PRIORITIZING AUDIO FEATURES SELECTION USING ANALYSIS HIERARCHY PROCESS AS A MEAN TO EXTEND USER CONTROL IN CONCATENATIVE SOUND SYNTHESIS

Noris Mohd Norowi *Eduardo Reck Miranda*

Interdisciplinary Centre for Computer Music Research,
University of Plymouth
PL4 8AA, United Kingdom
noris.mohdnorowi@plymouth.ac.uk, eduardo.miranda@plymouth.ac.uk

ABSTRACT

User control is one of the most important heuristic principles of a system design as it gives users the freedom to choose a system's functions and as a mean of communicating instructions to the system before performing a specific task. Existing concatenative sound synthesis systems call the need for a more flexible user control function, in particular during feature selection. This paper studies the effect of different combination of audio features, their orders and weights have on the segment match. It also proposes that a decision making methods such as the Analysis Hierarchy Process should also be incorporated into the system to enable weights to be generated automatically for each feature.

1. INTRODUCTION

In this computer age where almost all problem-solving processes are mechanized, the inclusion of artificial intelligence (AI) in performing any tasks is unavoidable, including the process of music making. One of the ways to create music intelligently is through the use of a concatenative sound synthesis (CSS) system. CSS is the art of producing new sounds from a composite of many small snippets of audio. Analogous to photomosaicing [1], it assembles large number of unrelated small sound segments called tiles together according to a specification given by an example sound referred to as the sound target, to form a coherent, larger sound framework.

Currently, uses of CSS is already widely exploited in concatenative speech synthesis, via the text-to-speech technology, in commercial applications such as the talking watch, the talking calculator, the natural voice reader or voice announcer. Although more popularly used in the speech domain, its data-driven nature of has pushed forward more potential use of CSS in music. As music is more artistically perceived, its concatenation can be more flexible compared to synthesized speech, which needs to be coherent and human-like. This dexterity allows more space for music creation using CSS.

However, many existing CSS systems for music can be criticized on the grounds that they provide limited control to the users, i.e. the degree of freedom a user has to make on variables such as selection of source files, audio features to analyze, prioritizing the features according to weight and order, setting the threshold on the unit selection process, etc.

One of the more important variables is the audio features. In CSS, the small snippets of sounds undergo feature extraction, where a compact numerical representation of the audio segment is obtained. Different audio features correspond to sound characteristics. Examples of audio features are the pitch, loudness, spectral centroid, zero crossing rate (ZCR), etc. Typically, more than one feature is exploited at any one time from the audio segments in order to draw any significant correlation or patterns that might exist. However, not all audio features might be needed for extraction – this very much depends on the task at hand. Moreover, even when several features are considered relevant for extraction, they may possess different level of importance from one another, i.e. it might be more important to find a match for audio segments with closer ZCR values than it is the pitch. In such case, it is more computationally economic and time saving to have only the relevant features extracted, in order of importance.

This paper demonstrates the effect of prioritizing audio features according to feature order and weight has on the concatenation result and suggests the use of the Analysis Hierarchy Process (AHP), a simple weight-applying mechanism that can be utilized to differentiate the different level of importance between the features [2], to be used for this purpose.

This paper is organized as follows: the first section of the paper gives a basic introduction to CSS. Section 2 describes the importance of extending the user control in CSS and discusses the features selection options of several existing CSS systems. Section 3 then describes the experimental setup. Examples and discussion are presented in Section 4, whilst conclusion and future expansion are included in Section 5.

2. AUDIO FEATURES SELECTION IN CSS

Control can be defined as an act 'to determine the behaviour or supervise the running of (something)' [3]. Limited user control usually leads to frustration, where user would associate a negative attitude towards the entire system, regardless of the system's potentials [4]. It is therefore important that a positive relationship between user and system is established from the start.

Systems with good user control are typically known to be effective, easy to use and have the overall ability to aid user [5]. Extending user control in a CSS system would mean increasing the flexibility of options offered to users during the music making process.

In order to fully appreciate user control in CSS, it is crucial that the three main processes that take place in a CSS system are understood: (1) analysis; (2) unit selection; and (3) synthesis. Figure 1 shows the dataflow of a typical CSS system.

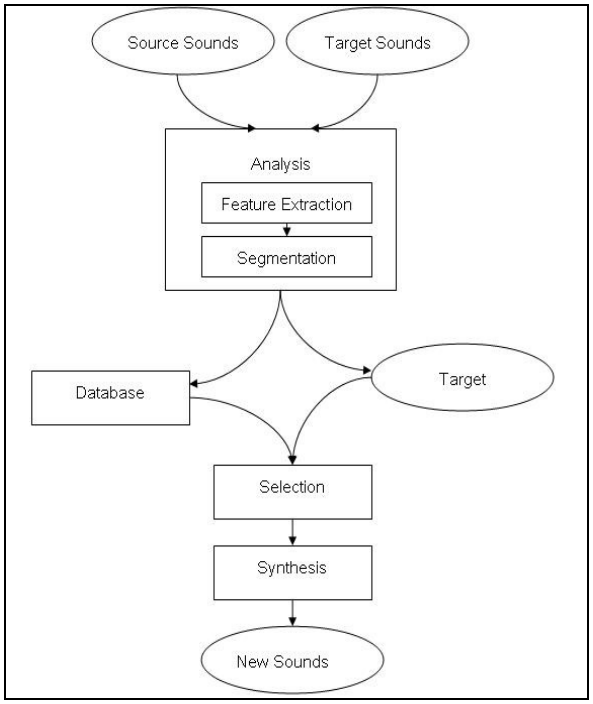


Figure 1. The dataflow of a typical CSS system.

At the start, both target and source sounds need to undergo some form of audio analysis that usually includes sub-processes such as feature extraction and segmentation. Audio information of the source sounds are then stored in the database whilst the information on the target sound is used to find matching sound units. Unit are then concatenated together during synthesis. Control can be offered at every level of these processes, i.e. segmentation, feature extraction, unit selection and synthesis.

Although existing CSS systems are already offering ample options to users with regards to segmentation mode and audio features for analysis, little effort have been focused on providing users with the ability to prioritize the features in order of relevance. As the computational complexity is quite high during feature extraction, including only the more relevant features reduces this load, whilst at the same time giving users the flexibility in choosing which features to exploit.

Aptly, several existing CSS systems such as Mosievius [6], MATConcat [7], Soundspotter [8], Caterpiller [9] and CataRT [10] offer this control, enabling their users to experiment with numerous combinations of audio features. MATConcat offers a slightly higher level of flexibility where it not only lets users select which features to include, but also their order of importance. However, it does not take into account the weight of each feature with respect to one another.

Musical Mosaic [11], on the other hand, includes the use of weights, but its use is not targeted on differentiating the importance between features, but as a mean to prioritize different cardinal rules, i.e. it is three times more important to obtain the correct pitches than to obtain all unique samples. Furthermore, the weights are assigned manually by users, which can be arbitrary and may result in some form of inconsistency.

Currently, MATConcat and Musical Mosaic are the only CSS systems that let users prioritize the order of the features, but this can only be done by manually. Automating this process is a highly appealing prospect, and the following section proposes a method to achieve this.

3. METHODS

The experimental setup and methods used in developing a weighted order dependent feature selection mechanism is described as follows.

3.1. Dataset

The study was performed using two different sets of corpus – source and target. The source sounds that were stored in the database came from a collection of 270 classical tracks, segmented based on their perceptually relevant onset times to give 9906 units of small, non-uniformed audio segments in total. Target sounds were compiled from recordings of music from several different genres and also included recordings of natural sounds, which was also segmented prior to finding a matching source sound. Segmentation of both corpora is derived from the work of [12].

3.2. Feature Extraction

As a preliminary study to illustrate the significance of prioritizing audio feature selection in CSS, only five basic feature-vector dimensions were currently implemented; spectral centroid, spectral rolloff, zero crossing rate, beat and pitch. These features were selected as they can be used to extract the general characteristics of a sound, covering the timbral, beat and pitch elements. Individually, they represent the brightness of a sound, the skewness of the spectral distribution, the noise level in a sound, the pulse and the frequency of in a sound segment [13].

3.3. Weighted Audio Feature Selection

It is already understood that in order to save time and space, only the more relevant features are extracted. However, between the selected features themselves, a rank of importance may exist i.e. it may be twice as

important to find the units in the source database that match the spectral centroid value of the target unit than it is the zero crossing rate. A weighted order dependent feature selection mechanism is thus needed in order to make an organised and prioritized decision.

This can be achieved through the use of the Analysis Hierarchy Process (AHP), a decision making method of measurement through pairwise comparisons and relies on the judgments of experts to derive priority scales [14]. Using a fundamental scale of importance as a guide, users assign one of the five available states to each feature (see Figure 2).

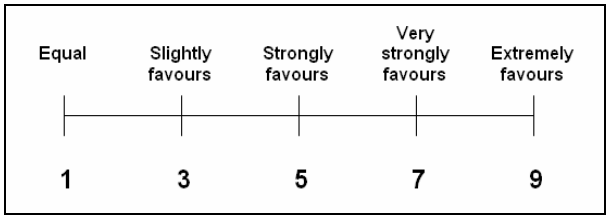


Figure 2. The fundamental scale of importance.

From here, the weights for each feature can be generated automatically. First, the pairwise comparison matrix is tabulated, where the number of criteria (features) is assumed to be n, giving $A_1, A_2,\ldots,A_n$ respectively with corresponding weights $W_1, W_2,\ldots, W_n$. (See Equation 1).

$$A = [a_{ij}] = \begin{bmatrix} W_1/W_1 & W_1/W_2 & \cdots & W_1/W_n \\ W_2/W_1 & W_2/W_2 & \cdots & W_2/W_n \\ \vdots & \vdots & \vdots & \vdots \\ W_n/W_1 & W_n/W_2 & \cdots & W_n/W_n \end{bmatrix}$$
$$\text{where } a_{ij} = W_i/W_j,\ a_{ji} = 1/a_{ij},\ i,j = 1,2,\ldots,n \tag{1}$$

To get the numerical ranking of the features, the sum of each column in the matrix (the Eigenvector of Matrix A) is calculated. To obtain the normalized relative weight, each element of the matrix is then divided with the sum of its own column. The average of all cells in a row of Matrix A is then summed up to get the priority vector, W, which is also the weight of the feature (see Equation 2).

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix}, i,j = 1,2,\ldots,n \tag{2}$$

Further reading on the mechanism of AHP can be found in [14] and [15].

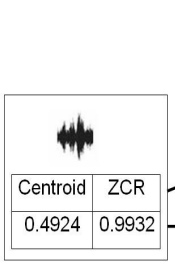
3.4. Unit Selection Search Methods

The k-Nearest Neighbour (KNN) is the search method used this study. It has already been used in a few CSS systems [6], [9]. It is most advantageous when little prior knowledge is known about the distribution of the data, but strong consistency in the result is required. Its fairly simple algorithm is allows quick computation, an ideal criteria for a search method was the basis of its implementation in this study.

4. DISCUSSION

The significance of prioritizing audio features selection in CSS is illustrated in Figure 3 of this section. The case of example needs to find a match for a target sound that is based on two features; centroid and ZCR. No exact match is found for those two feature values in the database. Using the fundamental scale of importance, it has been specified by the user that it is more important to find a match for the centroid value than it is the ZCR. A weight value of 0.8 is assigned to the feature centroid through the methods described earlier in Section 3.3.

A standard Euclidean distance calculation will return Segment 2 (dotted line) as the closest matching segment, as it has the smallest sum of features distance between the two. However, the method proposed in this study has taken into account the weight of each feature based on their importance as well as the distance from target, hence returning Segment 1 as the closest matching segment (bolded line). Although the numerical difference between the two instances is quite small, it can significantly affect end product of the concatenation, as the selected segment will be synthesized into sound. Therefore, the order of the features does affect the overall outcome of concatenation.



	Centroid	ZCR	Sum of features distance	Sum of weighted features distance, W=0.8
1	0.4927	0.9847	0.0087	0.0193
2	0.4991	0.9930	0.0069	0.0539

Figure 3. Comparison between the results returned for standard audio features match and weighted audio features match.

5. CONCLUSION

A CSS system with good controls allows users to specify desired criteria of their targets. In return, this provides a clearer description to the system on what needs to be searched. Giving users the flexibility of choosing different combination of features, prioritizing their orders and assigning weights to each of them during the segment matching process are small steps taken towards extending the user control in CSS. This is especially useful since it is shown that the order and weight of the features selected can affect the overall concatenation result. A method to generate weights automatically

through AHP is also suggested in this paper. The notion of allowing users to fine tune the constraints is hoped to be able to boost the value and potentials of general CSS systems.

Future works involve studying other search methods that have proven to work well in other domains such as text search or speech synthesis, and observes its performance in music synthesis. An interactive user interface for CSS system that promotes ease-of-use, higher level of user control is also being designed and developed.

6. REFERENCES

[1] Tran, N. Generating photomosaics: an empirical study. *In Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 105-109.

[2] Saaty, T.L. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9-26, 1990.

[3] Oxford Dictionaries. Compact Oxford English Dictionary of Current English, 2008.

[4] Krichmar, A. Command language ease of use: a comparison of DIALOG and ORBIT. *Online Information Review*, 5(3):227-240, 1993.

[5] Dunlop, M.D., Johnson, C.W. and Reid, J. Exploring the layers of information retrieval evaluation. *Interacting with Computers*, 10(3):225-236, 1998.

[6] Lazier, A. and Cook, P. MOSIEVIUS: Feature driven interactive audio mosaicing. *In Digital Audio Effects (DAFx)*. Citeseer, 2003.

[7] Sturm, B. L. MATConcat: an application for exploring concatenative sound synthesis using MATLAB. *Proceedings of Digital Audio Effects (DAFx)*, Naples, Italy, 2004.

[8] Casey, M. Soundspotting: a new kind of process?, 2009.

[9] Schwarz, D. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3-22, 2006.

[10] Schwarz, M. A system for data-driven concatenative sound synthesis. *In Digital Audio Effects (DAFx)*, pages 97-102. Citeseer, 2000.

[11] Zils, A. and Pachet, F. Musical mosaicing. *In Digital Audio Effects (DAFx)*. Citeseer, 2001.

[12] Brossier, P. (2006). Automatic annotation of musical audio for interactive systems. PhD thesis, Centre for Digital music, Queen Mary University of London.

[13] Tzanetakis, G. and Cook, P., 'Musical genre classification of audio signals', *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, 293-302 (2002).

[14] Saaty, T.L. and Sodenkamp, M. Making decisions in hierarchic and network systems. *International Journal of Applied Decision Sciences*, 1(1):24-79, 2008.

[15] Lin, C.T. and Wu C.S. Selecting a marketing strategy for private hotels in Taiwan using the analytic hierarchy process. The *Service Industries Journal*, 28(8):1077-1091, 2008.