Knowledge-Driven, Interactive Animation of Human Running

Armin Bruderlin
ATR Media Integration & Communications Research Labs
Seika-cho Soraku-gun
Kyoto 619-02, Japan
email:armin@mic.atr.co.jp

Tom Calvert
Simon Fraser University
Burnaby, B.C., V5A 1S6, Canada
email:tom@cs.sfu.ca

Abstract

A high-level motion control system for the animation of human-like figures is introduced which generates a wide variety of individual running styles in real time. Sequences such as a leisurely jog, a fast sprint or a bouncy run are conveniently obtained by interactively setting the values of "running" parameters such as desired velocity, step length, "flight" height or "heel/toe" strike while observing a running figure on the screen.

The algorithm incorporates knowledge of how humans run at several levels: *empirical* knowledge defines the relationships between the running parameters; for example, a change in running velocity by the user triggers a change in step length to maintain a "natural" running stride. *Physical* knowledge calculates the trajectory of the body for the current running step. Knowledge about limb-coordination of a running stride is utilized for establishing both, *state-constraints* which define the support and flight states, and *phase-constraints* to "guide" the internal joint-angle interpolation for the stance and swing phases.

Keywords: human figure animation, motion control, procedural animation.

1 Introduction

Animating the motion of human figures is a challenging task. Traditionally, an animator has to tediously specify many keyframes for many degrees of freedom to obtain a desired movement. At the same time, spatial and temporal components of a movement, coordination of the limbs, interaction between figures as well as interaction with the environment need to be resolved. Keyframing supports the animation process only at the lowest level providing tools for the manipulation of joint angles or coordinates (i.e. low level motion parameters), and the animator has to explicitly account for higher level interactions based on experience and skills.

We propose a higher level motion control technique to animate human running which alleviates the tedious detail the animator has to specify. A higher level of control is achieved by incorporating knowledge about how people run into the motion control algorithm. High-level motion parameters such as velocity, flight height, pelvic rotation or stride width generate variations in runnings styles with different expressions. These parameters can be changed interactively to give real-time feedback on their effect on the motion — analogous to 'tweaking' low-level parameters such as control points of joint angle trajectories in a keyframing system.

Several other approaches have been proposed to animate movements of human-like figures at a higher, above joint-manipulation level. *Inverse kinematics* has been successfully applied to ease limb positioning [5] and motion generation [20, 25] by encoding knowledge on how the joint angles change given the position of an end-effector. *Physically-based* approaches [14, 29] incorporate knowledge in form of dynamic models to produce realistic although often expressionless motion, which is also difficult to control since the driving forces and torques need to be approximated for a particular movement. Hybrid kinematic-dynamic techniques [18, 30] have been proposed to avoid some of the disadvantages associated with purely dynamic systems. *Optimization* techniques have been applied to simplified articulated structures to generate motions such as throwing [9, 23] and lifting [22]; here, knowledge is enbedded in a cost function term and constraints whose solution approximates a desired movement. Motion control at a very high level is provided by *planning* systems which have knowledge of an entire scene, and a planner autonomously generates the animation [6, 20], including gestures and conversations between characters [7]. However, such high-level systems often produce plain animations lacking expression and personality.

Most higher level approaches also share the drawback of not performing in real time, making it difficult for the animator to interactively and iteratively construct and refine a desired movement. There is usually a trade-off between how much and what the animator has to specify, and how expressive and close the resulting motion is to what the animator had in mind, that is how much the system "assumes" about motion. For instance, traditional keyframing is an 'assisting' tool which assumes very little about a motion by mainly taking care of interpolation and administration of the input, while it is really the animator who does motion control. On the other hand, animating a bouncing ball in a physically-based system leaves the animator with very little control for fine tuning after the equations of motion have been set up and initialized.

The disadvantages of lack of control, lack of expression and non-interactivity usually attached to higher level systems have been overcome by our motion control algorithm for human running. The creative control stays with the animator who can interactively adjust high-level parameters to obtain a desired running style. In section 2, related research in animating human locomotion is outlined and the main concepts of human running are defined. Section 3 describes the motion control algorithm for running in detail followed by a discussion and examples in section 4. Conclusions and extensions for future work are addressed in section 5.

2 Bipedal Running

In human locomotion, walking and running are the most important and most frequently used gaits. A lot of investigation has been done on human walking (see [17] for a good overview), and at least conceptually and kinematically it is well understood. On the other hand, there is not as much consensus in research on running. This has much to do with the fact that there is a larger number of running styles compared to walking. For example, humans can run at any speed at which walking is possible, they can run leisurely at a modest pace, they can jog or sprint at maximum speed, which results in a much greater variability in the kinematics than, say, between a slow and a fast walk. Furthermore, the choice between one of the two basic running styles—either the toe or the heel impacting with the ground first at the end of a running step—has different effects on the kinematics (and the dynamics which are ignored here) of the foot and leg during stance. Variations between running subjects are also more pronounced than in different people walking because of the flight or airborne state which does not occur during a walking stride. Lastly, we believe that anatomical differences between humans such as unequal leg lengths and muscle strengths produce a wider range of kinematic patterns in running than in walking since strength becomes more of a factor.

2.1 Related Work

Early work in animating legged figures includes PODA [12], which produces a variety of gaits. Simple dynamics are applied to calculate the motion of the body as a whole, while the legs are animated kinematically, using a pseudoinverse Jacobian technique to determine the leg angles while keeping the feet on the ground during support. A different approach to animate legged locomotion was taken by Raibert et al. [26]. An internal control algorithm for a dynamic running model is decomposed into three independently treated functions: a vertical component to regulate hopping height, and a horizontal component regulating body attitude (balance) and forward velocity. Hodgins [13, 14] introduced a related approach to dynamically animate human running. The control algorithm relies on a cyclic state machine which determines the proper control actions to calculate the forces and torques such that the desired forward speed is satisfied. Stewart [27] presented another active control dynamically based system which allows the user to write an algorithm (in LISP) for a particular motion. The equations of motion are constructed by the algorithm, which then controls the motion by setting values of variables and keeping track of the state of the simulation. Although convincing examples of walking sequences of a simple biped have been demonstrated, this approach requires the user to know a lot about the motion to be animated by writing an appropriate algorithm. In an effort to produce more "human-like" motion, techniques have been developed to generalize rotoscoped data for animating human walking [2, 19] such that variations in direction and step lengths for figures of different height can be obtained.

A different approach was taken with the development of KLAW [3, 4]. By incorporating knowledge about how humans walk, the system generates realistic human walking animations at a high level while still allowing for variations in the movement. In this way, a user can conveniently create various styles of walking by specifying parameters like step length, velocity, or stride width. Compared to the other locomotion algorithms above, this approach performs interactively, providing real-time feedback and the ability to produce different styles of locomotion. The method introduced here to animate human running is similar to KLAW in satisfying these criteria. However, a different control scheme is adopted (see section 3) since running is a different motion from walking (see next section) requiring a different set of control parameters (see section 3.1).

2.2 Running Concepts

A bipedal running stride consists of 2 steps as shown in Figure 1. If we assume the motion of the legs to be symmetric we can restrict our analysis to one step, for instance from heel-strike of the left leg to heel-strike of the right leg. A running step is made up of a single support state where one foot is off the ground, and a flight state where both feet are off the ground. With respect to one stride, each leg cycles through a stance phase and a swing phase, shifted in time. Compared with walking where the stance phases for the two legs overlap, in running the swing phases overlap.

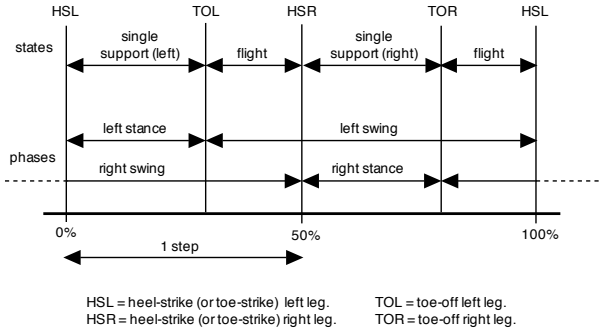

Figure 1: Running stride.

A running stride can be characterized by the following four high-level parameters: velocity (v), step length (sl), step frequency (sf) and flight height (H), where

$$v = sl \times sf. \quad (1)$$

Humans when told to run at a particular velocity or step length naturally "choose" the other parameters to maintain a comfortable running stride. For a high-level motion control scheme it is therefore crucial that the inter-relationships between these parameters are determined. Research on human running indicates that an increase in velocity is accompanied by a linear increase in step length up to a speed of about $400\,m/min$ ($\sim 24\,km/h$), then the step length levels off and only step frequency is increased to further increase velocity [15, 16, 21]. On the other hand, step frequency increases in a curvalinear manner with respect to velocity; small increases in step frequency at lower velocity, and larger increases at higher velocity [10]. This linear relationship between v and sl is now established more formally based on actual human running data from [15, 16, 21, 28]. The formula below was derived by linear regression applied to the data with $v \leq 400 m/min$. A multiple r^2 coefficient of 0.94442 was obtained (for $n = 80$ data pairs) indicating very good correlation (sl is in m and v in m/min):

$$sl = 0.1394 + 0.00465\, v.$$

To this equation, a term $level$ is now added which models the level of expertise in running based on the observation [10, 21] that better runners have a greater stride length at a given velocity than less skilled or poor runners, where -0.001 (poor) $\leq level \leq 0.001$ (skilled):

$$sl = 0.1394 + (0.00465 + level)\, v.$$

Another factor influencing this relationship between v and sl is leg length. For walking, Inman derived a normalization formula [17] relating step length to body height (which is a function of leg length). For running, we adopt an approach suggested by Alexander [1]. He observed that geometrically similar animals of different sizes have runs which are dynamically similar whenever their speeds made their *Froude* number equal. The Froude number is defined as the ratio between kinetic and potential energies, $v^2/(2gl)$, where l is the leg length. Since leg length is proportional to body height, the Froude equality can be expressed as $v^2/body\_height = v_{1.8}^2/1.8$, where $1.8\,m$ is the default body height. Putting this information into our equation relating velocity to step length, we now have

$$sl = 0.1394 + (0.00465 + level)\, v \sqrt{\frac{body\_height}{1.8}}. \quad (2)$$

Figure 2 illustrates equation 2 as well as the real data pairs (crosses). The three lines close together indicate relationships with $level = 0$ and body height equal to $1.7\,m$ (lower), $1.8\,m$ (middle) and $1.9\,m$ (upper). For the two lines with the smallest an largest slope, body height is $1.8\,m$ with $level$ set to minimum for the former and maximum for the latter, respectively.

Equation 2 represents the normalization formula for running. In correspondence with research on running,

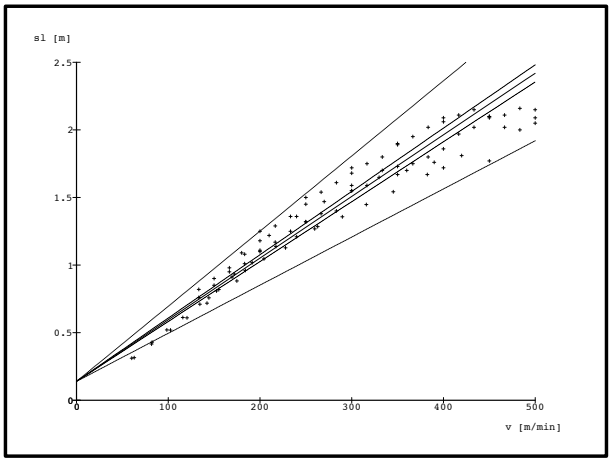

Figure 2: Step length as a function of velocity.

we apply this formula up to v_{norm}, where $v_{norm} = 400\,m/min$ ($\sim 24\,km/h$) for a human with a body height of $1.8\,m$. If v increases further, sl is kept constant (sl_{norm}) and only sf is increased ($v = sl_{norm} \times sf$).

Equation 1 and 2 establish a relationships between v, sl and sf. The fourth running parameter, flight height H, is a function of flight time t_{flight} and the vertical position of the pelvis at toe-off (y_1) and heel-strike (y_2); g denotes the gravitational acceleration:

$$H = \left(t_{flight} - \frac{2\,(y_1 - y_2)}{g\, t_{flight}} \right)^2 \frac{g}{8}. \quad (3)$$

Thus, in order to calculate H the duration of flight needs to be known. Research on human running suggests that an increase in velocity has little effect on the time for flight [15, 21, 24]. However, the time for support in running decreases significantly as the speed is increased [10]. Therefore, an increase in step frequency is due mainly to a decrease in the time for support. Correlations on the running data by [15, 21, 24] suggest that the time of flight initially increases with step frequency, reaches a maximum at about $190 steps/min$ after which it levels off slightly. A best fit (residual mean square = 0.00087 for $n = 20$ data pairs) was obtained by a 3rd order polynomial. As noted in the literature [21], expert runners tend to have a longer flight period than poor runners at comparable step frequencies. Incorporating a $level$ attribute into the equation similar to above (here, -0.0001 (poor) $\leq level \leq 0.0001$ (skilled)), we have

$$\begin{aligned} t_{flight} = \quad & -8.925 + (0.131 + level)\, sf \\ & -0.623\, 10^{-3}\, sf^2 + 0.979\, 10^{-6}\, sf^3. \end{aligned} \quad (4)$$

The running data available ranged about between 160 $step/min$ and 230 $steps/min$ (t_flight is in sec). To get a more general expression for the relationship between sf and t_{flight}, extrapolation becomes necessary. From experience, we think that the following quadratic equation provides a good relationship below 180 $step/min$:

$$t_{flight} = \begin{array}{l} -0.675\ 10^{-3} - (0.15\ 10^{-3} + level)\ sf \\ +0.542\ 10^{-5}\ sf^2. \end{array} \quad (5)$$

The above leads to the following calculation of t_{flight} from sf: from 0—180 $steps/min$, equation 5 is used; from 180—230 $steps/min$, equation 4 is applied; above 230 $steps/min$, t_{flight} is kept constant.

Now we are ready to calculate the durations for the leg phases in running. From Figure 1 it follows that

$$\begin{aligned} t_{stance} &= t_{step} - t_{flight}\ ; \\ t_{swing} &= t_{step} + t_{flight}\ ; \end{aligned} \quad (6)$$

where t_{step} is $1/sf$. These durations manifest timing constraints which guarantee natural looking interpolations of the joint angles of a desired running stride. This is explained as part of the running algorithm in the next section.

3 Running Algorithm

Our motion control algorithm for human running has been developed with the following four design goals in mind to make it a useful tool for an animator: ease of motion specification, interactivity and real-time feedback, generation of believable motion, ease of customizing and personalizing motion. To meet these goals, knowledge about human running has been incorporated at various levels. *Empirical* knowledge is applied to determine the kind of control parameters and how they are interrelated: the four main running parameters introduced in the previous section define a basic running stride while nineteen attributes can be set to individualize a run such as amount of torso tilt, arm swing, and choosing between heel-strike or toe-strike. *Physical* knowledge determines how the center of the body moves during a running stride: during the support state its motion is defined by an interpolating cubic spline, whereas during flight it follows a parabolic trajectory. *Limb-coordination* knowledge is used to set up both state-constraints for support and flight, and phase-constraints to "guide" the joint-angle interpolation of the stance and swing phases.

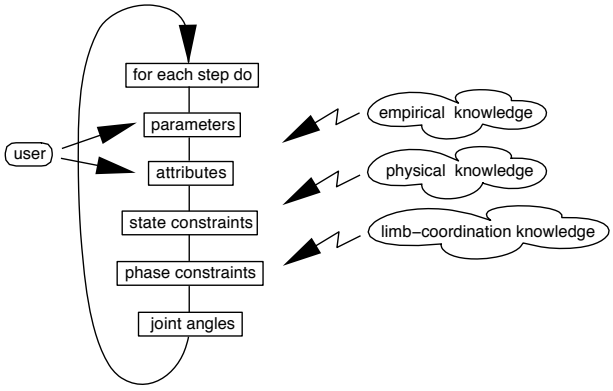

Figure 3: Running algorithm.

The basic algorithm is illustrated in Figure 3. The main loop is executed for each running step, which means that changes the user makes to any of the parameters or attributes become active on the next running step. At the end of each step, the current front leg and hind leg are switched to initialize the next step. This way, acceleration and deceleration are possible, as well as starting and stopping which are just special cases of a "normal" step (see section 3.3 below). In the following, a closer look is taken at each part of the algorithm in turn.

3.1 Parameters and Attributes

The parameters and attributes of the running algorithm are the interface to the user and control the current running stride. They have default values which can be changed interactively via sliders to customize a run. A distinction between *parameters* and *attributes* has been made such that the parameters define the basic running stride while the attributes change the expression or personality of the stride. A change in one of the parameters causes a change in the other parameters to maintain a natural stride, as well as in some attributes (see section 3.1.2 below), whereas a change in any of the attributes is independent of the parameters.

3.1.1 Parameters

The parameter panel is illustrated in Figure 4. There are four main parameters: velocity, step length, step frequency and flight height, plus an additional slider for the level of expertise in running as discussed in section 2.2 above. If one of the parameters is changed, the other ones are updated using equations 1, 2, 3, and 4 or 5, respectively. The actual calculations of flight height and time of flight are a bit more complex and explained in section 3.2, since we do not yet know the vertical position of the pelvis (kinematic center of the body) at toe-off and impact (y_1 and y_2 in equation 3).

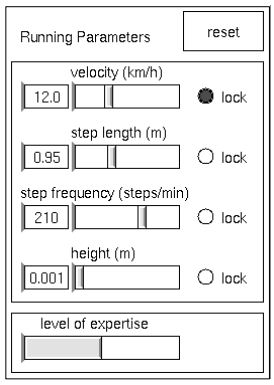

Figure 4: Running parameters.

Parameters can be *locked* to produce somewhat unnatural strides. This is shown in Figure 4, where the velocity was locked at 12 km/h and then step frequency was increased from the default 187 $steps/sec$ to 210, which decreased the normally chosen step length of 1.07 m to 0.95 and the flight height from 0.004 m to 0.001.

3.1.2 Attributes

The attributes allow the user to individualize a running stride. As shown in Figure 5, nineteen attributes have been implemented at this time. Five of these control the movement of the arms and shoulders, two attributes control torso tilt and sway, three attributes are provided to alter the motion of the pelvis and nine attributes change the motion of the legs in one way or other. Among the attributes for the legs are a heel-toe-strike button (heel or toe touches ground first at impact), a bounciness slider (amount of knee bend during support), an overstride slider (amount by which the foot impacts the ground ahead of the body), and a stride-width slider (lateral distance between the feet).

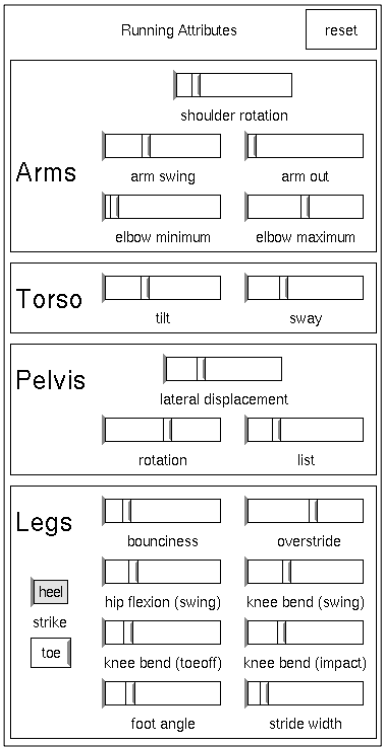

Figure 5: Running attributes.

As with the parameters, default values for the attributes are chosen to produce a natural running stride. This involves the values of some attributes being automatically adjusted when a parameter value changes. The adjustment is necessary since there is a large range in the kinematics of running from a slow run to a fast run. For example, the overstride value decreases with increasing velocity, pelvic rotation increases with step length, bounciness increases with flight height, whereas the knee angle at toe-off decreases with step length. These relationships are implemented as linear functions of the current parameters, the attribute values and their extreme values. Some attributes are also coupled. For instance, lateral displacement increases as stride-width increases and both decrease with step frequency. Lastly, there are some "hidden" attributes which can not be set directly by the user but are automatically calculated; for example, the ankle and metatarsal angles at toe-off which are functions of step length, or the ankle angle at impact which is a function of the overstride and the heel-toe-strike attributes.

Based on the parameters and attributes settings for the current running step, the state constraints including the motion of the pelvis are now determined, followed by the phase-constraints which guide the interpolation of the joint angles (section 3.3).

3.2 State Constraints

The state constraint principle is illustrated in Figure 6 for a step beginning with heel-strike (subsequently used to mean heel-or-toe-strike) of the right leg and ending at heel-strike of the left leg. For the opposite step from heel-strike left leg to heel-strike right leg, all the calculations below are mirrored. The constraints establish the leg angles for the stance leg at the beginning of a step (HSR) and at toe-off (TOR), as well as the leg angles of the swing leg at the end of the step (HSL). These internal "keyframes" serve as the basis for interpolating the leg angles in section 3.3.

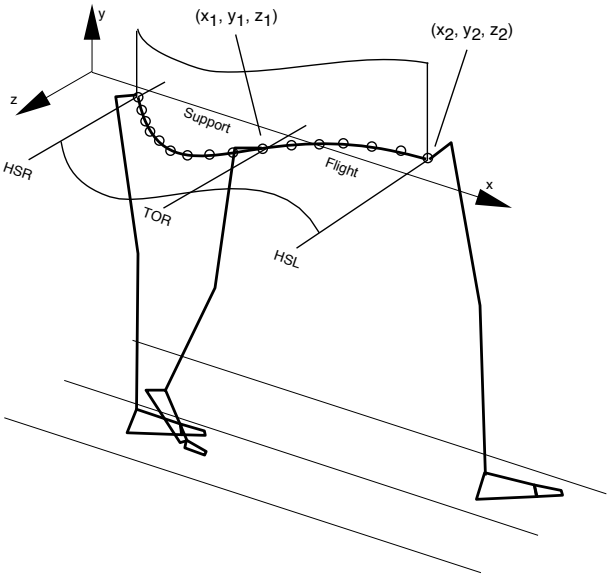

Figure 6: State constraints.

We now explain how these constraints are calculated given the current parameters and attributes, assuming that the leg angles at the beginning of the step (HSR) are known from the end of the previous step or the resting position on the initial step. All the computations are done in 3-D. For the explanations below it is noted that according to research on human running, the actual values for lateral displacement of the body at toe-off and heel-strike are 80 % (to either side of neutral) of the maximum value given by the attribute. Similarly, pelvic rotation (in the transverse, x–z plane) is a maximum at toe-off and about 20 % at heel-strike, whereas pelvic list (in the coronal, y–z plane) is a minimum (zero) at toe-off and about 80 % at heel-strike. This is shown in Figure 7. It is also illustrated that at mid-support, lateral displacement as well as pelvic list are a maximum and rotation is a minimum (zero).

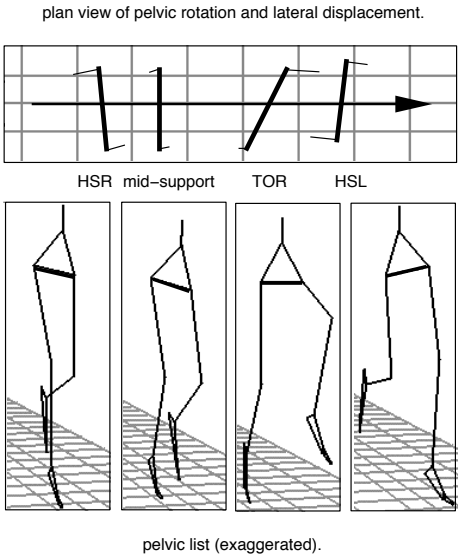

Figure 7: Motion of pelvis for one step.

The first stage in calculating the state constraints for the current step is to add the current step length to the heel position at HSR to obtain the heel position at the end of the step (HSL). Then the position of the center of the pelvis at HSL, (x_2, y_2, z_2), is determined "bottom-up" using the current attributes for stride width, overstride, knee bend at heel-strike, as well as the percentages of lateral displacement of the body, pelvic rotation and list at heel-strike. Given the position of the pelvis at heel-strike, we now calculate "backwards" using information on the flight state to obtain the position and orientation of the toe-off leg (TOR). For this purpose, the length of the toe-off leg (rad) from toe to the center of the pelvis is computed first using the current attribute values. Note that we can not directly calculate the pelvis position at toe-off since the orientation of the leg in the sagittal $(x-y)$ plane is unknown (unlike at heel-strike where it is known from the overstride attribute).

Two cases need to be considered now: (I) the flight time (t_{flight}) is given from equation 4 and the flight height H is still to be determined (which is the case if any of v, sl, sf were changed by the user); (II) H is given and t_{flight} is still unknown (which is the case if H was changed by the user). In case (I), there is an analytic solution to calculating the pelvis (x_1, y_1, z_1) at toe-off:

$$x_1 = x_2 - v\, t_{flight}\,; \tag{7}$$

$$y_1 = \sqrt{rad^2 - (x_1 - x_{toe})^2}\,; \tag{8}$$

where x_{toe} is known from the previous step and z_1 is the percentage of lateral displacement at toe-off as explained above. Given $y1$ and $y2$, H is obtained from equation 3. For case (II) the pelvis at toe-off is solved numerically by a two-dimensional Newton-Raphson method [11] using equations 3 and 8, whereby $\frac{x_2 - x_1}{v}$ is substituted for t_{flight} in the former. Once x_1 is known, we determine t_{flight} by equation 7.

With these "keyframes" at HSR, TOR and HSL in place, the translation of the pelvis can now be determined. During flight, the pelvis follows a parabolic trajectory, with $0 \le t \le t_{flight}$ and z being interpolated between z_1 and z_2:

$$x = x_1 + v\, t\,;$$
$$y = y_1 + \sqrt{2\, g\, H}\, t - \frac{1}{2}\, g\, t^2.$$

During support, the pelvis moves along an interpolating cubic spline segment, whose four control points are at HSR, mid-support, TOR and mid-flight. Whereas the coordinates at HSR, TOR and mid-flight are known from above, the control points for mid-support are chosen as follows: from research on human running [8], it is known that the kinetic and potential energy changes within a stride are simultaneous and are both lowest about the middle of support. Assuming that the motion of the whole body is represented by the pelvis, the vertical position of the pelvis at mid-support is a minimum, defined as a function of the vertical pelvis position at HSR and TOR, as well as bounciness and flight height, such that the bigger H the lower y; the x-position of the pelvis at mid-support is also a minimum (behind average forward position p) where a value of $0.8 \times p$ has given good results. Finally, the z control point of the pelvis at mid-support is set to the maximum lateral displacement as mentioned above. The translation of the pelvis is illustrated in Figure 6, with changes in velocity indicated by differently spaced circles along the path; also shown is lateral displacement of the body.

The calculation of the orientation of the pelvis—rotation and list—during the current step completes the state constraints. This is done by linear interpolation between the four keyframes shown in Figure 7. The positions of the hip for the stance and swing leg are now known and used next to interpolate the leg angles.

3.3 Phase Constraints

Given the constraints on the support and flight states of the current running step introduced in the last section, the phase-constraints further subdivide the constraints with respect to the stance and swing phases of the legs in order to naturally interpolate the joint angles. The subdivisions are based on observations of how real humans run. This is illustrated in Figure 8 and explained below.

3.3.1 Single Support

During single support from HSR to TOR, the stance leg angles (right, solid leg in Figure 8) are calculated given the hip and the heel/toe position, as well as the leg angles at HSR and TOR. We assume the motion of the foot to be planar (vertical). At the beginning of stance, the foot rotates around the heel (heel-strike) or metatarsal joint (toe-strike) until it is flat on the ground at mid-support. In this phase, the position of the ankle and therefore the ankle-hip distance are known and from this the other leg angles are derived trigonometrically. Subsequently, the foot stays flat on the ground until the beginning of

the next sub-phase during stance which lasts until TOR and is triggered by either the body passing through the vertical or the ankle-hip distance becoming bigger than the length of the extended leg. In both cases, the heel comes off the ground. This is implemented by interpolating both the metatarsal and knee angles from the current time and values until TOR. Then the circle around the toe with the radius toe-ankle (with current metatarsal angle) is intersected with the sphere around the hip and radius thigh-shank (with current knee angle) to compute the remaining leg angles.

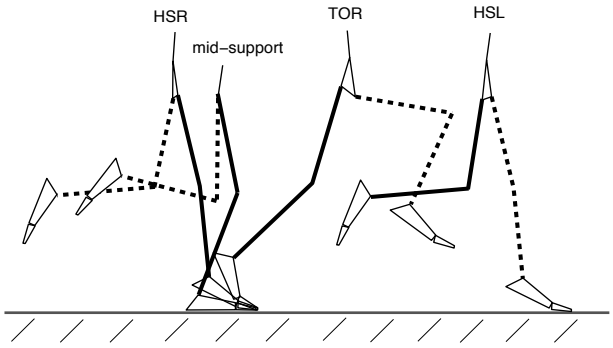

Figure 8: Phase constraints.

Automatic recovery procedures during stance are built in; if the hip is too high so the above intersection fails (which can happen, for instance, if bounciness and flight height are significantly reduced), the control point for the pelvis at mid-support is lowered. On the other hand, if the hip is too low (due to an increased bounciness and flight height), the intersection might push the foot through the ground. In this case, the knee angle is automatically temporarily increased.

The swing leg angles during single support (left, dashed leg in Figure 8) are interpolated in the first subphase between the values at the end of the previous step (HSR) to mid-support, where the thigh is vertical (sagittal hip angle is zero) and the knee is maximally flexed. The next subphase for the swing leg goes until the end of support (TOR) where the hip is maximally extended and the toe is vertically under the knee. Both the maximum knee flexion and hip extension are functions of the current velocity, but can be adjusted via attribute sliders by the user. If part of the foot stubs the ground during swing (which might occur if bounciness is increased or maximum knee flexion during swing is decreased), an automatic recovery procedure inserts a new control point with temporarily increased knee flexion to lift the foot above the ground. Finally, we note that on the initial step when starting a run from a standing position, the first subphase during swing is omitted and its duration is added to the second phase.

3.3.2 Flight

During flight (TOR to HSL in Figure 8), both legs are in their swing phases. The interpolation of the leg angles for the former stance leg (hind leg) are done such that at HSL, the angles are a percentage (p) of the values at mid-support of the next step. The percentage automatically varies depending on whether the figure is currently accelerating, running at a constant speed, or decelerating; values for p ranging between 0.6 and 0.9 have given good results. On the last step before stopping, this subphase is omitted while the previous stance phases for this leg are extended until HSL. The leg angles for the other leg during flight (dashed leg in Figure 8) are interpolated between their values at TOR and their values heel-strike known from the state-constraints.

3.3.3 Upper Body

Several degrees of freedom of the upper body are animated. Torso tilt and sway in the sagittal plane have default values which can be changed via sliders. Tilt automatically increases with velocity, while sway is interpolated between maximum forward lean reached at mid-support and the maximum backward value at toe-off. In order for the head to always point straight ahead, compensation for pelvis rotation and list are performed in the spine. List compensation is distributed over the five lumbar vertebrae, and rotation over the lumbar and the twelve thoracic vertebrae such that below the seventh thoracic vertebra the rotations are towards the pelvis, whereas above it the rotations are counter to the pelvis. The amount of this counter rotation is adjustable by an attribute.

Arm swing is implemented such that on the forward swing it is equal to the sagittal hip rotation of the opposite (swing) leg multiplied by a default factor adjustable via a slider. On the backward swing, instead of the hip angle which increases and decreases during support, the angle between the vertical and the hip-ankle vector of the opposite leg is used which decreases continuously. The amount of elbow flexion during a running step can also be adjusted by the user through a minimum and maximum flexion attribute. Minimum flexion occurs during the backward swing of the arm at toe-off, maximum flexion during forward swing at toe-off.

This concludes the discussion of the running algorithm. Most of the implementation is based on research and observations on human running, and the main contribution of this approach is that it pulls together different knowledge and techniques to provide an interactive environment in which a user can experiment and animate a wide variety of runs in real-time without having to know about the intricacies of limb-coordination during locomotion.

4 Results

A system called RUNNER has been implemented in C++ according to the principles introduced above. An illustration of the interface is given in Figure 9. The program performs in real-time on a Silicon graphics $Indigo^2$ R4000 workstation. For calculations done at 30 frames/sec, this means that an animator can interactively change any of the parameters and attribute sliders while viewing a real-time running stick-figure on the screen. Our model of the human figure has 37 joints and 71 degrees of freedom, and anthropometric data such as body height and relative limb lengths are variable.

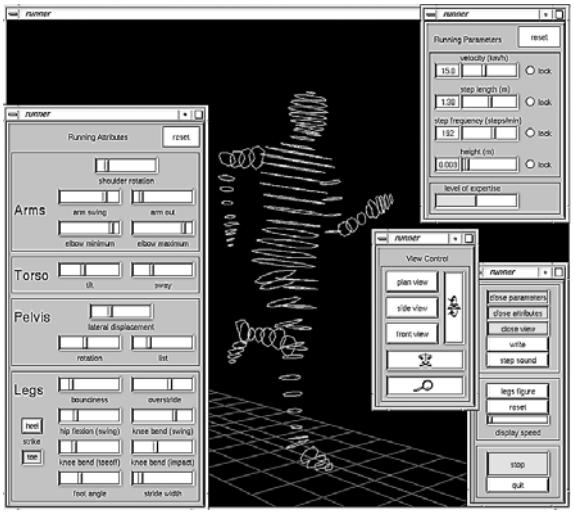

Figure 9: Running interface.

Snapshots of a few sample runs are shown in Figure 10. For example, the top left run was obtained by just reducing the velocity slider to about 5 km/h; the second top run from left was generated by increasing the velocity to about 15 km/h and increasing elbow flexion for both minimum and maximum. The third top run from the left was produced from the settings of the previous run by increasing arm-swing and knee-bend during swing as well as switching from heel-strike to toe-strike. In general, a large number of running styles can be animated. For example, ranges in the parameters from a very slow run at 2 km/h to a fast run at 25 km/h with a step length well over 2 m are possible. Also, moving the attribute sliders to their extreme values results in runs which can look very "stiff" or very "loose".

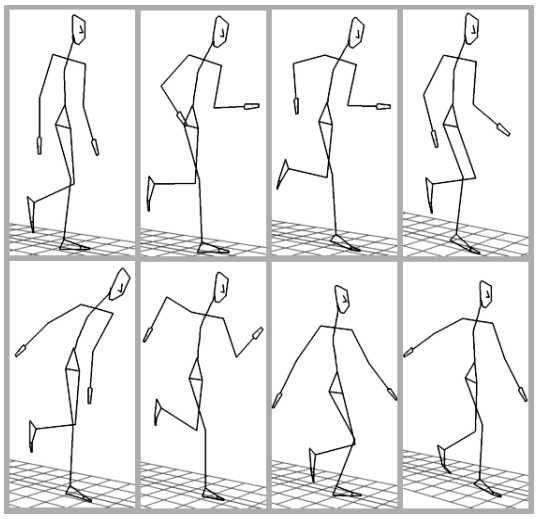

Figure 10: Various sample run snapshots at heel-strike of the right leg.

Figure 11 demonstrates a comparison between a human running on a treadmill and a run generated by our algorithm to match the real run. By setting the body height of our figure to the height of the subject and the velocity to the speed of the treadmill, step length, step frequency and flight height successfully matched with the real run. In addition, the default overstride attribute value was reduced slightly and elbow flexion was increased to closely match the leg and arm movements of the treadmill run.

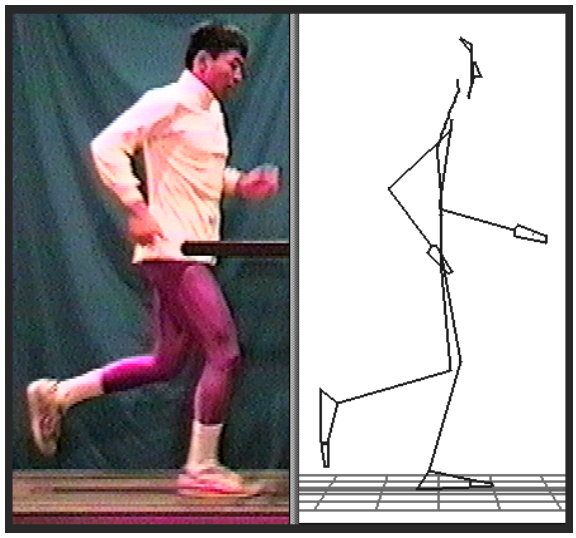

Figure 11: Real treadmill and generated run.

5 Conclusions

A high-level motion control technique has been introduced which allows a user to conveniently create a large variety of human running styles interactively and in real-time. Unlike other high-level techniques, the creative control over the motion remains with the animator. Convincing running animations are achieved by incorporating knowledge on how people run into the algorithm. The approach has proven useful for customizing "real" looking runs or "funny" runs for human-like figures of different sizes and shapes. The results are better than motion-captured data in a sense that true 3-D motion is generated which is easily modifiable on the fly.

We are currently extending the system to running along inclines and arbitrary paths. In this way, the control could be hooked up to devices like joysticks to drive figures in interactive games or virtual environment applications. Also, the keyframes defined by the state and phase constraints can be exported to general-purpose animation systems for rendering or further manipulation.

The initial research introduced here was carried out at Simon Fraser University supported in part by grants from the Natural Sciences and Engineering Research Council of Canada. We are thankful to Dr. Nakatsu and Dr. Mase at ATR MI&C for their continuing support of new developments.

References

[1] Alexander, R. M. The gaits of bipedal and quadrupedal animals. *The International Journal of Robotics Research 3*, 2 (1984), 49–59.

[2] Boulic, R., et al. Human free-walking model for a real-time interactive design of gaits. In *Computer Animation '90, Proceedings* (April 1990), N. Magnenat-Thalmann and D. Thalmann, Eds., Springer-Verlag, pp. 61–79.

[3] Bruderlin, A., and Calvert, T. Goal-directed, dynamic animation of human walking. In *Computer Graphics (SIGGRAPH '89 Proceedings)* (July 1989), vol. 23, pp. 233–242.

[4] Bruderlin, A., and Calvert, T. Interactive animation of personalized human locomotion. In *Graphics Interface '93, Proceedings* (May 1993), pp. 17–23.

[5] Calvert, T., et al. Desktop animation of multiple human figures. *IEEE Computer Graphics & Applications 13*, 3 (1993), 18–26.

[6] Calvert, T., et al. Towards the autonomous animation of multiple human figures. In *Computer Animation '94, Proceedings* (1994), pp. 69–75.

[7] Cassell, J., et al. Animated conversation: Rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 413–420.

[8] Cavagna, G., et al. Mechanical work in running. *journal of Applied Physiology 19* (1964), 249–256.

[9] Cohen, M. Interactive spacetime control for animation. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (July 1992), vol. 26, pp. 293–302.

[10] Dillman, C. Kinematic analysis of running. *Exercise and Sport Sciences Reviews 3* (1975), 193–218.

[11] Fletcher, R. *Practical Methods of Optimization.* John Wiley & Sons, New York, 1987.

[12] Girard, M., and Maciejweski, A. Computational modeling for the computer animation of legged figures. In *Computer Graphics (SIGGRAPH '85 Proceedings)* (1985), vol. 19, pp. 263–270.

[13] Hodgins, J. Simulation of human running. In *IEEE International Conference on Robotics and Automation, Proceedings* (1994), pp. 1320–1325.

[14] Hodgins, J., et al. Animating human athletics. In *Computer Graphics (SIGGRAPH '95 Proceedings, Los Angels, CA, August 6-11)* (1995), pp. 71–78.

[15] Hogberg, P. Length of stride, stride frequency, "flight" period and maximum distance between the feet during running with different speeds. *Arbeitsphysiologie 14* (1952), 431–436.

[16] Hoshikawa, T., et al. Analysis of running pattern in relation to speed. *Medicine and Sport: Biomechanics III 8* (1973), 342–348.

[17] Inman, V., et al. *Human Walking.* Williams & Wilkins, Baltimore, 1981.

[18] Isaacs, P., and Cohen, M. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (1987), vol. 21, pp. 215–224.

[19] Ko, H., and Badler, N. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Graphics Interface '93, Proceedings* (May 1993), pp. 9–16.

[20] Koga, Y., et al. Planning motions with intentions. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 395–408.

[21] Kurakin, M. Relationships among running parameters. *Yessis Review 8*, 1 (1973), 1–4.

[22] Lee, P., et al. Strength guided motion. In *Computer Graphics (SIGGRAPH '90 Proceedings)* (1990), vol. 24, pp. 253–262.

[23] Liu, Z., et al. Hierarchical spacetime control. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), pp. 35–42.

[24] Nilsson, J., et al. Changes in leg movements and muscle activity with speed of locomotion and mode of progression in humans. *Acta Physiologica Scandinavica 123*, 4 (1985), 457–475.

[25] Phillips, C., and Badler, N. Interactive behaviors for bipedal articulated figures. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (1991), vol. 25, pp. 359–362.

[26] Raibert, M., and Hodgins, J. Animation of dynamic legged locomotion. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (1991), vol. 25, pp. 349–358.

[27] Stewart, J., and Cremer, J. Beyond keyframing: An algorithmic approach to animation. In *Graphics Interface '92, Proceedings* (May 1992), pp. 273–281.

[28] Weber, W., and Weber, E. *Mechanics of the Human Walking Apparatus.* Springer-Verlag, Berlin, 1992.

[29] Wilhelms, J. Using dynamic analysis to animate articulated bodies such as humans and robots. In *Graphics Interface '85, Proceedings* (1985), pp. 97–104.

[30] Wilhelms, J. Virya–a motion control editor for kinematic and dynamic animation. In *Graphics Interface '86, Proceedings* (1986), pp. 141–146.