An Interface Test-Bed for 'Kansei' Filters Using the Touch Designer Visual Programming Environment.

Rodney Berry, Masahide Naemura, Yuichi Kobayashi, Masahiro Tada, Naomi Inoue, Yusuf Pisan*, Ernest Edmonds*

ATR Media Information Science Laboratories
2-2-2 Hikaridai, 'Keihanna Science City',
Kyoto 619-0288, JAPAN
*Creativity and Cognition Studios, UTS, Sydney Australia

rodney@atr.jp

Abstract

In the context of a larger project dealing with *kansei* analysis of movement, we present a basic method for applying real-time filters to human motion capture data in order to modify the perceived emotional affect of the movement. By employing a commercial realtime 3D package, we have been able to quickly prototype some interfaces to an as yet non-existent system. Filters are represented as physical objects whose proximity to an animated dancing human figure determine how much they modify the movement.

Keywords: augmented reality, kansei engineering, rapid prototyping, motion capture.

1 Introduction

The purpose of this paper is to introduce a test-bed we are using to prototype interface ideas for future character animation systems, informed by analysis of human movement. We have used Touch Designer, a commercial realtime 3D modeling and animation tool, to quickly build some test systems, and suggest that its usefulness to others

This work is part of an on-going research project dealing with the visual analysis of human movement using computers. Humans are able to discern from subtle cues, something of another person's emotional state and intentions. Artists use these cues to convey emotional nuance or to awaken emotional states in their audience. In the Japanese language, the word *kansei* refers both to a person's sensitivity to nuance in others, and to an artists sensitivity in communicating this nuance to other people. Research that attempts to achieve this kind of sensitivity using technology has given birth to the field known as *kansei engineering.*

According to the Japan Society of Kansei Engineering web page (JSKE):

"…KANSEI is the integrated functions of the mind, and various functions exist in during receiving and sending. Filtering, acquiring information, estimating, recognizing, modeling, making relationship, producing, giving information, presenting and etc. are the contents of KANSEI."

2 Kansei in Movement

In our laboratory, the focus of our group is on two aspects of *kansei* in human movement. The first is to study the way a professional dancer performs a simple set dance routine compared to how a beginner performs the same set of movements. By analysing data from both performances, we hope to find information that will help a beginner to identify the parts of their movement that need changing in order to emulate the model performer. (Naemura, Suzuki 2005)

The second aspect of human movement targeted by our research is the way in which people recognise another person's emotional state by how that person moves, and how a dance performer conveys these emotional cues to an observer. The research presented here is more concerned with this second aspect, but we believe it may also be applicable to the first in terms of interface design at a later stage.

3 Prior Work

The project is informed and inspired by previous work across a variety of fields. Perlin and Goldberg (1996) in their *Improv* authoring environment placed a lot of emphasis on creating believable emotional cues with computer-generated characters, and also the importance of real time responses and gestures. Perlin's facial animation applet (Perlin 2000) also shows an engaging level of emotional communication from subtle visual cues manipulated in real time through a simple screen-based interface.

Motion capture data is often modified by movie animators, and filters are typically applied to smooth out inconsistencies in the capture process. Usually they are

reluctant to modify the original data too much, as they do not want to lose the essential quality of the particular performer's movement, but occasionally filters are used to create caricatures of the original movements. Recently techniques have been developed to apply characteristics of motion captured performance to keyframe animations as a kind of *movement texture* (Pullen & Bregler 2002), creating the potential to add an individual performers *kansei* to otherwise artificial movement.

Emotion-related work stemming from theory and practice of dance can be seen in the EMOTE 3D gesture engine (Chi et al 2000). In contemplating 3D interfaces to emotional expression, we are drawn to the various spatial models of emotion found in social robotics literature (Breazal 2003, Miwa 2003) as a possible metaphor to explore in interaction. We intend to extend this work by providing a realtime user interface using augmented reality or physical elements to facilitate the communication of subtle body signals that are part of human *kansei*.

4 Interface Problem

Our ideal is to use *kansei* analysis of movement data to define algorithms for use as real time filters of live human dance movements. Such a system would provide an immediate 'emotional mirror' by providing feedback about how the system 'sees' their movement, maybe in an abstracted or exaggerated form. The feedback might involve a computer-generated character who copies a person's movements, but reinterprets the emotional content of the movement in some way. However, at the time of writing, computer systems lack the power to perform such a feat in anywhere near real time. Also, the group's research on movement analysis is still at a theoretical, data-gathering stage and has yet to yield usable filter algorithms.

Even in the absence of 'something to interface to', it is still important to consider the user interface in the early stages of such a project. This is because an interface can help provide a focus for the lower-level and theoretical aspects of the research. To be able to make early mock-ups is also desirable. If the researchers are able to try even a fake version of a working system, it can help rekindle the excitement sometimes lost in the grind of running experiments and gathering data. For these reasons, we felt that it was worthwhile to start trying some interface ideas despite the absence of fully functioning software or a concrete application.

We looked for a tool that would enable the following:

- Import motion capture data files.
- View the data as a waveform.
- Animate an articulated *skeleton* based on this data.
- Use the above skeleton to capture and animate the mesh of a humanoid model.
- Allow for ways to readily process the data both offline and in real time.
- Allow interface via on-screen widgets and external devices.
- Construction of a 'fake' application allowing realtime modification of the original dancer's movement.
- Provide some way of incorporating data from future datasets that have been filtered offline using as-yet-unmade kansei filters.
- Quickly try different visual forms in order to reference and extend existing interface conventions.

The tool we chose was Touch Designer, a commercial 3D modelling package by Derivative (Derivative 2005).

5 Touch Designer

Touch Designer differs from most 3D modelling packages in a number of ways. Firstly, it is designed primarily for realtime interactive 3D rather than pre-rendered cinematic animation. Secondly, it uses a procedural flow paradigm reminiscent of an analog audio synthesiser where simple geometric *generators* are fed through a series of *filters* to modify the geometry. Components are connected by together by drawing *wires* between them. This procedural chain generally runs continuously so that changes to early parts of the chain are fed through to the final visible object. The third way in which touch Designer differs from other 3D packages is that most data in the program is interchangeable and can be represented as a set of data channels. This means that motion capture data could just as easily be used to construct a static 3D object, or geometric data from an object can be fed through the same filters as control data. This, coupled with the realtime procedural filters enables us to import the capture data and modify it on-the-fly.

6 Implementations

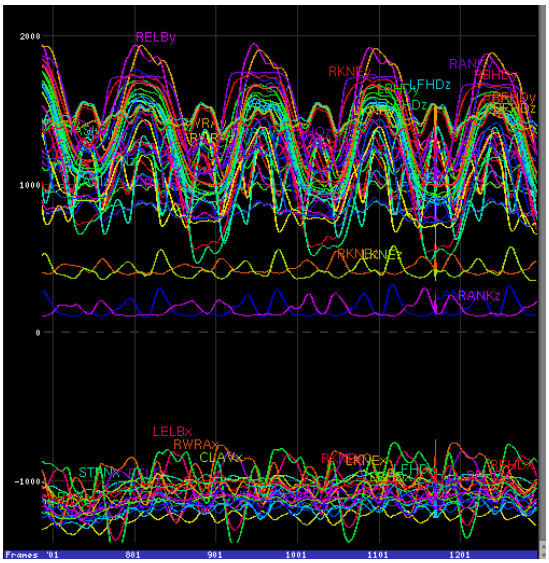

Figure 1: Raw data channels in Touch Designer

First we created a tab-delimited ascii file of the motion capture data of a dancer performing a set of repeated dance movements. The file was then loaded into Touch

Designer as a set of data channels, one for each of the tracking points from the motion capture (**Figure 1**)

Then we built a simple articulated skeleton to join the tracked body-joints (**Figure 2**). Each point was assigned the *x*, *y*, and *z* translations for the corresponding joint in the motion-capture data.

Our next task was to make a set of 'fake' realtime *kansei* filters with which to change the apparent mood of the dancer.

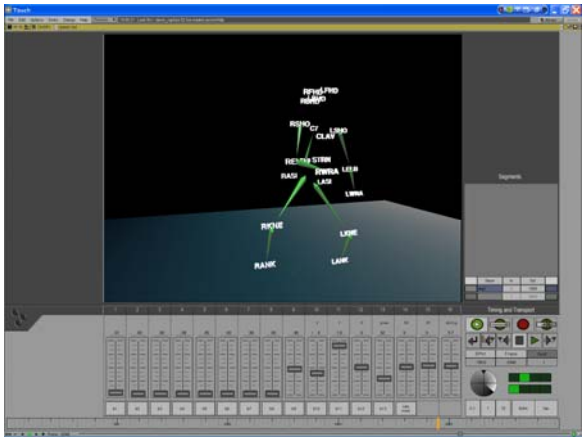

Figure 2: Dancer skeleton with default onscreen slider widgets and buttons for live control of filters.

To do this, we used Touch Designer's *channel operators* to feed the raw capture data through a series of modifiers (**Figure 3**). In the figure, a Math operator rectifies the different axis orientations between touch and the Motion-capture software.

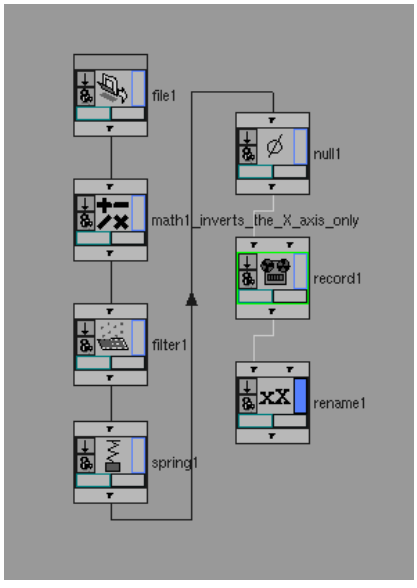

Figure 3: Motion-capture data fed through filters and modifiers to create different changes in the dancer's style of movement.

The *filter* object then applies a Gaussian filter that smooths the data causing a more lethargic, fatigued motion in the dancer. The *spring* object applies a simple spring model driven by the input data. This adds bounce or jitter to the dancer's movements depending on the *spring constant*, *mass* and *damping* settings of the spring. The *null* object simply serves as a point from which to export the channels to other parts of the program (such as the dancer) while allowing intervening objects to be added or removed. The *record* object takes the result and allows us to save it as a file incorporating all the changes made by the filters etc. for later importing via a *file* operator. Finally, the *rename* operator applies names to the channels for easy identification. If the filter is changed to a *de-spike* filter, it has the effect of making the dancer's movements more jerky and mechanical.

The performance screen in **Figure 2** has a default set of slider widgets that can be assigned to the various parameters of the filters as well as camera and lighting controls etc. A USB or MIDI fader box can be used to replace these onscreen sliders with real mechanical ones. This provides very comfortable live access to the parameters. With this arrangement, a user can quickly alter several parameters and immediately see the results. Although we have not tried this, but it should not be difficult to use live motion capture directly from the motion capture system (ignoring for the moment, tracking errors that inevitably happen in such a situation)

These modifiers create realtime changes in the character's movement that can have some emotional meaning for the observer, but the actual filters planned by our group will not be able to run in real time, so they will not be able to function in the same way. These *kansei* filters will need some time to process offline the data from a file, and render the result as a new file. This requires a slightly different approach. We need to take the data from the original file, and interpolate between it and data from the modified file. Fortunately this is possible with the *blend* operator in Touch Designer (**Figure 4**). First the *record* operator in **Figure 2** is used to save a *springy* version of the original data, then the data is fed in via the *file* operators: *original* and *springy*.

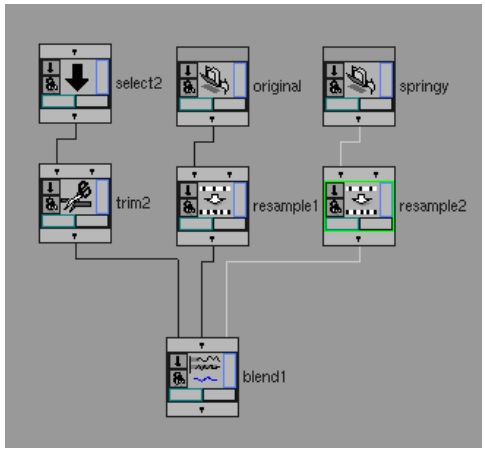

Figure 4: Morphing between two data files

Then the *select* operator to the left in **Figure 4** imports data from the MIDI faders, or whatever control device is to be applied. This data controls the morph between the two data sets. Currently there are some discontinuities between the transitions between the two. This is partly alleviated by resampling the file data to a lower sample

rate, but at the expense of overall smoothness of the motion.

7 Augmented Reality

To take this interface in a different direction, we implemented a simple augmented reality interface to the 'fake' *kansei* filters. Using DesignAR (Berry 2004), a prototype add-on for Touch Designer, the dancer was attached to a fiducial tracking marker (**Figure 5**).

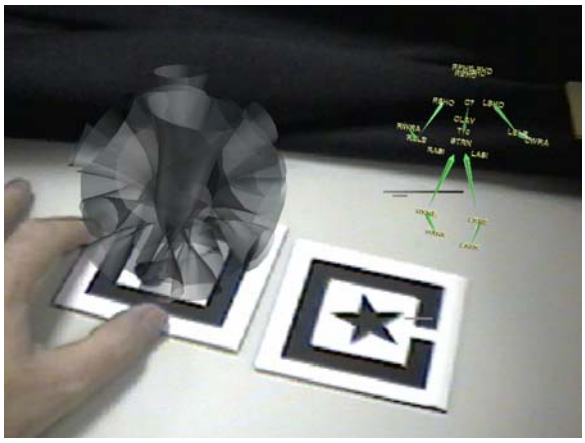

Figure 5: Augmented reality view. Proximity to the *fatigue* entity (left) causes the dancer's movement to become lethargic.

An animated *fatigue* entity, or monster, was attached to another marker. The distance between the two markers controls a Gaussian filter that makes the dancer appear more lethargic, the closer it gets to the *fatigue* thing. When the *fatigue* is taken away, the dancer recovers its normal vigour. When the dancer moves near a *spring* entity, it increases the effect of the *spring* filter described earlier (**Figure 6**). The proximity of a *robot* entity applies a de-spike filter that makes the dancer's movements jerky and mechanical. The combination of the two makes the movement robotic but springy.

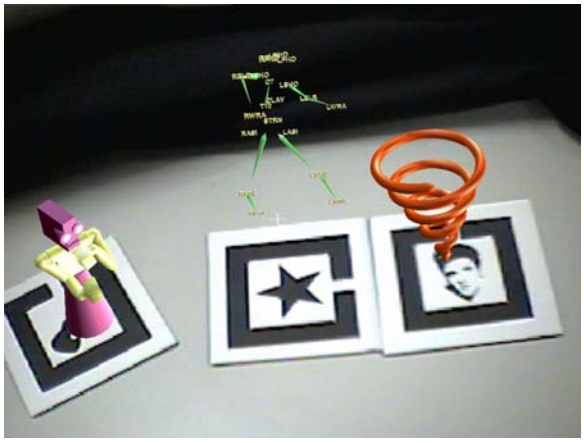

Figure 6: Proximity to *robot* and *spring* entities causes robotic, springy movements.

8 Conclusions

In this paper we have described our use of Touch Designer for prototyping interfaces for imparting emotional changes in control of an animated figure. The augmented reality interface allows a sense of agency where the filters are seen as almost spiritual forces acting on the dancer. This, we believe, has potential to create mini narratives that would not be possible by simply manipulating sliders for example. We have achieved a 'live' appearance by morphing between data sets pre-rendered by non-realtime filters. From this experience, we advocate the quick development of 'premature' interfaces in the early stages of a project, even for the simple morale boost of seeing something working. We also suggest that Touch Designer may be useful for others in this field outside of the uses we have found for it.

9 References

Berry, R. (2004) Poster0125 Authoring Augmented Reality *Conference CD-ROM SIGGRAPH 2004* project page: http://www.mis.atr.jp/~rodney/designAR

Breazeal, C. (2003) Emotion and sociable humanoid robots. *International Journal of Human Computer Studies, Vol 59, pages 119-155.*

Chi, D., Costa, M., Zhao, L., Badler, N. (2000) The EMOTE model for effort and shape. *Proc. 27th annual conference on Computer graphics and interactive techniques p173 - 182*

Derivative: Derivative company home page http://www.derivativeinc.com Accessed 26 Aug. 2005.

JSKE: Japan Society of Kansei Engineering http://www.jske.org/whatiskansei.html Accessed 26 Aug. 2005.

Miwa et al, Miwa, H., Okuchi, T., Itoh, K., Takanobu, H., Takanishi, A. (2003) A New Mental Model for Humanoid Robots for Human Friendly Communication – Introduction of Learning System, Mood Vector and Second Order Equations of Motion *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, p3588-3593, 2003.*

Naemura, M., Suzuki, M. (2005): Extraction of Rhythmical Factors on Dance Actions Thorough Motion Analysis. *Proc. Seventh IEEE Workshops on Application of Computer Vision(wacv-motion),* vol. 01, no. 1, pp. 407-413, Seventh 2005.

Perlin, K. (2000): Responsive face demo http://mrl.nyu.edu/~perlin/experiments/facedemo/ Accessed 26 Aug. 2005.

Perlin, K., Goldberg, A., (1996) Improv: A System for Scripting Interactive Actors in Virtual Worlds *Proc. 23rd annual conference on Computer graphics and interactive techniques; Vol. 29 No. 3 p 205-216*

Pullen, K., Bregler, C. Motion capture assisted animation: texturing and synthesis, *Proc. 29th annual conference on Computer graphics and interactive techniques, July 23-26, 2002, San Antonio, Texas p501-508*

This research was partially supported by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research(B), 1600038, 2005.