v2MSy
(almost a palindrome)

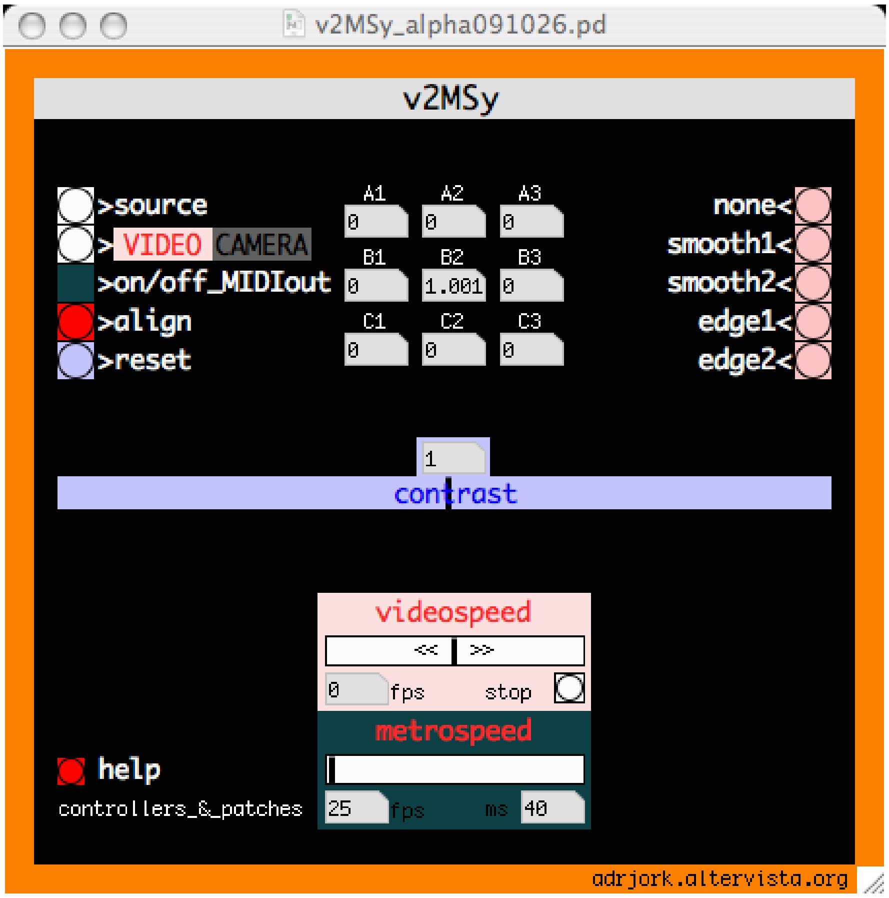
...is a...

video-to-midi synth

by adrjork (http://adrjork.altervista.org)
(download v2MSy)

QUICK GUIDE

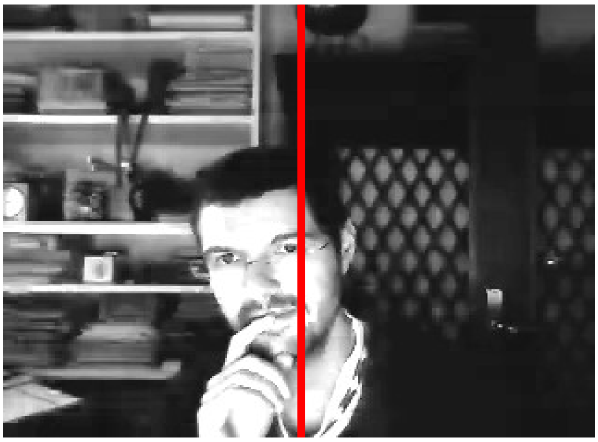
interface



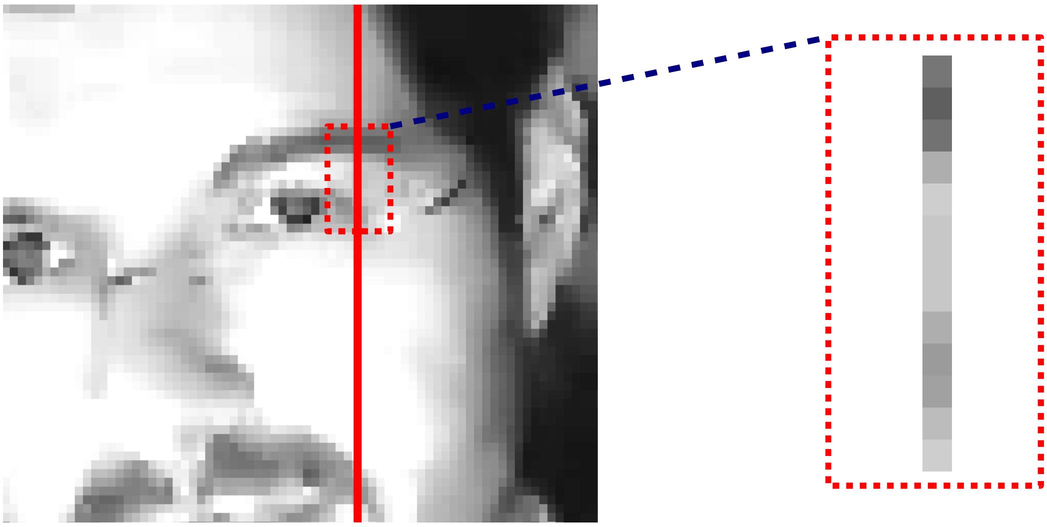
">source"	of video files and/or webcam device
">VIDEO|CAMERA"	shifts on-the-fly between source's type
">on/off_MIDIout"	starts/stops MIDI flux
">align"	verifies Behringer's connection to this patch
">reset"	for contrast's slider
"A1...C3"	is the image-convolution's grid
">none"	default convolution's preset
">smooth1...edge2"	various convolution's presets
"contrast"	for image's contrast
"videospeed"	video's speed (fps) in forward and backward
"metrospeed"	speed of MIDI output (fps and ms)
"help"	opens subpathes and controller's interface

main target

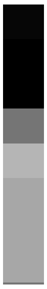
v2MSy loads video files and/or webcam and outputs a MIDI flow on-the-fly. The mechanism is smilar to carillons' cylinder...
Any video is a sequence of frames, every frame is a grid of pixels, and every next frame changes (even if a little) the color of the former frame's pixels. Now let's take the central vertical axis of a video (in red):



this axis is a column of pixels, from bottom to top...

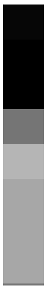


In a grey-scale image, every pixel is described by a single gray value (from 0.0=black to 1.0=white). An extract of our central vertical axis could be...



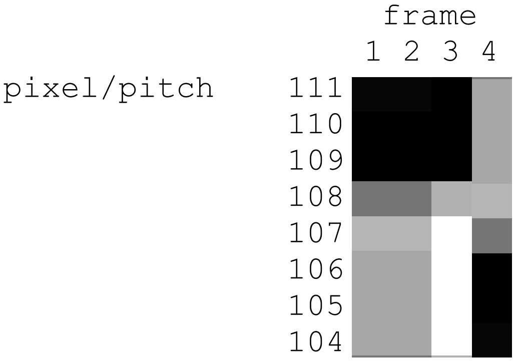
= 0.0
= 0.0
= 0.0
= 0.2
= 0.5
= 0.3
= 0.3
= 0.3
… = …

In PureData, default gemwin (the video screen of Gem, an external video library for PureData) is a square of 500*500 pixels, but just image to reduce this size to a 128*128 grid. In this grid we have a central vertical axis made by 128 pixels. Why 128?
You probably know that MIDI is a protocol that couples musical notes' pitch with integers from 0 to 127 (128 values). Every MIDI note number is described with the value of its volume (named velocity). So, we have 128 couples of numbers: pitch + velocity.



	pitch	+	velocity
=	111	+	0.0
=	110	+	0.0
=	109	+	0.0
=	108	+	0.2
=	107	+	0.5
=	106	+	0.3
=	105	+	0.3
=	104	+	0.3

In v2MSy, 0.0 (black pixel) matches with velocity=0 (note-off, silence), while 1.0 (white pixel) matches with velocity=127 (the maximum value of volume).
In our video, or webcam, our central vertical axis is refreshed at 25 frames per second. It means that at every frame, v2MSy analyses the same number of pixels (the 128 pixels of the central vertical axis), and then it ouputs the same number of MIDI notes (128, that's the whole MIDI range) 25 times per second, but every time, the velocity of each note could change!
This is the behaviour of v2MSy's MIDI output after pixels' analysis (this is a 4 frames extract)...



	1	2	3	4
111	…	…	…	D#
110	…	…	…	D
109	…	…	…	C#
108	C		C	
107	B		B	B
106	A#		A#	…
105	A		A	…
104	G#		G#	…

At 1st frame three notes are off (black), and 5 notes are on but have different volume: C plays *pp* (*pianissimo*), B plays *mp* (*mezzopiano*), and A# A G# play *p* (*piano*).
At 2nd frame nothing changes, then this chord will not be repeated, and the result is that we obtain a two-frames long chord!
At 3rd frame the pitch of C B A# A G# changes: now C plays *p* (*piano*), and B A# A G# play *ff* (*fortissimo*). There is a change of velocities, then our chord will be repeated!
At 4th frame there's another change: D# D C# start to play *p* (*piano*), C doesn't change so C will be maintained, while B will be repeated in *pp* (*pianissimo*), and finally A# A G# will be stopped.

requirements|installation|settings

Recommended system: 1GB RAM, 128MB video card, dual-core CPU.

v2MSy outputs MIDI notes, so you need a software that receives MIDI values (pitch, velocity) and converts them into sounds or score or whatever you want (i.e. Logic, Rosegarden, Cubase, etc.) To connect v2MSy's MIDI-out to receiver-sw's MIDI-in, you need also a virtual MIDI cable (*)

If you want to use the patch with a webcam, you can use USB devices or DV-cams with firewire (iLink). If you don't have a webcam, you anyway can use v2MSy loading a video file.

-Download and install Pd-extended (http://puredata.info/downloads)
-Install a virtual MIDI cable, and create two buses(**)
-Download v2MSy and unpack v2MSy.pd wherever you want
-Start Pd-extended and open v2MSy.pd
-Setup Pd's MIDI preferences with two output devices
-Start your favourite MIDI-receiver software

**) v2MSy outputs two MIDI flows:
-MIDI channel 1 for Behringer's motorized faders feedback
-MIDI channels 17...24 for MIDI-notes-out
...
In MIDI protocol, channels travel through “buses”, every bus manages up to 16 channels: channels_1-16 for bus_1, channels_17-32 for bus_2, etc. To manage two separate MIDI flows, we just have to work with separate buses. To do that, we have to assign two different channel's numbers that refer to two different buses (i.e. channel_1 and channel_17: the first will travel atomatically through bus_1, while the second will travel through bus_2). Obviously, first of all we need to setup multiple buses in our virtual MIDI cable.

*) OSX HOWTO (IAC Driver)
In your system, go to Applications > Utility > MIDI Setup > IAC Driver, click on “Device is online” then “Add Port” to obtain Bus_1 and Bus_2, then “Apply”.
Go to Pd-extended's Preferences > MIDI settings > use multiple devices, and choose:
-input device 1: BCF2000
-output device 1: BCF2000 (because in v2MSy is channel_1)
-output device 2: IAC Driver bus 2 (in v2MSy MIDI-notes-out is channels_17...24)
In this way, Behringer has feedback and Logic receives MIDI notes (to play them through a virtual instrument or to record them into a score).

*) LINUX HOWTO (Jack and ALSA) ...to be tested

*) WINDOWS HOWTO (MidiYoke) ...to be tested

let's start

Select a type of source clicking on ">VIDEO|CAMERA" (selected type is lighted up) and then choose your source by clicking on ">source". The gemwin will appear with your video or webcam. Then, if you select a source also for the other type, ">VIDEO|CAMERA" will allow you to shift between video and webcam on-the-fly.
You can modify the image through the "contrast" slider (towards the right for positive values, towards the left for negative values - "reset" button for default contrast value=1), or using the convolution-grid's values A1...C3 (a great tutorial for image convolution-grid is http://www.eas.asu.edu/~karam/2dconvolution/).
You can also use convolution's presets (pink buttons on the right). To change the presets' values, simply click on "help" button to show the subpatches, then click on the little box [pd convolution_presets] and change the values.
If your source is a video, probably it doesn't go on...this because the "videospeed" box indicates "0 fps". Drag the slider towards the right to increase speed up to 25fps, or drag the slider towards the left to rewind the video up tp 25fps. Your video will run in loop: when it arrives to the end it automatically starts again from the first frame (obviously, if it moves backward, when it arrives to the first frame it starts again from the last frame).
Now you have a video/webcam-window on your desktop, but v2MSy doesn't output any MIDI flow. To output MIDI flow, first of all you have to set MIDI preferences in PureData (...later on this guide). Then, you must active the "on/off_MIDIout" button. Now v2MSy outputs a 128 MIDI notes chord every 40ms (25 times per second). You can also change the speed of this flux, from 40ms to 2000ms (a chord every 2 seconds).
"help" shows also two other subpatches: [pd keyboard_control] and [pd behringer_control].

KEYBOARD_CONTROL
Click on [pd keyboard_control] to open v2MSy's keyboard-based controller. This interface is useful for performances: a single slider controls all the functions of the main patch, so you don't need to move around your mouse. To select the function you want to control with the slider, simply use the keyboard shortcuts indicated on this subpatch: click on...
"key_ON/OFF" to active the keyboard shortcuts
"q...c" to control some sectors of the convolution's grid
"r...n" to control a symmetrical selection of the grid
"y...p" to active the convolution's presets
"l...h" to shift convolution/contrast/metrospeed(*)/videospeed
"bar" (spacebar) for video-pause.
*) metrospeed is the speed of the MIDI-notes-out in milliseconds

BEHRINGER_CONTROL
Click on [pd behringer_control] to open v2MSy's BCF2000-based controller. After you have properly setup Pd's MIDI output devices (as described above), try to move one of the motorized fader on

your BCF2000, and then press the “align” red button to verify the connection between your BCF and v2MSy and the feedback of the motorized faders. In this subpatch the functions are the same as keyboard_control subpatch.
First BCF's knob controls video speed, while second knob controls metro-speed.

thanks to

http://www.puredata.it/ and http://puredata.hurleur.com/ for the advices and the patience of all the guys. Grassie fioj!

***

adrjork is...

Adriano Castaldini

http://adrjork.altervista.org/