COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2013; **24**:65–83
Published online 30 May 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.1436

RESEARCH ARTICLE

An embodied approach to arthropod animation

Llyr ap Cenydd* and Bill Teahan

School of Computer Science, Bangor University, Bangor, UK

ABSTRACT

We describe a system for dynamically animating the locomotive behaviour of arthropods (insects, spiders and numerous other species) in real-time, facilitating realistic and autonomous traversal across an arbitrary environment. By combining a decentralised reactive behavioural model with a hybrid approach to motion that utilises the comparative advantages of physical simulation and kinematic control, our system is capable of automatically generating complex organic motion over a wide range of surface features, independent of structural complexity. The reactive embodiment of the creature, combined with the physical simulation of the virtual world enables the formation of emergent behaviours that are entirely based on circumstance, including rigid-body interaction, grip recovery and adaptive wall climbing. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

dynamic animation; procedural animation; real-time; arthropods; spiders; insects

***Correspondence**

Llyr ap Cenydd, School of Computer Science, Bangor University, Bangor, UK.
E-mail: llyr.ap.cenydd@bangor.ac.uk

1. INTRODUCTION

Real-time virtual environments are becoming increasingly complex, realistic and physically simulated. Similarly, advances in graphics allow for the rendering of increasingly life-like creatures. However, there largely remains a disconnect between the two, as virtual characters are still predominantly animated using pre-created data. Although perfectly adequate for specifically designed environments, the interaction between agent and environment is still primarily an illusion. Removing, or at least limiting this discrepancy potentially allows for much more interesting, complex and realistic animations to form automatically.

Arthropods are the largest phylum of animals on Earth, and include all arachnid, insect, crustacean and numerous other species. Most arthropods are capable of unrestricted movement within their environment. They can navigate cluttered surfaces, scale walls and walk across ceilings with relative ease, displaying a vast range of dexterous motions in going from one surface to another. The subtleties and variance of these motions are very difficult to capture using current data-driven techniques, especially for the general case.

The real-time animation of arthropods therefore presents a considerable challenge. Although the locomotion of various insects and arachnids have been extensively studied in the fields of robotics and biology, to the best of our knowledge, there has been no prior attempt to dynamically animate the locomotive behaviour of any arthropod in real-time within a completely arbitrary environment, let alone their ability to interact with objects in the scene and navigate large changes in surface orientation. Natural looking animations of such feats could greatly enhance vividness and immersion in interactive applications, as well as potentially being of benefit to fields such as phobia therapy and artificial life research.

In this paper, we present a system for dynamically animating the majority of ground based arthropods in real-time, capable of displaying the range of motion and dexterity found in real-life equivalents. By introducing an abstraction between the stepping limbs and the aggregate motion of the creature, we can evoke a level of animation realism comparable to that produced by ethological simulations in far more complex environments, while retaining global control. By utilising a novel stance reflex mechanism, locomotion is produced by the complex interaction between simple internal systems—facilitating the generation of dynamic and contextual motion. Furthermore, such self-organising functionality allows for the emergence of behaviour not explicitly programmed, the most prolific example being the tentative stepping exhibited when navigating large changes in surface orientation (such as climbing walls), a behaviour which mimics the actions of real-life equivalents.

Our test models have mostly been various species of spider. However, the majority of arthropod species are

Copyright © 2012 John Wiley & Sons, Ltd.

similar enough in body-limb structure that our technique can effectively simulate any creature of similar morphology, including other arachnids, insects, crustaceans and even reptiles, or any hypothetical character with a low centre of gravity and stable posture.

2. RELATED WORK

The creation of autonomous characters capable of realistically navigating around a virtual environment continues to be a challenge in computer animation. The steady increase in computational power has facilitated the development of many techniques of various abstraction, resulting in an expansive body of literature dedicated towards generating animation for bipedal figures [1], with many techniques also applicable to other morphologies such as quadrupeds [2]. One of the main overarching issues in real-time animation research concerns the trade-off between how natural a motion appears, and how much control can be exerted.

There has also been exceptional work in deriving legged locomotion control from entomological studies, resulting in a large number of robotic and simulatory systems. We cannot directly compare our system with the approaches taken in ethological, biomechanical and robotics research because of our differing goals; they are largely limited by their requirement to accurately physically simulate ambulation, or to produce control systems for machines destined for the real world, although we have no such restriction. However, we do share the need to mimic anthropod ambulation as closely as possible, and the underlying control theory used in our system is based upon research in these areas.

2.1. Animation Techniques

A common technique in both research and commercial software is to synthesise animation using conventional [3,4] and constraint based data-driven approaches [5], which usually require a large library of motion captured sequences to produce realistic results. Here, the response of an animated character in different situations can be extracted from a motion database and blended to create plausible and natural motion. Research has been conducted into capturing motion sequences for various arthropods, such as in [6] where a system was developed to capture and synthesise the 3D motions of insects and arachnids in order to drive a computer-generated model. Although it would certainly be possible to use data-driven animation techniques with captured (or hand crafted) arthropod motion sequences, we do not believe that such an approach can adequately capture the dexterity and range of motion displayed by real-life arthropods outside specific situations or carefully designed environments. An alternative approach is to synthesise the animation in real-time, which potentially allows for much more vivid, flexible and contextual motion to emerge automatically. Such techniques can be broadly divided into two categories—procedural animation and physical simulation.

Procedural animation techniques generally use parameterised formulas to create motion primitives, which can describe joint rotations directly or describe the paths of end-effectors, usually in conjunction with an inverse kinematics (IK) solver [7,8]. Alternatively, in physical-based approaches, animation is generated by the direct application of force and torque on an articulated figure. A wide variety of control strategies have been proposed, including the use of neural networks such as in Natural Motion's Euphoria engine [9], learning and optimisation strategies [10,11] and inverted pendulum models [12].

Many of the underlying techniques used in synthesising animation for bipedal and quadruped locomotion are also applicable for arthropod morphologies. For example, we abstract the motion of the creature's body from the underlying locomotion mechanism, the articulation of each limb being defined by an end-effector path and IK solver. However, in our approach, we also physically simulate the figure, with its motion defined by an underlying kinematics template. This aims to combine the comparative advantages of both procedural and physical simulation, although also effectively allowing for switching between kinematic, hybrid and wholly physically based simulation, similar to the approach Shapiro *et al*. proposed for switching between data-driven and physical simulation in [13].

2.2. Arthropod Simulation

There has been substantial research into the physiology of arthropods, including analysis of locomotion control in arachnids [14,15], assorted insects [16,17] and crustaceans [18]. An excellent survey on the modelling, analysis and challenges associated with insect locomotion dynamics can also be found in [19].

Early research into the simulation of insect ambulation can be seen in [20], where a forward dynamics algorithm was proposed for the locomotion coordination of a simulated cockroach, capable of negotiating planar and uneven terrain. The cockroach model consisted of a simple dynamical model with springy legs, with the load of the body distributed across supporting legs.

An alternative approach was taken by Randall Beer in [21], being one of the most well known early approaches to adaptive behaviour. The model consisted of a simplified cockroach consisting of joint-less 2D line segments, controlled by an artificial neural network. By specifying a requirement to maximise forward velocity, the artificial creature was made to learn six-legged locomotion. The approach was extended in [22] to incorporate sensory feedback and strategy switching mechanisms, and further applied to hexapod robot locomotion in [23].

The primary goal of the Walknet system outlined by Cruse *et al*. in [24,25] was to develop a system, which could be used as a scientific tool for studying insect walking. Walknet consisted of a simple (static) neural network,

Comp. Anim. Virtual Worlds 2013; **24**:65–83 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

which reproduced the behavioural properties of hexapod walking. Their insights, although investigating stick insect ambulation, indicated that they gain their adaptivity and flexibility mainly from the extremely decentralised organisation of the control system that generates leg movements, with neither the movement of a single leg or gait coordination appearing centrally pre-programmed. They deduced from this that the production of gait is an emergent property of the whole system, in which each of the six leg controllers obey a few simple, localised rules regarding the states of neighbouring legs [26]; this modus operandi also forms the foundation of our system's design.

An alternative technique for autonomous arachnid locomotion was explored in [27]. They proposed that a physically simulated eight-legged virtual creature can learn to control its own muscles and achieve a viable walking gait by using a stochastic minimisation algorithm to maximise forward momentum, whilst simultaneously minimising total muscle-energy expenditure.

Our initial research [28] used a procedural animation system inspired by insect gait and stepping functions, combined with a point mass model of a spider and an IK-controlled skeletal system. Here, the orientation and elevation of the body was taken directly from ray-polygon intersection tests, and although producing viable animation of a spider walking across an uneven surface, natural movement over angular or more complex surfaces were limited to controlled situations. This approach was used to create a pre-rendered animation sequence in [29], and is similar to other commonly used arthropod animation rigging techniques [30].

In [31], the paths of the creature's feet were directed by rigid body spheres, allowing for a more substantial (albeit indirect) method of sensing the environment. Although this approach resulted in the generation of plausible animation over increasingly arbitrary surfaces, there remained a disconnect between the creature's motion and the underlying surface, being especially apparent in cluttered or significantly angular environments.

2.3. Lessons from Robotics

There is significant crossover between the biomechanical study of insects and its application in robotics, especially concerning the generation of plausible locomotion and associated gait control. There has been a vast amount of research into insect-inspired six and eight-legged robots of varying degrees of capability, complexity and expense, with the underlying designs being predominantly based on cockroaches and stick insects attributable to extensive ethologic and neurophysiologic knowledge.

Although the initial work into insect-inspired controllers [32,33] lead to a number of robots being developed, they were limited to walking over a flat surface. Arthropods, however, are capable of traversing natural terrains, which are often not level, are slippery, provide poor support and contain significant vertical variations, obstacles and sparse footholds. More advanced machines were developed to emulate these abilities.

The research outlined in [34,35] describes a robot capable of adjusting its gait to walk across uneven terrain. In order to cope with these irregularities, a more sophisticated mechanism to that used in [32] was developed. The machine was loosely based on a stick insect, with the body suspended by its legs to enhance stability. Walking across complex terrain required dynamic adjustments, as opposed to the statically stable gaits previously implemented. Inspired by the assorted strategies and reflexes used by insects, four main extensions were identified:

Passive and Active Compliance. Used to maintain a stable posture on complex terrain. Passive compliance denotes the springs in each leg joint, whereas active compliance was split into three components: proportional feedback position control at each (rotational) joint, reduction of joint stiffness in response to increased loads, and distribution of vertical load amongst supporting legs.

Stepping Reflex. A stepping reflex that allows the robot to compensate for leg perturbations. Any significantly perturbed leg will step back into a more stable posture.

Search Reflex. If no footholds are available at the end of a swing, or support is lost in a stance leg, a searching reflex is triggered, which results in increasingly concentric steps biassed in the direction of movement.

Elevatory Reflex. If a swinging leg encounters an obstacle, it is briefly retracted and lifted higher before continuing its swing. The process is repeatable for large elevations.

By combining a distributed architecture with the reflex mechanisms outlined earlier, the robot could display smooth locomotion over significantly irregular terrain, despite the absence of any sensing mechanism. As we will describe in the remainder of this paper, our animation system also contains analogues of these four reflex strategies.

3. AN ARTHROPOD MODEL

Our simulation uses Object-Orientated Graphics Rendering Engine (OGRE) [36], and the Newton Game Dynamics physics engine [37]. Our system is designed for simulating an arbitrary arthropod, and therefore, all of the system's parameters are stored in a separate species specific attribute file, facilitating the development of many variations of figure composition, locomotion strategy and overall behaviour. The contents of this file (and therefore the main parameters of our system) can be seen in the Appendix (Table I).

3.1. The Articulated Figure

Our creature models consist of a polygonal mesh rigged with an underlying skeleton (Figure 1). Given this skeleton at initialisation, a convex rigid body is created for every segment in the creature's body (head, thorax and

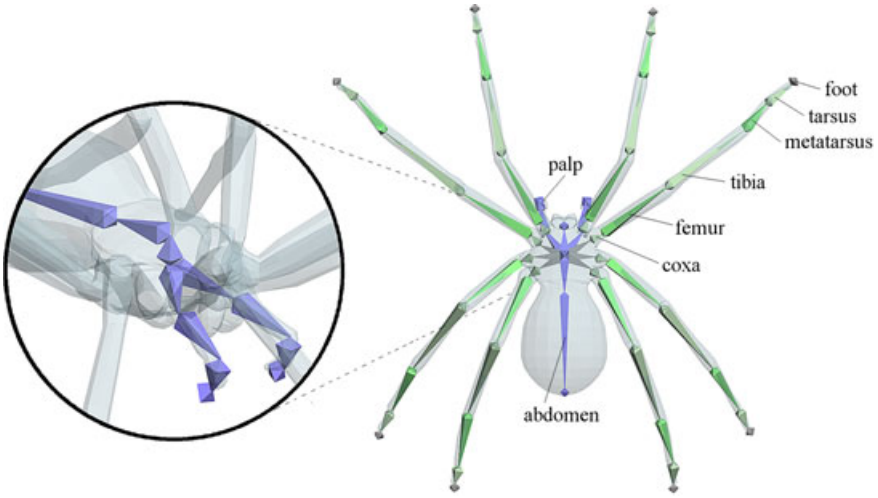

Figure 1. Huntsman spider skeleton and underlying mesh.

abdomen), approximating each segment's collision hull. A rigid body cuboid or cylinder is also created for every limb segment, the mass and moment of inertia automatically calculated.

Following the creation of the root thorax rigid body, subsequently created rigid body segments are attached to their parent in the hierarchy with an actuated joint. Although the hierarchical structure and base orientation of each bone is described within the skeleton itself, specific joint configurations are taken from a separate species specific attribute file, which describes for each (bi-symmetrical) joint pair the rotational axis, maximum and minimum angles of rotation, rotational comfort range, stiffness and elasticity. Given these specifications, the rigid body segment is restricted to powered joint rotation around a single defined (hinge) axis, whereas potentially allowing for limited flexibility in other directions. Deliberate articulation is achieved by powering the aforementioned joints with a corresponding rotational force, which affects any child bodies of the system. Any movement of the rigid bodies is mirrored in the creature skeleton, which in turn deforms the polygonal mesh.

There are parts of an arthropod's physiology that operate largely independent of movement, such as a spider's chelicera and palp appendages (Figure1), the jaw-like mandible horns of some beetle species, and insect antennae. Being largely independent of the underlying ambulatory mechanism, it is possible to animate these sections of the skeletal hierarchy separately using key-frame sequences in order to compliment the dynamical locomotion.

3.2. Limb Control

Each limb has an associated target node (Figure 2). These nodes denote the ideal or target position of the leg tip for a simulation step. Limb configuration is controlled via

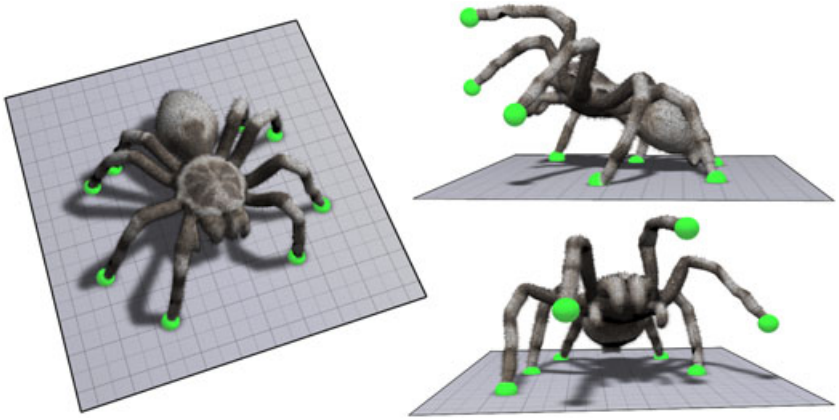

Figure 2. Posing a tarantula model using target nodes (green). The elevation and orientation of the thorax is derived from the average foot position (Section 5.4).

DOI: 10.1002/cav

a simple cyclic coordinate descent IK solver, which calculates based on the foot's current position the rotation required at each joint to drive the positional error between target node and foot (end-effector) toward zero. This incremental rotation is translated into a corresponding torque force for each actuated joint. Given adequate parameters, the comfort range, absolute limit, elasticity and stiffness of each joint will curb unnatural or erroneous limb poses.

3.3. The Virtual Environment

We aim to produce an animation system that enables the simulated creatures to explore and navigate around their virtual environment completely autonomously, with no pre-processing or explicit labelling of specific mesh features. Therefore, to effectively evaluate the capabilities of our simulated creatures, we require that the environment be arbitrary in detail and structure, from geometrically relatively simple environments such as the surface of a cuboid, sphere or torus knot to complex virtual worlds such as cluttered forest floors and man-made environments.

The majority of the interaction with the virtual environment is processed using a collision tree, being the preferred method within the physics engine for processing collisions between meshes of arbitrary complexity. Any object or surface too heavy for an arthropod to have any effect through interaction is considered static geometry, and processed using collision trees. Interactive objects are processed using convex collision primitives as dynamic rigid bodies.

4. THE BEHAVIOURAL SYSTEM

At the core of our technique is the idea that it is the creature's body that instigates movement, and although the legs play no immediately active role, they animate as though the inverse is true. Most arthropods are comparatively small and lightweight, with the act of generating forward momentum in the limbs in effect occurring concurrently with the resultant movement of the body. It follows that the order by which these events occur can be reversed with little to no loss in animation accuracy.

Therefore, in simulator reality, the creature's thorax is the instigator of forward propulsion. By actively maintaining a balanced stance relative to the body, the legs act to maintain the appearance of ambulation. Furthermore, the physical presence of each limb through the articulated skeleton effectively embodies the creature in its environment, enabling both passive and active feedback between the legs and body.

Our hypothesis is that organic, dynamic animation can be achieved in real-time by reducing the ambulatory mechanism inherent in all arthropod species into a series of modules, each regulating a specific function required in maintaining visually coherent locomotion. Through feedback between these components, a constant re-balancing reflex mechanism can be achieved that ensures that the limbs step in accordance with the body, effectively negating the fact that no actual physical propulsion is taking place.

4.1. Steering Behaviours

In our system, the velocity of the creature, and therefore both its speed and relative heading, is governed by a physically based vehicle model, similar to that proposed by Craig Reynolds in [38]. The paper describes a method of controlling global movement using steering behaviours, allowing a virtual character to achieve higher level goals (such as exploration and obstacle avoidance) independently of underlying locomotion mechanism.

Reynolds divided motion behaviour into three layers—action selection, steering and locomotion. Action selection is concerned with the abstract concepts of strategy, goals and planning, which are outside the context of our research. The middle layer of the behavioural hierarchy is concerned with path determination, with the steering behaviours of this layer combining to direct the character's speed and heading. Our research is primarily concerned with the locomotion layer, where actual figure animation and articulation is performed. However, it is necessary to first describe how global movement is controlled by the steering layer.

4.2. Speed and Heading Adjustment

Global motion in our simulation is produced by applying a force and torque on the creature's thorax rigid body, the target velocity specified either by user input (keyboard, joypad), or as an aggregate output from AI-driven steering behaviours (wander, seek/flee, etc.). The calculation is split into two steps—the target speed and target heading.

Speed adjustment is performed by applying an acceleration force to the creature's thorax rigid body, by default in the direction of its current heading (forward axis). However, it is also possible to move the thorax independently of heading, for example using dual analogue sticks (left stick for movement, right stick for heading).

The current target speed of the creature is found by the scalar quantity of the target velocity, curbed by the maximum speed. However, under certain conditions, in order for the creature to react appropriately to specific surface characteristics, obstacles or internal conditions, the target speed is also scaled by a normalised brake percentage parameter. The force applied to the creature's thorax is found using Newton's second law of motion $F = ma$, with the rate of acceleration controlled by acceleration and deceleration parameters.

For heading adjustment, given the body's current heading v_c and a corresponding target heading v_t,the torque τ required in rotating the creature body to a target heading is found by

$$\tau = (v_a \delta c_1) - (\omega c_2)$$

where the rotational axis v_a is found by the cross product of v_c and v_t, and the rotational angle δ by the inverse

cosine of the dot product between v_c and v_t. The c_1 and c_2 parameters represent the rates of body rotation and deceleration, respectively, allowing for adjustment to the rate of target convergence.

5. GENERATING ARTICULATED MOTION

The behavioural system corresponds to a steering layer, which actively applies force and torque on the thorax. It is the task of the locomotion layer components to produce cohesive animation around this motion—-that is, from an observer's perspective, the creature appears to be physically ambulating.

Figure 3 shows a diagram of the simulation components. The overall design is quite simple, with each of the locomotory modules (blue, red) affecting some part of the creature's anatomy (yellow) in order to balance around the thorax's global motion (green). However our design is also very robust, and allows for much dexterity and emergent behaviour. It is the aim of this section to explain the purpose, function and implementation of each module in the locomotion layer, the exception being the gait generator which is explained in the next section. Although we aim to highlight salient parameters and their functions in this section and throughout the paper, in the interest of space, we omit implementation specifics.

It is important at this stage to explain that the effects of gravity are not taken into account when the creature is actively ambulating. There are two main reasons for this. Firstly the absence of gravity removes a considerable computational strain on the rigid bodies that comprise the articulated models, and secondly it removes the need to explicitly simulate grip when climbing. However gravity is activated in certain situations, which we will highlight in Section 7.

5.1. The Stepping Modules

Each of the creature's limbs has an associated Stepping Module which, given the parameters of a step, actively moves the associated target node along a calculated path. The Stepping Module's purpose is essentially to guide the creature's foot in a parameterised arc, acting as a procedural step generator. Each Stepping Module accepts the parameters step direction, length, height, duration and type, which are passed to the module by the system's directorial gait generator. Essentially, the gait generator requests when, where and how a step is taken, and the Stepping Module acts to iteratively move the target node (foot) to this new position. The most basic source for a step path is an ellipsoid function, an example of which can be seen in Figure 4. The step here starts at position 'a' and ends at 'b', the path describing the ideal trajectory taken by the foot in transit. More advanced stepping paths can also be invoked using Bezier curve representations.

The step path is initially described in 2D Cartesian space. Localised to the foot's position and creature orientation at step invocation, the path of the target node is scaled by step length, height and duration parameters, and rotated by the step direction parameter.

5.2. Step Termination

By design, no goal position is explicitly defined at step invocation. The termination of a step occurs when the limb

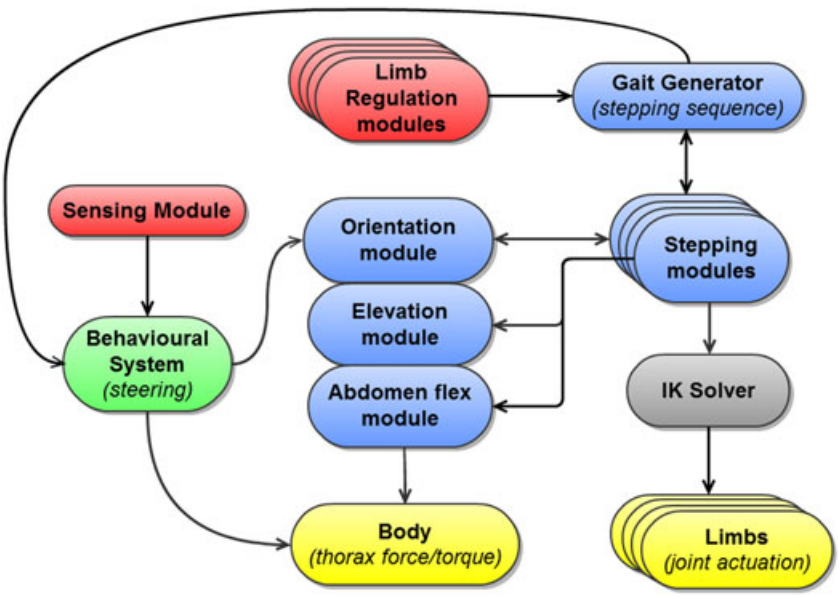

Figure 3. Component overview diagram. Green represents the steering layer, blue and red the active and passive components of the locomotion layer, and yellow the physical layer.

Comp. Anim. Virtual Worlds 2013; **24**:65–83 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

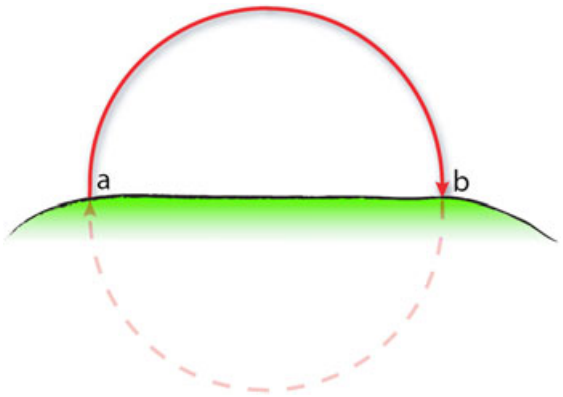

Figure 4. Step path, in an ellipsoid function, describing the ideal trajectory taken by the foot in transit, starting at position 'a' and ending at position 'b'.

(usually the foot, but can be any tested for every segment) collides with the environment or a rigid body object of significant mass. For example, in Figure 4, the path terminates at 'b' because of the fact that the limb strikes the surface. In the absence of this collision, the leg would continue to retract back to its original position at 'a'. The Stepping Modules effectively invoke 'blind' steps, where the step only terminates when the leg strikes something that impedes its path.

The detection of a possible collision between one of the limb's rigid body segments and the environment results in a callback function being called. Here, provided that the limb is active and in the process of taking a step (as opposed to colliding with the object due to falling, or being hit), the percentage of stride complete is tested against a threshold value ($\sim 2\%$). This test is necessary as it eliminates a situation where a surface collision is detected at step invocation owing to an initial intersection between limb and environment. Passing this initial test signifies that the creature has attempted to step the leg, and that in transit, a collision has occurred between it and the environment.

Although it is possible to flag certain objects (and the static environment) so that any collision between it and the creature's foot will automatically result in a stride termination, a more unified approach is to process the rate by which the limb decelerates. A sharp percentage drop in velocity signifies that the attempted stride has reached its conclusion. The absence of a large deceleration implies that the leg is still free to move for whatever reason, such as a glancing collision, low friction contact or easy displacement of the struck object. In the latter case, a limb striking a piece of detritus and knocking it out of the way would not result in a stride termination, whereas a downward step onto the same object would. Different effects and behaviour can be achieved by modifying the properties of the environment and the creature (for example, low friction glass surface; and sensitivity to step deceleration).

However, although effective in evaluating foot grip on the ground, this technique is not adequate for simulating grip when a creature is walking across a wall or ceiling. As gravity is not simulated when the creature is active, a foot striking a 90° glass surface perpendicularly would terminate just as it would on concrete, whereas in the real world, the effectiveness of the grip would depend on how much purchase the appendage's microscopic hairs could achieve.

In order to optionally simulate the effect of friction, it is necessary to subsume the physical dynamics of the collision and process the event numerically. An approximation of grip is achieved by analysing the friction properties of the collision surface and its orientation relative to gravity. The specifics of the calculation are based on an arbitrary weighted random number generator (depending on the type of behaviour required). For example, a glass surface could always result in a step termination at surface angles of 50° relative to gravity, provided that the aforementioned deceleration test is passed, with the probability of step termination at subsequent angles falling until eventually zero at 90°. Although being a rather crude method of testing grip, this technique does provide a simple mechanism for giving the appearance of tentative, laborious or abortive surface traversal.

Following step termination, the target node position is set to the foot's termination point, and the limb enters its stance phase. Although the IK solver will flex and twist the limb around this pivot point as the creature moves, there could potentially be slight movements in the foot's position because of adjustments higher up the chain. Therefore, a universal joint is also created at the termination position, effectively anchoring the foot segment to the surface. The universal joint is made weak enough so that any significant force (such as the creature being struck or falling) will break it.

5.3. Limb Regulation

The gait generator monitors and reacts to erroneous limb configurations by triggering a correctional step that brings the creature back into a balanced pose. We have identified four types of error: imbalance, over-stretching, restriction and limbo. At each simulation step, each Limb Regulation module monitors these errors and if detected raises the appropriate flag.

Imbalance. An imbalance error signifies that the foot of the limb in question is too far away from its most comfortable position. This is by far the most common error during locomotion, where imbalance errors effectively operate as the trigger for the stepping reflex mechanism that consistently moves legs back into neutral positions.

The initial position of each foot as described by the model's default skeletal pose represents its most comfortable position in relation to the body. At initialisation, origin nodes are created at these positions, and although the aforementioned target nodes move independently of the body, the origin nodes inherit the rotation of the thorax, and therefore describe the most comfortable position of each foot in relation to the body at all times, even if that

Figure 5. Comfort range of the creature's feet. Origin nodes determine the centre point of each sphere.

point is currently positioned above or below the surface. Imbalance errors are tested each frame by comparing the distance between the foot end-effector and the origin node against a comfort range parameter, unique for each leg pair. A visualisation of these comfort ranges can be seen in Figure 5.

Over-Stretching. A stretching error signifies that the limb in question is anchored on the surface, but significantly outstretched. This error aims to counter (otherwise valid) poses where the leg has successfully terminated, but cannot feasibly provide adequate support. The regulatory module tests for limb stretching by summing the current (local) rotation of each joint. If the cumulative angle between joints surpasses a minimum cumulative angle parameter ($\cong 20°$), a correctional step is requested.

Restriction. A restriction error signifies that a segment of the leg has struck the environment before the foot, potentially requiring a correctional step. Testing for such an event is simply a case of detecting a collision between the leg and the environment, as is performed for step termination.

Limbo. A limbo error signifies that the limb in question was unable to terminate after the foot completed a path trace. The limb is said to be in 'limbo' because of the fact that without correction, such a step will repeat—effectively stranding the creature at its current position. Although the prior three errors result in a request for a standard correctional step, a 'limbo' error requires a more complicated foot probing sequence, detailed in Section 6.3.

5.4. Body Orientation

The rotation of the body around the up vector axis (or the yaw component) was previously outlined in Section 4.2, where a target vector is translated into forces that adjust the heading of the creature. The actual localised plane by which this steering occurs is dependent on the other two components of the creature's orientation, which are derived directly from the combined positions of the feet.

The pitch component of the creature's orientation is calculated by finding the normalised average vector between adjacent feet positions along the creature's x-axis, parallel to its heading, as seen in Figure 6. The relative positions of the target nodes along the z-axis are ignored, as their influence will be calculated in the roll component. Similarly, the roll component is given by the normalised average vector between feet positions perpendicular to the creature's heading, along the z-axis. Here, the x-axis is ignored, having been previously taken into account in the pitch component.

In order to update the creature's rotations along the three principle axes, a similar method to that previously used to steer the thorax is used. The orientation module passes the calculated roll and pitch vectors to the thorax rigid body callback, which proceeds to translate the rotational target vector into the appropriate torque force.

The calculations of both these components can be simplified by only taking into account the difference in elevation between front and back limb couplings, disregarding the influence of the middle set(s) of feet. Moreover, the orientation module can be switched between the alternative methods, with more accurate calculations requested only when the creature detects a sufficiently uneven surface or a potentially large change in elevation.

Although the callback is called every frame, continually updating the body orientation by applying the appropriate torque, by default, the actual target rotations are only updated when a limb successfully anchors on the surface. An alternative approach is to constantly update the target vectors by measuring the relative difference between

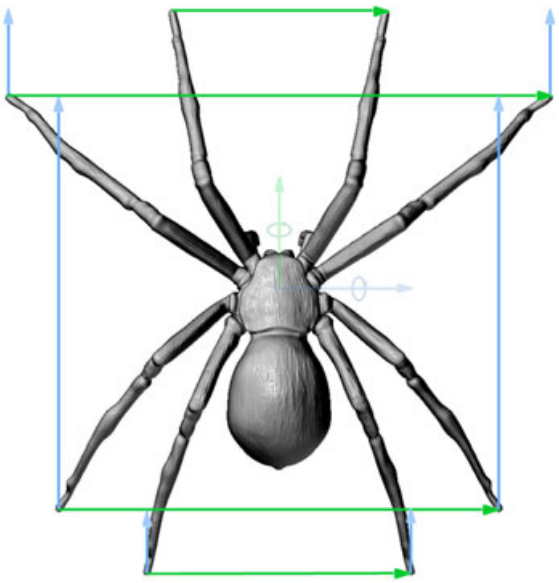

Figure 6. Pitch (blue) and roll (green) components of body orientation are derived from the average positions of the creature's feet.

Comp. Anim. Virtual Worlds 2013; **24**:65–83 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

all foot positions, even when in transit. Although this can result in the generation of more natural and continuous motion (especially lighter creatures such as ants and beetles), it can also result in every step being amplified in the orientation of the creature, the animation bordering on the cartoon as the stepping legs generate a rhythmic bobbing of the thorax and abdomen. If undesired, this effect can be dampened by appropriate adjustment to the parameters.

Although the c_1 and c_2 parameters of the torque equation defined the rotational rate and deceleration in yaw component steering, here, they are used to specify how quickly the creature reacts to the surface being traversed. Varying c_1 and c_2 allows for faster or slower reactions to orientation changes, which can not only be changed during simulation to reflect different creature gaits, locomotion speeds and underlying terrain structure, but also its global sensitivity to changes in surface characteristics—allowing for more jittery or relaxed reactions to the environment. For example, setting c_1 to a higher value for a domestic house spider would reflect its lighter, more erratic nature compared to a larger tarantula, which would tend to maintain a smoother average orientation when traversing the same irregular surface.

In pitch calculations, it is also possible to add bias to the front pair or quadrant of limbs. This allows for extra sensitivity to changes in frontal elevation, useful in traversing large orientation changes approaching and exceeding 90°. As it is the front legs that will almost always contact the new surface first, this frontal bias can aid the creature in smoothly navigating radical changes in orientation. As with the reaction time variable c_1,the bias can be varied during simulation to reflect circumstances.

5.5. Body Elevation

Although the IK solver aims to maintain contact between the feet and the surface, and the Orientation Module adjusts the orientation and heading of the body, the lack of simulated gravity means that there is no force to stop the thorax and abdomen from being pushed out of position (or floating away). Although linear and angular damping forces help to curb inertial movement, a means of regulating body elevation is required.

A mass-spring model calculates the force necessary to elevate the body at the appropriate height. Its effect is to apply a consistent force on the thorax, pulling it back into a target height relative to the creature's foothold. The first step is to transform the required vectors so that they can be represented as displacements, regardless of the creature's orientation. This is achieved by taking the current position of the thorax and the average position of the anchored feet, and multiplying them by a rotation matrix derived from the creature's orientation quaternion. The y-component of the transformed vectors yields the y-axis displacement between the two. The acceleration acting on a body of mass m is found by the following equation, derived from Hook's Law of elasticity:

$$a = \left(\left(\frac{k}{m}\right)(s-d)\right) - \left(\frac{\mathrm{b}}{m}\right)v$$

where k is the spring stiffness, s the current position of the thorax, d the target rest position of the body, b the damping constant (friction) and v the velocity of the body. The value of d represents the ideal displacement of the thorax, being the average of grounded feet positions plus a target elevation.

The damping constant b is given a value of ~ 1.0, the system being critically damped, with the body converging on its target position in a single oscillation. The speed by which this convergence occurs is controlled by the spring stiffness parameter k.

5.6. Abdominal Flexing

In many arthropod species, the abdomen is connected to the thorax by the stalk-like pedicel. Although the movement and orientation of the thorax is directly tied to the connected limbs, the abdomen is free to flex semi-independently.

A pedicel joint is generated similarly to those found in the creature's limbs. The difference in this case being that the stiffness along both the hinge and perpendicular axis are made significantly laxer, allowing for the abdomen to flex under external force. The abdomen is connected to the thorax via a ball-socket joint, with active rotation along a principal axis acting to proactively flex the abdomen in situations such as when navigating large changes in orientation. Here, the pedicel joint is aligned with the average elevation of the back two pairs of limbs as the creature climbs around the feature.

5.7. Sensing the Environment

In navigating an arbitrary environment, it is necessary that the creatures have a means of proaction in order to appropriately adjust their speed and profile in preparation for especially problematic obstacles. This sensing of surrounding environmental features is analogous to an arthropod's sight, although the only function explicitly implemented for the simulation concerns the proximity of nearby objects and surface features.

Proactive environment sensing is performed by casting a number of rays in the direction of the creature's heading. The simulation requires that a (zero weight) bone be placed at the average position of each model's eyes, serving as the origin point of each ray. Specified in the creature attribute file are parameters for sight resolution, width, angle and range, which are used at initialisation to construct the Sensing Module.

By processing the average of the intersected polygon positions and normals across all sense rays, an approximation of the immediate environment is found as a scalar

distance and angle. Although being a fairly crude approximation, the average surface angle and proximity serves in providing some sense of the environment's composition directly in the creature's path. The Sensing Module is updated every simulation step; the two measurements providing a continuous snapshot of the environment.

The Sensing Module computes a braking percentage based on its proximity measurements, which is used by the Behavioural System to appropriately decelerate in preparation for potential traversal, found by simply dividing the average (relative) surface normal by a default value of 120. In a scenario where the creature is approaching a vertical wall, this calculation would yield a normalised percentage of 0.75, that is a 75% reduction in speed upon its detection, although this absolute percentage is linearly scaled based on the detected proximity of the obstacle. In a case where little (or no) surface is detected (such as near a precipice), the maximum brake is applied automatically, slowing the creature down in preparation for potential traversal over the edge.

The sensing mechanism is also linked with a parameter, which scales the radial distribution of the origin nodes to bring them closer or further away from the creature's origin (root thorax bone) depending on the detected proximal surface. This radial scale parameter enables the creature to dynamically adjust its footprint profile in preparation for the type of surface being traversed. The function of this parameter is highlighted in Section 8.3.

6. GAIT GENERATION

In order for an arthropod to remain in a balanced configuration at all times, at least half of its legs should remain on the surface in a supporting role. Furthermore, most should adhere strictly to the same stepping pattern—that is, the hard-wired rules that govern the relationship between grounded and stepping limbs.

Recalling the section on robotics, four reflex strategies were identified in [31], with parallels also found in entomological literature. Although the effect of passive and active compliance is largely automatic in the dynamics of the articulated figure, and the elevatory reflex a by-product of blind strides, it is the aim of the gait generator to simulate the stepping and search reflexes. The gait generator is designed to create generalised stepping sequences that mimic the most common gaits found in six and eight-legged creatures—ensuring that every step in every circumstance is legal and appears plausible given the current limb configuration, whereas also allowing for ample flexibility during difficult or slow traversals.

6.1. Gait Sequence

When walking across a relatively uniform and uncluttered surface, most arthropods walk in a distinct manner. These characteristic gaits, determined from studies of arthropod locomotion, fall into two basic models: the metachronal wave; and the alternating tripod gait. Although the former is epitomised in the gait of the non-insect millipedes (and therefore not applicable within the scope of the simulation), the latter is seen in all insect species, with the similar alternating tetrapod gait characteristic to arachnids, in that they are both based on the alteration of ipsilateral and contralateral legs.

The tripod gait is the best known hexapod gait. A tripod consists of the front and back legs on one side of the body, and the middle leg on the opposite side. An insect walking using a tripod gait will step with all three tripod limbs at the same time, and once these are grounded will step with the other tripod. Similar to a bipedal walk, each stepping tripod shifts the weight of the creature to the other (grounded) tripod. This ensures that three limbs remain grounded at all times, providing support for the stepping legs.

The equivalent in arachnid locomotion is known as the alternating tetrapod gait. Here, legs on opposite sides work as tetrapods, where four limbs remain on the surface whereas the remaining four limbs step. However, rather than stepping at the same time as is the case in the tripod gait, here, the limbs move in waves, with limbs from each tetrapod stepping alternatively. A diagram of this classic arachnid gait can be seen in Figure 7, with tetrapod R4-L3-R2-L1 green, and tetrapod L4-R3-L2-R1 blue.

6.2. The Stepping Reflex

The objective of the gait generator's stepping reflex is to ensure that limbs remain in a viable configuration at all times. If a limb is in an illegal position or configuration, the gait generator triggers a rectifying step. This a purely reactive method of walking, as steps are only triggered when the corresponding limb enters an erroneous state.

When a step is triggered, the gait generator passes parameters that define the step path to the appropriate limb'stepping module:

Step length. The distance between the foot and its origin node.

Step height. Taken directly from the attribute file, optionally scaled by the creature's speed.

Step direction. The normalised vector between foot and origin node.

Step type. Set in the attribute file as either eliptical or bezier; can be subsumed by a special searching step.

This stepping reflex is based on the reflex strategy previously outlined in Section 2.3. Parallels can also be drawn between its function and the mechanical aspects of legged locomotion as described in [39], where it is hypothesised that the primary requirement of an animal's locomotion control strategy is to stabilise its body around limit cycles. Similarly, in [40], it is stated that to characterise more complex responses to perturbations, one must first define the steady-state condition to which the animal will return, and that stability analysis used in concert with a template (a simple, general model that serves as a guide for control) can lead to testable hypotheses of function.

DOI: 10.1002/cav

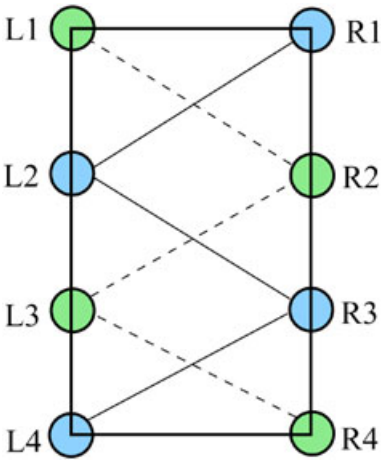

Figure 7. Stepping sequence of a classic arachnid gait.

An illegal condition will be acted upon only if the leg is currently anchored on the surface, and not if the limb is in the process of taking a step. For example, although it is entirely possible that a limb will become temporarily overstretched or be in an imbalanced state when stepping, it is only a concern if the problem persists at step termination. The only exception to this rule involves the limb being in a limbo state, the issue in this case being the reverse—the limb has failed to anchor at step culmination.

The order by which steps are triggered is regulated by placing any limb specified to be in an illegal state into a stepping queue. The gait generator algorithmically adheres to the previously described gait sequences by specifying that a queued limb is free to step provided that neighbouring limbs (e.g. in Figure 7; R2 = R1, R3 and L2) are grounded, where the neighbouring limbs of frontal and distal limbs wrap around (i.e. R1 = R4, R2 and L1).

At slow speeds (where perturbations are infrequent), this rule allows flexibility in the stepping pattern, whereas at high speeds, the creature will be forced to enter the classic tetrapod/alternating tetrapod gait.

The queue therefore effectively limits the number of steps that can be in progress at any one time (three for insects, four for arachnids), and will always process steps in order of request—thereby ensuring that the creature's stance remains stable. If the number of queued steps exceeds the maximum, then it can be inferred that the system is becoming overly imbalanced, in which case, a large deceleration force (brake percentage) is requested to the behavioural system, effectively slowing the ambulation to an eventual halt until balance is restored.

6.3. Searching Steps

Although a standard elliptical or Bezier step is taken by default, specialised steps can also be performed under certain circumstances. The previous section noted four types of limb errors that require immediate attention from the gait generator. Although limb imbalance, restriction and over-stretching can be rectified by a standard stepping request, a limbo error requires a special step.

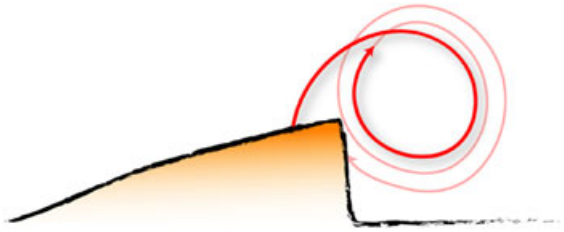

Figure 8. Searching steps at a precipice. If a step fails to terminate, the foot will move in increasingly concentric circles until it can find purchase.

There are two outstanding circumstances where limbo errors are most prevalent: either the object the foot terminated against at the last successful step has since been displaced, or the creature has approached a precipice at significant speed.

Based on insect stereotypical leg searching movements [41], the solution to a limb being in limbo is shown in Figure 8. The leg is randomly probed in increasingly concentric circles and perturbed directions, up and until it can successfully strike the surface. Although it would be possible to automatically trigger the searching steps as the path trace passes 100%, in keeping with the reactive structure of the simulation, subsequent steps following a limbo error are explicitly triggered by the gait generator (with incrementing step length and height, and randomly perturbed direction).

6.4. Kinematic–Dynamic Switching

Although our system can physically simulate the creatures at all times, we previously showed in [30] that a simpler approach of only testing for foot contact points can also produce viable animation over uniform and uneven surfaces. As the underlying calculations of our simulation are kinematic, it is possible to normally animate the creatures using only this simple technique, with dynamic simulation only activated under certain circumstances. For example, in Figure 9, the environment largely consists of flat terrain with small inclines, where the animation will not really benefit from full physical simulation. However, if the creature's speed is sufficiently curbed it is likely that the local environment is more difficult to traverse or interactive, prompting a switch to dynamic simulation. In Figure 9, this is achieved simply by switching between kinematic and dynamic walking when the speed falls below 5 cm/s.

7. GRAVITY ACTIVATION

As previously mentioned, gravity is not simulated during creature locomotion. However, there are certain situations where a gravitational effect becomes paramount in generating cohesive animation.

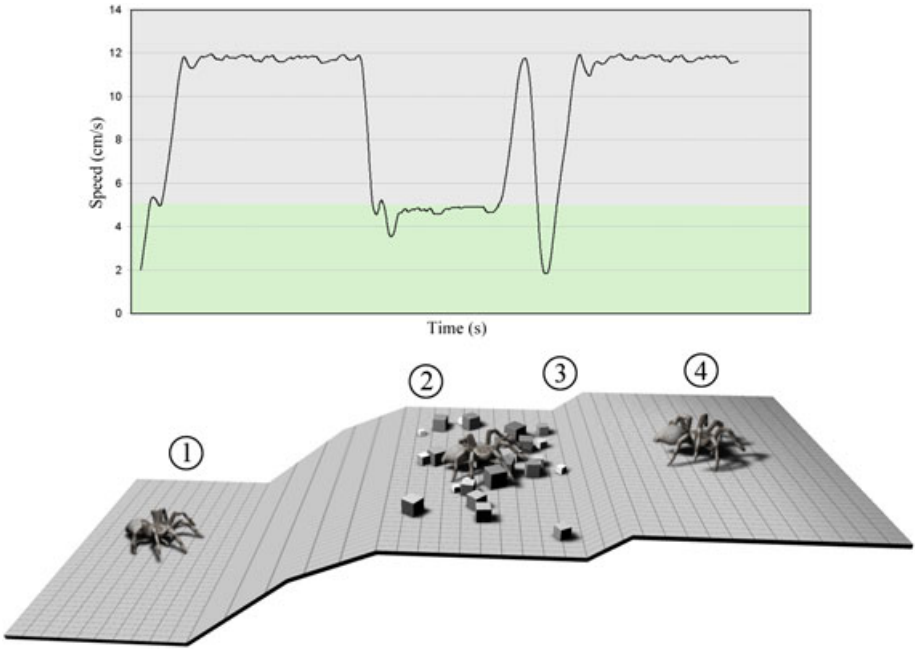

Figure 9. Time lapse screenshot of a tarantula navigating a semi-cluttered surface using kinematic (<5 cm/s) and dynamic (>5 cm/s) simulation. Inset graph shows relative speed of creature with time.

Gravity is activated at every instance where less than half of the creature's limbs are anchored on the surface. The rule aims to simulate the fact that in order to remain in a stable stance, half of the limbs will continually be in a supporting role. Although gravity is active, all component calculations are suspended apart from a function within the gait generator that tracks which feet are currently in contact with the surface. Therefore, although all direct animation callbacks are inactive, the creature remains able to check for a valid stance configuration. Herein, if contact is re-established between enough of the creature's feet and surface, all simulatory components are reactivated and gravity is disabled.

When gravity is activated, all joints in the articulated figure maintain their last angular row restriction, effectively weakly locking the limbs in place at the last rotational angle as requested by the inverse kinematics system. Although external forces such as striking an object or surface while falling at a significant velocity will temporarily buckle the limb, the active angular rows will eventually flex the limb into its prior configuration.

There are numerous example situations that highlight the dynamic nature of gravity activation (and eventual deactivation). At initialisation, creatures are simply dropped onto the surface of the environment, their locomotion system only activating when the feet contact the ground. The implementation of a creature's grip on the surface yields whether a limb can successfully grasp the surface or not. In certain conditions, a situation can arise where a creature undergoing locomotion fails in maintaining an adequate foothold on the surface. Activating gravity at this stage is both physically realistic (the creature cannot sufficiently support its weight), and advantageous from an animation point of view (the real-life equivalents being far from infallible themselves). Given appropriate parameters, the combination of attempting to climb up a vertical surface and gravity activation can lead to complex motion where multiple momentary losses in grip give weight and organicity to the animation.

Lastly, although the locomotory system is very robust, there are certain situations where coherent motion cannot be maintained—such as the body being too out of balance with the limbs, loss of supporting foothold, and so on. In this instance, gravity activation acts such as a panic button for the animation system, forcing the system to reset. In the unlikely situation where a creature lands on its back, or cannot find enough purchase, random pulses of excitement to the limbs can be used as a righting strategy.

8. RESULTS

We intend to highlight in this section that although it is possible to produce stable and natural locomotion over flat and uneven terrain using simpler or data-driven approaches, embodying the creature in its environment facilitates the situated coordination required to successfully (and realistically) traverse within arbitrary environments.

Our test environments are designed to evaluate our creatures' ability to navigate specific types of landscape features. We will focus on analysing the animation generated in environments where effective navigation requires

DOI: 10.1002/cav

a higher degree of coordination and dexterity (which only our simulation can produce), such as angular and cluttered surfaces. A video featuring example animations produced by our system can be seen in [42].

8.1. Uneven Surfaces

An uneven surface presents the most typical locomotory challenge, paramount in evaluating the effectiveness of the simulation. A terrain environment (with heightmap elevation) was previously used in [31] to evaluate the traversal of this type of feature. The environment contained both even and significantly perturbed features in order to assess not only whether plausible animation can be continuously generated over difficult terrain, but whether a rhythmic walking gait can also be achieved over more uniform sections. The example video [42] also shows numerous examples of spiders efficiently navigating uneven terrain.

8.2. Climbing a Wall

Of particular interest during the development of the simulation was the degree of dexterity and coordination displayed by real life arthropods in navigating over angular surfaces, where a complex range of motions are performed in going from one surface to another. The flexibility and animation complexity possible using our technique becomes increasingly apparent when we substitute a relatively uniform environment to that consisting of angular corners.

The inside of a cube, although being geometrically simple, poses constant challenges in terms of climbing from one surface to another. Although an uneven landscape requires constant small adjustments in gait, the act of climbing onto perpendicular surfaces presents a far greater challenge, demanding constant evaluation and correction to limb placements in order to maintain balance and generate net forward momentum.

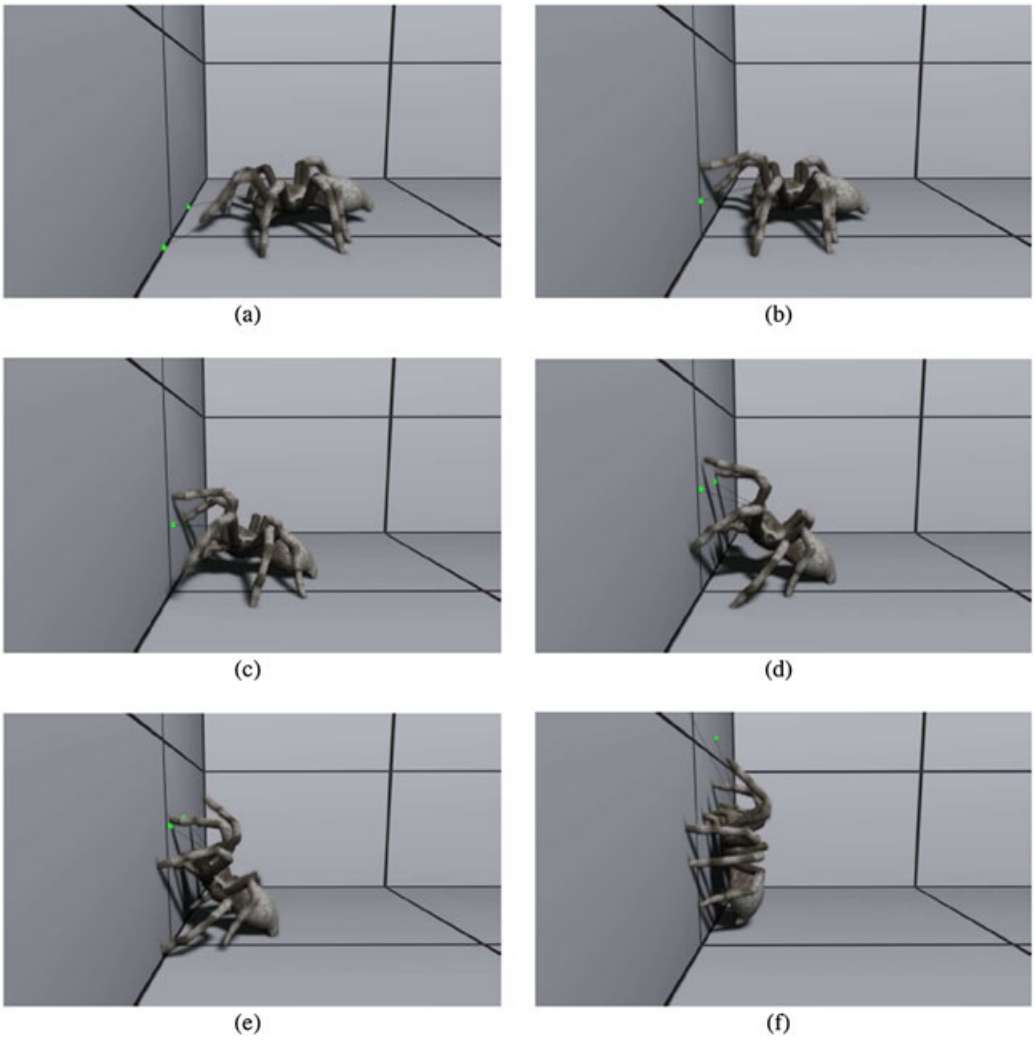

Figure 10. Select frames of a tarantula climbing onto a vertical wall. Green dots show the intersection bounds of the rudimentary sensing mechanism, used to judge the proximity and relative angle of objects in the creature's path.

Figure 10 shows a series of screenshots where a tarantula is climbing onto a vertical wall. In order to effectively analyse the creature's climbing ability, the surface of the cube is given sufficiently high friction properties so that all but the most glancing of steps will successfully terminate.

In Figure 10(a), the spider is walking on the ground, approaching the wall. By this point, it has sensed a surface at 90° relative to its current orientation. This causes the creature to slow down (through the link between the Sense Module and the steering behaviour brake percentage)—the rate of deceleration based on both the surface angle and proximity.

By Figure 10(b), the creature has slowed down considerably and moving very cautiously, this slower speed resulting in less need for constant, or even consistent, adjustment steps. The front pair of legs have already terminated at a higher elevation relative to the remaining legs, resulting in the Orientation Module adjusting the pitch (and roll) components of the body in reaction to the change in balance. Every consequent step will be along this new forward heading.

For the second pair of legs, this orientation change also results in the origin nodes becoming suspended in mid-air. The first effect this has is to quickly cause the pair to enter a non-balanced state (due to the imbalance caused by now too-distant origin nodes). The aforementioned orientation and resultant new step-path direction causes the two legs to take steps that struggle to strike a comfortable position (as most positions satisfying this goal are now in mid-air). Typically, the steps will terminate after approx 270° of the elliptical path (or 75% termination), with the creature potentially taking more circular, searching steps attributable to limbo errors. Eventually, this will result in the second pair of legs finding purchase on the wall, as can be seen in Figure 10(c).

At this stage, the frontal limbs are perceived to be tapping on the wall because of the inherent imbalance of the system, mimicking similar behaviour found in

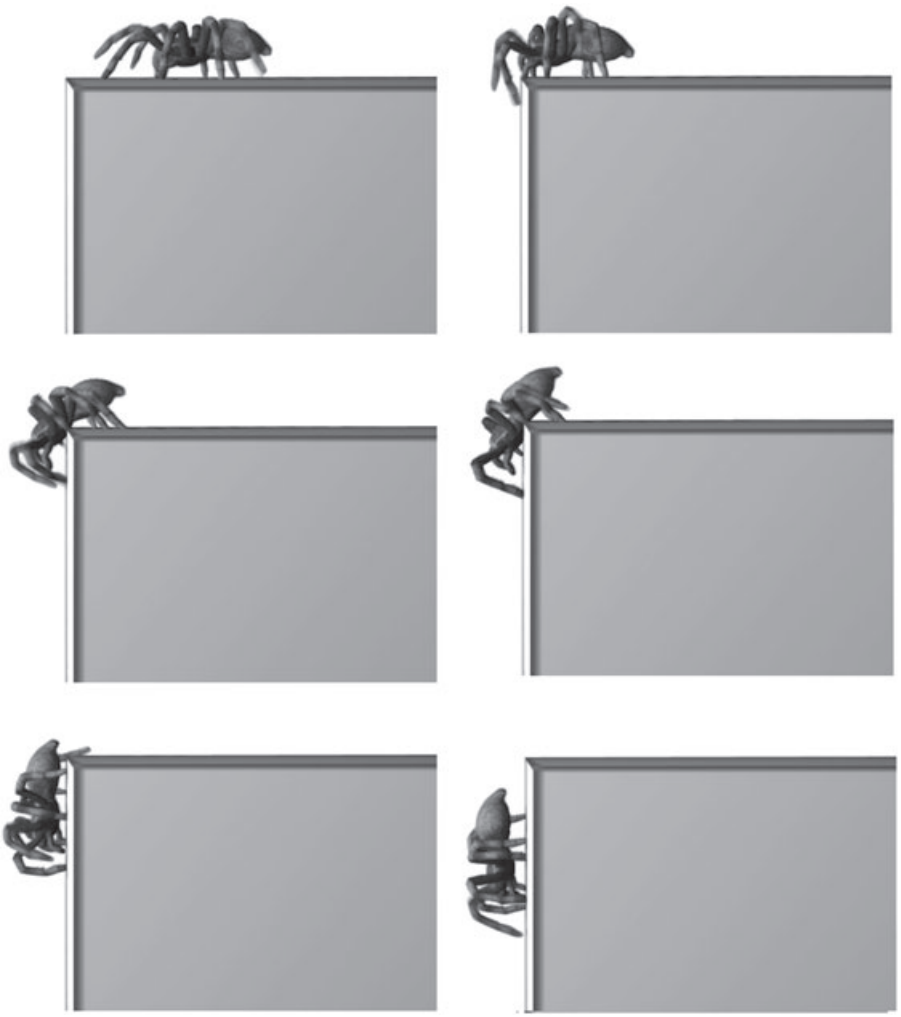

Figure 11. Side view of tarantula climbing over the edge of a precipice.

Comp. Anim. Virtual Worlds 2013; **24**:65–83 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

real-life arthropods as the creature is consistently fighting between moving forward, balancing and finding a reasonable foothold for each of its eight legs. Every step results in a change of body orientation, which in turn affects the overall balance of each limb placement.

Figure 10(d) and (e) represents before and after 'tipping point' in the traversal. By Figure 10(d), half of the tarantula's legs are touching the wall, pivoting the body towards and eventually past 45°. Once this orientation is reached, the creature will begin to accelerate (due to the sensors now detecting a shallower surface ahead, combined with increased stability). Eventually, all eight limbs have struck and anchored on the vertical wall, resulting in the creature's body levelling out at an orientation parallel to this new surface at Figure 10(f).

8.3. External Corners

The functionality that enables the simulated Arthropods to climb around corners also allows the creatures to climb down onto 270° shear surfaces relative to its position (Figure 11), although the technique as outlined in the previous section will potentially result in a much more laborious traversal. The reason for this is that although steps usually terminated at $< 50\%$ path in the previous example, here, the legs are likely to terminate at $\cong 75\%$ path. In other words, steps will terminate much earlier at a corner than a precipice, where the creature has to initially reach over the edge and grasp at a lower elevation. Moreover, the wall itself acts as a barrier to motion in Figure 10, as creature approaching the wall will be physically inhibited as the thorax/abdomen presses against the geometry. In the inverse corner's case, an inattentive approach is far more likely to result in a problematic traversal because of stagnation caused by over-caution or even falling over the edge.

Efficiency when traversing an incline can be significantly improved by adjusting the radial scale parameter outlined in Section 5.7. The parameter describes the radial distance offset of the origin nodes relative to the creature's body (or to be more precise, the skeletal root node). Although radial scale $= 1$ denotes the default position, smaller values result in the origin nodes (and therefore ideal foot position) moving closer to the creature's body. Similarly if radial scale > 1, the creature's leg span at rest is increased.

Moving the origin nodes closer to the body in effect reduces the physical profile of the creature. By linking this parameter to the creature's visual system, the default leg span can be adjusted in real-time as the environment indicates. The creature's step speed will also be affected by the radial scale parameter, as a decrease in profile will lead to imbalance errors at slower speeds. This will result in shorter, shallower steps occurring at an increased rate. This is an advantage when climbing over an incline, as the creature can quickly move its feet around the edge and re-orientate. Moreover, the technique can further serve as an advantage to the corner traversal of the previous section, as an increased leg span allows the creature to search for and grasp onto the wall more effectively.

8.4. Cluttered Surfaces

The advantage of a reactive, physically simulated arthropod is further evident when we introduce rigid body objects into the environment. Again, a motion data driven animation would simply not be able to react realistically in the presence of interactive objects, something which is automatic in a more dynamic approach. As each leg is physically simulated, the creatures are not only able to disrupt lightweight objects in their path but also actively walk on top of larger, heavier objects.

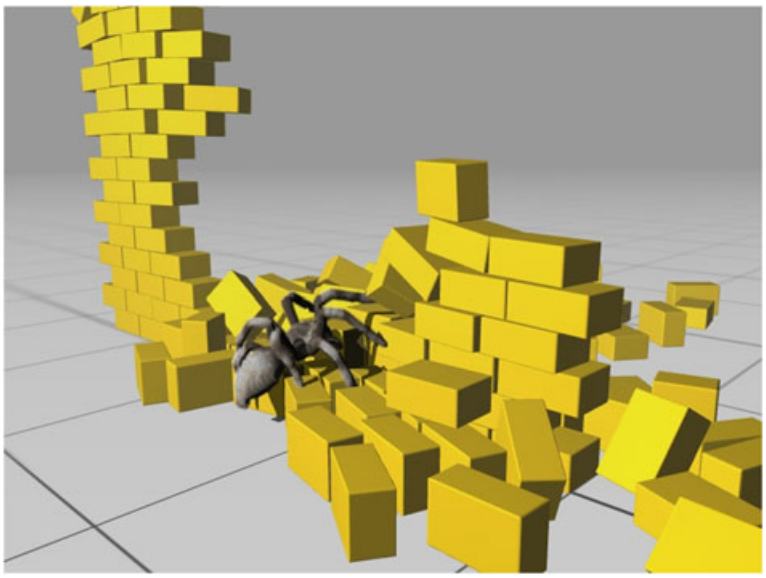

Figure 12. Tarantula climbing onto a pile of physically simulated blocks.

Figure 12 shows an example of a tarantula climbing over a pile of moderately weighted rigid bodies. As each step the creature takes is situated, and termination occurs upon any valid collision between foot and object, the creature is capable of seamless locomotion from static surface to rigid body pile.

9. CONCLUSION

The novel aspect of our work is a result of many interdependant components, some of which were derived from biological and robotics concepts, others from artificial life and behavioural animation. The contribution of the work documented in this paper is therefore synergistic, the unique features of which are outlined below.

Navigation of Arbitrary Surfaces. The simulation is capable of generating natural looking animation for walking arthropods within arbitrary environments, regardless of structural complexity. The exploration and navigation of the environment is completely autonomous, with no pre-processing or explicit labelling of specific environment features.

Behavioural Abstraction. A process of inverse dynamics allows for an abstraction between steering and underlying locomotion mechanism. Although the simulation of higher-level (instinctive) behaviours are outside the immediate scope of our research, this abstraction facilitates the development of higher level species specific ethological phenomena without having to specify the specifics of motor coordination (and therefore locomotion animation). This abstraction also facilitates the simulation of crowds, which are commonly implemented using steering behaviours.

Increased Vividity. Although more biomechanically accurate models of specific insects have been developed, our kinematic–dynamic hybrid approach is capable of generating a comparable level of motion complexity using far more complex and visually realistic models.

Scalable Precision. Although our implementation is capable of complete physical simulation of the creature's articulated figure, an underlying kinematic approach means that the vast majority of the simulatory components can produce viable animation using a simplified model, where only the translations of the feet are physically simulated.

Creature–Object Interaction. By physically simulating the creatures and the virtual world around them, we achieve a level of agent–environment interaction simply not possible using more abstract representations. Because of their physical presence and dynamic motion generation, the virtual arthropods are capable of climbing, affecting and reacting to rigid-body objects.

Admittedly, physically simulating an arthropod's interaction with rigid bodies is a far simpler task than the equivalent for quadrupedal and bipedal locomotion owing to their extremely stable posture and low centre of gravity. However, our implementation produces interactive, situated animation at a level of realism we have yet to witness in any real-time animation of similar creatures, including biological and robotic simulations, or commercially available animation software and games.

9.1. Limitations and Future Work

The primary limitation of our approach is that although the morphology of arthropods allows for the effects of gravity to be disregarded during locomotion, the technique does not transfer well to creatures of considerably different structure. The majority of arthropod species are similar enough in morphology that we can effectively simulate any of them, given an adequate model, creature attribute parameters and gait controller. It is their light mass, low centre of gravity and extremely stable posture that allows for an effective abstraction between behaviour and underlying control, and we do not believe that our system can easily transfer to creatures that rely heavily on proactive balance to ambulate, such as mammals (including humans).

Although automatic gravity activation enables the simulation of falling when contact with the surface is lost, its absence during active locomotion limits the interaction possible with other objects in the environment. For example, a creature walking across an overhanging object would not cause it to topple as a direct result of its supported weight. Viable interactive animation can only occur around light objects, or objects heavy enough to be considered immovable. One possible solution is to physically simulate locomotion (as in robotics) when walking on the ground, with our hybrid approach being activated when climbing.

Although our approach generates extremely vivid animation in many circumstances, there are certain types of surface features where the current approach results in extremely laborious navigation. For example, edges that require even larger orientation changes than those outlined in this paper pose an increasingly difficult task as the angle sharpens. One potential failure case concerns a situation where the creature's thorax catches on the geometry. We can currently rectify this problem by momentarily disabling collisions between the thorax/abdomen and the environment. We are currently exploring a more unified approach, where the elevation and orientation modules react automatically when this situation is detected.

Our current work has concentrated on simulating a single spider. Although it would certainly be possible to simulate many arthropods at once using the full technique outlined in this paper, a more interesting approach could be to introduce a level-of-detail system. As previously mentioned in Section 6.4, our system has a degree of scalable precision, facilitating switching between our previous kinematic and our new hybrid method based on circumstance. A more advanced switching system could utilize simple key-frame motion sequences for background creatures, or ones traversing uniform terrain, with our procedural technique only being applied in more difficult or foreground situations.

Finally, we are also exploring methods of simulating many creatures in parallel using Nvidia's CUDA

Comp. Anim. Virtual Worlds 2013; **24**:65–83 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

architecture (Nvidia Corp, Sta. Clara, CA, USA), facilitating not only the ability to simulate large groups procedurally but also to refine the aesthetical or locomotive behaviour of the creatures. In conjunction with simulated annealing or evolutionary algorithms, parameters could be finely tuned for navigating extreme changes in orientation. There are a large number of manually defined parameters in our simulation, and although we have been able to invoke smooth transitional animation when navigating angular surfaces through trial and error, such techniques could be used to make traversal even more efficient.

REFERENCES

1. van Welbergen H, van Basten BJH, Egges A, Ruttkay ZM, Overmars MH. Real time animation of virtual humans: a trade-off between naturalness and control. *Computer Graphics Forum* 2010; **29**(8): 2530–2554.
2. Skrba L, Revéret L, Hétroy F, Cani M, O'sullivan C. Quadruped animation. In *Eurographics '08, State-of-the-Art Report*, Crete, Greece, 2008; 7–23.
3. Kovar L, Gleicher M, Pighin F. Motion graphs. *ACM Transactions on Graphics* 2002; **21**(3): 473–482. DOI: 10.1145/566570.566605.
4. Lee Y, Wampler K, Bernstein G, Popović J, Popović Z. Motion fields for interactive character locomotion. *ACM Transactions on Graphics* 2010; **29**(6): 1–8. DOI: 10.1145/1882261.1866160. Article 138.
5. Fang AC, Pollard NS. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics* 2003; **26**(3): 417–426. DOI: 10.1145/882262.882286.
6. Gibson D, Oziem D, Dalton C, Campbell N. A system for the capture and synthesis of insect motion. *Graphical Models* 2007; **69**(5–6): 231–245.
7. Meredith M, Maddock S. Real-time inverse kinematics: the return of the Jacobian. *Technical report*, Department of Computer Science, University of Sheffield, 2004.
8. van Basten BJH, Stüvel SA, Egges A. A hybrid interpolation scheme for footprint-driven walking synthesis. In *Proceedings of Graphics Interface 2011 (GI '11)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2011; 9–16.
9. Natural Motion. Dynamic motion synthesis, (April 2011). http://www.naturalmotion.com/euphoria.
10. Yin K, Loken K, van de Panne M. Simbicon: simple biped locomotion control. *ACM Transactions on Graphics* 2007; **26**(3). DOI: 10.1145/1276377.1276509.
11. Wang J, Fleet DJ, Hertzmann A. Optimizing walking controllers. *ACM Transactions on Graphics* 2009; **28**(5): 1–8. DOI: 10.1145/1661412.1618514.
12. Coros S, Beaudoin P, van de Panne M. Generalized biped walking control. *ACM Transactions on Graphics* 2010; **29**(4): 1–9. DOI: 10.1145/1778765.1781156. Article 130.
13. Shapiro A, Pighin F, Faloutsos P. Hybrid control for interactive character animation. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, Washington, DC, USA, 2003; 455.
14. Bowerman RF. The control of walking in the scorpion. I. Leg movements during normal walking. *Journal of Comparative Physiology* 1975; **100**: 183–196.
15. Bowerman RF. Arachnid locomotion. In *Locomotion and Energetics in Arthropods*, Herreid CF, Fourtner CR (eds). Plenum Press, New York, 1981; 73–102.
16. Delcomyn F. Motor activity during walking in the cockroach Periplaneta Americana II. Tethered walking. *Journal of Experimental Biology* 1973; **59**: 643–654.
17. Cruse H, Frantsevich L. The stick insect Obrimus asperrimus (Phasmida, Bacillidae) walking on different surfaces. *Journal of Insect Physiology* 1997; **43**: 447–455.
18. Macmillan DL. A physiological analysis of walking in the American lobster (Homarus americanus). *Philosophical Transactions of the Royal Society* 1975; **270**(901): 1–59.
19. Holmes P, Full RJ, Koditschek D, Guckenheimer J. The dynamics of legged locomotion: models, analyses, and challenges. *SIAM Review* 2006; **48**(2): 207–304.
20. McKenna M, Zeltzer D. Dynamic simulation of autonomous legged locomotion. *SIGGRAPH Computer Graphics* 1990; **24**(4): 29–38. DOI: 10.1145/97880.97882.
21. Beer RD. *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
22. Beer RD, Gallagher JC. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behaviour* 1992; **1**(1): 91–122.
23. Beer RD, Chiel HJ, Quinn RD, Espenschied KS, Larsson P. A distributed neural network architecture for hexapod robot locomotion. *Neural Computation* 1992; **4**(3): 356–365.
24. Cruse H, Kindermann T, Schumm M, Dean J, Schmitz J. Walknet a biologically inspired network to control six-legged walking. *Neural Networks* 1998; **11**(7-8): 1435–1447.

25. Cruse H, Bläsing B, Dean J, Dürr V, Kindermann T, Schmitz J, Schumm M. Walking - Biological and technological aspects. In *CISM Courses and lectures*, Pfeiffer F, Zielinska T (eds). Springer, Wien, New York, 2004; 81–118.
26. Cruse H, Durr V, Schmitz J. Insect walking is based on a decentralised architecture revealing a simple and robust controller. *Philosophical Transactions of the Royal Society* 2006; **365**: 221–250.
27. Alshurafa NI, Harmon JT. Artificial spider: eight-legged arachnid and autonomous learning of locomotion. In *Unmanned Systems Technology VIII, Proceedings of The International Society for Optical Engineering, volume 6230*, Kissimmee, Florida, USA, 2006.
28. ap Cenydd L, Teahan WJ. Arachnid Simulation: scaling arbitrary surfaces. In *EGUK Theory and Practice of Computer Graphics*, Kent, Canterbury, UK, 2005; 125–131.
29. Sullivan C. Technical Animation using maya and mental ray. *Master's thesis*, Bournemouth University, 2006.
30. Murdock A. *Boris the Spider*. Autodesk Master Class, SIGGRAPH Boston, 2006.
31. ap Cenydd L, Teahan WJ. The dynamic animation of ambulatory arthropods. In *EGUK Theory and Practice of Computer Graphics*, Bangor, Wales, UK, 2007; 21–28.
32. Beer RD, Chiel HJ, Sterling LS. Heterogeneous neural networks for adaptive behaviour in dynamic environments. *Advances in Neural Information Processing Systems* 1989; **1**: 577–585.
33. Brooks RA. A robot that walks; emergent behaviors from a carefully evolved network. MIT AI Lab Memo 1091, February 1989.
34. Espenschied KS, Chiel HJ, Quinn RD, Beer RD. Leg coordination mechanisms in the stick insect applied to hexapod robot locomotion. *Adaptive Behaviour* 1993; **1**(4): 455–468.
35. Espenschied KS, Chiel HJ, Quinn RD, Beer RD. Biologically-inspired hexapod robot control. In *Proceedings of the Fifth International Symposium on Robotics and Manufacturing*, Maui, Hawaii, 1994; 14–18.
36. Torus Knot Software. OGRE (Object-oriented graphics rendering engine). April 2011. http://www.ogre3d.org.
37. Newton Game Dynamics. Newton game dynamics physics engine. April 2011. http://www.newtondynamics.com.
38. Reynolds C. Steering behaviors for autonomous characters. In *Game Developers Conference*, San Jose, California, 1999.
39. Koditschek DE, Full RJ, Buehler M. Mechanical aspects of legged locomotion control (Invited Paper). *Arthropod Structure and Development* 2004; **33**: 251–272.
40. Full RJ, Kubow T, Schmitt J, Holmes P, Koditschek D. Quantifying dynamic stability and maneuverability in legged locomotion. *Integrative and Comparative Biology* 2002; **42**(1): 149–157.
41. Durr V. Stereotypic leg searching movements in the stick insect: kinematic analysis, behavioural context and simulation. *Journal of Experimental Biology* 2001; **204**(9): 1589–1604.
42. Realtime Arthropod animation video, April 2011. http://vimeo.com/user7178755/realtime-arthropod-animation.

AUTHORS' BIOGRAPHIES

Llyr ap Cenydd is a researcher at the School of Computer Science, Bangor University, UK. He obtained his PhD in 2008. His principal research interest is in the field of real-time computer graphics, especially areas that relate to biological subjects such as behavioural animation, artificial life, biomedical simulation and visualisation.

Bill Teahan is a lecturer in the School of Computer Science, Bangor University, UK. He obtained a PhD in 1998 from the University of Waikato, NZ. He leads the Artificial Intelligence and Intelligent Agents research group at Bangor and has broad interests in many areas including natural language processing, data mining, multi-agent systems, machine learning and artificial life.

DOI: 10.1002/cav

APPENDIX

Table I. The main parameters of the animation system.

Category	Name	Function	Type
General	Scale	Global scale of creature	Float
	Initial_position	Initial position	Vector3
	Initial_orientation	Initial orientation	Quatern.
	Thorax_mass	Base mass of creature's thorax	Float
	Abdomen_mass	Base mass of creature's abdomen	Float
	Limb_seg_mass	Base mass of creature's limb segments	Float
Steering	Max_speed	Max speed of creature	Float
	Max_acceleration	Max acceleration of creature	Float
	Acceleration_rate	Acceleration rate of creature	Float
	Deceleration_rate	Deceleration rate of creature	Float
	Turn_rate_c1	Speed of heading adjustment (yaw)	Float
	Turn_rate_c2	Speed of heading convergence (yaw)	Float
	Wander_displacement	Displacement of virtual wander circle	Float
	Wander_radius	Radius of wander circle	Float
	Steering_rate	Rate of wander steering	Float
	Proximity_brake_max	Maximal factor to curb speed	Float
Sense	Sense_range	Range of creature's sense rays	Float
	Sense_angle	Frontal elevation angle of sense rays	Float
	Sense_fov	Sense field of vision	Float
	Sense_density	Number of rays in creature's vision	Integer
	Max_radial_scale	Maximum radial scale	Float
	Min_radial_scale	Minimum radial scale	Float
Stepping	IK_rate	Maximum rate of IK solver	Float
	Max_step_height	Maximum height of step	Float
	Min_step_height	Minimum height of step	Float
	Max_step_duration	Maximum duration of (360°) step	Float
	Min_step_duration	Minimum duration of (360°) step	Float
	Termination_sensitivity	Velocity reduction required for step termination	Float
Thorax	Thorax_elevation	Target elevation of thorax above feet	Float
	Orientation_rate_c1	Rotational rate of thorax (pitch/roll)	Float
	Orientation_rate_c2	Convergence rate of thorax (pitch/roll)	Float
	Frontal_limb_bias	Default bias of orientation towards frontal limbs	Float
	Update_on_termination	Orientation target is updated only at step termination	Bool
Leg (x)	Stance (x)	Initial foot adjustment from default skeletal pose	Vector3
	Comfort_range (x)	Comfort range of foot in relation to origin node	float
	Joint (x,j)	Joint element, where j is joint number for leg x	Null
	Joint (x,j) : axis	Rotational hinge axis of joint	Vector3
	Joint (x,j) : max	Local maximum angle of rotation for joint	Float
	Joint (x,j) : min	Local minimum angle of rotation for joint	Float
	Joint (x,j) : max_range	Maximum comfort range for joint (before restriction)	Float
	Joint (x,j) : min_range	Minimum comfort range for joint (before restriction)	Float
	Joint (x,j) : stiffness	Stiffness of joint	Float
	Joint (x,j) : elasticity	Elasticity of joint	Float