# The Motion Dynamics of Snakes and Worms

Gavin S. P. Miller

Alias Research Inc.

110 Richmond St. East,

Toronto, Canada. M5C 1P1.

#### Abstract

Legless figures such as snakes and worms are modelled as mass-spring systems. Muscle contractions are simulated by animating the spring tensions. Directional friction due to the surface structure is included in the dynamic model and legless figure locomotion results. Various modes of locomotion are described.

Keywords: Modeling, Deformation, Elasticity, Dynamics, Animation, Simulation, Locomotion, Rendering, Texture.

CR categories: I.3.5 - Computational Geometry and Object Modeling (Surface and object representations); I.3.7 - Three-Dimensional Graphics and Realism (Color, shading, texture and animation).

## 1. Introduction

Modelling biological forms using computer graphics poses several difficult problems. The models must look convincing both in their static appearance and in the way they move. Legged figures, based on hierarchies of transformations, have received considerable attention in recent years. Both kinematic and dynamic models have been presented in the literature [5]. Whilst they are successful at animating skeletal structures, the problems of modelling skin and muscle remain . Waters presented a facial animation technique which modelled the effects of muscle tensions over a region of skin [13]. Whilst successful in that application, the model was purely geometrical rather than dynamic. It did not respond in a physically realistic fashion to external forces. For animations to be flexible and realistic, biological forms should not only work in isolation but they should be able to interact with each other and with their environment.

Elastically deformable models simulate the interaction of objects with external constraints by modelling the physical properties of the materials [12]. By modelling biological structures in this way, it should be possible to create life-like animations. The models will be made to look "alive" by animating the elastic properties of the muscles as a function of time.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Unfortunately, of course, real plants and animals have highly complex structures, leading to complicated and extremely expensive models. However, certain classes of creature are elegant in their simplicity, namely snakes, caterpillars and worms.

### 2. Motivation

Animals have been a favourite topic of animation from its earliest days. Being either loveable or frightening, they have a strong emotive effect on an audience. At the same time, the real-life counterparts are notoriously difficult to train as well as being potentially dangerous. Puppets have been used to good effect although they either require direct physical controls or tedious alteration for every frame [3]. Exceptions to this include models which have one actuator for each degree of freedom of the creature. Computer animation holds the promise of creating subtle motions automatically and synthesising large numbers of creatures without additional work on the part of the animator. Reynolds modelled the interaction of schools of fish and flocks of birds [11]. The individual creatures were comparatively simple, but collision avoidance strategies led to complex and beautiful motions.

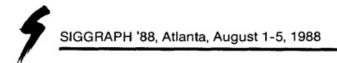
Individual snakes and worms, on the other hand, change shape with every frame. They slither over the ground in a way which depends on how well they grip it. They deform elastically when external forces are applied. A simple model with these characteristics is described in the next section.

#### 3. Modelling Elastically Deformable Strands

Snakes and worms have complex internal structures. For the purposes of this paper, a greatly simplified model was used. Each segment of the creature was modelled as a cube of masses with springs along each edge and across the diagonal of each face. For each time interval, the spring lengths and spring length velocities were used to compute the forces exerted on the masses at the end of each spring.

$$f = k (L - l) - D \frac{dl}{dt}$$

where f is the force along the spring direction, k is the spring constant, D is the damping, l is the current length of the spring and L is the minimum energy spring length. (L is animated as a function of time to simulate muscle contractions.) In addition, external forces such as gravity were computed, and the total force was divided by the mass to give the acceleration. The new position was then computed by integrating the acceleration twice with respect to time.



$$\vec{x}_p = \frac{1}{m_p} \iint \vec{f}_i \, di \, di$$

where  $\vec{x}_p$  is the position of point p,  $m_p$  is the mass of the point and  $\vec{f}_t$  is the total force acting on the point. This simple mass spring model is adequate to describe many physical objects such as pieces of rope and strands of hair. (Unfortunately a head of such hair would be extremely expensive to compute). The edge and diagonal springs together control the Young's modulus of the strand, whilst the diagonal springs affect the shear and twisting moduli.

The method of integration appropriate to compute the position of each point mass depends in part on how external constraints are implemented. Impulse-based collisions detect whether the motion of a point intersects a surface. If it does, the new position and velocity are computed analytically (see Figure 1).

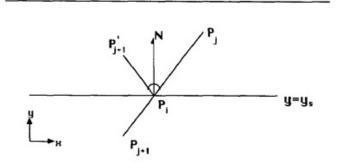


Figure 1. Point-plane constraint intersection.

For a point mass travelling from  $P_j$  to  $P_{j+1}$ , the intersection position  $(x_i, y_i)$  will be given by:

$$y_i = y_s$$
$$x_i = x_{j+1} + (y_s - y_{j+1}) \frac{(x_j - x_{j+1})}{(y_i - y_{j+1})}$$

For an inelastic collision, the new position  $(x_{j+1}, y_{j+1})$  is given by:

$$y'_{j+1} = y_i + r_n(y_i - y_{j+1})$$
  
 $x'_{j+1} = x_i + r_t(x_i - x_{j+1})$ 

where  $r_s$  is the coefficient of normal reflection, and  $r_t$  is the coefficient of tangent reflection.

The advantage of such a scheme is that the collision detection calculation is simple to compute. It is equivalent to intersecting a ray with a surface. Note that the collision calculation is done in the static coordinate frame of the constraint, so that  $P_j$  and  $P_{j+1}$  take account of the motion of the constraint relative to the world coordinate system between successive time steps. Another advantage is that the points are guaranteed not to penetrate the constraint. Impulse based constraints are most easily implemented when used in combination with Euler integration.

$$v_{j+1} = v_j + a_j \Delta t$$
$$x_{j+1} = x_j + v_j \Delta t + \frac{1}{2}a_j \Delta t^2$$

This is slower to converge than higher order methods such as the Runge-Kutta [14] or predictor-corrector methods [6], which assume that the forces vary smoothly as a function of time. These higher order methods have to be restarted when impulse based techniques are used. For such methods the constraints are usually expressed as a force of repulsion as the particle approaches the surface. The more localised the force, the smaller the time-step needed for correct integration. As explained later, forces such as directional friction can occur very suddenly, so this paper was illustrated using Euler integration, although the physical principles would apply equally well to other techniques.

Unfortunately, applying constraints to the point masses is not a general solution to collision detection. Sharp discontinuities in the constraints, such as at a cliff edge, should be tested against the edges of the cubes as well as the vertices. So there should really be both particle-surface and edge-surface collision detection. This will be especially important when creatures interact with each other and with themselves.

However, for this paper the actual locomotion of snakes and worms was of primary interest, so the constraints were kept simple: a horizontal floor and a rounded cliff edge.

### 4. Muscle Contractions and Directional Friction

In order to make the mass-spring systems move it is necessary to exert forces on the masses. This may be done globally using gravity and viscous drag [12] and directly, using force vectors applied to individual masses. This is equivalent to having a rubber puppet with strings and rods pulling and pushing it. However, to make the model look really alive it is necessary to generate the motive power internally. The massspring system must use frictional forces with constraints and changes of shape to achieve locomotion. Walking is one example of using friction to move the centre of mass. One foot stays in place on the ground as the other is moved through the air. In this way the frictional force with the ground is exploited whilst the geometry is reconfigured. Lifting feet is necessary because the frictional forces are isotropic i.e. they are independent of the direction in which the foot is slid. Sliding one foot forward causes the other foot to slide back equally far if the pressure on each foot is the same, hence the need to change pressure from one foot to the other. Snakes and worms, on the other hand, remain in contact with the ground at all times (sidewinding is an exception to this). So, in order to get along, they must have frictional forces which vary with the direction of sliding. This is achieved by the scales which cover the skin. Just as the microstructure of a surface affects its macroscopic optical properties, the small scale features of an object determine the frictional forces generated when it interacts with its environment.

If the body segment moves forwards, the scales slide relatively easily over the ground exerting little friction. When the body segment slides backwards, however, the scales dig in and the frictional force suddenly becomes very great. Worm locomotion is a simple application of this idea. Figure 2 illustrates a two mass, one spring worm.

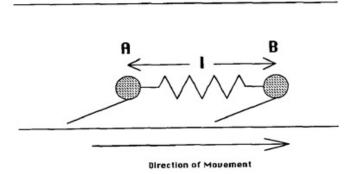


Figure 2. A two mass, one spring worm.

If the spring is expanded (l increased) scale B will slide over the ground and scale A will grip. If the spring is contracted (l decreased) scale B will dig in and scale A will slide. Both these motions lead to the system travelling in the same direction, so oscillating the spring length as a function of time will lead to forward motion.

A real worm, of course, consists of a number of such sections. If all of the spring lengths were varied in phase, only the front and back most scales would ever grip the floor. All of the intermediate scales would be constantly sliding. So, to prevent undue stresses at either end, a worm sends waves of compression from its head to its tail. The familiar worm-like motion results. In a real worm the travelling compression wave seems to be approximately a square-wave. Unfortunately, because of the coarse nature of the mass-spring approximation, squarewaves lead to peculiar distortions of the shape of the worm. However, a travelling sine-wave gives well-behaved and acceptably realistic results for the worm.

In the computer simulation, the directional friction was implemented as follows. The local forward spine unit vector  $\vec{s}$  was computed from the centre of the next and last segments.

$$\vec{s} = \frac{\vec{x}_{p+1} - \vec{x}_{p-1}}{|\vec{x}_{p+1} - \vec{x}_{p-1}|}$$

The velocity was then modified as follows :

if 
$$(3^{\circ}, v^{\circ} < 0.0) v^{\circ} = v^{\circ} - 3^{\circ}(3^{\circ}, v^{\circ})$$

This stops any backwards sliding of the worm. The frictional effect was only applied if the point mass was less than a certain distance from the floor so that is could be said to be in contact. Because the model is a dynamic one, it was possible to apply gravity as well. Image 1a shows a worm in free-fall colliding with the constraint, in this case a book. The nose slides forwards on impact. Image 1b shows the body mostly in contact, the tail exhibits a little "bounce". Image 1c shows the worm wriggling towards a vertical edge. The front begins to bend downwards under the weight. Image 1d shows the worm sliding off the edge. The tail flops over with the momentum from the sliding.

In a real worm, as the muscles contract they bulge out. This may be included in the model by making the circumferential spring lengths increase as a function of the axial compression. The model used for this animation was to keep the total volume of the worm constant.  $1/\sqrt{l}$  along the worm was used to scale the value of L for the springs around the circumference of the worm. This then helps to conserve volume.

The anatomy of a snake is very different from a worm, both in its internal structure and in the formation of its scales (squamation). Snakes have a skeleton with a flexible spine and ribs. The scales on a snake are diamond shaped on the upper part, but on the bottom they are a linear array of slanted blades. Snakes come in a wide variety of shapes and sizes [8] but move in only four basic ways, just as there are various "gaits" for four-legged figures [4].

The worm motion described above is called "rectilinear progression" and is achieved by the snake sliding its skin over its ribs [7]. The more familiar sinusoidal motion is called "horizontal undulatory progression".

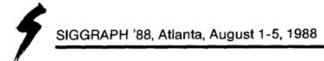
To model this, compression waves are again sent down the mass-spring system, but the springs on the left hand side of the snake are 180 degrees out of phase with those on the right hand side. This has the effect of bending the snake into the familiar waves. The directional friction with the floor leads to the snake being propelled along. When a snake has a good grip on the ground and has highly curved coils and is moving slowly, the body segments all follow along the same curve. It is as if the cross-sections of the snake follow along the same spline curve at constant distances. Indeed this technique has been used to model and animate snakes [2] and [10]. However, if the snake only has a poor grip on the surface, or if it is acted on by external forces, or if it is changing the amplitude of its coiling, then the segments will slide sideways as well.

Image 2a shows the mass spring system "at rest" with no muscle tensions. Image 2b shows the effects of undulations deforming the snake. The head-end has stayed virtually fixed while the tail is dragged forwards as the coils form. Image 2c shows the snake with the coils fully formed. Little forward motion of the head-end has been achieved. Image 2d shows the snake at a later stage. The segments are now all very nearly following each other along the same path.

The dynamic snake model is interesting in the way that it deviates from the spline following paradigm. The tail, because it is not laterally constrained, swishes slightly from side to side. (It looks like a happy snake). In addition, the mass spring system has secondary reactions to the spring contractions and the friction. This leads to a slightly irregular motion of the snake which looks both convincing and realistic. Finally, of course, the snake can fall under gravity and collide with constraints in the same way that the worm did.

A third form of snake locomotion is called "sidewinding". When grip on the ground is poor such as in open areas of sand, some snakes adopt a method of reducing their contact area with the sand so that the effective pressure increases. This also prevents undue heat exchange with the hot sand. This motion may be achieved using a vertical sinusoidal flexing of the snake which is 90 degrees out of phase with respect to the horizontal undulations. The vertical flexing has the effect of lifting all but a small portion of the snake off the ground. The wavelength used for the simulation in this paper was the length of the snake divided by 1.4.

Image 3a shows the sidewinder as the coils are forming. The back arches and the contact with the ground is localised. Image 3b shows the snake reaching its fully coiled state. Image 3c shows the snake as progression begins to take effect properly and Image 3d shows the snake at a later time. The



diagonal nature of the locomotion is apparent.

The fourth form of snake locomotion is called "concertina progression" and involves successive flexing and straightening of the snake. It is only very rarely used and has not been simulated by the author.

Whilst most snakes are not adept at flexing their backs vertically in a sinusoidal fashion through large amplitudes, creatures such as caterpillars are. Since the underside of the caterpillar is lifted and thrust forward and then placed down again, there is no need for directional friction. Instead it is better to eliminate all tangential movement for any segments in contact with the surface. This gives the caterpillar the ability to scale steep gradients both up the slope and down. Image 4a shows the caterpillar at rest. The fanciful creature shown, called the Arctic caterpillar, is covered in fur to highlight the changes in orientation of the surface. Image 4b shows the caterpillar arching its back.

## 5. Timings for the Dynamics

The dynamics calculations for the snake animations took 30 seconds per frame on a Silicon Graphics 4D/70 workstation. In part this was due to the implementation in a convenient but interpreted procedural modelling language (Alias SDL) and in part it was due to the comparatively unstable nature of Euler integration. 25 subintervals per frame were used to ensure accuracy in the animation of the snakes. For the less rigid worms this was reduced to 10 subintervals.

The different creatures were animated using different values for the spring constants and the damping. The worm had a k spring strength of 1.5 and a damping D value of 0.9. The worm tends to bounce on contact with the floor and as it dangles over the edge of the book it takes on interesting chain-like oscillations. The snake, on the other hand, used a k of 0.5 and a D of 3.5. The difference is that the snake swishes its tail as it moves. Too little damping leads to standing waves in its body which interfere with locomotion and mean that the snake is in danger of shaking itself to pieces. The caterpillar had a k of 0.3 and a D of 0.6.

# 6. Rendering Worms, Snakes and Caterpillars

In order to render the snakes and worms, the lattice of masses was used to create control points for cardinal bicubic parametric patches. They were generated so that the crosssections of the snakes and worms just touched the edges of the lattice. This prevented any possible intersections with the ground plane.

The worms for this paper were rendered using a colour map and a bump map [1]. The colour map gave the pink underside and the grey top. The bump map gave the wrinkles in the skin. The snake in Image 5 was rendered using a bump map and a colour map for the scales. The depth value for the scales was also used to blend between the texture mapped "markings" and the underlying skin colour. The hairs on the caterpillar were modelled as separate pieces of geometry. The roots of the hairs were generated as a dithered lattice in parameter space for each patch. The other end of the hairs were computed using linear combinations of the tangent vectors and the surface normal at the root to offset from the root position. This technique is discussed in more detail in [9]. The individual hairs in Images 4a to 4d also had collision detection with the floor. In the event of penetration of the ground plane, the hairs were individually rotated upwards about their roots until the ends were just in contact with the ground. This approach is trivial for an extended planar surface but would be more difficult for arbitrary obstructions. The hairs for these images were composed of three forward-facing triangles arranged in the form of a cone. The results were raycast with supersampling everywhere to avoid aliasing. Each hair was colour mapped along it length. The tips of each hair were black whilst the roots were white. The caterpillar images at 500 lines resolution each took 0.5 hours to render on a Silicon Graphics 4D/70 workstation.

## 7. Areas for Future Work

Now that the basic principles of legless figure locomotion have been implemented it is necessary to improve the collision detection computations. This will allow several snakes or worms to interact with each other and will enable the snakes to coil up without penetrating themselves. The task of directing the snakes to specific action needs to be investigated, giving the creatures goals and path planning skills. On a different tack, the dynamic models for legless creatures may be combined with legs to extend the models to such creatures as crocodiles and Chinese dragons which use their tails for locomotion extensively.

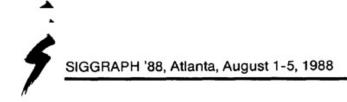
## 8. Acknowledgements

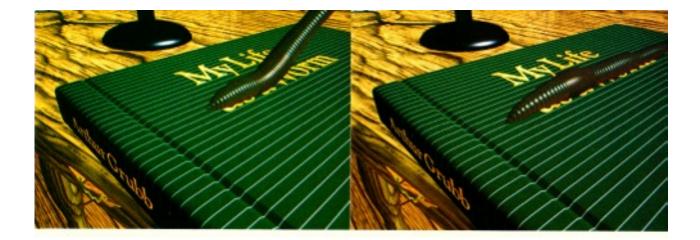
This research took place at Alias Research Inc. in Toronto as part of the Natural Phenomena Project. Thanks to Alias for encouraging this work. A big thank you to Patricia Anderson at Alias for modelling the snake head, to Freddi Gitelman for modelling the background to the worm sequence, and to Jim Craighead and Bob Leblanc for additional help with the snake rendering, and to Andrew Pearce, Steve Williams and Gary Mundell for help with the background to Image 6. A special thank you to David Ross for revealing to me the joys and complexities of procedural motion. Finally, thanks to Eileen O'Neill for introducing me to the beauty of snakes, and for leading the worm hunt without which the images in this paper would have been much less realistic.

#### 9. References

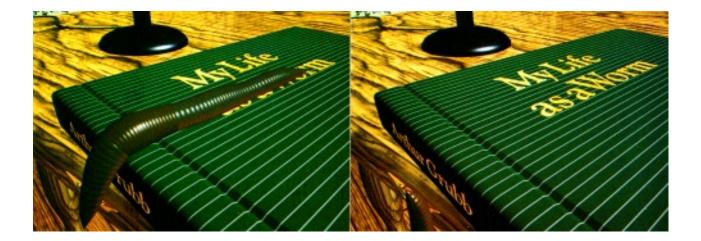
- Blinn, James F. "Simulation of Wrinkled Surfaces." Computer Graphics, Vol. 12, No. 3, August 1978.
- Coquillart, Sabine, "A Control-Point-Based Sweeping Technique", IEEE Computer Graphics and Applications, November 1987 pp 36-45.
- Culhane, John, "Special Effects in the Movies", Hilltown Press, Inc., ISBN 0-345-34536-3, November 1981.
- Girard, Michael and Anthony A. Maciejewski, "Computational Modelling for the Computer Animation of Legged Figures", Computer Graphics, Vol. 19, No.3, July 1985 pp 263-270.
- Isaacs, Paul M. and Michael F. Cohen, "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics", Computer Graphics, Vol. 21, No. 4, July 1987 pp 215-224.

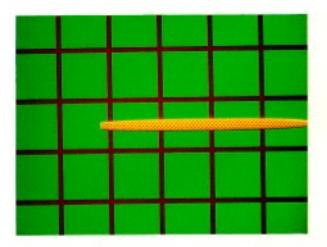
- Johnson, Lee W., R. Dean Riess, "Numerical Analysis", Addison-Welsey Publishing Company Inc. 1982. ISBN 0-201-10392-3.
- Klauber, Lawrence M., "Rattlesnakes", University of California Press, 1982. ISBN 0-520-04039-2.
- Mehrtens, John M., "Living Snakes of the World", Sterling Publishing Co. Inc., New York. Blandford Press Dorset, England. 1987. ISBN 0-8069-6460-X.
- Miller, Gavin S. P., "Computer Display and Manufacture of 3-D Models", Ph.D. thesis, June 1987, Cambridge University Engineering Department, Cambridge, England.
- 10. O'Neill, Eileen, personal communication Nov. 1987.
- Reynolds, Craig W., "Flocks, Herds and Schools : A Distributed Behavioral Model", Computer Graphics, Vol. 21, No. 4, July 1987 pp 25-34.
- Terzopoulos, D., J. Platt, A. Barr, K. Fleischer, "Elastically Deformable Models", Computer Graphics, Vol. 21, No. 4, July 1987 pp 205-214.
- Waters, Keith, "A Muscle Model for Animating Three-Dimensional Facial Expression", Computer Graphics, Vol. 21, No. 4, July 1987 pp 17-24.
- Willhelms, Jane and Matthew Moore, "Dynamics for Everyone", Appendix 1, SIGGRAPH '87 Course 10, Computer Animation : 3-D Motion Specification and Control pp 145-146.

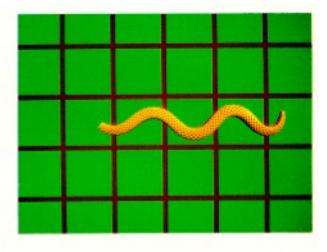




Images 1a to 1d (top left to bottom right): Eric the Dynamic Worm

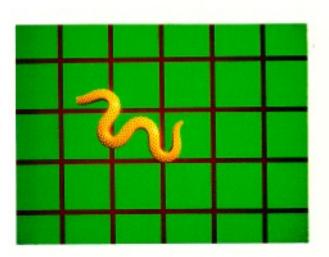


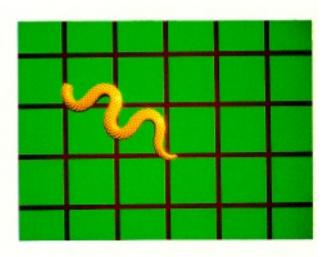




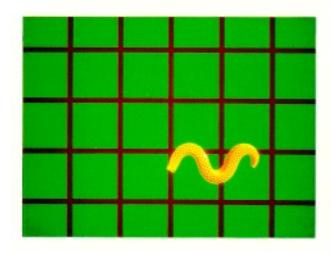
Images 2a to 2d (top left to bottom right): Horizontal Undulatory Progression

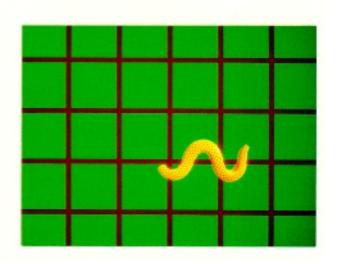
ecm





Images 3a to 3d (top left to bottom right): A Sidewinder







acm

# Images 4a and 4b: An Artic Caterpillar

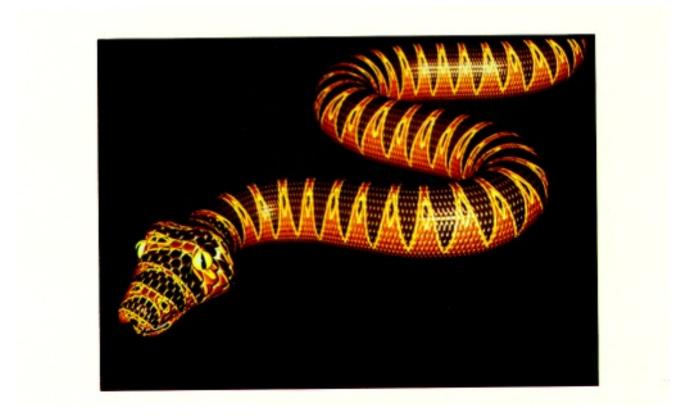


Image 5: A Missisauga Rattler.



Image 6: Curling Up by the Fire