



HAL
open science

Macintosh Graphical Interface and Improvements to Generalized Diphone Control and Synthesis

Xavier Rodet, Adrien Lefèvre

► **To cite this version:**

Xavier Rodet, Adrien Lefèvre. Macintosh Graphical Interface and Improvements to Generalized Diphone Control and Synthesis. ICMC: International Computer Music Conference, Aug 1996, Hong Kong, Hong Kong SAR China. pp.336-338. hal-01105434

HAL Id: hal-01105434

<https://hal.science/hal-01105434>

Submitted on 20 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Macintosh Graphical Interface and Improvements to Generalized Diphone Control and Synthesis

Xavier Rodet, Adrien Lefevre

Proceedings of the International Computer Music Conferences, Hong Kong, 1996
Copyright © Ircam - Centre Georges-Pompidou 1996

Abstract

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 on Macintosh is presented. It communicates with Diphone through a simple command language. It allows for the managing dictionaries, the building and the articulation of parallel sequences of diphones, and for the editing 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 times*. 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 $F_0(t)$, since a desired fundamental frequency trajectory, say $G_0(t)$, is obtained by applying the transposition factor $G_0(t)/F_0(t)$ 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) since none of each 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, $G_0(t)$ 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.

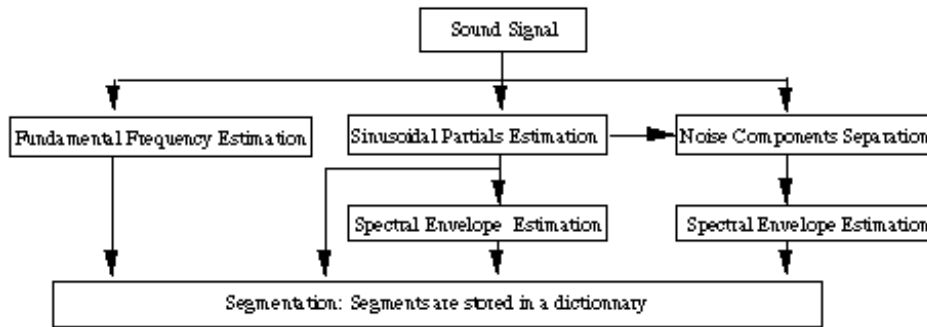


Fig. 1: Analysis data are segmented into segments stored in dictionaries

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).

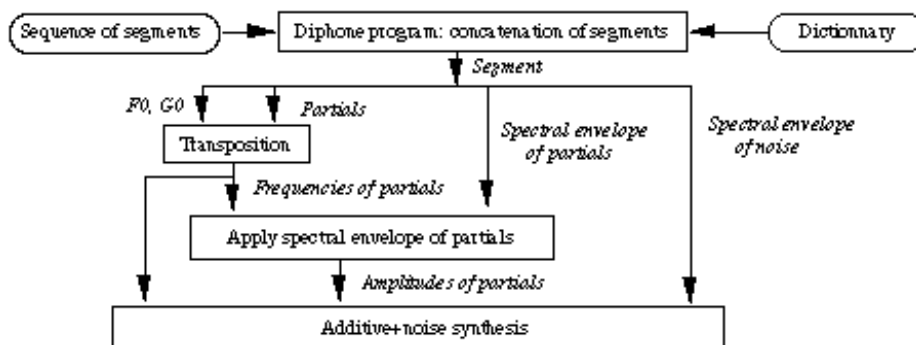


Fig. 2: Concatenation and synthesis of a sequence of segments

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 and tested separately and is totally independent of any GUI which is usually platform dependent.

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 tested on PowerPC and Macintosh-68K platforms. It is written in C++, compiled with Metrowerks-CodeWarrior IDE 1.4, and built on the Metrowerks-PowerPlant set of classes. From these classes, two groups of classes have been derived. They are not specific to MacDiph but designed for general graphic 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. instruments, 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 and paste, as well as drag and drop facilities, and follow the Wysiwig 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 huge 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 data base 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).

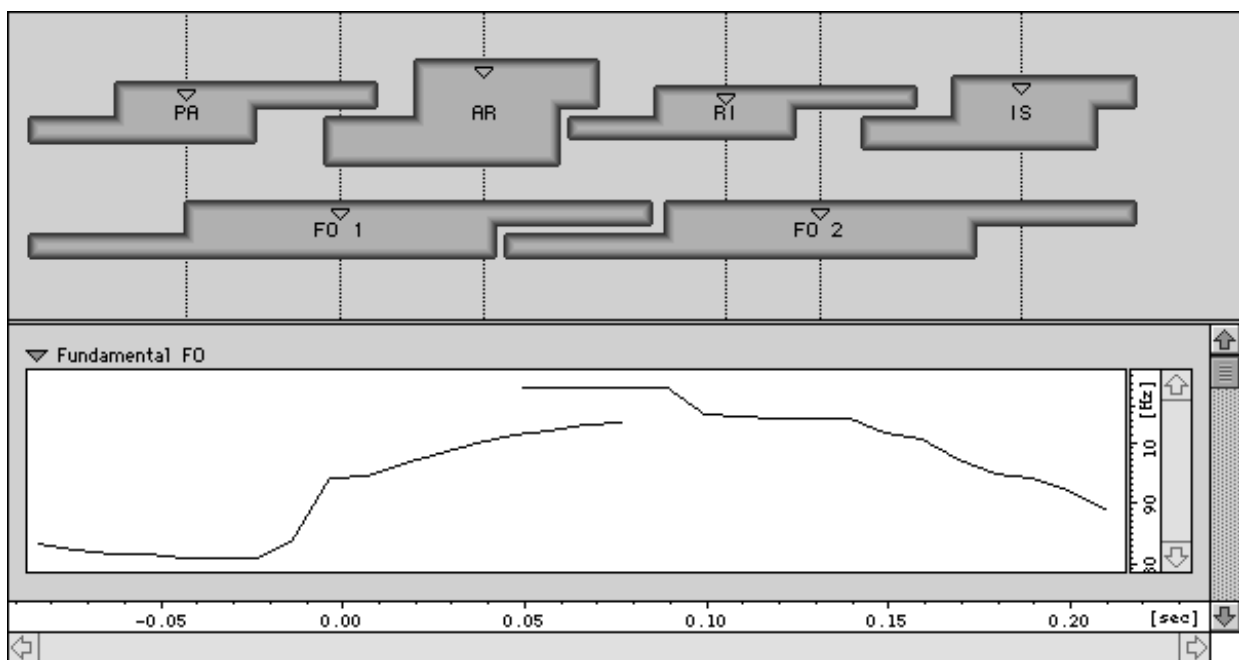


Fig. 3: Two parallel sequences and Fundamental frequency evolution

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 segments 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 Wysiwig style.

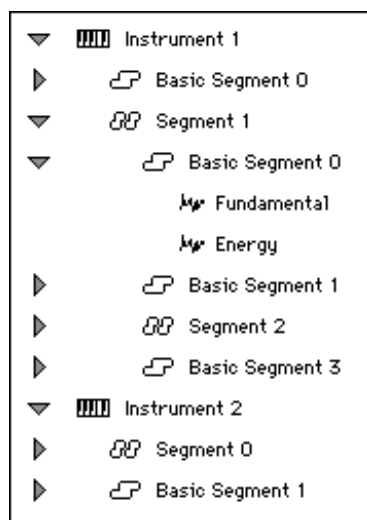


Fig. 4: Management of a dictionary of instruments, segments and basic segments

Parameter evolution, such as sinusoidal 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 visualisation of synchrony. Modify, cut, copy and paste are fully supported on sequences and on parameter evolution.

4. Conclusion

MacDiph and Diphone constitute a promising tool for musicians. On one hand, diphone control offers new possibilities for precise and powerful control of synthesis, which could 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 opposite to discrete values, such as notes, the control of continuous quantities [Rodet 84] has always been a difficulty in computer music. MacDiph and Diphone should bring some help in that domain by establishing a close connection between discrete and continuous representations.

References

[Rodet 84] X. Rodet, P. Cointe, *Formes: Composition and Scheduling of Processes*, C. M. J., MIT Press, Vol. 8, No 3, Fall 84.

[Rodet 85] X. Rodet, P. Depalle, *Synthesis by Rule: LPC Diphones and Calculation of Formant Trajectories*, IEEE-ICASSP, Tampa, Fl., March 85.

[Rodet 88] X. Rodet, P. Depalle, G. Poirot, *Diphone Sound Synthesis*, Int. Computer Music Conference, Koeln, RFA, Sept. 88.

[Depalle 93] P. Depalle, G. García & X. Rodet, *Tracking of partials for additive sound synthesis using hidden Markov models*, IEEE *ICASSP-93*, Minneapolis, Min., Apr. 1993.