



HAL
open science

A graphical interface for trajectory design and musical purposes

Jean-Baptiste Thiébaud

► **To cite this version:**

Jean-Baptiste Thiébaud. A graphical interface for trajectory design and musical purposes. Journées d'Informatique Musicale, Association Française d'Informatique Musicale; Centre de recherche en Informatique et Création Musicale, Jun 2005, Saint-Denis, France. hal-03121399

HAL Id: hal-03121399

<https://univ-paris8.hal.science/hal-03121399>

Submitted on 26 Jan 2021

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.

A GRAPHICAL INTERFACE FOR TRAJECTORY DESIGN AND MUSICAL PURPOSES

Jean-Baptiste Thiebaut

CICM - Université Paris VIII

MSH Paris Nord

jbthiebaut@mshparisnord.org

ABSTRACT

This paper presents a generic tool useful to design trajectories in the Max /MSP environment and applicable for musical purposes. The eight graphical tools of the *trajectory* program are exposed. Then, the creative potential of the use of trajectories is presented with examples. Conceptually based on the wave terrain synthesis, we present a usage of trajectory that allows to control two parameters through time. Based on previous works [7], the trajectory can also retrieve parameters according to a map on which the trajectory is drawn. Examples of synthesis and transformation according to this process are presented. Another interface based on the waveform segment synthesis is exposed: considering the coordinates as sample values, trajectory is used to fill two waveform segments in real-time by the successive values send by the tool. Trajectory can be used too as an interpolator, considering the design surface as a two-dimensional controller. Examples of use of trajectory as an external controller for vst plugins and for adaptive effects are presented. At least, I present future developments that use the new features of Max/MSP, javascript and user interface. The future *trajectory* patch will be able to manage up to 20 objects in a bigger interface, and will take under account the elevation axis.

1. INTRODUCTION

The first goal of the *trajectory* program has been to draw geometrical trajectories to spatialise different sound sources [4,5,6]. In the software Egosound, the *trajectory* program retrieves a position (abscissa and ordinate) that is used by an ambisonic algorithm [1] or a VBAP algorithm [2] to spatialise sound in a multichannel environment (from 2 to 16 loudspeakers). The main difference with the other spatialisation software (the sony program *music space*, the Spat from Ircam or Holophon from GEM) is that this spatialisation is an intensity panning controller and does not transform the sound with a reverberation, in order to focus on the spatial position of the sounds, not on a simulation of a virtual room. This strategy leads us to the project "La terre ne se meurt pas", an interactive composition by Anne Sedes, Benoit Courribet and the author that allows the user to virtually move himself in a multichannel environment in which sounds are moving according to specified trajectories. As it's a composition and not a virtual room, it was necessary to spatialise the sounds only with an intensity panning process.

Using the *trajectory* program in the Max/MSP environment leads us to use it as a generic interface of

parameters control. As data are shared in this environment in a very simple way, it allow us to use the coordinates retrieved by *trajectory* as a pair of abstract values usable for any control purpose. The interface become then a generic two-dimensional controller like the GRM Tools controller.

This paper presents the graphical properties of the trajectory tool, and some examples of its manipulation in sound and musical processing.

2. THE TRAJECTORY TOOL

The first version of *trajectory* has been developed in 2003. A recent release is available on the CICM website (<http://cicm.mshparisnord.org/>). Trajectory is an abstraction build in the Max/MSP environment.

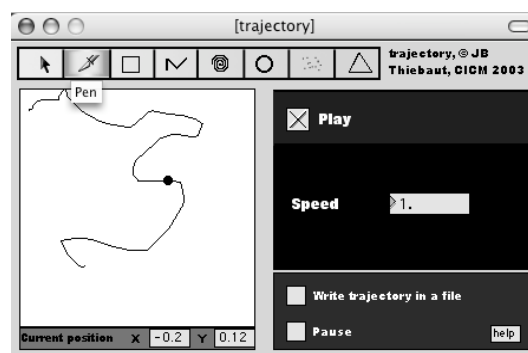


Figure 1. Interface of trajectory. In this example, the pen tool draws the trajectory on the white plan. The trajectory is then played at the same speed it has been drawn.

It contains eight graphical tools to draw trajectories. All of them are usable in real-time. While the user draws, a pair of coordinates (x,y) is output each 40ms or at a lower rate depending on the user interaction. Even if it has mainly been developed for a drawing usage, it can be processed with automation and different kind of control within Max/MSP. Messages connected to the *trajectory* allow the user to control the current trajectory. For example, the message "rectangle -0.5 0.5 0.5 -0.5 0. 1" means draw a rectangle where the upper left corner is at (-0.5, 0.5) coordinates and the lower right corner is at (0.5,-0.5). The rotation angle is 0. and the speed is 1. The message "rectangle start" plays the trajectory.

The eight tools are:

- **Drag and throw:** The movement follows the mouse. The small circle which represents the position can be thrown by the mouse; it then rebounds against the walls losing energy and may be slowed by a friction parameter.
- **Pen:** Writes a trajectory and play it at the same speed it has been drawn. The global speed can be change in real time.

- **Rectangle**: Defines a rectangle trajectory by the drawing of a rectangle
- **Ellipse**: Defines an elliptic trajectory contained in a rectangle
- **Spiral**: Defines a spiral trajectory whose biggest loop is the biggest circle contained by the editing rectangle.
- **Random**: Defines a rectangular zone in which the movement is randomly played and the speed can be controlled.
- **Broken Line**: This interface allows the user to draw segments and to affect them a duration. It's also possible to edit, save or load a trajectory.
- **Triangle**: Defines a triangle trajectory

Musicians and composers use *trajectory* in the emerging area of the surrounding sound. As multi-channel environments are more used than ever, it exists a demand on a useful tool to compose sound in space and time. It has already been used by the composer Anne Sedes and the choreograph Laurence Marthouret in the performance "Espaces sensibles". It is also used in the opera composed by Jose Manuel Lopez Lopez, "La noche y la palabra". Students of University Paris VIII use it for live performance and mixed composition.

The interface of trajectory is philosophically close to the GRM Tools interface. In fact, we can use trajectory as a controller of sound transformation. The parameters (x,y) are the controllers, as in the GRM approach. The following part deals with different way to use trajectory as a transformational interface.

3. TRAJECTORIES AS A SOUND PROCESSING INTERFACE

We present here some possible uses of the trajectory interface to generate sound, and hopefully, music.

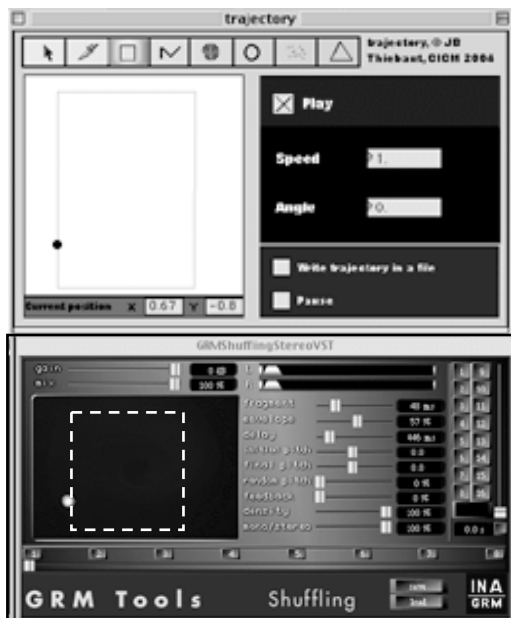


Figure 2. Using trajectory as a controller of the GRM Tools interface.

As a first example, it's possible to use trajectory as a controller of VST plugins within Max/MSP. As each parameters can receive a value between 0 and 1, we just have to map the (x,y) value outputted by trajectory to use them as the controllers of two parameters. In the case

we use a GRM Tools VST, like Shuffling Stereo, the trajectory interface can control the manipulation interface of Shuffling, that's to say control the delay and fragment parameters. The value of the parameters will then respond to a specific trajectory.

We consider *trajectory* as a generic tool for controlling parameters. The most important thing to use these parameters is the interface of mapping towards an interactive and creative interface for sound design and sound transformation.

3.1. Control a large set of parameters

We have developed with *Egosound* a program that allows the user to control the spatialisation of multiple sound sources. With the same interface, it is possible to control a higher set of parameters than the intensity of the sounds. For example, the position of the user can correspond with a data vector of transformation parameters. Recent works at CICM have shown different ways to use a n-dimensional matrix as an array of parameters. A trajectory within this matrix (viewable as an image) can then control the sound. For an image, the data vector is constituted of four planes, alpha, red, green and blue. We focus only on the visible parameter, i.e. red, green and blue planes. The *trajectory* object offers different strategies for the control of these parameters. To a position sent by *trajectory* corresponds a unique vector of data. A trajectory within a matrix causes a succession of data that could be map to synthesis or audio effect. A regular and continuous trajectory such as an ellipse or a rectangle may causes transformation repetitions. Acceleration and deceleration allow to control the period of these repetitions. A random movement constrained to a significant part of the image may cause random variations in a small range of values.

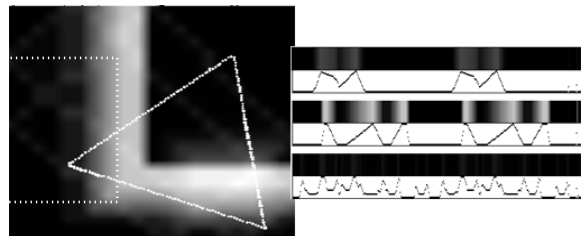


Figure 3. The triangle trajectory played on the left causes three parameters lists. The values that correspond to each point are represented with their color intensity and, below, by a graph. The trajectory started on the upper corner, clockwise.

3.2.Using trajectory as a waveform synthesis interface

This example refers conceptually to the waveform synthesis [3]. This synthesis is based on a waveform segment of a variable size (between 2 and 1000 ms) that will be read repeatedly. The waveform is written according to the current position of a trajectory drawn by the user. As a point is constituted of two values (x,y), we can write two waveform segments.

3.2.1. Technical precisions

The program writes for each value send by the trajectory tool a sample of the waveform. The trajectory tool sends a couple of values each 40 ms. In order to design a 100 ms sound at a 44.1 kHz rate, we need exactly 4410 values, obtained in 176,4 seconds. We can elaborate different strategies to reduce the writing time of the segments. First we can reduce the buffer size. A 2ms buffer is written in 3,528 seconds. The second strategy consists in interpolating the values send by trajectory. According to the level of interpolation, the buffer will be written more or less quickly. For example, with a ten points interpolation, the buffer will be written ten times quicker.



Figure 4. The coordinates of the rectangle trajectory on the top write themselves in the two buffers on the bottom. The upper buffer corresponds to the abscissa variation, the lower buffer to the ordinate variation.

3.2.2. Aesthetic results

The short size of the buffer and its repetition places the aesthetic result in the fields of loop music. The large possibilities of trajectories allow to create different kind of sounds with a progressive approach that fits to real-time music and live electronic. With a small buffer size, the repetition is so frequently that the sound will be heard as a textured sound flow. In this case, the manipulation of the trajectory causes an important variation of sound in real-time. For a buffer size higher than 100 ms, we should obtain a rhythmic loop. The influence of the trajectory on the sound is rather slow. This tool can be used in real-time to create progressive sounds texture or rhythm and as a sound design tool.

3.3. Others possible musical applications

We also can consider the trajectory surface as a graphical controller. At each corner is attached an image. A visualization window represents a mixed image, based on the four basic images. The current position determines the level of mixing between these images. The more the position is close to a corner, the more the image attached to this corner appears on the visualization window. The images could be Short Time Fourier Transform, and the result is then synthesized to output a hybrid sound. They could also represent matrixes of parameters. This interface is a graphical four-sources interpoler.

Inspired by adaptive audio effects [8], it is also possible to control a trajectory according to sound parameters. This sound may have been produced by parameters that come from a trajectory. The resulting sound will then modify the original parameters. Such a process may typically be used in an interactive environment.

4. PERSPECTIVES AND FUTURE DEVELOPMENTS

The current and future developments focus on the feedback given by the users of the program. The main developments concern the management of several objects in the same interface and the possibility of control the objects on the elevation axis.

4.1. Multiple objects in trajectory

Control several moving objects in the same interface is possible due to the new features recently add in the Max/MSP environment. The number of objects can be changed dynamically or defined from the beginning. When the number of objects changes, the patch dynamically changes its number of *outlet* (a max gate that outputs a pair of coordinates).



Figure 5. The outlets dynamically fit to the specified size.

Each object has a gradient colour from white to black and his coordinates are output on the corresponding outlet. An object can be selected with a click while the control button (ctrl) is pressed. Then, the selected object became red and all the geometrical manipulation will have an effect on it. The design palette persists but will be enhanced, in order to define for each trajectory a beginning and an end. The synchronisation between all the objects will be possible with a simple message sent to the patch that triggers the beginning of the specified trajectories. The scene will be bigger than in the current version of *trajectory*, in order to draw trajectories behind the loudspeakers. The square surrounded by the loudspeakers is materialised (see figure 6).

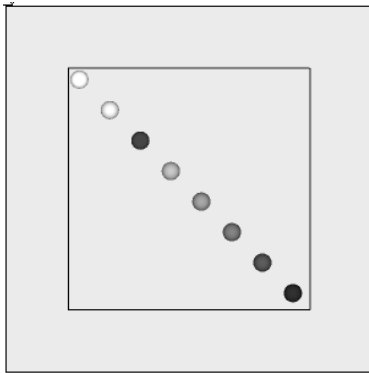


Figure 6. The scene containing all the objects. The third object is selected for design. It is possible to draw outside of the little square.

4.2. Managing the elevation axis

The third dimension raises several problems of representation. As the computer screen is two-dimensional, we have to find tricks to represent the third axis. I thought about two strategies to do that. The first strategy consists of defining three-dimensional shapes with OpenGL on the surface of which the user designs its trajectory. For example, we design a sphere on which one can draw a rectangle. The resulting trajectory follows the surface of the sphere and is three-dimensional. Such a solution would be quite long to explore and not that much intuitive for the user. The second solution consists of choosing two planes on which the trajectory will be drawn. It could be either (x,y) , (x,z) or (y,z) . Once the trajectory is defined, the specific palette of the object allows to draw a curve that represents the evolution of the third axis. Considering that a trajectory has a beginning, an end and a known duration, the third axis's curve will be mapped to fit the duration of the trajectory. The user will have a feedback of the three-dimensional movement, considering that he is upon the scene and look at it from an elevation position (along the z axis). The size of the objects will change according to their position along the elevation axis.

5. CONCLUSION

Firstly designed to draw trajectories in a space environment, trajectory can also be used to generate sound, and hopefully, music. The future features of the *trajectory* program will allow to control up to 20 triplets of coordinates. As the surrounding sound systems evolve regularly, the third dimension could be useful to design cubic trajectories in order to enhance immersion in a sound environment. It already exists a three-dimensional object developed by Rémi Mignot thanks to ambisonic algorithm [1].

Depending on the strategy, trajectory can be a parameter generator for a synthesis approach or for a transformational approach. This leads us to think about the importance of the interface in electronic music and

of the role of the mapping of abstract data toward aesthetic intentions.

6. REFERENCES

- [1] Gerzon, M. « Ambisonics in Multichannel Broadcasting and Video » in Journal of the Audio Engineering Society 33 :11, New York, AES, 1985.
- [2] Pulkki, V.: "Virtual Sound Source Positioning Using Vector Base Amplitude Panning", Journal of the Audio Engineering Society, vol 45, no 6 1997 June
- [3] Roads, C. Computer Music Tutorial, MIT Press, Cambridge, Massachusetts, 1996.
- [4] Sedes, A., Verfaillie, V., Courribet, B. & Thiebaut J.-B., "Visualisation de l'espace sonore, vers la notion de transduction" in Espaces sonores, actes de recherches, sous la direction d'Anne Sedes, éditions musicales transatlantiques, pp. 125-143, Paris, 2003.
- [5] Sedes, A., Courribet, B. & Thiebaut J.-B.,: "Visualisation du sonore avec le logiciel egosound : work in progress" in Actes des journées d'informatique musicale 2003, Montbéliard.
- [6] Sedes, A; Courribet B. et Thiebaut JB. "From the visualization of sound to real-time sonification : different prototypes in the Max/MSP/jitter environment." in Proceedings of the International Computer Music Conference (ICMC 2004), Miami, Florida, USA, November 1-6, 2004.
- [7] Sedes, A., Courribet, B. & Thiebaut J.-B.,: "Egosound, an egocentric interactive and realtime approach of sound space", in Proceedings of the 6th international conference of DigitalAudioEffects DAFX-03, London, 2003.
- [8] Verfaillie V., "Effets Audionumériques Adaptatifs: Théorie, Mise en Oeuvre et Applications en Création Musicale Numérique", PhD dissertation, Université Aix-Marseille II, 2003