Expressive Character Animation with Energy Constraints

Jae-Kwon Park, Young-Min Kang, Sung-Soo Kim, Hwan-Gue Cho

Graphics Application Lab.,
Department of Computer Science,
Pusan National University, Kum Jung Ku, Pusan 609-735, Korea.
E-mail: {jkpark,ymkang,sskim,hgcho}@pearl.cs.pusan.ac.kr

ABSTRACT

The main problem in computer animation is how to control the act of a creature in a desired way efficiently. There are two approaches in this problem: one is to apply laws of physics as constraints, and the other is to synthesize a controller. Recently a very appealing approach, named Spacetime Constraints(SC) paradigm was introduced, whereby the animator specifies what animated figure should, not how to do. Though SC paradigm works well for generating a natural-looking act of a creature, it needs huge amount of computations in optimization phase. In addition, it can not express the mood of character, e.g., briskness, tiredness, since SC only focuses on minimizing the total energy consumed in animating procedure.

In this paper, we propose a new modeling technique for the gait of an artificial creature with emotions. The basic idea of our approach is that we limit the energy consumption to control the creature in a desired way. Thus if we want to make a creature move energetically (e.g., walking briskly, and running or hopping), we only need to supply enough energy to activate its all articulations fully. Similarly if we want to make it walk in a tired state, we force it walk with a very small amount of energy. In this way, we can easily animate limping or staggering locomotion by controlling the utilization ratio of energy consumption in each leg independently.

Our animation algorithm was implemented and tested in SGI *Indigo*^{2} machine with OpenGL/Inventor environment. Several snapshot pictures taken in the real-time animation procedure are given in the final section.

Keywords: character animation, energy constraints, genetic programming, emotion.

1 MOTIVATION

Computers have been used to assist people with the creation of animated sequences for more than a decade. One of multiple aspects of computer animation is the objective to simulate reality taking into account in biological, mechanical or physical constraints, yet the resulting animation still can not reproduce the lifelike qualities imparted to creatures through traditional hand animation[10]. There have been several methods to create realistic motion of inanimate objects such as chain and some articulated machine by simulating physical laws. However the lifelike motion of animated creature can not be captured by a direct application of Newton's laws of motion since we have to know how to make it move or control muscles in a desired way[5, 8].

In this paper we consider the problem of how to model the emotion-based behaviors of an artificial creature for the purpose of advertising and entertainment use, including such as cartoon-like exaggerations or expression. There have been several studies for animation of emotion-based behavior. One common approach to cope with this problem is to build huge database of human behavior[9]. Other dynamic simulation techniques can be used to generate a desirable motion by trial and error off-line experiments[3, 8].

Recently one approach is proposed. In their method, characteristics of human behavior are extracted in the method simply from the empirical data of actual human movements without any physical-based simulations[10]. The authors compute the Fourier series on the expansion of the original measure data to make a prototype of the functional model for describing the emotional aspect of human locomotion through the Fourier analysis and synthesis(interpolation)[5]. Though this Fourier series functional model approach enables us to make realistic animation, it has some drawbacks. At first, this method requires lots of empirical data, namely motion data, by experiments. So if there is no motion data for human swimming, it can not make any animation for swimming. Another disadvantages is that it can not animate the behaviors of artificial creatures or non-human being, for example insect-like robot, or animals[3, 6].

In this paper, we propose a new animating method for emotion-based creature by controlling the amount of the energy to be consumed within the creature. We can make it walk slowly and tired by only reducing its energy supplying or run or jump briskly by supplying energy fully. In this procedure, we use genetic programming approach to find the optimal behavior within a given energy amount.

2 GENERAL TECHNIQUES FOR CHARACTER ANIMATION

General animation techniques can be divided into two major categories, animation techniques using motion data and those using optimization. The former techniques have powerful control abilities and the later create animation with low cost and effective procedures.

2.1 Editing Motion Data

Animation techniques that utilize the motion data enable animator to control the motion of characters precisely. The representative technique of this category is the keyframe animation technique. This technique makes animators able to specify the motion of characters corresponding to their purpose. However, keyframing procedure is tedious and requires much time. Another type of using the motion data is generating an animation with the data extracted by motion sensor. This type of technique also has the defects that implementation of them needs much cost. Many techniques that are involved in this category have been proposed. Michiel von de Panne proposed Parameterized Gait Synthesis which extracts the common properties of various types of gait motions and synthesizes the properties to generate a specific animation[3]. The drawback of this method is that an animator should gather various motion data from which motions are synthesized.

2.2 Motion Optimization

Motion of character is too complicated to control all its motion by manual work. We can not control detailed motion of a character. Therefore we need an automated technique to avoid manual work. The automation of character animation is obtained by optimization of motion. Motion optimization technique is divided into three representative categories.

The first is technique using Dynamics[8, 10]. Since this technique is based on physical dynamics, it generates realistic motion. On the other hand, it requires too much time to calculated motion of a character.

The second is using neural network. In this approach described by van de Panne and Fiume[11],a weighted, nonlinear network with time delays is used to connect sensors indirectly to the actuators. This structure is referred to as a Sensor-Actuator Network(SAN).

The third is using Spacetime Constraints. This technique supposes that the motion satisfying the space-time constraints would be the best motion[1, 4]. Since it needs heavy computation to solve the optimization problems that are expressed as many nonlinear equations, the sequential quadratic programming method has been proposed for the approximation method for the optimization.

Genetic Programming methods are also intensively used as a new paradigm for solving the optimization problems[2, 6, 12]. Though these techniques have advantage in automating the producing procedure, they have so poor control ability that it is difficult for the animators to control and even predict the motion of characters.

2.3 Motion Control with Energy Supply

As shown in previous sections, so far almost every animation technique stresses on only one of two main goals, the automation and control. Those techniques cannot avoid the limitation of control or automation. New control technique we present in this paper creates animation frame sequences automatically and controls the motion effectively with energy constraints or secondary control methods. To achieve the goal, creating animation automatically and controlling it with effective control methods, we exploit genetic programming technique.

Fig.1 shows the overview of our animation system implementation. The main control is achieved by the energy constraints and secondary constraints control the motion in detail. Genetic programming component selects the best gene sequence to generate natural motion.

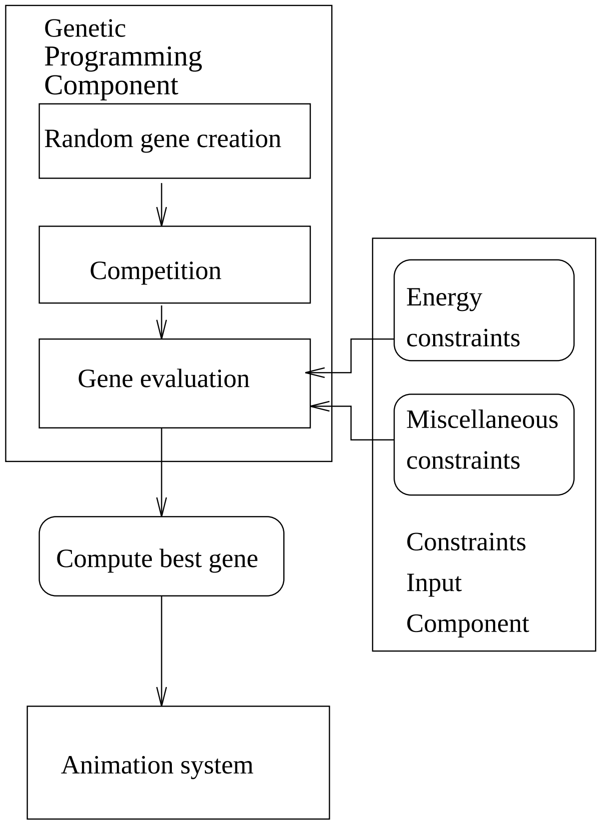

Figure 1: Animation System Overview

3 THE CHARACTER AND CONTROL METHODS

3.1 The Structure of Virtual Creature

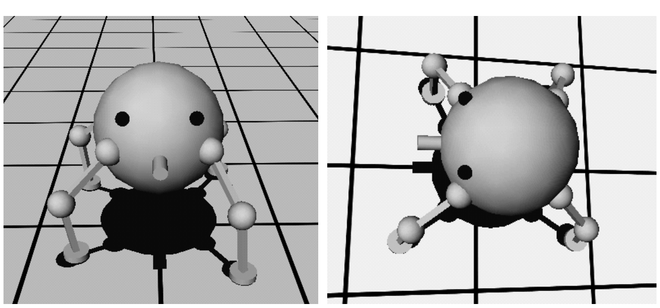

Figure 2: A Virtual Creature

We prepared a simple creature to simulate our technique. The physical appearance of the virtual creature is shown in Fig.2. This virtual creature has a spherical body and four legs that have two articulated joints. Each leg has three degree of freedom(DOF). The rotational angle of each joint is not specified explicitly but determined by inverse kinematics according to the location of feet and body.

3.2 Control Methods

The behavior of a virtual creature is determined by control program that calculates the location of feet and body at a specific time. When the energy constraints are given to the creature, our system generates virtual creatures with randomized genes and lets the creatures walk. Then we evaluate the speed per a given energy for every creature. After the evaluation, the system produces a new generation by the methods of reproduction, crossover, and mutation. A set of good genes are selected by the evaluation system, and those genes are used for next generation. We designed foot control program and the body control program that compute the physical location of feet and body, respectively.

3.2.1 Foot Control

The feet of a virtual creature proceed according to time span. Suppose that xz plane is the surface and z direction is the proceeding direction, y is the height. When the virtual creature walks to the positive direction of z axis, the feet are lifted up to a specific height and get down on the surface. For the representation of such behavior, we use $sin(x)$ function for y-coordinate of the i-th foot, and quadratic function is assigned for z-coordinate of the i-th foot as following,

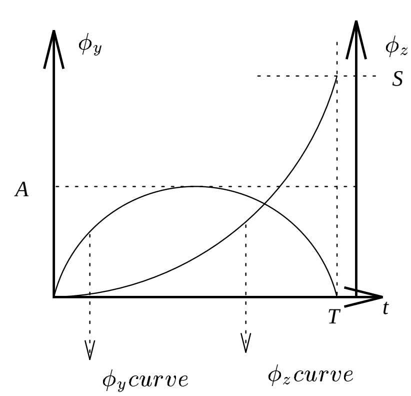

Figure 3: Foot Control Curve

$$
\begin{aligned}
\phi_{iy}(t) &= A \cdot sin((t-\sigma_i)\cdot \pi/T) \\
\phi_{iz}(t) &= \phi_{iz}(0) + (t-\sigma_i)^2 \cdot \frac{S}{T^2}
\end{aligned}
\tag{1}
$$

where $\phi_{iy}(t)$ denotes y-coordinate of the i-th foot at time t, A is the amplitude of a pace, T is the total time required for one pace, S is the length of a pace, and σ_i is the delay of starting time of the i-th leg. A, T, S, and σ_i are updated automatically to fit the given constraints

3.2.2 Body Control

The control program that determines the location of body is composed of two parts. One is responsible for calculating the location on xz plane and the other determines the height of creature, y-coordinate. A simple method to determine the location of body on xz plane is to place body on the centroid of four feet, but it is not realistic. When a foot is lifted, the body must be moved to the center of other feet on the ground. We use the following function to determine xz location of body C_{xz}.

$$C_{xz} = \frac{\sum_{i=1}^{\eta}(\phi_i + \alpha \cdot (\varpi - \phi_i) \cdot (1 - \frac{1}{\phi_{iy}+1}))}{\eta} \quad (2)$$

where η is the number of legs, and ϖ is the centroid of four feet in xz plane. The variable α affects the distance between body and the centroid of the feet on the ground when a foot is lifted. Now we can determine the height of body by a function using the projected coordinate on xz plane of joints (J_{xz}) and feet (ϕ_{xz}). Following function controls the height of body, C_y.

$$C_y = \frac{\sum_{i=1}^{\eta} \beta \cdot H(J_{i_{xz}}, \phi_{i_{xz}})}{\eta} \quad (3)$$

where the function $H(J_{i_{xz}}, \phi_{i_{xz}})$ goes up to maximum when $\mid J_{i_{xz}} - \phi_{i_{xz}} \mid$ is zero, and this value decreases, when $\mid J_{i_{xz}} - \phi_{i_{xz}} \mid$ increases. The variable β is determined by genetic programming and it controls the height of body.

3.3 Genetic Programming

The functions in the control program generate a specific pose of a character according to the current time t. Since computing the optimal values of A, S, T, α, β, and the activation time of each leg requires very heavy computations, those variables are automatically generated through genetic programming.

3.3.1 Gene Evaluation

The virtual creatures are generated with randomized gene sequence and competed with others during the given period. After the competition, gene sequences of all the creatures are evaluated. For the experiment of this paper, we adopted the evaluation criteria as the utilization ratio of energy, that is, the speed per a given energy. The total evaluation value of gene g is denoted as ψ_g, and two criteria are used for the evaluation. One is energy fitness ratio of the gene E_g, which is the difference between used energy E_u and the given energy of constraints E_c, and the other is the speed evaluation S_g which is defined by a function of O_g which is the arrival order of a creature with gene g.

$$\begin{aligned} \psi_g &= E_g + S_g \\ E_g &= \mid E_c - E_u \mid \\ S_g &= (E_c \cdot O_g)/N \end{aligned} \quad (4)$$

where N is the number of virtual creatures for competition. The smaller the ψ of the gene is, the more easily the gene is selected for the set of gene sequences of the next generation.

4 CONTROL TECHNIQUES

4.1 Energy Constraints

Since our method is controlled mainly by the energy comsumption, the defintion of the energy is needed. The energy is calculated with several parts such as the torque of legs, the change of potential energy of lifted legs and body, and force to accelerate the velocity of body.

4.1.1 The elements of Energy

- **Torque of Legs(τ)**
  The first element of energy is the torque of legs placed on the ground. Each leg on the ground consumes different amount of energy according to the joint angle. The smaller the angle at which a leg meets the surface is, the more forces are needed for the leg to support body. See Fig.4. A leg consists of two rigid

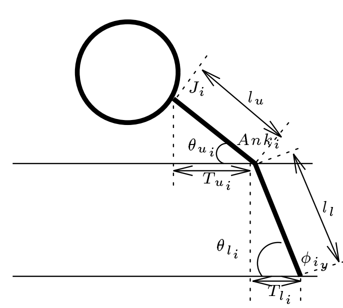

Figure 4: Torque of a leg

  parts, l_u and l_l. Total torque of a leg is the summation of the upper torque T_u and the lower torque T_l. We assume that the force in the i-th leg, F_i, is distributed evenly to both of solid parts. Thus, the torque of the i-th leg τ_i is obtained as

$$\begin{aligned} \tau_i &= T_{u_i} + T_{l_i} \\ T_{u_i} &= F_i \cdot cos\theta_{u_i} \cdot l_{u_i} \\ T_{l_i} &= F_i \cdot cos\theta_{l_i} \cdot l_{l_i} \end{aligned} \tag{5}$$

where T_{u_i} and T_{l_i} are the i-th upper torque and the i-th lower torque, respectively. If the i-th foot is lifted ($\phi_{i_y} > 0$), τ_i is 0. Otherwise

$$\tau_i = F_i \cdot (cos\theta_{u_i} \cdot l_{u_i} + cos\theta_{l_i} \cdot l_{l_i}) \tag{6}$$

The weight of body, $m_c \cdot g$, is distributed to each force F_i. F_i increases as the coordinate of the i-th foot projected on xz plan ($\phi_{(x,z)_i}$) gets closer to the coordinate of body ($C_{(x,z)}$). The sum of these forces should be the same of the weight of body. g is the gravitational acceleration and m_c is the mass of body.

$$m_c \cdot g = F_\alpha + F_\beta + F_\gamma$$

where $F_\alpha, F_\beta, F_\gamma$ are in proportion of the distance between body projected on xz plane, (C_{xz}) and each foot that is placed on the ground, respectively. For example, F_α is the force on the leg whose distance to C_{xz} is α. When one foot is lifted ($\phi_{i_y} > 0$), and three feet are on the ground, torque of the lifted leg goes to 0 and the torque of the i-th leg that is not lifted is calculated by the following.

$$F_i = \left(\frac{1}{2} - \frac{D(\phi_{i_{(x,z)}}, C_{(x,z)})}{2 \cdot \sum_{j=1}^{\eta} D(\phi_{j_{(x,z)}}, C_{(x,z)})}\right) \cdot m_f \cdot g \tag{7}$$

Function $D(\phi_i, C)$ denotes the distance between ϕ_i and C. Thus, if the i-th foot is lifted ($\phi_{i_y} > 0$), the value of $D(\phi_i, C)$ goes to zero.

- **Potential Energy in a Lifted Leg (ΔU_f)**
  This means the energy consumed when the creature lifts its legs. This potential energy is proportional to the difference of height of the i-th foot in a unit time.

$$\Delta U_f = m_f \cdot g \cdot (\Delta\phi_y - \Delta C_y) \tag{8}$$

  ΔC_y is subtracted from $\Delta\phi_y$ because the height change caused by the movement of body should not affect the leg energy consumption. m_f is the mass of the lifted leg. If $\Delta\phi_y$ is negative, the energy is ignored. Because we assume that the creature does not consume energy while a leg moves down. As each leg moves, the locomotion of the leg gets easier or harder depending on the rotation inertia. If the i-th foot is close to the pivot of rotation, the movement requires less force. The i-th pivot of rotation is the coordinate of the i-th joint. The rotation inertia, R_i, is computed as

$$R_i = \mid R_c - \phi_i \mid$$

  where R_c is the pivot of rotation.

- **Potential Energy in Body (ΔU_C)**
  While the creature is lifting body, it consumes some energy in proportion to the difference of body's current height and the previous height. Consumed potential energy in the body, ΔU_c, is

$$\Delta U_c = m_c \cdot g \cdot \Delta H_c \tag{9}$$

  where ΔH_c is the height difference of body in a unit time. If ΔU_c is negative, the value is ignored.

- **Force to Accelerate Body (F_a)**
  The force to accelerate body is calculated with the mass and the acceleration of the body.

$$\begin{aligned} F_a &= m_c \cdot \Delta v_t \\ \Delta v_t &= v_t - v_{t-1} \\ &= C_t - C_{t-1} - (C_{t-1} - C_{t-2}) \end{aligned} \tag{10}$$

  where Δv_t is v_t - v_{t-1} and used as acceleration of the body, and C_t is position of the body at time t.

E_G, the total energy of the creature G. is the summation of all torques T_G and potential energy M_G.

$$\begin{aligned} E_G &= T_G + M_G \\ T_G &= \sum_{i=1}^{\eta} \tau_i \\ M_G &= \sum_{i=1}^{\eta} \Delta U_{f_i} \cdot R_i + \Delta U_c + F_a \end{aligned} \tag{11}$$

The total consumed energy during the whole frame is $\sum_{t=t_s}^{t_e} E_{G_t}$, where t_s is the starting time, t_e the ending time.

4.2 Emotion-based Gait Generation

We use energy constraints as the main control method. However, other secondary control methods are required. When an animator wants to specify the pose of a creature in detail, he can make the constraints as an element of fitness evaluation. For these pose control, we used the height of creature and width of pace. Fig.8 is a snapshot of animation with height constraints that limit the minimum height of body. This kind of constraints can be easily adopted to the genetic programming phase of our new technique. Fig.10 shows gait with a limping leg. In order to make one leg to be handicapped, we modified the control program. The location of body must be moved to the center of three normal legs. Other legs must get down on the ground quickly when a handicapped leg is on the ground and others are lifted.

5 EXPERIMENTS

We experimented our technique with C++ using Open Inventor on SGI Indigo[2]. We prepared 1000 individual creatures for competition with randomized gene sequence and forced them to evolve to meet the given constraints. Table 1 and Fig.5 show the result of genetic programming when supplied energy is fixed as 300. The vertical axis represents the fitness of a creature. For a creature v with gene g, we define the fitness evaluation function $F(v)$ as following,

$$F(v) \quad = \quad \mid E_c - E_u \mid + \frac{E_c \cdot O_g}{N} \qquad (12)$$

where E_c is the energy constraints, E_u is the energy used by the creature v, N is the total number of creatures, and O_g is the order of arrival of the creature with gene g. The fitness value of a creature is minimized when the creature uses exactly the same energy as the given energy and arrives target sooner than any other creatures.

Table 1 and Fig.5 show that the fitness goes to zero as a creature is evolved.

Generation	Energy E_u	Arrival O_g	Fitness $F(v)$
2	302.35	85	27.85
4	299.87	20	6.12
6	299.87	3	1.02
8	300.98	1	1.28
10	300.94	3	1.84
12	300.97	1	1.27
14	300.98	2	1.58
16	300.96	2	1.56
18	300.97	2	1.57
20	300.96	0	0.96

Table 1. Gene Evaluation Table (Energy:300)

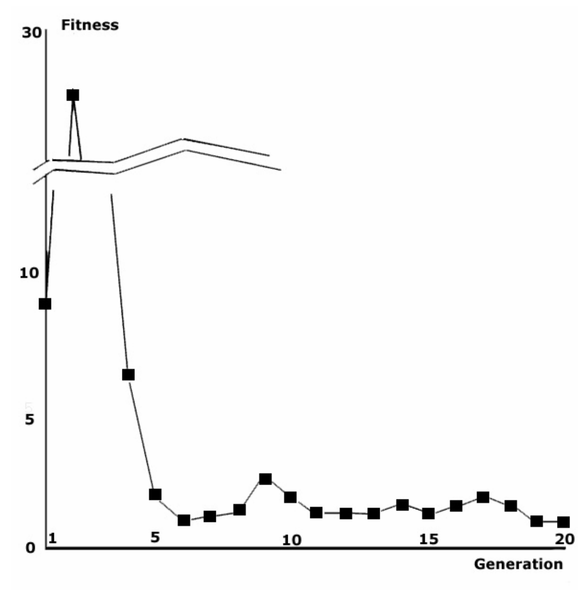

Figure 5: Fitness Curve

Fig.6 and Fig.7 show the result of energy constraints 30.0 and 400.0 respectively. Under the low energy condition, a virtual creature walks in slow pace and tardy pose to reduce the amount of energy consumption. However, it walks fast and looks very vigorous under the high energy condition. Table 2 shows the best gene under energy level 30 and Table 3 shows those under the energy level 400.

Generation	Energy	Arrival	Fitness
0	49.79	1	19.82
2	33.01	70	5.11
4	32.73	13	3.12
6	33.32	5	3.47
8	32.04	4	2.16
10	31.03	11	1.36
12	31.37	3	1.46
14	31.37	1	1.40
16	31.36	1	1.39
18	30.82	6	1.00
20	30.56	5	0.71
...	...	...	...
38	30.24	2	0.30

Genes

A	S	T	α	β
0.239	2.545	329	0.520	0.669
l_1	l_2	l_3	l_4	E
1.23	0.00	0.82	0.41	30

Table 2. Gene Evaluation Table (Energy:30)

As shown in Fig.6, if an animator gives little energy to a virtual creature, it does not lift

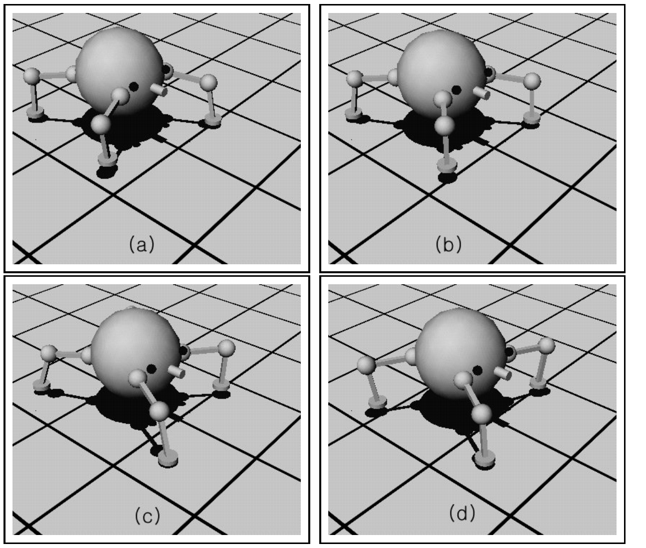

Figure 6: "Tired" Walking with Energy = 30.0

its feet high while the virtual creature lifts its feet higher and stretch to reach further under the high energy condition. The gene under the low energy shows small values for the amplitude (0.239) and long cycle (329) of a pace while the genes for high energy condition involve larger values for the amplitude (0.712) and shorter cycle time (114). l_i denotes the relative activation time of the i-th leg. The actual activation time of the i-th leg is $l_i \cdot T$.

Generation	Energy	Arrival	Fitness
0	407.79	6	10.19
2	407.79	7	10.59
4	407.79	23	16.99
6	394.26	2	6.53
8	394.26	5	7.73
10	394.26	3	6.93
12	394.26	1	6.13
14	396.79	3	4.40
16	396.79	0	3.20
18	396.79	1	3.60
20	396.79	2	4.00
$\cdots$	$\cdots$	$\cdots$	$\cdots$
33	396.79	0	3.20

Genes

A	S	T	α	β
0.712	1.443	114	1.284	0.936
l_1	l_2	l_3	l_4	E
2.58	1.72	0.00	0.86	400

Table 3. Gene Evaluation Table (Energy:400)

5.1 Secondary Motion Control

Fig.8 shows the effects of height control. Fig.8(left) is the snapshot of a creature where the minimum height is limited as 1.5, and Fig.8(right) is that of the same creature where the maximum height is limited as 1.2. Both creatures are evolved under the same energy constraints (Energy:100).

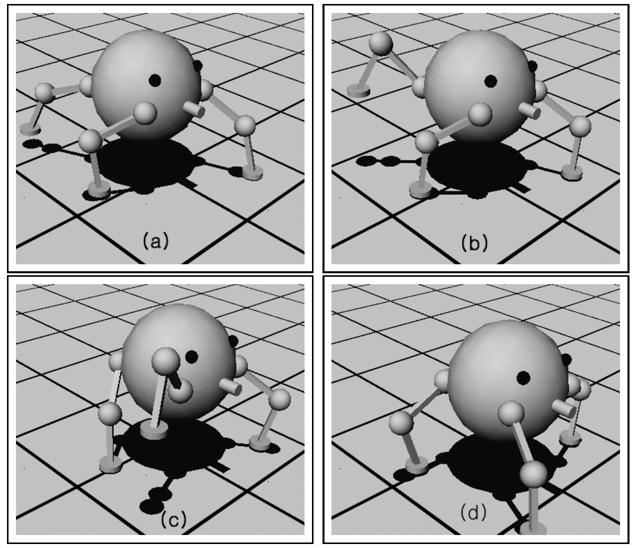

Figure 7: "Brisk" Walking with Energy = 400.0

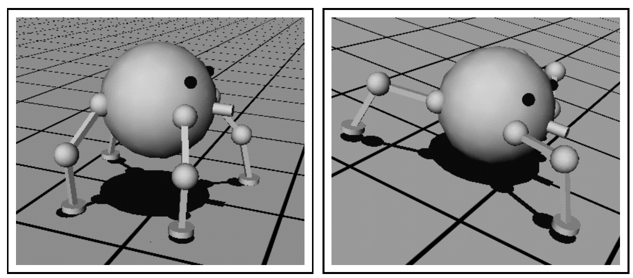

Figure 8: Height Control

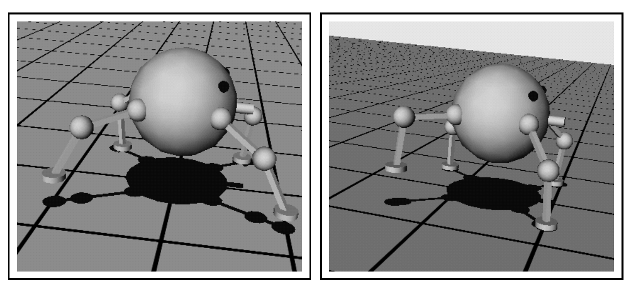

Figure 9: Pace Control

Fig.9 is a snapshot of pace control. Fig.9(left) shows the movement of a creature of which the length of a pace is forced to be longer than 1.5 and the other is limited to be shorter than 0.5. Fig.10 shows gait with a limping leg. We limited the total force on the right-rear leg to simulate handicapped leg. The gene for gait with a limping leg is shown in Table 4.

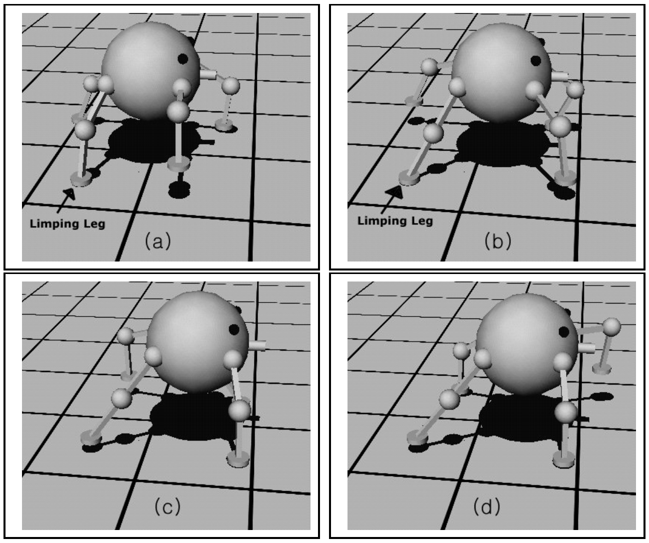

Figure 10: Gait with a Limping Leg

A	S	T	α	β
0.483	1.60	66	1.000	1.019
l_1	l_2	l_3	l_4	E
1.53	0.52	2.51	0.00	100

Table 4. Gene Table (Limping Leg)

We cannot simulate the gait with a limping leg only with gene of Table 4 because the control program which determines the location of body (Equation 2) does not consider that a leg is handicapped. The factor $(1-(1/\phi_{i_y}+1))$ in the equation 2 affects the significance of the leg i in calculating the location of body. When the i-th leg is handicapped, this factor is not calculated by $(1-(1/\phi_{i_y}+1))$ but just assigned 1.

6 CONCLUSION AND FUTURE WORKS

We propose a new animation technique to generate emotion-based behavior of an artificial creature. The main idea of this technique is to limit the amount of energy to be supplied to the creature. So if we allow a small amount of energy, the creature have to walk in a "tired" form. However the optimal motion control with a given amount of energy needs a hard optimization procedure, so we adopted the genetic programming approach to find the optimal motion. By this approach we do not consider how to control the detailed mechanical movement of each joint and leg to produce "tired" or "energetic" motions. Also Our technique does not need real motion data obtained by rotoscopy.

The major features of our technique are as followings.

- In order to animate emotion, we do not need to capture motion data, which was the main strategy of the previous works.
- We control the amount of energy and its consumption rate to animate its emotion-based behaviors.
- Our technique works fast due to the genetic programming approach to find a nearly optimal motion within a limited energy provided by animator. Once some optimal genes were obtained for some interesting behavior in off-line simulation, we can animate the creature in real-time by combining these gene in our experiment.

In the future we will try to apply this technique to a creature with more articulations such as a human or animal. And we will develop the other control techniques to make a more natural and realistic emotions such as controlling mouth, eye, neck and waist. In this paper, we did not consider the situation when the virtual creatures have to clear obstacles such as wall and stairs. We are working to make a new gene structure for these complicated job, e.g., "go to there avoiding that obstacle" or "go to there within t_0 time and an energy E_0".

REFERENCES

[1] A. Witkin, M Kass, *Spacetime Constraints*, Proceedings of SIGGRAPH '88, vol. 22(4), pp.159-168, 1988.

[2] L. Gritz, J. K. Hahn, *Genetic Programming for Articulated Figure Motion*, The Journal of Visualization and Computer Animation, vol. 6. pp. 129-142, 1988.

[3] M. van de Panne. *Parameterized Gait Synthesis*, IEEE Computer Graphics and Applications, vol. 16(2), pp. 40-49, 1996.

[4] C. Rose, B. Guenter, B. Bodenheimer and M. F. Cohen, *Efficient Generation of Motion Transitions using Spacetime Constraints*, Proceedings of SIGGRAPH '96, pp. 147-154, 1996.

[5] M. Unuma, K. Anjyo and R. Takeuchi, *Fourier Principles for Emotion-based Human Figure Animation*, Proceedings of SIGGRAPH '95, pp. 91-96, 1995.

[6] K. Sims, *Evolving Virtual Creatures*, Proceedings of SIGGRAPH '94, pp. 15-22, 1994.

[7] Z. Liu, S. J. Gortler and M. F. Cohen, *Hierarchical Spacetime Control*, Proceedings of SIGGRAPH '94, pp. 35-42, 1994.

[8] A. Bruderlin and T. W. Calvert, *Goal-Directed, Dynamic Animation of Human Walking* Proceedings of SIGGRAPH '89, pp. 233-242, 1989.

[9] A. Witkin and Z. Popovie, *Motion Warping*, Proceedings of SIGGRAPH '95, pp. 105-108, 1995.

[10] J. K. Hahn, *Realistic Animation of Rigid Bodies*, Proceedings of SIGGRAPH '88, pp. 207-216, 1988.

[11] M. van de Panne and E. Fiume, *Sensor-Actuator Networks*, Proceedings of SIGGRAPH '93, pp. 335-342, 1993.

[12] J. T. Ngo and J. Marks, *Spacetime Constraints Revisited*, Proceedings of SIGGRAPH '93, pp. 343-350, 1993.