ENCODING SOUND BY POLYNOMIAL INTERPOLATION FOR INTELLIGENT DYNAMIC MUSIC IN COMPUTER GAMES

M. A. Burley, N. E. Gough, Q. H. Mehdi[1] and S. Natkin[2]

[1] Games, Simulation and AI Centre
Research Institute for Advanced Technologies
University of Wolverhampton, Wolverhampton, UK
E-mail: M.Burley@wlv.ac.uk

[2] Centre De Recherche en Informatique du CNAM
Conservatoire National des Arts et Métiers, Paris, France
E-mail: natkin@cnam.fr

KEYWORDS

AI, Polynomial interpolation, sound, music, computer games

ABSTRACT

Current research in computer music composition almost exclusively involves the manipulation of music stored as MIDI data. While this allows direct access to the structure of music, it creates limitations in realism for the end result of such techniques. This paper describes a method designed to represent music in a form that facilitates the use of existing processing techniques while conserving the 'real-world' attributes of music recorded in PCM format giving computer-game developers a facility for the production of variations on a pre-recorded theme, whatever the original source. Experimental results are presented to demonstrate that polynomial interpolation is a viable technique.

INTRODUCTION

This paper explores the use of polynomial interpolation to improve the generation of audio tracks for computer games. Traditionally, there is a recurring tendency for computer music research to tackle the processing of music at a grammatical level. Music is often described as a language and, indeed, can be quite legitimately thought of as so. There is however evidence to suggest that working at a higher-level than that of the note-sequence has considerable potential for analysis and composition. As far back as 1979, it was becoming apparent that simply applying techniques similar to those used in Natural Language Processing (NLP) was falling short of the mark in unlocking the secret of what makes music sound musical (Meehan 1979). The concept of 'Shenkerism', whereby initial parsing of a piece of music is performed at the note-group level rather than delving into every facet of its structure, was a hint that being 'less-precise' could in fact make the task of instilling creativity in computer music easier. This was also the case with the later POD system of Truax (1977) that introduced the concept of 'Digital Sound Objects'. A survey by Roads (1985) is quick to criticise many of the automatic-composition systems developed around the middle of the twentieth century for their rigidity – something that could reasonably be seen as a necessary compromise to achieve the required degree of success when using an abstracted representation of music.

Nevertheless, computer music research seems to be anchored to the concept of musical notes whether the technique in use is a Neural Network, Genetic Algorithm, Stochastic or Grammatical algorithm or an Iterative Formula (with some exceptions in the latter case). A cursory glance through core texts in computer music such as Roads (1995) and Miranda (2001) will make this apparent. One possible reason for this is the fact that, when working at a higher syntactic level, one faces the choice of either being limited by having to work with note-groups or relative pitch structures as atomic components or, if these high-level symbols are made more flexible, loosing some of the very information one is trying to process.

The computer-games industry largely ignores existing automatic composition techniques. The game 'Halo' (O'Donnell, 2002) which is recognised as having one of the most advanced dynamic-music systems currently in existence only stretches to event-driven transitions between manually-composed segments of music whilst other landmark games such as Quake seem to treat background music as a technical afterthought.

A major driving factor behind the technique presented here, was the desire to preserve as much data as possible when improvising around an existing composition. Of paramount importance in this respect is the issue of timbre. Usually defined as the characteristic of a sound that allows us to identify it as emanating from a particular source (a musical instrument for instance), timbre is an issue for any composition stored in MIDI format as the composer is restricted to whatever sounds the synthesis module of the sound-card can generate. This issue becomes augmented when working with existing compositions recorded from real instruments in PCM form. A conversion to MIDI format allowing computer improvisations destroys all of the original timbre information resulting in (despite the use of advanced

synthesis techniques) an artificial sounding end-product that fails to preserve any of the nuances and idiosyncrasies of the original performers. The aim of the work described here was to allow any recording to be used as the basis for background music in a game so as to create the same emotional effects that a film-soundtrack, which is be tailored to a predefined script, creates for its audience. Consequently, MIDI was discarded as a viable option.

The disadvantage of working with wave-data as opposed to MIDI is that while it might be possible to decipher and work on the regular, amplitude samples provided in wave-data files (using say a neural network), this involves a complex procedure just to obtain a single note that can be identified from the mass of fundamentals, partials and general background noise. This limitation was the initial hurdle in the process of developing a professional dynamic music system for use in computer-games. While the idea of a musical-improvisation system composing the soundtrack for a game in real-time and in response to environmental and narrative factors present in the gaming environments is not unresearched (Casella & Paiva 2001), there seems to be an automatic choice of MIDI as the format to work with, presumably for the reasons already mentioned. It is felt that a compromise is possible if some of the complexity of such a representation system were to be handled by Artificial Intelligence (AI).

The aims of this research are thus to produce variations on an existing sound track by means of AI; to limit the representation of that theme by defining only at a conceptual level; to segment the track and represent each segment parametrically; and to use the parameters to generate new instantiations of the sound.

The paper is organised as follows: In the next section we examine the possible use of AI techniques to solve this problem and outline the use of polynomial interpolation as a basic data representation for this process. This is followed by a description of the experimental methodology and the results obtained. The paper concludes by examining the limitations and possible improvements for the proposed technique.

METHODOLOGY

AI as a Facilitator

Artificial Intelligence (AI) provides a way of tackling problems without having to think about the fine detail. As an eventual aim of the work is to produce variations on a theme by means of AI, the representation of that theme needs only to be defined at a conceptual level. What is required is a level of quantisation whereby each code represents not just pitch information but rhythmic and timbral information as well. Our approach is based around the theory of wavelets and is designed to allow segments of a soundtrack to be categorised as instantiations of dynamically identified generic waveforms. While these 'waveform-objects' are extremely difficult to work with manually, it is believed that a stochastic technique such as a Markov Model (Russell & Norvig 1995) or an AI technique such as the Kohonen Self Organising Map (Kohonen 1989) will be able to identify the relationships between them in the context of specific musical tracks. These relationships can then be manipulated in order to induce variations on the original theme, theoretically producing music that sounds as though performed by the original artists.

Segmentation and Polynomial Interpolation

In order to identify segments of PCM data as specialisations of generic waveforms, it is necessary to find a representation of those segments that allows rigorous comparison. The approach taken here is a functional one. Lagrange Interpolation is applied to successive segments of the waveform representation of a track resulting in a sequence of polynomial equations, each of which representing a particular segment. As the Lagrange formula allows determination of the degree of a polynomial before performing calculations, the only parts of an equation that need to be stored are the coefficients of the various terms.

The Lagrange formula adopted is as follows (Butler & Kerr 1962):

$$l_j(x)=\frac{(x-x_0)(x-x_1)..(x-x_{j-1})(x-x_{j+1})..(x-x_n)}{(x_j-x_0)(x_j-x_1)..(x_j-x_{j-1})(x_j-x_{j+1})..(x_j-x_n)} \qquad (1)$$

$j\in[0,n]$, $j\neq n$, where $x_0 \ldots x_n$ represent a series of values for the independent variable (in this case, instants in time) and $l_j(x)$ is the polynomial for the wave segment at instant x. A complete approximating polynomial for a given segment is obtained by summing the products of the various $l_j(x_j)$ and their corresponding $f(x_j)$ (the amplitude at instant x_j):

$$L(x)=l_o(x)f_0+l_1(x)f_1+\cdots+l_j(x)f_j+\cdots+l_n(x)f_n \qquad (2)$$

It was discovered pragmatically that, in the context of this paper, Lagrange polynomials generated from large sets of PCM data are generally unreliable in terms of accuracy. Another important issue affecting the choice of parameters was that of sampling frequency. The danger is that by taking PCM values in close proximity to each other, little variation is picked up, with the result that each 'wavelet' reduces effectively to a simple curve or, in extreme cases, a line. In practice this would lead to identical classifications of most segments giving no significant outcome.

Currently the sampling rate (11,025 Hz), number of interpolation points (6 per segment) and distance between interpolation points (5 samples) are fixed at values chosen through informal experimentation. While this configuration is sufficient to demonstrate the potential of the technique, it is unlikely that this approach will be sufficient to take the work forward. As there is no relation between the aforementioned parameters and the structure of the music being processed, it is purely a matter of chance as to whether or not the more (structurally) significant parts of a wave are picked up or missed by the interpolating-quantiser. The next step is therefore to add a degree of 'intelligence' to the algorithm, taking into account the structure of the music on which it is working in both the time and frequency domains.

EXPERIMENTS

Various recordings of a musical soundtrack were made in PCM format using the low-level wave functions provided by the Win32 API in a modified version of the simple buffered recording program provided by Petzold (1998) in order to reduce development time. This approach provided memory-buffers of wave data that were then processed by the interpolating-quantiser. The Lagrange coefficients were written to a file. The quality of this representation was then validated by reconstructing the waveform and comparing the result with the original. While the technique is not designed to be an alternative storage-format for music due to an inevitable loss in sound-quality caused by the geometric properties of the Lagrange polynomials, this exercise was necessary in order to verify that the generated wavelets bore sufficient relation to the original wave-segments. Once it had been determined that the LIP data, when played back, was recognisable as the original PCM recording, graphical representations of some of the wavelets were made with their associated wave-segments. These graphs clearly illustrate the potential for success of this approach to sound representation while simultaneously highlighting areas for improvement. A low sampling rate was deliberately chosen in order to determine the maximum 'strain' that the system could deal with.

RESULTS

The following graphs illustrate the effect of interpolative-quantisation using the Lagrange-based technique described above on a piano rendition of the C Major scale sampled at 11,025 Hz (CD audio is generally recorded at 44.1KHz). The start and end samples are given in the titles. Also, note the differing ranges of the Y axes.

The reproduction in Fig. 1 is very close in form to the original, however the Lagrange technique is at the mercy of two factors. We will discuss the most innate of these shortly, but an evident side effect of taking points at fixed intervals is the fact that, by missing a peak or trough, the resulting polynomial will flatten out that part of the waveform as the subsequent group of samples shown in Fig. 2 demonstrates.

One should also be aware that making even a slight change to a waveform can introduce many new partials (component waves that, when added together, form the complex wave seen) and that because of the way the brain interprets sound waves (Zotkin et al, 2003) this can have unwanted side effects such as single notes being turned into chords and pseudo-random timbres replacing the sounds of the original instruments.

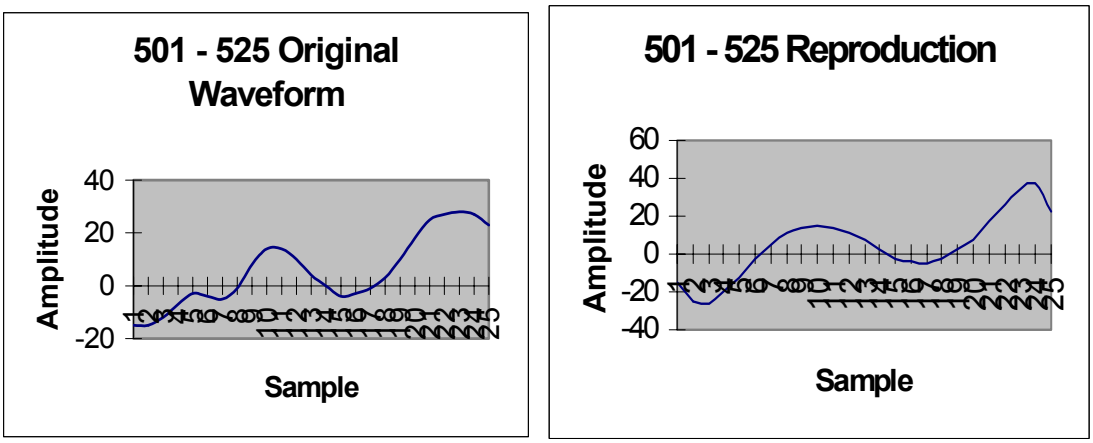

Fig. 1 Acceptable interpolated reproduction of wave-segment

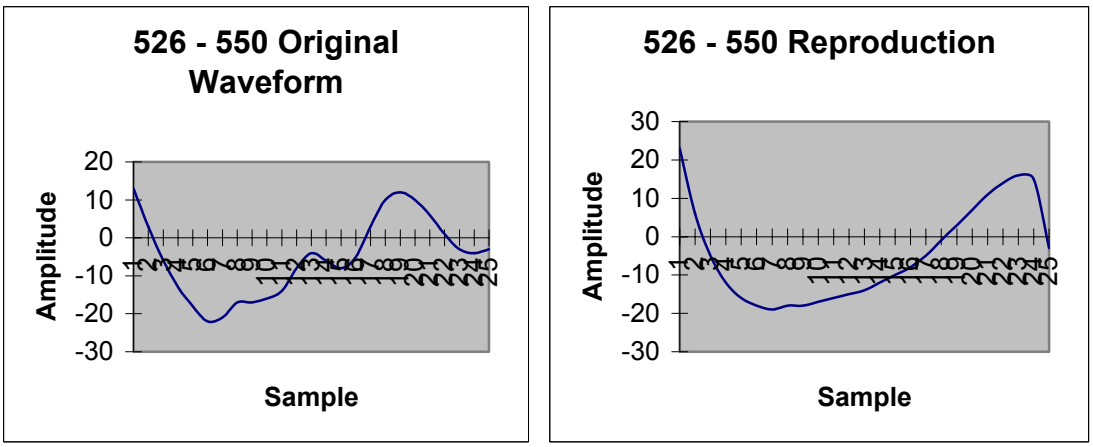

Fig. 2 Inaccurate reproduction

LIMITATIONS AND POSSIBLE SOLUTIONS

The primary risk in using Lagrange Interpolation is that the formula does not offer a way to determine how well a polynomial fits the original waveform (Hosking *et al*, 1986). One technique which is less than satisfactory and high on the priority list for replacement is that of overlapping the regenerated wavelets to compensate for a tendency whereby (at least with the current, empirically determined operating parameters of the system) the Lagrange Polynomials usually become very inaccurate after the last reference point (I.e. the last point given to the Lagrange formula from the original PCM data). This may be resolved by the measures described below, but if not will warrant the replacement of the Lagrange formula by another polynomial technique.

As has already been stated, the final aim of this work is not to develop an alternative sound storage technique. It is to create, in parallel to the PCM data, a meaningful digest of a piece of music that can be used to alter it without loosing timbral information. We must therefore take into account the fact that some parts of a piece of music are more significant than others. Immediately, fixing the interval between interpolation points is highlighted as a problem. While Lagrangian Interpolation can implicitly deal with varying distances between these points, the problem again arises that we have no way of determining the accuracy of a polynomial and, with so many polynomials resulting from even a one second sample of PCM data at 11,025 Hz, no way of flagging 'bad' polynomials for treatment by a corrective algorithm.

On the other hand, other interpolation techniques tend to require data tabulated at equal intervals (Hosking *et al*, 1986). This may turn out to be unacceptable for reasons already mentioned. Also, the order at which these constant intervals are found often determines the order of the resultant polynomial. This makes the storage structure required more complex but may provide a payoff in terms of accuracy.

With some options clearly available, we then face the task of identifying significant events in a piece of music. While the efficacy of techniques with which to accomplish this is yet to be investigated, it is felt that a protocol-analytic study of composers and audience members will be of value.

Assuming for a moment the worst case scenario; it may become apparent that interpolating polynomials will be sufficient for the manipulation of music by AI techniques but not sufficient to completely replace PCM as a recording **and** playback format. It will therefore be necessary to map changes made to the approximating polynomials to the raw PCM data that will actually form the output of the dynamic-music system. It is here that a more significant overlap with Computer Sound Synthesis occurs. Slaney, *et al* (1996) have conducted research into the problem of "Automatic Audio Morphing", a means of smoothly transitioning from one sound to another. They achieve this by isolating each different aspect of a sound-wave to it's own dimension. These dimensions are then warped according to freely-definable rules governing the relationships between them. This approach may well provide a satisfactory Polynomial-PCM bridge in our dynamic-music system.

CONCLUSION

While currently in it's infancy, we have demonstrated a way of representing sound that has the capacity to facilitate the manipulation of any music stored in PCM format while preserving all of the original data not concerned with the piece's 'grammatical' structure. This is almost an opposite approach to that taken by systems using the MIDI standard or similar. Any pitfalls currently inherent in the technique indicate their own solutions and have enabled us to construct a solid methodology with which to take the work forward.

References / Bibliography

Butler, R. and Kerr, E. (1962) *An Introduction to Numerical Methods*. 1st ed., London: Sir Isaac Pitman & Sons Ltd.

Casella, P. and Paiva, A. (2001) MAgentA: an architecture for real time automatic composition of background music. *Intelligent Virtual Agents'2001*, Springer.

Hosking, R. J., Joyce, D. C. and Turner, J. C. (1986) *First steps in numerical analysis*. 1st ed., Kent: Hodder and Stoughton Educational.

Kohonen, T. (1989) *Self-Organisation and Associative Memory*, 3rd Ed. Springer-Verlag.

Meehan, J. (1979) An Artificial Intelligence Approach to Tonal Music Theory. *in* ACM: Association for Computing Machinery *Proc. 1979 Annual Conference, 1979*. New York: ACM Press, pp.116-120.

O'Donnell, M. (2002) *Producing audio for Halo* [online]. [accessed 7th September 2004]. Available from: < http:// www.gdconf.com/archives/2002/marty_odonnell.doc >

Petzold, C. (1998) *Programming windows*. 5th ed., Microsoft Press International.

Russell S. and Norvig P. (1995), *Artificial Intelligence a Modern Approach*, Prentice Hall Inc.

Slaney, M.,, Covell, M. and Lassiter, B. (1996) Automatic Audio Morphing. *In Proc. IEEE int. Conf. Acoust., Speech and Signal Processing, 1996, Atlanta.* pp.1-4.

Truax (1977) The POD System of Interactive Composition Programs. *in Computer Music Journal,* 1(3), 1977, pp. 30-39

Zotkin, D., Shamma, S., Ru, P., Duraiswami, R. and Davis, L. (2003) Pitch and timbre manipulations using cortical representation of sound. *in Proc. ICASSP, 2003, April, 2003, Hong Kong.* vol.5, pp.517-520.