Appear in Proceeding of 2003 Computer Graphics Workshop (CG2003), Taiwan, 2003.

Planning Efficient Walking Gaits in Real-Time for Human Characters

Pei-Feng Chen and Tsai-Yen Li
Computer Science Department
National Chengchi University,
Tel: +886-2-29393091 Ext. 62266
Email: {g9009, li}@cs.nccu.edu.tw

Abstract

Simulating humanoid motion has always been a challenging problem in computer animation. It has become an indispensable task in emerging virtual environment applications such as on-line games. Although there have been much research on this topic, most of them are only applicable to a specific terrain or only for a given set of footsteps. It is rare to have a system capable of automatically generating collision-free lower-body motion for a given terrain profile. In this paper, we have proposed such a system that can generate the motion plan for humanoid walking in real time. We use inverse kinematics to analyze the motion characteristics of human walking on various ground conditions in order to define the key frames of a walk cycle. We use Bezier curve to adjust the motion of a swing leg to avoid collisions with the ground or obstacles. In addition, the system is capable of generating legal and efficient footsteps for the next few steps according to an energy consumption model. According to observation and measured data, we designate appropriate timing for each phase between key frames to make the animation more realistic. Finally, we have succeeded in integrating such a system with a 3D browser and demonstrated that the planner can generate the locomotion motion for human characters in real time.

Keywords: humanoid animation, walking motion simulation, human walking gait generation, digital actors, motion planning, and computer animation

1. Introduction

In the rapidly growing virtual environment applications, generating avatar motions has always been a challenging and indispensable problem. Among the various locomotions that human characters can perform, walking is the most frequently used motion. Therefore, there has been much research on this topic in the past two decades in the field of computer animation and robotics. However, most of them usually assume a flat ground or a given set of feasible foot placements. However, these assumptions may not be valid for many applications where the environment consists of uneven terrain and when designating foot placements become too tedious.

An ideal humanoid motion simulation system should be capable of generating collision-free human walk motions adaptive to the ground condition of an uneven terrain. Additionally, the system should provide the mechanism of planning legal foot locations for next few footsteps in an efficient way like a human does. Furthermore, since the target application of this research is an interactive virtual environment, the system is also required to generate the collision-free motions in real time.

In this paper, we propose a planning system that can generate humanoid lower-body motions in real time according to the ground profile on an uneven terrain. The system can not only plan the footstep locations automatically, but also guarantee that there exists collision-free locomotion for these footsteps. In addition, among the feasible footstep locations, the system will choose the most efficient one according to an energy model based on experimental data reported in the literature.

The rest of the paper is organized as follows. We will first review related work in humanoid motion simulation in the next section. We will then give an overview of our system architecture in Section 3. In Section 4, we will describe how to plan the footsteps while in Section 5 we will explain how the locomotion is generated. The experimental results will be demonstrated in Section 6. Finally, we will conclude our work in the last section with some future extension suggestions.

2. Related Work

The research on humanoid walking motion simulation can be mainly divided into two categories: *kinematics-based*[2] and *dynamics-based*[1][5]. Nevertheless, there also exist many works that fall between these two extremes and use a hybrid solution to take advantages of these two approaches [4].

In a kinematics approach, the key frames of a humanoid walk cycle are usually determined by observing a real human walking. Although we can specify the joint angles of humanoid lower body for a given key frame, there is no guarantee that the motion will satisfy constraints such as not penetrating the ground. Therefore, these joint angles are usually computed by solving inverse kinematics equations for a given end-effector location in a key frame pose. The frames between the key frames can then be computed by appropriate interpolation between two key frames. Although the lower body motion can be computed efficiently in this way, a common drawback of such an approach is that the generated motions may not look realistic.

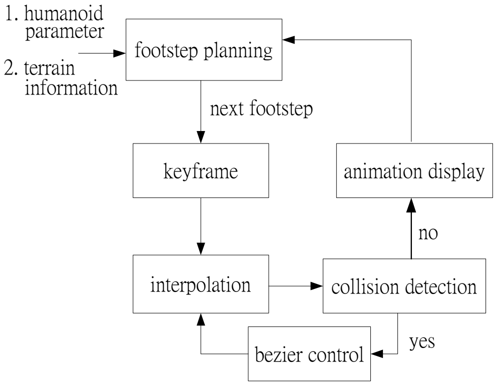


Figure 1. System flowchart

The other category of approaches uses dynamics methods to model the physical properties of a humanoid figure and simulate the motion by numerical methods. This type of methods usually generates more realistic motions but the computation is also more expensive. Among the dynamics methods, inverse dynamics methods are commonly used because the experimental data about the inserted forces on the end-effectors can be obtained more easily than the joint torques. However, due to the computational cost of such an approach, this type of methods usually cannot generate motions fast enough to be used in a real-time application.

3. System Overview

Figure 1 shows the data flow chart of the system proposed in this paper. The inputs to the system include the parameters of a humanoid model and the geometric description of the ground profile. For a humanoid model, we only consider the lower body in order to simplify the problem. This model includes parameters such as the lower leg and thigh lengths and must satisfy the requirement of LOA 1 (Level of Articulation) [10] in the standard virtual humanoid model. The other input is the geometric description of ground profile. This profile describes the locations where height changes along a given path.

After inputting the geometric parameters of the humanoid model and terrain, the system will use the footstep planner to generate the next feasible foot location. Then the system will use the current and the next footstep locations to compute the key frames as well as the frames between the keys by interpolation.

For each of the interpolated frames, we will check the possible collisions of the given pose with the environment. If a collision does exists, we will modify the Bezier curve for the swinging leg by adjusting the control points in order to avoid the collision. In addition, all generated motions must also satisfy the kinematics constraints imposed by the humanoid model. After generating the joint angles of a given cycle for one leg, the system will continue to plan for the other leg.

4. Footstep Planning

When human beings walk, they plan subconsciously. They usually only plan a few steps ahead and decide on feasible footsteps that move their body forward. If there exist multiple choices, the one that consumes the least energy is usually taken. We will first describe an energy model for human walking and then introduce the footstep planner taking into account energy consumption.

4.1. Energy model for walking gait

According to [7], in a natural setting, human walks in the most energy-saving way. Therefore, we will need to build an energy model for walking gait in order to take the energy factor into account when we plan the lower-body motion. However, there is no ready equation in the literature that can accounts for various walking styles on uneven ground of various conditions. Although dynamics simulation can provide us with an accurate answer, the computational cost does not allow us to do so in real time. Therefore, in this subsection we will describe an energy model that can quickly estimate the energy consumption for various ground conditions based on statistic data. The model considers three factors: *horizontal*, *vertical*, and *internal* as described below.

In [5], an energy consumption model on a flat ground has been proposed. The relation between energy and speed is listed as follows:

$$E_n = \frac{E_w}{n} = \frac{32 + 0.005 \times v^2}{n}, \tag{1}$$

where n is the number of steps in a time unit, and v is the walking speed. In other words, Eq. (1) is the energy component for horizontal movement. For vertical movement, one needs to consider the extra potential energy to overcome when walking upstairs. After consulting the statistic data in [7] about climbing up stairs, we come up with the following equation:

$$E_v = -0.18h^2 + 7.14h, \tag{2}$$

where h is the positive height of a stair (h<40cm). In addition, for the same ground condition, different walking styles may result in different energy consumption because the amounts of joint rotations could differ. Therefore, we

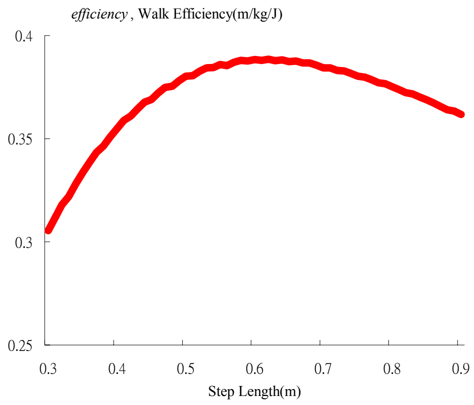


Figure 2. Relation between step lengths and energy efficiency on a flat ground

use the following equation to approximate the internal energy consumption:

$$E_j = c_{thigh} \cdot (\delta\theta_1 + \delta\theta_3) + c_{calf} \cdot (\delta\theta_2 + \delta\theta_4), \tag{3}$$

where $\delta\theta_i$ are the amounts of joint rotations during a motion cycle and c_{thigh} and c_{calf} are the coefficients for the masses for thigh and calf, respectively. We have used a linear combination of the above three energy components to obtain an estimate on the overall energy consumption as in the following equation:

$$E = \alpha \times E_n + \beta \times E_v + \gamma \times E_j \tag{4}$$

where α, β, γ are the empirical weighing coefficients for the three components.

For the purpose of energy saving, we can not only consider the absolute energy consumption, but also need to account for the progress being made such as the moved distance. We assume that the efficiency of a walk cycle (*eff*) is defined as follows.

$$eff = \frac{step\ length}{E}, \tag{5}$$

where E is the overall energy consumption in Eq.(4). We compute the efficiency of walk cycle for different step lengths and depict the curve of their relation in Figure 2. The most energy efficient step length is around 0.6 meters, which is consistent with our daily experience.

4.2. Footstep planning algorithm

Our footstep planner is a local planner that only consider the next few steps at a given foot location. Before making a step forward, the system will call the planner to verify if there exist legal configurations in the next few steps. Assume that the *footstep configuration* q_f is defined as the set of footstep lengths, s_i, for the next n steps: $q_f = (s_1, s_2, s_3, \ldots, s_n)$. Generally speaking, humans plan for the next three to four steps as they walk. Therefore, n is usually set to three or four. The goal of the planning problem is to find a collision-free footstep configuration that is the most efficient in terms of energy consumption. Since n is not large in general, it is possible to enumerate all possible footstep configurations in order to find the globally most efficient one. However, the real-time requirement of our target application prevents us from doing the exhaustive search. In our system we use a Best-First approach to find a local minimum on efficiency only in order to return a feasible footstep plan as early as possible.

```
FIND_INIT ()
INSERT ($q_0$, INITIAL);
mark $q_0$ visited;
SUCCESS $\leftarrow$ false;
while $\neg$ EMPTY(INITIAL) and $\neg$ SUCCESS do
    $q$ $\leftarrow$ FIRST(INITIAL);
    if Collide($q$) then
        for every neighbor $q'$ of $q$ in the grid do
            if $q'$ is not visited then
                INSERT($q'$, INITIAL);
                mark $q'$ visited;
    else SUCCESS $\leftarrow$ true;
if SUCCESS then
    return $q$;
else return failure;
```

Figure 3. Algorithm to search for the first collision-free footstep configuration

```
BFP ()
INSERT ($q_{init}$, OPEN);
mark $q_{init}$ visited;
while $\neg$ EMPTY(OPEN) do
    $q$ $\leftarrow$ FIRST(OPEN);
    for every neighbor $q'$ of $q$ in the grid do
        if legal($q'$) then
            if Eff($q'$) > Eff($q$) and $q'$ is not visited then
                INSERT($q'$, OPEN);
                mark $q'$ visited;
            if all Eff($q'$) < Eff($q$) then
                return $q$;
```

Figure 4. Best-First Planning algorithm

When the search starts, we initialize the footstep configuration q_{init} with (s_{init}, s_{init}, …. , s_{init}), where s_{init} is the initial neutral footstep length. Then the search is comprised of two steps: searching for an initial legal configuration and searching for a legal configuration with the minimal energy consumption. We will use the algorithm in Figure 3 to find the first collision-free footstep configuration. The algorithm is basically a Breadth-First al-

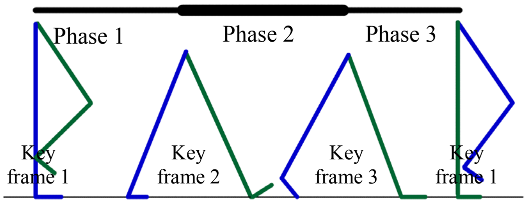


Figure 5. Three key frames and phases of a walk gait

gorithm that expands nodes outward from the initial node until a legal one is found and returned. In the second step, a Best-First Planning (BFP) algorithm is used to find the configuration with the minimal energy consumption as shown in Figure 4. During the search, the algorithm always expands from the node with minimal energy and stops when a local minimum is reached.

5. Kinematics-Based Locomotion Generation

In the planning algorithm above, the legality of a footstep location is checked by systematically trying different locomotion for the given footstep length. The walk gait is generated by an inverse kinematics approach in whcih we first compute the key frames in a gait cycle and use interpolation to generate the frames between keys. If the interpolated frames result in the swinging leg to collide with the environment, we will modify the control points of a Bezier curve to adjust the trajectory of a swinging leg.

5.1. Key frame generation and interpolation

We determine the key frames of a walking gait by observing the different phases of a human walk cycle as shown in Figure 5. Each key frame can be clearly defined according to the given footstep length. For example, the first key is defined at the pose where one leg is standing still while the other leg is slight above the ground. The second key is at the location where the front leg touches the ground. The third key is defined as the pose where the back leg leaves the ground. The fourth key will return back to the first key and complete the cycle. These keyframes are computed according to the inverse kinematics methods in [11].

We use interpolation to generate the frames between two key frames. Specifically, we interpolate the positions of the pelvis and the ankle in each frame and compute the knee angle according to the inverse kinematics equation for the two links. Each phase between two key frames has its own way for doing interpolation. The general principles for interpolation are as follows [11]. When a leg is stretching out, we use the ankle as the center to rotate the pelvis location. If the leg is swinging, we will use the most energy efficient way to specify the ankle trajectory.

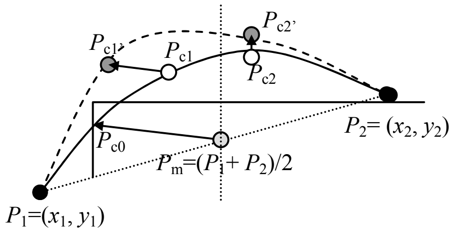


Figure 6. Adjusting the Bezier curve for the swinging leg to avoid collisions

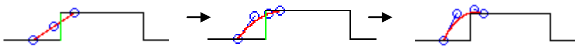


Figure 7. Example of adjusting the Bezier curve

5.2. Ankle trajectory adjustment

The leg motions may have collisions with the environment in the interpolated frames. However, some of these collisions can be avoided by adjust the trajectory of the ankle. In our system, we use a Bezier curve to represent the ankle trajectory of the swinging leg. The end points of the curve are the given locations of footsteps. By adjusting the control points of the curve, we can change the shape of the curve to escape from the collision situation. Since each control point has two degrees of freedom and the Bezier curve has two control points, the total degrees of freedom are four. How to adjust these two points in order to position the curve segment above the ground profile becomes the problem we will address next.

The locations of the control points start at some location slightly above the midpoint between two endpoints. When a collision between the curve and the environment is detected, the control points are moved upward and outward to lift the curve as shown in Figure 6. The curve is divided into two parts (left and right) from the summit point. Each part computes the intersected points of the curve and moves the corresponding control point for some distance along the direction from the midpoint of two endpoints toward the intersection points. If one part does not cause intersection while the other does, its control point also moves upward for the same amount as the other control point moves. The search process stops when a legal curve is found or the locations of the control points have left a given legal region. In the second case, the planner returns failure. An example of how the adjustment works is shown in Figure 7.

5.3. Timing arrangement

In addition to being collision-free, correct timing is a key factor for a walk motion to look realistic. Although we can use dynamic simulation to generate the timing information, lacking a good force exertion model and requiring real-time performance prevent us from using

Table 1. Timing information of three phases for three ground conditions

	T_1	T_2	T_3
flat	44 %	28%	28%
up-hill	$(44-30\times\frac{h}{leg\ length})\ \%$	$(44-30\times\frac{h}{leg\ length})\ \%$	$(44-30\times\frac{h}{leg\ length})\ \%$
down hill	$(44-30\times\frac{h}{leg\ length})\ \%$	$(44-30\times\frac{h}{leg\ length})\ \%$	$(44-30\times\frac{h}{leg\ length})\ \%$

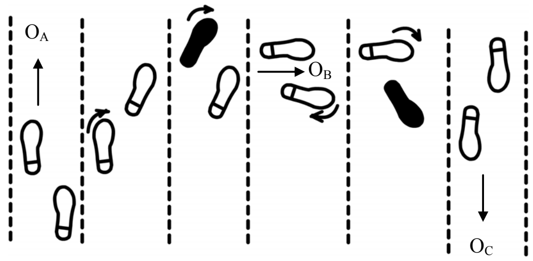


Figure 8. Turning foot location on a curved path

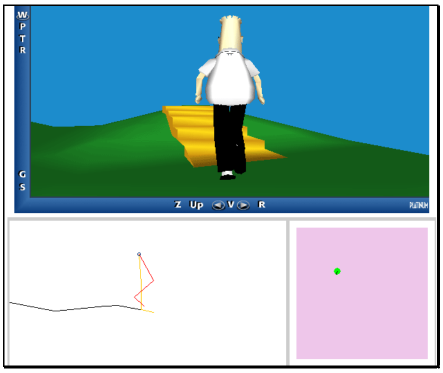
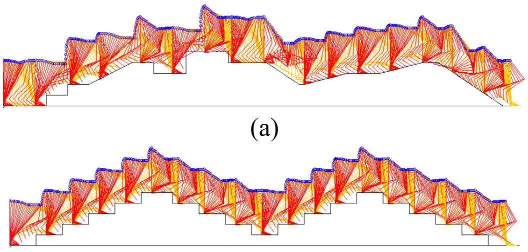


Figure 9. Snapshot of the interactive user interface

this approach. What we propose is by assigning each phase an appropriate timing (percentage of the overall time in a cycle) according to statistic data in the literature. In [9], the timing information, T_1, T_2, and T_3, for the three phases in subsection 5.1, as shown in Table 1, can be observed in a regular walk cycle with nominal speed on a flat ground. According to [8], the timing for uphill and downhill situation can also be adjusted according to the height differences as summarized in Table 1.

5.4. Consideration for curved paths

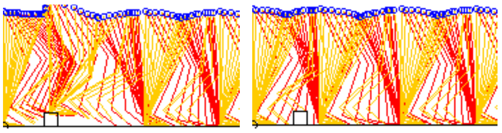
Although the ground profile given to the system is a two-dimensional polyline extracted from a path, the actual path could be a curve path. After the locomotion is found, we need to restore the possible orientation change along the path and reflect the change on the foot placement to avoid slipping on the ground. In our system, we have proposed a simple mechanism, similar to the one proposed in [3], to restore the orientation changes gradually along a path. Depending on the front leg and the turning direction, we determine which leg should make the turn. If the amount of orientation change is greater than some threshold, the system will divide the rotation into several smaller ones. An example of how the feet make a 180-degree turn is shown in Figure 8. Note that it takes four smaller rotations on both feet to complete the turn.



(a)

(b)

Figure 10. Examples of walking on uneven terrains



(a) (b)

Figure 11. Comparison of locomotion efficiency with and without footstep planning

6. Experimental Results

The above proposed system has been fully implemented in the Java language. A standard VRML browser connected to the planning system has been used as the front-end 3D display. A user can use a regular mouse to navigate through a 3D scene with an uneven terrain and stairs such as the one shown in Figure 9.

6.1. Walking gait examples

Two examples of the walking gaits on an uneven terrain is depicted in Figure 10. The gaits succeed in avoiding collisions with the ground or stairs by adjusting the Bezier curve for the ankle trajectories.

In Figure 11, we show an example that illustrates how the footstep planner accounts for energy consumption and generates a more efficient gait. The motion without footstep planning in Figure 11(a) steps onto the short obstacle while the other one with planning in Figure 11(b) strides across the obstacle. The latter one is considered to be more energy efficient since it avoids lifting the body weight while making the same progress.

6.2. Planning efficiency

The planning times for several examples are reported in Table 2. These planning times are taken on a regular

Table 2. Timing information of three phases for three ground conditions

Types of Terrain	Planning time (sec.)
Flat Ground	0.015
Upstairs	0.067
Uphill	0.019
Downstairs	0.103
Downhill	0.013
Example in Fig. 10 (a)	0.044
Example in Fig. 10 (b)	0.069

PC with a 1.3GHz CPU. The range of reasonable footstep lengths is divided into 13 intervals such that an increment of the footstep length is about 5 cm. The footstep planner takes less than 0.02 seconds on average to generate the motion for one step on the flat, uphill, and downhill ground conditions. It takes a bit longer for the cases of walking upstairs and downstairs. This is mainly due to the fact that walking on stairs need to carefully adjust the ankle trajectories of the swinging leg by controlling its Bezier curve in order to avoid collisions with the stairs. The collision checks between the curve and the ground is the more time consuming part of the computation.

For the example in Figure 10(b), the planning is around 0.069 seconds and the display frame rate is around 15 frames per second. For application requiring even higher frame rate or for the even more difficult scenarios, we can reduce the resolution of the search space in the footstep planner in order to reduce the planning time. For example, if we change the resolution from 13 to 11 in the range of acceptable footstep length, the planning time can be reduced to 0.054 seconds. It can even be further reduced to 0.03 if only seven intervals are used. By trading completeness with efficiency, the user can allocate appropriate time budget to the system so that it can achieve the requirement of real-time performance.

7. Conclusions and Future Work

In this paper, we have described a motion planning system that can generate humanoid lower body motion in real time. Given the ground profile along a given path, the system can generate a legal and efficient footstep plan according to a proposed energy model. Detail locomotion for the walking gaits can then be generated by an inverse kinematics approach according to these footstep locations. We use a Bezier curve to represent the ankle trajectory of a swinging leg. If the leg along the trajectory collides with the ground, the Bezier curve is adjusted until a collision-free trajectory is found. Furthermore, in order to make the generated motions look more realistic, we adopt a timing model obtained from measured data to determine the appropriate timing for each phase of a walk cycle. In addition, we also have designed a mechanism to determine the orientation change of a footstep in a walk cycle. Finally, several examples are shown to demonstrate the capability of such a planning system, especially in a real-time or interactive environment.

The current system only considers planning the motion of the lower body and the arm motion is only generated in accordance with the legs during a walk cycle. However, it would be interesting to extend the planner to consider upper body motions. In addition, we should be able to use the same principle to generate the motion of various types such as running and jumping.

8. Acknowledgement

This work was partially supported by National Science Council under contract NSC 91-2213-E-004-005.

References

[1] A. Bruderlin and T. W. Calvert, “Goal-Directed, Dynamic Animation of Human Walking,” *Proceedings of ACM SIGGRAPH*, 1989.

[2] M. Girard and A.A. Maciejewski, “Computational Modeling for the Computer Animation of Legged Figures,” *Computer Graphics, (Proceedings of SIGGRAPH95)*, 19(3), pp. 263-270, 1985.

[3] K. Hase and R. B. Stein, “Turning Strategies During Human Walking,” *The American Physiological Society*, 1999.

[4] H. Ko, *Kinematic and Dynamic techniques for Analyzing, Predicting, and Animating Human Locomotion*, PhD thesis, University of Pennsylvania, 1994.

[5] M. G. Pandy and N. Berme, “A Numerical Method for Simulating the Dynamics of Human Walking,” *Journal of Biomechanics*, 1988.

[6] HJ. Ralston, “Energy-Speed Relation and Optimal Speed During Level Walking,” *Int Z Angew Physiol*, 1958.

[7] J. Rose and J. G. Gamble, “Human Walking,” *Williams and Wilkins*, 1994.

[8] K. Tsutsuguchi, S. Shimada and Y. Suenaga, “Human walking animation based on foot reaction force in the three-dimensional virtual world,” *The Journal of Visualization and Computer Animation*, 2000.

[9] Winter DA, *Biomechanics and Motor Control of Human Movement*, New York: Wiley, 1990.

[10] HANIM (The humanoid animation working group), URL :<http://www.hanim.org>.

[11] P.F. Chen, and T.Y. Li, 2002.6, "Generating Humanoid Lower-Body Motions with Real-time Planning," in *Proceeding of 2002 Computer Graphics Workshop*, Taiwan.