SPIRIGATE

User Manual

HUMMELING engineering

January 4, 2011

Contents

1 Background **2**

2 Installation **3**

2.1 Microsoft Windows 3

2.2 GNU/Linux 3

2.3 Mac OS X 3

3 Getting Started **3**

3.1 Volume Mixer Panel 3

3.2 Dithering Panel 4

3.3 Tone Generator Panel 5

3.4 MIDI Audio Panel 5

3.5 Filters Panel 6

3.6 Record & Playback Panel 6

1 Background

Whereas past scientific efforts were aimed at reproduction of the electronic voice phenomenon –or EVP– by using technology of alleged successful recordings, the SpiriGate project works from a scientific hypothesis and uses modern Digital Signal Processing (DSP) techniques to attain the best audible results. Currently, EVP is part of the paranormal realm, as were many phenomena from the past until scientific proof has shown otherwise. Likewise, the SpiriGate project uses current-day science to build its hypothesis to give this phenomenon a chance to proof itself. The hypothesis which forms the basis of the SpiriGate project is that voices heard on recordings are manifestations of deceased people and that these entities require some form of energy –a *carrier*– to manifest like, for instance, white noise. When we further assume that the entities are able to just slightly alter this noise to form something that sounds like a voice, we might be able to distil this voice from the recording. In scientific terms, we filter the carrier from the recorded signal.

But which carrier is best, a carrier that can be filtered easily, like a pure tone, or a carrier that is optimal for the entities to manifest? This is a trade-off between our technology and 'theirs', and the optimum should become apparent by experiment. A non-audible carrier could even be considered, like a light source, a laser, or any other electromagnetic form of energy. Since at this time there is no known indication that other forms of carriers lead to any results, these are abandoned for the time being. So, on the one hand, there is the question of carrier technology, and on the other hand, the choice of filtering technology. SpiriGate software leaves this quest to the user, or experimenter, that's you!

Different audio sources can be used simultaneously, as well as different filtering techniques. Not all ideas are implemented yet. Carriers include: white noise; a tone generator; and MIDI audio. Filters include an in-place recursive moving average filter. Both time-domain and frequency-domain filtering techniques will become available in due time.

Another important aspect is the accuracy of the measurement. In other words, the sensitivity of the recording. A very sensitive microphone should have preference but there is more. Multi rate sampling is a technique that can be used to make a recording more sensitive. By recording with a high sample rate, say 60 000 Hz, followed by decimation to a lower sample rate, say 6000 Hz, a more sensitive recording is attained compared to a 'normal' recording of 6000 Hz. These aspects enable us to hear more.

Yet another important aspect is that SpiriGate works in real time, meaning that we can tune settings while we're recording, and hear the result almost immediately. SpiriGate makes use of the fact that our minds are very well capable of detecting optimal settings. See SpiriGate as an analog radio tuner and we're adjusting the frequency band knob to attain the best reception...

SpiriGate is a non-profit side project of Hummeling Engineering simulation software development. See www.spirigate.com and www.hummeling.com for more details. Don't hesitate to contact us when you're faced with installation problems, or when this manual isn't clear, or for bug reports, or for moral objections, theological elaborations, principal disagreements, or just to kill time. Simply send an e-mail to info@spirigate.com.

2 Installation

Installation of the SpiriGate software is different for different operating systems. In order to reach the largest population, the SpiriGate software is written in the platform independent Java[1] programming language. The Java virtual machine (JVM), required for SpiriGate, is available for most popular operating systems, and you'll probably already have it installed on your computer. SpiriGate is tested on Windows and Linux and should also run successfully on a Mac.

2.1 Microsoft Windows

First, extract all files from the downloaded archive (SpiriGate.rar or SpiriGate.zip) and place in a folder named 'SpiriGate'.

SpiriGate can be launched by double-clicking on the SpiriGate.jar file in newly created folder from Windows Explorer. Or alternatively, from a Command Prompt: `java -jar "SpiriGate.jar"`. A Command Prompt can be opened via: Start | Run... and enter `cmd`.

2.2 GNU/Linux

First, extract all files from the downloaded archive (SpiriGate.rar or SpiriGate.zip) and place in a folder named 'SpiriGate'.

The SpiriGate.jar file needs to be made executable. Open a Terminal and 'cd' to the newly created folder which contains SpiriGate.jar. Enter `chmod a+x SpiriGate.jar`. SpiriGate can now be launched by issuing the command: `java -jar "SpiriGate.jar"`.

2.3 Mac OS X

First, extract all files from the downloaded archive (SpiriGate.rar or SpiriGate.zip) and place in a folder named 'SpiriGate'.

Please refer to a general guide on how to run Java programs.

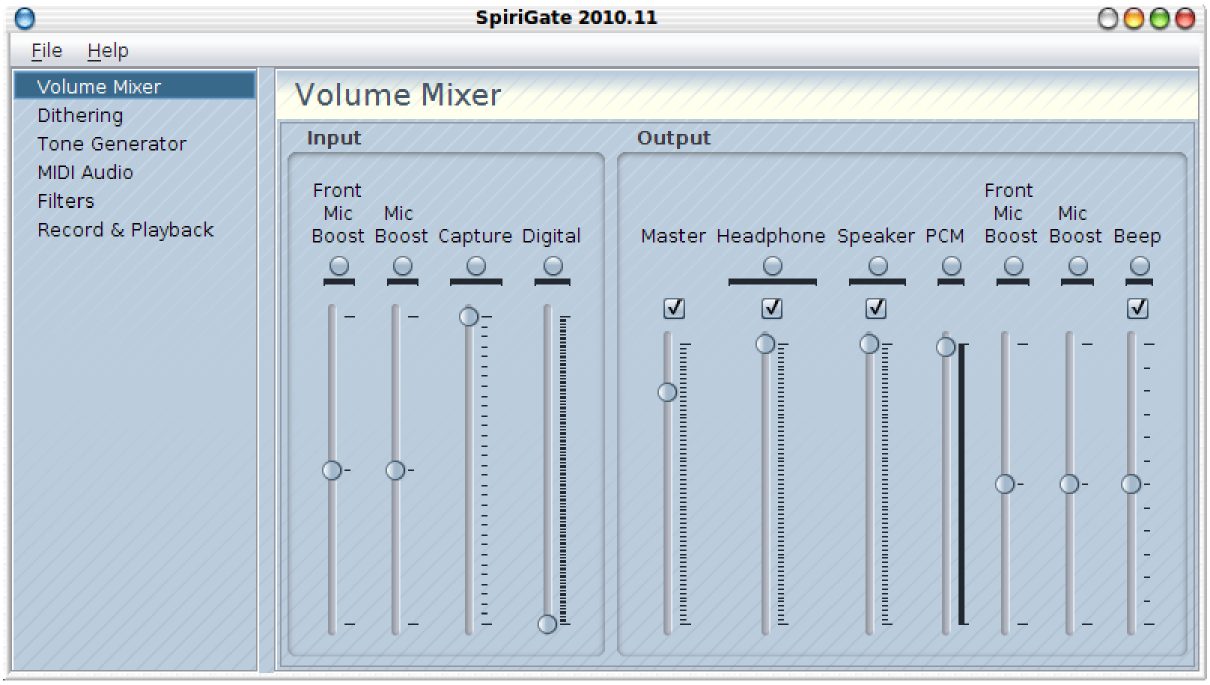
3 Getting Started

SpiriGate consists of several panels which will be discussed in detail in the following subsections. These panels can be accessed by simply clicking on the tree on the left.

3.1 Volume Mixer Panel

The volume mixer panel looks different depending on the audio channels available on your computer. SpiriGate looks for input and output channels and sets the volume sliders and mute check boxes to the current values. Note when any of these settings are adjusted outside of SpiriGate, it won't be noticed by the volume module.

[1] Oracle and Java are registered trademarks of Oracle and/or its affiliates.



3.2 Dithering Panel

Dithering is a technique known to science for being able to raise otherwise undetectable sounds above the background noise level. According to a theory called *stochastic resonance*, adding a little bit of noise can amplify weak signals 'hidden' within the signal.

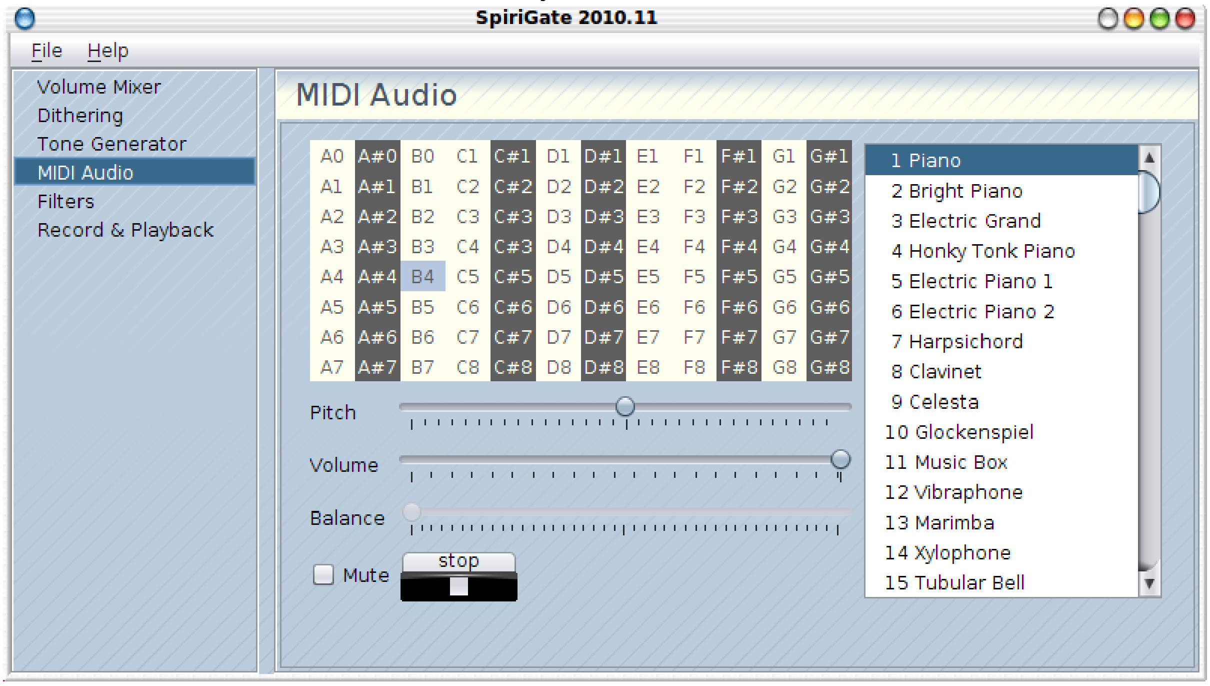
Currently only white noise is available.

3.3 Tone Generator Panel

The tone generator simply sounds one or more pure tones, separated by spaces. All at the same volume level. By default, the frequencies of the Spiricom device are given.

3.4 MIDI Audio Panel

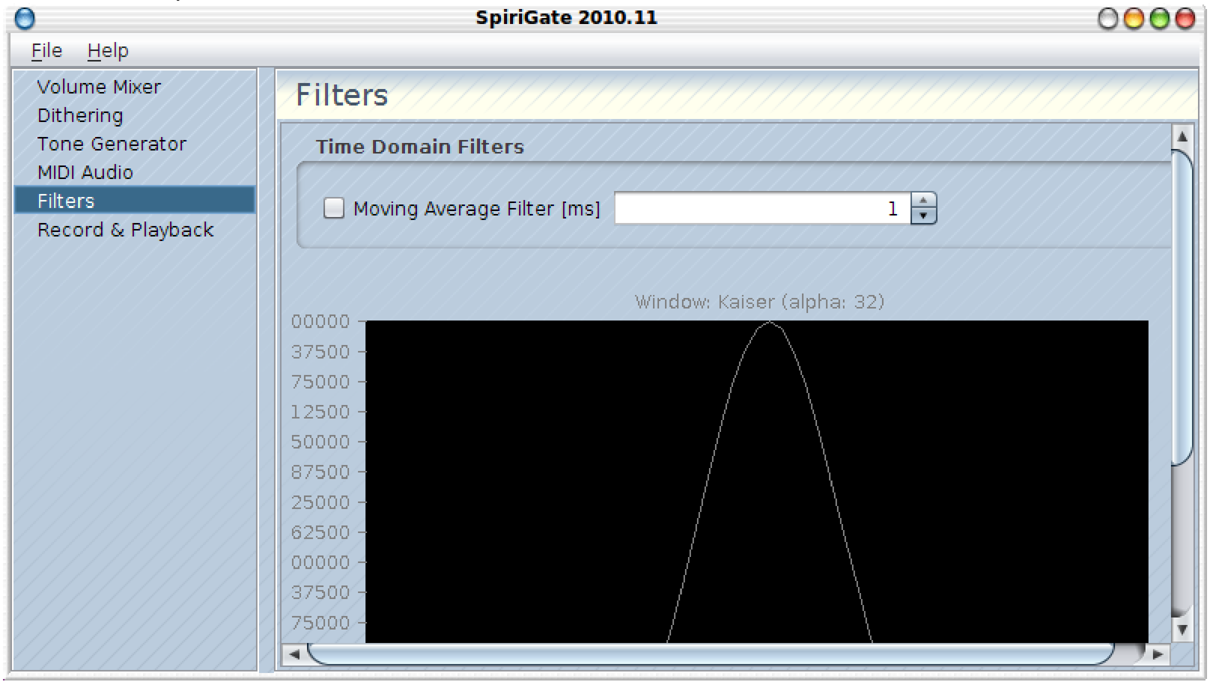
The available instruments listed in the MIDI audio panel differ per Java installation. Look for instruments that sustain sound indefinitely for a useful carrier.



3.5 Filters Panel

Currently only one time domain filter is currently implemented. An *inplace recursive moving average filter* is most effective to filter noise from a recording. Frequency domain filters will be implemented in due time.

It also shows the used *windowing kernel*. A future software release will include an option to select the preferred window kernel.



3.6 Record & Playback Panel

Digital sound recordings are characterised by several parameters that require some explanation. Contrary to analog recordings, digital recordings have a *sample rate*, which is the number of acquired samples per second. And each sample is recorded with a given accuracy, which is determined by the *sample size*. Greater sample size leads to higher accuracy.

An important fact to keep in mind is that the sample rate of a recording puts a limit to the highest tones that can be represented by that signal. In practice, the sample rate should be at least double of the frequency of the highest tones we want to capture. (This is called the *Nyquist criterion*.) Human hearing is limited to sounds with a frequency ranging from about 20 Hz to 20 000 Hz, while human speech is limited to about 3000 Hz. Hence, for voice recordings the *decimation frequency* shouldn't be lower than 6000 Hz. If we would make a recording of a pure 3000 Hz tone with a sample rate of, say, 4000 Hz, the recorded tone will be *aliased*, meaning that playback of the recording will sound like another (lower) tone which wasn't really captured. Aliasing is an artefact of digital sampling and must be prevented in order not to obscure the recording. This is done automatically by filtering (a low-pass filter) of the signal just before the decimation step.

A window containing four graphs appears when a recording is initiated. These represent time domain and frequency spectra of the recorded signal both before and after decimation.

The *sample gain* is the amplification factor of the incoming signal. Increase this value until the signal *clips*. Clipping is when you start hearing glitches in the recording represented by spikes appearing in the *wet time-domain signal* graph. Using headphones enables higher sample gains, use them whenever you can!