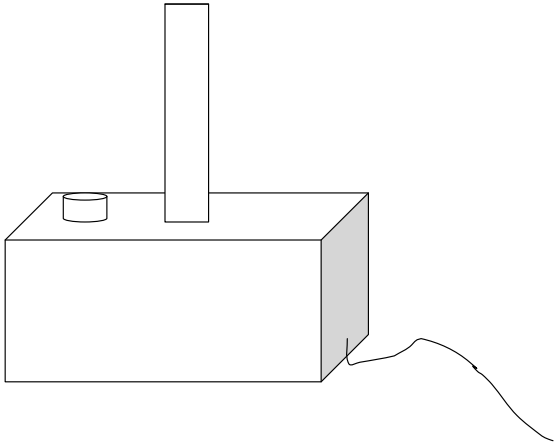
Spectral Tourist

Joystick operated sound production software
for Macintosh Computers (OS 9.1 & .2)
Documentation and
CD recording of performances
The Spectral Tourist

Martin Parker
June 2003



Spectral Tourist
September 2002-June 2003
Software documentation

The relationship between a musician and an acoustic instrument is not solely concerned with converting human gesture into pitch and timbre: there is a process between player and instrument that involves resistance. It is within the struggle to produce and then control a sound that conflict between a musician and his instrument can give rise to complex and interesting results.

It is a continuing struggle to make computer-based sounds that breathe with their own life and to play the computer like a musical instrument. Interfaces for transferring real-world data into the computer such as fader boxes, tracker-ball mice, graphics tablets, game controllers, wireless devices, motion tracking etc., are increasing in flexibility and interest and through software such as Max MSP[1], these interfaces can offer reasonable ways of converting human movement into sound. Joysticks, for example, are a ubiquitous, inexpensive and a generally reliable piece of technology. An external MAX object was written by Adam Schabtach[2] in 2001 to collect joystick data and view it within Max MSP.

However, there is nothing inside the hardware of most joysticks to curb a gesture. They can be pulled between extremes in milliseconds, while with a real world instrument such as a saxophone, extremes are very hard to reach and have to be found with care to avoid complete breakdown of the system and a split note. Therefore, grafting the joystick's extremes to the extremes of one's software may result in an artificial and possibly unfulfilling musical experience.

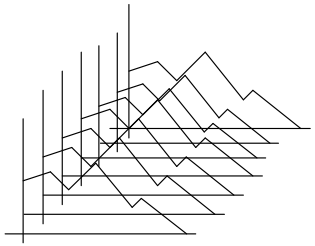
One way in which to create some 'resistance' is to imagine the joystick as a tool for travelling rather than for gesture mapping. In computer games for example, one is never in the same place for long. Pushing forwards with differing degrees of intensity means that one travels forward at different speeds. This approach has been implemented in the *Spectral Tourist,* although the landscape traversed is not one of snipers and guard dogs but the hilly terrain of a spectrogram.

A spectrogram can be an interesting landscape to behold but as with most spectral analysis and processing, the results depend on the quality and harmonicity of the source. Essentially, a performance with the *Spectral Tourist* involves taking a journey through a sequence of spectra stored within the computer's memory. The spectrograms can be generated from live input or from sound files stored on the computer.

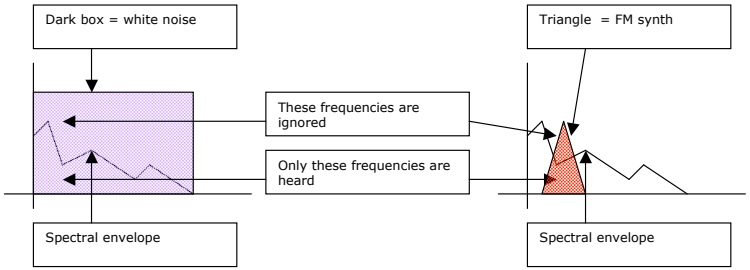
---

[1] Max MSP, graphical programming environment version 4.1 *http://www.cycling74.com*

The spectra stored within the *Spectral Tourist* preserve to a high degree the frequency and timbral quality of their sources only the sound is freed from its original place in time. Imagine each sheet of spectral data lined up: the joystick can jump from sheet to sheet, morph from one sheet to another or smooth between sheets over time:



The spectral data is re-animated while you travel with anything from live signal to white noise and an FM synthesizer. Imagine again the spectral data behaving like a filter for white noise. The noise could be described as a fairly complete sound source featuring most frequencies almost all of the time. When the white noise is filtered by the spectrum, the resulting sound will be a close representation of the source. However, if the re-synthesis sound source is less stable than white noise, the effect is to bring a mesh between the stored spectrum and the impulse sound:



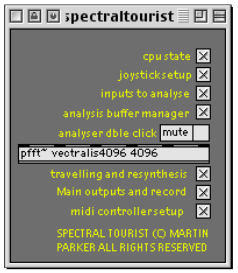
The following pages explain in some detail how to get a result from the SPECTRAL TOURIST (version 1); like most instruments, familiarity with the idiosyncrasies and the internal resistance of the system will yield better results.

[2]Adam Schabtach author of the **insprock** object, downloadable from
http://www.studionebula.com/files/IS_Object_010901.sit.hqx
(URL valid 20 April 2003)

Spectral Tourist – System Requirements

Absolute minimum 500mhz Macintosh G4 processor
256mb RAM
Allow plenty of disk space for storing data and recording performances
Input sprocket compatible joystick
OS 9.1 (or 9.2.2)
This software will not work in any version of OSX. Future versions may be ported to OSX. Please contact mp@tinpark.com for more details.
If you are getting desperate with the processor, try reducing colours to 256 and mute any patches that are not in use.

Spectral Tourist – main window



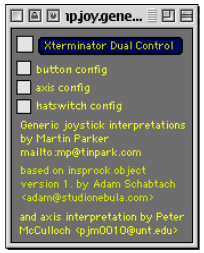
When you first open the *Spectral Tourist*, you will encounter this window in the top left hand corner of the screen. The default set-up of the screen has been optimised for a resolution of 1152x768, the same dimensions as a Titanium Powerbook 15'. It is possible to move these windows around the screen to find an optimal look for your system.

The grey boxes on the right open and close windows depending upon whether you need to view them.

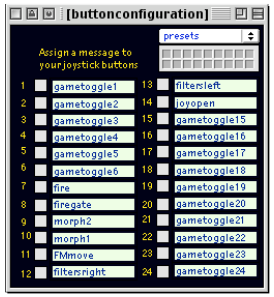
CPU state

This small window allows you turn on the software by clicking inside the grey box next to the DAC (Digital Analogue Converter). You can also monitor the CPU level. This is important. In trial versions of this software the sound tended to break up when CPU cycles peaked above 51%. This was always defeatable by muting the processing within the patch pfft~ vectralis4096. It is possible to mute the analysis patch directly from your keyboard using the M button.

Configuring your joystick[3][4]

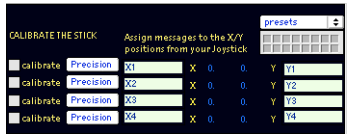


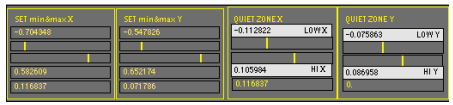
It is possible to plug any input sprocket, USB compatible gaming device into your computer and configure it for use inside the *Spectral Tourist*. Select your device from the dropdown menu and turn it on with the button to the left of the menu.



Then configure the buttons. Click the buttons on your stick and when you find the two buttons you want to use to allow travelling, type *fire* and *firegate* inside the boxes. These two buttons open the gates that allow you to move the stick and travel through the spectra. The other button assignments are fairly self-explanatory but here is a table of the messages and their action.

Message	Action
fire	Opens gate for travel with stick, also triggers new preset in FM synth
firegate	Allows movement when fire is down
Morph1 and morph2	Allows morphing in the two travel patches
FMmove	Allows changing parameters of the FM synth
Joyopen	Allows the stick and fire button to change synthesis parameters
Filtersright/Filtersleft	Allows stick to change the filter amplitudes in the two travel patches





You are allowed up to 4 different axis on your stick but the Spectral Tourist only uses the main stick. They may be a little 'noisy' so calibrate them using the calibrate dialogue and set the extreme X and Y numbers to be set at –1 and 1. You can also establish a quiet zone where the stick may spit data out even though you aren't touching it.

---

[3] A note on the *mp.joy.generic* object

This is the control centre where data is collected from the USB port of Apple Macintosh computers. The computer should be running OS9.x and can only be used with input sprocket compatible joysticks. (This software is currently not OSX compatible and will not work on Windows machines.)

I have successfully tested four joysticks made by GRAVIS, Kensington Technology Group, a division of ACCO Brands, Inc. http://www.gravis.com

The following sticks and game pads have been used.

Exterminator dual control (very 'noisy' stick)

Destroyer Extreme

Eliminator Game pad

Eliminator Shock Game pad

mp.joy.generic calls the USB sprockets to ask if any gaming devices are connected. This is achieved through the **insprock** external designed by Adam Schabtach[4]. It is possible to use as many gaming devices as your system can handle but you need a separate instance of mp.joy.generic to read and send data simultaneously. To choose between different devices one at a time, you only need one instance of mp.joy.generic

Data from the sprockets is interpreted and parsed to various places where the user can assign a unique message to be read in other patches. Axis numbers range –1.0 t0 1.0, with 0 at the centre. Hat switches send out a number (e.g. 8 switches will send 0-7) and buttons give a toggle (0 = off, 1=on)

Each button, stick and switch can be assigned a unique message that is sent wherever the user wants. This object can cope with any compatible controller boasting up to 4 axis controllers, 24 buttons and 16 sets of hat switches.

[4] Adam Schabtach author of the insprock object, downloadable from
http://www.studionebula.com/files/IS_Object_010901.sit.hqx
(URL valid 20 April 2003)

Midi Setup

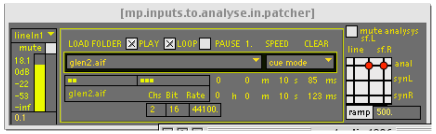
You will need hardware such as a KENTON CONTROL FREAK that has access to the first 16 midi controller numbers. They are assigned thus:

Parameter	**Controller number**
White Noise Amplitude	5
FM synthesis Amplitude	12
Click synthesiser Amplitude LEFT	6
Pan LEFT	4
Pan RIGHT	13
Re-synthesis Amplitude (LEFT, 1)	1
Re-synthesis Amplitude (LEFT, 2)	2
Re-synthesis Amplitude (LEFT, 3)	3
Re-synthesis Amplitude (RIGHT, 1)	14
Re-synthesis Amplitude (RIGHT, 2)	15
Re-synthesis Amplitude (RIGHT, 3)	16
Clicker Synth Amplitude (RIGHT)	11
Travel speed (LEFT)	7
Travel Speed (RIGHT)	10
Offset Left	8
Offset Right	9

If you have several controllers attached to your system, you will need to select which controller you want to use so open the midi set-up dialogue box.

If you move your sliders around, you will see the dialogue box sliders change but only if you are moving *controller* numbers 1-16.

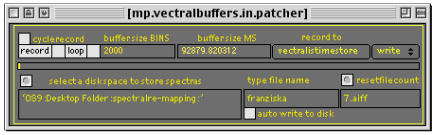
Routing input to analysis and/or re-synthesis



This is the input router for the analysis. Drag and drop a selection of sound files onto the window or load a folder of sound files by clicking on LOAD FOLDER. It is important to click CLEAR before dragging files into the player from the desktop. Alternatively analyse a live signal.

The matrix on the right allows you to route these signals to the analysis patch (anal) or to the resynthesis patches (synL and synR).

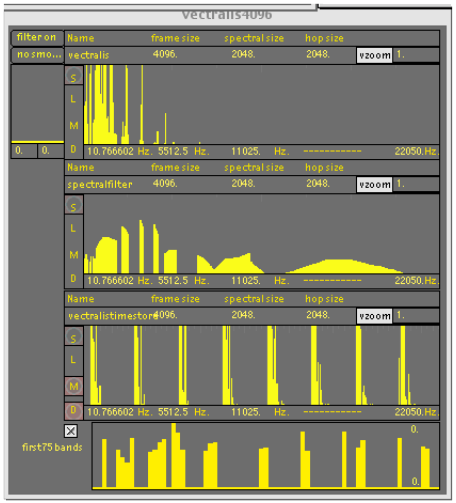
Vectral buffers – store the analysed signal



This is the buffer manager where incoming spectral analysis is stored. Up to 15000 frames can be held in up to four buffers named *vectralistimestore (2,3,4).* (You must have at least 256mb of RAM assigned to the software before you start to run the *Spectral Tourist*).

You can also load spectra from disk, automatically cycle from one buffer to the next when each one fills up and automatically write the buffer to disk when full with a unique file name and location specified before recording begins. This keeps hands free and the analysis constant during a long analysis session.

The Spectral Analyser – Vectralis4096



Turn on filtering

Smooth between frames

The stored spectra

Incoming spectral analysis, 2048 bands, *hanning* window

Filter the analysis by drawing an envelope (click D on the right)

Control the first 75 bands of the filter more preciselv

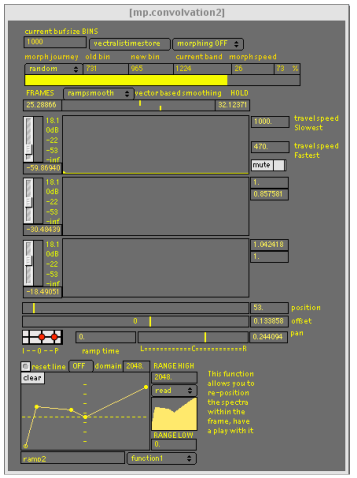
Vectralis4096 is a pfft~ sub patch that creates and holds the spectral data. It can easily be muted with the M key on your keyboard to save processing power (the re-synthesis window opens automatically to show you that analysis is not happening and that re-synthesis can begin without overloading CPU). If you have a fast and stable system, it is possible to re-synthesise while analysing.

With a 4096 window size, there are only 2048 useful samples, the last half of the window is ignored in this patch because it is a direct inversion of the FFT analysis. Interesting effects might be achieved if this was stored as well, the computer has to work out this inversion anyway, so future versions might make this implementation.

With this window it is possible to filter the incoming signal by drawing a filter envelope, click the D on the left of the spectral filter window. To get more accurate results, use the sliders at the bottom of the patch to access the first 75 frequencies of the analysis. Alternatively zoom into individual samples and draw a new volume using the M icon (just above the D on the left side of the spectral filter window). Filtering the incoming signal may have several uses: for example reducing noise or enhancing/masking certain frequencies that might prove dangerous during re-synthesis.

It is also possible to smooth between each frame of the analysis, smoothing over samples and frames linearly (*rampsmooth*) logarithmically (*slide*) or temporally (*deltaclip*).

Managing the stored spectra - convolution filtering – manipulating the data



Select the buffer with spectral data, there are four

Smooth between frames

Current re-synthesis window used as a filter for re-synthesis signal

Amplitudes of the re-synthesis of each window

Position in the buffer from 1- number of BINS

Mix the signals and pan

Function to spread the spectrum, have a play with it

Morph between frames either sequentially or randomly. Change morph speed from 1(slow) to 100(fast)

First filter window select frequencies with amplitudes between two numbers

Second filter window select frequencies with amplitudes between two numbers

Offset position from –1 to 1, 0 is normal

Inside this window, it is possible to manipulate the spectra that were stored, filter in and out certain frequencies, travel through the spectra and control the amplitude of the re-synthesis.

From the top:

Select which buffer you want to travel through (vectralistimestore 2, 3 or 4). Selecting a buffer automatically sends the number of frames inside that buffer to the joystick controller and GUI, so it is a good idea to select a buffer before performing any other operations. Select whether morphing is off with the mouse or with the button you assigned to morphing. You can also set the morphing parameters – to travel through each frame in sequence or to morph to randomly chosen frames (none are repeated) and to select the morphing speed. 1 is slow, 100 is very fast.

It is also possible to smooth between each frame using *rampsmooth*, *slide* or *deltaclip* parameters.

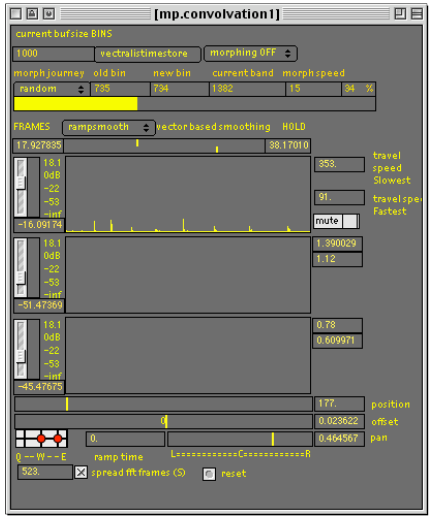
The top waveform shows the selected frame, the two other windows show the selected frame with certain amplitudes filtered out. These can be set with the mouse or by holding down filterLEFT, (*convolvation1)* or filterRIGHT *(convolvation2)* and moving the analogue stick.

The mouse or a midi fader box can control the volume of each window of re-synthesis. It is generally important not to push these levels too much to avoid distortion, unless of course, that is what you want to achieve.

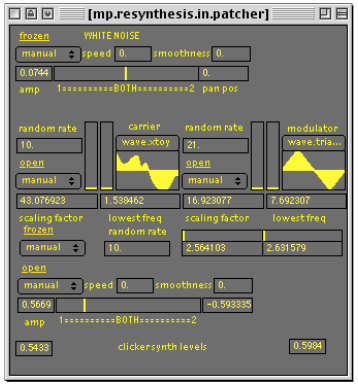
The slider below the waveform windows shows the current frame within the stored spectra (from 1 to the numbers of frames) and the slider below that shows your offset. This means you can change the fundamental frequency of the bin.

Below the slider, the matrix allows you to send the signals to the output of the computer and you can adjust the stereo pan position. The pan can be manipulated by midi controller numbers 4 and 13 and the speed of travelling can be changed with contrller numbers 7 and 10.

The function at the bottom of the window allows you to draw a curve for the spread of the FFT frequencies. A straight line running from bottom left to top right is the normal frame view but pulling the line around allows you to re-map the spectra. Have a play with it.



The left side of the screen is almost the same as the right. The only difference is that rather than use the function window to alter the spectra, you can spread out or reduce the bands instead. This means you can crunch all the bins of the stored spectra into one window or spread the lower frequencies out across the whole spectrum.



RESYNTHESIS

By pressing *SHIFT J*, the joystick is active in this window. Hold down *joyopen* to control the parameters. You can also hold *FMmove* and press *fire* to toggle through different presets of the FM synthesis. Press *SHIFT R* to make the parameters shift randomly and *SHIFT M* to make adjustments manually. The horizontal slider allows you to send the re-synthesis signal only to convolvation1 or 2 or to both. The amplitude of the white noise is mapped to midi controller number 5 and the amplitude of the FM synthesis is mapped to controller number 12. Amplitudes of the clicker synth are mapped onto controller numbers 6 and 11.

By holding down the FIRE button on your stick, the click synthesizer will come into play. This sends a harsh audio click into the filter envelope and the rate of the clicks is adjusted with the X,Y positions of the stick.

The Spectral Tourist makes use of the following third party externals

From the Real Time Composition Library vs. 3.3
© 1993-2000 by Karlheinz Essl <essl@eunet.at>
Round
From the "jimmies" by Zack Settel, © 1994-98 IRCAM.
Scale

Db1

From Timothy Place – tap.tools 1.3

tap.split~ © Timothy Place

Adam Schabtach

Input Sprocket object version 1.0

The Spectral Tourist

Audio CD Duration 29.10

The eight tracks on this short CD represent some of the possibilities for performing solos with the *Spectral Tourist*. Each track was recorded in one take travelling through spectrograms that were generated from sound files stored on my laptop hard disk. Of course, the *Spectral Tourist* is not limited to solos and it is just as possible to work with a live musician.

The tracks play well together and they are best considered as a complete whole.

1.	*A word from the composer*	02.06
2.	*Frank London's Klezmer All Stars (a)*	03.05
3.	*A shapely Buffer~*	03.45
4.	*J.S.Bach, 1st Prelude, Book 1 (a)*	01.11
5.	*Frank London's Klezmer All Stars (b)*	04.51
6.	*Another word from the composer*	01.54
7.	*J.S.Bach, 1st Prelude, Book 1 (b)*	02.39
8.	*Anton Webern, Lansammer Satz*	09.31
	Total playing time	29.10