Musical Exoskeletons: Experiments with a Motion Capture Suit

Nick Collins, Chris Kiefer, Zeeshan Patoli, Martin White
Department of Informatics
University of Sussex
Falmer, Brighton, BN1 9QH
N.Collins@sussex.ac.uk, C.Kiefer@sussex.ac.uk, m.z.patoli@sussex.ac.uk, M.White@sussex.ac.uk

ABSTRACT

Gaining access to a prototype motion capture suit designed by the Animazoo company, the Interactive Systems group at the University of Sussex have been investigating application areas. This paper describes our initial experiments in mapping the suit control data to sonic attributes for musical purposes. Given the lab conditions under which we worked, an agile design cycle methodology was employed, with live coding of audio software incorporating fast feedback, and more reflective preparations between sessions, exploiting both individual and pair programming. As the suit provides up to 66 channels of information, we confront a challenging mapping problem, and techniques are described for automatic calibration, and the use of echo state networks for dimensionality reduction.

Keywords

Motion Capture, Musical Controller, Mapping, Agile Design

1. INTRODUCTION

Technologies to track the movements of dancers and musicians extend back to Theremin's 1930s experiments with the Terpistone platform, and continue through a host of remote sensing apparatus from radar to sonar [8, 14]. The most common current technique is computer vision, through direct body recognition or using fiducial markers [1, 2]. Some advanced systems employ multiple cameras for a three dimensional representation, such as the Vicon 8 system, as employed by Dobrian and colleagues for musical control data [3] and with an associated Max/MSP tool kit (`http://music.arts.uci.edu/dobrian/motioncapture/mcmmax.htm`). Kia Ng has investigated the use of such a system to record musician's performances for the analysis of expression and musical pedagogy [9].

This paper explores the use of a motion capture suit, as used to provide high quality non-occludable capture data for games and film special effects. Because of the price of such systems, and with a primary focus on visually dominated entertainment,[1] they have previously been used little as a musical controller, though attention has recently been turning to their potential [7]. Some artists have built custom systems themselves by co-opting accelerometer sources; for instance, Tom Tlalim's *W_music* adapts eight Wiimotes as wearables, two per limb for the costume.[2] The DIY electronics site instructables.com offers another wearable suit solution.[3] Most famously perhaps, Michel Waiswisz's The Hands and Laetitia Sonami's Lady's Glove present important examples of wearable fine motor tracking musical interfaces for the arms, intricately coupled respectively to live sampling and time-varying effects [8].

The particular suit investigated herein is a research prototype by the Brighton based company Animazoo, which is being explored for application areas in the Centre for VLSI and Computer Graphics at Sussex [10]. This prototype is meant to substantially reduce the cost of motion capture suits, with an eventual aim of creating a mass produced game controller. A MIDI enabled variant of this suit is commercially available, the Gypsy MIDI (`http://www.sonalog.com`), which offers 27 sensors per arm (nine 3-axis mechnical rotation sensors) and is sold as a single arm or double arm suit, wired or wireless, with supporting Max/MSP built standalone (the wireless suit including VAT is around 1500 pounds, but one arm with wires is about 600). The experimental suit we have access to in the lab has more channels of data than Gypsy MIDI, including head position, and can be scaled up to a full body suit with spine, hips and leg sensors.

2. THE MOTION CAPTURE SUIT

The suit prototype currently available to us consists of a predominantly mechanical exoskeleton rather than an electromagnetic or optical device. We are using an upper body suit, consisting of two arms and head sensors. There are 33 active orientation sensor channels sampled at 30fps,[4] consisting of 12 rotational readings for each arm (three axes of measurement each at wrist, elbow, shoulder and collar),

[1] An example of missed opportunities is the work of Praga Kahn, who despite being a musical team, employ the xsens MVN technology to let dancers control visual avatars, rather than contribute musically (`http://www.xsens.com/en/entertainment/live-entertainment/performance-praga-khan`)

[2] `http://www.popsci.com/entertainment-gaming/article/2008-02/dancing-song-full-body-wiimote-music-controller-suit`

[3] `http://www.instructables.com/id/Puppeteer_Motion_Capture_Costume/`

[4] A full body version of this suit exists, with 66 channels and 120 frames per second resolution, but has not been available yet for musical application tests

three axes of head rotation from head-strapped accelerometers, and three buttons for the joystick which each hand grasps. The suit is wired rather than wireless.

Data from the suit is sent to a custom executable on a Windows machine, provided by Animazoo. This passes data on to an application built in the Computer Graphics Centre, via a private SDK for development. This C++ application then in turn passes the data via UDP over a network to the Macs that will deal with sonification. The average latency for pinging over the high speed LAN is under a millisecond. We are using SuperCollider for audio in this project, which doesn't directly receive UDP packets, so we go via a Processing application, exploiting the udp and oscP5 libraries to create Open Sound Control messages for SuperCollider, tagged /motioncapturedata. The data is sent as a single long string; SuperCollider breaks up the string by space delimination, and converts from characters to an array of floating point values. The triggers on the joysticks are binary values, but others are rotation values in degrees with variable range, depending on the affordances of the human body; for example, there is a greater range of movement lower down the arm.

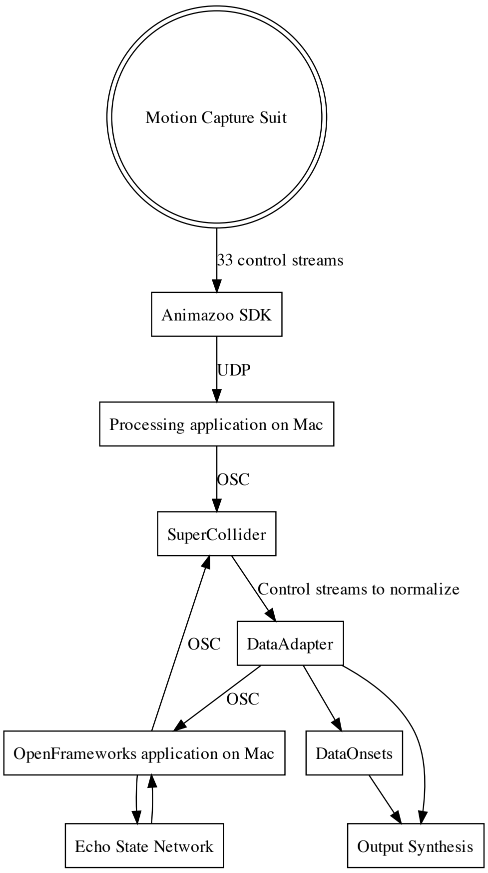

Figure 1: Signal processing flow for suit data

Figure 1 illustrates the data handling required to obtain realtime data from the suit as well as the additional data analysis processes we explored, and Figure 2 depicts a user wearing the suit in musical control flow.

3. 'AGILE' DEVELOPMENT

In order to explore the musical uses of the suit, the first

Figure 2: Zeeshan Patoli models the motion capture prototype suit; he is interacting with a discrete triggered piano patch

two authors (as the musician-programmers) adopted a fast development cycle intended to maximise what lab time we had available with the motion capture suit specialist Zeeshan Patoli. We wanted to evaluate the benefits of techniques for fast reactivity in situ. We employed both individual and pair programming in preparation sessions outside the lab, and then a fast swap of lead application developer within the sessions themselves, so that one could code new variations, whilst the other was solving an interfacing issue or testing examples. For full musical trials, we continually swapped the person in the suit, so that all three of us had chance to try patches and give dynamic feedback. We recorded active sessions (when not engaged in more low level debugging tasks, but actively trying musical control algorithms) by video camera, for later review.

This rather more manic development cycle is not one traditionally tackled within task-based HCI experiments or even concert or rehearsal based review of developed systems [13, 6, 5, 11], but provided continually interesting and immediate feedback within a rapidly changing environment. The lab itself was a relatively busy place; had we not been locked to working where the suit and its host computer were based, we would have preferred private studio sessions. Not all lab residents were happy with the audio outputs ('It's not too loud it's just hurting my ears' was one candid description of our exploration of a rather more noisy patch), and themselves generated extraneous noise; we ended up moving our sessions later in the day to avoid core hours.

An unexpected obstacle quickly appeared, absorbing almost all the time in the very first session: networking problems between the PC and Macs were eventually traced to a bug in the Windows application code. We were however able to save the day by making the breakthrough before the end of the session, and recording lots of sample suit data for a separate analysis session. This then motivated building a flexible, adaptive data manager, as described below.

To illustrate a further disruption, misunderstandings in the nature of the control streams were at first frequent, and only properly clarified after we'd prepared patches that expected more data than they actually got. Frozen voices within the sound patches were a clue that something was amiss. It turned out that we had been misled to expect 66 channels (the full body suit) but were only getting half that. In later patches, once this specification of particular channels was resolved, we created much more specific patches which were qualitatively more successful.

4. MAPPING TECHNIQUES AND SOUND PATCHES

In a pair programming session where we analyzed example live data from the suit, Collins and Kiefer co-designed a SuperCollider class for a general control data frontend. An instance of this DataAdapter class can accommodate any size array of floating point numbers, and automatically calibrates itself to the ranges of feature values exhibited, normalizing to the 0.0 to 1.0 range for ease of use. This maximises the range of useful values provided from any given sensor, though it may distort the relative relations of sensors. Nonetheless, for our purposes, it was an essential tool to speed the process of creating musical mappings, without constantly tripping over the differences between the various sensors. The implementation is relatively straight forward, just to extend the max and min values associated with an individual channel, and thereby normalize within a range.[5] Whilst this may lead to initial transient behaviour whilst a range is established, in practice there were no adverse effects, as the adaptation could take place before invoking any sounding patch.

Each musician-programmer then prepared individual patches drawing upon these frontends. Table 1 lists various example sonifications from our trial sessions. Some of these used variations on the basic DataAdapter frontend. Additional analysis modules were investigated for some patches; the experiments with Echo State Networks are further examined below. The DataOnsets class was written to look for higher order differences in data, reacting to sudden movements, providing one transient trigger signal for each input data channel. The onset detection algorithm involved summing the absolute difference between the most recently arrived value and the last N, with comparison to a threshold. N and threshold could be specified, and N=5 and threshold values between 0.1 and 0.5 were used in these investigations. Threshold was modified to explore sensitivity factors in sound grain generation.

5. EXPERIMENTS WITH ECHO STATE NETWORKS

Echo state networks (ESNs) [12] are a form of recurrent neural network that use a trainable set of output weights to exploit the dynamics of a random *reservoir* of interconnected neurons. ESNs can be trained to approximate nonlinear dynamical systems; they respond to a stream of inputs over time, and so show promise for musical applications [4]. From a previous experiment with echo state networks and tangible interaction, it was observed that by stimulating a specific output while streaming a particular pattern to the inputs in training, this output would then respond to this particular movement of these inputs after training. The precise mapping is difficult to predetermine without careful training, but nonetheless potentially interesting for musical control, and a useful method of dimensionality reduction from a large set of input streams. This method was trialled in these sessions to reduce the large amount of suit data to a smaller set of streams. Holzmann's *AUReservoir*[6] C++ library was embedded within an OpenFrameworks program that controlled training and displayed data streams. An ESN was configured with 33 inputs, 6 outputs and a reservoir of 200 nodes; motion data received from SuperCollider was fed through the network and the outputs returned, all through OSC. To train the ESN, motions such as waving an arm were recorded and associated with individual outputs. After training, motion data was fed through the network and the outputs mapped to musical parameters.

Front end	Sound mapping	Notes
DA	Audio input is processed by pitch shifting and comb filtering	Version with 8 voices with 4 parameters each, and one with 2 voices with 6 parameters each (one per lower arm)
DA	Dynamically create enveloped subtractive synthesis voices constrained to move in pitch on a diatonic scale	Each voice uses a random octave and a choice of six control streams
DO	Onsets trigger notes; right arm controls bass notes, left higher pitches, head rotation selects major chord from chromatic scale	MIDI sent through to Logic MainStage to utilise high quality sampled patches
DA	Data streams are split into groups and used to control parameters of five grain clouds.	The clouds work on different random areas of the same sample.
DA	Each data stream controls the amplitude of a sound source.	Sound sources are harmonised sine waves or small loops from a sample.
ESN	As above	The data is reduced to 6 streams through the ESN

Table 1: Table of example musical mappings: DA is DataAdapter, DO is DataOnsets, ESN is Echo State Network

6. MUSICAL RESULTS

Qualitatively, the most successful patches were most typically those with the most direct mapping. This might be initially attributed to the novelty value of the suit as a controller for the participants, when the experience was fresh, though there was a continuing sense of immediacy and accessibility across various sessions, from suit wearers and observers alike. Interested parties in the lab kept wandering back to see what we were up to, providing an informal stream indicating those patches which proved crowd pleasers. Those onset controlled patches which triggered a MIDI enabled sampler seemed universally successful. This seemed to follow from both the reactivity in terms of a granularity of discrete responses, and the sense of body control, with clearly differentiated roles for the two arms and the

[5]The only awkward case is when a very small range is observed, perhaps a single value or extremely close values; the code has a pragmatic heuristic which picks up on the overall range being under 0.000001, and returns only 0.0 in that case.

[6]`http://aureservoir.sourceforge.net/`

head. Individually, the second author observed that the piano sound was a fascinating choice for him, in that it transfigured his experience of piano lessons to an unfamiliar physical interface. Although the sonic result for the audio input patch was not quite as strong, it was interesting to explore the use of the suit as a control to processing rather than direct generation, and gave a sense of duetting as one performer used the microphone and one transformed their voice. However, within the strictures of this patch, the alignment with limbs again proved more successful, rather than using an increased number of voices spread across the body.

However, although the ambient soundscapes were hardly as immediate, they did provide a solid sense of discovery. Wearing the suit for a prolonged session, the first author discovered that there was a 'home position' for the patch which he could return to and which provided a point of aural respose; the suit's own physical presence actually assisted achieving this neutral position. From that home point, he could navigate through the complex synthesis space, gradually hearing out correlations of action to aural resultant, and achieving some subtlety of performance.

The dynamic remapping, with new sound events taking on a random configuration of control streams, could still lead to some interesting results. This may be because the sound events themselves were rich enough, even if the patch itself was unsurprisingly inconsistent in control. The performer attended much more to the fact that they had a rich effect on the sound, than consistency of mapping in their movement; when grain density was increased, the aggregate effect tended to be that of continual influence anyway, even though individual sensors were continually swapping their roles.

The result of the ESN mappings were less encouraging, the performer perceiving a lack of connection to the music where before, direct mappings had worked well for the same patch. While this technique had previously worked successfully on a smaller scale with a different controller, the rapid-prototyping environment meant it was difficult carry out the more detailed choreography and trial and error of the ESN training process to achieve a more useful set of mappings. This technique could however show promise in a more controlled setting.

7. CONCLUSIONS

This paper has outlined experiments with a new motion capture suit interface within a laboratory. Whilst we have not yet proceeded to a concert setting, exposure to this interface has widened our knowledge of the sonic control possibilities. Although the more direct sound mappings provided more palpable hits, there was potential in a selection of more subtle materials. However, use of dimensionality reduction techniques (namely, the use of the ESN) did not lead to particularly successful outcomes in this context. We certainly found that strongly correlating components of patches to arm and head positions had a beneficial effect, and the ESN most likely suffered from losing this specificity.

Because of our aim to maximise the musical exploration, there are a number of more technical evaluations we would like to return to. We would also next like to invite in third parties who have not been privy to the experimental sessions, to explore our favourite sound mappings. And of course, the possibility of concert performances will bring a further strong incentive to our ongoing evaluations.

8. ACKNOWLEDGMENTS

Thanks to Martin White's Computer Graphics Centre for hosting the experiments and their continuing research efforts into applications for the suits. For essential help resolving UDP via TCP/IP networking issues for connecting PCs and Macs, a big thankyou to Cash Garman, who honed in on the problem via the nc command line app. Jake Slack also assisted with an initial try-out session with the suit.

9. REFERENCES

[1] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, 2008.

[2] A. Camurri, G. De Poli, A. Friberg, M. Leman, and G. Volpe. The MEGA project: Analysis and synthesis of multisensory expressive gesture in performing art applications. *Journal of New Music Research*, 34(1), 2005.

[3] C. Dobrian and F. Bevilacqua. Gestural control of music: using the vicon 8 motion capture system. In *Proceedings of New Interfaces for Musical Expression (NIME)*, pages 161–163, 2003.

[4] G. Holzmann. Reservoir computing: a powerful black-box framework for nonlinear audio processing. In *International Conference on Digital Audio Effects (DAFx)*, 2009.

[5] W. Hsu and M. Sosnick. Evaluating interactive music systems: An HCI approach. In *Proceedings of New Interfaces for Musical Expression (NIME)*, 2009.

[6] C. Kiefer, N. Collins, and G. Fitzpatrick. HCI methodology for evaluating musical controllers: A case study. In *Proceedings of New Interfaces for Musical Expression (NIME)*, Genoa, Italy, June 2008.

[7] V. Lympourides, D. K. Arvind, and M. Parker. Fully wireless, full body 3-d motion capture for improvisational performances. In *Proceedings of Computer-Human Interaction (CHI)*, Whole Body Interaction workshop, Boston, 2009.

[8] E. R. Miranda and M. M. Wanderley. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard.* A-R Editions, Inc., Middleton, WI, 2006.

[9] K. Ng, T. Weyde, and P. Nesi. i-Maestro: Technology-enhanced learning for music. In *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, 2008.

[10] M. Z. Patoli, M. Gkion, P. Newbury, and M. White. Real time online motion capture for entertainment applications. In *The 3rd IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning DIGITEL 2010*, 2010.

[11] D. Stowell, A. Robertson, N. Bryan-Kinns, and M. D. Plumbley. Evaluation of live human-computer music-making: quantitative and qualitative approaches. *International Journal of Human-Computer Studies*, 67(11):960–975, 2009.

[12] D. Verstraeten. *Reservoir Computing: computation with dynamical systems.* PhD thesis, Unversity of Gent, 2009.

[13] M. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from HCI. *Computer Music Journal*, 26(3):62–76, Fall 2002.

[14] A. O. Wilson. Sensor- and recognition-based input for interaction. In A. Sears and J. A. Jacko, editors, *The Human-Computer Interaction Handbook (2nd Edition)*, pages 177–199. Lawrence Erlbaum Associates, New York, NY, 2008.