Modeling and Animating Myriapoda: A Real-Time Kinematic/Dynamic Approach

Jingyi Fang* Chenfanfu Jiang† Demetri Terzopoulos‡
Computer Science Department
University of California, Los Angeles

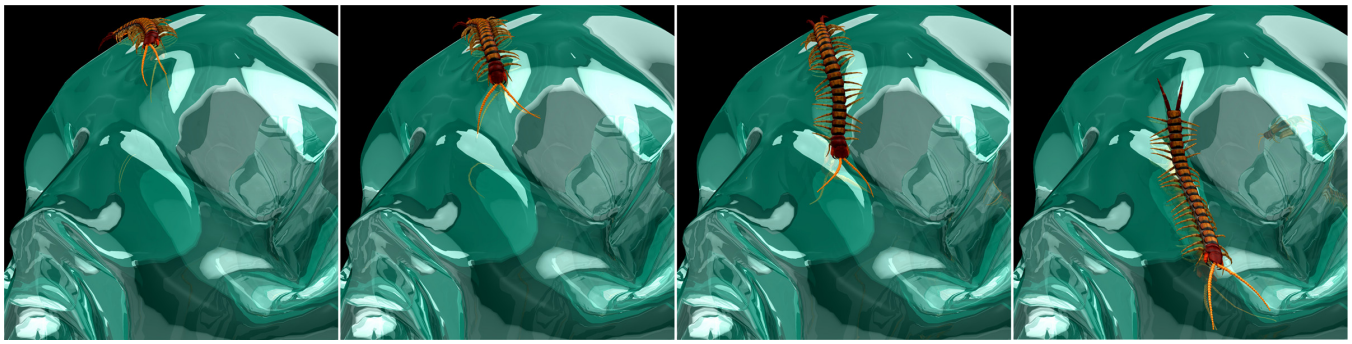

Figure 1: *A simulated centipede walking autonomously over a crystal skull.*

Abstract

Unlike two, four, six, and eight legged animals, Myriapoda—i.e., centipedes, millipedes, etc.—have been largely overlooked in the computer graphics literature. We present an artificial life framework for modeling these arthropods and animating their locomotive behavior over regular or irregular surfaces in real time with compelling physical and biological realism. Our hybrid approach combines kinematic and dynamic simulation, as well as a decentralized, distributed leg control system whose emergent behavior is suitable for animating simulated myriapoda of different morphologies with the characteristically vivid leg wave patterns of their biological counterparts. The simulated creature's antennae sense its virtual environment and the sensory information guides its adaptive behaviors, including obstacle avoidance and foraging.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: Myriapoda, Centipedes, Millipedes, Physics-Based Modeling, Behavioral Animation, Artificial Life

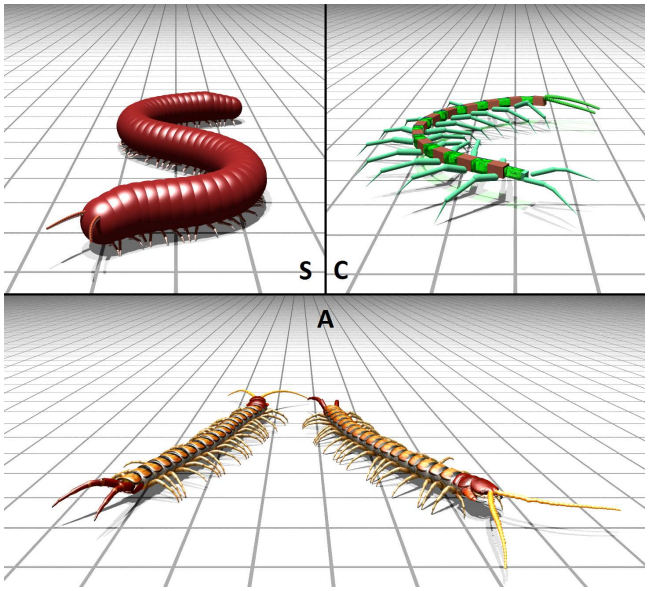

Figure 2: *Different simulated myriapoda forming the letters SCA.*

1 Introduction

Many movies and games can make use of loathsome creatures such as spiders and centipedes to rouse the viewer.[1] Entomophobia therapy can also take advantage of lifelike animations of such creatures. Motion capture techniques are difficult to apply to multi-legged arthropods such as centipedes [Gibson et al. 2007]. It is a daunting task to manually rig a realistic 3D centipede model such that it can locomote over an irregular surface, since at least two requirements must be met, (i) the physical realism of the deformable body and leg contacts—the legs must land precisely on the surface—and (ii) the natural appearance of the distinctive leg wave pattern. Advanced

[1] For example, the *Indiana Jones*, *Harry Potter*, *The Mummy*, *Metro 2033*, and *Diablo III* series of games and motion pictures.

*e-mail:ffang@cs.ucla.edu
†e-mail:cffjiang@cs.ucla.edu
‡e-mail:dt@cs.ucla.edu

SCA 2013, July 19 – 21, 2013, Anaheim, California.

simulation techniques are therefore desirable for the realistic synthesis of the locomotion patterns of such creatures.

In biology, the centipede is but one of the 13,000 identified species that form the Myriapoda subphylum of arthropods. Their primary characteristic is that they have tens or even hundreds of legs distributed over an elongated, segmented body structure, plus a pair of sensory antennae on the frontmost segments [Ruppert et al. 2003]. During its locomotion, the wave pattern of a myriapod's legs can steadily move the entire body over highly irregular surfaces. If in the legged animal kingdom we compare four-legged animals to cars, then myriapoda are like trains. Because of their unusual structure and unique locomotion systems, they are an interesting subject both for robotics [Hoffman and Wood 2011] and graphics research.

To tackle the task of synthesizing realistic animations of myriapoda as well as to gain a better understanding of how to coordinate hundreds of legs, we take a bottom-up, artificial life approach (see, e.g., [Terzopoulos 1999]) to synthesizing autonomous virtual myriapod creatures. Our work makes three major contributions:

1. We devise a novel biomechanical body structure suitable for myriapoda, which comprises both rigid and deformable segments, thus ensuring ease of control through the rigid components while emulating the deformable nature of myriapod bodies by applying a fast, robust elasticity simulation method.

2. We develop a biologically plausible decentralized and distributed leg control system that is suitable for different myriapod morphologies and enables our simulated creatures to locomote over arbitrary surfaces with an emergent, naturally wave-like ambulatory gait (see Figs. 1, 2, and 13)

3. To achieve realistic real-time animation of myriapoda, we develop a hybrid simulation method that incorporates both kinematics and dynamics in different body states. Our hybrid approach in conjunction with the rigid/deformable body structure enables efficient kinematic control of the relatively light legs of myriapoda while achieving realistic, physics-based motion and deformation of their relatively heavy bodies.

Fig. 3 shows an overview of our framework, emphasizing the interplay between the locomotion control system and the physical simulation. The former is composed of identical, modular, local leg controllers plus a motor center in the creature's head segment that governs higher-level locomotion variables such as ambulatory speed and turning angle. The decentralized leg control system is easy to understand and computationally inexpensive, yet robust enough to allow the simulated creature to walk over irregular surfaces. A pair of mobile antennas enhance the realism of our animated myriapoda, but also function as environmental sensors that support their adaptive behaviors. A simple Braitenberg vehicle [Braitenberg 1986] like behavioral mechanism proves to work well for obstacle avoidance and foraging. By positioning obstacles and food sources, animators can easily induce our simulated myriapod creatures to follow desired paths over irregular terrain.

The remainder of the paper is organized as follows: Section 2 surveys relevant prior work on autonomous virtual creatures, biomechanical modeling, and legged locomotion and control. Section 3 presents our hybrid, kinematic/dynamic model of myriapod bodies. In Section 4, we introduce our locomotion control system for myriapoda. Section 5 explains how our simulated myriapoda interact with their environment. We describe our implementation and present results in Section 6. Section 7 concludes the paper with a discussion and ideas for future work.

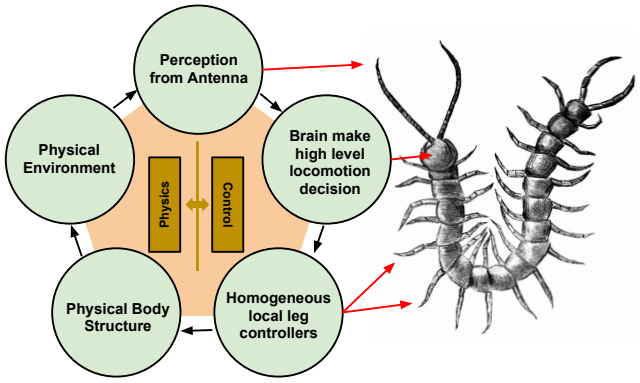

Figure 3: *Overview of our simulation framework. The physical environment and controller influence each other through antenna and contact sensors.*

2 Related Work

Autonomous Virtual Creatures: Graphics researchers have done interesting and important work on simulating autonomous creatures in physics-based virtual environments. In [Miller 1988], mass-spring-damper systems are used to generate animation of snakes and worms. Tu and Terzopoulos [1994] introduced an artificial life framework for simulating autonomous creatures, including mass-spring-damper systems for the biomechanical simulation of deformable piscine bodies capable of producing muscle-based locomotion. Such biomechanical models are suitable for animals lacking legs. Subsequent work on salamander simulation [Ijspeert et al. 2007] used rigid-body dynamics to simulate the 4 legs with 8 kinematic degrees of freedom and a global artificial neural network to work as a central pattern generator (CPG) for controlling them. Chiel and Beer [1990] built a 2D cockroach equipped with simple sensory feedback and artificial neural networks, demonstrating six-legged locomotion and adaptive behaviors in a complex environment and they also built a hexapod robot [Beer et al. 1992].

Biomechanical Modeling: Because of their simplicity and computational efficiency, mass-spring-damper systems were a popular method for biomechanical modeling of virtual creatures [Tu and Terzopoulos 1994; Miller 1988]; however, their uniaxial elements do not model 3D material properties, and the biomechanics of the simulated animal body depends on how the spring-damper elements are assembled and how their parameters are tuned, which can be tricky. Alternatively, robust and efficient simulation of continuum mechanics based deformable objects is now possible [Stomakhin et al. 2012; McAdams et al. 2011]. Shinar *et al.* [2008] create realistic creatures by combining rigid and elastic simulation, where the elastic components are passive and the creature is actuated by internal rigid bones. In [Coros et al. 2012], deformable objects are animated by changing the rest shape of the deformable mesh over time. State-of-the-art muscle models [Chen and Zeltzer 1992; Teran et al. 2003] also employ continuum mechanics based elasticity. Our work on myriapoda incorporates elastic and rigid simulation to achieve hybrid, kinematic/dynamic simulation.

Legged Locomotion and Control: Graphics and robotics researchers have devoted considerable effort to legged locomotion [Raibert and Hodgins 1991; Golubitsky et al. 1998; Wang et al. 2012]. Holmes *et al.* [2006] survey the modeling, analysis, and challenges associated with insect locomotion dynamics. There has been much robotics research on insect-inspired four, six, and eight legged robots [Kimura et al. 2007; Raibert 2008; Saranli et al. 2001], predominantly based on etiologic and neurophysiologic

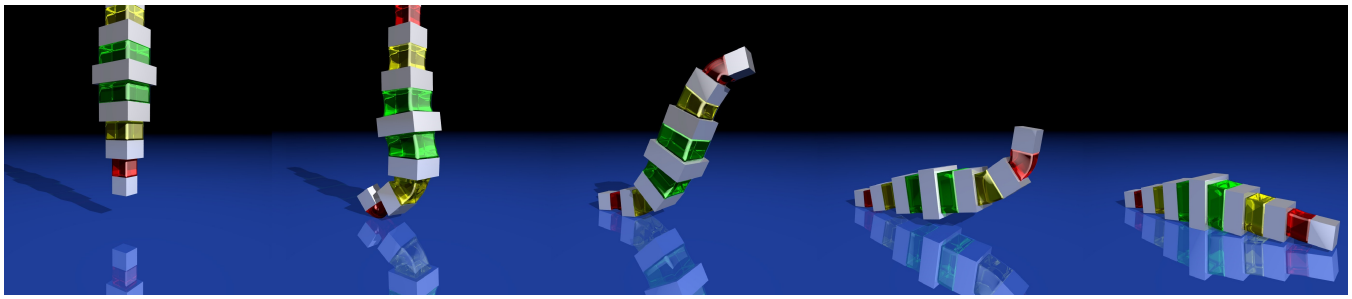

Figure 4: *A dynamic structure with alternating, coupled rigid and deformable segments dropping onto a slippery ground plane.*

knowledge about cockroaches and stick insects [Full and Tu 1991]. Although there exists no prior computer graphics work specifically on myriapoda animation, robotics researchers have started building centipede robots [Odashima et al. 1998; Inagaki et al. 2003; Inagaki et al. 2011]. However, there exists a substantial body of graphics literature on bipedal [Hodgins et al. 1995; Faloutsos et al. 2001; Van Welbergen et al. 2010], quadrupedal [Skrba et al. 2008; Coros et al. 2011], and hexapodal [Cenydd and Teahan 2013; McKenna and Zeltzer 1990] figure animation. McKenna and Zeltzer [1990] proposed a forward dynamics algorithm for locomotion coordination of a simulated cockroach, capable of navigating irregular terrain. Attention has been paid to legged locomotion control from entomological studies by building accurate biomechanics models of leg muscles [Wang et al. 2012] with the purpose of accurately simulating ambulation for medical purposes. Specialized neural CPG structures have been the subject of experiments [Mellen et al. 1995] and simulations [Ijspeert et al. 2007]. Coupling simplified dynamics such as the Inverted Pendulum Model with a learnt CPG has been a popular approach for multi-legged locomotion [Coros et al. 2011; Tsai et al. 2010]. The Walknet of Cruse *et al.* [2000; 1998] successfully reproduced the behavioral properties of hexapod locomotion via a decentralized organization of the control system—neither the movement of any single leg nor gait coordination is centrally preprogrammed, yet adaptivity and flexibility emerge from each leg controller applying only a few simple, localized rules involving the states of neighboring legs. This decentralized controller paradigm also forms the basis of our system design.

3 Physical Model

Prior work on animating arthropods has been limited to insects with only one major abdominal segment, such as a stick insect or a spider. The unique, elongated and deformable body structures of myriapoda require special attention to the biomechanics. The numerous slim legs of typical myriapoda have much lower mass than does the remainder of their body, which enables us to take an efficient hybrid kinematic/dynamic approach to modeling them in order to achieve real-time animation performance.

Fig. 5 illustrates the physical body structure that we developed for our artificial myriapoda. First, we simplified our model by neglecting the mass of the legs, thus making them kinematic. Initially, we used only deformable segments to model the body. However, coupling the slim legs to the deformable segments significantly increased the control challenge, since the leg attachments have a very local effect on the deformable body. By contrast, connecting a leg to a rigid segment emulates the natural exoskeleton of millipedes and enables the leg to easily control the position and orientation of the entire segment.

Thus, in our model, a pair of rigid, kinematic legs actively control each of the rigid dynamic exoskeletal segments, which are then

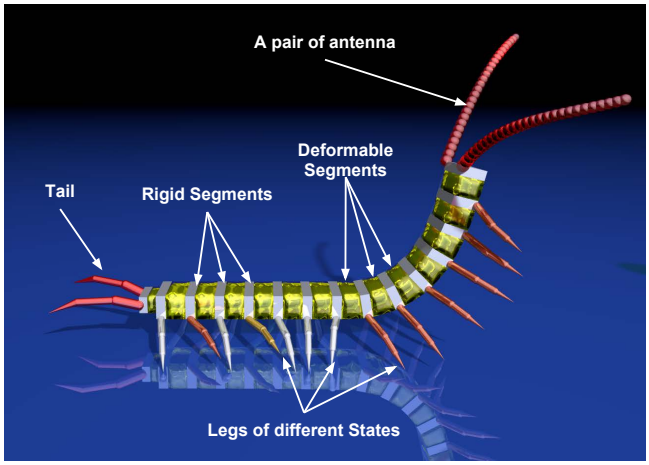

Figure 5: *Myriapod body structure, comprising rigid segments with rigid legs, deformable segments, deformable antennas, and tail. The color of each leg indicates its current state (see Fig. 13).*

connected in series using passively dynamic elastically deformable segments. When hanging in gravity, the body model will naturally elongate (Fig. 10). The motion of the rigid segments is governed by rigid-body dynamics with collision and friction [Baraff 1997]. The deformable segments are governed by energetically consistent invertible elastodynamics with a fixed corotational constitutive model [Stomakhin et al. 2012]:

$$\Psi = \mu \sum_i (\sigma_i - 1)^2 + \frac{\lambda}{2}(J-1)^2, \qquad (1)$$

where μ and λ are Lamé parameters, J is the determinant of the deformation gradient $\mathbf{F}$, and σ_i are the singular values of $\mathbf{F}$. Appendix A presents the details of the deformable segments and their simulation using the finite element method.

Several authors have demonstrated two-way coupling between rigid and deformable objects [Shinar et al. 2008; Sifakis et al. 2007; Baraff and Witkin 1997]. We couple the rigid and deformable parts of our alternating rigid/deformable myriapod body structure using the following staggered simulation method:

1. Boundary nodes on deformable tetrahedral meshes that make contact with rigid segments are fixed to those rigid segments.
2. At each simulation time step, the elastic forces from the mesh deformation computation evaluated at the contact nodes are applied as external forces to their respective rigid bodies.
3. Finally, the positions of the contact nodes are updated in accordance with the rigid body dynamics simulation.

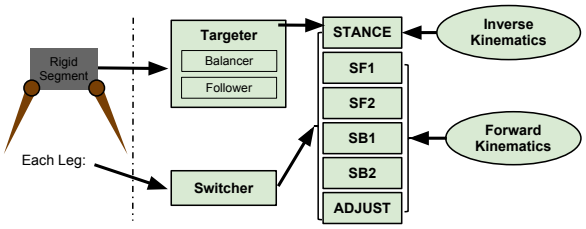

Figure 6: *Overview of the local locomotion controller.*

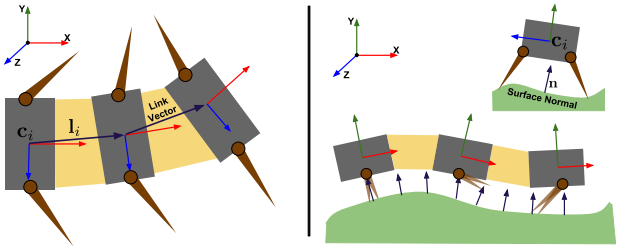

Figure 7: *The Targeter of a rigid segment, comprising a Follower (left) and a Balancer (right).*

Since the deformable and rigid parts time-stepped independently, our staggered method is not fully coupled as in [Shinar et al. 2008]; however, by virtue of the robust, invertible elasticity method of [Stomakhin et al. 2012], stability is not an obvious issue. Our approach achieves two-way coupling yet is very straightforward to implement. Fig. 4 demonstrates the simulation of a coupled rigid-deformable object.

The legs of the myriapod creature are simulated as rigid links that rotate around joints. The leg tip and root positions are inputs and outputs for our locomotion control system and the leg joint angles are determined by an inverse kinematics (IK) solver. Although it falls short of full biomechanical simulation and control, our efficient method meets the visual realism requirements. Furthermore, given the numerous legs involved, the animation of myriapoda does not present anywhere near as severe an "Uncanny Valley" to observers as does human bipedal locomotion. Hence, rather than trying to improve the realism of individual leg movement by modeling complex leg biomechanics, and pay the associated computational costs, we have focused on the development of the locomotion control system described in the next section, which can realistically synthesize the natural wave motion pattern of the numerous legs.

4 Myriapod Locomotion System

The Dragon Dance, a tradition that is performed during Spring Festivals in Chinese culture, requires approximately 9 to 15 dancers to control a long dragon whose segments are connected by joints. Three key observations can be made about this dance: First, the person controlling the head of the dragon makes the locomotion decisions and implements these decisions using his/her own two legs. Second, each of the remaining performers have one major locomotion goal—to follow the performer ahead of them. Third, applying the previous two simple local rules results in a global emergent behavior—the global wave pattern of the dragon movement.

The leg control model that we have developed for our myriapod creatures applies the idea of the head segment leading and the subsequent segments following. At any moment, the head segment synthesizes information sensed by the antennas to make high-level locomotion decisions, such as turning and changing speed. These decisions are processed by the head segment to determine the desired future configuration of the head; specifically, the position and orientation of the rigid head segment in the next time step. The desired configuration is provided to the leg controller for execution. The subsequent segments have identical local leg controllers, which comprise two components shown in Fig. 6—(i) a *Targeter* that determines the desired target configurations of the segment and (ii) left and right leg *Switchers* that switch the leg states (Fig. 8).

4.1 Rigid Segment States

A rigid segment can have two states: *supported* and *unsupported*. When the segment is supported, it is kinematically transported to the desired position and orientation by the legs. In the unsupported state, its motion is governed by the rigid-deformable coupling dynamics. The major benefit of this hybrid, kinematic/dynamic approach is that it circumvents complex leg dynamics to drive the body dynamics without sacrificing appreciable dynamical realism in the body. The legs are rotated via forward kinematics in most states and they control the segment through inverse kinematics during the stance state. The details of the six leg states and the switching mechanism are discussed in the following sections.

Targeter: A Targeter in each rigid segment i continually outputs its configuration (position $\mathbf{c}_i$ and orientation $\mathbf{R}_i$) to the legs. When the segment is unsupported, it outputs the updated configuration computed by dynamic simulation, whereas when the segment is supported, it outputs the desired configuration of the rigid segment for the segment's legs to achieve kinematically. As is described later, the left and right leg can be synchronized in order to maximize the time that a rigid segment is in the supported state. The Targeter has two objectives (Fig. 7)—to follow the previous segment, which is accomplished by a *Follower*, and to balance the body, which is accomplished by a *Balancer*. The Follower generates a target position $\mathbf{c}_i^T$ for the current rigid segment by modifying its current position in the direction of the link vector $\mathbf{l}_i = \mathbf{c}_{i+1} - \mathbf{c}_i$ such that the length of the link vector remains constant and it generates a target rotation $\mathbf{R}_i^T$ of the segment around its y axis so as to orient the x axis toward the link vector direction. The Balancer balances the y axis of the rigid segment toward ground surface normal vector $\mathbf{n}$ and adjusts the elevation of the rigid segment by further modifying the target position in the direction of $\mathbf{n}$. The combined adjustments of the current position and orientation by the Follower and Balancer result in a new target position and orientation that are outputted to the leg controllers as goals to achieve during the stance state.[2]

4.2 Leg State Machine

Leg States: Typically, a leg will periodically cycle between 6 different states (Fig. 8) under the control of a Switcher. Referring to Fig. 9, a pose is uniquely defined by three rotational angles: θ, α, and β (γ is fixed). Angle θ is the rotation of the leg plane around the y axis of the rigid segment, while α and β are the 2 degrees of freedom of the leg within the leg plane. Each leg state starts in one pose and ends at a target pose.

[2]With the target position $\mathbf{c}_i^T$ and orientation $\mathbf{R}_i^T$ of segment i initialized to its current position and orientation, the update equations for the Follower are $\mathbf{c}_i^T \stackrel{\pm}{=} k_1(|\mathbf{l}_i| - l_i^0)\mathbf{l}_i\Delta t$ and $\mathbf{R}_i^T \stackrel{*}{=} \mathbf{R}([0,\pm1,0],\omega\Delta t)$, with the $\pm$ sign determined by the sign of the projection of $\mathbf{l}_i$ onto the z axis, and the update equations for the Balancer are $\mathbf{c}_i^T \stackrel{\pm}{=} k_2(h_i - h_i^T)\mathbf{n}\Delta t$, where h_i and h_i^T are the segment's current height and target height, and $\mathbf{R}_i^T \stackrel{*}{=} \mathbf{R}([0,0,\pm1],\omega\Delta t)\mathbf{R}([\pm1,0,0],\omega\Delta t)$, with the $\pm$ signs determined by the projection of $\mathbf{n}$ onto the x-z plane.

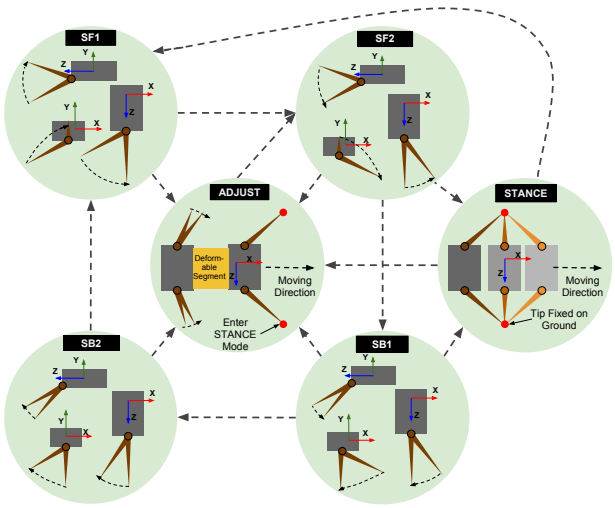

Figure 8: *The six leg states and transitions.*

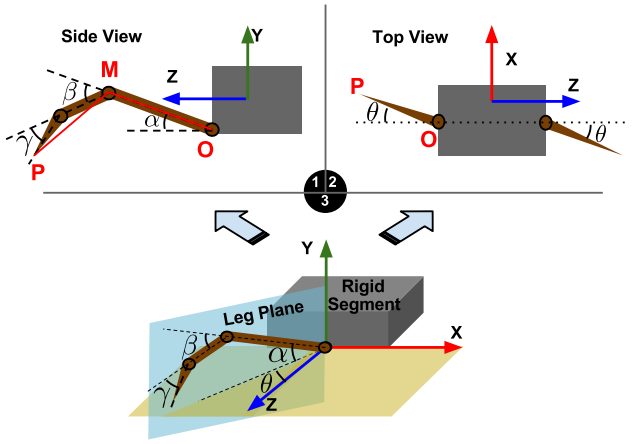

Figure 9: *Leg structure. The leg rotates around the y axis (θ) and can articulate in the leg plane around joints O and M (α, β). P is the tip of the leg.*

- **Sway Forward 1 (SF1):** Ideally, the leg sways forward up from the posterior extreme pose (PEP) ($\theta = \theta_m$, $\alpha = \alpha_S$, $\beta = \beta_S$) to the middle up pose (MUP) ($\theta = 0$, $\alpha = \alpha_M$, $\beta = 0$). This state allows the leg to leave the STANCE state and elevate off the ground.
- **Sway Forward 2 (SF2):** Ideally, the leg sways forward down from the MUP to the anterior extreme pose (AEP) ($\theta = \theta_M$, $\alpha = \alpha_S$, $\beta = \beta_S$). This state sends the leg to the STANCE state;
- **Sway Backward 1 (SB1):** Ideally, leg sways backward down from the AEP to the middle down pose (MDP) ($\theta = 0$, $\alpha = \alpha_m$, $\beta = 0$). This is to help the leg touch the ground.
- **Sway Backward 2 (SB2):** Ideally, the leg sways backward up from the MDP to the PEP. This is the last state of a cycle comprising SF1, SF2, SB1 and SB2.
- **STANCE:** In the stance state, the leg tip is fixed on the ground and the root of the legs push the segment forward. Ideally, it starts at the PEP and ends at the AEP. Inverse kinematics is used to compute the rotational angles (θ, α, and β) from the tip and root positions, as is detailed in Appendix B. The root position of the leg is calculated from the output of the Targeter. The leg will leave the STANCE state to enter SF1 when the IK solver cannot resolve the current outputted root position from the Targeter.
- **ADJUST:** This state generates the collective wave pattern in the leg motions. A leg will enter the ADJUST state immediately after the previous leg enters the STANCE state. During the ADJUST state, the leg will move to a target pose (θ_T, α_T, β_T). The differences $\theta_T - \theta_M$, $\alpha_T - \alpha_S$, and $\beta_T - \beta_S$ determine the desired rotational phase difference between the legs. Changing their values will change the frequency of the leg wave patterns.

Note that the word "ideally" above indicates that during locomotion the legs do not always start or end in precise poses, but attempt to reach target start and end poses; e.g., when the ground is irregular, the SF1 state can end prematurely and enter the STANCE state, as will be detailed below. Only during the STANCE state will output from the Targeter be used, whereas in other states, the leg rotates by forward kinematics toward the target rotations.

Switcher: The following rules are used to update leg states by the Switcher associated with each leg:

1. A normal loop is from SF1 to SF2 to SB1 to SB2 and back to SF1 (Fig. 10(A)), this normal loop can be interrupted by STANCE and ADJUST states;
2. Whenever the leg tip touches the ground surface, enter the STANCE state;
3. When the IK solver cannot solve for the STANCE state, enter the SF1 state;
4. Whenever the previous leg enters the STANCE state, enter the ADJUST state;
5. At the end of the ADJUST state, enter the SF2 state;

For example, when the simulated myriapod is suspended off the ground (Fig. 10(A)), the legs will periodically go from the SF1 to the SB2 states without entering the STANCE and ADJUST states. If it is dropped to the ground, the legs will start to enter the STANCE and ADJUST states (Fig. 10(B),(C)). When the creature starts walking, its leg motions quickly converge to a wave pattern in which each leg undergoes the state cycle SF1→SF2→STANCE with short appearances of the ADJUST and SB1 states (Fig. 10(D)).

In particular, Rule 4 enforces a stable phase difference locally between legs, finally resulting in a wave-like leg formation during locomotion. The wave-like locomotion pattern has several merits: First it ensures that at any moment the body will be supported by a fixed ratio of legs such that stability is guaranteed. Second it allows each leg to stretch to extreme poses (AEP and PEP) with maximal energy efficiency, since the work done to raise and lower each leg can be reduced for a given locomotion distance.

Synchronization of the Left and Right Legs: Synchronizing the leg waves on both sides maximizes the duration that a rigid segment stays in the efficient IK-driven supported state. The synchronization of left and right leg rotations is achieved by adding an extra rule to the first segment's Switcher—when the left leg enters the SF1 state, the right leg will also immediately enter SF1.

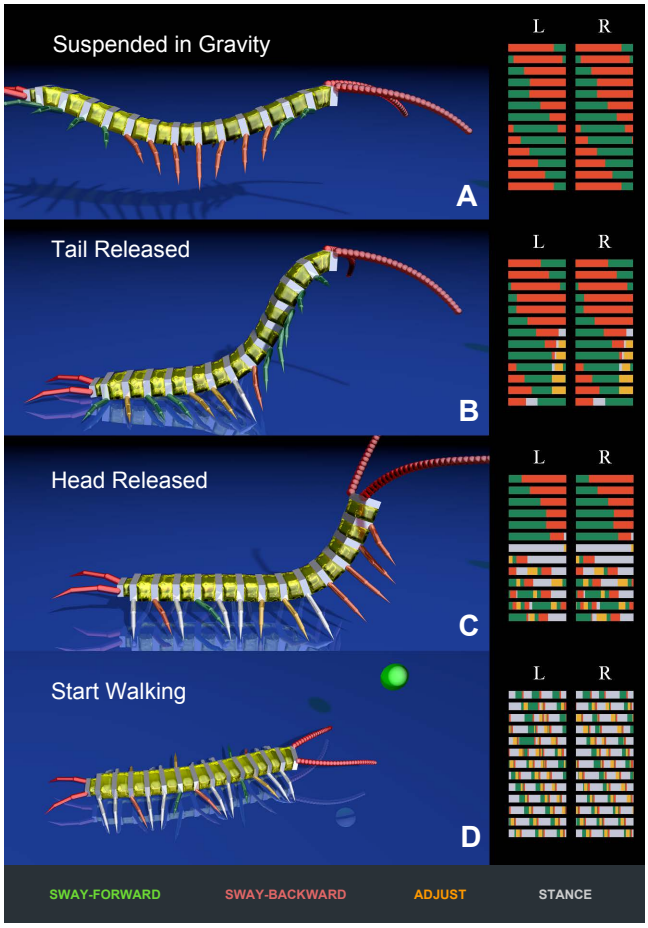

Figure 10: *A centipede dropped to the ground. The right panel shows the states of each leg during the last second.*

4.3 The Head

The head segment is almost always kinematically updated by its own Targeter, which will employ the Balancer for balancing, but not have any anterior segments to follow. The Targeter updates the configuration of the head segment based on three variables: (i) a moving speed (the head will always move forward along its x axis), (ii) a turning speed around the y axis, and (iii) a binary turning direction. There are various brain modes that hold different values of the three variables. A special Switcher updates those brain modes as well as the variables based on internal states and signals from the antennae. The brain modes and associated switching mechanisms are as follows:

- ADJUSTMENT: The head polls each segment to check if the majority of the legs are on the ground. If not, it will stay in the ADJUSTMENT mode and the head segment is controlled by the dynamics. This brain mode enables the animation of a creature dropped onto the ground to adjust its legs automatically before it starts walking (Fig. 10).
- RANDOM-WALK: When not in ADJUSTMENT mode, the brain automatically enters random-walk mode. Turning direction and target turning angles are set randomly and renewed when they are attained.
- AVOIDANCE: When the left/right antenna sends an obstacle contact signal, the brain enters the obstacle AVOIDANCE mode. The turning direction and speed will be adjusted to avoid the obstacle. After the obstacle is cleared, the head returns to RANDOM-WALK mode.
- TROPHOTROPISM: similar to the obstacle AVOIDANCE mode, antenna sensed signals of food in the environment modify the turning speed and direction such that the creature orients itself toward the food. After the food is reached, the head returns to RANDOM-WALK mode.
- CONTROL: The head is externally controlled by locomote and turn commands issued by a user.
- For obvious reasons, the AVOIDANCE mode has a higher priority than the TROPHOTROPISM mode.

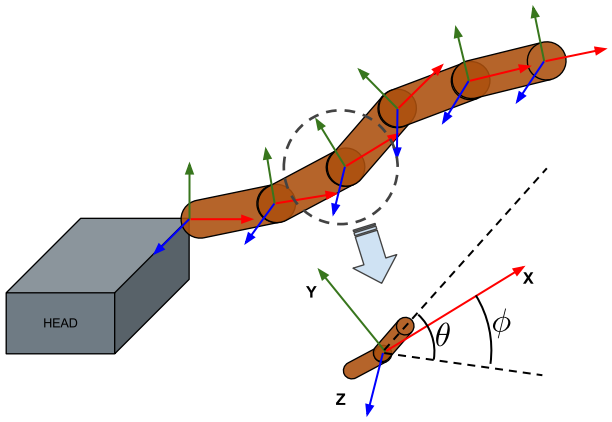

Figure 11: *Antenna modeling and animation.*

5 Environmental Interaction

5.1 Locomotion on Irregular Terrain

Our decentralized locomotion system works robustly in the presence of various types of terrains. Our simulated myriapoda can locomote over arbitrary triangular meshes (Fig. 1). The terrain model supports three basic queries: (1) determining if the tip of a leg is touching the ground by doing an inside/outside test, (2) obtaining the height h_i of the center $\mathbf{c}_i$ of the rigid segment relative to the terrain surface, and (3) obtaining the normal vector of the terrain surface directly below $\mathbf{c}_i$. The queries are easy to perform if the terrain is stored as a 2D height field, but for closed meshes such as the cranium in Fig. 1, we built a spatial hash table to improve the performance.

5.2 Sensors and Environmental Stimuli

Antennae Modeling and Animation: The antennae are modeled as a chain of N short links, each of which has two rotational degree-of-freedom (θ, ϕ) joints relative to the previous link (Fig. 11). We observed from video footage of a real centipedes that during locomotion the antennae will articulate randomly for exploration. Meanwhile, there is an outward traveling wave from the root of the antenna to the tip. To synthesize lifelike antenna movements, we devised the following two-step method: First, to achieve randomness in antenna rotation, we set random target rotations θ_T, ϕ_T for the root link ($i = 0$). The root link will rotate toward this target rotation at a constant speed (ω_0) and will generate new target rotations when the current ones are achieved. The formula is simply $[\theta_0, \phi_0]^{t+\Delta t} = [\theta_0, \phi_0]^t + \omega_0 \Delta t$. Second, for the links

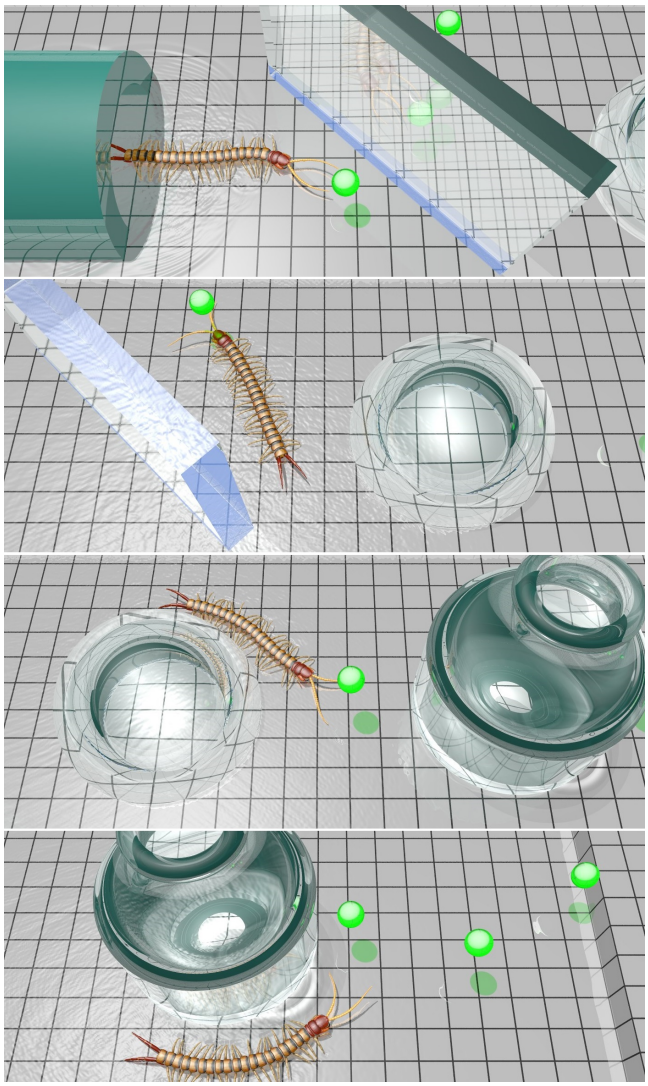

Figure 12: *A centipede walking in shallow water, attracted by food (green spheres) and avoiding obstacles.*

$(i = 1, 2, \ldots, N-1)$, their relative rotation angles (θ_i, ϕ_i) are sampled from a wave function $[\theta_i, \phi_i] = A_{[\theta,\phi]} \sin(\omega t - \lambda i) + B_{[\theta,\phi]}$, where A_θ, A_ϕ, B_θ, and B_ϕ are constant amplitudes and base values, and ω and λ determine the wave frequency and wavelength.

Food, Obstacles, Etc: The antennae detect obstacles through physical contact, when an antenna-obstacle collision occurs, the θ_T, ϕ_T, and ω_0 will be adjusted to retract the antenna in the direction of the obstacle's surface normal. A physical contact signal will be passed into the brain for locomotion adjustment. Our simple steering mechanism is similar to that in Braitenberg vehicles [Braitenberg 1986]—whenever there is a contact signal from the left antenna, the head will enter AVOIDANCE mode and turn right, and vice versa. One antenna has higher priority when both antennas sense a collision, in order to avoid the myriapod becoming trapped by walking straight into a wall. Food is modeled as point sources with intensity gradients that decrease with squared radial distance. The antenna can sense food intensity in the environment. Once the stimulus exceeds a certain threshold, it will send an intensity signal to the brain. The myriapod will turn in the direction of maximal food intensity. Our simulation shows that this simple foraging method works very well. Furthermore food sources can be used to plot the path of an artificial myriapod.

6 Implementation and Results

We implemented the biomechanical body structure and leg controller and parallelized the code with OpenMP. Our Intel Core i7-3930k (6 core, 3.2GHz) machine can update an 18 segment centipede model with $3 \times 3 \times 3$ deformable mesh resolution at a rate of 1.7×10^4 time-steps per second. Typically, a time-step of $1/3000s$ is used, resulting in approximately $6\times$ real-time speed. Most of the time is consumed by the semi-implicit FEM simulation of the deformable segments (fully implicit simulation should improve stability and speed). The update procedure is as follows:

1. **Update of the Antenna:**
   - The rotation angles for the root link are updated with a constant rotation velocity toward the target rotations;
   - The rotation angles for the remaining links are updated by sampling from a wave function.
   - If a collision with an obstacle occurs, a retraction phase will be triggered.
2. **Update of the Head:**
   - The sensory information and internal states are processed and brain modes are switched if necessary.
   - If the brain is in ADJUSTING mode, the head segment's position and orientation is updated dynamically; otherwise, it is updated kinematically as follows:
   - First, the Balancer adjusts the orientation and elevation of the segment;
   - Second, the head segment's position and orientation are updated according to the brain's mode. The head always moves in the direction it is pointing and its orientation is updated to execute a turn.
3. **Update of the Rigid Segments:**
   - If the segment is not supported by the legs, it will be updated dynamically, otherwise it will be updated kinematically, as follows:
   - First, the Balancer will adjust the orientation and elevation of the segment;
   - Second, the Follower will adjust the position and orientation of the segment. The leg root positions are updated accordingly with the rigid segment.
4. **Update of the Legs:**
   - The Switcher checks the leg states and updates them accordingly;
   - If in the STANCE state, the IK solver (Appendix B) computes the leg rotation angles from the tip and root position of the leg;
   - Otherwise, forward kinematics will update the leg rotations and tip positions towards the target poses of that state.
5. **Update of the Deformable Segments:**
   - The deformable segments are dynamically updated with the elasticity simulation (Appendix A) and proposed coupling method.

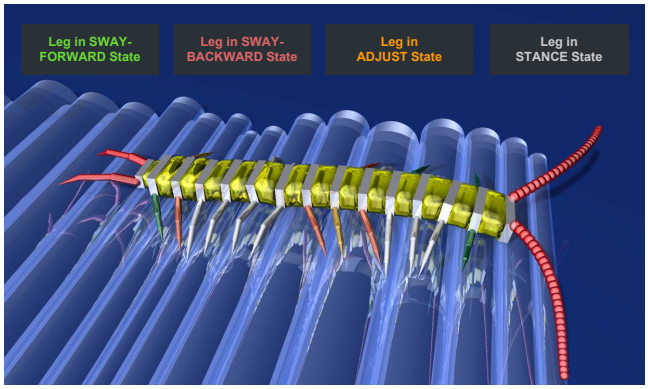

Figure 13: *Walking over an irregular surface paved with random cylinders.*

6.1 Rendering

The physical model may be rendered simply as rigid primitives (cube, sphere, cylinder) and as jelly for the deformable segments (Fig. 2(C) and Fig. 5). For the textured myriapod mesh models shown in Fig. 1 and Fig. 2(A), we rigged a skeleton that represents the biomechanical model and attached it to the mesh. The physical parameters, such as segment size and leg length, are calibrated and set in the simulation. Per-frame data, including rigid segment position, orientation, leg rotation angles, etc., are exported to Maya to set keyframe of the rigged model.

6.2 Simulations

Our myriapod simulator includes a ground plane with additional surface obstacles such as cylinders and cubes for the myriapoda to crawl over (e.g., Fig. 13). We have also implemented a height field based terrain generator that can create arbitrary terrains from mixtures of Gaussian functions. In addition to terrain, we extended our simulator to support closed triangulated meshes. Fig. 1 shows a centipede walking over the mesh of a human skull. In our locomotion simulations, contact friction is set high enough for the tips of the legs not to slide appreciably on the ground surfaces.

In our real-time simulator, food and obstacles can be generated randomly in the scene to elicit the emergent behavior of an autonomous myriapod creature exploring its environment. For animation purposes, the user can place food particles and vertical obstacles on the terrain surfaces to guide the locomotion path of the creature.

Different body parameters can be used to model different types of myriapoda, as shown in Fig. 2. To add realism to our animation, we coupled the legs of a centipede to a shallow water simulator [Kass and Miller 1990]. The water ripples realistically under the legs of the myriapod, leaving a wake along the path. Fig. 12 shows stills from an animation in which a centipede walks around obstacles toward food sources in shallow water. One-way coupling of the myriapods with sand or snow would also generate realistic traces on such granular surfaces.

7 Conclusion

We have developed a framework for animating myriapoda with lifelike ambulation and simple autonomous behaviors. To deal with the unique body structure of myriapoda relative to other Arthropods, we devised a novel physical body structure that is composed of mixed rigid and deformable body segments. In conjunction with the physical body structure, we devised a decentralized locomotion control system composed of identical local leg controllers. The locomotion controller demonstrates competence in various irregular terrains and over surface obstacles. In addition to their motor control system, our myriapoda are equipped with antennae for sensing food and obstacles, leading to interesting autonomous exploration behaviors in a complex environment. Animators can take advantage of these high level behaviors to guide the locomotion path of our simulated creatures.

Given its simplicity and physical realism, our use of efficient and highly robust elastic components has the potential to supplant traditional mass-spring-damper systems for simulating the deformable body structures of artificial animals [Miller 1988; Tu and Terzopoulos 1994]. There are various possible ways to actuate such structures. In our myriapod models, the rigid segments serve as active parts that motivate the passive deformable body parts. This is suitable for manually designed motor controllers such as ours, since the motor control of rigid body parts is simpler than controlling deformable meshes. The alternative is to actuate the elastic deformable body structures to achieve locomotion goals through the use of optimization algorithms such as the genetic algorithm [Sims 1994] or simulated annealing [Grzeszczuk and Terzopoulos 1995].

Interesting future work would include biomechanically simulating the swimming abilities of myriapoda, by including actively contractile elastically deformable body segments that can serve as muscle actuators, and automatically learning locomotion controllers for simulated myriapod creatures in both terrestrial and aquatic environments.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. We are grateful to Xinli Cai for her centipede sketch in Fig. 3 and to Wenjia Huang for her assistance with rigging in Maya the beautiful geometric model of the centipede, which was purchased from the Turbosquid website. We made use of the outstanding open source libraries Eigen, OpenMP, and OpenGL for linear algebra, parallelization, and rendering, respectively.

A Finite Element Modeling and Simulation

This appendix details our simulation of the deformable segments in our myriapod model using the finite element method (FEM).

Let $\mathbf{X}$ be the material coordinates of a deformable body, and $\mathbf{x}(t) = \phi(\mathbf{X}, t) = \mathbf{F}(t)\mathbf{X} + \mathbf{b}(t)$ be the world coordinates. We discretize the material space with a uniform tetrahedral mesh. In each tetrahedral element, the deformation gradient $\mathbf{F}$ can be computed as

$$\mathbf{F}_e = \mathbf{D}_s \mathbf{D}_m^{-1}, \tag{2}$$

where

$$\mathbf{D}_m = (\mathbf{X}_4 - \mathbf{X}_1, \mathbf{X}_3 - \mathbf{X}_1, \mathbf{X}_2 - \mathbf{X}_1) \tag{3}$$

are the edge vectors of the undeformed tetrahedron and

$$\mathbf{D}_s = (\mathbf{x}_4 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_1, \mathbf{x}_2 - \mathbf{x}_1) \tag{4}$$

are the edge vectors of the deformed tetrahedron.

For hyperelastic material, we use the fixed corotational energy density function

$$\Psi = \mu \|\mathbf{F} - \mathbf{R}\|_F^2 + \frac{\lambda}{2}(J-1)^2 \tag{5}$$

$$= \mu \sum_i (\sigma_i - 1)^2 + \frac{\lambda}{2}(J-1)^2, \tag{6}$$

where λ and μ are Lamé parameters,[3] J is the determinant of $\mathbf{F}$, $\mathbf{R}$ comes from the polar decomposition $\mathbf{F} = \mathbf{RS}$, and σ_i are the singular values of $\mathbf{F}$, which we compute using the fast SVD method proposed by [McAdams et al. 2011].

The elastic forces on nodes are computed as

$$\mathbf{f}_i = \sum_e V_e^0 \frac{\partial \Psi(\mathbf{F}_e)}{\partial \mathbf{x}_i} = \sum_e V_e^0 \mathbf{P} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_i}, \tag{7}$$

where V_e^0 is the undeformed volume of tetrahedral element e, and $\mathbf{P}$ is the first Piola-Kirchhoff stress, given by

$$\mathbf{P} = \frac{\partial \Psi(\mathbf{F})}{\partial \mathbf{F}} = 2\mu(\mathbf{F} - \mathbf{R}) + \lambda J(J-1)\mathbf{F}^{-T}. \tag{8}$$

According to (7), the explicit force formula for each element can be written as

$$[\mathbf{f}_1^e, \mathbf{f}_2^e, \mathbf{f}_3^e] = V_e^0 \frac{\partial \Psi(\mathbf{F}_e)}{\partial [\mathbf{x}_1^e, \mathbf{x}_2^e, \mathbf{x}_3^e]} = V_e^0 \mathbf{P} \mathbf{D}_m^{-T}, \tag{9}$$

$$\mathbf{f}_0^e = \frac{\partial \Psi(\mathbf{F}_e)}{\partial \mathbf{x}_0^e} = -\mathbf{f}_1^e - \mathbf{f}_2^e - \mathbf{f}_3^e. \tag{10}$$

The calculation of elastic forces on each mesh node must take into account the contributions from each tetrahedral element; i.e.,

$$\mathbf{f}_i^E = \sum_{e, i \in e} \mathbf{f}_{i_e}^e, \tag{11}$$

where i_e is the local index of node i in element e. The mass of each node is calculated by averaging the mass over neighboring elements:

$$m_i = \frac{1}{4} \sum_{e, i \in e}^{e} \rho V_e^0, \tag{12}$$

where ρ is the density of the soft material.

We also introduce the damping force as

$$\mathbf{f}_i^D = -\gamma \sum_{e, i \in e}^{e} V_e^0 (\mathbf{v}_i - \frac{1}{4} \sum_{k=1}^{4} \mathbf{v}_{e_k}), \tag{13}$$

where e_k is the index of node k of element e, and γ is the damping coefficient.

A semi-implicit time integration scheme is applied at each time step:

$$(\mathbf{I} + \Delta t \mathbf{M}^{-1} \mathbf{D}) \mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t \mathbf{M}^{-1} (\mathbf{f}^E + \mathbf{g} + \mathbf{f}), \tag{14}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1}, \tag{15}$$

where $\mathbf{v}$ is the nodal velocity vector, $\mathbf{x}$ is the nodal position vector, $\mathbf{M}$ is the diagonal mass matrix assembled from (12), $\mathbf{D}$ is the damping matrix assembled from (13), $\mathbf{f}^E$ is the internal, elastic force vector, $\mathbf{g}$ denotes gravity, and $\mathbf{f}$ are externally applied forces. In our simulations, we use a time step of $\Delta t = 1/3000$ sec.

For the deformable segments in our myriapod body model, we set the density of the hyperelastic material to $\rho = 1.0$, its Young's modulus $E = 4000$, its Poisson's ratio $\nu = 0.4$, and the damping coefficient $\gamma = 50$.

[3] In terms of Young's modulus E and Poisson's ratio ν, Lamé's first parameter $\lambda = E\nu/(1+\nu)(1-2\nu)$ and second parameter $\mu = E/2(1+\nu)$.

B Inverse Kinematics of the Legs

Referring to Fig. 9, the leg inverse kinematics algorithm takes as input the position of the leg tip P and leg root O and computes the joint angles θ, α, and β. If no convex solution (i.e., with $\beta > 0$) exists, it will inform the Switcher to switch to the SF1 state. Fixing γ for simplicity, the steps to solve for θ, α, and β are as follows:

1. By restricting all the leg segments to lie in a plane, θ can be computed independently from α and β using the positions of P and O, as Fig. 9(2) shows.

2. Angles α and β can be computed by solving for the position of point M (Fig. 9(1)), which is determined by rotating segment OM around point O and segments PM (with γ fixed) around point P. The two circles will either not intersect (no solution), intersect at one point (tangent), or intersect at two points. If a convex solution with $\beta > 0$ exists for point M in the leg plane, then α and β are computed using simple geometric calculations.

References

BARAFF, D., AND WITKIN, A. P. 1997. Partitioned dynamics. Tech. Rep. CMU-RI-TR-97-33, The Robotics Institute, Canegie-Mellon University, Pittsburgh, PA.

BARAFF, D. 1997. An introduction to physically based modeling: Rigid body simulation I—Unconstrained rigid body dynamics. In *An Introduction to Physically Based Modeling, SIGGRAPH '97 Course Notes*, 97.

BEER, R. D., CHIEL, H., QUINN, R., AND ESPENSCHIED, K. 1992. A distributed neural network architecture for hexapod robot locomotion. *Neural Computation 4*, 6, 356–365.

BEER, R. D. 1990. *Intelligence as adaptive behavior: An experiment in computational neuroethology*. Academic Press, San Diego, CA.

BRAITENBERG, V. 1986. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA.

CENYDD, L. A., AND TEAHAN, B. 2013. An embodied approach to arthropod animation. *Computer Animation and Virtual Worlds 24*, 65–83.

CHEN, D. T., AND ZELTZER, D. 1992. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. *Computer Graphics (Proc. ACM SIGGRAPH 92) 26*, 89–98.

COROS, S., KARPATHY, A., JONES, B., REVERET, L., AND VAN DE PANNE, M. 2011. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics 30*, 59:1–12.

COROS, S., MARTIN, S., THOMASZEWSKI, B., SCHUMACHER, C., SUMNER, R., AND GROSS, M. 2012. Deformable objects alive! *ACM Transactions on Graphics 31*, 69:1–9.

CRUSE, H., KINDERMANN, T., SCHUMM, M., DEAN, J., AND SCHMITZ, J. 1998. Walknet: A biologically inspired network to control six-legged walking. *Neural Networks 11*, 1435–1447.

CRUSE, H., DEAN, J., DÜRR, V., KINDERMANN, T., SCHMITZ, J., AND SCHUMM, M. 2000. Control of hexapod walking: A decentralized solution based on biological data. In *Proc. 4th Int. Conf. on Climbing and Walking Robots*, 13, 1–10.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proc. ACM SIGGRAPH 01*, 251–260.

FULL, R. J., AND TU, M. S. 1991. Mechanics of a rapid running insect: Two-, four- and six-legged locomotion. *Journal of Experimental Biology 156*, 215–231.

GIBSON, D. P., OZIEM, D. J., DALTON, C. J., AND CAMPBELL, N. W. 2007. A system for the capture and synthesis of insect motion. *Graphical Models 69*, 231–245.

GOLUBITSKY, M., STEWART, I., BUONO, P.-L., AND COLLINS, J. 1998. A modular network for legged locomotion. *Physica D: Nonlinear Phenomena 115*, 56–72.

GRZESZCZUK, R., AND TERZOPOULOS, D. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *Proc. ACM SIGGRAPH 95*, 63–70.

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proc. ACM SIGGRAPH 95*, 71–78.

HOFFMAN, K., AND WOOD, R. 2011. Myriapod-like ambulation of a segmented microrobot. *Autonomous Robots 31*, 103–114.

HOLMES, P., FULL, R. J., KODITSCHEK, D., AND GUCKENHEIMER, J. 2006. The dynamics of legged locomotion: Models, analyses, and challenges. *SIAM Review 48*, 207–304.

IJSPEERT, A. J., CRESPI, A., RYCZKO, D., AND CABELGUEN, J.-M. 2007. From swimming to walking with a salamander robot driven by a spinal cord model. *Science 315*, 1416–1420.

INAGAKI, S., YUASA, H., AND ARAI, T. 2003. CPG model for autonomous decentralized multi-legged robot system generation and transition of oscillation patterns and dynamics of oscillators. *Robotics and Autonomous Systems 44*, 171–179.

INAGAKI, S., NIWA, T., AND SUZUKI, T. 2011. Decentralized control of centipede-like multi-legged robots with passive intersegment joints based on follow-the-contact-point gait control. *Transactions of the Society of Instrument and Control Engineers 47*, 282–290.

KASS, M., AND MILLER, G. 1990. Rapid, stable fluid dynamics for computer graphics. *Computer Graphics (Proc. ACM SIGGRAPH 90) 24*, 49–57.

KIMURA, H., FUKUOKA, Y., AND COHEN, A. H. 2007. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *International Journal of Robotics Research 26*, 475–490.

MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Transactions on Graphics 30*, 4, 37:1–12.

MCKENNA, M., AND ZELTZER, D. 1990. Dynamic simulation of autonomous legged locomotion. *Computer Graphics (Proc. ACM SIGGRAPH 90) 24*, 29–38.

MELLEN, N., KIEMEL, T., AND COHEN, A. H. 1995. Correlational analysis of fictive swimming in the lamprey reveals strong functional intersegmental coupling. *Journal of Neurophysiology 73*, 100–1030.

MILLER, G. S. P. 1988. The motion dynamics of snakes and worms. *Computer Graphics (Proc. ACM SIGGRAPH 88) 22*, 169–173.

ODASHIMA, T., YUASA, H., AND ITO., M. 1998. The autonomous decentralized myriapod locomotion robot which is consist of homogeneous subsystems. *Journal of the Robotic Society of Japan 16*, 81–88.

RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. *Computer Graphics (Proc. ACM SIGGRAPH 91) 25*, 349–358.

RAIBERT, M. 2008. Bigdog, the rough-terrain quadruped robot. In *Proc. 17th International Federation of Automatic Control World Congress*, 10822–10825.

RUPPERT, E. E., FOX, R. S., AND BARNES, R. D. 2003. *Invertebrate Zoology, 7th Ed.* Cengage Learning.

SARANLI, U., BUEHLER, M., AND KODITSCHEK, D. E. 2001. Rhex: A simple and highly mobile hexapod robot. *International Journal of Robotics Research*, 616–631.

SHINAR, T., SCHROEDER, C., AND FEDKIW, R. 2008. Two-way coupling of rigid and deformable bodies. In *Symposium on Computer Animation*, 95–103.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *Symposium on Computer Animation*, 81–90.

SIMS, K. 1994. Evolving 3D morphology and behavior by competition. *Artificial Life 1*, 4, 353–372.

SKRBA, L., REVERET, L., HÉTROY, F., CANI, M.-P., AND O'SULLIVAN, C. 2008. Quadruped animation. In *Eurographics State-of-the-Art Report*.

STOMAKHIN, A., HOWES, R., SCHROEDER, C., AND TERAN, J. M. 2012. Energetically consistent invertible elasticity. In *Symposium on Computer Animation*, 25–32.

TERAN, J., BLEMKER, S., HING, V. N. T., AND FEDKIW, R. 2003. Finite volume methods for the simulation of skeletal muscle. In *Symposium on Computer Animation*, 68–74.

TERZOPOULOS, D. 1999. Artificial life for computer graphics. *Communications of the ACM 42*, 8, 32–42.

TSAI, Y.-Y., LIN, W.-C., CHENG, K. B., LEE, J., AND LEE, T.-Y. 2010. Real-time physics-based 3D biped character animation using an inverted pendulum model. *IEEE Transactions on Visualization and Computer Graphics 16*, 325–337.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proc. ACM SIGGRAPH 94*, 43–50.

VAN WELBERGEN, H., VAN BASTEN, B. J. H., EGGES, A., RUTTKAY, Z. M., AND OVERMARS, M. H. 2010. Real time animation of virtual humans: A trade-off between naturalness and control. *Computer Graphics Forum 29*, 2530–2554.

WANG, J. M., HAMNER, S. R., DELP, S. L., AND KOLTUN, V. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics 31*, 25:1–11.