CENTRE NATIONAL DE
GMEM
CREATION MUSICALE

Holophon

Documentation

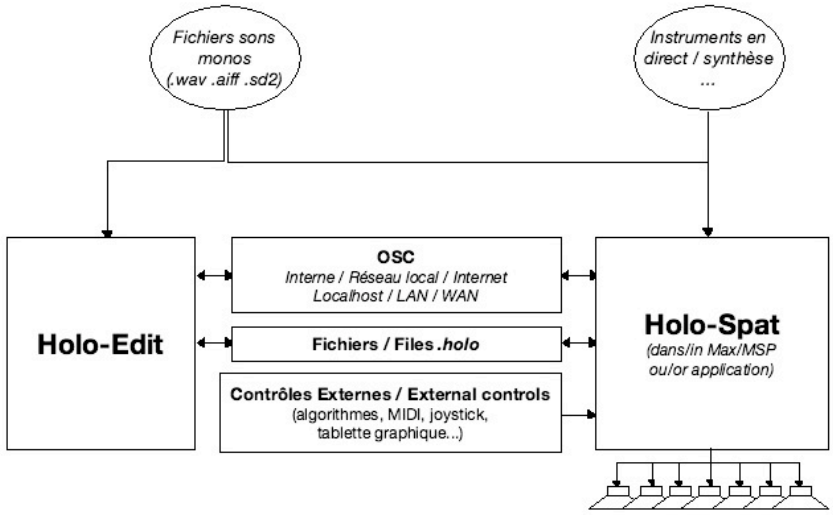
7th edition, July 2008 (version 4.2)

GMEM
Centre National de Création Musicale
15-17 rue de Cassis
13008 Marseille
Tél. : 04 96 20 60 10
Fax : 04 96 20 60 19
Email : holophon@gmem.org
http://www.gmem.org/

Introduction

Holophon is a set of tools for the programming and real-time manipulation of sound trajectories across different speakers. There are two different software packages:

- Holo-Edit: A graphical and algorithmic editor of sound trajectories. Holo-Edit makes it possible to draw and graphically edit trajectories across a complex sound system. It's also possible to program those trajectories with different automatic functions.
- Holo-Spat: A group of spatializers working in real-time on Max-MSP[1], allowing the user to spatialize several sound sources independently over an array of multiple speakers. The control of the source trajectories can be made automatically by either using *.holo* files (made with Holo-Edit), directly by Holo-Edit using OSC[2] or directly by various means in Max-MSP.



Holophon v4.0 – System Overview

Despite the care taken to design and develop these tools and to write the documentation, some errors could persist. Please share your remarks by writing to the following email address : holophon@gmem.org. If it appears useful to us, we will make the necessary modifications to the software, but this is without any guarantee of time, except in the case of severe problems preventing their use.

Programmers willing to use whole or part of the source code for one of these software packages should not hesitate to contact us at the same address. Given the complexity of the code, we cannot guarantee that one will easily find the section of interest.

[1] http://www.cycling74.com/
[2] Open Sound Control http://cnmat.berkeley.edu/OpenSoundControl/

Summary

Installation

I. Requirements

Holo-Edit is based on the Java 1.5 programming language and Jogl[3]. This implies that it will only be able to work under operating systems compatible with java Java 1.5 and Jogl. (Mac Os X **Tiger**, Windows XP)

Holo-Spat is based on the Max/MSP 4.5 software/programming language. This implies that it will only be able to work under operating systems compatible with Max/MSP 4.5. (Mac Os X et Windows XP).

People wishing to modify Holo-Spat must own a licensed copy of Max/MSP. For the other package, a standard application is available with versions for Apple PowerPC, Apple Universal Binary and Windows XP.

[3] Java bindings for OpenGL https://jogl.dev.java.net/

II. Installation

A. Apple Mac Os X Tiger

Check via software updates that you have the latest java version installed.
Copy Holophon’s folder to the directory of your choice.

B. Microsoft Windows XP

Check that you have the latest java version installed at the following address:
http://www.java.com/en/download/installed.jsp
Download the Bonjour installer for Windows at the following address:
http://www.apple.com/macosx/features/bonjour/
Copy Holophon’s folder to the directory of your choice.

C. Firewall configuration

For correct communication between Holo-Edit and Holo-Spat, it is necessary to open the following UDP ports on your firewall: 5353 (Bonjour), 13008 (Holo-Edit), 13005 (Holo-Spat).

D. Max/MSP Configuration

The following steps are only necessary if you own Max/MSP and wish to modify or use the extended functionality of Holo-Spat.

At the end of the file *max.java.config.txt* (in Cycling’74/java/), add the following line:

max.java.jvm.version 1.5

Add holoedit.jar (in *Holo-Spat/src/To_C74_java_lib*) to Cycling’74/java/lib or add the following line to the file *max.java.config.txt*:

max.dynamic.jar.dir [absolute path to the folder containing holoedit.jar].

Add the *To_Max_Folder-XXX* to Max’s *Files Preferences…* or move/copy this folder to the Cycling’74 folder.

Holo-Edit

I. Introduction

A. Principles

Holo-Edit allows for the precise positioning of multiple sounds in time and space (defined by a set of speakers).

In order to do so, it associates sounds to trajectories (a set of points defined by their position in space (x, y, z) and their date).

The software is a set of graphical editors and algorithmic functions for creating and manipulating sounds in space.

B. Definitions

- Point: a point in space defined by its coordinates (Cartesian x, y, z or polar ray, angle) and its type: editable or non-editable.
- Trajectory: a set of successively dated points.
- Waveform: a visualization of a sound in time, (the amplitude curve of this sound.)
- Track: a set of successive trajectories associated with waveforms.
- Session: a set of tracks.

II. Editors and their functionalities

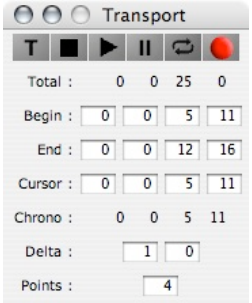
The following windows share the same set of data, i.e., the same session. Two different sessions cannot be opened at the same time.

Each editor offers a visualization of the set of data from a different point of view.

They all share what is called the **time selection**, or the portion of the session to visualize. This makes it possible to work only on a given interval of time without viewing the totality of a session.

??But they don't share object's selection.

A. Transport



Total : 0 0 25 0
Begin : 0 0 5 11
End : 0 0 12 16
Cursor : 0 0 5 11
Chrono : 0 0 5 11
Delta : 1 0
Points : 4

The number boxes in this window allow the precise handling of time in the software.

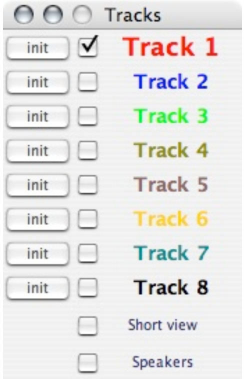
Total indicates the total duration of the session.

Begin and **End** respectively indicate the upper and lower limits of the time selection. **Cursor** indicates the position of the cursor in the *Score*. **Chrono** displays the time while playing. Times are expressed in hours / minutes / seconds / 1/100th seconds.

Delta is used to set the time (seconds / 1/100 seconds) between the new point you want to create with the mouse and the last point of the track. **Points** determines the number of intermediate points which will automatically be created in-between.

There are also traditional transport commands (stop / play / pause / loop / record).

B. Tracks



init ☑ Track 1
init ☐ Track 2
init ☐ Track 3
init ☐ Track 4
init ☐ Track 5
init ☐ Track 6
init ☐ Track 7
init ☐ Track 8
☐ Short view
☐ Speakers

This window allows the management of tracks.

Each track is named and numbered.

The name of a track is for the user's reference; its number is a unique identifier for the machine or for more complex applications (see *V.C. Track Numbering*).

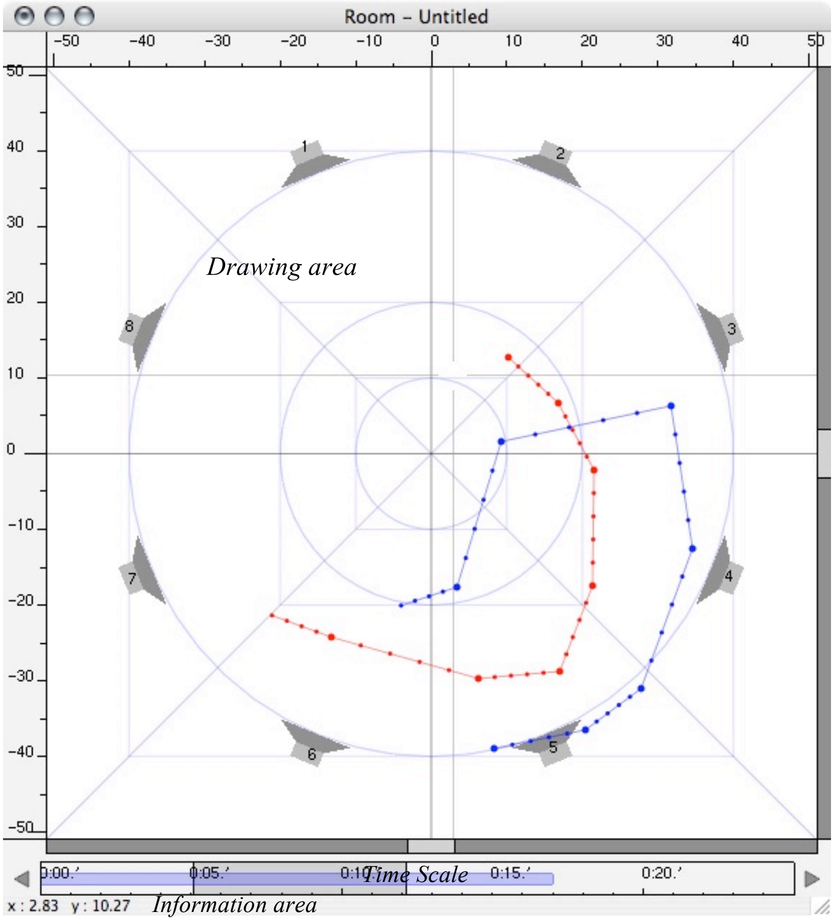
The cross below the name indicates whether or not this track is visible. This will affect all editors except the *Score*.

The active track is written in bold. This is the track that will be modified by the user interface and the menus (copy, paste…).

The **Short view** mode modifies the time selection so that the editors only display the last point of the active track in the real time selection (+/- **delta**).

Speakers allows the handling of speakers.

C. Room Editor

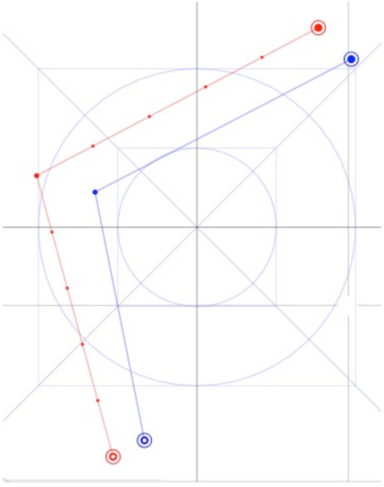


This editor offers a top view of the space and the trajectories contained in the time selection.

Shown here are all the checked tracks in the *Tracks* window having content in the time selection.

Here we handle trajectories and speakers from a spatial point of view (move, add/remove points…).

a. Adding points



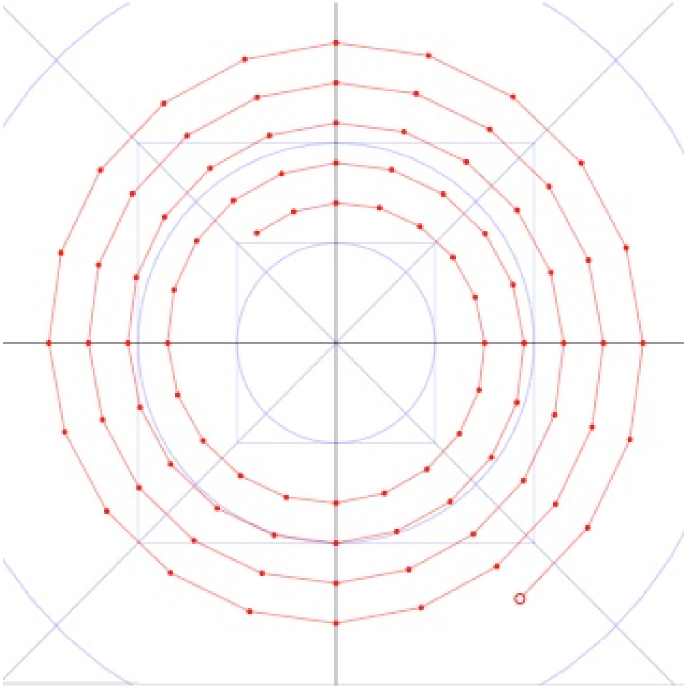
To add points defining trajectories in Holo-Edit, select the track you want to add points to in the *Tracks* window. It is now visible and active.

Before adding a point, ensure that values for **delta** and **points** have been entered in the *Transport* window.

To add a point, click in the drawing area at the desired position while holding the Command/Ctrl key (Max/Win) down. A new segment appears. If *Points* = 0, no intermediate points will separate this new point from the previous one, and sound will jump from one position to another, see illustration below.

Intermediate points are referred to as **non-editable**, as opposed to the larger points on the screen, which are **editable**.

The first point in a track is represented by a filled circle surrounded by an empty one. The last one is represented by an empty circle surrounded by another. If the last point of a track is not visible, the last drawn point appears empty, helping the user to visualize the direction of the trajectory as in the example below.



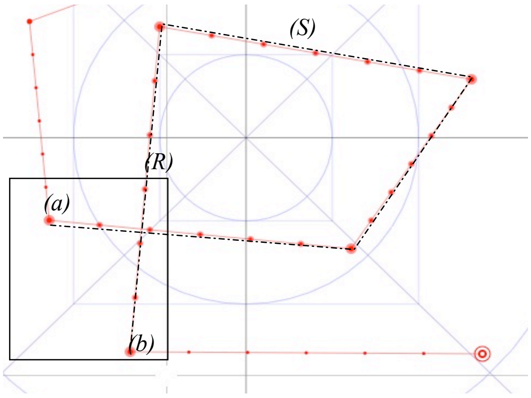
If the time selection is empty on the active track when we insert a new point, only one point is created (no intermediate points) at the *Begin* date. This represents the beginning of a new trajectory.

b. *Selecting several points*

To select a point and modify it, first make sure that its track is active, or that the option *Automatic track selection* is activated and then click on that point.

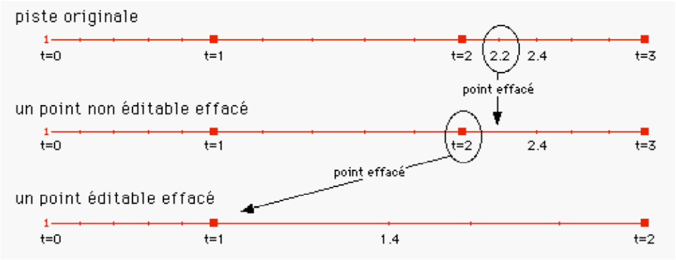
To select several points, there are two possibilities:

- By default, the selection will include all points between the first *(a)* and last *(b)* editable points contained in the selection rectangle *(R)*. The selection in this case is all points on the segment *(S)*.
- If the Shift key is held down while dragging the selection rectangle, only points in this rectangle *(R)* will be included in the selection.



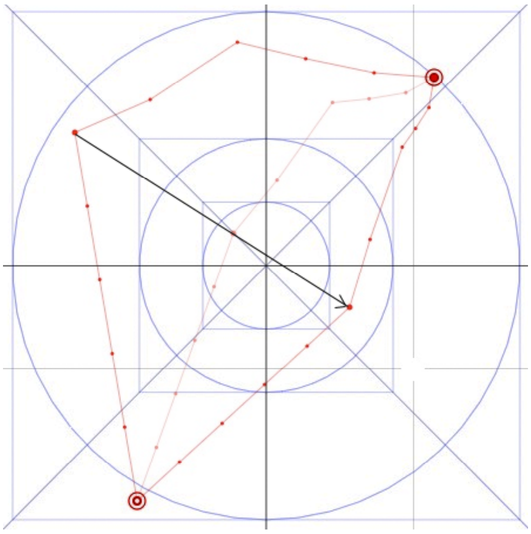
c. Removing points

To remove a point, click on it while holding the Alt key down. Removing an editable point erases all non-editable points preceding it and shifts the date of all following non-editable points.



d. Moving points

While dragging an editable point with the mouse, its movement distorts the segments surrounding it.



If the Command/Ctrl (Mac/Win) key is held down while moving a point or a selection, the motion will be constrained to the X-axis.

Conversely, if the Shift key is held down while moving a point or a selection, the motion will be constrained to the Y-axis.

To move a point or a selection along the Z-axis, hold Command/Ctrl (Mac/Win) + Shift down while dragging. The new Z value is displayed at the bottom of the window. The same rules apply to this motion as to the motion along the X/Y-axis. Use the *Time Editor* to visualize it or to modify it more precisely.

e. *Inserting and changing point types*

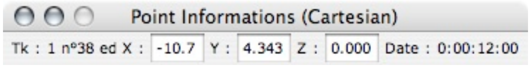
While holding down the Command/Ctrl (Mac/Win) key:

- Clicking on a segment inserts a new point on this segment.
- Clicking on a point makes it editable if it was not before (and vice-versa).

f. *Point Information*

The information area below the drawing area quickly gives information on the point under the cursor.

Right-clicking, or clicking while holding the Ctrl key down (Mac) opens a new editor where this point's position can be numerically modified. These modifications are confirmed by pressing the Enter key.



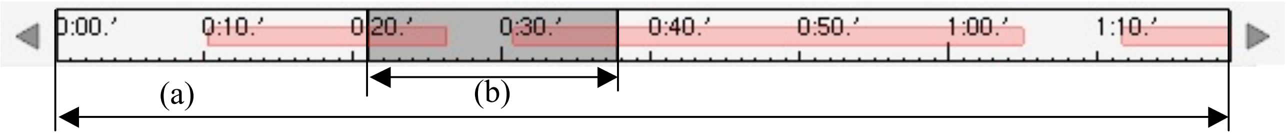
Point Informations (Cartesian)

Tk : 1 n°38 ed X : -10.7 Y : 4.343 Z : 0.000 Date : 0:00:12:00

g. *Handling speakers*

If the *Speakers* mode is activated in the *Tracks* window, speakers can be added/removed/moved in similar way (Command/Ctrl + Click to add a speaker, Alt + Click to remove it, and, while moving, Command/Ctrl (Mac/Win) to move it only along the X-axis, Shift to move it only along the Y-axis, Command/Ctrl + Shift move it along the Z-axis).

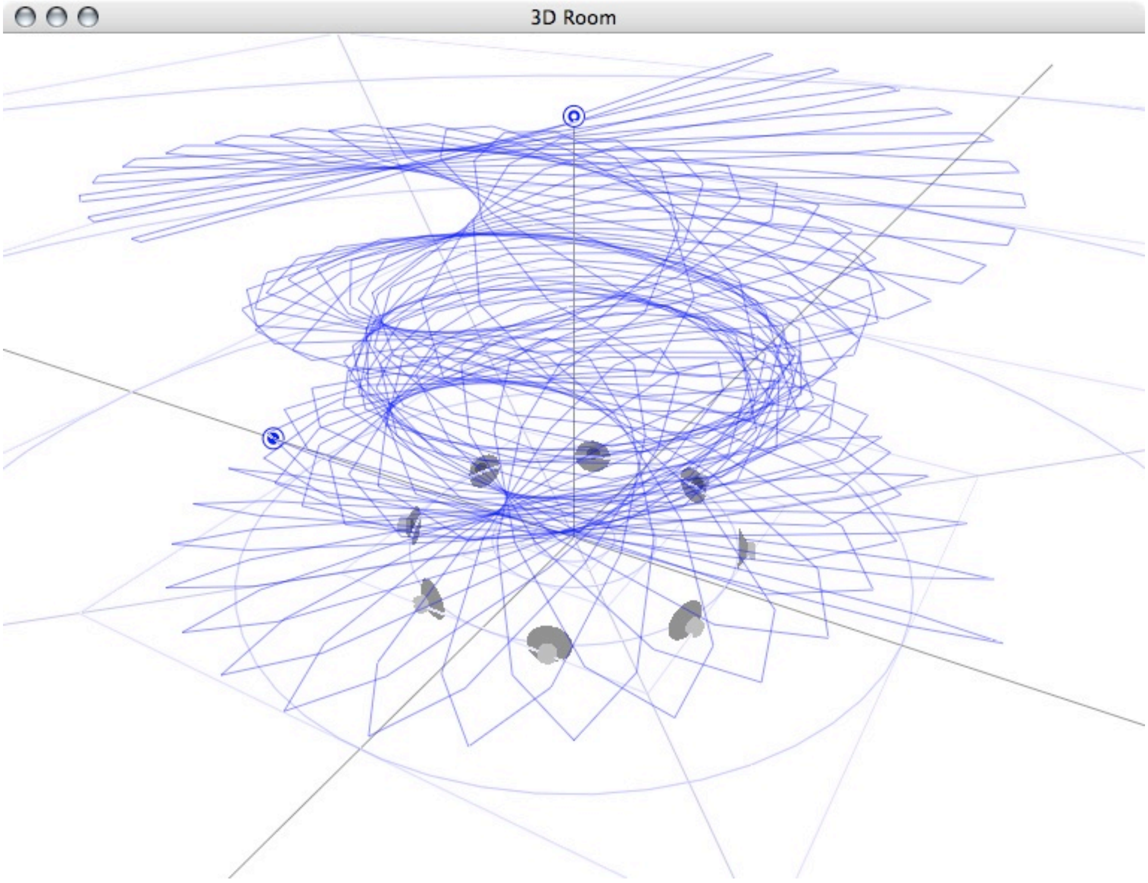
h. *Time scale*



The time scale below the drawing area displays the temporal information of the active track.

(a) Total duration of the session.
(b) Time selection.

D. 3D Room

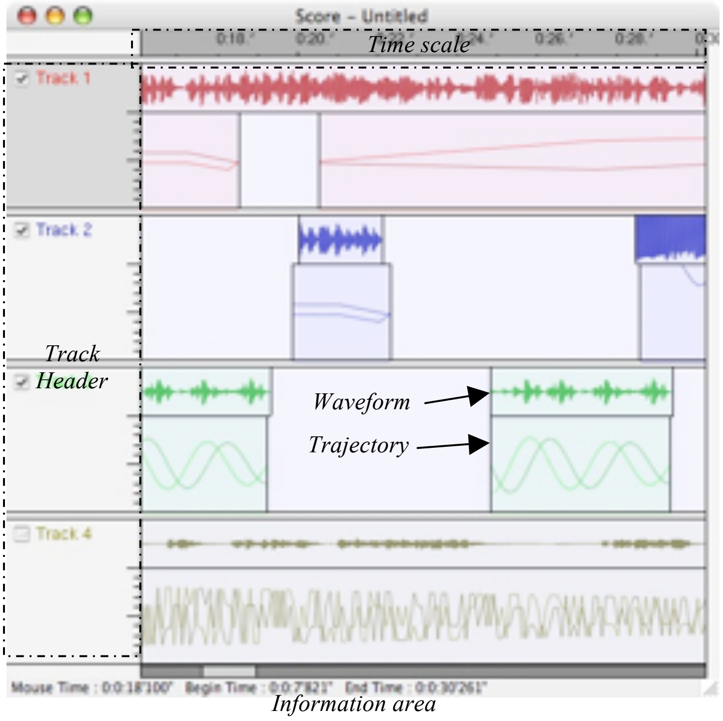


This window offers a three-dimensional view of the tracks. It is useful in visualizing modifications like elevation of the points and speakers. No editing is possible in this window.

E. Score

This window represents the score of the session. It displays trajectories as blocks, (condensed views of the x, y and z curves over time) as well as the data and the waveforms of the sound files associated with those trajectories.

Trajectories, waveforms and data appear here as graphical objects that are easily editable as in traditional audio mixing softwares: They may be moved, duplicated, stretched/compressed, split…



Tracks are arranged under a time scale, each of which has a header that shows the status of its activity and visibility.

a. Selection

Dragging in an empty space creates a selection. All trajectories and waveforms contained (at least in part) in the selection's rectangle are selected. If the Command/Ctrl (Mac/Win) key is held down while dragging, only trajectories are selected. If the Shift key is held down while dragging, only waveforms are selected.

Selected blocks appear darker.

To cancel a selection, click in an empty space.

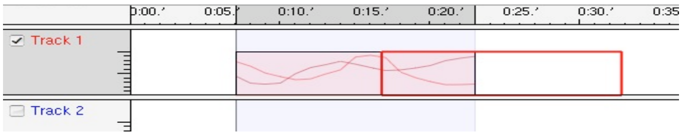
b. View and Zoom

The View function of the *Score* menu modifies the time selection, which is displayed in the other editors. The Zoom function modifies the view of this window.

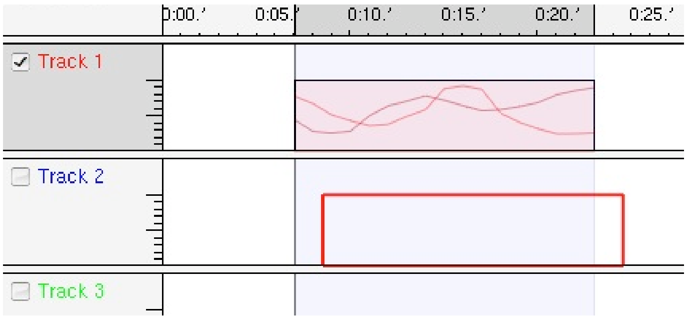
c. Moving

(The following behaviors are valid for selections of one or several waveforms, trajectories or datas)

- Moving objects within tracks: If the Alt key is held down while moving, the object is duplicated. If the Shift key is held down while moving, the motion will be more precise.



- Moving objects between tracks: If the Alt key is held down while moving, the object is duplicated. If the Shift key is held down while moving, the motion will be more precise. If the Ctrl key is held down while moving, the object will move to another track and keep the same time.



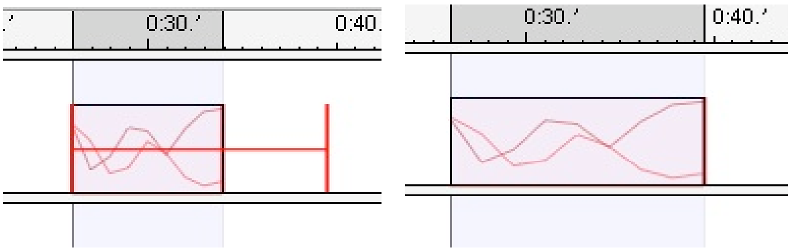
- It's also possible to numerically enter a new start time for the object/selection with the *Move* function in the *Score* menu.

d. *Transformations*

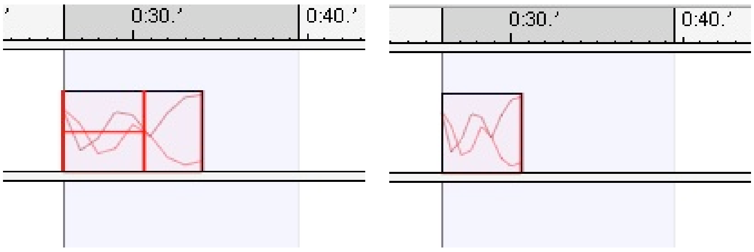
(Valid for selections of one or several waveforms , trajectories or datas)

- Stretching/Compression

If the side of a trajectory is pulled, the trajectory will be stretched or compressed to correspond to the new time. In the following examples, the trajectory is modified by pulling its right side, or end. The left side of the example shows the operation in progress, and the right side it's result. It is possible to perform the same operations on the start point of the trajectories, as well.

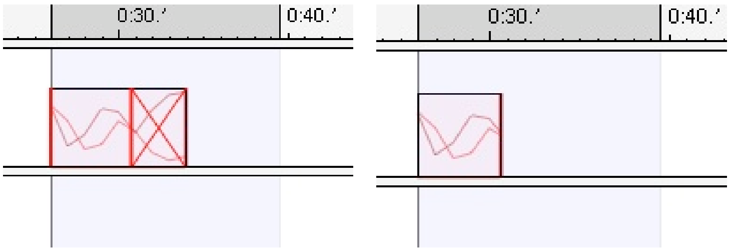


Stretching



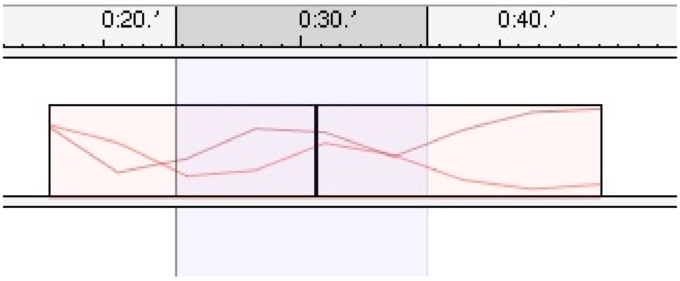
Compression

If the Alt key is held down while compressing, then the part marked by a cross will be cut from this trajectory.

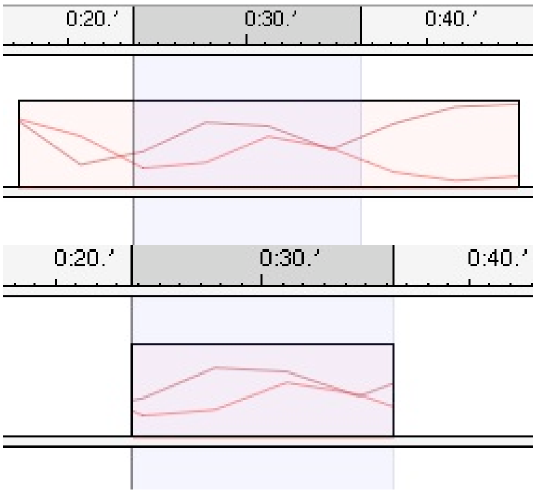


Cutting a part

- Division

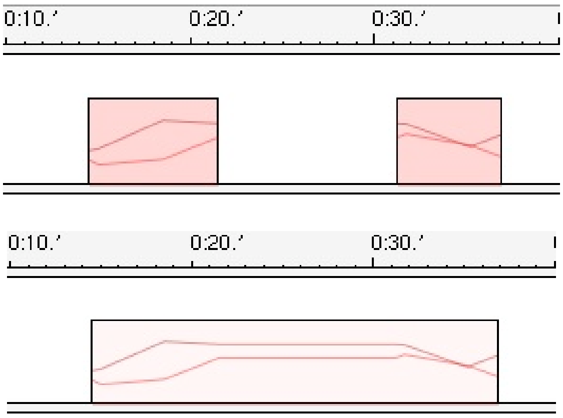


Clicking on a trajectory while holding the Command/Ctrl (Mac/Win) key down will cut the trajectory in two at the time corresponding to the mouse position.



It is also possible to trim a trajectory to the current time selection with the *Trim* function from the *Score* menu.

- Joining

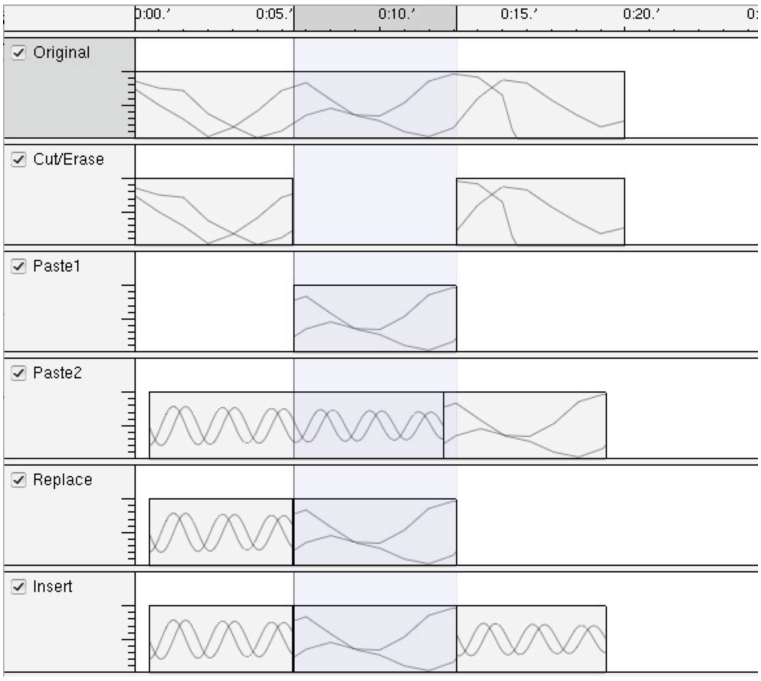


The Join function from the *Score* menu joins two different trajectories. No points will be added to the trajectory.

e. Cut/Copy/Paste/Insert/Replace…

The functions *Cut, Copy, and Replace Paste, Insert and Erase* can be found in the *Edit* menu, the *Score* menu or from the contextual menu (accessible from a right-click on a trajectory, a waveform, a data, a selection or the background of a track). These different methods offer different possibilities. The default behavior (*Edit* menu) will act on the content of the active track between *Begin* and *End*.

- *Cut* erases the content of the active track or the selection between *Begin* and *End* and places it in the clipboard.
- *Copy* places the content of the active track or the selection between *Begin* and *End* in the clipboard.
- *Paste* places the content of the clipboard at the end of the active track.
- *Replace* places the content of the clipboard at the *Begin* date of the active track and erases what was previously at this time.
- *Insert* inserts the content of the clipboard at the *Begin* date on the active track and shifts what was previous at this time from the size of the clipboard.
- *Erase* erases the content of the active track between *Begin* and *End*.



f. Adding waveforms

- Dragging and dropping a sound file on the score will import this sound into the sound library and places it at this point in the score.
- It's also possible to drop an element from the *Sound Pool* list and to place this waveform more precisely in the score.
- Another method consists of calling the *Import Waveform* function from the *Score* or contextual menus.

g. *Adding datas*

Dragging and dropping a data from the *Sound Pool* on the *Score* places it at this point in the *Score* (see §.F for more information about importable data files).

h. *Removing trajectories, waveforms or datas*

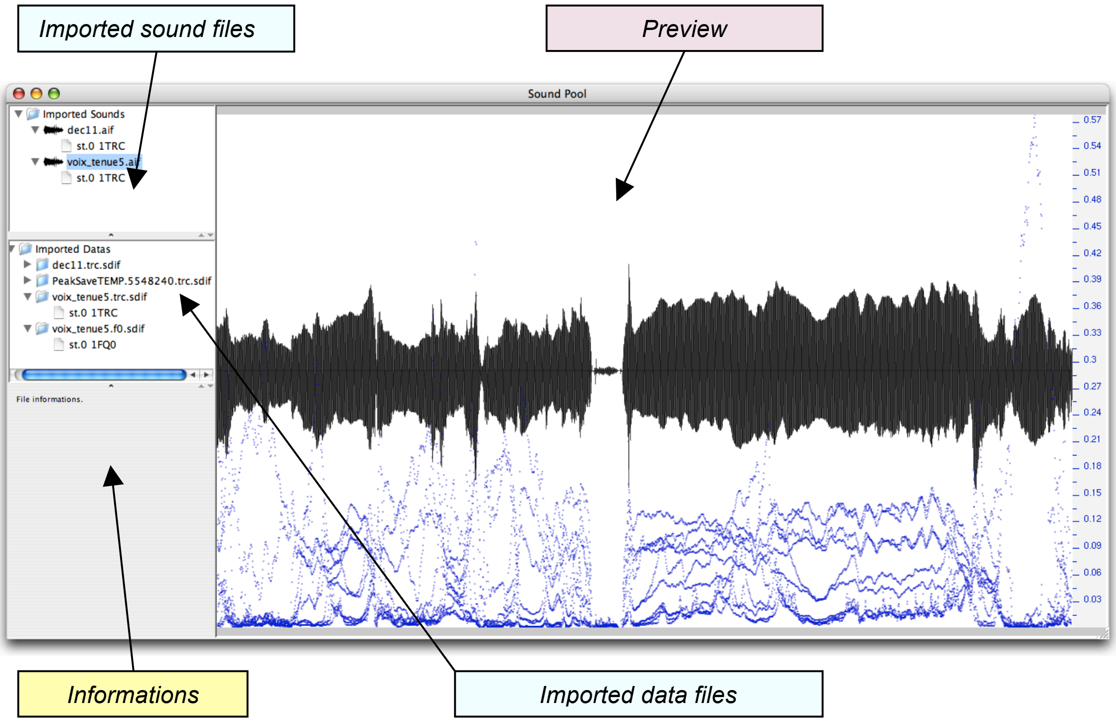
Holding the Command/Ctrl (Mac/Win) + Shift key down while clicking on an object or selection deletes it from the session.

i. *Editing Trajectories*

Double-Clicking on a trajectory opens the *Time Editor* for this trajectory.

F. Sound Pool

This editor lists the different data and sound files loaded in the session, providing file information and a preview of the data and waveforms of the sounds.



a. Sound file formats

- For now, Holo-Edit can only import and handle mono files. However multi-channel sound files will be considered as mono ones (the first channel for multi-channel files, the left channel for stereo ones).
- The sound file formats supported by Holo-Edit are AIFF, WAVE and SD2F (Protools). We do not recommend the use of SD2F formats because they can become unreadable while passing to an external hard drive, or over a network.
- The sound card that will play the sounds in Holo-Spat must support their sampling rate. All sound files used in the session must have the same sampling rate.
- Sound files cannot be modified in Holo-Edit; they can only be placed in time, which implies that they must be clean, well cut and mixed before being imported. They can however be modified after their first import.
- *Do not use blank spaces in the filenames, which cannot be longer than 32 characters*.

b. Sound files Import

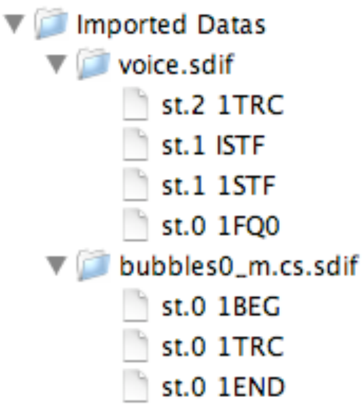
- Importing a sound file means extracting the essential data of this file to construct its waveform (amplitude curve).
- This method can take a while, so after an import, this waveform is saved as a file in the *cache* folder of the software. The next time this session/file is opened/imported, the cache file will be loaded (more quickly) instead of the original file (if a cache file is present either in the *cache*

folder of the application, or in the session's folder or in the folder containing the sound).

- If the file is modified after its first import, it will be imported again to update its waveform. If it cannot be found when the session is opened, the cache file will be used instead.
- *Advice: if you want to work on another computer, take the cache files with the session.*

c. SDIF files import

To import a SDIF file, you can either drop it in the *'imported datas'* area, or select *'import datas…'* from the contextual menu of the same area.
During the import of a SDIF file, data are segregated depending on the different streams and matrix types it contains. Thus this leads most of the time to the creation of many *'holoData'* from only one SDIF file. These last can be seen as tree nodes in the *soundPool* :



- Imported Datas
  - voice.sdif
    - st.2 1TRC
    - st.1 ISTF
    - st.1 1STF
    - st.0 1FQ0
  - bubbles0_m.cs.sdif
    - st.0 1BEG
    - st.0 1TRC
    - st.0 1END

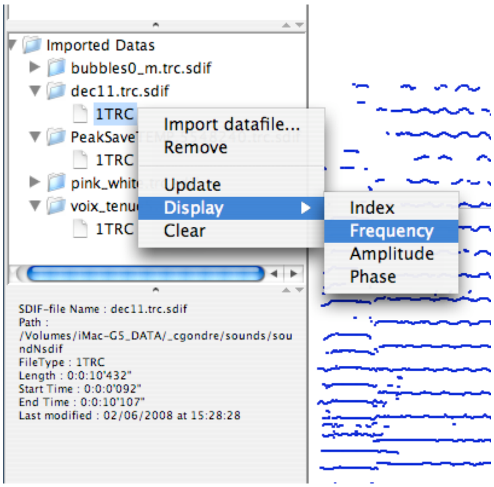
The holoDatas created from a SDIF file are shown as different tree nodes

These *HoloData* are named according to the SDIF streamIDs ('st. 2' stands as 'stream n°2' as an example) and the type of data contained by those streams (i.e. 1FQ0, 1TRC, etc…).

It is then possible to add these *holoData* directly to the *Score* (by drag-and-drop from the *Sound Pool* to the *score*), or after you 'linked' it to a sound by drag-and-drop over a sound in the *'imported sounds'* area. Linking a data to a sound ensures that they will always be 'synchronized' during their manipulation within the *Score*.

d. holoData visualization :

When a *holoData* is selected, it is displayed in the *soundPool* in two dimensions only. The X axis represents the time, while the Y axis represents one field (i.e. frequency, or amplitude…) of the SDIF data type that may be chosen from the contextual menu (right-click on a selected data -> '*Display'* ->….) :

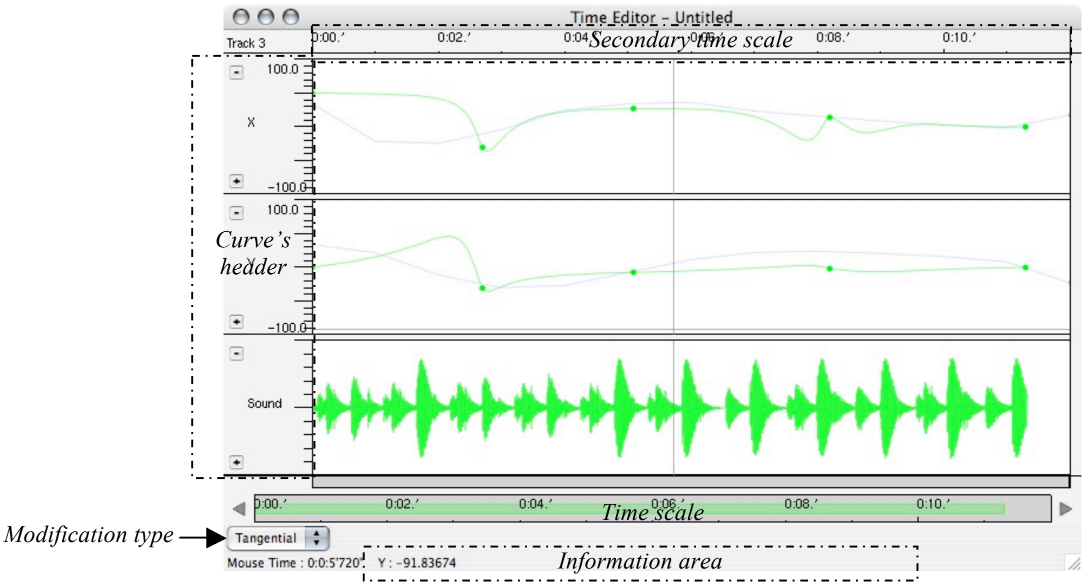


SDIF-file Name : dec11.trc.sdif
Path :
/Volumes/iMac-G5_DATA/_cgondre/sounds/soundNsdif
FileType : 1TRC
Length : 0:0:10'432"
Start Time : 0:0:0'092"
End Time : 0:0:10'107"
Last modified : 02/06/2008 at 15:28:28

The displayed field can be defined through the contextual menu

G. Time Editor

This last editor permits the precise temporal manipulation of trajectories.



Mouse Time : 0:0:5'720
Y : -91.83674

Different curves are available (X, Y, Z, Ray, Angle, Sound, data) according to time. It is possible to add or remove the visualization of a curve and thus concentrate on one or more parameters by clicking on the plus and minus buttons in the headers of the curves.

Clicking on the name of the curve makes a drop-down list appear. Choosing an element in this list modifies the parameters of the curve.

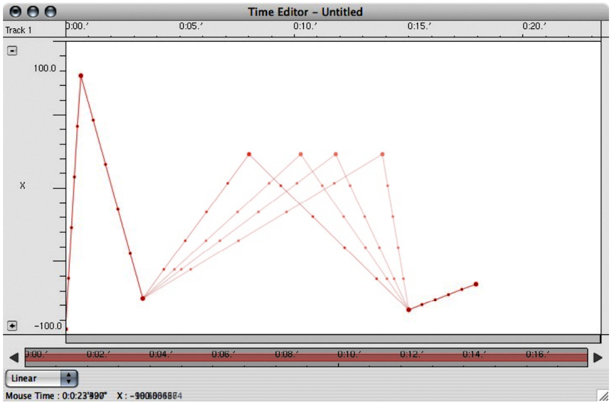
Only the active track can be changed. The other (inactive) tracks are displayed as hatched or shaded.

The following functions apply similarly to any curve, except the sound curve, which cannot be changed.

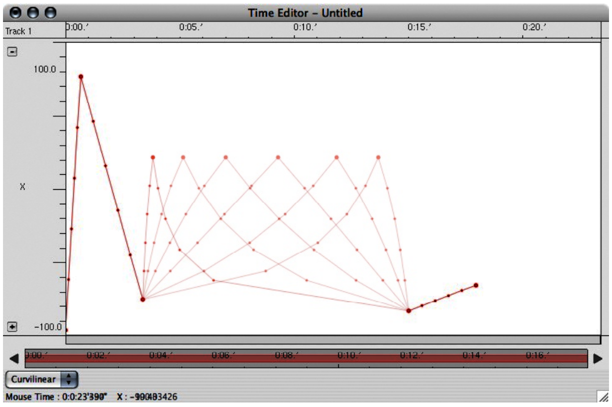
a. Temporal displacement

According to the value of the modification type (linear, curvilinear, tangential), moving a point or a selection will have different results.

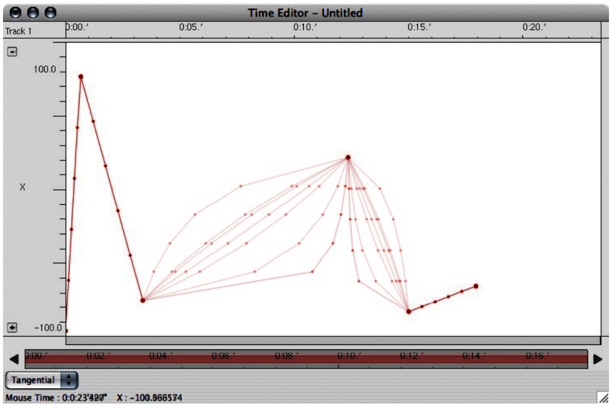
In linear mode, this point will move normally, and the points surrounding it (between the editable points preceding and following this one) will move in time in a linear way.



In curvilinear mode, this point will move normally, and points surrounding it will move so as to create an acceleration or a deceleration on either side.



In tangential mode, this point won’t move, but the points surrounding it will move so as to create an acceleration followed by a deceleration (or the opposite).



b. Spatial displacement

By default, moving a point will only modify its date. If the Shift key is held down while moving a point, the point will also move along the vertical axis.

If the Shift key is held down before moving a point, the point will only move along the vertical axis.

III. Functions

All available functions in the *Functions* menus are algorithms in the form of java files contained in the *algo* folder in the application folder. Each function saves its own presets in an external file with the same name and having the extension *.pst* in the same folder.

Generative functions allow the creation of trajectories defined by algorithms. These functions can modify one or more tracks at the same time. Their result can replace the current time selection or can be added at the end of each track.

Transformative functions use the content of the track in the time selection to modify it. They apply from one track A to another track B (A and B can be the same).

All functions are applied between *Begin* and *End*.
All generative functions are marked with an asterisk in the list below.

Symmetry
Duplicates the content of the Input track to the Output track after mirroring it about a central point or about the X or Y axes.

Rotation
Duplicates the content of the Input track to the Output track after rotating it about the central point of the graph.

Rotation Progressive
Duplicates the content of the Input track to the Output track after rotating it about the central point of the graph. The value of the angle evolves linearly between two values over the duration of the transformation.

Translation
Duplicates the content of the Input track to the Output track after moving it.

Proportion
Duplicates the content of the Input track to the Output track after reducing or enlarging it.

Proportion Progressive
Duplicates the content of the Input track to the Output track after reducing or enlarging it. The value of this reduction or enlargement evolves linearly between two values over the duration of the transformation.

Interpolation
Makes a dynamic interpolation between the content of two tracks between *Begin* and *End*. Interpolation is independent on the X, Y, and Z-axis.

Smooth
Smoothes the input track.

Resample
Modifies the temporal resolution of the starting curve by placing a constant number of one point per unit of time between the editable points.

Time Shift

Shifts in time the portion of the active track starting from *Begin*.

Time Stretch

Modifies the duration of the content of the active track between *Begin* and *End* by multiplying it by a percentage.

Time Reverse

Duplicates the content of the active track between *Begin* and *End* and reverse the data.

Acceleration

Modifies the progression in time of the content of the active track between *Begin* and *End* (without changing its total duration) so as to create an acceleration/deceleration.

*Circle**

Create a circular trajectory (if initial ray = final ray) or in spiral (if not).

*Lissajou**

Create a Lissajou figure.

*Brownian**

Create a step-by-step random trajectory in a defined area. The length of a step is expressed by the Brownian value: percentage of the defined area to cover between each steps.

*Random**

Create a random trajectory in which each point is placed in front of a randomly chosen speaker. The distance to speakers evolves linearly between two values over the duration of the generation.

Other functions place speakers in space. Like the generative algorithms, these functions can replace existing speakers or be added next to them.

*Circular Speakers**

Places the speakers in circle around the center of the graph.

*Rectangular Speakers**

Places the speakers symmetrically around the vertical axis passing through the center of the figure.

IV. Scripting

A. What is a script?

A script is a set of commands to be executed by Holo-Edit in order to perform specific actions, such as the application of many successive transformations over a trajectory.

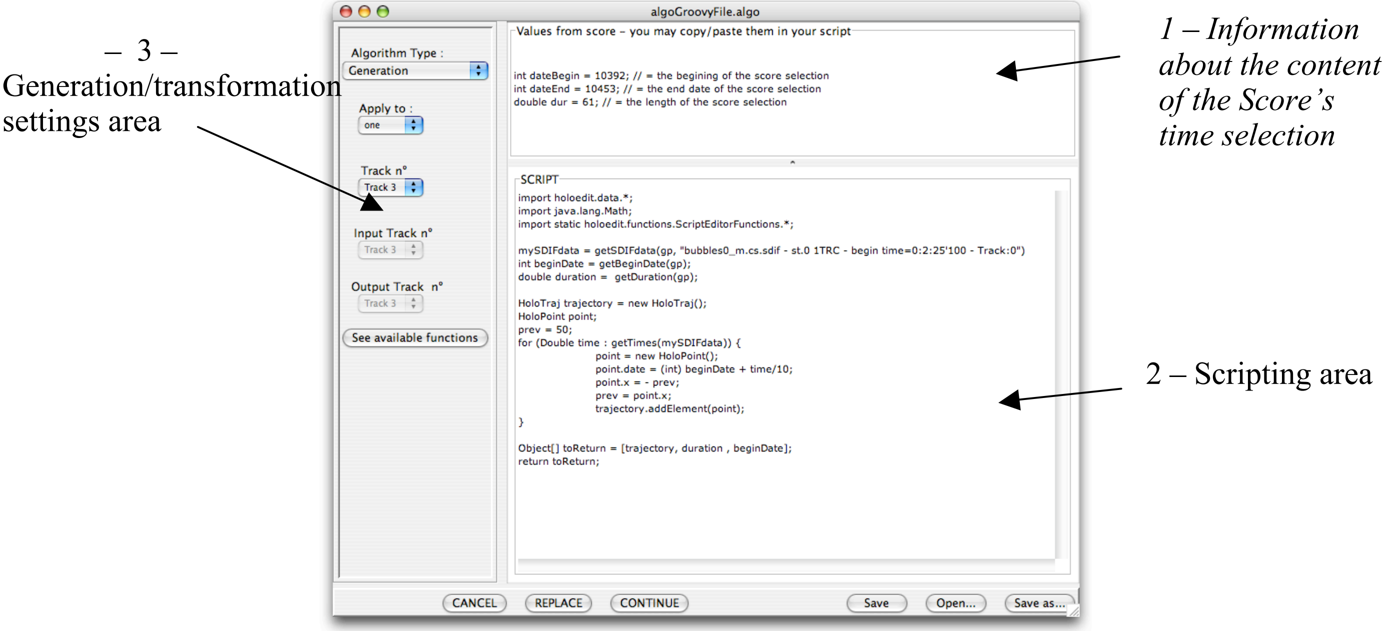
Thus, scripts may be particularly useful in different cases:

- Performing repetitive tasks efficiently.
- Applying high precision algorithms on some data.
- Getting some information about the available data.
- Using predefined mathematical functions.
- Creating trajectories transformations and generations libraries.
- Fast experimentations.

In HoloEdit, scripts are particularly useful to use the different fields/elements of some SDIF data for trajectories generation/transformation.

B. Scripting interface

An interface dedicated to scripting is available from the menu (*Functions -> Script*).



Algorithm Type : Generation

Apply to : one

Track n° Track 3

Input Track n° Track 3

Output Track n° Track 3

Values from score – you may copy/paste them in your script

```
int dateBegin = 10392; // = the begining of the score selection
int dateEnd = 10453; // = the end date of the score selection
double dur = 61; // = the length of the score selection
```

SCRIPT

```
import holoedit.data.*;
import java.lang.Math;
import static holoedit.functions.ScriptEditorFunctions.*;

mySDIFdata = getSDIFdata(gp, "bubbles0_m.cs.sdif - st.0 1TRC - begin time=0:2:25'100 - Track:0")
int beginDate = getBeginDate(gp);
double duration =  getDuration(gp);

HoloTraj trajectory = new HoloTraj();
HoloPoint point;
prev = 50;
for (Double time : getTimes(mySDIFdata)) {
                point = new HoloPoint();
                point.date = (int) beginDate + time/10;
                point.x = - prev;
                prev = point.x;
                trajectory.addElement(point);
}

Object[] toReturn = [trajectory, duration , beginDate];
return toReturn;
```

– 3 –
Generation/transformation settings area

1 – Information about the content of the Score's time selection

2 – Scripting area

Holo-Edit scripting interface

This one is divided in three parts:

1- A non-editable text area named '*Values from Score*', that provides some information from the

Score, such as :

- The *holoData* contained in the time selection of the *Score*.
- The current time selection (begin and end times ; duration).

2- An editable text area used to write the scripts.

3- An area containing different choice lists and buttons to define the type of script (generative or transformative) and the track(s) where the trajectories have to be written.
This area also contains a button named *'see available functions'* that opens the documentation of two HoloEdit java classes that contain some useful methods for scripting trajectories.

C. Writing scripts for holoEdit

Scripts are written in *Groovy*[4], a dynamic language built upon Java. This means that you can write your scripts in pure Java, and access to the whole Java API.

a. Templates

The interface has been conceived to make trajectories generations and transformations by script as easy as possible. The next two templates constitute the minimal code required for (1) a trajectory generation and (2) a trajectory transformation. The "**…**" represent some fields to be filled.

1. Generative script template :

```
// imports and declarations
...
HoloTraj trajectory = new HoloTraj();       // creates a new trajectory
int beginDate = …                           // trajectory's begin date
double duration = …                         // trajectory's duration

// defines the trajectory by adding some points
for (…) {
        HoloPoint point = new HoloPoint();          // creates a new point
        point.date = ...                            // point settings…
        point.x = ...
        point.y = ...
        point.z = ...
        trajectory.addElement(point);       // adds the point to the trajectory
}
// The trajectory, its duration and its begin date must be returned :
Object[] toReturn = [trajectory, duration, beginDate];
return toReturn;
```

[4] http://groovy.codehaus.org/

2. Transformative script template :

```
// imports and declarations
...

HoloPoint point;

// running though the trajectory contained in the current time selection
int trajectoySize = getTrajectorySize();
for (int i = 0 ;  i < trajectoySize ; i++) {
        // gets a point of the trajectory
        point = getTrajectoryPoint(i);
        // modifies point settings
        point.date = ...
        point.x = ...
        point.y = ...
        point.z = ...
}
```

More templates can be found in the directory holo-edit/scripts/templates. A description of these last can be found in the file *‘templates-doc.pdf’* of the same directory. Reading through this document is probably the best way for you to get to know the most useful HoloEdit java methods, and how to use them to script trajectories generation/transformation.

HoloEdit scripts are saved with a ‘.algo’ extension, and may be open and edited in any text editor.

V. File Input / Output

A session of Holo-Edit is saved in a file with the extension *.holo*. This file is a XML file easily readable by a human being or a computer.

This file is also readable by Holo-Spat: once the composition is finished, it is not necessary to use Holo-Edit to play it, reading the *.holo* file is enough.

It is also possible to export and import tracks separately (with the extension *.tk*). This can be done from the *Track* menu or the contextual menu (accessible by right clicking on the background of a track in the *Score* window).

Trajectories can also be exported/imported (with the extension *.tj*). This can be done from the *Track* menu or the contextual menu (accessible by right clicking on a trajectory in the *Score* window).

It is also possible to import old versions of Holo-Edit files.

Supported formats for sound files are detailed in part *II.F Sound Pool*.

VI. Communication with Holo-Spat & Transport

A. Startup

Communication with Holo-Spat is fundamental to listening to your composition as it is written.

Both applications use the *OSC*[5] network protocol and *Bonjour*[6], a network protocol developed by *Apple* on the *zeroconf*[7] norm, "a standards-based networking technology that automatically connects electronic devices on a network, allowing them to interoperate seamlessly without any user configuration".

To enable communication, you must activate the *Enable Transport* menu from *Transport* menu.

If several instances of Holo-Spat and Holo-Edit are running simultaneously on the local network, Holo-Edit will provide a list from which you must choose the appropriate one.

In the current and subsequent versions, transport commands from Holo-Edit activate the same commands in Holo-Spat and vice versa.

B. Configuration

If the local network is very large and Holophon functions unpredictably on that network, deactivate the Bonjour option to avoid conflicts with other applications once communication has been established between Holo-Edit and Holo-Spat.

If Bonjour is activated, it is possible in Holo-Spat to precisely name the service that Holo-Spat will declare to Bonjour, and thus to Holo-Edit.
If not, it is possible to manually configure the communication between the two applications by entering their IP addresses (or their domain names for WAN applications) and their UDP port numbers in the *Preferences…* menu, under the *OSC* tab.

Once you have made modifications, deactivate and reactivate the transport command or restart the application to validate the modifications.

C. Track Numbering

The number of a track in Holo-Edit (which can be changed in the *Track* menu) indicates which track in Holo-Spat will be controlled by Holo-Spat.
One can only have three tracks (numbered 5, 6 and 8) in Holo-Edit respectively controlling the same numbered ones in Holo-Spat. The remaining tracks must be controlled by other means in

[5] Open Sound Control http://cnmat.berkeley.edu/OpenSoundControl/

[6] http://www.apple.com/macosx/features/bonjour/

[7] http://www.zeroconf.org/

Holo-Spat.

D.Playing & Recording

See Holo-Spat.

VII. Options & Preferences

Maximum recent files: the maximum number of files to list in the *Open recent files* menu.

Automatic Track Selection: enables automatic track selection when the mouse passes over a point in the *Room Editor* or when an object is moved in the *Score*.

View only editable points: hide/show non-editable points.

View speakers: hide/show speakers.

Load last file on startup: automatically load last file read on startup.

Coordinates: Cartesian or Polar. Changes the type of coordinates in the point or speaker editor.

Draw points number: the number of points to draw while playing. If the computer slows down while playing, you should decrease this number. For a more aesthetic effect, increase it.

Scroll speed: the speed of the zoom when using the mouse wheel. Increase this number if it’s too slow and vice versa.

Holo-Spat

I. Introduction

Holo-Spat is an application for the realization of the work done in Holo-Edit. It can also work as a stand-alone application for the real-time spatialization of sounds (voices, instruments, synthesizers…).

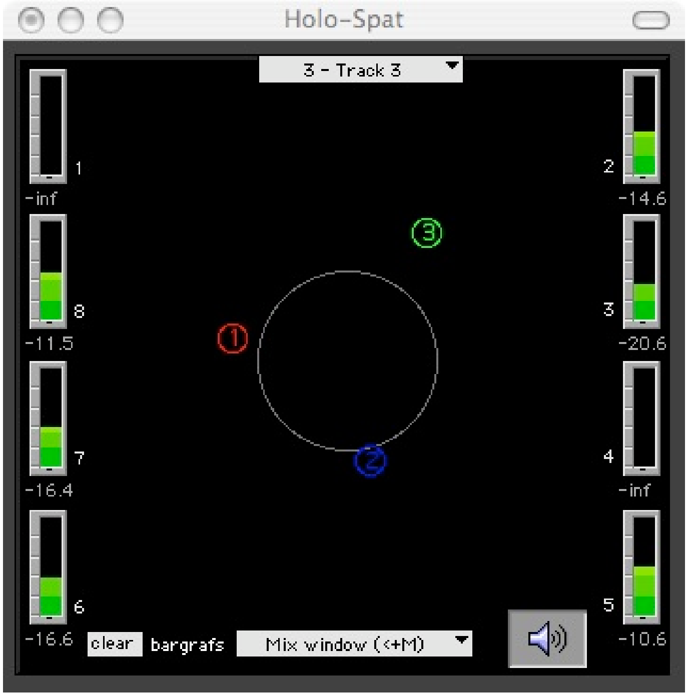
Holo-Spat functions as both a multi-channel mixer for an array of loudspeakers and a psycho-acoustic sound processor.

Each track algorithmically places a sound file or an external input in space. By default, this algorithm corresponds to an implementation of John Chowning[8]'s works with eight speakers.

Other algorithms are available (in the *algo* folder of the application) and allow work with different speaker configurations (4,16,cube…).

To use or create other algorithms, or more complex speaker configurations, please see the *Advanced functions* section.

II. User interface



A. Room

This window is at the same time a top view of the speaker configuration and an area for controlling sound positions in real time.

It allows one to visualize the position of each sound played at a given time as well as the speaker's volume as represented by the peak meters.

The bottom menu of this window gives access to the other windows of Holo-Spat.

The top menu allows one to choose the track controlled by the mouse in the control area.

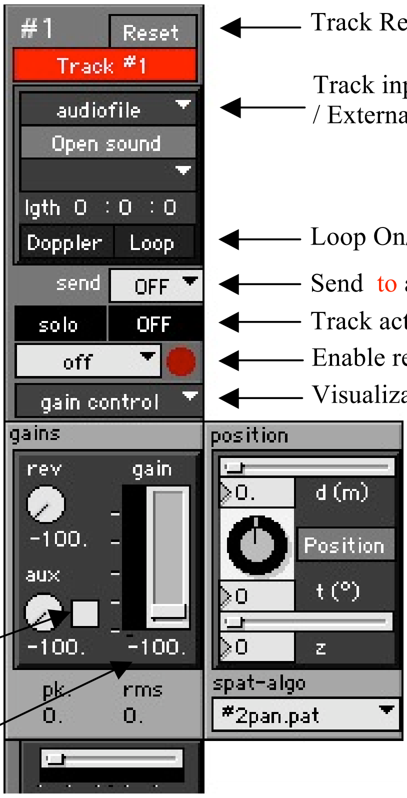
The button with a speaker is called the DAC, which must be engaged to run sound processing.

[8] John Chowning, « The simulation of Moving Sound Sources », in *Computer Music Journal*, June, 1977, pp. 48-52, reprinted from the *Journal of the Audio Engineering Society*, 19 :2-6 , 1971

B. Mix Window

This window displays all tracks available in the form of a mixing desk track. There is also a Master section.

Description of a track:



Track number

Name

Add a sound to the sound list

List of available sounds

Length of the selected sound

Doppler On/Off

Solo On/Off

Control method of this track by Holo-Edit or *.holo* files

Reverb send gain (dB)

Aux send gain (dB)

Auxiliary post/prefader (default post)

Track gain before spatialization (dB) + peak meter

Track Reset

Track input menu Sound file / External input

Loop On/Off

Send to a bus

Track activity On/Off

Enable recording (of trajectory)

Visualization (gain/position controls)

Distance to center (m)

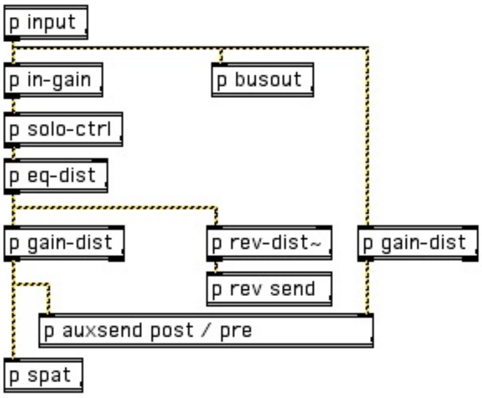
Position

Angle in degrees

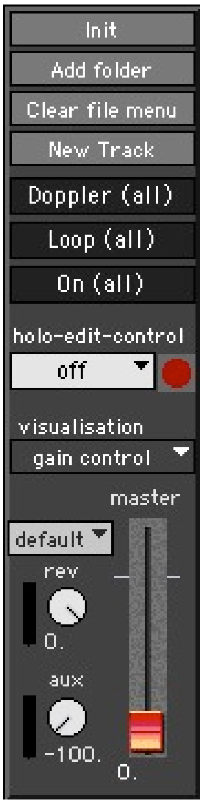
Elevation (0-100)

Algorithm (list of all patches available in the *algo* folder of the application)

DSP diagram of a track:



Description of the master Section:



Global initialization

Add a folder to the sound file list

Add a track to the session

Loop On/Off for all tracks

Holo-Edit Control method for all tracks

Track visualization

Reverb Type (default or VST plugin)

Reverb Gain (dB) + peak meter

Aux Gain (dB) + peak meter

Clear the sound file list

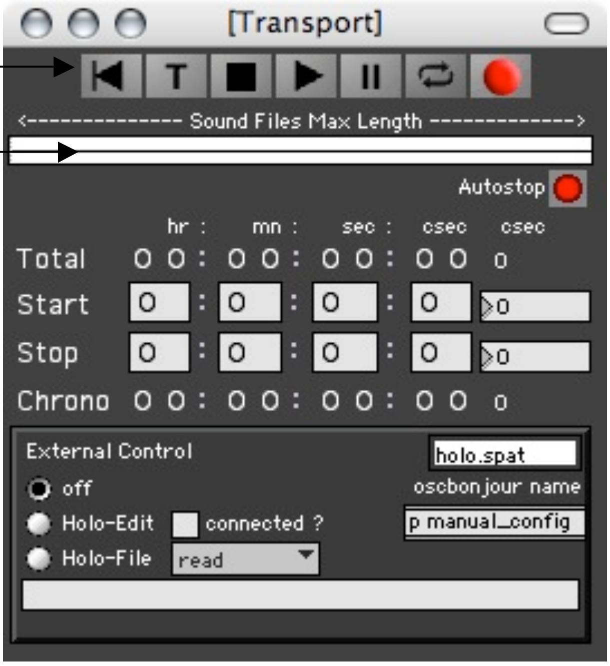
Doppler On/Off for all tracks

On/Off for all tracks

Enable recording (of trajectories) on all tracks

Master gain (dB)

C. Transport



Transport (rewind, preload, stop, play, pause, loop, record)

Time scale representing the total duration of the session and the time selection's minimum and maximum limits

Total duration of the session

Beginning of the time selection

End of the time selection

Play position

Choice of external control, read/write commands of *.holo* files

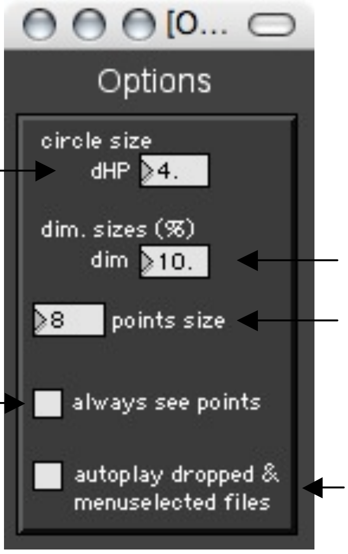
Name of the last read *.holo* file

Autostop determines if playing will stop at the end of the time selection.

Name of the *Bonjour* service.

Manual configuration of network communications.

D. Options



Options

circle size

dHP 4.

dim. sizes (%)

dim 10.

8 points size

□ always see points

□ autoplay dropped & menuselected files

Size of speaker circle : adapt to the natural reverberation of the room

Scaling factor of Holo-Edit trajectories

Size of drawn points

By default, points are visible only when sound comes out of its track

Automatically play a chosen sound file in the sound file list

III. Use

Several modes are possible with Holo-Spat.

A. With Holo-Edit

Ensure that the External Control portion of the *Transport* window of Holo-Edit is connected and selected (this should be automatically done when Holo-Edit is launched and its transport command activated).

On each track controlled by Holo-Edit, select a control method:

- *Position*: changes the sound file while playing or spatializes an external input (voice, instrument…).
- *Sound*: controls the position in real time by another means (algorithm, external controller…).
- *Sound + Position*: controls both sound and position parameters of the track.

If the sound part of the track is not controlled by Holo-Edit (method set on *position*), you must:

- Either select a sound file (command *Open file* of the track, or choose a file in the file list, which can be filled with the command *Add folder* of the *Master* section),
- Or select *input patch* in the input menu and configure the input patch.

B. With .holo file

Using a *.holo* file is similar to using a Holo-Spat file with Holo-Edit. In the External Control portion of the *Transport* window, check Holo-File, and choose the file to *read* with the read command.

Then select a control method for each track (see preceding paragraph).

C. Other methods...

Select a sound file (command *Open file* for the track, or choose a file in the file list, populated by the command *Add folder* under the *Master* section) or select *input patch* in the input menu and configure the input patch.

The position of the sound can then be controlled by:

- The mouse in the main window,
- An algorithm in real time (see *Advanced Functions I*),
- By an external midi controller (see E. Midi Control).

D. Playing & Recording trajectories

Transport commands are common to both Holo-Edit and Holo-Spat. Once they are connected, both applications react to the same commands (setting start and stop position, play, stop…).

Don't forget to start the DAC (the button with a speaker) before playing.

Just as for the reading and recording of trajectories, it is necessary to specify (in Holo-Spat) which track will be recorded the next time the play command is launched.
In the External Control portion of the *Transport* window, select the record destination (Holo-Edit or Holo-File).
For each track, select a control method (see paragraph A) and check the *prepare recording* box.
In the transport window, prepare the recording and execute the play command.

In recording mode, each change in the position of a track will be recorded regardless of the method used (user interface, algorithm, external controller…).

Tracks whose record mode is deactivated will be read in the file or in Holo-Edit if they contain information.

If the recording destination is a Holo-File, save your recording with the *write* command. It is then possible to open and rework this session in Holo-Edit.

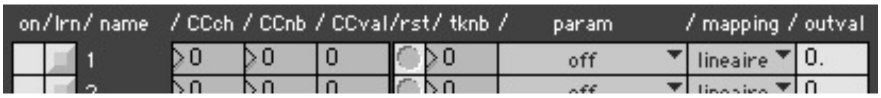
E. Choosing the Spatialization Algorithm

All patches available in the *algo* folder of the application are spatialization algorithms. By default, the algorithm p08pan.pat is used.
(See *Advanced Functions IV*)

F. Midi Control

The *Midi Control* window allows the control of some parameters with the midi protocol (external application, physical controller…).

Several controllers are defined and then assigned to a parameter of a track.



on	lrn	name	CCch	CCnb	CCval	rst	tknb	param	mapping	outval
□	□	1	0	0	0	○	0	off	lineaire	0.
□	□	2	0	0	0	○	0	off	lineaire	0.

on: activate the controller
lrn: midi learn button to automatically detect the next moved button
name: name
CCch: midi channel
CCnb: number
CCval: last received value
rst: resets the assignment of the controller
tknb: track number to control (0 = Master Section)
param: parameter to control
mapping: mapping type (linear, exponential, inverted, extra (user-defined mapping), boolean (on/off))
outval: effective value sent to the parameter.

Different parameters are available in the *param* list:

- *on*: activate/deactivate the track
- *vol*: track gain
- *rev*: reverb send gain
- *aux*: auxiliary send gain
- *auxpre*: auxiliary output pre/post fader
- *doppler*: doppler on/off
- *loop*: loop on/off
- *d*: distance from center
- *t*: angle
- *z*: elevation
- *p1*, *p2*, *p3*: additional parameters.

G.Input/Output Patches

The input patch is useful when one wishes to spatialize voices or instruments in real-time. It connects the physical inputs of the sound card to Holo-Spat's tracks.

It is also possible, for example, to connect a Max variable to the input of a track to spatialize a synthesis patch. For that purpose, 8 variables have been provided (1extpatch – 8extpatch). (*NB: these variables use traditional send/receive and not send~/receive~).*

The output patch connects the virtual speakers and the auxiliary channel to the physical outputs of the sound card.

It is also possible to record in a 8-track sound file the sound result with the *Recorder* window. It is necessary to connect speakers to the recorder in the output patch to make this possible.

The Output window allows you to visualize the gain of each of the 16 available speakers (by default) in Holo-Spat (only 8 have level meters in the main window). This window also permits the fine adjustment of your speakers (gain, delays…).

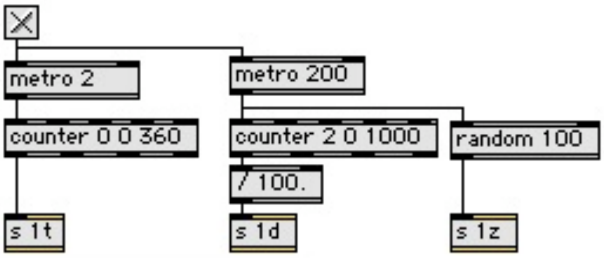
Advanced functions[9]

I. Realization of a real time control algorithm

For each track, the Max/MSP variables controlling the positions follow (where #1 is the number of the track):

- #1d: distance from center
- #1t: angle in degrees
- #1z: elevation

Example: algorithm for the creation of a spiral with a random elevation.



These variables cannot only be used for creating a control algorithm (like the example above), but also for connecting a control interface to a track (graphics tablet, joystick…).

II. Obtaining the source code of Holo-Edit

You can obtain the **latest version** of the source code for Holo-Edit with Subversion[10]. Enter the following command:

svn co svn://gmem.homeip.net

The source code of Holo-Spat is provided in this distribution.

[9] These functions require skills in Java concerning Holo-Edit, and in Max/MSP concerning Holo-Spat.

[10] http://subversion.tigris.org/

III. Development of algorithms for Holo-Edit

(Skills in java needed)

All generative or transformative algorithms in Holo-Edit inherit the holoedit.functions.Algorithm class.
To compile your class, you need the package holoedit.jar in your classpath (in the application bundle (Mac) or in the *resource* folder (Windows).

Add the compiled *.class* file in the *algo* folder of the application and restart Holo-Edit.

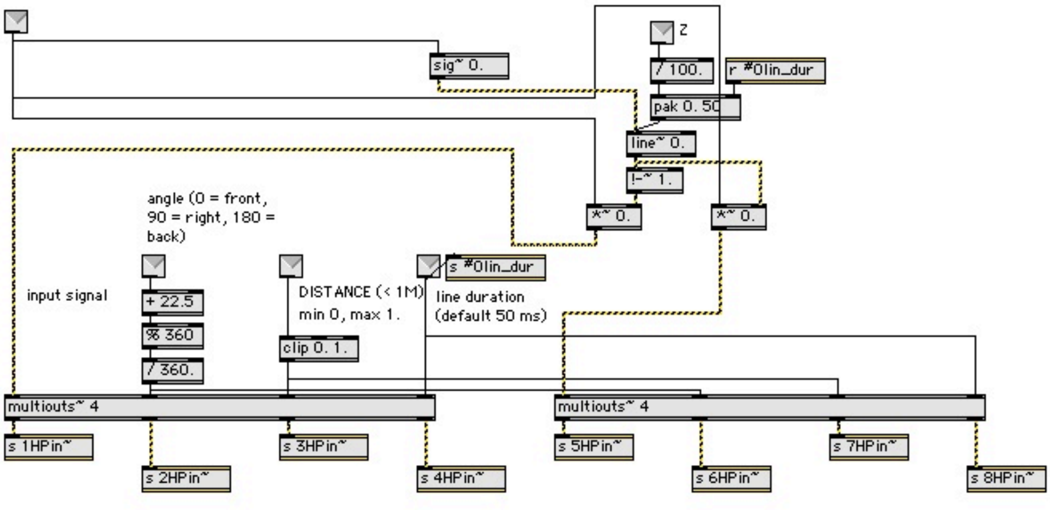
All necessary information to develop your own algorithm can be found in the **Example.java** file in the Documentation folder of the application.

IV. Development of spatialization algorithms for Holo-Spat

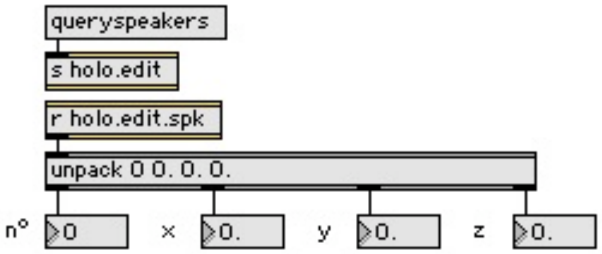
To develop your own spatialization algorithm, you need to create a patch having these five inputs:

- Input signal
- Angle in degrees (between 0 and 360)
- Distance between 0 and 1 (= real distance of the point / dHP)
- Interpolation time in ms
- Elevation between 0 and 100

In addition, you can get the number of the track calling your algorithm with the #1 variable.
By default 16 speakers numbered from 1 to 16 are accessible with #1HPin~ (where #1 is the number of the speaker) in Holo-Spat. (*NB: these variables use traditional send/receive and not send~/receive~).*
Place your patch in the *algo* folder of the application and restart Holo-Spat.
The example below illustrates the implementation of an algorithm for cubic speaker configuration:



You can also retrieve the position of the speakers in Holo-Edit by sending the command « queryspeakers » to the variable *holo.edit*. Speakers will be sent to the variable *holo.edit.spk* one after the other as a list (number, x, y, z) as in the example below.



```
queryspeakers
s holo.edit
r holo.edit.spk
unpack 0 0. 0. 0.
0    0.    0.    0.
```