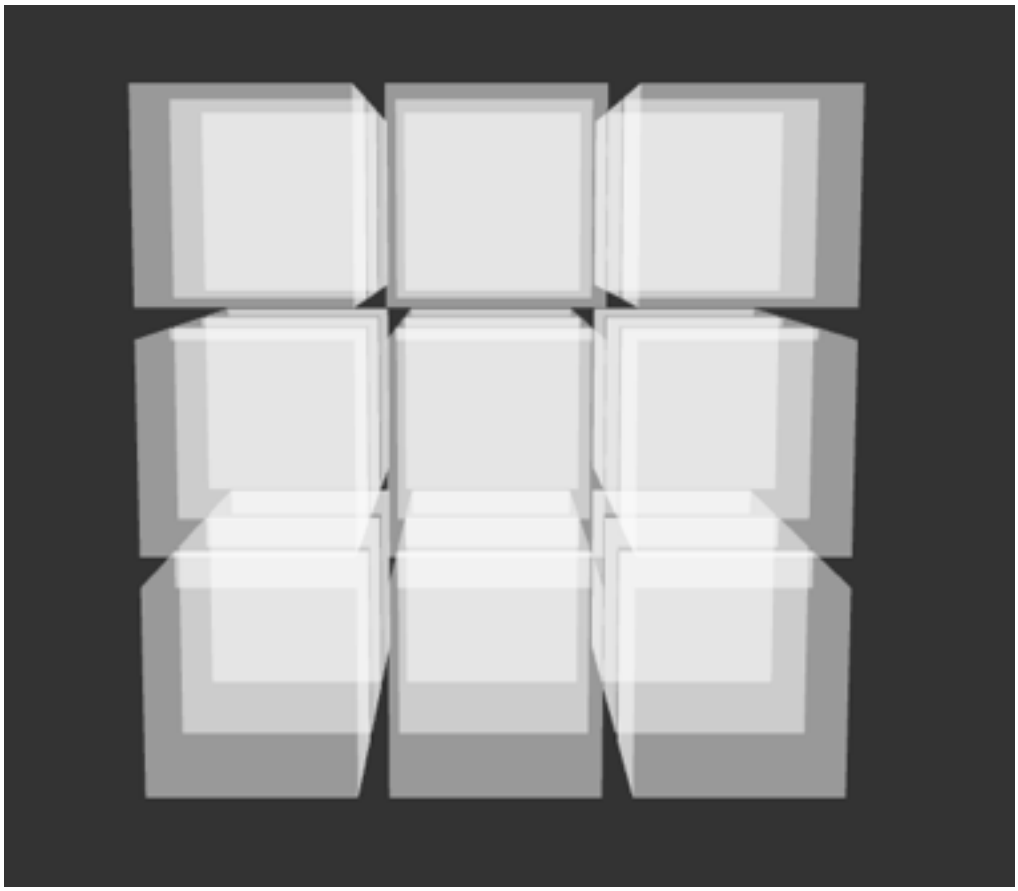
SONORIUM

(Version 1.2.1 for OSX)



USER MANUAL (ENG)

Digitópia — Educative Service of Casa da Música, 2016

1. WELCOME 5

2. FIRST STEPS 5

2.1. Requirements *5*

2.2. Dependencies *5*

2.3. How to install *6*

2.4. How to use *6*

3. BASIC USAGE 7

3.1. Concept *7*

3.2. How to set Sonorium in a physical space : limitations and recommendations *8*

3.3. Out of the box (presets) *8*

3.4. How to configure the sound output *8*

3.5. Accessing help *9*

4. ADVANCED USAGE 9

4.1. How to setup Kinect's capture *9*

4.2. How to set the capture area *9*

4.3. How to set the virtual grid *9*

4.4. Configuring the sampler *10*

4.5. Making preset banks / adding, saving and deleting presets *10*

4.5.1. Making a preset bank *10*

4.5.2. Opening a preset bank *10*

4.5.3. Creating a new preset *10*

4.5.4. Saving a preset *11*

4.5.5. Changing the preset name *11*

4.5.6.How to delete a preset *11*

4.6.Extending functionality (MIDI, OSC, grid and skeleton data) *11*

4.6.1. MIDI Output *11*

4.6.2.OSC — Tridimensional grid *12*

4.6.3.OSC — Skeleton data *13*

4.6.4. Remote control *13*

4.7.Software architecture *16*

4.7.1. Video Input and Capture Area *16*

4.7.2. Virtual Cube Grid and Output *17*

4.8.About the perspective correction *18*

4.9.About presets *19*

5. LABEL **20**

5.1. Sections *20*

5.2. Presets *21*

5.3. Configuration panel *22*

5.4. Kinect Configuration *22*

5.5. Capture Area *24*

5.6. 3D Virtual Grid *25*

5.7. Output *26*

5.8. Sampler *27*

5.9. MIDI *29*

6. Frequently asked questions (FAQ) **31**

7. ACKNOWLEDGEMENTS **33**

8. LICENSE **33**

1. WELCOME

Sonorium is an application developed by Digitópia (Education Service of *Casa da Música*) that aims to create a sound map of a physical space, allowing to trigger specific sounds according to the presence and position in space of the participants, which in a whole contribute to a collective composition. This software uses a depth sensing camera to create a virtual grid, allowing to configure a tridimensional sound map of the space where it sits.

To download Sonorium-1.2.1 visit the following link:
https://github.com/Digitopia/Sonorium/releases

To get the source code go to the following link and press the 'Download ZIP' button:
https://github.com/Digitopia/Sonorium

After downloading you should read the following chapters on how to install Sonorium.

Thanks for your preference!

2. FIRST STEPS

2.1. Requirements

To use Sonorium you will need a computer with OSX 10.6.8 (or higher) and a Microsoft Kinect Camera (or an Asus X-tion PRO, although this camera was not tested).

Note that Sonorium's sampler loads audio to your computer's RAM so, when using the sampler, you are limited by your computer's RAM memory (please note that your operative system and other applications also use your computer's RAM memory).

2.2. Dependencies

To use Sonorium you will need to download and install the following dependencies:

- libusb-1.0.9
- OpenNI 1.5.4.0
- PrimeSense NITE 1.5.2.21
- PrimeSense Sensor KinectMod v0.93 (based on 5.1.2.1)

You can find download links at https://github.com/Digitopia/Sonorium/wiki/How-to-install-Sonorium-%3F-(mac-OSX-only) or at http://hidale.com/jit-openni/

Sonorium is an open source project, although you will need MaxMSP (version 6.0.7 or higher) in order to access and modify the code.

2.3. How to install

To correctly install Sonorium's dependencies please follow the instruction below in the following order:

1. install *libusb-1.0.9* — double-click *libusb-1.0.9.pkg* package and follow the provided instructions in the installer;

2. install *OpenNI 1.5.4.0* — double-click *OpenNI-Bin-Dev-MacOSX-v1.5.4.0.tar.zip* package, then open the Terminal and navigate to the folder you have just uncompressed. To do so write *"cd "* in the Terminal window, drag the folder to that window and press *<enter>*. Then write *"sudo ./install.sh"*, press enter again and insert your password to proceed with the installation.

3. install *PrimeSense NITE 1.5.2.21* — double-click the *NITE-Bin-MacOSX-v1.5.2.21.tar* package, then open the Terminal and navigate to the folder you have just uncompressed. To do so write *"cd "* in the Terminal window, drag the folder to that window and press *<enter>*. Then write *"sudo ./install.sh"*, press enter again and insert your password to install.

4. install PrimeSense Sensor KinectMod v0.93 : double-click the *SensorKinect093-Bin-MacOSX-v5.1.2.1.tar.bz2* package, then open the Terminal and navigate to the folder you have just uncompressed. To do so write *"cd "* in the Terminal window, drag the folder to that window and press *<enter>*. Then write *"sudo ./install.sh"*, press enter again and insert your password to proceed.

2.4. How to use

To use Sonorium you will have to install the dependencies as described in the previous subchapter and download the installer from https://github.com/Digitopia/Sonorium/releases. Then you just need to open the installer ("Sonorium Installer.dmg") to your "Applications" folder.

If you pretend to modify Sonorium's code you will need to install MaxMSP (version 6.0.7 or higher) and the following libraries: *TA-Lib (*https://github.com/p1nh0/TA-Library*)* and *Open Instruments (*https://github.com/p1nh0/OpenInstruments*)*, plus the *jit.openni v0.88* MaxMSP external for *OSX* available at http://hidale.com/jit-openni/*.*

3. BASIC USAGE

In this chapter we will describe the core concepts of Sonorium as well as a basic usage of the application.

3.1. Concept

Sonorium applies a tridimensional virtual grid to a physical space (see Figure 1 below) allowing to create a set of hotspots that are activated by the user's position, triggering a predetermined sound or MIDI note.

The application implements a perspective distortion correction, allowing to a create an interaction area corresponding to a cube or a parallelepiped if desired. This area can then be filled with a predetermined number of hotspots of equal dimensions. Thus making a 3D virtual grid where you can then map an audio sample or MIDI note to each one of these hotspots, triggering them whenever a user, or object (note that Sonorium does not discriminate between people, animals or objects) occupies that hotspot.

Sonorium allows the creation of 16 hotspots per axis *(x, y and z)*, making a total of 4096 available hotspots.

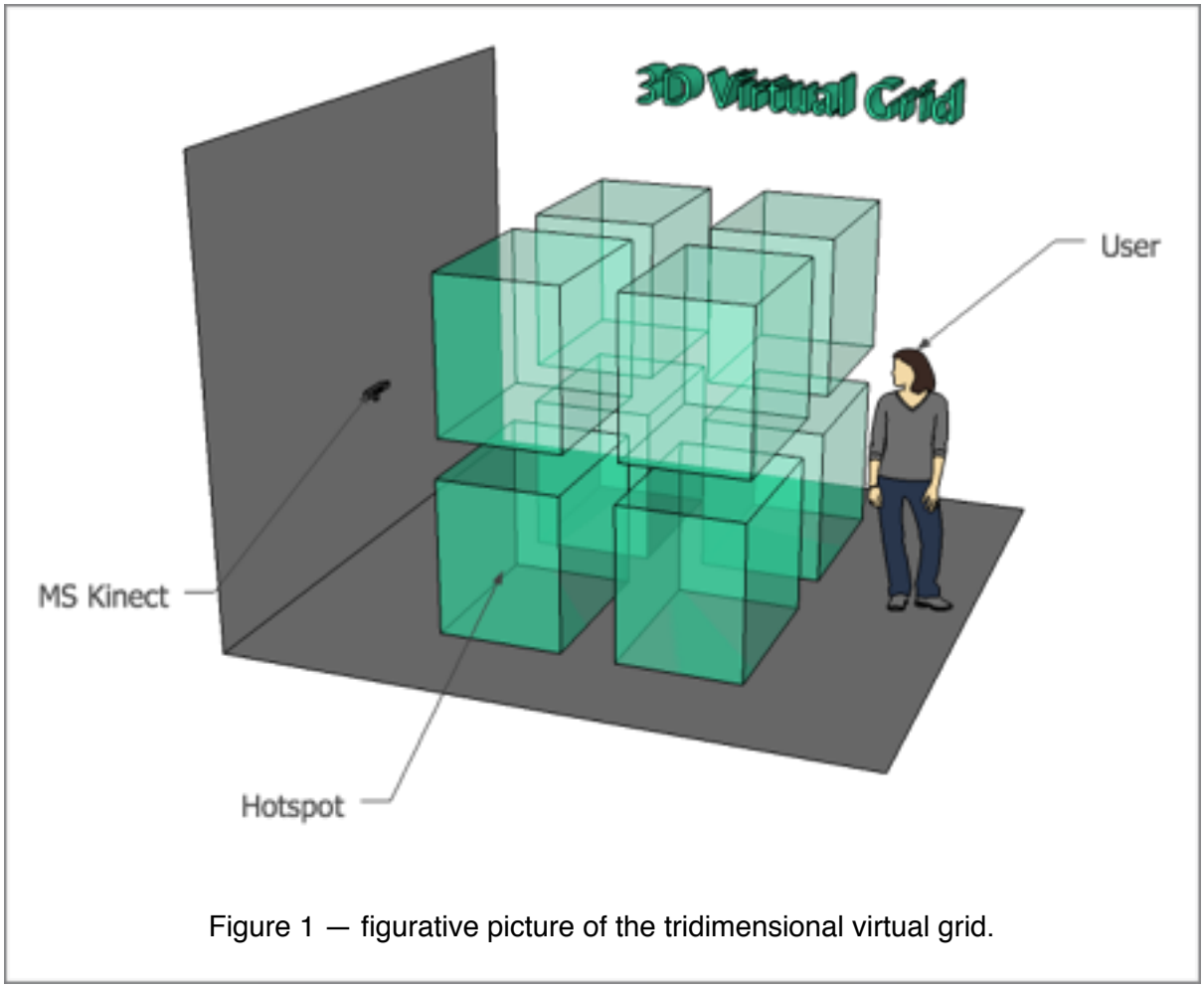


Figure 1 — figurative picture of the tridimensional virtual grid.

3.2. How to set Sonorium in a physical space : limitations and recommendations

After correctly installing all dependencies, as described in *chapter 2.3*, you can then open and execute the Sonorium application. To do so please carefully read the following limitations and recommendations beforehand:

- Connect the USB cable from Kinect to your computer before opening the application;
- Disconnecting this USB cable while Sonorium is running will cause the application to crash;
- Sonorium does not make a distinction between humans, objects or other animals, thus you should have a clear space without any obstructing objects (see *How to setup the capture area*);
- The capture area (see *chapter 4.2*) is simultaneously linked to the camera field of view and depth range (see http://msdn.microsoft.com/en-us/library/jj131033.aspx and http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges);
- The camera should be parallel to the floor and its height will correspond to the capture area's center point.

3.3. Out of the box (presets)

Sonorium brings four presets (or configurations) containing four soundscapes composed by Digitópia Collective members. In the top left corner of the application window you can select between these four presets (see Fig. 2). Note that starting up Sonorium will randomly load one of the available presets.

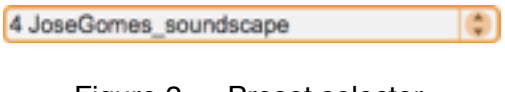


Figure 2 — Preset selector.

3.4. How to configure the sound output

In the *Configuration Panel* press the dark blue tab named *Output*. Here you can turn Sonorium's audio on or off, set the master volume (sampler only) and open the *Audio Settings* window (*'+'* button, see Fig. 3 and 4), where you can set your audio output device, sampling rate, etc.

Figure 3 - Audio output (on).

Figure 4 - Audio output (off).

3.5. Accessing help

Besides this manual you can access more information in the application *Help Window*, which can be opened in the *Presets section* of the application by pressing the *'?'* button. This will open a window presenting information whenever you hover your mouse over any graphical interface. Note that this information is presented in both english and portuguese in the *Status Bar* (small gray bar on the bottom of the application window).

There is also an online wiki available at https://github.com/Digitopia/Sonorium/wiki.

In case there is not enough information in the aforementioned places you can send an email to digitopia@casadamusica.com or p1nh0.c0d1ng@gmail.com. But please, check the help window and manual first.

4. ADVANCED USAGE

4.1. How to setup Kinect's capture

On the light blue tab (look for *Kinect Config.*) you can set the capture parameters: i.e. inverting the vertical or horizontal capture image using *flip vertically* and *flip horizontally* boxes, will let you place the Kinect in the best way.

4.2. How to set the capture area

The *Capture Area* sets the action space for Sonorium, making use of a virtual grid. This area is defined by four factors: a maximum and minimum depth, set in meters to the sensor; and the width and height of the capture area, both set in percentage. It is also possible to check the real dimensions of the capture area in the box (see *Real Dimensions*).

4.3. How to set the virtual grid

On the green tab *3D Virtual Grid* you can set the amount of hotspots you can use inside the *Capture Area*. You can set how many hotspots you will need for each axis *(x, y and z)* up to a total number of 4096 (16 hotspots per axis).
You can preview each *depth slice*, or each group of hotspots distributed in each part of the *z* axis using the *Z slice preview* menu.
It is also possible to set the spacing (also called *"dead area"*) between each hotspot using the numeric box *Spacing*. In every axis, spacing is done the same way, it's values are percentile and set from *'0.0'* to *'1.0'*, meaning that a spacing value of *'0.5'* determines that both the *dead area* and the hotspots have the same size.

After making changes to the virtual grid, the user should then hit the *Apply* button, in order for those changes to become active. You should note that this process might take a while, depending on your computer. (See *FAQ*)
Note that by hitting the *Apply* button changes won't be automatically stored into the current preset, so the user should hit the *Save* button in order to store current changes to the virtual grid. (Also see *chapter 4.5.4*)

4.4. Configuring the sampler

In the *Configuration Panel* select the *Output* tab and click *Configure Audio Samples* to open the sampler window. In this window you can drag audio files to each hotspot. Compatible audio file formats include '*.wav'* or '*.aiff'* only!
Hotspots are shown in different *Y slices*.
To check a specific hotspot at a specific slice (or height level) use the *Y Slice Selector* menu.
Furthermore it is possible to change parameters, such as *Loop*, *Fade In, Fade Out*, *Gain* and *Panning* in each hotspot, either individually or globally. (see *Sampler*)

4.5. Making preset banks / adding, saving and deleting presets

4.5.1. Making a preset bank

To create a group of presets (*Preset Folder* or preset bank) you will need to use the graphic interface in *Pro mode* and then press *Make New Preset Folder* on the *Preset Section*.

After pressing *Make New Preset Folder* a pop up window will open warning you that you should make a folder with the intended name for the group of presets. Close the *Warning Window* and a *Finder* dialog window will open. Then go to the desired directory, create a new folder (by pressing the *New Folder* button on the bottom left of the dialog window), set the desired folder name and press *Open*.

4.5.2. Opening a preset bank

To select a preset bank (or *Preset Folder*) press *Open Presets File* on the *Preset Section*. On the opened *Finder* dialog window navigate to the folder you want and open the *'.json'* file that has the same name as the folder.

4.5.3. Creating a new preset

To add a new preset press *'+'* on the *Preset Section*. The new preset will be copied from the previously selected one and saved on the current *Preset Folder*. The newly created preset will be added to the end of the preset bank (or folder).

4.5.4. Saving a preset

To save a preset press the *'S'* button on the *Preset Section*. All Sonorium parameters will be stored with the preset, except for the *Audio Settings* parameters.

4.5.5. Changing the preset name

To change a preset name just press the *orange circle* in the *Preset Selector* .
After pressing it, a window will pop up to introduce the new name.
The preset name will be automatically stored in the preset.

4.5.6.How to delete a preset

It is not possible to delete presets in version 1.2.1, thus the user has to manually delete and edit the preset files. (see *About Presets*)

4.6.Extending functionality (MIDI, OSC, grid and skeleton data)

Sonorium is not limited to the embedded sampler, since it allows to extend its own functionalities by communicating with other applications through MIDI or OSC *(Open Sound Control)* protocols. (see the following subchapters)

4.6.1. MIDI Output

Sonorium contains a dedicated window for MIDI output, very similar to the *Sampler* which can be accessed by pressing the *Output* tab and then the button *Configure MIDI notes*. This will open a dedicated window displaying all hotspots in a determined height slice *(y)*, where the user can define a MIDI note for each hotspot. To select different height slices click the *slice selector* on the top of the MIDI output window. (see *MIDI*)

Besides sharing the *loop* parameter with the sampler, which allows to repeat a MIDI note while the hotspot remains active, it is also possible to determine the loop duration (in milliseconds) through the *loop duration* number box. In addition, you can also set the note duration, which can be fixed or randomized, allowing you to define a minimum and maximum duration and a randomness amount (*duration variation*). In a similar manner the user can also set constant or randomized MIDI velocities for each note. Allowing it to have duration and velocity variations, which can lead to a slightly more expressive output. Just like the sampler, you can set these parameters to all MIDI notes, through the *header*, or you can have different values for each MIDI note.
Furthermore it is also possible to set the *MIDI channel* of each hotspot, allowing for example, to drive each hotspot note to different virtual instruments associated with the MIDI channel. While to set the *MIDI output device* the user should go to the *Output*, where it is also possible to update the device list, pressing the button *refresh device list*, if you connect your MIDI device after opening Sonorium.

4.6.2. OSC — Tridimensional grid

Besides the sampler and MIDI output Sonorium also allows to interact with other software or hardware devices compatible with OSC (Open Sound Control) by sending the information of the *3D grid* (basically, activity data of all hotspots) in real-time. This gives the possibility of extending Sonorium's core features by allowing, for example, to control a video projection.

The *3D grid* is sent over the network through UDP (http://en.wikipedia.org/wiki/User_Datagram_Protocol). Note that while this protocol is faster than TCP (http://en.wikipedia.org/wiki/Transmission_Control_Protocol) it is susceptible to data loss.

To send the *3D grid* data through the network it is necessary to set the *IP address* and the desired *port* number of the device in the *Output* tab and then you just have to activate the *grid* switch within this tab (see Figure 5).

Figure 5 - *Open Sound Control output.*

To receive the *3D grid* data just connect a device to the same *IP address* and *port* number set on the *Output* tab, as shown in the example of Figure 6 using a MaxMSP patch.

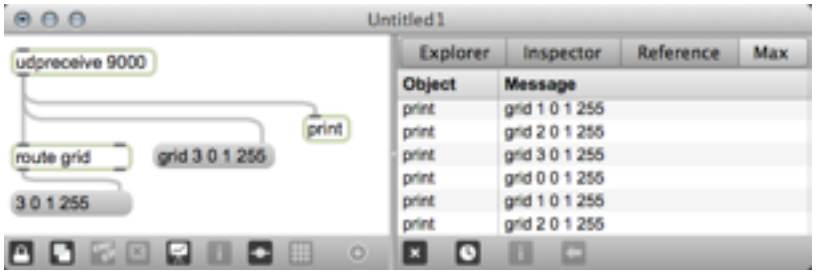


Figure 6 - Receiving 3D grid data through network (example).

To better explain the *3D grid* messages sent through OSC, we will reference its format, range and trigger, providing a concrete example at the end.

Format:
List with four integer numbers: *< x >< y >< z >< activity >*

Range:
x = { 0, $x_{hotspots}$ - 1 }
y = { 0, $y_{hotspots}$ - 1 }
z = { 0, $z_{hotspots}$ -1 }
activity = 0 or 255

Output trigger:
The `3D grid` list is only sent when a change of activity values occur in a depth matrix `(z)`. Thus sending lists in the format *x/y/z/activity* serially, corresponding to the depth slice.
The number of lists sent for each depth slice depends on the number of hotspots in the `X` and `Y` axis, represented by the following formula:

number of lists for each depth slice = $x_{hotspots} * y_{hotspots}$

If, for example, changes in activity values happen in two depth slices simultaneously, Sonorium will send the list of the closest depth slice (smaller `Z` value) first.

Example:
Given the values,
$x_{hotspots}$ = 4;
$y_{hotspots}$ = 3;
$z_{hotspots}$ = 2.

If there is a change in the first depth slice (the closest to the Kinect camera, with `Z` = 0), the 3D grid output would be the following:

4.6.3.OSC — Skeleton data

Besides sending the `3D grid` through UDP it is also possible to send user skeleton tracking data. This functionality is inherent to the *jit.openni* external object which is also used for the Kinect depth image capture. This object gives us the total number of users within Kinect's sight, their bodies central position as well as the position of several articulations such as knees, hands, elbows, feet, etc. For more information on this topic refer to https://github.com/diablodale/jit.openni/wiki

4.6.4. Remote control

This functionality allows to control Sonorium remotely through a network connection. Making it possible, for example, to control the software in a performance situation, by having a computer on stage running Sonorium and another, on the *régie* (for ex.), to control the computer on stage.

(X Y Z Activity)	*Description*
0 0 0 0	*—> x = 0, y = 0, z = 0, activity = 0 (off)*
1 0 0 0	*—> x = 1, ...*
2 0 0 0	*—> x = 2, ...*
3 0 0 0	—> x = 3, ...
0 1 0 0	*—> x = 0, y = 1, ...*
1 1 0 0	*—> x = 1, ...*
2 1 0 255	*—> x = 2, y = 1, z = 0, activity = 255 (on)*
3 1 0 0	*—> x = 3, ...*
0 2 0 0	*—> x = 0, y = 2, z = 0*
1 2 0 0	*—> x = 1, ...*
2 2 0 0	*—> x = 2, ...*
3 2 0 0	*—> x = 3, ...*

The remote control is done through the *TCP protocol*, so that no control messages are lost during communication over the network.

In order to use the remote control you will need to activate the *debug mode* and *system messages* on the slave computer. Then, still on the slave, configure the *IP address* and *port number* corresponding to the master computer. (see Fig. 7)

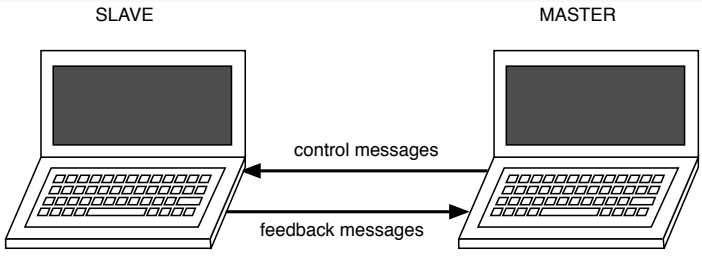


Figure 7 — Remote control.

On the master computer you should now be receiving feedback messages from the slave computer whenever you change any interface value.

To send control messages back to the slave computer you should send them to the slave IP address (see *Network Preferences* on you computer) and to the *remote control port* set on the slave computer.

Bare in mind that control messages are sent to pattr objects (see Fig 8). There is an example patch distributed with the source-code in the folder *'~/patchers/utilities/*

Remote Control'. If you're not aware of how this works in MaxMSP you should at least read the following:
https://docs.cycling74.com/max6/dynamic/c74_docs.html#pattrforward
https://docs.cycling74.com/max6/dynamic/c74_docs.html#pattrstorage
https://docs.cycling74.com/max6/dynamic/c74_docs.html#pattrchapter01
https://docs.cycling74.com/max6/dynamic/c74_docs.html#pattrchapter02

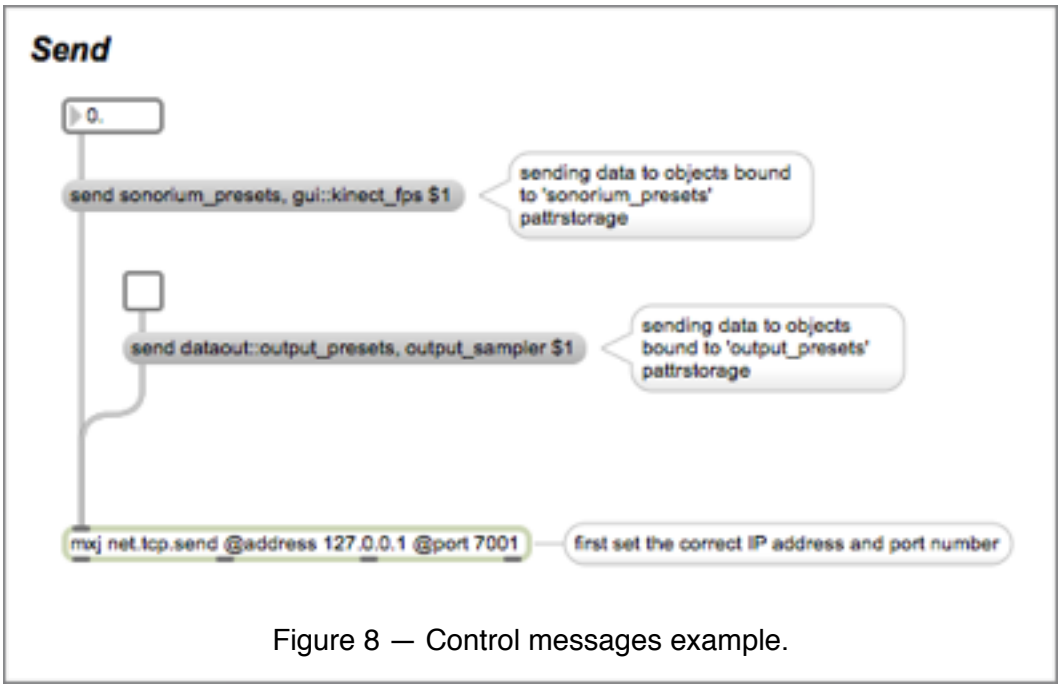
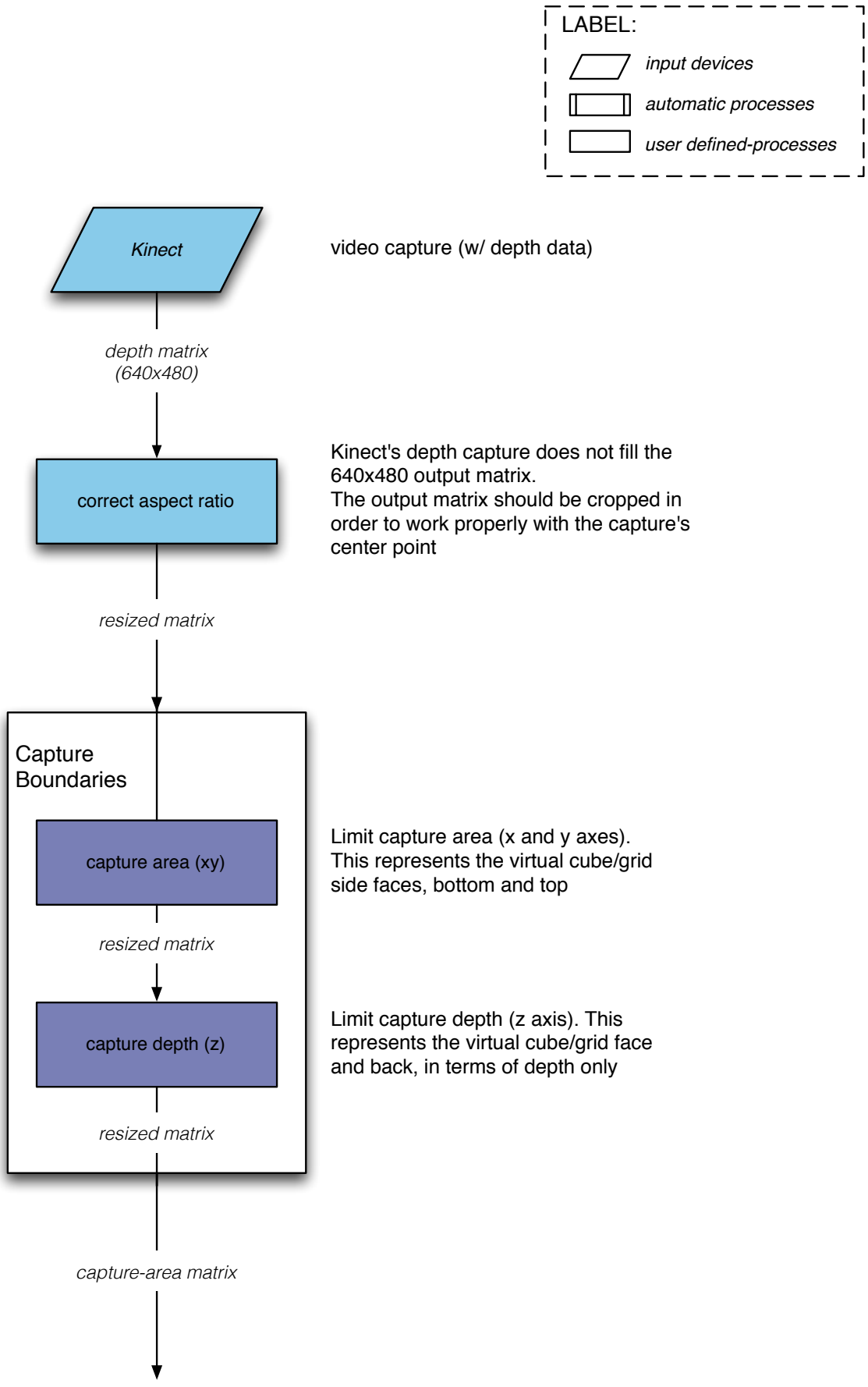


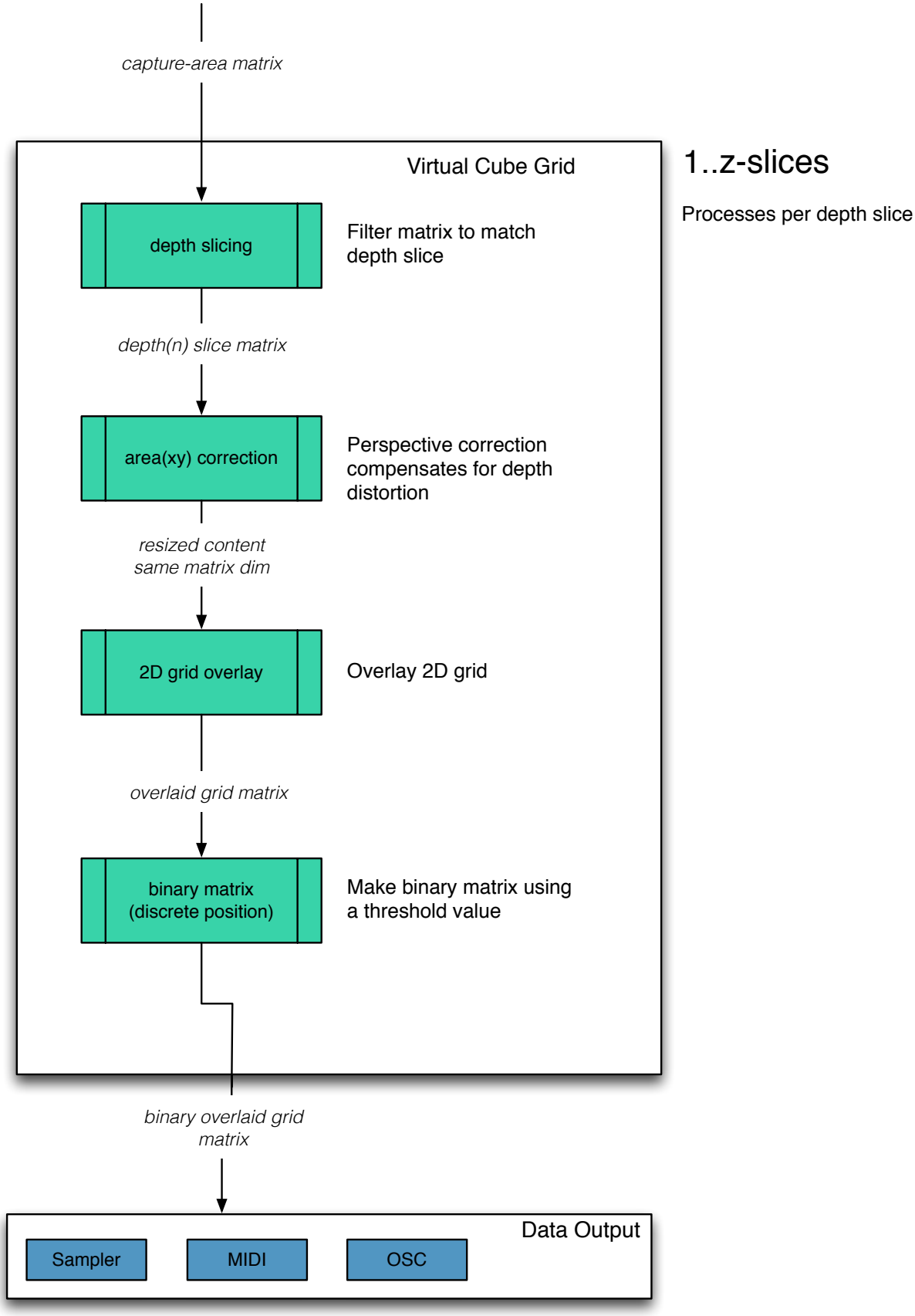
Figure 8 — Control messages example.

4.7.Software architecture

4.7.1. *Video Input and Capture Area*



4.7.2. *Virtual Cube Grid and Output*



4.8. About the perspective correction

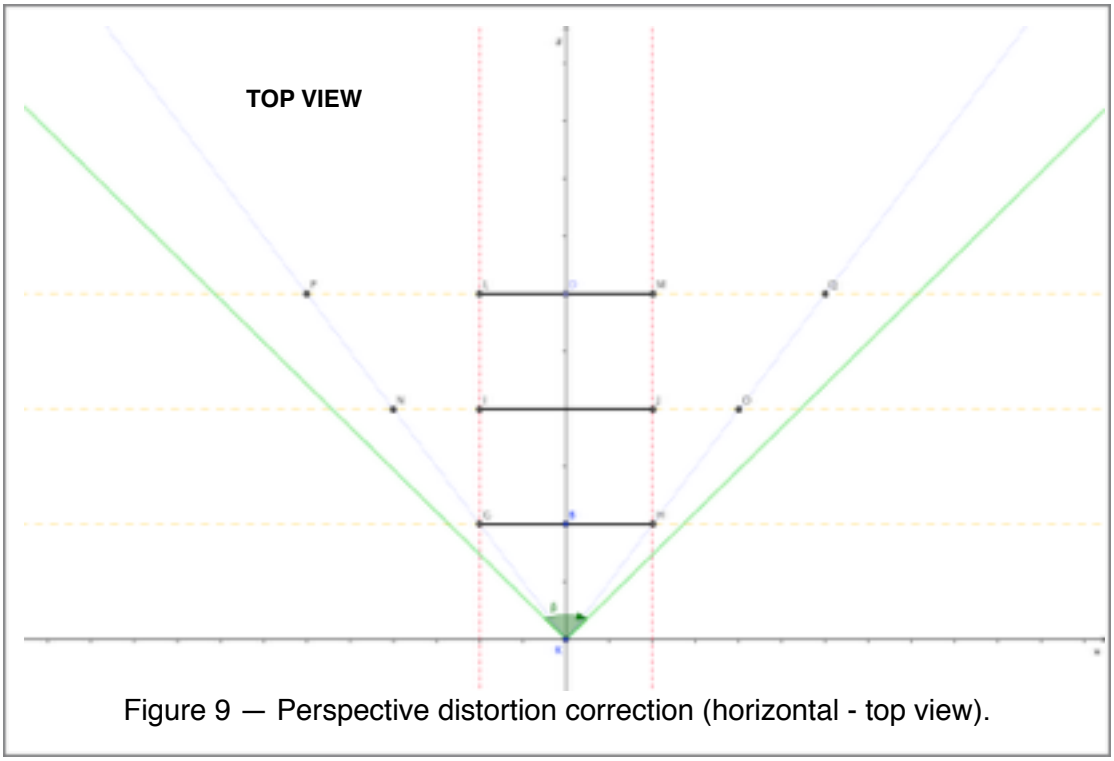


Figure 9 — Perspective distortion correction (horizontal - top view).

- K - central position of the Kinect camera;
- ß (green) - camera's *horizontal field of view* angle;
- Segment $\overline{KB}$ represents *minimum depth*;
- Segment $\overline{KO}$ represents *maximum depth*;
- The area defined by points G, H, L and M represent the *capture area*;
- The area defined by points G, H, I and J represents the second depth (z) layer;
- Segment $\overline{NO}$ represents the image captures at the beginning of the second depth (z) layer;
- Segment $\overline{IJ}$ represents a perspective correction of the image at the beginning of the second depth (z) layer. Note that $\overline{IJ}$ has the same pixel content as $\overline{NO}$ but is resized to correct the perspective distortion.
- The blue line represents the angle formed when the capture area is adjusted (see *Capture Area Correction*).

This process is also applied to the vertical perspective distortion, where the Kinect's field of view has an angle of 43 degrees.

Note that from the beginning of a depth slice to the next there is still a small distortion caused by the camera's field of view. Although this was not possible to correct with the current algorithm, it has a minimum effect on the functioning and concept of Sonorium's *3D virtual grid.*

4.9. About presets

Sonorium uses a preset bank to save several software configurations. This preset bank is stored in a folder, containing a *'.json'* file with the same name as the folder. This file contains information about all presets in the folder. Then, for each preset number there is a *'.json'* file and a *'.jxf'* file, where the nomenclature of these files starts with two digits corresponding to the preset number.

To better understand how presets are stored please see the *'sonorium_presets'* folder contained in the Sonorium application folder:

The folder *'sonorium_presets'* contains a file named *'sonorium_presets.json'* which contains information about all presets in this folder;
this folder contains four presets, where each preset is represented by two files, one with a *'.json'* extension and another with a *'.jxf'* extension. These preset files start with a two digit number and an underscore *("_")* and then have the same name as the preset folder. For example: *'01_sonorium_presets.json'* and *'01_sonorium_presets.jxf'* correspond to the first preset in the *'sonorium_presets'* preset bank.

5. LABEL

Sonorium has an embedded label of all visible interfaces, which is displayed in the status bar (gray bar at the bottom of the application window) or in the help with (see *Help button*)

5.1. Sections

Sonorium has three distinct sections:

- *Presets* : in this section the user may create, store and load presets;
- Configuration panel : in this section the user can set a determined software configuration;
- Preview : this section is used to provide visual feedback of the current configuration.

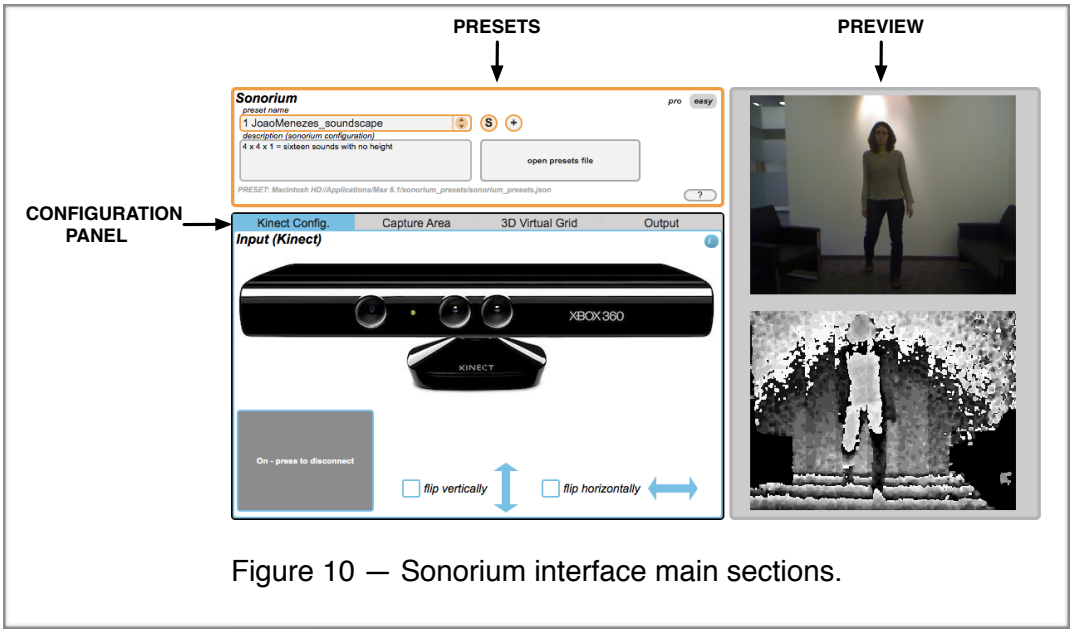


Figure 10 — Sonorium interface main sections.

5.2. Presets

Sonorium is distributed with a couple presets developed by Digitópia team and will, by default, load a random preset when opening the application.

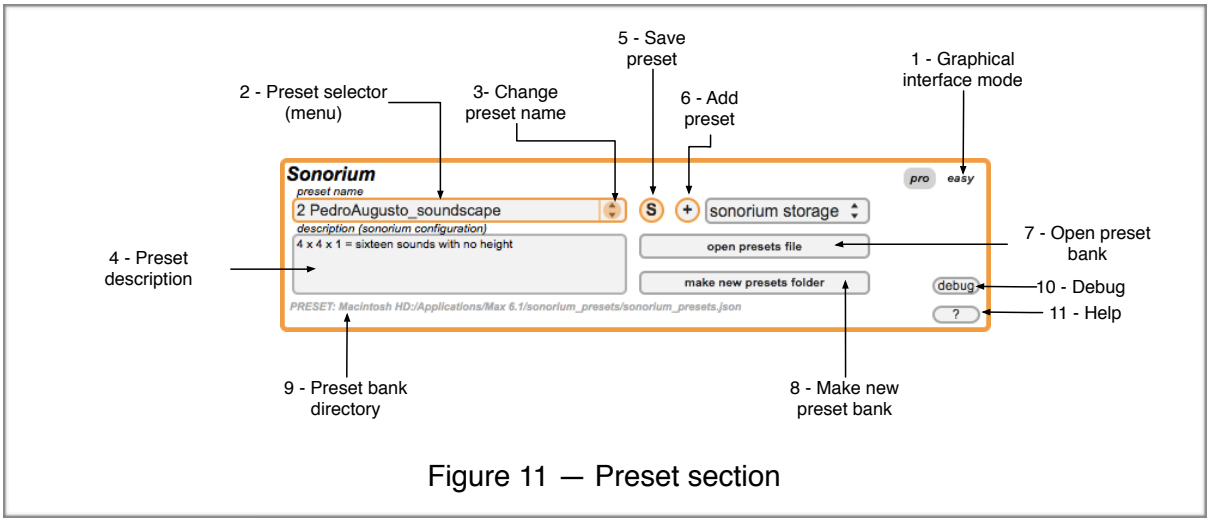


Figure 11 — Preset section

1. **Graphical interface mode (tab)**. Allows to switch between two different modes of presentation: easy (only displays basic interfaces) and pro (displays all interfaces);
2. **Preset selector (menu).** Allows to select a preset from a preset bank (or folder). See *`About Presets`*;
3. **Change preset name (button).** Pressing the small orange circle will open a dialog window to change the currently selected preset name;
4. **Preset description (text box).** View or edit the description about Sonorium's configuration stored in the currently selected preset. If you edit the this text don't forget to press the save button to store your description;
5. **Save preset (button).** Save all parameters to the currently selected preset file;
6. **Add preset (button "+").** Adds a new preset to the preset bank by copying the currently selected preset;
7. **Open preset bank (button).** Opens a *Finder* navigation window allowing you to load a preset bank (or folder). (see *`Opening a Preset Bank`*);
8. **Make new presets bank (button).** Create a new preset bank (or folder) (see *`Making a Preset Bank`*);
9. **reset bank directory (text).** Displays the directory of the currently selected preset bank;
10. **Debug (button).** By activating the debug mode all parameter changes (related to MaxMSP *'pattr'* objects) will be displayed in the console (also known as *Max Window*);
11. **Help (button).** Opens the clue window which displays information about the objects hovered by the mouse.

5.3. Configuration panel

To set a new sound interaction space follow every configuration step (from left to right). This panel shows 4 sections:

- **Kinect Config.:** In this section you can set the parameters for the video capture made by Kinect;
- **Capture Area:** In this section you can set the parameters for the desired capture area;
- **3D Virtual Grid:** In this section you can set the parameters of the virtual grid inside the Capture Area;
- **Output:** In this section you can set the behavior of each hotspot, by triggering samples, sending MIDI notes or OSC messages.

5.4. Kinect Configuration

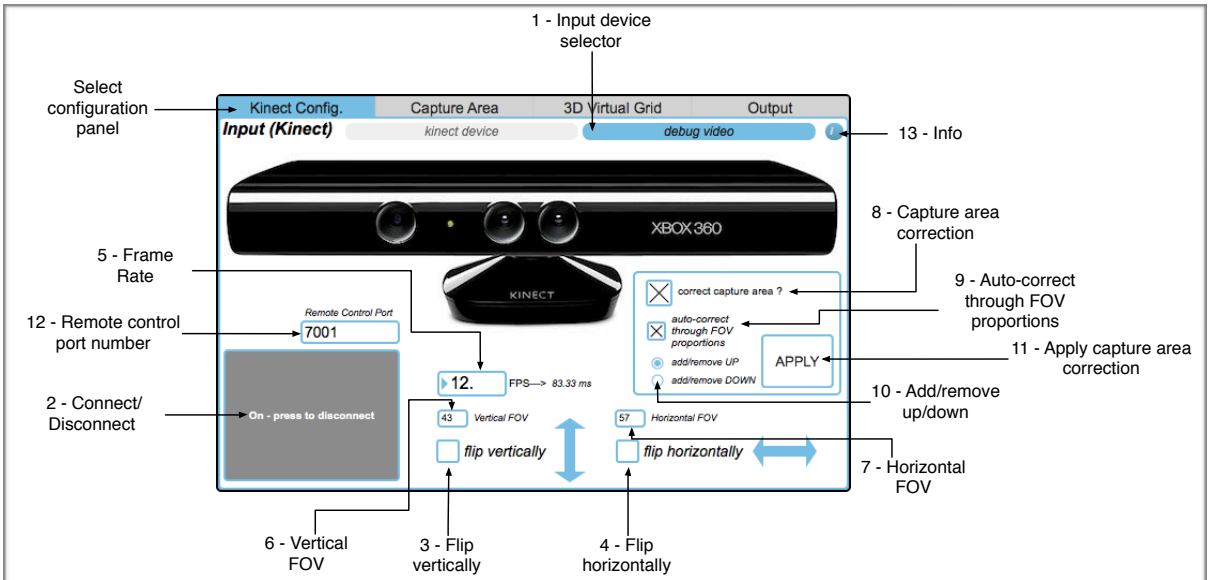


Figure 12 — *Kinect Config*. section.

1. **Input device selector (tab).** By default Sonorium will try to read from a connected Kinect camera. But if you don't have a Kinect camera you can use a recorded test video to experiment Sonorium. If you use a the debug video you should set the *`frame rate`* to 60 frames per second;
2. **Connect/disconnect (switch).** Start or stop video capture;
3. **Flip vertically (switch).** Vertically flip the video capture;
4. **Flip horizontally (switch).** Horizontally flip the video capture;
5. **Frame rate (number box).** Set the frame rate of the video capture. By default Sonorium uses a low frame rate of 12 fps (frames per second), which allows older computers to function better without compromising Sonorium's behavior;
6. **Vertical FOV (number box).** Set this number according to your camera's vertical field of view (FOV). With Microsoft Kinect cameras this value is usually 43 degrees;
7. **Horizontal FOV (number box).** Set this number according to your camera's horizontal field of view (FOV). In Microsoft Kinect cameras this value is usually 57 degrees;

8. **Capture area correction (switch).** Allows to correct the capture area by removing “dead”/unused pixels left by Kinect’s depth capture, thus maximizing Sonorium’s performance (since these “dead” pixels would be processed later);
9. **Auto-correct through FOV proportions (switch).** Uses the field of view angles to better correct the capture area;
10. **Add/remove up/down (tab).** Choose between removing the excess of pixels caused by the capture correction to be remove from the upper or the lower part of the video capture;
11. **Apply capture area correction (button).** Apply capture area correction (see list items 14 to 18);
12. **Remote control port (number box).** Set the port used for remote control (see *`Remote Control`*);
13. **Info (button).** Opens a web browser window directing to Microsoft’s Kinect specifications;
14. **Left (number box).** See list item number 17;
15. **Right (number box).** See list item number 17;
16. **Top (number box).** See list item number 17;
17. **Bottom (number box).** These number boxes allow to set the capture area correction, by manually identifying the “dead” area (black pixels) caused by the Kinect depth image. Note that Sonorium will try to maintain the same aspect ratio as the original image coming from the Kinect camera;
18. **Outcome (matrix).** Preview of the capture area correction after hitting the *`Apply`* button.

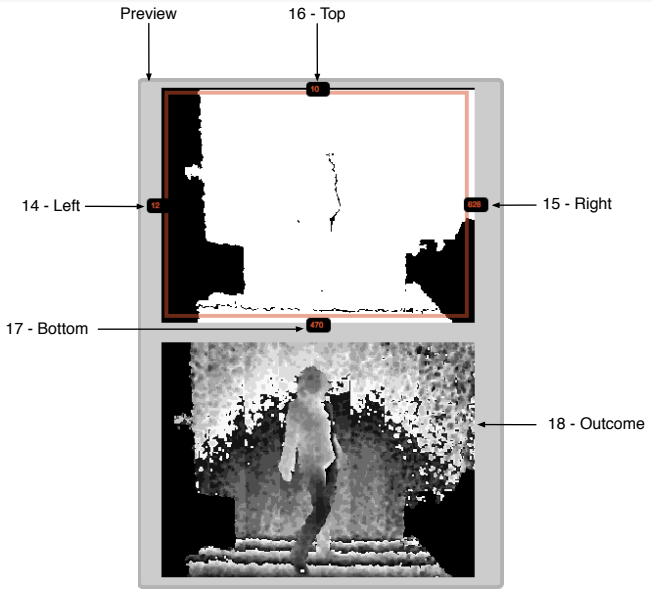


Figure 13 — Preview section

5.5. Capture Area

In this section it is possible to set the capture area where Sonorium will fit the hotspots for the 3D virtual grid.

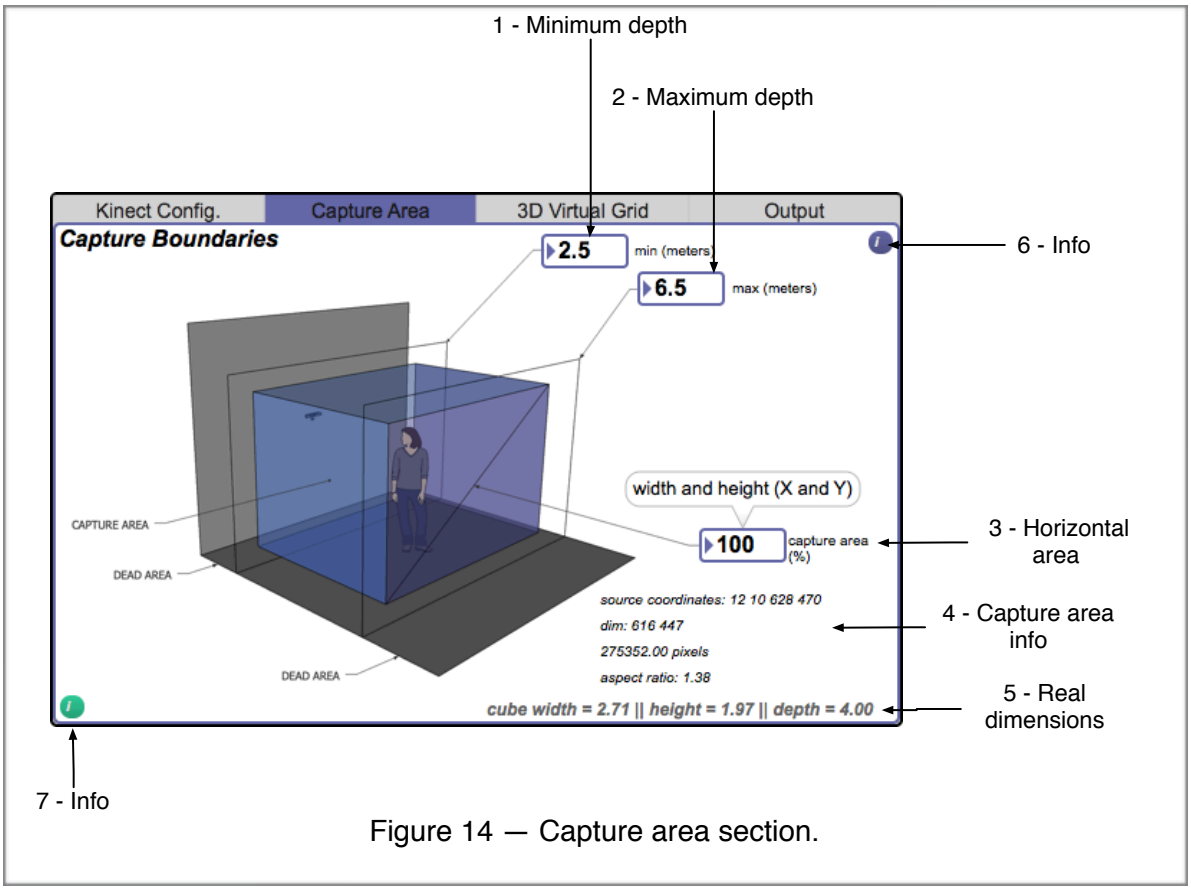


Figure 14 — Capture area section.

1. **Minimum Depth (number box).** Minimum depth measured from the Kinect in meters;
2. **Maximum depth (number box).** Maximum depth measured from the Kinect in meters;
3. **Horizontal area (number box).** Height and width of the capture area, defined as a percentage of the possible capture area;
4. **Capture area info (text).** Displays the coordinates of the captured matrix, its dimension in pixels, the total number of pixels and its aspect ratio;
5. **Real Dimensions (text).** Real dimensions (width, height and depth) of the capture area measured in meters;
6. **Info (purple button).** Opens a web browser pointing to Microsoft's web site containing information about the Kinect's depth capture specifications;
7. **Info (green button).** Opens a window explaining the orientation of the three axis x, y and z.

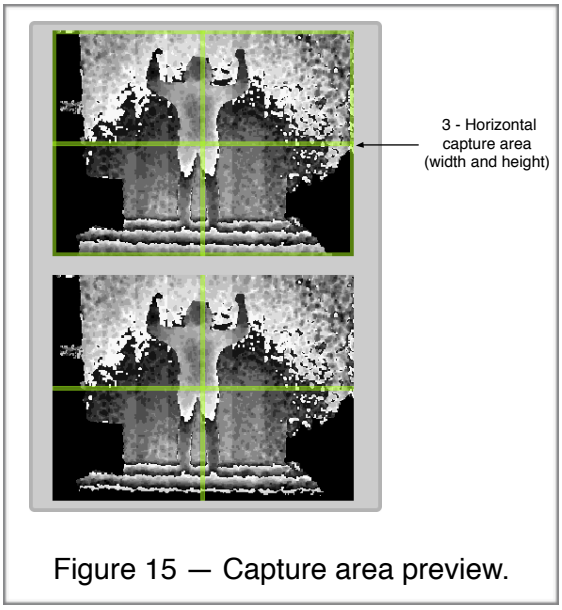


Figure 15 — Capture area preview.

5.6. 3D Virtual Grid

In this section you can set the number of hotspots contained in the capture area, thus forming a tridimensional virtual grid.

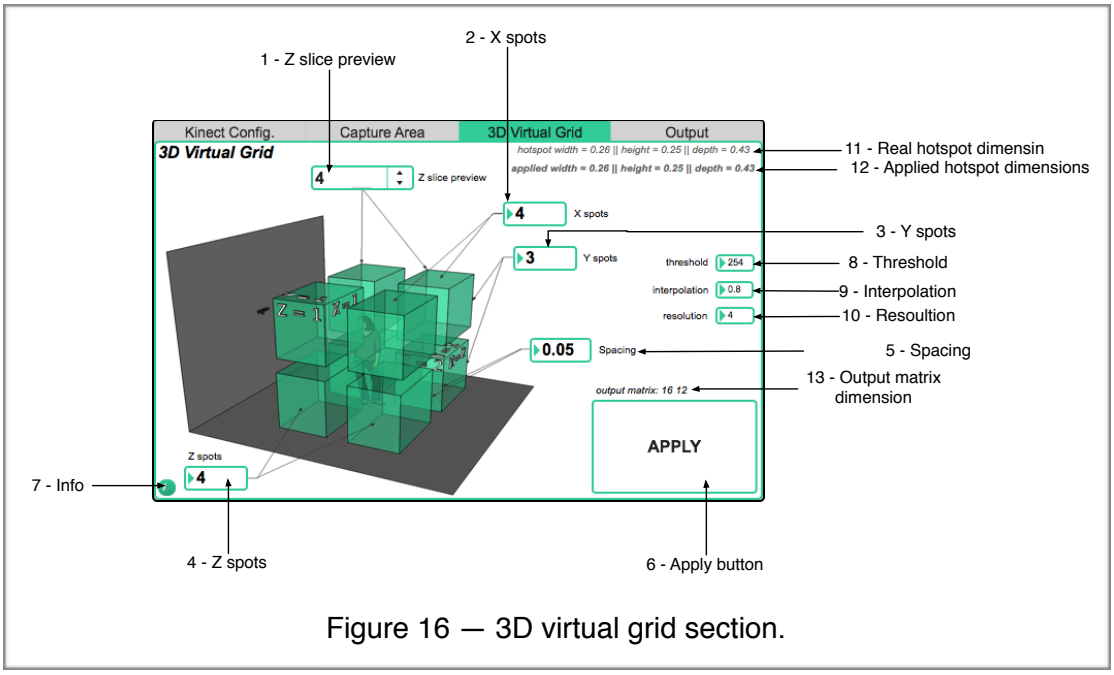


Figure 16 — 3D virtual grid section.

1. **Z Slice Preview (menu).** With this menu you can select which depth slice you would like to preview (on the preview matrices at the right of the section window). By selecting overall it is possible to have a basic preview of all depth slices;
2. **X spots (number box).** Set the number of hotspots placed on the x axis (width);
3. **Y spots (number box).** Set the number of hotspots placed on the y axis (height);
4. **Z spots (number box).** Set the number of hotspots placed on the z axis (depth);
5. **Spacing (number box).** Set the space between hotspots by percentage, from 0 to 1 (= 100%). A spacing value of 50% (0.5) means that the size of the empty space is the same as the size of a hotspot;
6. **Apply (button).** In order to confirm changes made to the 3D virtual grid it is necessary to press the apply button. Note that this button changes its color to orange to inform you that changes were not applied yet;
7. **Info (button).** Opens a window explaining the orientation of the three axis x, y and z;
8. **Threshold (number box).** Set the threshold value to trigger a hotspot on or off. This value ranges from 0 to 254. Note that threshold is placed after `interpolation`;
9. **Interpolation (number box).** Used to smooth hotspot activation by adding a video feedback stage. A bigger interpolation value will make hotspots active for a long period of time after being deactivated. This should be used in conjunction with the *threshold* value and the resolution (described below) to obtain an optimal hotspot activation behavior;
10. **Resolution (number box).** Set the hotspot activation resolution. The bigger the resolution the bigger the area need to activate the hotspot. For example, a resolution value of 4 means that at least a quarter of the hotspot needs to be filled in order to be activated. Note that higher resolution values will require more processing power.
11. **Real hotspot dimensions (text).** Preview hotspot dimensions in meters.

12. **Applied hotspot dimensions (texts).** Preview the applied hotspot dimensions in meters;
13. **Output matrix dimension (text).** Displays the output matrix size in pixels. This value changes according to the selected *resolution* (see list element number 10);
14. **Depth preview (matrix).** Displays all *z slices* affected by the virtual grid;
15. **Z slice grid preview (matrix).** Displays selected the grid result of a selected *z slice*.

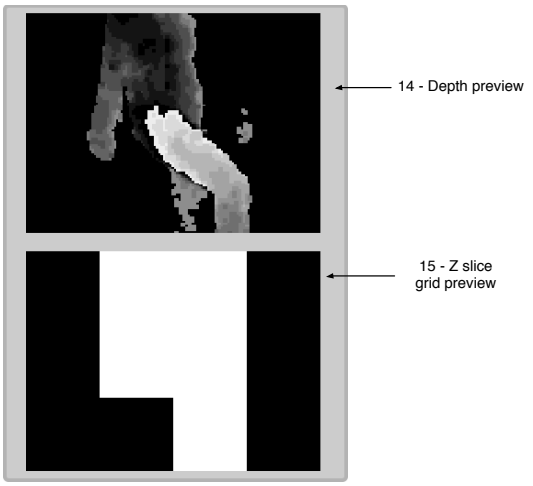


Figure 17 — Grid preview.

5.7. Output

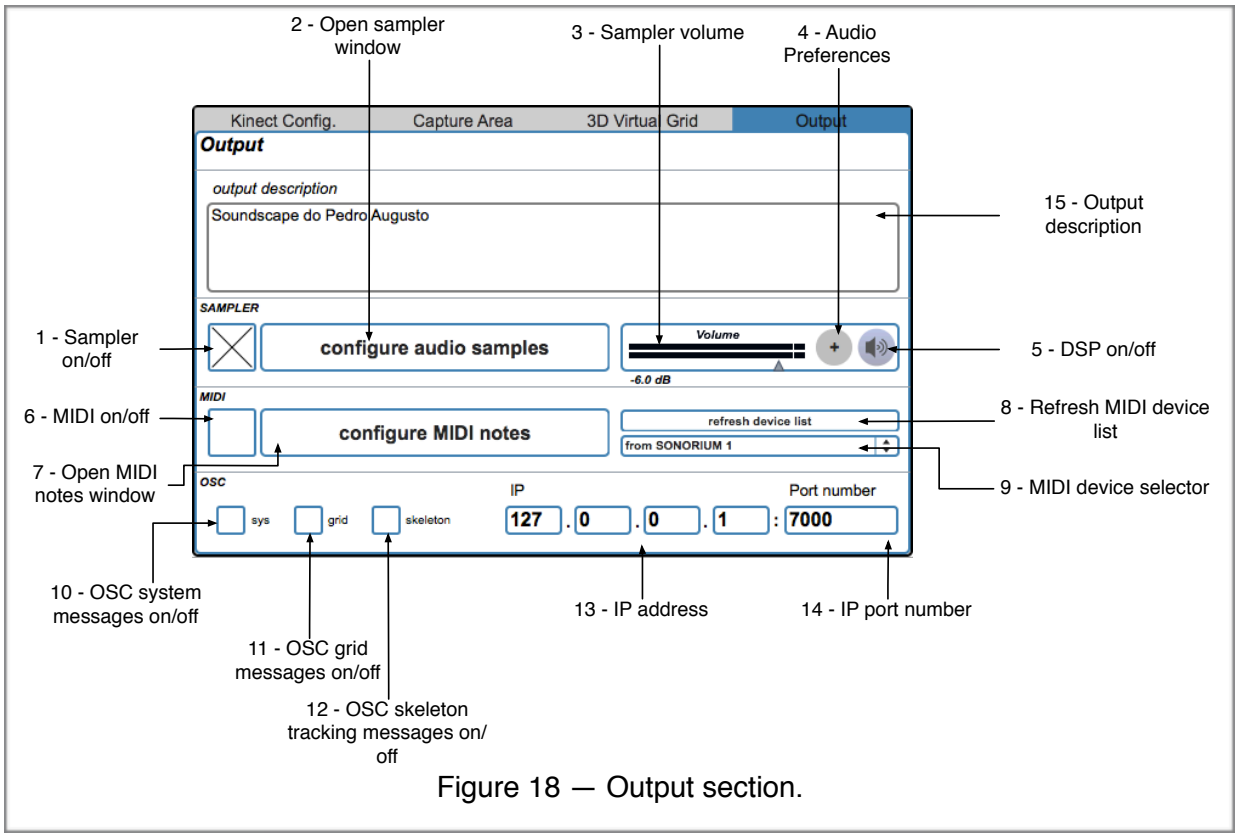


Figure 18 — Output section.

1. **Sampler on/off (switch).** Activate or deactivate the sampler. When deactivating the sampler all playing samples will be muted;
2. **Configure audio samples (button).** Clicking this button will open the sampler window (see *Sampler*);

3. **Sampler volume (fader+meter).** Sets the sampler output volume. The interface has a volume meter display and allows the user to set the volume in decibels via keyboard. Pressing the backspace key will set the volume to 0 dB;
4. **Audio preferences (button).** Opens the audio preferences window which allows to set the output device, sampling rate, etc. Note that Sonorium won't automatically adapt to the samples' sampling rate, so you should manually set the sampling rate and use audio samples with the same rate;
5. **DSP on/off (switch).** Turn audio processing on or off. By default DSP is on when Sonorium is launched. If the audio sampler is not in use the user should turn DSP off. Note that the DSP state is not stored in presets;
6. **MIDI on/off (switch).** Activate or deactivate MIDI output;
7. **Configure MIDI notes (button).** Opens *MIDI notes* configuration window allowing you to set different MIDI notes into different hotspots (see *`MIDI`*);
8. **Refresh device list (button).** Press this button to update the MIDI output device list;
9. **MIDI device selector (menu).** Select an available MIDI output device;
10. **OSC sys on/off (switch).** Activate or deactivate the output of system messages through network. These messages represent all parameter changes made in Sonorium (see *`Remote Control`*);
11. **OSC grid on/off (switch).** Activate or deactivate the output of grid messages through network (see *`OSC – Tridimensional Grid`*);
12. **OSC skeleton on/off (switch).** Activate or deactivate the output of user and skeleton tracking data (see *`OSC – Skeleton Data`*);
13. **IP (4 number boxes).** Set the IP (Internet Protocol) address to send network messages to. Note that most network messages are sent through UDP (User Datagram Protocol) which can sometimes face data loss (see *`OSC Tridimensional Grid`*);
14. **Port (number box).** Port number used for network communication;
15. **Output description (text editor).** Used to set and display output description;

5.8. Sampler

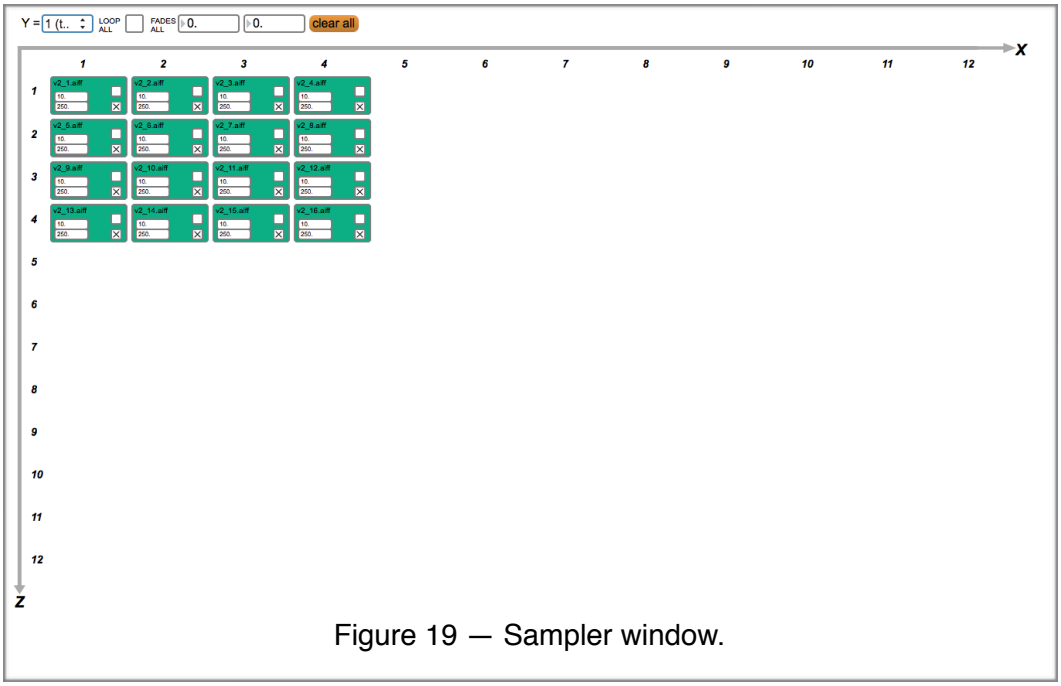


Figure 19 — Sampler window.

As shown in Figure 19, Sonorium's sampler window will only show each y axis "slice" at a time, showing just the samples placed in hotspots from x and z axis. To view the entire grid of samples the user must use the slice selector (see point 1 of the list above).

Sonorium's sampler allows the user to load a maximum number of 4096* audio samples, depending on the size of all samples and the available RAM on the computer (see *Prerequisites*).

*- 16 x 16 x 16 = 4096, which accounts for 16 hotspots per axis x, y and z.

Note that making changes in the sample header (see Fig. 20) will affect all hotspots while making changes in each hotspot interface will only affect that hotspot.

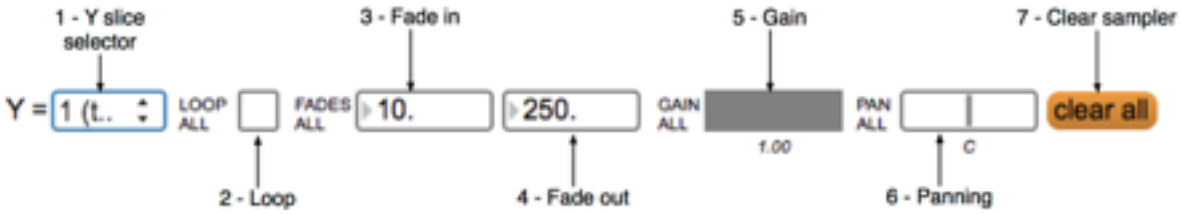


Figure 20 — Sampler header.

1. **Y slice selector (menu).** Use this menu to navigate through all hotspots. Note that a *y* value of 1 corresponds to the top slice;
2. **Loop (switch).** Loop playback of audio sample while its hotspot is active. Loop time is given in milliseconds;
3. **Fade in (number box).** Sample fade-in time in milliseconds;
4. **Fade out (number box).** Sample fade-out in milliseconds;
5. **Gain (slider).** Sets the sampler volume;
6. **Panning (slider).** Sets left-to-right panning with constant power.Double-clicking this interface will set the panning to the default center position;

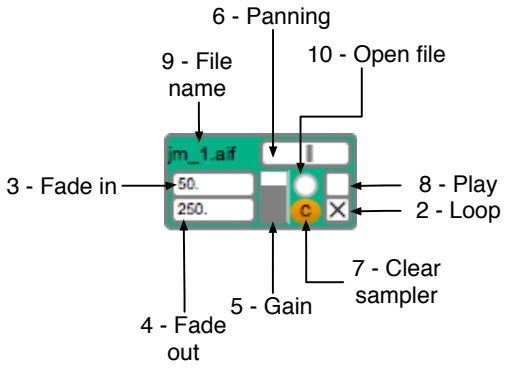


Figure 21 — Hotspot sampler with a preloaded audio file.

7. **Clear (button).** Clears audio samples leaving the hotspot empty. Bare in mind that it is not possible to undo this action. Although you can recall a stored preset, unsaved changes made to the hotspots' samples will not be restored after pressing the clear button;
8. **Play button (switch).** This button becomes active whenever a hotspot is triggered, also triggering the sound sample. You can also trigger this button manually to test the sound output;
9. **File name (text).** Displays the sample file name and extension;
10. **Open file (button).** Alternative method for loading samples. Press the button to open a dialog window and select a file.

Figure 22 — Empty hotspot sampler (without any audio file loaded).

5.9. MIDI

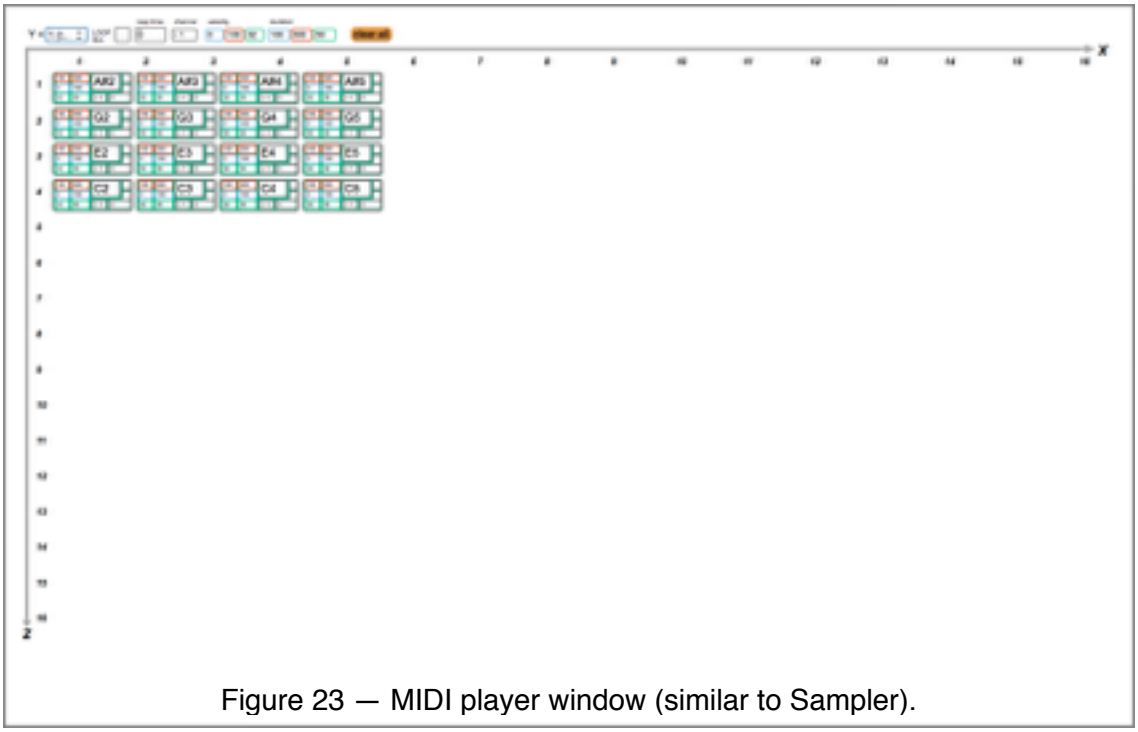


Figure 23 — MIDI player window (similar to Sampler).

The MIDI output window is very similar to the *Sampler* window, having only a few more features listed above. And obviously, instead of triggering audio samples, it outputs MIDI note messages to a selected output device (such as DAW's, synths or others), where each hotspot contains a note generator with varying velocity and duration parameters as described above.

Note that making changes in the MIDI player header (see Fig. 24) will affect all hotspots while making changes in each hotspot interface will only affect that hotspot.

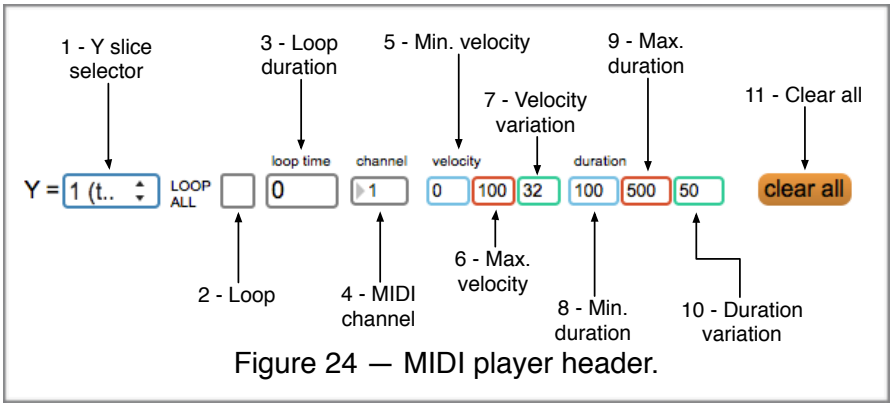


Figure 24 — MIDI player header.

1. **Y Slice selector (menu).** Use this menu to navigate through all hotspots. Note that a y value of 1 corresponds to the top slice;
2. **Loop (switch).** Loop playback of the MIDI note while the hotspot is active;
3. **Loop duration (number box).** Loop duration in milliseconds;

4. **MIDI channel (number box).** Set MIDI output channel (1..16);
5. **Minimum velocity (number box).** Set the minimum velocity value (0..127) of the hotspot's note generator;
6. **Maximum velocity (number box).** Set the maximum velocity value (0..127) of the hotspot's note generator;
7. **Velocity variation (number box).** Amount of velocity variation applied to the hotspot's note generator. (NOTE that setting this value to '0' the note velocity will be equal to zero! This problem should be fixed in later versions);
8. **Minimum duration (number box).** Set minimum note duration in milliseconds;
9. **Maximum duration (number box).** Set maximum note duration in milliseconds;
10. **Duration variation (number box).** Amount of variation applied to the generated note's duration;
11. **Clear all (button).** Clears all MIDI notes leaving all hotspots empty. Bare in mind that is not possible to undo this action. Although you can recall a stored preset, unsaved changes made to the hotspots' notes will not be restored after pressing the *clear all* button;
12. **Play button (trigger).** This button becomes active whenever a hotspot is triggered, also triggering the note generator. You can also trigger this button manually to test the note generator and/or MIDI device receiver;
13. **MIDI note (note box).** Select hotspot note (middle C = C3). It is possible to input note numbers or text notes (ex.: C#1) via the computer keyboard. When the hotspot is white it means that notes won't be generated. To clear a single hotspot click and hold the mouse and scroll down.

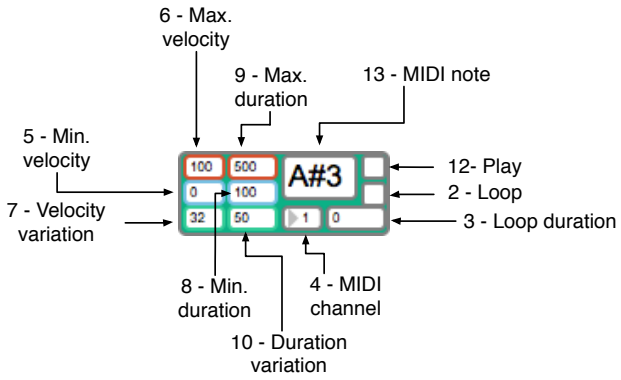


Figure 25 — hotspot MIDI note (with A#3 activated).

Figure 26 — empty hotspot MIDI note (no active note).

6. Frequently asked questions (FAQ)

- **The application is running slowly. How can I improve its performance ?**

  There are several tweaks that can improve Sonorium's performance, such as:
  - decrease *`Frame Rate`*;
  - decrease *`Resolution;`*
  - decrease the number of *`hotspots`*;
  - if you are not using the sampler make sure there are no samples placed on any hotspot (or just press *`Clear All`*) and also turn the *`DSP`* off.

- **My depth-sensing camera has different specifications. How can I adapt Sonorium to my device ?**

  In that case you should check the field of view (FOV) of your device and set Sonorium's FOV to match your device. You can do this by switching the *`Graphical Interface Mode`* to "pro", clicking the *`Kinect Config`* tab and setting the vertical and horizontal FOV values to match the FOV values of your camera.

- **When I press the *`Apply`* button in the *`3D Virtual Grid`* tab the application seems to jam. What can I do?**

  The process of generating the *`Virtual Grid`* can take a while, especially if there are many hotspots and Sonorium is running in a fairly old machine. In that you case you should wait as long as you can for the grid to be drawn. After drawing the *`Virtual Grid`* can be stored within the presets so you can later recall a preset that loads the pre-drawn grid instantly.

- **I want to use an external sound output device, how do I set it up ?**

  Click the *Output* tab and press the *`Audio Preferences`* button (*`+`*). This will open a window where you can select your output device from a drop-down menu. For more information about all the features in this window please visit the link : https://cycling74.com/docs/max5/vignettes/core/dsp_status.html

- **Sonorium crashes when I disconnect the USB cable from my computer or when camera power is turned of.**

  You should always quit Sonorium first and then disconnect your camera, otherwise the application will crash.

- **I have moved the preset folder or audio files to another directory or another computer and I can't get them to work. What can I do ?**

  It is possible to move files or presets to another computer as long as they stay in the same directory, for example: if you had your files on the *Desktop*, when moving them to another computer you should also place them in the *Desktop.* In case you do need to move them to another directory you should recreate

your presets, just like you made them the first time (see *chapter 4.5*), or you could edit the preset files with a text editor (ex: TextEdit).

If you have decided to edit the preset files you will need to:

1) edit matrix paths in the main preset file. For example,in the file *'sonorium_presets.json', edit all matrix paths to match the new path, like so:*

```
"2Dmatrix_path" : [ "/Applications/Sonorium-1.0/
sonorium_presets/01_sonorium_presets.jxf" ]
```

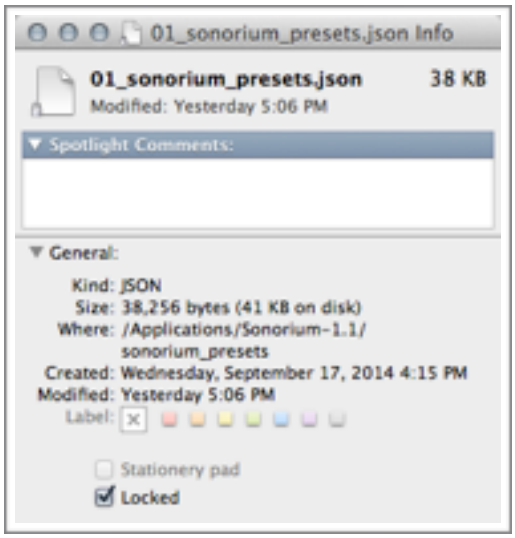
2) edit all sound file paths in all other presets (ex: *01_sonorium_presets.json*) that make use of the *Sampler* in order to match the new directory, like so:

```
"sampler::container[1]::s[1]::path" : [ "/Applications/
Sonorium-1.0/audio_files/JoaoM/jm_1.aif" ]
```

- **I accidentally changed my preset files. What can I do to prevent that from happening again ?**

In that case you should lock all preset files, that way Sonorium will not store any that on protected files.

To lock your files select them in a *Finder* window and open the *Information* window and check the 'Locked' toggle as shown in the figure on the right.



- **I want to compile the application after making some changes to the source code. How can I do that ?**

You should read the file *'build_instructions.txt'* under the folder *'~/misc/'* on the contents of the source code. To download the source code visit our github page at https://github.com/Digitopia/Sonorium

- **I want to compile the application for the Windows operative system. Is that possible ?**

If you want to compile Sonorium for Windows you should replace the *jit.openni* object by *dp.kinect* or any other MaxMSP object that can use the Kinect on Windows. The object does not need to have any skeleton tracking properties, a simple object that can provide the depth image from Kinect will be sufficient.

7. ACKNOWLEDGEMENTS

- Education Service of Casa da Música
- Dale Phurrough (for *jit.openni,* the MaxMSP external)

8. LICENSE