spatium · tools for sound spatialization

Manual · version 1.0

Index

Index 2

1. Introduction 3

2. Spatialization Techniques 4

2.1 HOA-DE · High Order Ambisonics with Distance Encoding 4

2.2. APbSP · Amplitude Panning between Stereo Pairs 4

3. Installation 6

3.1 spatium · renderers 6

3.2 spatium · interfaces 6

3.3 spatium · plugins 6

3.4 spatium · max 6

4. FAQ and troubleshooting 7

5. Spatialization renderers reference 9

5.1 spatium · ambi 9

5.2 spatium · player 14

5.3 spatium · diffusion 15

5.4 spatium · panning 16

6. Spatialization interfaces reference 19

6.1. spatium · flocking 19

6.2. spatium · gravity balls 2D 21

6.3. spatium · pendulum 23

6.4. spatium · pendulum 2D 26

6.5. spatium · springs 2D 28

6.6. spatium · polygonal chain 30

6.7. spatium · rotation 32

6.8. spatium · control OSC 34

6.9. spatium · linear rotation 36

6.10. spatium · trackpad 38

7. Plugins reference 40

7.1. Audio Unit plugin 40

7.2. Max for Live devices 41

7.2. spatium · live 41

7.2. OSCsend 42

1. Introduction

spatium is a set of free, open source and modular software tools for sound spatialization. It is comprised of 4 different types of software:

- spatialization renderers: standalone applications that render spatialized audio using ambisonics or amplitude panning;
- spatialization interfaces: standalone interfaces that generate spatial information to control the spatialization renderers via OSC;
- plugins: audio unit plugin and max for live devices to control the spatialization renderers via OSC;
- max objects: a library of objects for spatialization using ambisonics or amplitude panning in Cycling'74 Max.

If you want to know more about *spatium*, you can check this article I presented at SMC 2013 or, in much bigger detail, my PhD thesis (in Portuguese). If you want to quote *spatium*, please use the aforementioned SMC article:

Penha, R., & Oliveira, J.P. (2013). Spatium, tools for sound spatialization. *Proceedings of the Sound and Music Computing Conference 2013, SMC 2013*, Stockholm, Sweden.

2. Spatialization Techniques

spatium renderers and max objects use two different spatialization techniques: one based on Ambisonics (HOA-DE) and another based on amplitude panning (APbSP).

2.1 HOA-DE · High Order Ambisonics with Distance Encoding

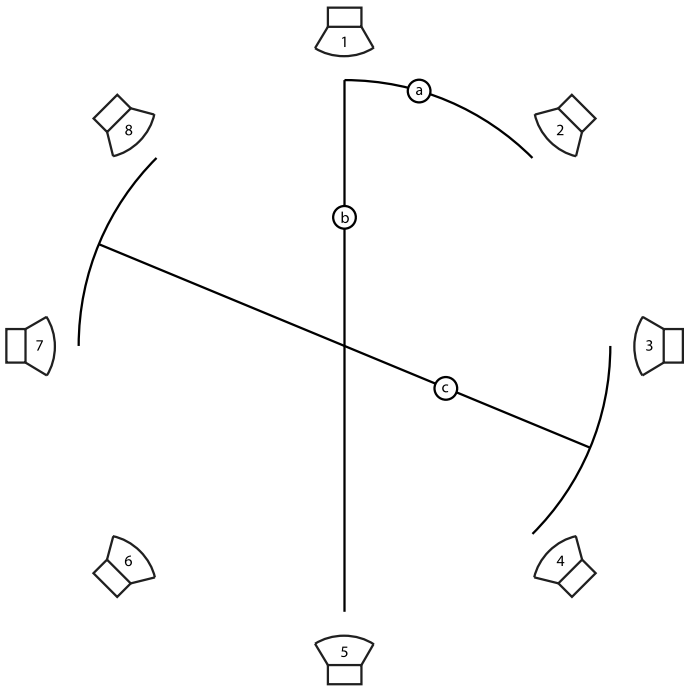
The coordinate system used is a spherical one: azimuth varies from -180° to 180°, with 0° in front and positive to the right; elevation varies from -90° to 90°, with positive above; radius varies from 0.0 to 1.0, with 1.0 on the surface of the sphere. The Ambisonics format of the *spatium · encode*, *spatium · rotate*, and *spatium · decode* max objects — as well as of the *spatium · ambi*, *spatium · player* and *spatium · diffusion* renderers — is fixed as a mixed-order soundfield, using a 3rd order horizontal, 1st order vertical approach. It uses, however, a 12-channel 3H1V instead of the traditional 8-channel 3H1P, in order to retain horizontal resolution even when going up in the soundfield. These 12 channels are, using the traditional FuMa notation, W (0th order), Y Z X (1st order), V T S U (2nd order) and Q O N P (3rd order). If you're not familiar with High Order Ambisonics, I recommend this article by Florian Hollerweger. Each loudspeaker layout choice automatically adjusts which Ambisonic orders to use and with what weights, defaulting to a *max-rE* decoder. These weights can also be manually adjusted.

To this 12-channel 3H1V mixed-order Ambisonics soundfield, an additional channel is added for distance encoding, as I proposed in this article. This approach is related to W-panning, as they both encode sounds inside the speaker array by increasing the omnidirectional component while decreasing the directional components of the Ambisonics soundfield. *spatium*'s proposal is distinguished by isolating the sounds at the center of the space in a specific channel, through the use of an additional angular coordinate. This system allows for the postponing of the application of cues for the perception of distance to the decoding stage, where they can be adapted to the characteristics of a specific space and sound system. The final sound file format of *spatium* is thus constituted by 13 channels, with the first 12 being completely compatible with regular Ambisonics.

Most Ambisonics-based objects for Max require the user to have considerable previous knowledge about the idiosyncrasies of Ambisonics (spherical harmonic orders, order weighting, etc.). This knowledge is of course helpful to get good results, as it is true with any technique, but their complexity can be overwhelming. *spatium*'s objects were developed as a simpler, though obviously less flexible, approach. As the popular VBAP object, they do not process audio, relying instead on the standard matrix~ Max object to render the audio. This also makes the objects' code easier to maintain and their output simpler to hack for specific needs.

2.2. APbSP · Amplitude Panning between Stereo Pairs

The *spatium · panning* object — as well as the homonymous renderer — uses amplitude panning with the least possible number of channels, ideally only a stereo pair, to place a sound within a 2D circle of loudspeakers. When placing a sound close to the circumference defined by the loudspeakers, it will pan the sound between the two closest speakers, as the VBAP object, using either sine or square root panning laws. When placing the sound inside the circle, it will use the azimuth to determine a stereo pair, the opposite azimuth (azimuth - 180°) to determine a second stereo pair and the radius to amplitude pan between them.



The figure above shows three scenarios: to pan sound a, the algorithm will do an amplitude panning between speakers 1 and 2; to pan sound b, the algorithm will do an amplitude panning between speakers 1 and 5; to pan sound c, the algorithm will do an amplitude panning between speakers 3 and 4, an amplitude panning between speakers 7 and 8 and finally an amplitude panning between these two pairs. This enables, e.g., the rotation of a sound placed in the middle of the circle, by keeping its radius 0.0 whilst changing its azimuth.

3. Installation

3.1 *spatium · renderers*

These are standalone applications and should work without any issue on a recent version of MacOs X. Please check the system requirements for Max 6.

3.2 *spatium · interfaces*

These are java applications and need a recent version of java to run.

3.3 *spatium · plugins*

The spatium · track 32-bit Audio Unit plugin should be copied to '/Library/Audio/Plug-Ins/Components' — available to all users — or '~/Library/Audio/Plug-Ins/Components' ("~" represents your home directory) — available only to the current user. It needs a compatible DAW, like Logic, Live or Digital Performer, amongst others.

The *spatium · live* and *OSCsend* max for live devices need a valid license of Max for Live or Live 9 Suite, which includes Max for Live.

3.4 *spatium · max*

These are Max 6 (they work with Max 7) objects and need a 32-bit version of Max.

4. FAQ and troubleshooting

Where is the Windows version?

Unfortunately, I don't own or have regular access to a Windows PC. Nevertheless, Paul Christian was kind enough to compile the Max objects for windows and that should make the *spatium* renderers easy to compile using Max on Windows. The *spatium* interfaces are Processing sketches and should compile without any issue on Windows.

Can I use the *spatium* objects/code/… inside my own project?

Yes you can, providing you acknowledge it and you release it under a compatible license — *spatium* is licensed under a Creative Commons Atribution-ShareAlike 3.0 license.

Can I use *spatium* to compose/perform my pieces and release them as a commercial CD/DVD?

Yes you can. I would appreciate if you acknowledge the use of *spatium* and would really like to get a copy!

I can't open …, a warning says that the application is damaged and cannot be opened or that the developer cannot be verified.

Go to your System Preferences, open 'Security & Privacy' and, inside this one, choose the 'General' panel. Check the option 'Anywhere' under 'Allow applications downloaded from:'. Your admin password may be required to unlock this option.

I can't send or receive OSC.

Please check your firewall settings and make sure all spatium components are able to accept incoming connections. Also, make sure UDP ports 11474, 11475, 11476 and 10000 are both open and free.

The renderer/interface/… doesn't receive OSC.

Only one application can listen on a given UDP port at any given time. If you already have one interface open, try changing its input port.

The multi-channel version of the soundflower driver starts crackling after some time.

This is a known problem with recent versions of this driver. There is a workaround:

- open the 'Audio Devices' window of the 'Audio MIDI Setup' application;

- click on the + sign on the bottom left corner and select 'Create Aggregate Device';

- select both the 'Soundflower (64ch)' and the Audio Device you're using for audio output;

- if you select first the 'Soundflower (64ch)' Audio Device, its channels will be the first on the new Aggregate Device, which will make configuring your audio setup easier;

- select the 'Clock Source' as the Audio Device you're using for audio output;

- activate the 'Drift Correction' for the 'Soundflower (64ch)' Audio Device;

- rename the new Aggregate Device as appropriate and close the 'Audio MIDI Setup' application;

- on the *spatium · ambi* renderer, open the 'audio settings' panel and set both the 'input device' and the 'output device' to the new Aggregate Device;

- set the output channels to the Audio Device you're using for audio output (channel 1 of the Audio Device correspondes to channel 65 of the Aggregate Device and so forth);

- set the output of your DAW to the new Aggregate Device.

The plugin doesn't alter the audio/doesn't receive midi.

Neither the *spatium · track* Audio Unit plugin, nor the *spatium · live* max for live device make any change to the audio. Please check their respective manual pages to see how they work.

5. Spatialization renderers reference

5.1 *spatium · ambi*

This renderer receives up to 16 mono audio channels and encodes them into an Ambisonics soundfield with distance encoding, using a technique described on the max objects page. The spatial information to encode each channel is received via OSC. The resulting soundfield can be decoded to several loudspeaker layouts, enabling the composition of spatial music with (almost) any loudspeaker setup and the fine-tuning of the piece to any given performance setup. To receive the audio from other applications, one should use an audio routing utility like Soundflower or Jack OS X.

interface

The top part of the interface shows the 16 input channels that feed the encoder, each with its own azimuth, elevation and radius displays, along with an input level meter. The spatial information can be edited directly on these displays, but it will be overwritten by the incoming OSC messages.

The bottom part of the interface is divided in two parts: the three left columns of buttons present options that affect the encoding of the soundfield and the two right columns of buttons present options that affect the decoding of the soundfield. In the middle, one can find level meters and a master fader, both below a menu of loudspeaker setups — currently: stereo (default), binaural, square, pentagon, surround, hexagon, octagon, dodecagon, hexadecagon, hexahedron, octahedron, dodecahedron and icosahedron. The 3D platonic solids have the speakers on the faces instead of on the vertices, so the prefix of each solid's name actually conveys the number of speakers to use (as it happens with the polygons). On the Max Window (available via cmd + M or Window -> Max Window), one can find the angles for the placement of each speaker.

OSC receive

This renderer receives OSC messages on port 11475.

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation and radius
/spatium/#1/azimuth f	with #1 being the channel, from 1 to 16, and f being the azimuth
/spatium/#1/elevation f	with #1 being the channel, from 1 to 16, and f being the elevation
/spatium/#1/radius f	with #1 being the channel, from 1 to 16, and f being the radius

buttons: external input

channel input on/off	when on, the 16 audio inputs will feed the encoder if being used inside Max, one can use send~ sp1, sp2,… to feed the encoder
ambi input on/off	when on, the first 13 audio inputs will feed the decoder directly, bypassing the encoding stage

buttons: spatial effects

granular on/off	when on, the 16 outputs of the granular engine will feed the encoder
granular settings	shows the granular settings panel, described below
spectral on/off	when on, the first audio input will feed the spectral engine and its 16 outputs will feed the encoder
spectral settings	shows the spectral settings panel, described below

buttons: ambi file record · play

choose record file	opens a file chooser to select a file name and location to record the encoded soundfield as a 13-channel, interleaved .aif file by adding '-16', '-24' or '-32' as a suffix to the name of the output file, it will be saved, respectively, as a 16-bit integer (default), 24-bit integer or 32-bit floating-point audio file
file record	toggles the recording of the soundfield to the selected audio file
choose play file	opens a file chooser to select an audio file (13-channel, interleaved and uncompressed) to be played as a pre-encoded soundfield
file play	toggles the playing of the selected audio file

buttons: ambi decoding

audio on/off	toggles the audio processing
decoder settings	shows the decoder settings panel, described below
reverb settings	shows the reverb settings panel, described below
audio settings	shows the audio settings panel, described below

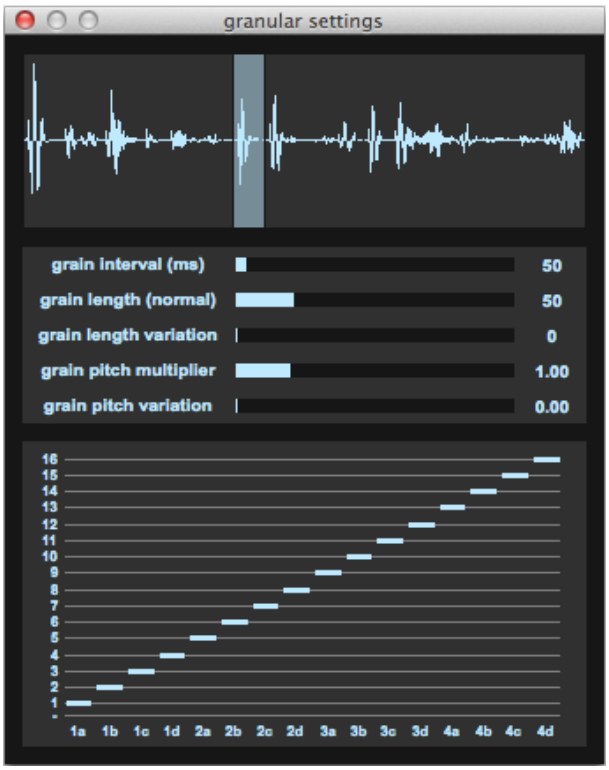
buttons: decoded record

choose file	opens a file chooser to select a file name and location to record the decoded output as an x-channel, interleaved .aif file (with x being the current number of loudspeakers) by adding '-16', '-24' or '-32' as a suffix to the name of the output file, it will be saved, respectively, as a 16-bit integer (default), 24-bit integer or 32-bit floating-point audio file
file record	toggles the recording of the decoded output to the selected audio file
open preset file	opens a file chooser to select a preset file to load the granular, spectral, decoder and reverb settings
save preset file	opens a file chooser to select a preset file to save the granular, spectral, decoder and reverb settings

granular settings panel

This granular engine uses objects from the granular toolkit to perform granular synthesis on an audio file. The top part of the interface shows the waveform: dropping an uncompressed audio file on top of this waveform loads an audio file and the starting points of the grains are chosen randomly inside the selected part of the audio.

Each of the 4 granular generators' output is iterated between 4 outputs, giving a total of 16 grain outputs that can be assigned to any of the 16 encoder inputs using the matrix on the bottom.

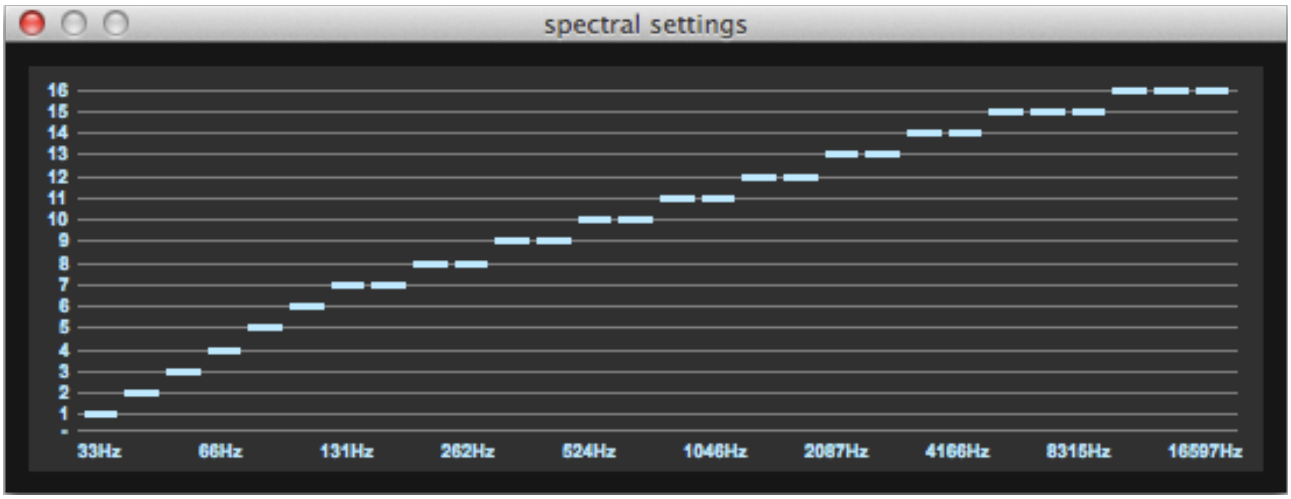


parameters

grain interval (ms)	the interval between grains
grain length (normal)	the length of the grains clicking over the fader name 'grain length (normal)' changes it to 'grain length (reverse)' and plays each grain backwards
grain length variation	the amount by which each grain's length can vary
grain pitch multiplier	the transposition of each grain, with 1.0 being the original speed, 2.0 sounding an octave above and 0.5 sounding an octave below
grain pitch variation	the amount be which each grain's pitch multiplier can vary

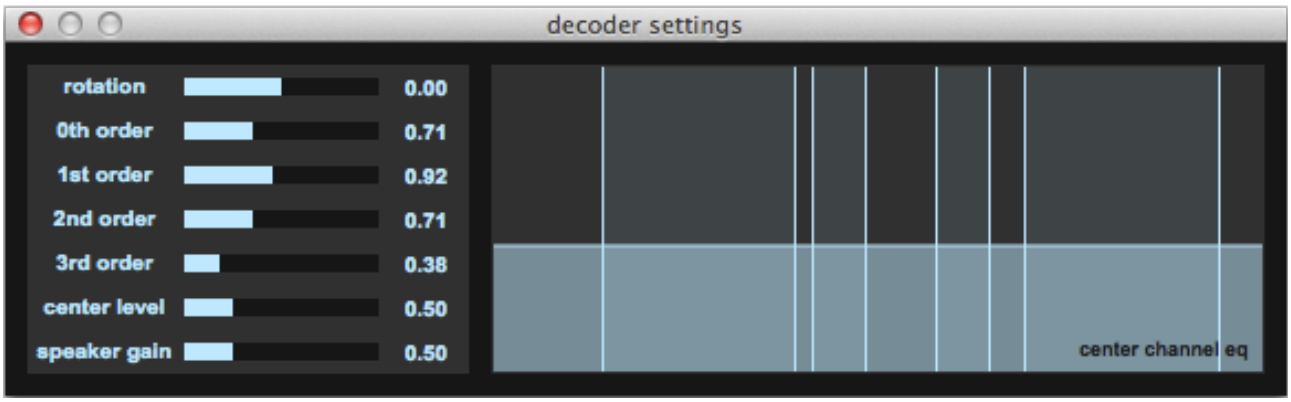
spectral settings panel

This spectral engine divides the 1st channel of the live audio input into 28 frequency bands that can be individually silenced or assigned to any of the 16 encoder inputs using the matrix shown.



decoder settings panel

This panel enables the control over the rotation of the soundfield, the decoding order weights and the equalization of the center channel, useful to simulate distance cues.

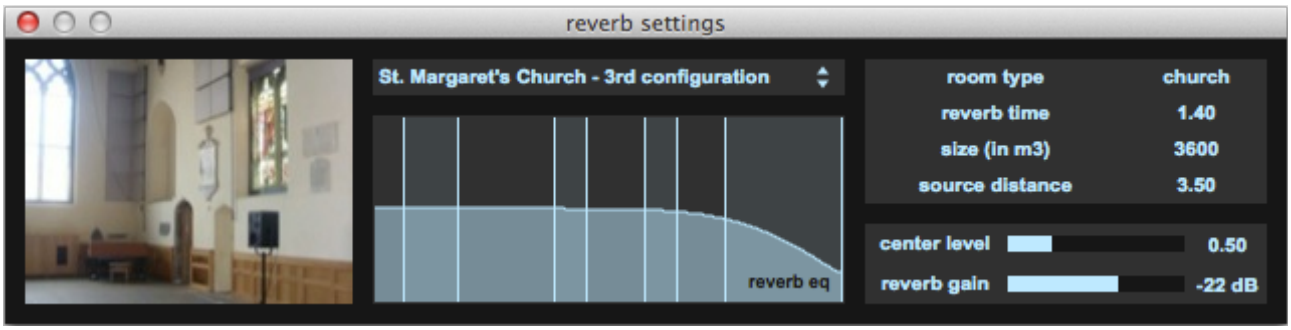


parameters

rotation	counter clockwise rotation of the soundfield
0th, 1st, 2nd and 3rd order	the weight of each Ambisonics order these values can be inactive if the current loudspeaker setup doesn't have enough resolution to support a specific order, in which case the correspondent value is grayed out
center level	the weight of the center portion of the soundfield
speaker gain	the gain applied to each speaker to avoid clipping

reverb settings panel

This reverb engine is a convolution reverb that uses HISSTools' Max objects and the B-format impulse responses available at the Open AIR Library. In order not to lose spatial resolution, the diffuse reverb is always added to the dry signal. The left part of the interface shows a picture of the currently selected space, the middle part a menu to select the current space and a control over the equalization of the reverb and the right part some information on the selected space, along with 2 controllable parameters, described below.

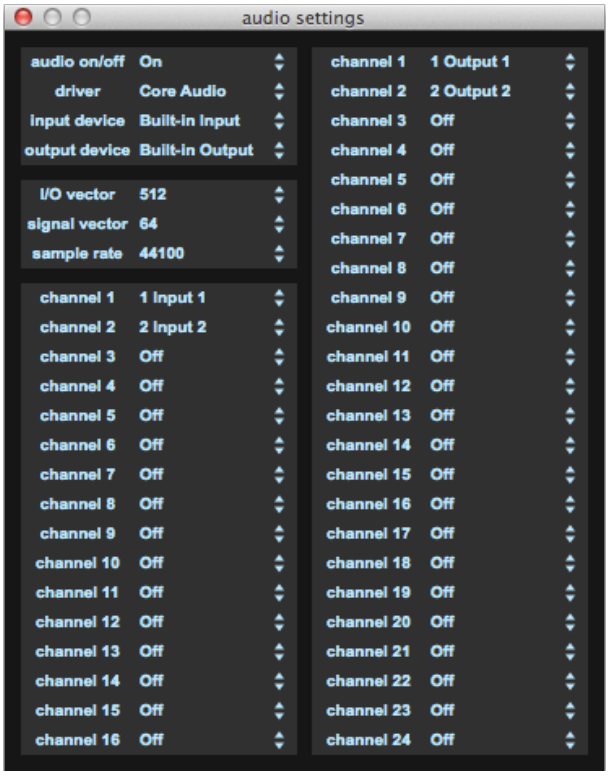


parameters

center level	the amount of the center portion of the soundfield that is feed into the reverb
reverb gain	the gain of the 100% wet reverb, added to the dry signal

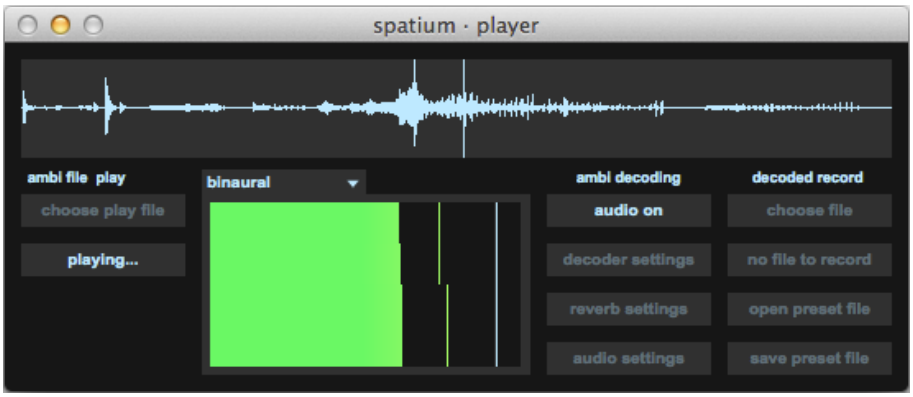
audio settings panel

This panel enables the control over the audio settings, the input and output channels. The left part of the interface shows menus to select the driver, input and output devices, vector sizes, sample rates and the input channels assignments. The right part shows the output channel assignments. The binaural and the reverb convolution engines use 44100Hz audio samples, so the sample rate should be kept at this value if they are to be used.



5.2 *spatium · player*

This renderer is basically a play-only, simplified version of *spatium · ambi*, made for the performance of a pre-encoded soundfield. This soundfield can be decoded to several loudspeaker layouts and fine-tuned to any given performance setup. The preset files are interchangeable between this renderer and *spatium · ambi*.



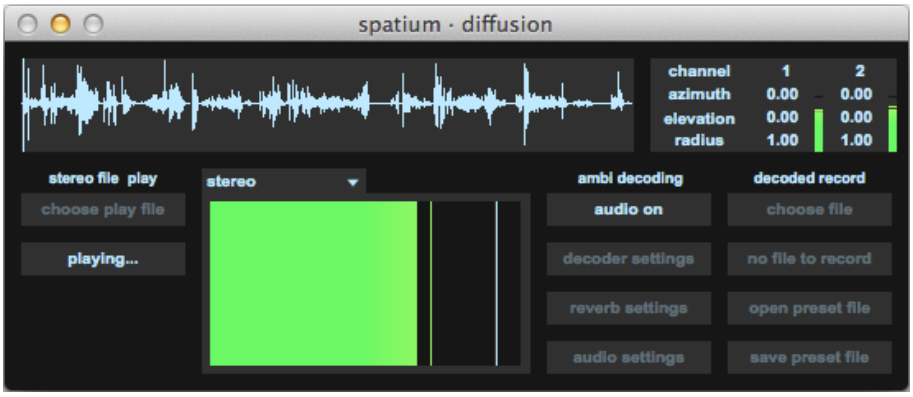
interface

The top part of the interface shows the waveform of the 1st channel (the W channel, or omnidirectional component) of the audio file. Clicking on any part of this waveform will play the audio file from that point onwards.

The bottom part of the interface is identical to that of *spatium · ambi*, with the exception of the audio input and spatial effects sections.

5.3 *spatium · diffusion*

This renderer plays a stereo file, encoding its two channels inside an ambisonics soundfield, being directly controllable by the *spatium* interfaces. This soundfield can be decoded to several loudspeaker layouts and fine-tuned to any given performance setup. The preset files are interchangeable between this renderer and *spatium · ambi*.



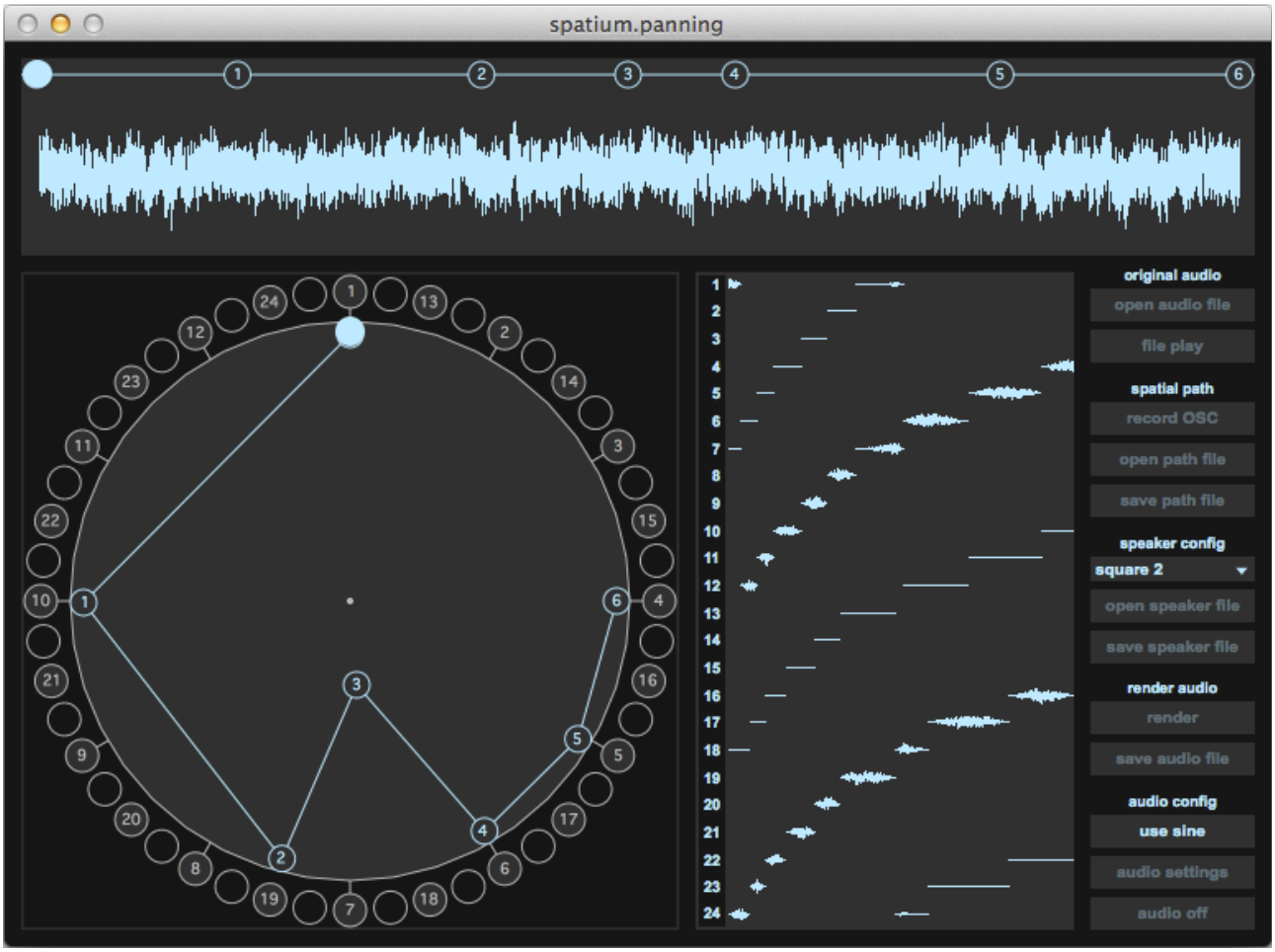
interface

The top part of the interface shows the waveform of the 1st channel of the stereo audio file. Clicking on any part of this waveform will play the audio file from that point onwards. Next to it there are the 2 input channels that feed the encoder, each with its own azimuth, elevation and radius displays, along with an input level meter. The spatial information can be edited directly on these displays, but it will be overwritten by the incoming OSC messages.

The bottom part of the interface is identical to that of *spatium · ambi*, with the exception of the audio input and spatial effects sections.

5.4 *spatium · panning*

This renderer takes an audio file and spatializes it to any 2D, concentrical loudspeaker setup (up to 24 channels) via APbSP. The spatialization path can be constructed and edited as a polygonal chain synchronized with the original audio file. It can also receive OSC from *spatium*'s interfaces to be recorded as the path. The end result can then be rendered as a set of mono audio files (one for each speaker) for further editing and mixing on any DAW.



interface

The top part of the interface shows the waveform of the current audio file, which can be loaded via drag&drop. On top of this waveform, there is a timeline of numbered anchors that correspond to the anchors along the path visible below.

The bottom part of the interface is divided in three parts. On the left there is an interface to control the spatial path and the loudspeaker location. On the middle there is a visualization of the rendered audio channels (up to 24). On the right there is a column of buttons that control several functions, as described below.

mouse click: top timeline

on the line	creates a new anchor
inside an anchor	moves the anchor shift + click: deletes the anchor

mouse click: bottom left interface

inside an anchor	moves the anchor
inside the surrounding circles	toggles the respective loudspeaker on or off

buttons: original audio

open audio file	opens a file chooser to select an audio file to spatialize (if multi-channel, only the first channel will be used)
file play	toggles the playing of the selected audio file

buttons: spatial path

record OSC	deletes the current path and records the incoming spatial data as a 30 segment path
open path file	opens a previously saved spatialization path
save path file	saves the current spatialization path as a YAML file

buttons: speaker config

loudspeaker config menu	menu to select one of the default loudspeaker setups
open speaker file	opens a previously saved loudspeaker setup
save speaker file	saves the current loudspeaker setup as a YAML file

buttons: render audio

render	renders the current audio file to the current loudspeaker setup using the current spatialization path note: it can take a while, shows 'rendering…' until its done
save audio file	saves the rendered audio channels as numbered mono audio files

buttons: audio config

use sine/srt	toggles between the use of sine or square root amplitude panning formulas
audio settings	shows the audio settings panel, described below
audio on/off	toggles the audio processing

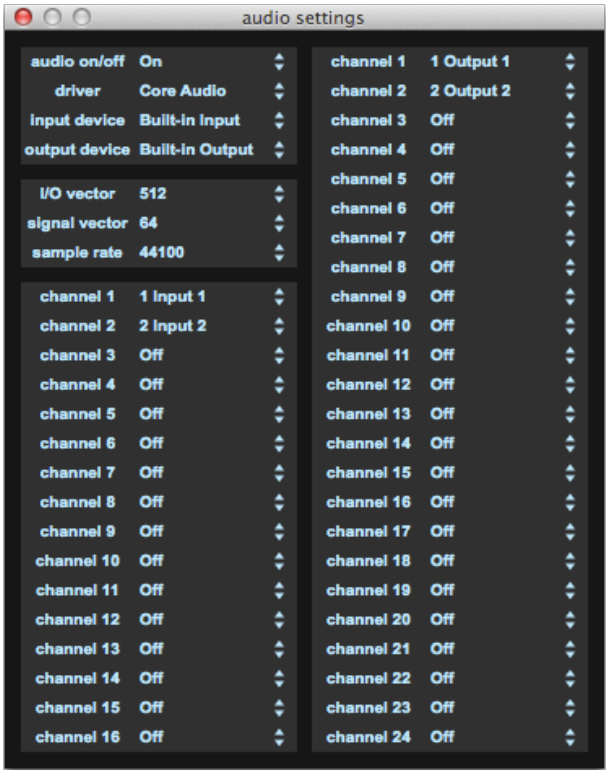
OSC receive

This renderer receives OSC messages on port 11475. It is limited to the first spatium channel and it ignores the elevation, as this spatial information is to be recorded as a 2D path.

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation (ignored) and radius
/spatium/#1/azimuth f	with #1 being the channel, from 1 to 16, and f being the azimuth
/spatium/#1/radius f	with #1 being the channel, from 1 to 16, and f being the radius

audio settings panel

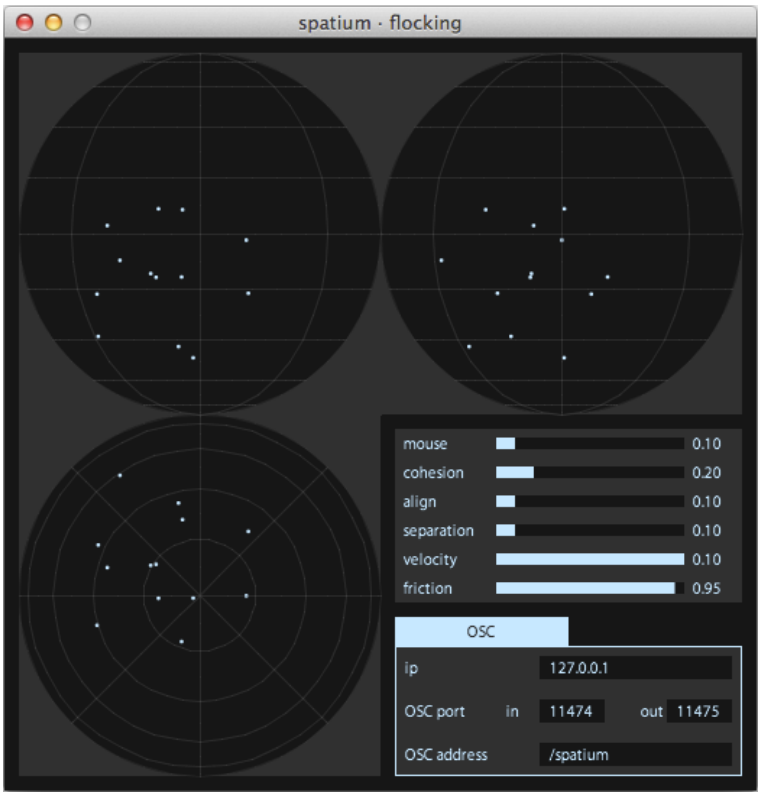
This panel enables the control over the audio settings, the input and output channels. The left part of the interface shows menus to select the driver, input and output devices, vector sizes, sample rates and the input channels assignments (unused, as this interface doesn't have an audio input). The right part shows the output channel assignments.



6. Spatialization interfaces reference

6.1. *spatium · flocking*

This interface uses a 3D flocking algorithm to move up to 16 birds in a 3D space. The control over a 3D space using a 2D computer environment is made possible by interacting with one of three 2D projections. These are, from bottom left and on clockwise motion, the projection on the horizontal plane, the projection on the frontal plane and the projection on the median plane. The menu on the bottom right corner can be hidden to reveal a one-point perspective 3D view over the scene.



key press

space bar	alternately starts or stops the simulation

mouse click

inside the soundfield	when stopped: creates a new bird when running: sets the mouse position, to which the birds will be attracted
inside the menu/3D view	alternately reveals or hides the menu

parameters

mouse	the mouse attraction factor, from 0.0 to 1.0
cohesion	the bird cohesion factor, from 0.0 to 1.0
align	the bird alignment factor, from 0.0 to 1.0
separation	the bird separation factor, from 0.0 to 1.0
velocity	the speed limit, from 0.0 to 0.1
friction	the friction factor, from 0.0 (stopped) to 1.0 (no friction)

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address can be defined in 'OSC address' (default is "/spatium").

/spatium/#1/aer f1 f2 f3	with #1 being the bird number, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation and radius

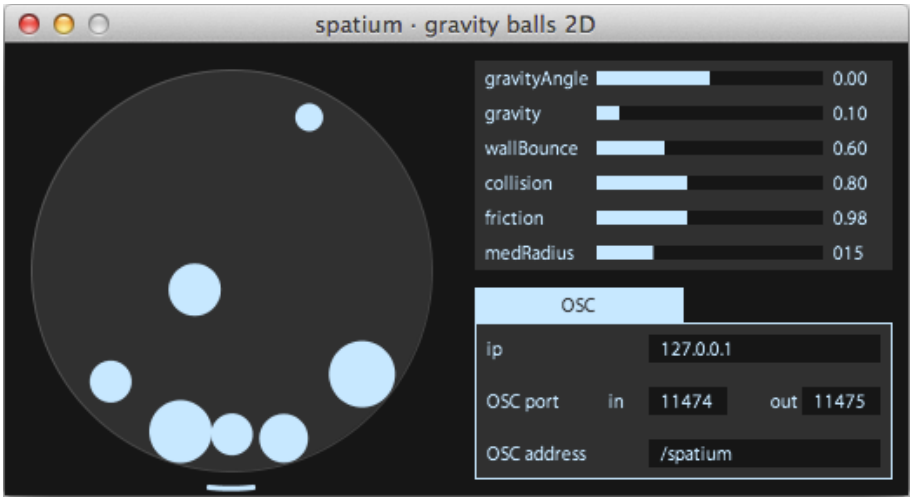
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/run i	with i being either 0 or 1, respectively starting or stopping the simulation
/spatiumControl/mouse f	controls the homonymous parameter with its respective range
/spatiumControl/cohesion f	controls the homonymous parameter with its respective range
/spatiumControl/align f	controls the homonymous parameter with its respective range
/spatiumControl/separation f	controls the homonymous parameter with its respective range
/spatiumControl/velocity f	controls the homonymous parameter with its respective range
/spatiumControl/friction f	controls the homonymous parameter with its respective range

6.2. *spatium · gravity balls 2D*

This interface simulates up to 16 balls that are enclosed within a 2D soundfield. Each of these balls is attracted by a gravitational force whose azimuth is controllable. They bounce on the walls and when they collide with each other. A ball can be launched against the others using a controllable slingshot.



key press

space bar	alternately starts or stops the simulation

mouse click

inside the soundfield	when stopped: creates a new ball
around the soundfield	sets the gravity angle
inside a ball	when stopped: moves the ball when stopped, right click: deletes the ball when running: starts a slingshot, ball is launched when mouse is released

parameters

gravityAngle	the angle of the gravity source, from -180.0 to 180.0
gravity	the relative force of gravity, from 0.0 to 1.0
wallBounce	the relative force of wall bouncing, from 0.0 to 1.0
collision	the relative force of bouncing between balls, from 0.0 to 1.0
friction	the overall friction, from 0.9 (negative acceleration) to 1.1 (positive acceleration) clicking over the fader name 'friction' resets the value to the default 0.98
medRadius	the median radius of the balls in pixels, from 10 to 30

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address can be defined in 'OSC address' (default is "/spatium").

/spatium/#1/aer f1 f2 f3	with #1 being the bird number, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation and radius

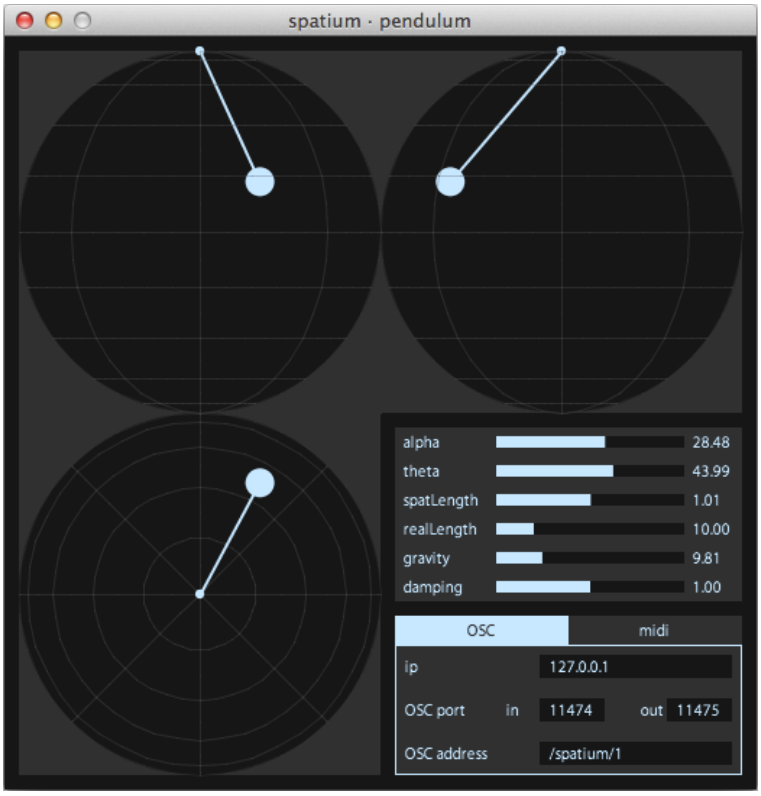
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/run i	with i being either 0 or 1, respectively starting or stopping the simulation
/spatiumControl/gravityAngle f	controls the homonymous parameter with its respective range
/spatiumControl/gravity f	controls the homonymous parameter with its respective range
/spatiumControl/wallBounce f	controls the homonymous parameter with its respective range
/spatiumControl/collision f	controls the homonymous parameter with its respective range
/spatiumControl/friction f	controls the homonymous parameter with its respective range
/spatiumControl/medRadius f	controls the homonymous parameter with its respective range

6.3. *spatium · pendulum*

This interface simulates up to 16 balls that are enclosed within a 2D soundfield. Each of these balls is attracted by a gravitational force whose azimuth is controllable. They bounce on the walls and when they collide with each other. A ball can be launched against the others using a controllable slingshot.



key press

space bar	alternately starts or stops the simulation

mouse click

inside the pendulum anchor	when stopped: moves the anchor
inside the pendulum ball	when stopped: moves the ball
inside the menu/3D view	alternately reveals or hides the menu

parameters

alpha	the pendulum angle on the horizontal plane, from -180.0 to 180.0 disabled when the pendulum is running
theta	the pendulum angle on the frontal plane, from -180.0 to 180.0 disabled when the pendulum is running
spatLength	the length of the pendulum inside the soundfield, in relation to its radius, from 0.0 to 2.0
realLength	the length of the simulated pendulum, in meters, from 0.0 to 50.0
gravity	the gravity acceleration, in m/s2, from 0.0 to 40.0 clicking over the fader name 'gravity' resets the value to the default 9.81
damping	the overall damping, from 0.98 (negative acceleration) to 1.02 (positive acceleration) clicking over the fader name 'damping' resets the value to the default 1.0

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is active by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

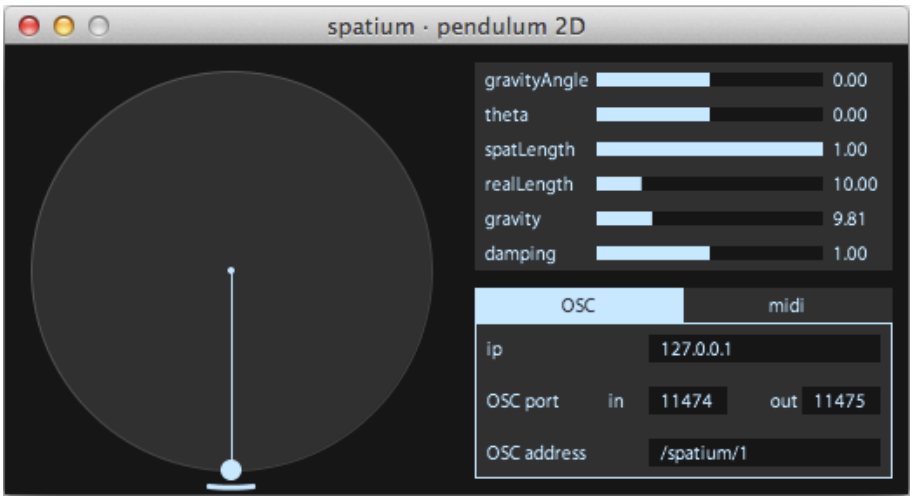
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/run i	with i being either 0 or 1, respectively starting or stopping the simulation
/spatiumControl/alpha f	controls the homonymous parameter with its respective range disabled when the pendulum is running
/spatiumControl/theta f	controls the homonymous parameter with its respective range disabled when the pendulum is running
/spatiumControl/spatLength f	controls the homonymous parameter with its respective range
/spatiumControl/realLength f	controls the homonymous parameter with its respective range
/spatiumControl/gravity f	controls the homonymous parameter with its respective range
/spatiumControl/damping f	controls the homonymous parameter with its respective range

6.4. *spatium · pendulum 2D*

This interface uses a 2D gravity pendulum simulation, with controllable gravity and damping.



key press

space bar	alternately starts or stops the simulation

mouse click

inside soundfield	when stopped: moves the ball
around the soundfield	when running: sets the gravity angle

parameters

gravityAngle	the angle of the gravity source, from -180.0 to 180.0 clicking over the fader name 'gravityAngle' resets the value to the default 0.0
theta	the pendulum angle, from -180.0 to 180.0 disabled when the pendulum is running
spatLength	the length of the pendulum inside the soundfield, in relation to its radius, from 0.0 to 1.0
realLength	the length of the simulated pendulum, in meters, from 0.1 to 50.0
gravity	the gravity acceleration, in m/s2, from 0.0 to 40.0 clicking over the fader name 'gravity' resets the value to the default 9.81
damping	the overall damping, from 0.98 (negative acceleration) to 1.02 (positive acceleration) clicking over the fader name 'damping' resets the value to the default 1.0

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is inactive by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

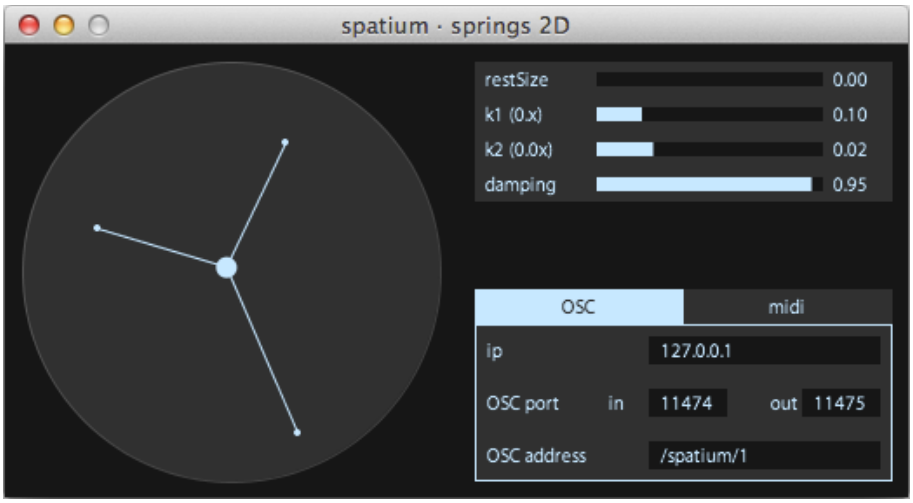
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/run i	with i being either 0 or 1, respectively starting or stopping the simulation
/spatiumControl/gravityAngle f	controls the homonymous parameter with its respective range
/spatiumControl/theta f	controls the homonymous parameter with its respective range disabled when the pendulum is running
/spatiumControl/spatLength f	controls the homonymous parameter with its respective range
/spatiumControl/realLength f	controls the homonymous parameter with its respective range
/spatiumControl/gravity f	controls the homonymous parameter with its respective range
/spatiumControl/damping f	controls the homonymous parameter with its respective range

6.5. *spatium · springs 2D*

This interface simulates up to 6 springs that connect as many anchors to the main ball.



key press

space bar	alternately starts or stops the simulation

mouse click

inside soundfield	when stopped: creates a new anchor when running: pulls the spring(s), ball is launched when mouse is released
inside an anchor	when stopped: moves the anchor

parameters

restSize	the resting size of the spring(s) in pixels, from 0 to 120 clicking over the fader name 'restSize' resets the value to the default 0
k1 (0.x)	the decimal part of the spring constant, from 0.0 to 0.5 clicking over the fader name 'k1 (0.x)' resets the value to the default 0.1
k2 (0.0x)	the centesimal part of the spring constant, from 0.01 to 0.05 clicking over the fader name 'k2 (0.0x)' resets the value to the default 0.02
damping	the damping factor, from 0.0 (stopped) to 1.0 (no friction) clicking over the fader name 'damping' resets the value to the default 0.95

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is inactive by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

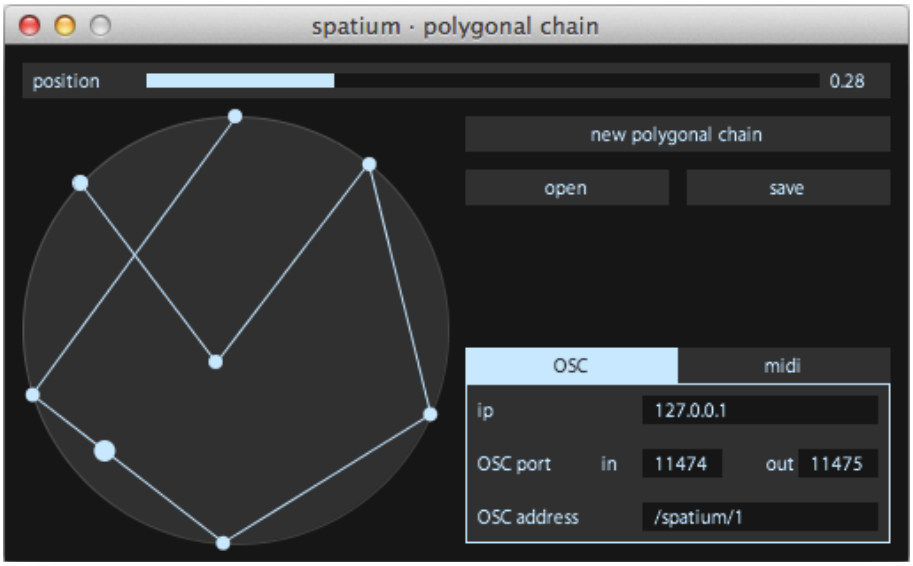
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/run i	with i being either 0 or 1, respectively starting or stopping the simulation
/spatiumControl/restSize f	controls the homonymous parameter with its respective range
/spatiumControl/k1 f	controls the homonymous parameter with its respective range
/spatiumControl/k2 f	controls the homonymous parameter with its respective range
/spatiumControl/damping f	controls the homonymous parameter with its respective range

6.6. *spatium · polygonal chain*

This interface enables the creation and performance of a polygonal chain as a spatialization path.



mouse click

inside soundfield	creates a new segment at the end of the chain
inside a ball	moves the ball and the respective segment right click: deletes the ball and the respective segment
on a point along a line	creates a new point at that position, dividing the respective segment in two

parameters

position	the position along the polygonal chain, from 0.0 to 1.0

buttons

new polygonal chain	restarts the polygonal chain
open	opens a polygonal chain file
save	saves a polygonal chain file

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is inactive by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

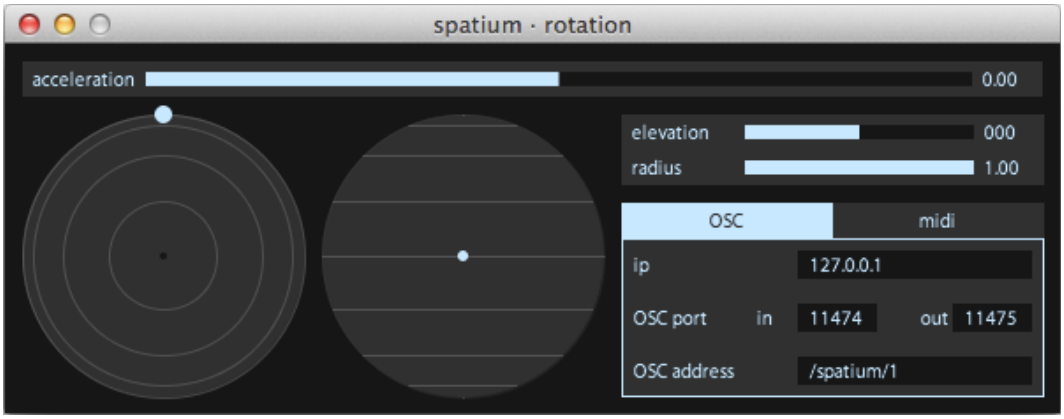
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/position	controls the homonymous parameter with its respective range

6.7. *spatium · rotation*

This interface enables the control over the angular acceleration on a horizontal plane, with controllable elevation and radius. The spatialization particle accelerates clockwise or counter clockwise, moving with either variable or uniform acceleration when the fader is respectively moving or left in a fixed position.



parameters

acceleration	the current acceleration, from -10.0 (negative acceleration) to 10.0 (positive acceleration) clicking over the fader name 'acceleration' resets the value to the default 0.0 (no acceleration, constant velocity)
elevation	the elevation, from -90.0 to 90.0 clicking over the fader name 'elevation' resets the value to the default 0.0
radius	the radius, from 0.0 to 1.0 clicking over the fader name 'radius' resets the value to the default 1.0

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is active by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

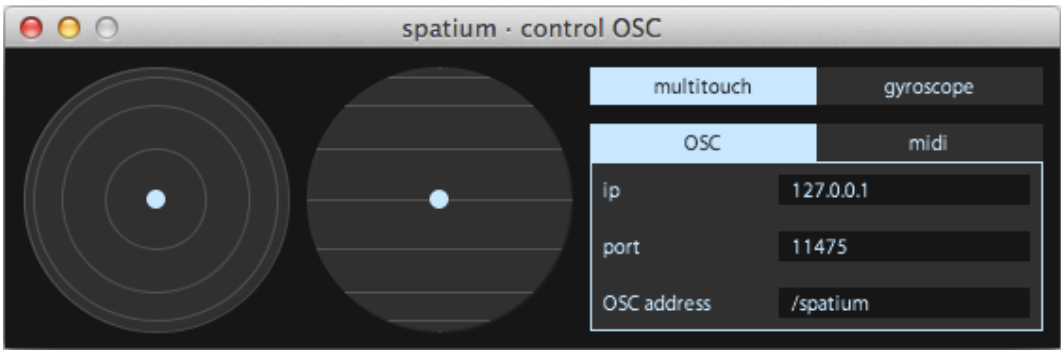
OSC receive

This interface receives OSC messages using the port defined in 'OSC port in' (default is 11474).

/spatiumControl/acceleration f	controls the homonymous parameter with its respective range
/spatiumControl/elevation f	controls the homonymous parameter with its respective range
/spatiumControl/radius f	controls the homonymous parameter with its respective range

6.8. *spatium · control OSC*

This interface has two modes that enable the control of spatium using Control, a free, open source and customizable touch controller that runs on both iOS and Android and outputs OSC via WiFi. The first is the multitouch mode, that enables the positioning of up to 11 particles on the horizontal plane via multitouch interaction with the iOs or Android device (it is better suited for a tablet). The second is the gyroscope mode, that enables the positioning of 1 particle on the spherical surface enclosing the soundfield by pointing the iOs or Android device to that direction (it is better suited for a phone or similar-sized device). In this last mode, it is important to start Control's Gyro + Accelerometer interface with the device pointing to the 0° azimuth, 0° elevation point.



buttons

multitouch	enables the multitouch mode, translating the OSC messages sent by Control's included MultiTouchXY interface
gyroscope	enables the gyroscope mode, translating the OSC messages sent by Control's included Gyro + Accelerometer interface

OSC send · multitouch mode

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address can be defined in 'OSC address' (default is "/spatium").

/spatium/#1/aer f1 f2 f3	with #1 being the bird number, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation and radius

OSC send · gyroscope mode

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send · gyroscope mode

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

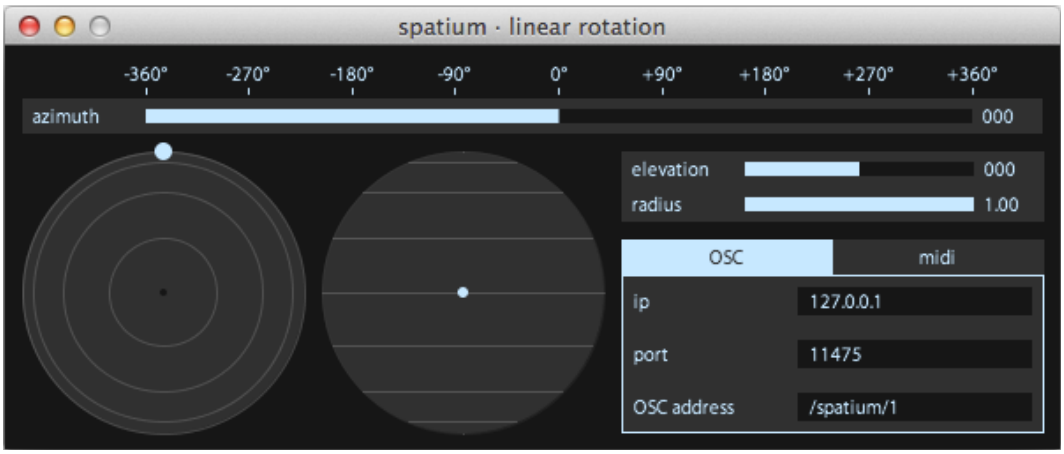
azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is active by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

OSC receive

The iOs or Android device should use the same WiFi as the computer running this interface and the Control app's Destination should be configured to the computer's ip address and to port 10000. This interface is configured to receive the standard messages sent by either the MultiTouchXY or the Gyro + Accelerometer interfaces included with Control.

6.9. *spatium · linear rotation*

This interface enables the control over the azimuth of the spatialization particle using a long horizontal fader spanning two complete revolutions, thus making it easier to perform circular spatialization movements with a mouse.



parameters

azimuth	the current azimuth, from -360 to 360 (two complete revolutions) clicking over the fader name 'azimuth' resets the value to the default 0
elevation	the elevation, from -90.0 to 90.0 clicking over the fader name 'elevation' resets the value to the default 0.0
radius	the radius, from 0.0 to 1.0 clicking over the fader name 'radius' resets the value to the default 1.0

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

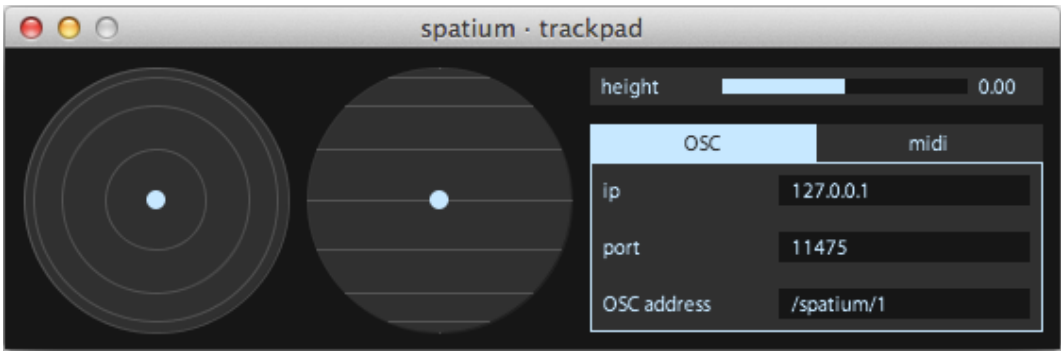
Midi send

When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is active by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

6.10. *spatium · trackpad*

This interface enables real-time, physical control over the spatialization of one particle using the MacBook trackpad or the Magic Trackpad as an absolute positioning touch controller.



key press

space bar	alternately starts or stops the simulation

trackpad

first finger	controls the horizontal position of the particle
second finger (optional)	controls the height of the particle along the vertical axis of the trackpad

OSC send

This interface sends OSC messages to the ip address defined in 'ip' (default is 127.0.0.1 — localhost) and to the port defined in 'OSC port out' (default is 11475). The first part of the address and the current channel can be defined in 'OSC address' (default is "/spatium/1").

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16 (the symbol '*' sends the message to all channels at once), and f1 f2 f3 being, respectively, its azimuth, elevation and radius

Midi send

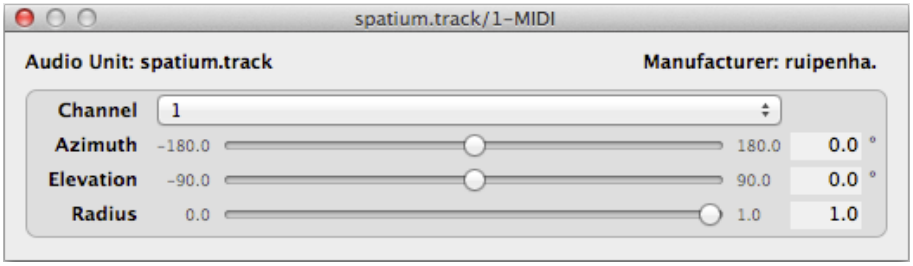
When the midi tab is selected, the interface stops sending OSC messages and sends the spatialization data via midi controller numbers to the selected midi interface and channel. The midi controller numbers can either be active (blue) or inactive (gray).

azimuth: 110	the azimuth is sent through midi controller number 110, with its value scaled from -180.0/180.0 to 0/127 it is active by default
elevation: 111	the elevation is sent through midi controller number 111, with its value scaled from -90.0/90.0 to 0/127 it is active by default
radius: 112	the radius is sent through midi controller number 112, with its value scaled from 0.0/1.0 to 0/127 it is active by default

7. Plugins reference

7.1. Audio Unit plugin

This Audio Unit plugin doesn't affect the audio. Instead, it can be placed in an audio track effects chain to send its three automatable parameters — azimuth, elevation and radius — via OSC, each every 10ms whenever one of the values has changed. Up to 16 instances of this plugin can be used simultaneously, but only one should use a given channel at a given time to avoid conflicting information.



parameters

Channel	the current channel, from 1 to 16 saved, but not automatable
Azimuth	the azimuth, from -180.0 to 180.0 saved and automatable
Elevation	the elevation, from -90.0 to 90.0 saved and automatable
Radius	the radius, from 0.0 to 1.0 saved and automatable

OSC send

This plugin sends OSC messages to the ip address 127.0.0.1 (localhost) and to port 11475.

/spatium/#1/azimuth f	with #1 being the channel, from 1 to 16, and f being the azimuth
/spatium/#1/elevation f	with #1 being the channel, from 1 to 16, and f being the elevation
/spatium/#1/radius f	with #1 being the channel, from 1 to 16, and f being the radius

Midi receive

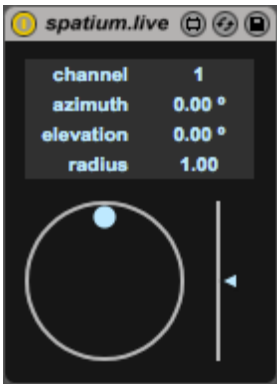
This plugin can record the midi controller values sent by the interfaces in midi mode, but the DAW must first be configured to control a specific instance of this plugin using these values. This needs to be done once per plugin instance. Please refer to your DAW's manual to learn how to configure the use of midi controllers.

7.2. Max for Live devices

These max for live devices do not affect the audio. Instead, they can be placed in an audio track effects chain to send and receive data via OSC. The *spatium · live* device receives, displays, stores and/or controls spatialization information to be sent via OSC. The *OSCsend* device sends up to three configurable parameters via OSC to any ip address and port. Their source is not distributed on the website, as all max for live content is distributed in source form.

7.2. *spatium · live*

The top part of the interface shows four parameters, described below. The bottom part of the interface has one 2D spatialization controller, controlling the azimuth and the radius, and one vertical fader controlling the elevation.



parameters

Channel	the current channel, from 1 to 16
Azimuth	the azimuth, from -180.0 to 180.0
Elevation	the elevation, from -90.0 to 90.0
Radius	the radius, from 0.0 to 1.0

OSC send

This plugin sends OSC messages to the ip address 127.0.0.1 (localhost) and to port 11475.

/spatium/#1/azimuth f	with #1 being the channel, from 1 to 16, and f being the current azimuth
/spatium/#1/elevation f	with #1 being the channel, from 1 to 16, and f being the current elevation
/spatium/#1/radius f	with #1 being the channel, from 1 to 16, and f being the current radius

OSC receive

This plugin receives OSC messages on port 11476, so the interfaces should be configured to send to this port in order to control this max for live device.

/spatium/#1/aer f1 f2 f3	with #1 being the channel, from 1 to 16, and f1 f2 f3 being, respectively, its azimuth, elevation and radius
/spatium/#1/azimuth f	with #1 being the channel, from 1 to 16, and f being the azimuth
/spatium/#1/elevation f	with #1 being the channel, from 1 to 16, and f being the elevation
/spatium/#1/radius f	with #1 being the channel, from 1 to 16, and f being the radius

7.2. OSCsend

This max for live device is not a part of spatium, instead being a generic device that can record up to three automatable parameters to be sent via OSC. It can be used to control spatium's interfaces by setting the appropriate addresses as parameters.

parameters

/address	the address of each parameter's OSC message
f1 * i1 + i2 = f2	f1 is the automatable value of each parameter, always between 0.0 and 1.0 i1 is the multiplier, to enable different ranges of the value to be sent i2 is the offset, to change the range of the value to be sent f2 is the result and the final value to be sent example: if i1 = 360 and i2 = -180, f2 will vary between -180.0 and 180.0
ip	the ip address to send the messages to (default is 127.0.0.1 — localhost)
port	the port to send the messages to (default is 11474, the default input port of the interfaces)