Generalized diphone control is a powerful means of building a musical phrase from dictionaries of analysed sound units by concatenating and articulating them. The Diphone program, developed at IRCAM for diphone control, has been improved with fundamental frequency, noise components, spectral envelopes and parallel sequences. A new graphical user interface for the Macintosh is presented. It communicates with Diphone through a simple command language. It handles dictionary management, the building and the articulation of parallel sequences of diphones, and the querying of parameter values.

1. Introduction

Generalized diphone control has been presented in [Rodet 93]. A spoken sentence can be modelled as a succession of transient sounds (called diphones). Any sentence can be reconstructed from a dictionary of diphones. In order that duration and fundamental frequency can be modified, an analysis-synthesis model, such as source-filter [Rodet 88], is used. Therefore, it is not the diphone sound signal itself which is stored in the dictionary, but the corresponding control signals of the analysis-synthesis model. A sentence is obtained by concatenation of control signals of a sequence of diphones. In [Rodet 93] we have extended the concept of diphone control to musical sounds in general. A dictionary can include any segment of sound considered as an atom for the musical usage in view. Such an atom, in a representation like additive representation or source-filter representation, is called a segment. Diphone control does not rely on a particular synthesis technique. We have focused on the additive model, i.e. a sum of sinusoidal partials with time-varying frequencies and amplitudes defined at frame time. A segment is a data structure containing, in the additive case, the frame times and the associated parameters for a segment of sound. A musical phrase is obtained by concatenating basic segments to produce a new segment from which a sound signal is computed by the additive synthesizer (Fig. 2). In [Rodet 93], we have detailed the Diphone program written at IRCam for diphone control and synthesis. We now present recent improvements to diphone control and synthesis, and a Graphical User Interface with novel characteristics, both from the conceptual and the implementation point of view.

2. Improvements

A segment has to contain also its original (time-varying) fundamental frequency, say Fp(t), since a desired fundamental frequency trajectory, say Gf(t), is obtained by applying the transposition factor Gf(t)/Fp(t) to the sinusoidal partial frequencies (Fig. 2). For music written in terms of notes with well defined pitch, the segment paradigm applies nicely to each segment with constant written pitch. Since notes can be largely independent of phones or other timbre attributes, a musical phrase has to be defined by two sequences of segments, one for the pitch, the other for timbre attributes such as phones. We say that these two sequences are parallel (Fig. 3) and that neither one should necessarily impose its metric structure on the other. On the contrary, vibrato tends to be synchronized on notes. In consequence, vibrato frequency and excursion can be defined by the same segments as pitch or by sub-segments of pitch segments. Finally, Gf(t) is computed by applying the defined vibrato to pitch values given by pitch segments. Other articulations, such as portamento or loudness are implemented in the same way.

Random components of sounds, like flute noise or voice fricatives, are not correctly represented as sinusoids with parameters recorded in a segment, but can be represented as white noise filtered through a time-varying spectral envelope. To be used at the synthesis stage, the values of the noise spectral envelope at all frame times are also stored in a segment (Fig. 2). In a preliminary stage, dictionaries of diphones, i.e. of segments, have to be constituted (Fig. 1). First, an additive-noise analysis [Depalle 93] is performed on the sound recordings. Secondly, the analysis data are segmented according to the segment time limits chosen by the musician. Finally the segments are stored in dictionaries. Another improvement brought to the Diphone program is the use of spectral envelopes for sinusoidal partials as well. A source-filter model is well suited for certain classes of sounds such as the voice. In this case, sinusoidal partial amplitudes are determined by the value of the spectral envelope at the frequencies of the partials. At the stage of the synthesis of a segment, these amplitudes have to be recomputed when the fundamental frequency is modified. Therefore, the spectral envelopes of sinusoidal partials have to be stored also in a segment and used at the synthesis stage to compute the amplitudes of the partials (Fig. 2). Finally, the Diphone concatenation and articulation program is given a textual interface in the form of a simple Command Language opening access to all its facilities. In this way, Diphone can easily be used in a routine without being an absolute user of a graphical interface. This is usually platform-dependent.

Rodet & Lefèvre

ICMC Proceedings 1996
3. A Macintosh Graphical User Interface

A Macintosh Graphical User Interface (GUI), named MacDiph, has been built to provide an easy access to the Diphone program. It is being tested on PowerPC and Macintosh-68K platforms. It is written in C++, compiled with Metroworks-CoderWarrior IDE v.4, and built on the Metroworks-PowerPlant set of classes. From these classes, two groups of classes have been derived. They are not specific to MacDiph but designed for general graphical programming. The first group is aimed at displaying and editing graphs and objects, such as diphone sequences. The second group of classes is aimed at displaying and editing tree structures, such as dictionaries of diphones and their constituents, i.e., instrument segments, composite segments, and basic segments. Finally, a break-point function editor is being built for the control signals contained in segments. The different tools provided by these classes are, as much as possible, compliant with the Macintosh Human Interface Guidelines (Inside Macintosh). In particular, they offer copy, cut, paste, as well as drop and drop functions, and follow the Macintosh guidelines.

We have taken care to separate the GUI from the Diphone program itself. MacDiph communicates with Diphone by using the Command Language mentioned above. This permits also to have a version of Diphone running on a fast Unix platform and be driven, through the network, by MacDiph running on a relatively slow Macintosh. This connection is implemented with sockets. Segments also can be read and written by MacDiph for display and editing. Since segments can be large data structures and, on the contrary of the Command Language, cannot be handled by users, this is done through binary streams.

MacDiph provides the usual functions of a database for a set of diphone dictionaries, i.e., browsing through different dictionaries, displaying their content, modifying them, selecting instruments, and segments, constituting new dictionaries and saving them (Fig. 3). The drag-and-drop paradigm is used in order to move segments between dictionaries and to build various parallel sequences. A sequence can contain basic segment and sub-sequences. Segments and sequences are represented as graphical objects (Fig. 4) directly displaying their characteristics, i.e., duration, center, interpolation portion between successive segments, loudness and articulation speed. Click-and-drag allows easy change of these characteristics in a Wysiwyg style.

Parameter evolution, such as sinesoidal partial frequency or fundamental frequency evolution, as stored in segments, or as computed from a sequence, can be displayed and edited in graphical windows (Fig. 5) placed under the sequence windows for easy visualization of synthesis. Diphone and paste are fully supported on sequences and on parameter evolution.

4. Conclusion

MacDiph and Diphone constitute a promising tool for musicians. On the one hand, diphone control offers new possibilities for precise and powerful control of synthesis, which would not be obtained in another way. The ability to build and articulate complicated sequences of segments from diverse origins appears to be an attractive feature. On the other hand, MacDiph implements a new direct representation and handling of segments in terms of intuitive graphical objects. As opposed to discrete values, such as notes, the control of continuous quantities [Rode & 84] has always been a difficulty in computer music. MacDiph and Diphone should bring some improvements in that domain by establishing a close connection between discrete and continuous representations.

References


---

![Diagram](image)

**Fig. 1:** Analysis data are segmented into segments stored in a dictionary.

ICMC Proceedings 1996

---

337

Rode & Lefever
Fig. 2: Concatenation and synthesis of a sequence of segments

Fig. 3: Two parallel sequences and fundamental frequency evolution

Fig. 4: Management of a dictionary of instruments, segments and basic segments