movement-I,II,III. *Journal of Biomechanics*, 23(1):59–83.

Yeadon, M. R., Athia, J., Hales, F. D. 1990. The Simulation of Aerial Movement-IV. *Journal of Biomechanics*, 23(1):85–89.