A Desktop Input Device and Interface for Interactive 3D Character Animation

Sageev Oore
Department of Computer Science
University of Toronto

Demetri Terzopoulos
Department of Computer Science
New York University

Geoffrey Hinton
Department of Computer Science
University of Toronto

Abstract

We present a novel input device and interface for interactively controlling the animation of graphical human character from a desktop environment. The trackers are embedded in a new physical design, which is both simple yet also provides significant benefits, and establishes a tangible interface with coordinate frames inherent to the character. A layered kinematic motion recording strategy accesses subsets of the total degrees of freedom of the character. We present the experiences of three novice users with the system, and that of a long-term user who has prior experience with other complex continuous interfaces.

Key words: Interactive character animation, Input device, Motion capture, Expert user interaction, Tangible interfaces

1 Background

Performance animation is the interactive creation of animation whereby the user manipulates an input device to continuously control the motion of a graphical character in real-time, and at the same time is provided with immediate feedback displaying the animation as it is being created [23, 21]. The animator is effectively a puppeteer; the computer graphics character is the puppet; and the mapping defines how the puppet is virtually strung. In principle, mappings can range from the simple triggering of scripted actions, to a continuous, low-level control over the character. It is in low-level control that we are interested, as that does not limit the animator to a specific set of pre-animated motions, and furthermore affords him with the opportunity to provide his own detailed human input.

The difficulty with this kind of control, however, is providing an interface to the very large number of degrees of freedom (DOF) of the graphical output. In fact, it has been claimed that performance animation

> [. . .] is particularly appropriate when the characters to be animated are simple and their range of movement limited [. . .] The great number of DOF that need to be controlled for complex human motion does not make [performance animation] a viable solution for realistic looking animation[19, pg.28].

For this reason, real-time animation of more complex 3D characters is typically done by motion capture [20], where an actor is covered in sensors, and her joints are mapped directly ("literally") onto the corresponding joints of the character. However, this requires a non-trivial post-processing stage to correct for the differences between the body proportions of the actor versus those of the character [6, 11]. In the case of an imaginary creature with a completely different body type, this issue becomes even more difficult or impossible. Furthermore, motion capture requires a costly, elaborate hardware studio setup, limiting its accessibility and making "retakes" inconvenient. Many of the bottlenecks stem from the essential limitation that, although this type of motion capture works in real-time, it is not interactive.

A fundamental characteristic of performance animation that differentiates it from the above approach is its highly interactive nature. The live continuous feedback of performance animation allows *non-literal* mappings, meaning that the motions of the user do not have to mirror those of the character. This non-literal approach has been used in a variety of interesting and creative ways, from having an actor's hands and feet control a cartoon worm character [7], to the impressive interactive control of a dynamic 2-D simulation by mouse and keyboard [16]. Furthermore, contrary to the claim quoted earlier regarding the limitations of performance animation, we contend and demonstrate that by capitalizing on the interactive feedback loop, it is even possible to design a real-time interface for low-level control of complex 3D character animation.

2 Approach

We achieve this within a desktop environment, using less than one tenth the number of sensors typically used for a motion capture session. By combining an appropriate input device design together with a multi-layered motion recording approach, we can use two 6-DOF Polhemus

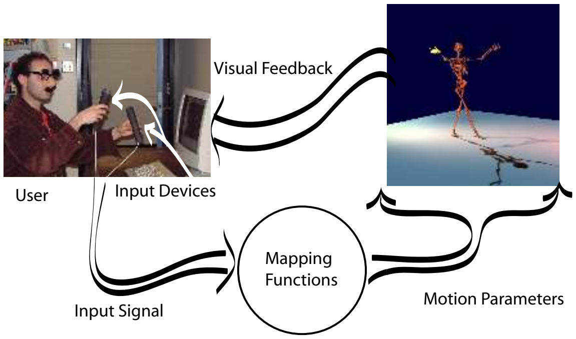

Figure 1: DIGITAL MARIONETTE*: Performance animation is a method of creating animation in real-time: The user manipulates real-time input devices to interactively drive the motion of a computer graphic (CG) character, and is provided with immediate feedback displaying the animation as it is being created. Our animation interface can be operated within a desktop environment.*

motion trackers [22, 18] to provide the user with real-time continuous interface to the 30-DOF joint angle space of an articulated 3D character. As will be described later, the multi-layered approach consists of partitioning the character's degrees of freedom into groups, e.g. left arm, right leg, etc., and designing bimanual mappings from the input device to the group. Most of these mappings are partially symmetric in the sense that they may have the same general task (e.g. during the leg mapping, each hand controls one of the legs), but different specific goals (e.g. each leg will have a different desired motion) [15]. Furthermore, the bimanual task has the advantage of being visually integrated [4] by the simple fact that each leg clearly belongs to the same character. On the other hand, the spine mapping makes use of asymmetric bimanual mappings in accordance with Guiard's Kinematic Chain theory [12] as applied within a computer graphics context [5]. In particular, as will be explained in more detail later, the left hand is used for the joint which is closest to the root of the hierarchy (the lower back), thus providing the reference frame for the joints controlled by the right hand (upper back and neck).

Together, these elements contribute to making our system extremely efficient, allowing an experienced user to create the motion parameters of a 1-minute long character animation in under 10 minutes (see Figure 5).

In designing this interface, two critical issues needed to be solved:

1. development of an effective way of acquiring user input, and
2. conception and specification of a strategy for mapping to the character's joint angles.

In the remainder of this paper we discuss our solutions to these issues, followed by a discussion of the resulting animations and user experience.

3 Input Device Design

The Polhemus sensors are small, oddly-shaped devices, about $2cm$ long, and very lightweight. Each tracker provides a 6-dimensional input vector consisting of positional and orientation information relative to a (fixed) source emitter.

Directly manipulating the translation and orientation of the sensors is awkward, as the small size and shape of the sensors makes them prone to slip, and hard to manipulate accurately over a continuous time window. Also, when the cables get too curled, then their stiffness usually causes the sensors to flip around, slipping on the user's fingers, unless the user grips really tightly. But gripping tightly makes it harder to maneuver fluidly.

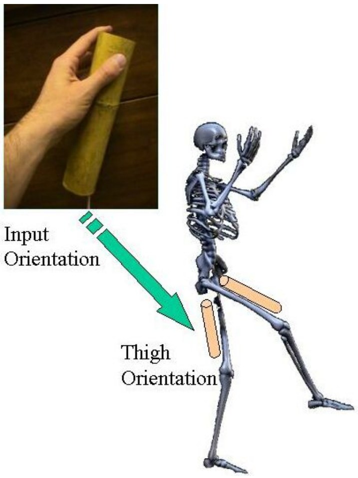

Figure 2: Bamboo Tubes For 3D Animation: The upper photo shows the user holding one of the two input tubes. Although not visible, the tracker is in the top of the tube, near the user's fingers. Below, cylinder icons next to the thigh bones illustrates the mapping from the input to the hip rotation, which keeps the thighs parallel to the tubes themselves.

We therefore redesigned the input device, with respect to both its geometry and physical characteristics, by embedding the trackers in cylindrical bamboo tubes, 15-20cm in length, and about 5cm in diameter. The upper part of Figure 2 shows the user holding onto the bamboo stick input device. This is related to other approaches taken for embedding trackers in outer shells

[1, 10, 13, 14]. However, this new design provided essential advantages in making the system easier to use both accurately and, ultimately, in a more natural manner for the task of interactive character animation, as we now discuss.

3.1 Grip

Our tubes afford the user both *power* and *precision* grips [17, Ch.2][24]. Interestingly, we found that the inexperienced or untrained subjects would naturally hold the interface devices in a power grasp. As the primary test subject became more advanced, the precision grip was used more often, and still quite naturally; the sticks are sufficiently light that they can be held without needing to engage all five fingers. The same device design is well-suited to accommodate both phases, and the bamboo has been described as being very comfortable to hold. Note that adding buttons to the device, as is often done, can make it much harder to maintain a precision grip while moving it through the orientations necessary for the animation task.

3.2 Coordinate Frame: Cues

The cylindrical input device provides visual and tactile cues indicating a reference frame with a “long-axis,” as well as axes of symmetry. These kinesthetic feedback cues [13] establish a tangible interface to coordinate frames inherent to the character. For example, rotating the input tube around its long axis can correspond to rotating a virtual object such as the humerus or thigh bone around its long axis as well, as shown in Figure 2. This modification was very helpful in the development and use of the mappings.

The mass of the bamboo also provides orientation feedback, by making the tubes naturally gravitate towards their neutral vertical orientation. Reorienting the axes so that the cable emerges from the bottom of the tube encourages holding the tube upright, again emphasizing a sense of the neutral orientation, as well as differentiating between the upwards and downwards directions, while visually retaining the existing symmetry along this axis.

3.3 Inertial Resistance and Smoothing

Hinckley points out that input tools with mass provide kinesthetic feedback due to gravity and inertial properties [13, Section 4.6]. This feedback— primarily the feeling that one was actually holding on to real object, rather than just waving one’s hands around in space— did indeed make the input control a far more satisfying experience. The lightness of the cables relative to the weight of the heavier tubes made the cables themselves much less noticeable than before. Furthermore, the inertial properties of such an input device reduces hand jitter.

4 Filtering

Although the mass of the sticks can indeed dampen the physically generated jitter, another source of input noise comes from the sensor readings themselves, e.g. as caused by the presence of electromagnetic devices in the environment. It is therefore still very important to be able to filter the raw input signal in some way. Let x_i represent the value received for one of the input parameters at time step t_i. Then the corresponding filtered value $y_i = f(x_i, x_{i-1}, \ldots, x_0)$ we use is given by

$$y_i = y_{i-1} + (1-\alpha)(y_{i-1} - y_{i-2}) + \alpha(x_i - x_{i-1}). \quad (1)$$

We can interpret Eq (1) as forcing the output velocity to be an average of its previous velocity with the most recent input velocity, thus adding a “viscosity” term to the output motion. This has been found to work quite well for our purposes, though for different animations, a different filter might be more appropriate. Figure 3 shows a sample of raw and filtered input data. Setting a value for α will be discussed in Section 6.6.

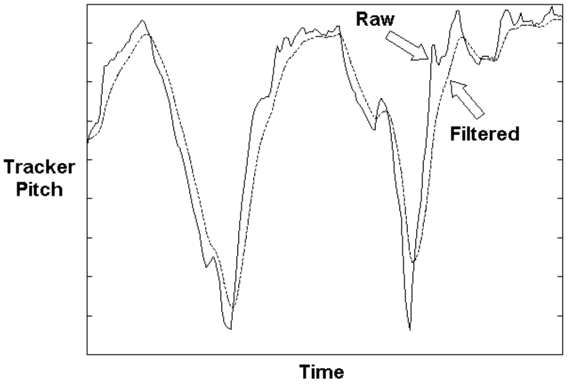

Figure 3: Filtered Input Signal: The solid line shows raw input data of one of the Polhemus tracker parameters. The dotted line shows the corresponding filtered data for $\alpha = 0.8$.

5 Layered Motion Recording

Multi-tracking— the recording and re-recording of various instruments in separate tracks— has been used in the music recording industry for many years. Applying this strategy to the animation problem, we control the articulated character in multiple layers, each layer controlling a subset of the output DOF.

We thus begin by subdividing the DOF of our character into layers as shown in Figure 4.

The multiple layers need to be accessible, to allow coordination of the playing and recording of multiple

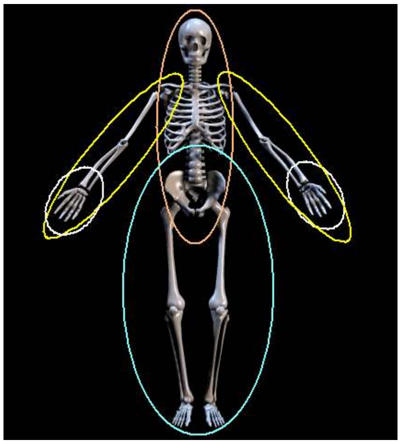

Figure 4: Kinematic Layering Strategy: Legs are usually recorded first, since they define the motion of the character's root. Spine and head are usually recorded simultaneously, followed by arms.

tracks. To create a functional recording studio environment for the animated motions, we therefore implemented modular components such as "Channel", "Mapping", and "Input Device", along with corresponding graphical user interfaces. Each channel includes its own clocking mechanism, and the user interface enables synchronization between clocks. This is crucial for layering motions in real time.

6 Kinematic Mappings

6.1 The Animated Character Output

The CG puppet we are controlling is a rigid articulated body consisting of a set of links connected by joints for a total of 33 controllable DOF, as summarized in Table 1.

Joint	D.O.F.	Child Link
Root	6	pelvis
Lower Back (L1)	3	back/torso
Lower Neck (C7)	3	neck
Head Nod (C1)	1	head
Left, Right Shoulders	3 each	upper arm
Left, Right Elbows	1 each	forearm
Left, Right Wrists	1 each	hand
Left, Right Hips	3 each	thigh
Left, Right Knees	1 each	lower leg
Left, Right Ankles	1 each	foot

Table 1: Output Degrees of Freedom

6.2 Legs

The hips, knees and ankles are all recorded simultaneously, and the mappings for each have been designed to facilitate this. The orientation of the graspable input tube is mapped onto that of the thigh bones, so that there is a direct correspondence between the tube and thigh orientation (as previously illustrated in Figure 2). That is, when the tube is held vertically, the thigh should be vertical as well, and similarly for rotations. The tracker height determines the orientation of the lower legs, or shins, relative to the world coordinate system. This is initially done by a linear mapping, and later modulated by a physics-based model (as will be described below). The tracker's $\mathbf{z}$-translation is mapped to control the flexion at the ankle. Ankle and knee motion is also influenced by physics-based filters, which are beyond the current scope of discussion, but described in detail elsewhere[21].

6.3 Arms

The arms can be controlled similarly to the legs. That is, the tube containing the tracker is mapped to control the orientation of the humerus by rotating the shoulder. The height of the tracker controls the bend at the elbow, and the tracker's $\mathbf{z}$-translation controls the flexion and extension of the wrists.

6.4 Spine

The spine is currently modeled by a 3 DOF joint at the lower back (vertebrae L5) which is attached to the root of the model, another 3 DOF joint at the upper back (vertebrae C7), and a hinge joint for the head (at C1). The ball and socket joints are controlled analogously to the hip control, while the head nod is controlled using a linear relationship as for the ankles and wrists. The left hand controls the lower back joint, nearest the root of the chain, while the right hand controls the upper back and head. Future models will use the same approach but allow a more flexible spine.

6.5 Root Motion and Ground Contact

A critical issue in achieving satisfying control of the puppet is keeping it grounded. This is also the basis for locomotion. Since the trackers translate freely in 3D space, it is virtually impossible to control the height of the pelvis directly without betraying the lack of any underlying ground constraint.

We solve this by imposing the constraint that one of the puppet's feet is touching the ground. Thus, as the character rotates at the hip, one foot is constrained to stay at a fixed position relative to the floor, hence becoming the center of rotation. Nailing a single foot to the floor will not let the character go very far, so a mechanism is provided for switching feet as the next foot touches the ground. By virtue of this ground contact, the puppet can be made to locomote in any direction. Our method can generalize to more contact points, and multiple ground levels of an uneven terrain. We currently use two contact points per foot— one at each of the heels and balls of the feet.

6.6 Setting Filter Viscosity Values

Adjusting the value of α between 0 and 1 in Eq (1) controls the smoothness of the input signal y_i (and therefore also of the resulting motion which depends directly on the input, as will be described in more detail in the next section). If α is too high (e.g. $\alpha = 1$ in the extreme case), then the input signal is essentially unfiltered, and the noise is very apparent in the animation. If α is too low, then the responsiveness is compromised, and significant lag is introduced as well. In the extreme case, for example, when $\alpha = 0$, the input value does not affect the output at all.

This leaves us with a range of possible values, with which we can control the smoothness quality of the resulting motion, to advantageous effect. For example, setting a high smoothness for the hip gives the character a certain appealing quality, related to what many viewers described as "very graceful motion". The local attentuation of higher frequency components can enhance what is seen as part of the character's "style" of motion. In contrast, for other joints such as head rotation, it is effective to use a much lower-smoothness filter, allowing more responsiveness to jerky motions such as quickly glancing over his shoulder.

The choice of a particular filter value helps emphasize corresponding qualities of the character's motion. This fact led us to try giving the user real-time control over the filter parameter α itself. Although doing so produced some interesting results, it was quite difficult to learn to control the interaction between this and the rest of the animated parameters (i.e. direct control of joint angle values), certainly when attempting to achieving realistically-based motion. In particular, changing the filter response in real-time can lead to highly non-linear effects, analogous to playing with delay, gain and various other parameters on effects pedals during a musical performance.

7 Results

A one-minute-long demonstration animation, "Digital Marionette Dance", was created by a fairly experienced user, having the character dancing alone to some music. Sample frames from the animation are shown in Figure 5. All of the animated parameters of this sequence were created at the desktop environment, using our interface, in a total of under 10 minutes, with no re-takes necessary. This is extremely efficient compared to traditional animation techniques. Images from another animation are shown in Figure 6

The expressiveness of the animation created with our system has been demonstrated in a wide variety of contexts, ranging from live television[2] to live theatre[3]. For the theatrical performance, the CG skeleton character was projected onto a very large movie-size screen. All tracks of the animation were created in front of the audience, together with live musical and vocal accompaniment when the fully layered animation was being played back. Projecting the character onto such a large screen involved the risk of finding and magnifying any weaknesses— perhaps subtle or otherwise unnoticeable in the screen-sized version— in his motion. However, the effect was quite strong, and audience and producer response was very positive.

8 User Learning & Experience

Like many complex control tasks, from playing a musical instrument to operating a physical marionette controller, this system, too, is designed with experienced users in mind; the goal is not to have an interface which makes each user an "instant animator", but a system which allows trained animators to create nearly instant animations. Hence, our discussion is based primarily on the observations of, and by, an experienced subject S—, as well those of some novice users[1].

S—'s learning experience involved three primary aspects: basic control tasks, refinement of complex motions, and general principles, including the importance of the overlap itself between the different learning stages.

8.1 Basic Control Tasks at the Initial Stage

The main motion chosen to be learned was walking, since it is a complex yet familiar motion, and leaves room for expressiveness. However, achieving a good, expressive walk right away was not possible, so initial simplifications and sub-tasks were necessary, leading to two main types of initial exercises: isolated motions and functional bipedal locomotion, which we discuss below.

Isolated Motions

Examples of isolation exercises included swinging one leg back and forth from the hip, and shifting support from one leg to the other. By isolating an output parameter, the user observed the specific combination, or *synergy*, of motions in his own body that lead to this constrained output, and thus focused his awareness on the relationship between his kinesthetic experience and the resulting effects.

---

[1] S— is also a musician, as well as being a user of the GLOVETALKII system [8, 9], and thus has extensive prior familiarity learning complex, continuous interfaces. In this light, there are some interesting parallels between this learning experience with that of GLOVETALKII and musical instruments, and these are discussed in considerable detail in [21]. S— was also the designer of the current system.

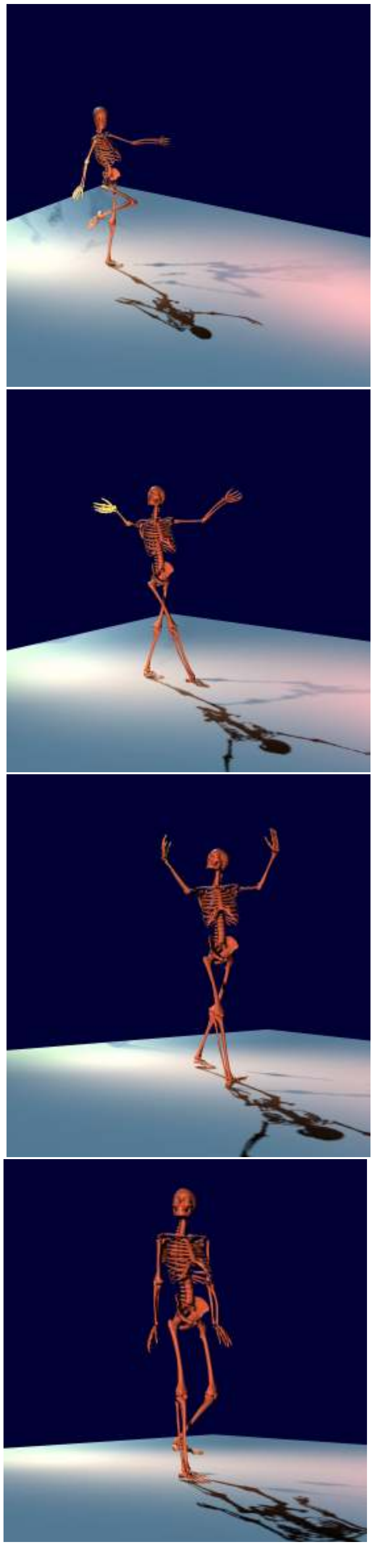

Figure 5: Digital Marionette Dance... Still Frames: These images were taken from an interactively-generated dance animation sequence created using our interface. All animated character parameters of the one-minute long sequence were generated in a total of 10 minutes at a desktop environment. The camera positions were set using a conventional mouse-based interface.

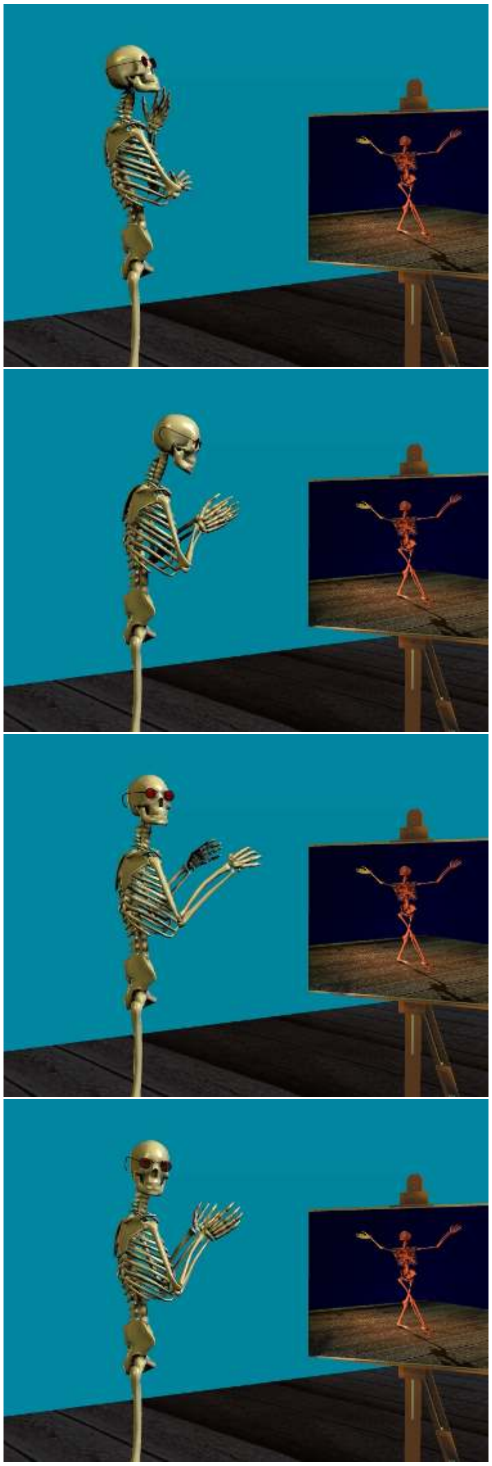

Figure 6: The Appreciator... Still Frames: Images taken from a short animation in which the character expresses his enjoyment of a painting. The frames shown here and in the previous sequence were sampled about one to four seconds apart (going down the columns), highlighting some of the postures achieved by the character.

Functional Bipedal Locomotion

"Functional locomotion" means that initially, the forward motion did not have to be graceful or even resemble conventional locomotion. Having first isolated concepts such as switching feet and swinging a leg, the user could now focus on getting the proper synchronization between them to propel forward. This often began as a "waddle", keeping the character's knees straight and just swinging his free leg out and around.

8.2 Refining Complex Motion: Walking

The exercise for functional locomotion was next refined in various ways to become a walking motion. Once the control of basic bipedal locomotion was "internalized" from the user's perspective (i.e. once the user became comfortable with the necessary coordination patterns), then most corrections and refinements were perceived as isolated details layered on top of a nearly automatic motion. The refinement tasks included aspects such as: controlling the arc of the swing; finding the right combination of "falling forward" and swinging the free leg; and velocity control.

8.3 Novice User Experience

Three novice users practised the system for about two hours each, over two or three separate sessions of 30-60 minutes in length. One of these users had minimal previous experience with animation, while the other two did not. None of them had any previous puppeteering experience. The format of these sessions consisted primarily of the users directly manipulating the CG character, occasionally commenting on the control, with interspersed short breaks of 5-15 minutes each. Due to the complexity of the task, it was essential for the sessions to be exploratory, and therefore structured more like *lessons* than *tests*[2].

Nevertheless, all three users were indeed able to achieve some form of basic bipedal locomotion within the first 20-30 minutes of practise, although after this brief practise time it was still quite inconsistent, and did not yet look like a realistic walk. By the end of the two hour period, all three test subjects had accomplished at least a couple of short walking sequences which, although not as refined as those achieved by the more experienced user, were clearly beyond the initial stages of "waddling". Some of these walking sequences showed early signs of refinement such as the leg swinging through with some smoothness.

9 Conclusion

We achieved our primary goal of providing an efficient, powerful and satisfying interface for expressive character animation. In doing so, we have brought the power of the performance animation tool to the desktop environment.

One of the challenges in achieving this was indeed handling the large number of DOF of the output. By embedding the trackers in a physical tubing, we established a tangible interface with various coordinate frames inherent to the character, providing a valuable synergy between the physical input device and the underlying transformations from input to motion. Building on this compatibility, we constructed a multi-track motion recording framework and a feasible set of layered mappings from the two input trackers to the thirty three DOF of the graphics character.

By demonstrating an input and interface solution for continuous, low-level control over a dancing 3D articulated character, we have provided a foundation and reference point for numerous future directions. One such direction, in particular, is the exploration of motion layers comprised of greater numbers of parameters and more complex mapping functions, including layers sharing parameters. This would allow application of the system to increasingly refined character models with greater numbers of degrees of freedom, while maintaining a desktop usability, real-time efficiency, and interactively-generated expressiveness. It will be interesting to additionally apply our 3D interface for the control of camera motion.

One limitation of the system is in the ground contact model, wherein at least one foot is making contact with the ground. A more flexible model is being developed. Also, due to the nature of non-literal kinematic-based control, it is relatively easy to allow the character to recover from unstable positions (e.g. including positions in which he would be expected to fall down in a dynamic environment). However, this can occasionally look non-realistic; adding appropriate constraints to the system is an interesting area of research.

Finally, another aspect of future work is to develop systematic evaluation strategies for the task at hand. We thereby intend to provide comparisons to other animation tools, and make explicit the suitability of different approaches for different tasks. Such an approach will also allow for rigorous comparisons between different mappings.

Acknowledgements

We thank to Chakra Chennubhotla for numerous helpful discussions and feedback. We thank Petros Faloutsos and Victor Ng for providing the DANCE platform and support. We thank the referees for their helpful comments

---

[2] Although informal, a more formal approach would be analogous to (and similarly difficult as) attempting to evaluate the quality of "the piano" as a musical instrument based on the first few times some beginners sit down to try it out.

and suggestions.

References

[1] Wand/Wanda (TM) VR Input Devices. Input Device, http://evlweb.eecs.uic.edu/research/.

[2] Breakfast Television on City TV. Television program, May 2001.

[3] Subtle Technologies (Toronto, Canada). Conference and Artshow, May 2001.

[4] R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'2000)*, pages 33–40, New York, 2000. ACM.

[5] Ravin Balakrishnan. *Issues in Bimanual Interaction for Computer Graphics*. PhD thesis, University of Toronto, 2001.

[6] B. Bodenheimer, C. Rose, S. Rosenthal, and J. Pella. The process of motion capture: Dealing with the data. In D. Thalmann and M. van de Panne, editors, *Computer Animation and Simulation '97 : Proceedings of the Eurographics Workshop in Budapest, Hungary*, Springer Computer Science, pages 3–18, NY, September 1997. Springer.

[7] B. deGraf. Protozoa. Company.

[8] Sid Fels. *Glove-TalkII: Mapping Hand Gestures to Speech Using Neural Networks – An Approach to Building Adaptive Interfaces*. PhD thesis, University of Toronto, 1994.

[9] Sid Fels and Geoffrey Hinton. Glove-talkII: An adaptive gesture-to-format interface. In *Proceedings of CHI'95 Human Factors in Computing Systems*, pages 456–463. ACM Press, 1995.

[10] George W. Fitzmaurice. *Graspable User Interfaces*. PhD thesis, University of Toronto, 1996.

[11] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of SIGGRAPH 98*, pages 33–42. ACM SIGGRAPH, 1998.

[12] Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behaviour*, 19(4):486–517, 1987.

[13] Ken Hinckley. *Haptic Issues for Virtual Manipulation*. PhD thesis, University of Virginia, 1996.

[14] Intersense. Is-900 precision motion tracker. www.isense.com.

[15] S. Kelso, D. Southard, and D. Goodman. On the coordination of two-handed movements. *Journal of Experimental Psychology: Human Perception and Performance*, 5(2):229–238, 1979.

[16] Joseph F. Laszlo, M. van de Panne, and E. Fiume. Interactive control for physically-based animation. In *Proceedings of SIGGRAPH 2000*. ACM SIGGRAPH, 2000.

[17] C. Mackenzie and T. Iberall. *The Grasping Hand*, volume 104 of *Advances in Psychology*. Amsterdam, 1994.

[18] I. Scott MacKenzie. Input devices and interaction techniques for advanced computing. In W. Barfield and T.A. III Furness, editors, *Virtual Environments and Advanced Interface Design*, pages 437–470. Oxford University Press, New York, 1995.

[19] Roberto Maiocchi. 3-D Character Animation Using Motion Capture. In Nadia Magnenat Thalmann and Daniel Thalmann, editors, *Interactive Computer Animation*. Prentice Hall Europe, London, 1996.

[20] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann, 1999.

[21] Sageev Oore. *Digital Marionette: Augmenting Kinematics with Physics for Multi-Track Desktop Performance Animation*. PhD thesis, University of Toronto, Toronto, Canada, 2001.

[22] Polhemus. `www.polhemus.com`.

[23] David J. Sturman. Computer puppetry. *IEEE Computer Graphics and Applications*, 18(1):38–45, January/February 1998.

[24] S. Zhai, P. Milgram, and W. Buxton. The effects of using fine muscle groups in multiple degree-of-freedom input. In *Proc. ACM CHI'96*, pages 308–315, 1996.