4 From pen gesture to expressive avatar gesture

The user gains expressive power with our interaction technique because they can modulate the motion of an avatar gesture along multiple dimensions. In contrast, previous techniques allow users to alter gesture along only a single dimension such as magnitude or speed. These techniques are limited by a user interface design that encumbers the addition more parameters. We solved this problem by moving away from the traditional graphical user interface and using a pen interface. Style features in the user's handwritten pen gesture are measured and then mapped to motion modulation parameters.

We produce the modulated expressive gestural motion by interpolation from a set of prototype avatar gesture motions. The prototype motions depict the same avatar gesture but with different expressive movement. Our interpolation scheme synthesizes an avatar gesture with the exact modulation specified by the user. This chapter describes how modulation parameters are derived from handwriting style and applied to gesture synthesis.

4.1 Mapping handwriting to animation modulation

There are two main parts to this technique: mapping handwriting features to motion modulation parameters and generating the modulated avatar gesture motion. The primary design issues are the selection of the pen gesture features to extract, the choice of avatar gesture animation parameters to control and the design of an appropriate mapping between the two. Once the animation parameters are selected, a final consideration is the choice of a suitable technique to produce the animation.

4.1.1 Style feature selection

The criteria for selecting pen gesture style features include its quantifiability and controllability. Because we want to produce a continuous range of expressive variation, the style features should be continuously valued. For this reason, we choose quantitative rather than qualitative features. In addition, they must be computable from digital ink.

That certain features of writing are more difficult to control than others is part of the basis of handwriting forensics and expert document examination, fields in which experts analyze handwriting to determine the authenticity of a document's authorship. For pen gesture to be a viable control technique, the style features should be drawn from features that are most controllable. Some of these features are those that people modify in their day-to-day lives. Fitting a signature inside a box requires controlling size, and filling out carbon copy forms requires adjusting writing pressure.

Another way to select style features is to choose those that most people consciously change to express particular emotions or ideas. Some psychological evidence suggests that commonalities exist in the ways that people consciously express themselves through the graphic aspects of writing. In one experiment[114], subjects were asked to write the letters "g" and "f" in the manner of an aggressive person, a submissive person, and an unstable person. Then the shuffled letters were presented to another subject who was asked to match the letters to the type of personality they were meant to express. The results indicate that a commonly accepted expression of these concepts exists. Other studies have found that certain emotional adjectives are represented by characteristic lines or forms [62][70]. For instance, "sad" is usually represented by lines pointing downward. Artists have used the relative spatial placement of words on a page as a form of imagistic expression as in the poems of e. e. cummings, or the rat's tail passage in *Alice in Wonderland*. The images and shapes created by the arrangement of the words contributes as much meaning to these works as the words themselves.

4.1.2 Avatar gesture modulation parameters

The selection of avatar gesture modulation parameters depends on the communicative power of the motion quality being modulated. For instance, size is a powerful parameter because it can be used in many ways. If the gesture is an illustrator, the size of the gesture might indicate the relative size of the "picture" being drawn. If the movement of the gesture reflects emotional state, the size is used to manage the noticeability of the gesture and intensity of the emotion.

4.1.3 Finding the map

The last problem is to determine a map between the handwriting features and the expression parameters. Ideally the mapping between the two is *natural* in some sense. We want the features extracted from the pen gestures to correspond in some meaningful and intuitive way with the gesture movement qualities they control. A degree of naturalness arises from a degree of congruity between the pen movement and the gesture movement. For instance, writing faster should probably result in gesturing faster. Here we present an ad hoc map. Through experience with using this map, we hope to gain insights into better methods for designing this map.

4.2 A design using physical features

We implemented a design for the system using a physical features map. In this system, two physical writing features are mapped to two physical motion parameters. The pen gesture style features are *size* and *speed*. We chose these particular features because they have intuitive meanings for writing, and they are readily controllable. By size, we refer to a measure of the area taken up by the written letter. Speed refers to the speed of the pen tip relative to the writing surface. When the letter is written, values for these style features are computed and then scaled to a value between *0* and *1*.

4.2.1 Calibration issues

Since the size of the letters and the writing speed depend on the individual person and the particular letter being written, it is best to measure these values relative to the user's personal style. We can do this by performing a calibration for each user. The user is asked to write each of the letters quickly, slowly, large and small, and with their usual speed and size. Since the size of a person's writing is partially determined by the amount of space they are given [3], the user writes into the same size window they will use when operating the system online.

4.2.2 Writing features

In our initial implementation we use the *duration* of writing as the measurement of speed. The time, measured in milliseconds, is noted from the time the pen is put down to the time it is lifted. As we used this system we found that this measurement corresponded well with the intuitive meaning of writing speed. The value for this feature was then normalized relative to the writing speed of a particular user in our lab.

For the *size* feature, we measure the bounding box area of the letter and normalize this over some range of sizes. Unlike speed, the design of the interface can determine the range of values for size. The size of the pen window constrains the maximum size for the letter. The minimum size for the letter can be calibrated for the user, but it can also be determined according to the minimum resolution of the character recognition software.

4.2.3 Motion parameters

The important aspect of motion parameters is that they make sense with respect to a particular gesture. This is because we define the space of variation using recordings of extremal variations. In our initial experiments, we controlled the speed and size of the avatar gesture motion. Intuitively, speed correlates with the duration of the movement and size with amount of space taken up by the motion.

4.2.4 Direct physical map

Our implementation maps the physical features of the writing to the analogous features in the gesture. That is, increasing the size of the letter increases the size of the gesture and increasing the speed of writing increases the speed of the movement. We apply the normalized values of the writing features as the values for their respective interpolation parameters.

4.3 Synthesizing expressive gesture motion

Each gesture in the avatar gesture library is constructed from a set of sample motions that were recorded using a performance capture system. We built our own motion capture system using the Flock of Birds sensors from Ascension Corporation [4]. The samples record the same gesture but with different expressive motions. Few people are adept at consciously controlling and modifying gestures, producing them on demand and reproducing the same movement consistently. Those who are adept are called actors (or dancers.) We enlisted the help of an actor/dancer to help in the design and recording of the gesture. The actor actively aided in the design of the movement in addition to letting us stick sensors on them. Our experiences recording gestures is described in Appendix B.

At runtime, new gesture motions within the expression space are generated by multilinear interpolation. Producing new motions from existing animations is a technique that has been studied extensively in computer graphics. Applications of linear interpolation over a set of example motions to produce new motions are described in [44][113]. A technique for multivariable interpolation of motions that are parameterized by emotional expressiveness is described in [88].

4.3.1 Motion samples

Each sample in an avatar gesture is annotated with its expression parameter values and its time duration. The samples are chosen so that they lie on the extremal vertices of the expression space. If the space has n dimensions, then 2^n samples are required for each gesture. Say the space has three dimensions: size, speed and forcefulness. Every motion sample lies at the coordinates given by the values of its parameters. Let a particular sample be denoted by $S(p, q, r)$, where p, q, and r are the values of the parameters and these values vary from 0 to 1. Then our sample set consists of the eight variations

$$S(0,0,0)$$
$$S(0,0,1)$$
$$S(0,1,0)$$
$$\ldots$$
$$S(1,1,1)$$

and each sample is made up of its joint trajectories, parameter values and timing data.

4.3.2 Motion data

A joint trajectory is an array of angle rotations over time. If a gesture motion involves three joints, then each joint trajectory is of length three. That is, for trajectory *J(t)* and joint angles θ_1, θ_2 and θ_3

$$J(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \end{bmatrix}$$

where each $\theta_i(t)$ is a time indexed array of rotation angles. The rotation values are sampled at regular time intervals during the duration of the gesture, though this constraint is not a requirement of the technique. The main constraint is that the samples correctly capture the structure of the motion including all of the inflection points.

The animation samples are recorded using Flock of Birds magnetic position/orientation trackers (from Ascension Technologies [4]) at a rate of 50Hz. The trackers are placed on the segments of the limbs involved in the gesture. In the system we built, we use only the orientation data from the trackers. After recording, the raw orientation data is converted to joint angle data represented by quaternions. Finally, joint angle data is smoothed and aligned by hand. They are aligned so that the gesture motion begins at the same time for all of the samples.

4.3.3 Multilinear interpolation

Given values for the expression parameters, we compute the new motions using multilinear interpolation. With a single parameter, the linear interpolation of two samples *S(0)* and *S(1)* by interpolation value a to produce *S(a)* is straightforward. The interpolation is shown schematically in Figure 4-1. For each pair of corresponding joint trajectories in *S(0)* and *S(1)*,

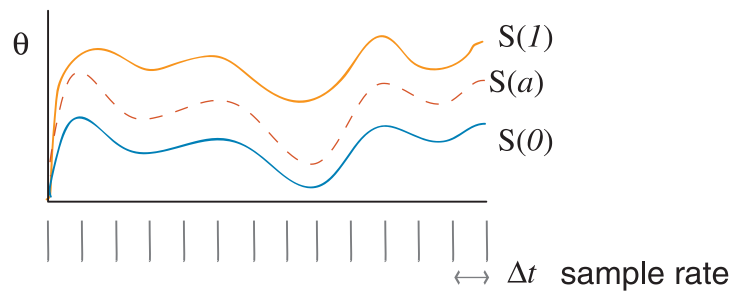

Figure 4-1. Interpolation between two trajectories by parameter value a.

we sample the trajectories at regular intervals, and perform a pairwise interpolation of the sampled angle values. The result is a new interpolated trajectory.

Interpolation between joint angles is accomplished using spherical linear interpolation. This interpolation finds points along the great circle between two quaternions on the unit hypersphere as shown in Figure 4-2. Interpolation along the surface of a sphere allows us to

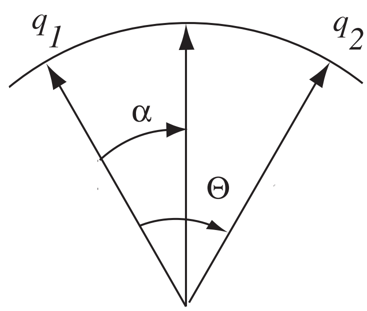

Figure 4-2. Spherical interpolation.

interpolate the angles without speeding up. In contrast, interpolating between two quaternions along the straight line that cuts through the sphere would speed up the rotation in the middle.

We use the *slerp* (**s**pherical **l**inear int**erp**olation) formula to compute the interpolation [98]. Given two unit quaternions q_1 and q_2, and an interpolation parameter α which varies from *0* to *1*, the formula is given by:

$$\mathrm{slerp}(q_1, q_2, \alpha) = q_1\frac{\sin((1-\alpha)\Theta)}{\sin\Theta} + q_2\frac{\sin(\alpha\Theta)}{\sin\Theta}$$

Note that this formula keeps the rotation smooth between two points, but that rotation changes along multiple segments may change direction abruptly. For our purposes, this formula works adequately because our key frames are fairly close together. Higher order continuity can be preserved by finding splines between successive quaternions. [98] proposes the use of Bézier curves for smooth interpolation along a series of key frames.

In multilinear interpolation we interpolate along only one parameter at a time while holding the other parameters constant. Then we recursively interpolate along each successive parameter. We can most easily illustrate this procedure for the case of two parameters p_1 and p_2. This procedure is illustrated in Figure 4-3.We wish to produce a new motion for $p_1 = p$ and

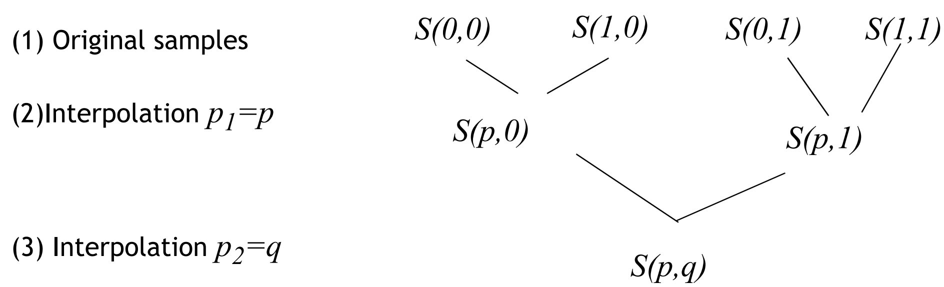

Figure 4-3. Multilinear interpolation for the case of two parameters

$p_2 = q$ using the sample motions *S(0,0), S(0,1), S(1,0), S(1,1)*. In the first round of interpolations we hold p_2 constant and interpolate along p_1. The interpolation between *S(0,0)* and *S(1,0)* yields the intermediate result *S(p,0)*. Interpolating between the other pair, *S(0,1)* and *S(1,1)*, results in the motion *S(p,1)*. Then we interpolate between the intermediate motions *S(p,0)* and *S(p,1)* by the parameter p_2=q to arrive at the desired motion *S(p,q)*.This same procedure can be applied to any number of parameters.

4.3.4 Gesture speed

The interpolation is more complex than this because not all gesture instances will have the same duration. A faster version of a gesture will require fewer time steps to complete than a slower version of the same gesture, as shown in Figure 4-2. Before interpolating, we take the

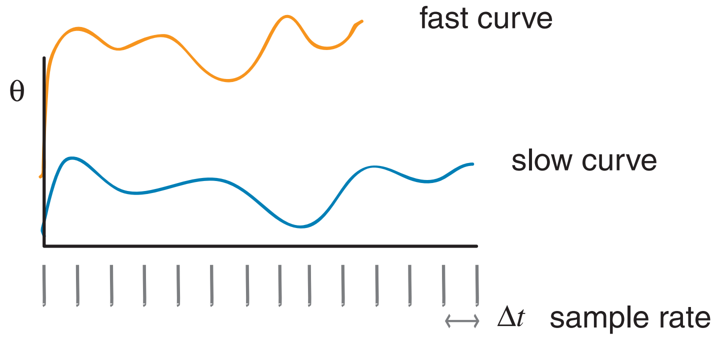

Figure 4-3. Fast and slow curves of different durations.

extra step of time warping the prototype curves. This is done by resampling the prototype curves in such a way that we can realign them in order to perform the interpolation.

The first step is to determine the duration of the interpolated curve. Suppose the current interpolation parameter has value α. Let d_0 be the duration of the slower speed and d_1 be the duration of the faster. Then the duration d of the interpolated curve is

$$d = \alpha d_1 + (1 - \alpha) d_0$$

To compress the slower curve to the new duration, we resample the curve with a new sample rate that is proportional to the base sample rate. The equation for the new sample rate is given by the equation.

$$\Delta t' = \frac{d}{d_0} \Delta t$$

where Δt is a sample rate on the prototype trajectories. Similarly, the faster curve is expanded by a new sampling rate.

$$\Delta t'' = \frac{d}{d_1} \Delta t$$

These new sample points are aligned and linear interpolation takes place as usual as shown in Figure 4-4.

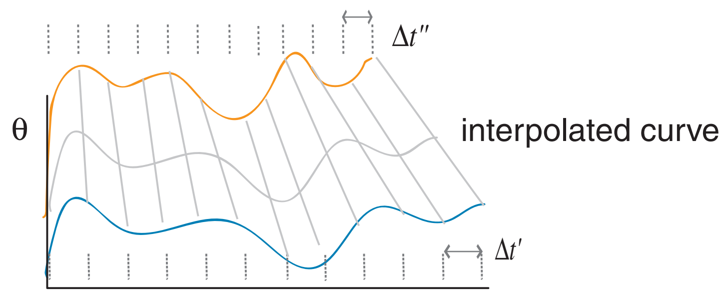

Figure 4-4. Time dilation for trajectories of different durations.

4.4 Evaluation

Using our implementation in the lab we are able to provide a preliminary evaluation of the feasibility and possibilities of our technique. We found that modulating either the speed or the size of the gesture individually is very straightforward. Modulating the two together is a little more difficult since writing a letter smaller often results in writing faster and vice versa. With additional practice and appropriate tuning of the modulation parameters, simultaneous control of the two becomes easier.

Similarly, we found that the ease of talking while controlling the avatar depended on the amount of practice one had with using the system. Initial use required concentrating a lot on the feedback offered by watching the avatar gesture. It is likely some visual feedback will always be necessary though not necessarily by fully rendering the whole avatar. With practice it became easier to drive the gestures while talking. The experience may be likened to talking while driving a car: it can be done but it depends on the complexity of the conversation and the conditions of the road.

During demonstrations we found that the expressions were readily understood by the audience. A few people questioned whether simultaneously specifying the symbolic and expressive aspects of gesture was desirable. An alternative is to choose the gesture with one action, and then to invoke and modulate the gesture with a second action. We agree that determining the desirability of simultaneous control needs to be further investigated.

4.5 Discussion

We have presented a framework for modulating avatar gesture motion using the values of pen gesture style features. The motion can be modulated within a multidimensional expression space. The framework requires a suitable mapping from handwriting style features to the motion modulation parameters, and then a technique for producing the modulated animation. One implementation that we describe maps the size and speed of handwriting to the size and speed of the avatar gesture.

Though we found that our implementation worked well, there are many possible mappings that remain unexplored and that may yield more intuitive or richer expressiveness. Besides speed and size, there are other handwriting features that have continuous values and can be readily computed. For instance, we can also measure the pen point pressure or the writing slant.

In our map, we directly mapped the values of the handwriting features to the values for the modulation parameters. Although this worked fairly well, it ignored the fact that speed and size are not independent factors. Especially if we extract more style features from the pen gesture, we may find that a better relationship between the handwriting features and gesture motion parameters exists. Ultimately, user-testing will be required to validate these designs.

Our framework proposes synthesizing new motions by interpolating from a set of motion captured avatar gesture samples. By using interpolation, a small number of gesture samples can be used to produce new gesture motions with a continuous range of expressive variation. This makes avatar gesture more expressive than if the same gesture motion were played back each time the gesture is invoked.

Using multi-linear interpolation requires $O(2^n)$ samples for an expression space with n dimensions. However, there are other interpolation techniques that require fewer samples. Because motion capture is a time consuming process, adopting one of these other techniques may be a better design alternative. Another limitation of using linear interpolation is that gesture motion variations must be relatively similar in shape. Too much difference in shape may produce unexpected motion. However, one can imagine that a meaningful expressive variation may change a gesture's structure. For instance, a periodic motion such as waving may have a variation in which the frequency of the periodic movement changes, resulting in a structural variation with more inflection points. In these cases, an alternate representation for the animation data may be more appropriate.