

OMax

The Software Improviser



© Martin lartigues

OMaxVideo

version 2.12

Georges Bloch, December 2018

OMax has been designed and developed by *The OMax Brothers*: G. Assayag, G. Bloch, M. Chemillier, Benjamin Levy

ImprotoK and DYCI2 have been developed by Jérôme Nika.

This document presents a new version of OMaxVideo ideveloped by Georges Bloch. The first version was conceived by G. Bloch with contributions from E. Rossez, A.-S. Joubert and V. Robischung

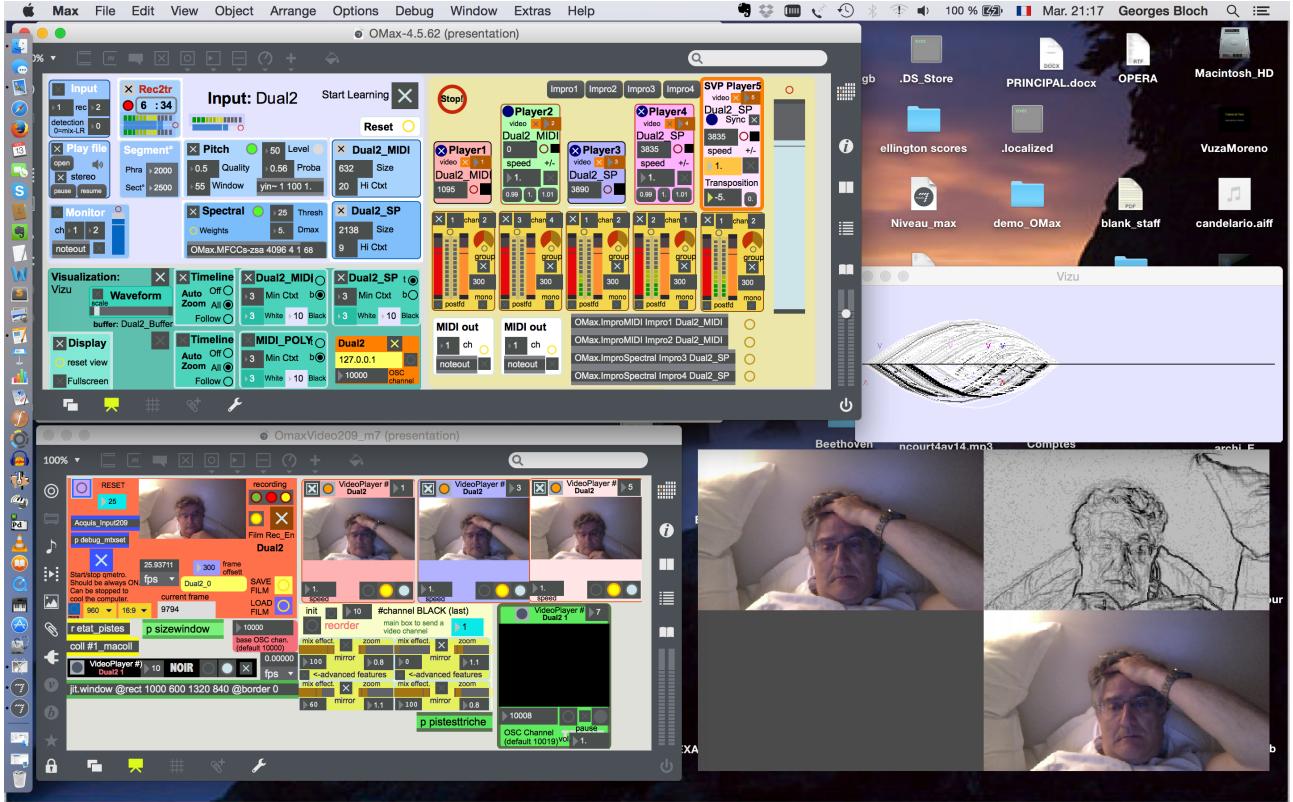
<http://www.ircam.fr/ircam/equipes/repmus/OMax>

OMaxVideo

Table of contents

<i>Introduction : Sound-Driven Video</i>	p. 3
I- What is Needed	p. 4
I-1 Two-Computers Principle	p. 4
I-2 Setup	p. 5
II- What you need to know before starting	p. 6
II-1 Online/Offline	p. 6
II-2 Possibilities and expressions used	p. 7
II-3 Use of OMaxVideo in a General Context	p. 7
III- Tutorial 1: OMaxVideo Online Quick Start	p. 8
IV- Initialization and performances	p. 11
IV-1 The Double-recording Trick	p. 11
IV-2 The Importance of Online Recording in Your Session	p. 12
IV-3 Memory and Performance Considerations	p. 13
IV-4 Modifying the Default Values	p. 13
V- Tutorials 2: Offline Video Quick Start	p. 14
V-1 Loading Existing Films	p. 14
V-2 OMax with an Existing Film: QuickStart	p. 15
V-3 Simulation of Offline Playing: Using OMaxVideo With Another [Any?] Musical Program	p. 17
VI- Tutorial 3: Acquisition	p. 20
VI-1 Setup	p. 20
VI-2 Controls	p. 20
VI-3 Camera Selection	p. 21
VII- Tutorial 4: Video tracks	p. 22
VII-1 A Video Track and its Corresponding Audio Track	p. 22
VII-2 Videoplayer Reference	p. 22
IX- Rendering, Matrix and Effects	p. XX
X- Tutorial 7: Cheating	p. XX
Appendix	p. XX
Using an Existing Musical Film	p. XX
X1 Constructing a Reference Table (funbuff)	p. XX

OMaxVideo



A typical OMaxVideo setup on one single computer. On the bottom right, the rendering window (which can be displayed on another screen). On its left, the OMaxVideo window with, left to right, acquisition - in orange - and reception with three players and one special player for displaying various movies or photographs during the performance. Since it's a single computer setup, one can see OMax 4.6 on the top, with its blue input and its five colored outputs .In this particular case, the iSight camera of the computer was used.

Readers of this part must be already familiar with OMax 4.5.

OMaxVideo requires jitter, the video extension of Max/msp. OMax 4.56 and above work with Max 7, as do OMaxVideo 209 and above.

OMaxVideo works on Intel Apple Macintoshes. The most recent machines are recommended.

Introduction: Sound-driven Video

OMaxVideo basically follows the same principle as the audio version of OMax.

In OMax and DYCI2 – whether pitch detection or spectral descriptors are used – events are listed into a data structure called an *oracle*; to each oracle event and datas corresponds a date in an audio buffer, the actual date of the recorded event.

In OMaxVideo, a video is captured as the musician is playing : usually it is a capture of the performance, but anything could be filmed as the music goes along. Even, another film of the same length as the sound file could later be used. As the musical improvisation goes on, there is a live montage of the film is according to the sound re-injection of the played passages. Therefore the

filmed performer is always in sync with the improvisation: the « live-filmed » musician can actually be seen playing something never played before. The video is *sound-driven*.

Although it is not its most interesting feature, OMaxVideo offers a pedagogical approach to OMax: and DYCI2. Whereas in its most successful improvisation patterns OMax and DYCI2 make it difficult to hear when recombinations take place, it is often easier to see it on OMaxVideo, since a film cut is generally not as hard to catch, especially when there is only one single shot of the musician.

OMaxVideo is just fascinating in the way it allows sound-driven video improvisation. You can even load films with no relationship at all with the music and have them be recombined according to the logic of the music played: each repetition of the same passage will provoke the same visual illustration.

I- What is Needed

I-1 Two-Computers Principle

During the conception of OMaxVideo, the video had to be run on a (second) different computer from the one running OMax or DYCI2. For many reasons: both programs are quite demanding in CPU and memory. It was (and still is) also more convenient. Furthermore, in recent versions, OMax allows several inputs at once: it becomes impossible for a single machine to handle several video inputs and outputs. This two-computer architecture remains the reference.

On more recent machines, two computers are no more mandatory. However, with two computers, it is easier to understand the way OMaxVideo works; it is also easier to manipulate, because one has combine two relatively large user interfaces, hard to master with only one single keyboard and mouse or trackpad.

Last but not least, the double computer concept makes it also easier to understand: you have a first audio computer running OMax, and a second one (or several others, one per audio input) running the video. If the video computer crashes, OMax (or the Video computer) continues running and it's much easier to restart (as we all know it's a very remote possibility: most computer programs are bug-free, and ours certainly are). For the sake of clarity, we'll speak of two computers, one audio (Computer_Audio) and one video (Computer_Video), even if we use the same machine (as in the picture presented above).

In any case, Computer_Audio and Computer_Video communicate with OSC. In the case of one single machine, it still does with the address 127.0.0.1. Computer_Audio, the OMax computer, is the boss, and the video part is slaved to it.

IMPORTANT: With only one machine, OMax video works MUCH better with multiples instances of Max7. You just copy your Max7 program and rename it, for example, MaxVideo. You then launch OMax on regular Max7 and OMaxVideo on MaxVideo.

By a way of consequence, OMaxVideo consists of several different programs:

1- Computer_Audio. In the OMax folder, a video-friendly version of OMax (with outputs of the dates of the players) is included into OMax 4.x.

- *To_video.maxpat*, (orange) with its Connect ON-OFF button takes care of the OSC connection
- The Audio Players from OMax have a video toggle and a Video channel number (by default, a Player names Player#N has the N channel number at startup).

To_video.maxpat is for acquisition (recording), the Video Players are for lecture.

IMPORTANT: OMaxVideo is not implemented for the MIDI part of OMax.

2- Computer_Video: the OMaxVideoW212 (or above) folder with OMaxMaxvideo212 folder and, generally, a ovideo-films folder somewhere to keep and record the films. The main program (now OMaxVideo212) can be aliased in the folder.-

I-2 Setup

“Ideal” setup.

- OMax computer (Computer_Audio), with a high-quality sound setup
- OMaxVideo computer (Computer_Video), with jitter installed, and the OMaxVideo Package
- High quality camera connected to Computer_Video
- Network connection between both computers (a WiFi computer to computer network is perfectly acceptable). In any case, a dedicated network between both machines is preferable, in order to keep a reasonably constant network latency.

“Minimal” setup.

- One computer with *two* instances of Max and all programs
- the internal audio input (microphone) and output (loudspeaker)
- a built-in iSight camera or webcam

This setup is sufficient for most of the Tutorial presented here.

“Reasonable” setup.

- A reasonably powerful computer with all packages
- A good soundcard and sound system
- A good quality camera.

II- What you need to know before starting

II-1 Online/Offline

Actually OMax and DYCI2 can be used in many different ways: but a very important aspect is whether or not you want to use video *online*, that is, to capture the ongoing performance. Sometimes, DYCI2 or OMax make sole use of « memory » repertoires; live improvisation, if any, is not taken into account by the machine. In this particular case, there is no need of live capture. The use of *online* audio by the machine is important, since it also entails the use of *online* video. The relative importance of the online and offline parts can vary according to the project, but have to be decided beforehand for setting up the machine. More is said in part IV

II-2 Possibilities and expressions used

- Video (audio) tracks : the video streams available at a given moment. Generally, they correspond to the available audio channels. However, there are exceptions: a constant one is the *NOIR* channel, resulting in a black screen. Another possibility is possible *TRICHE* channels, where existing images – films, photos – are projected independently of the incoming audio stream.
- Video (audio) sources : the sources from the channels. Typically, OMax has only *one* source (the audio stream from the program). This source is itself divided into different treatments (oracles), themselves rendered by different players. The players correspond to tracks, but the source is generally always the same one, that is, the actual original improvised performance. In DYCI2 and ImprotoK, there is a frequent call to different *memories* which constitute different sources. For example, in the « Three Ladies Project », Billie Holiday, Edith Piaf and Elisabeth Schwarzkopf were singing together. The sources were evidently distinct!
- OMaxVideo does *not* work with the MIDI part of OMax.

III-3 Use of OMaxVideo in General Context

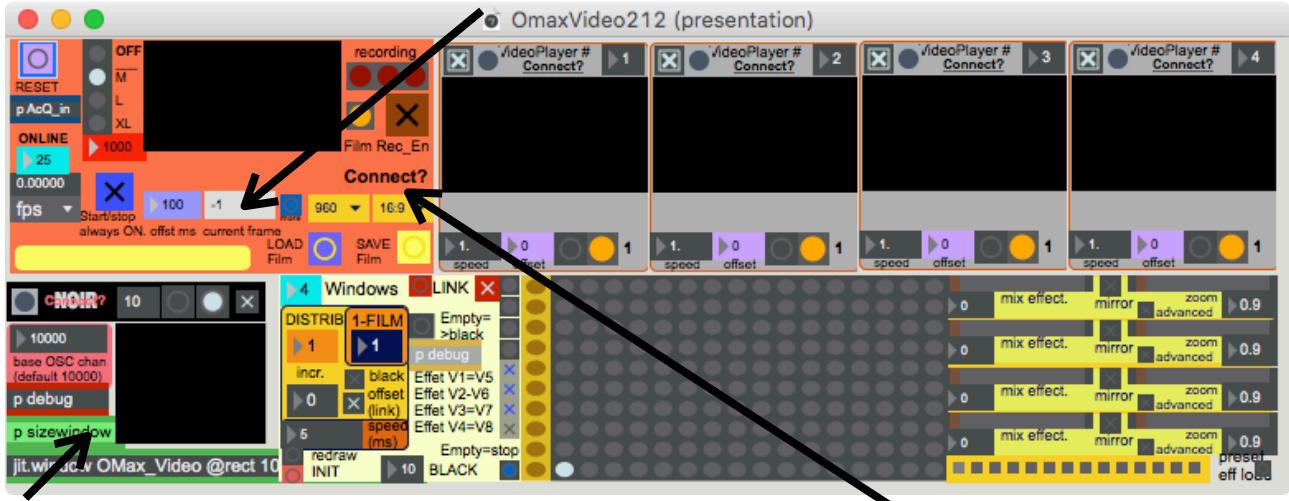
OMaxVideo is simply a video editor-mixer optimised being driven by the sound of the video file – or any other soundfile, but this is another story. It relies on three simple prerequisites:

1. Sound file and images are *separated*: this is the case for online recording, but this has to be, in order to be able to manipulate the sound file from a sound program. For existing films, countless softwares exist allowing to extract the sound and the image separately.
2. There is a *precise* table of correspondence between the sound and the images: the famous « funbuff » file. It can easily be constructed with the utility file **Make_funkbuff_OK present** in the Utility folder. The fun buff is connected to the film by its name (the same, with _fnb at the end of it)
3. Any editing date in the sound file is sent to the program. It realises on the fly an editing in the corresponding image.

III- Tutorial 1: OMaxVideo Online Quick Start

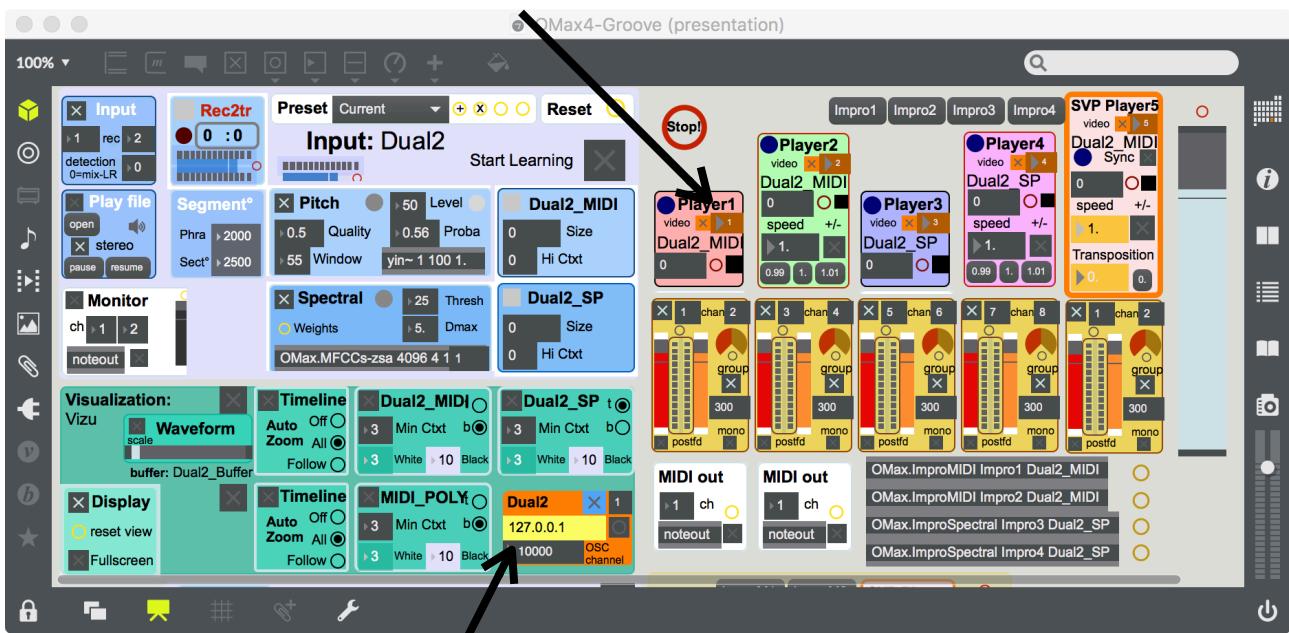
(If you don't understand the steps referring to OMax, refer to the OMax Tutorials.)

1. Launch OMaxVideo [on Computer_Video]. A rendering window appears on the right side of



the screen. You can make it larger or smaller if you want through the window called sizewindow; for the moment, keep it as it is. You'll notice two strange features on the acquisition part. First, the one yellow light above the word **Connect?** show that there is no connection with OMax (or DYCI2): it is quite normal, since there is no OMax, the same question can also be seen on top of each video player. Second, the current frame is numbered -1 (it will start at 0)

2. Launch OMax [on Computer_Audio]. Do the proper initialisation setup (turn on audio, etc.). The general connection to Video player takes place in the *ToVideo* orange box (bottom of the picture). The individual track connections are on each player, displaying the respective video channel and whether it is connected or not of the The MIDI part of OMax is not shown, as Video does not function with MIDI .

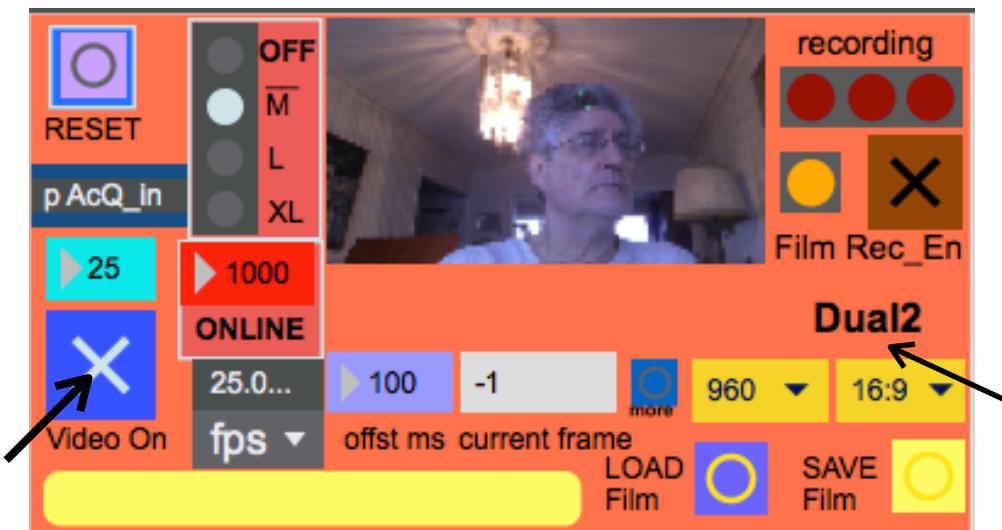


To_video (for input)

3. [ONLY IF YOU ARE USING TWO COMPUTERS, and/or if the OSC ports 10000 and 10001 are already used.] On To_video patcher [Computer_Audio], click on the button right to the patcher *To_Video* (pointed by the arrow on the figure above). You can set the IP address of Computer_Video and change the OSC port if necessary. Two main ports are used : the port given for acquisition (default 10000) and the port immediately superior (10001) for lecture. The same port should be set on the OMaxVideo window (below the orange acquisition rectangle). In general, you just need to adjust the address of Computer_Video, and can keep to the default ports.

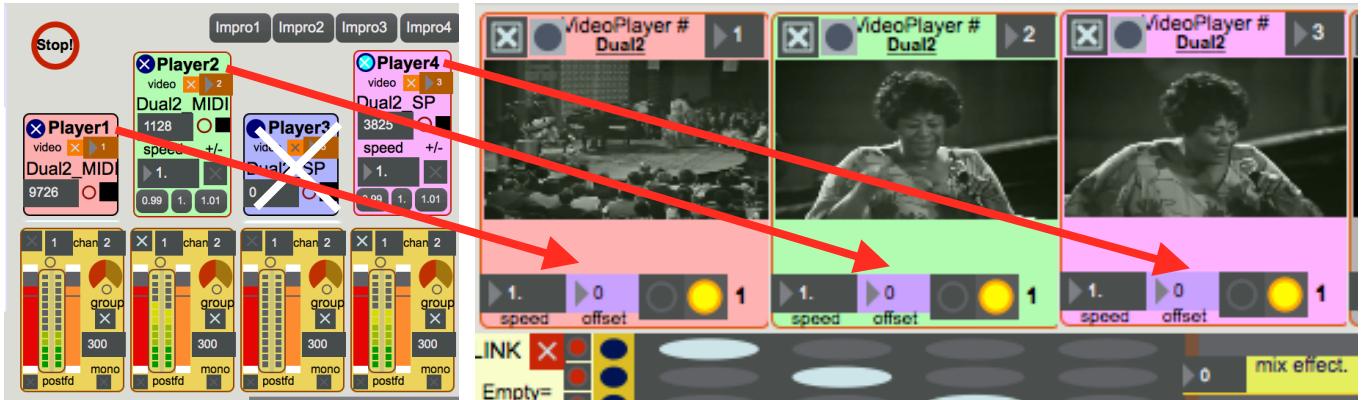


- OMax :
4. Also on the To_video patcher [Computer_Audio], turn ON the blue video toggle (otherwise you don't transmit anything). The **Connect?** messages should be changed to the default session name – **Dual2** –, as written on the OMax window. If it does not work the first time, do it again (OFF and ON)



5. Turn ON Audio [on Computer_Audio]. A good tip is to do a reset on OMax when everything is connected.
6. Turn on the video on the **qmetro Start/stop** large blue toggle, left of the OMaxVideo acquisition window (the orange-colored rectangle).
- The orange led Film Record_Enable starts to blink.
 - If you press the Film Rec_En toggle (brown), a dialog opens asking for a place to save the movie, with a default name like Dual2_0. By allowing this name (or a new one), it will be displayed on the yellow band; the orange led will stop blinking and the left led of the upper led panel recording will become green.

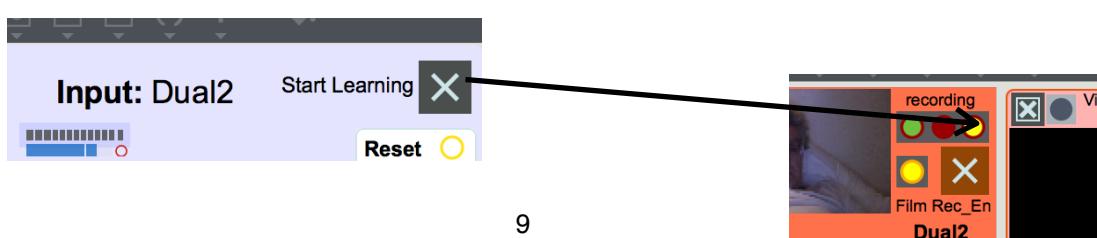
7. Enable the video toggle of the OMax players you want to see on screen. Make sure the video players have the same channels (change them if necessary). When enabling the OMax player, they change to the same color as the players. In the example below, the Player3 from OMax is *not* connected to the video system [of course, we can hear it], and Player4 is connected to Video player #3.



8. By default, the built-in iSight camera works. That should go for a quick start. Otherwise, open the patcher *AcQ_in* (top left in the orange part). You get inside all for finding the right camera and opening it as a grabber. The camera image should appear on the screen. If you don't see the right image (not the external camera image but the iSight), either don't bother for the moment, or open the subpatcher *image_control* in the orange rectangle and play with the input menu. If it does not work, take a look at chapter VI, Tutorial 3
9. Prepare the video playback. For a quick start, we decide for only one screen, that is, we use one video track on the entire rendering screen, with no divided screen. Modify the 4 into 1. The matrix goes from 16 column to one single column.

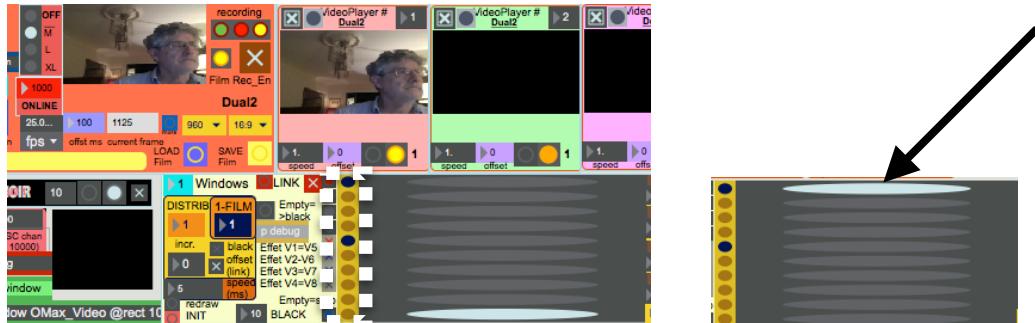


10. Turn ON Start learning on OMax [on Computer_Audio]. The right led of the upper led panel recording will turn yellow.



11. Start the recording: the middle led becomes bright red.

Then, play. Press one of the Players (for example, Player 1). Blue points will appear on the lines 1 and 5, respectively corresponding to the image made by player 1, and the same image with effect (line 5): this mean that these two tracks are available for the rendering window. By checking one or the other, the image will appear in the window.



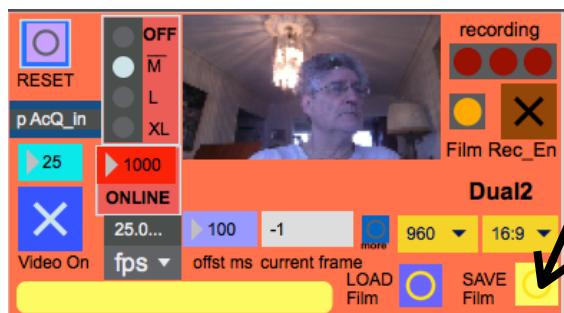
Improvise with OMax.

Normally, the video should be improvising in sync with the sound. With the matrix, you can decide which track is supposed to show on the screen.

When you are done, if you are happy with your session and want to save it:

12. Save your audio session [on Computer_Audio].

IMPORTANT: when you are done, DO NOT PRESS RESET on Computer_Audio before saving the video. It does save the film and a corresponding synchronisation table file.



13. Press **SAVE FILM** on your acquisition window (yellow button bottom right) [on Computer_Video].

Your film is actually written on disk, as is the synchronisation buffer is written, with the same name as the film, only with fnb.pat at the end. These names should *not* be modified: only, the name of the film and of the fnb can be modified simultaneously, as long as they stay the same except for their ending.

When the film is saved, the orange led Film Record_Enable starts to blink again, and you can resume at point 5.

14. You can press RESET [on Computer_Audio] and start a new session, or reload the former one.

IMPORTANT: with the actual version, you cannot add to an existing film: therefore, a saved film is a definitive one, usable only as an archive. Once the film and the oracle are saved, they can be replayed at will, but improvising on top of it is possible but cannot be saved as a video.

IV- Initialization and performances



Here are the default values for the video:

- frame per second: 25
- color (4 planes) 960 X 540 video
- length of the video buffer (called matrixset in jitter): 1000 frames (approximately 40 seconds).

These values seem to be OK for OMaxVideo running on a single relatively recent Macbook Pro (as on March 2018). However, they do work with *two* instances of Max7. For safety, a version in black and white (one video plane) is proposed (but not really sustained). In the example above, with three video tracks running on the same film, the speed sometimes slows down to 16fps. The open GL version, however, gives much better results.

IV-1 The Double-recording Trick

The length of the video buffer could look surprising. How could OMax Video work if you play more than 40 seconds ?

How does it work ? The new thing in recent OS and Max7 is that now a film can be read on as it is being written (with avf engine). Therefore, in parallel to the video buffer (jitter matrixset), the film is written on the hard disk. The matrixset is recorded as a circular buffer. The current frame written on it is the same as the current frame of the film, but modulo the size of the matrixset (default 2000). If you currently receiving frame #5346, the 5346th frame is written on the disk file and simultaneously on the #346 matrix of the matrixes. Why? Because when reading the disk file, it is loaded in memory with its current size. So the last recorded frames are missing. If they are needed for playback, they are read from the memory buffer (the jitter matrixset).

The setup for recording is :



Time: ->

now

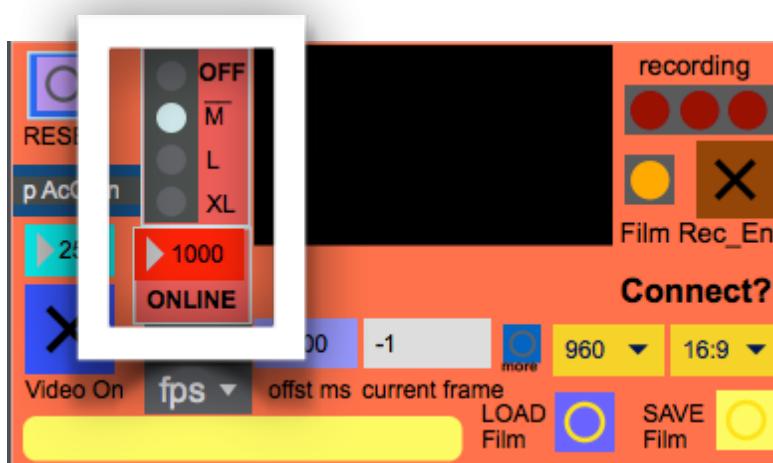
For playback, the reading of the film is updated, when it misses more than `size_of_video_buffer - 500` frames, by default 1500 frames. It is also updated when the recording process is stopped. Playback is achieved either by reading the film, or, for the last frames, by reading the matrixset. Each time, you restart recording (by pressing the *reset* button) a new disk file is recorded. These disk file are, by default, called **Dual2_0**, **Dual2_1**, **Dual2_2**, etc. Generally several tries are made for setup before the real session begins. It means that these useless files will have to be disposed of after the session. The film corresponding to unsaved oracles cannot be used with OMaxVideo anymore.

IV-2 The Importance of Online Recording in Your Session

This last version of OMaxvideo only works with Max7 64 bits. For former versions of the program, we recommend version 208 (although it is no longer supported).

In some instances, online recording will NOT be used. Therefore, the matrixes and the grabber are useless. In other extreme instances, the session will perhaps be relatively short, but the online performance is paramount.

OMaxVideo is designed to support all these situations, as there are four possible choices, visible on the Acquisition(orange) part of the window.



There are precisely four presets, which can be seen in the framed part of the window:

1. *Online* OFF. The acquisition grabber is off and the matrixset is on 50 (unused) frames.
2. Medium (default). The set is on 2000 frames. This should work on most situations.
3. Large (a lot of real-time online capture). The set contains 6000 frames (this would correspond to four minutes of video at 25 fps).
4. Extra-Large: same as above, with 9000 frames (six minutes of video).

There is less reading from the hard

disk. However it is taxing on the memory

4. Extra-Large: same as above, with 9000 frames (six minutes of video).

The size of the buffer can be set by hand. In case of Online use, *a value less than 500 would cause the computer to crash*. The default value is recommended: however, larger values are practical if online images are paramount and if there is a lot of RAM on your computer (this is often the case with a two-computers setup).

IMPORTANT (but obvious): all modifications should be made *before* the beginning of the session.

IV-3 Memory and Performance Considerations

Having a comfortable disk size available and a large quantity of RAM is practical; the use of two computers, with one dedicated to video, is a definitive asset. Two average machines are more efficient (and secure) than a top computer alone.

Another solution, not to be forgotten, is to reduce the basic fps (frame per second) rate. 20, even 15 fps can be enough in a live performance situation, or to use black and white one plane video [not supported anymore, but a relatively easy hack]. Of course, smaller images can do the trick.

IV-4 Modifying Default Values

IMPORTANT (but obvious): all modifications should be made *before* the beginning of the session.

1. Changing the fps value. The default is 25, but the frame rate can be chosen between 1 and 50 fps. A lower fps can help to get better performances.
2. Modifying the size of the frame is another solution.

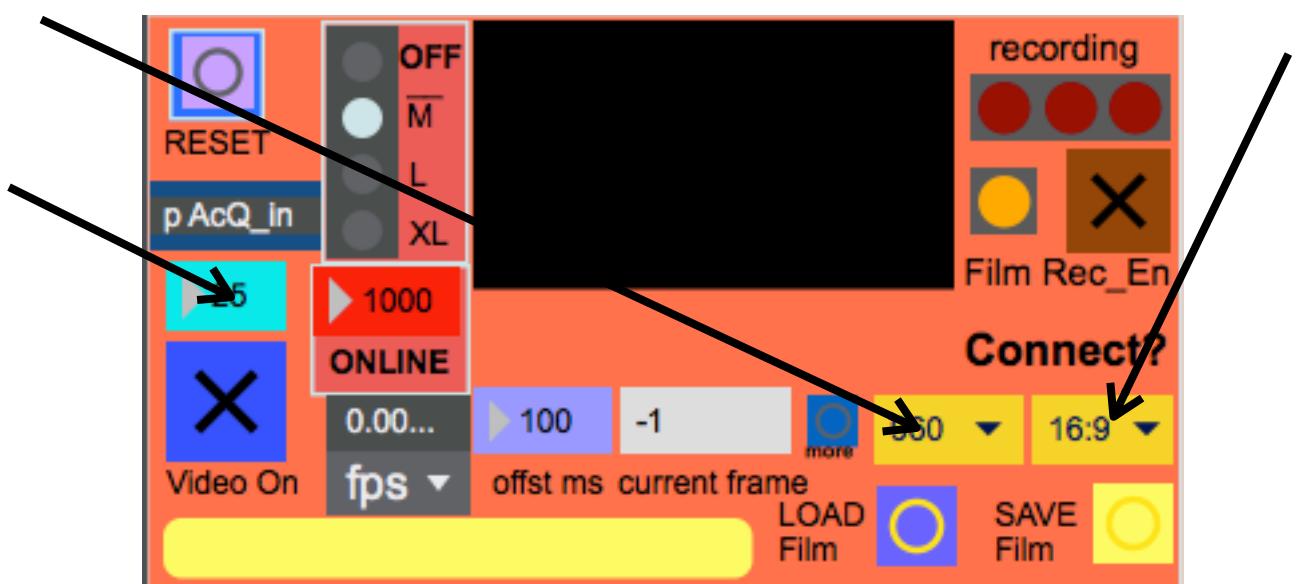
The image size can easily be changed through two menus :

- either 4:3 ratio, or 16:9 (default)
- width of 1024, 960 (default), 640, 480, 320, 160, 80

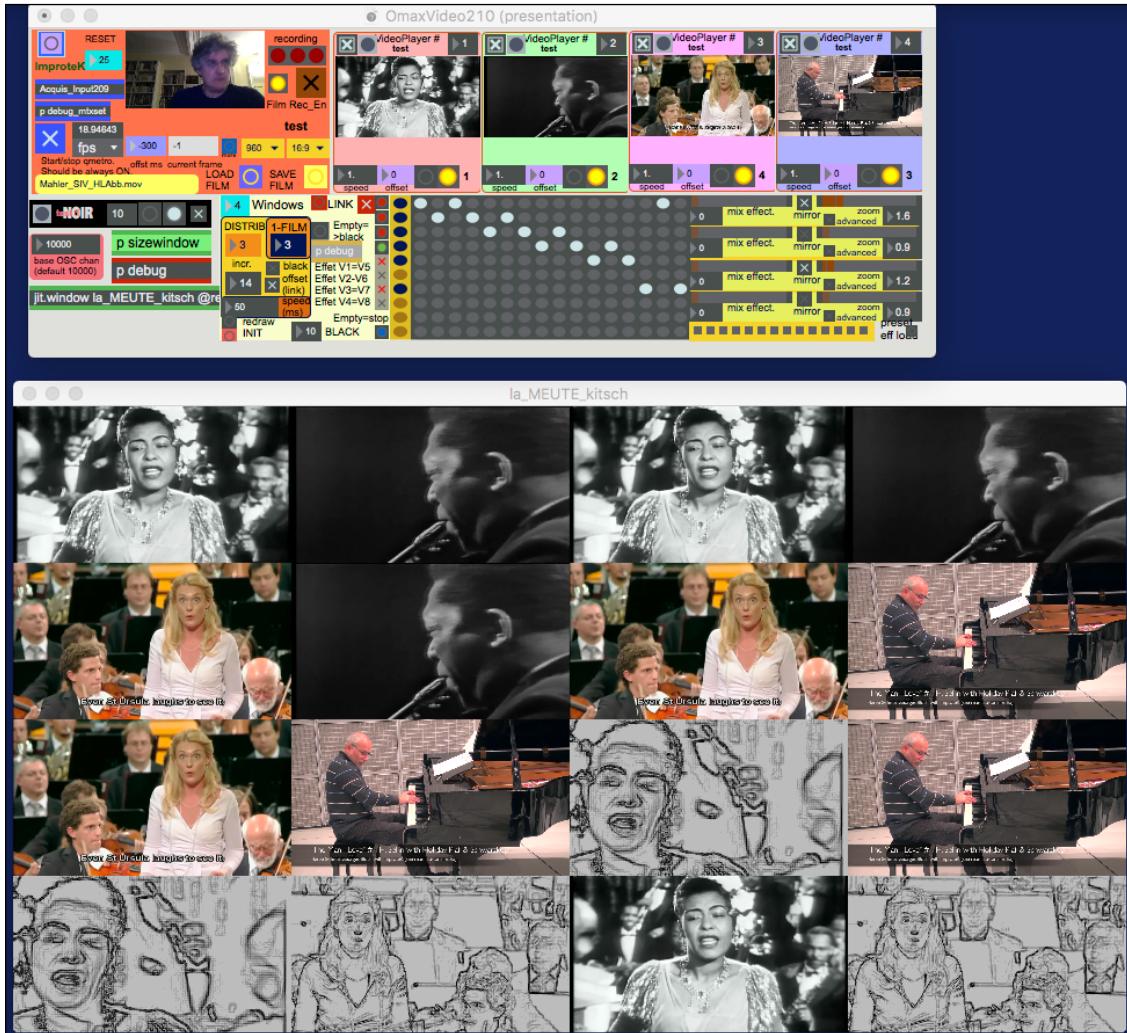
The corresponding sizes are:

In 16:9, 1024X576, 960X540, 640X360, 320X180 160X90 80X45

In 4:3, 1024X768, 960X720, 640X480, 320X240 160X120 80X60.



V- Tutorial 2: OMaxVideo Offline Quick Start

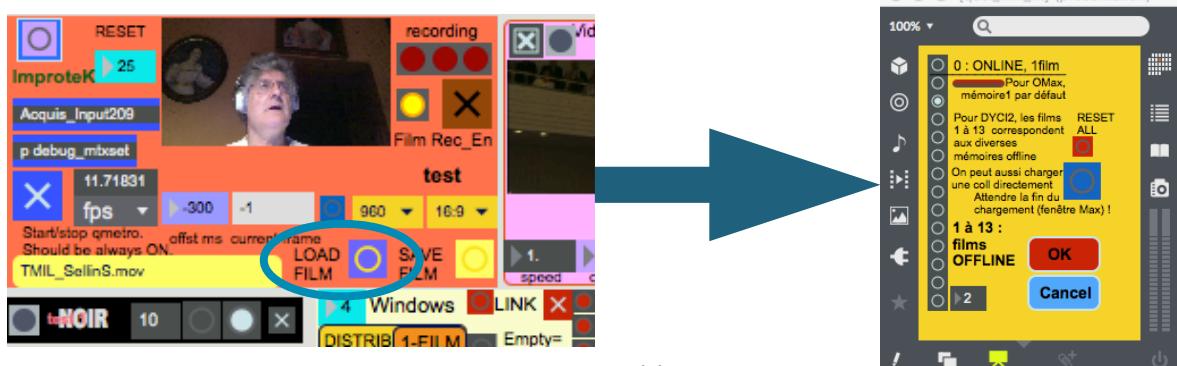


IV-1 Loading Existing Films

OMax Video has been used, especially in an DYC12-ImproveK kind of environment, with existing musical films. If you did the Tutorial 1 to the end, you actually saved an OMax file – that is, the soundfile and its corresponding structure allowing to navigate in OMax – and the corresponding film, by default named something like Dual2_3.mov, and its corresponding timetable, called Dual2_3_fnb.pat, or, more generally [MyFilmName]_fnb.pat, where MyFilmName is the actual name of your movie .mov file.

Two Loading modes.

There are two ways of loading film files. In any case it starts with the blue button LOAD Film on the Acquisition part (orange) of the OMaxVideo window:



1. First way: one by one (white dotted lines). Your click in a memory slot for a line.
 - slot 0 is reserved for *Online* film.
 - by default, the dot is positioned in the first free position. In the example shown, it means that there is already a film in slot 1, but that slot two is free to a memory. In other words, you usually *do not* have to chose a slot.
 - you then click OK, and a standard dialog box opens, in order to select the file.
2. Second (and prefered) way of loading films, by loading a whole collection of films (blue dotted lines and button).

Loaded films are set into a collection, that is, a text film containing the address of the film and their index in the list. The file looks like this:

```
1, Macintosh_HD:/Users/johndoe/Music/OMax-DYCI2/
ovideo_films/Films/ELLA_Montreux3JP.mov 328737
8169;
5, Macintosh_HD:/Users/johndoe/Music/OMax-DYCI2/
ovideo_films/Essais_IK_Films/hendrix_960.mov 298422
7462;
```

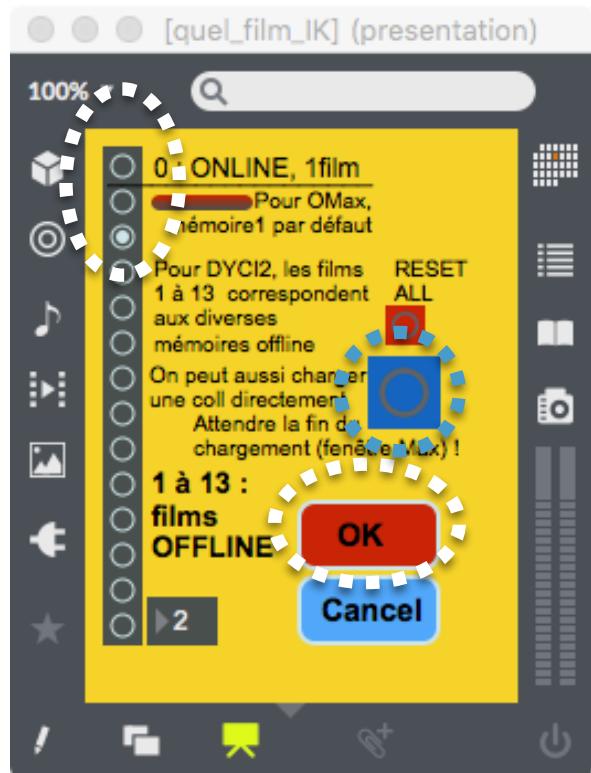
This file contains two films, in slots 1 and 5. The file path is followed by the length in milliseconds and the number of frames. You can directly load a text file containing the list of films. Conversely, you can save the actual file with the films you have loaded one by one for future use. This is done by saving the actual collection containing the films.

Saving and Remembering the Loaded Films.

There is a small red patcher on the left called [p db_film].

If you open it, you have access to the current collection of films, called #1_lesfilms.

By clicking on [write], you can save it the wy you want (name, place). By double-clicking on the box containing the collection, you can examine the actual list in memory.

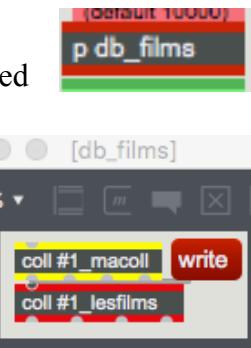


IV-2 OMax with an Existing Film: Quick Start

We will use the ELLA_Montreux film, currently in slot 1.

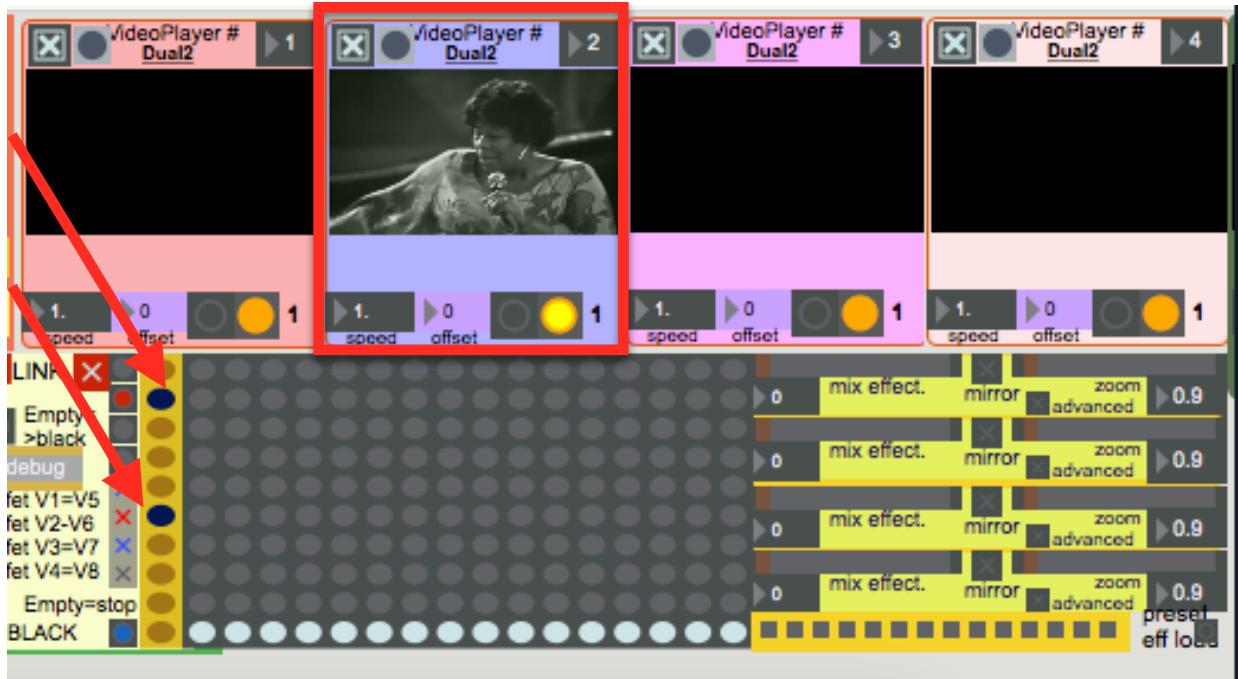
IMPORTANT: in order for OMax to take into account the fact it is using an existing film. OMax has to be reset *before* loading the new oracle. Therefore, OMax refers to a loaded (referenced) film and not the online film (which is chosen by default).

1. Reset OMax, Load the oracle Ella-Montreux.
2. Verify the connection between OMax and OMaxVideo. A good tip is to undo-redo the connection from OMax (this resets the connection). Connect also the video players and verify the connections with the Video players (same as Tutorial 1, part 7)
3. Load the film ELLA_Montreux, or a collection containing it. For the sake of the demonstration, we will assume that the film is index 1 of the collection. On OMax, set the value #1 on the To-video orange box.

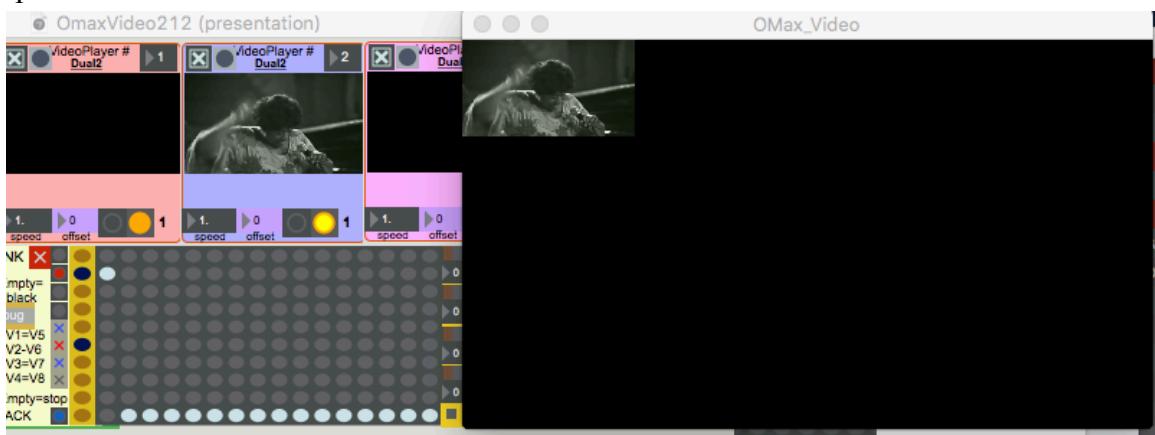


Actually, it is the default value for Offline films. But it can be set to a different value. You can also chose to connect a completely unrelated film... as long as it loaded in the collection.

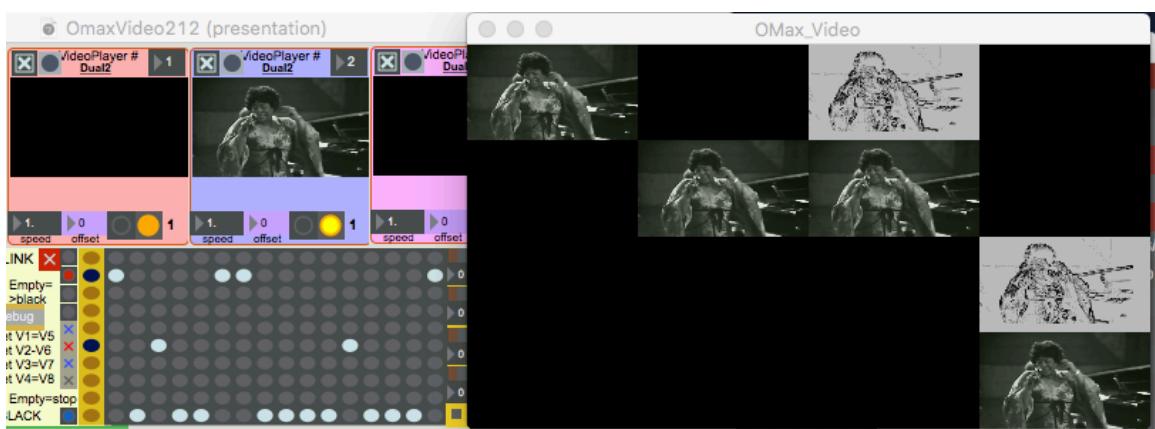
4. Play one voice (let's assume it is the voice corresponding to VideoPlayer #2). The film will appear in sync in the corresponding VideoPlayer window, and lines 2 and 6 of the matrix will be checked in blue.



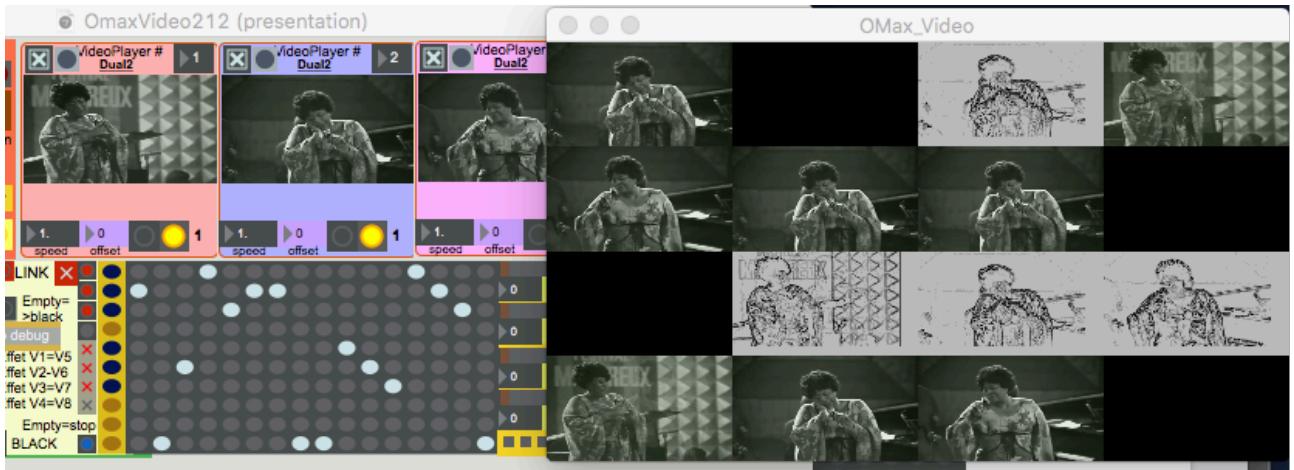
5. The checked line 2 corresponds to VideoPlayer2 as it is, and Line 6 (that is, translated by 4) is Videoplayer with effect. If I check the first dot of line 2, the image will appear in the left upper corner of the rendering window. Here we use a relatively small rendering window to put it on top of the matrix



6. Several boxes can be checked: corresponding videos will appear.



7. Finale, any combinations of tracks and rendering positions can be used.

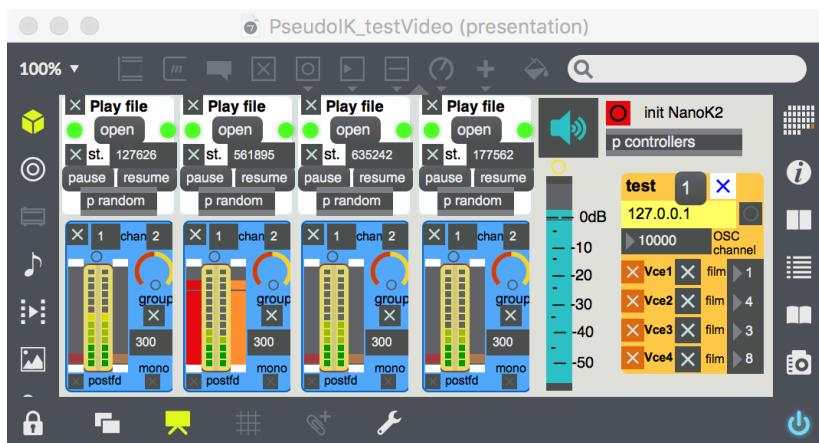


Note that:

- the dots on the bottom line ensure that the non connected boxes are black.
- if you put them on the 9th line (bottom but last), the image freeze

Enjoy...

IV-3 Simulation of Offline Playing: Using OMaxVideo With Another (Any?) Musical Program



Basically, OMax is a single sound source program. Most other audio software are audio sequencers-like, and actually rely on the mixing of several sound sources (this remark applies ago Improtex and DYCI2 as well). Therefore, for single computer testing, we designed a simple application, PseudoIK_testVideo. It just performs random editing on sound files. If these sound files are OMaxVideo ready, that is, as we said earlier, if there is a *precise* table of correspondance between this sound and some images – the famous « funbuff » file, the corresponding image should appear on the screen. But, contrary to the example of the preceding chapter, several different soundfiles and films can be used.

PseudoIK_testVideo is a bogus program, giving a starting point for people wanting to connect their own realtime sound editing program to OMaxvideo. Its most important part is the yellow one, which is very similar to the actual DYCI2 (or Improtex) transmission module. It receives messages of track, date, and speed, and send these to corresponding tracks in OMax video. Since it is a test program, generally the OMaxvideo players are set on tracks 1, 2, 3 and 4 respectively.

In order to have a transmission working, you should:

- set the address of the Video computer (in that case, it can easily be the same, and one single instance of Max can be used)
- accept the transmission for some given voices (brown toggles on the left part)
- decide on a corresponding *loaded* film for each track (on the picture, these are films 1,4,3 and 8 for video voices 1, 2, 3 and 4). The films can be changed during the performance. We'll notice

that this is generally decided by the computer running the sound (OMax, DYCI2 or any other), and will be generally accomplished in an automatic fashion.

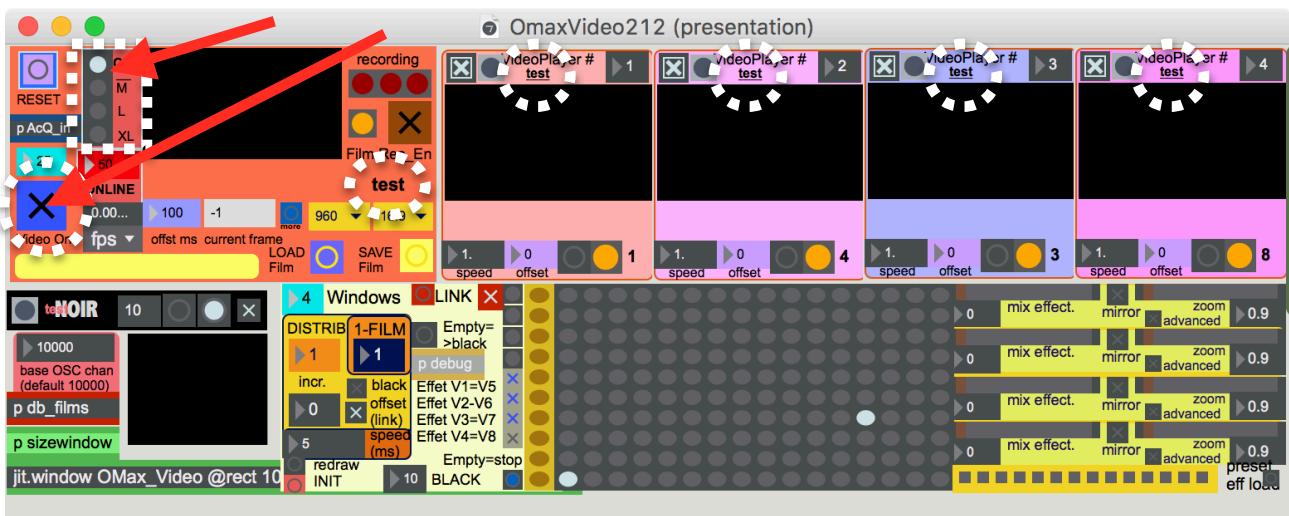
- the sub-patch random allies to simulate a random edit (by default every 4 seconds).

Linking the program and the video computer



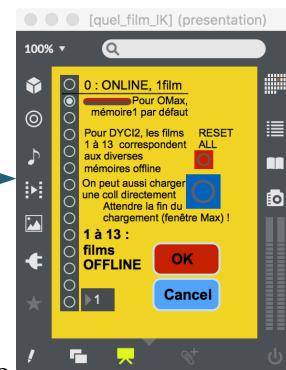
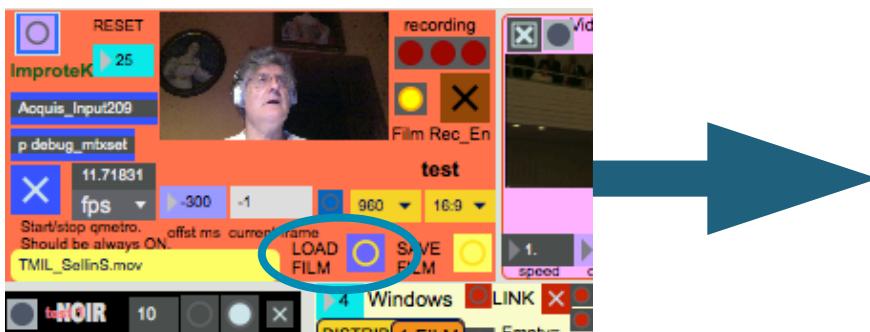
In order to link the sound and video computer, you should:

1. first launch OMaxVideo
2. on the sound computer, set the address of the video computer (by default, 127.0.0.1 since one single machine is used)
3. Press the [1] black button, then the white toggle. The Video computer should display the name of the window (here **test**). Otherwise, *Connect?* will appear and the link is inactive.
4. for this exercise, we suggest you stop the video grabber on the Video computer, by setting the online matrixset preset on OFF
5. Launch the Main blue video metro.
6. Set the numbers of the films and activate (some of) the voices. The Video Players should get coloured.



First try

- Open PseudoIK_testVideo, and OMaxVideo. Connect them as explained above
- in PseudoIK_testVideo, open one sound file (with open in the first Playfile white box)
- In OMaxVideo, load the corresponding film, as explained above . You can use *any* slot, except the top one; the second from the top corresponds to film 1



- Check that PseudoIK_testVideo is really connected to OMaxVideo. (You can press again 1 and the toggle, as explained above).
- play the sound file, possibly simulating an edit with the random box.

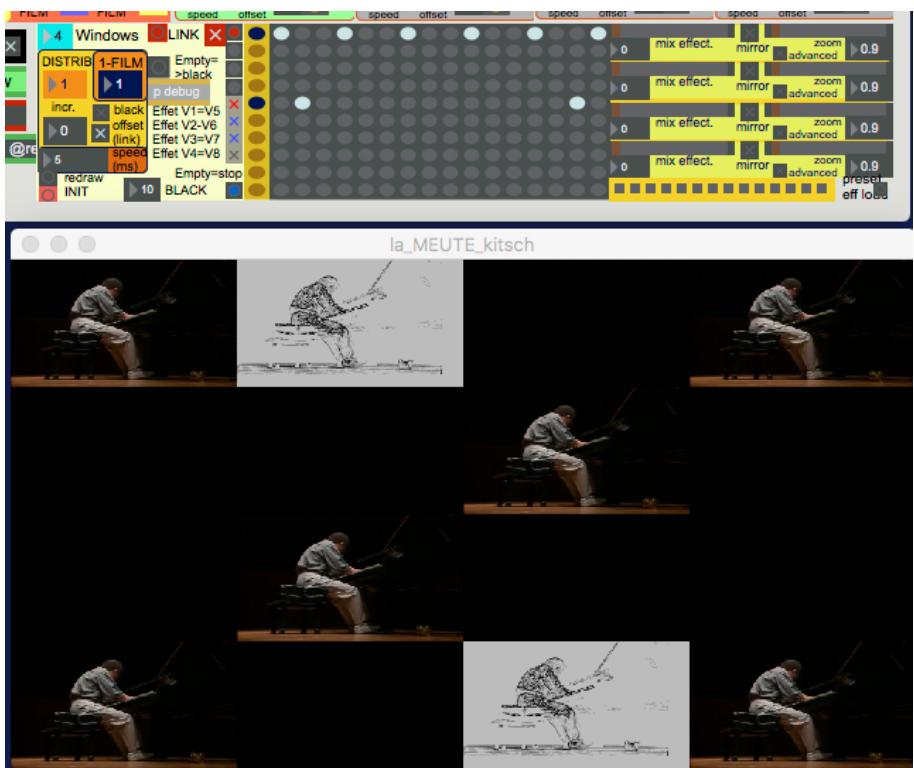
Several things should happen

1. The image of the film appears in the first video player. The film number appears in the bottom right corner (number 6 in the example)
2. The VideoPlayer track is coloured, the color varies according to the film number. When the sound and the film run, the yellow light of the player becomes bright yellow
3. the first red led of the matrix becomes bright red, showing that the video is playing.
4. In most cases, (when the toggle LINK is on), two blue points will appear on the render-enable yellow bar, one on the first rank (corresponding to the film), one on the fifth rank, corresponding to the modified film



You are now ready to send the film one to the render window using the matrix. In the example below, we have a screen divided in 16 parts, and the original film is sent to sub-screens 1, 4, 7, 10, 13 and 16; The same film, transformed, is sent in sub-screen 2 and 15.

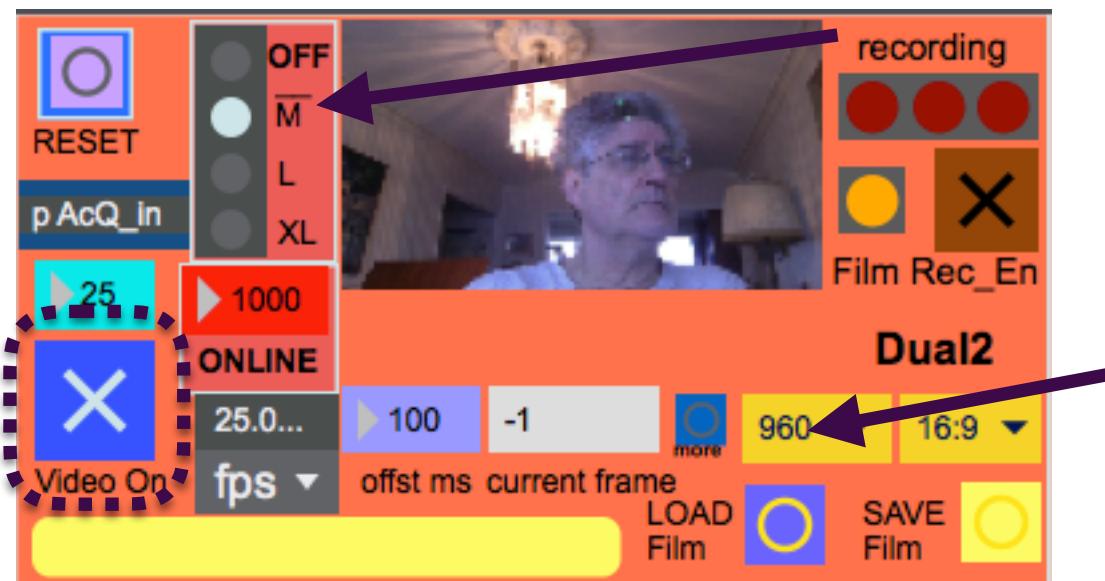
IMPORTANT (but already said): by unchecking the points of the matrix, the image freezes and is NOT replaced by a black screen. You need to check the last rank (10) to replace the image by a black screen.



VI- Tutorial 3: Acquisition

VI-1 Setup

- You need a camera (or webcam) going into the computer with a USB or Firewire cable. A webcam or the iSight can be used.
- The picture size is by default 960 by 540 pixels, for a different size change the default size in the (orange) acquisition window.
- However, as always, the optical quality of the camera matters, and its capacity to adapt to different kinds of light.
- You can even plug different cameras and choose from them. Since it takes some time to switch, you must check it before the performance.
- by default, the grabber (system allowing to grab the video stream on the computer) is automatically opened when loading the program. The program looks for any available camera (most often, the iSight inside the computer) and grabs its video stream. The grabber is closed when the ONLINE radio button is OFF (top position).



WARNING: Don't forget to TURN ON the **Start/stop** metronome button! The image from the camera must appear on the screen-window of the acquisition rectangle.

VI-2 Controls

As said above, the acquisition patcher is the orange rectangle to the left of the OMaxVideo window: An important subpatch is the [p_Acq_In]. It allows to set the incoming video stream as a grabber (to grab the video into the computer). It also allows to chose the camera. It includes:

- The *image_controls* setup. By double-clicking into the patcher box, you open the actual control window (arrow on the picture)
- a subwindow *inVid* allows to use a film as input [See Tutorial 6, chap. IX-2]

On the orange acquisition part, we already saw the **Start/Stop** button for the video metronome [Tutorial1, chap. II] **Record Enable** button. The window also includes several displays:

- one display shows the current **fps**
- another shows the last recorded frame (**actual frame**) of the current movie. The actual frame of the matrixset never goes above the shortened length of the matrixset. (2000 by default), but the movie can go for a long time: we tested recordings above 60 minutes (the hard drive size should match!). If the matrixset length is 2000 frames, at 25 fps, this means

it lasts 80 seconds. Then it restarts on matrix 0 (which is the movie frame 2000), and so on... the last 80 seconds are available by an indication of the frame number modulo 2000.

- a value for the **delay** of the Video recording. The audio is recorded with 400 ms delay. There is an actual video delay of the same amount (with a circular buffer), allowing not to miss the first images of the performer. But, due to the equipment, some more adjustment could be necessary. You can move the film forward or backward in the playback by adjusting the date associated to the frame. (The image is more in advance with a positive value). Generally, a value of 0 is fine, but some fine tuning is necessary before the concert and this can vary quite a lot with different equipments.
- The SAVE FILM and LOAD FILM button have already be seen in chapter VI.

RESET

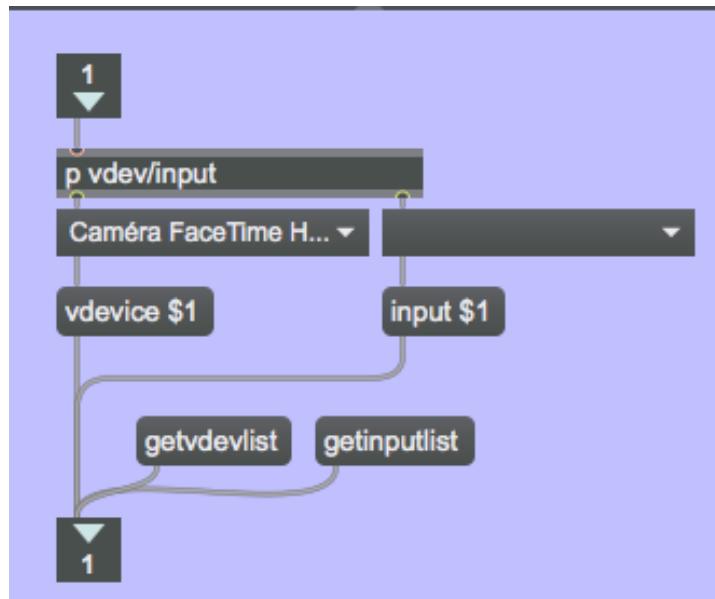
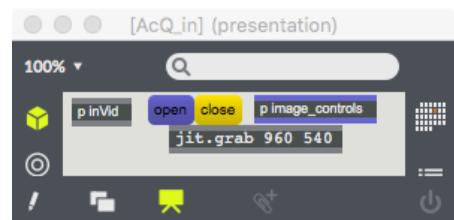
This is generally made from OMax: by making Reset in OMax, it reinitiates everything, erase the films and so on. It is mandatory to do it at the beginning of a session, but it is better done from OMax. This button does not exist in DYCI2 and ImprotEK: it could be preferred to reset between different sessions.

VI-3 Camera Selection

A double-click on [p_Acq_In] gives access to the grabber.

You get the button to open and close the grabber. If, when opening, the program choses the wrong camera, you have to close the grabber, open *image_control* to chose the right camera, and reopen the grabber again.

When you open the *image_controls* subpatch, you actually a generic jitter help patch with very little modification.



In order to get the right camera, you have to click in **getdevlist** and **getinputlist**. The list of available Input will be printed in the max video. By choosing the right input (iSight on the above figure) and device (if you choose USB video, your Firewire camera risks not working even if it is selected), you get the right picture in the acquisition screen.

When the right camera is chosen you reopen your grabber. The image should appear on the acquisition window, allowing you to make eventual controls, if available. VI- Tutorial 3: Acquisition

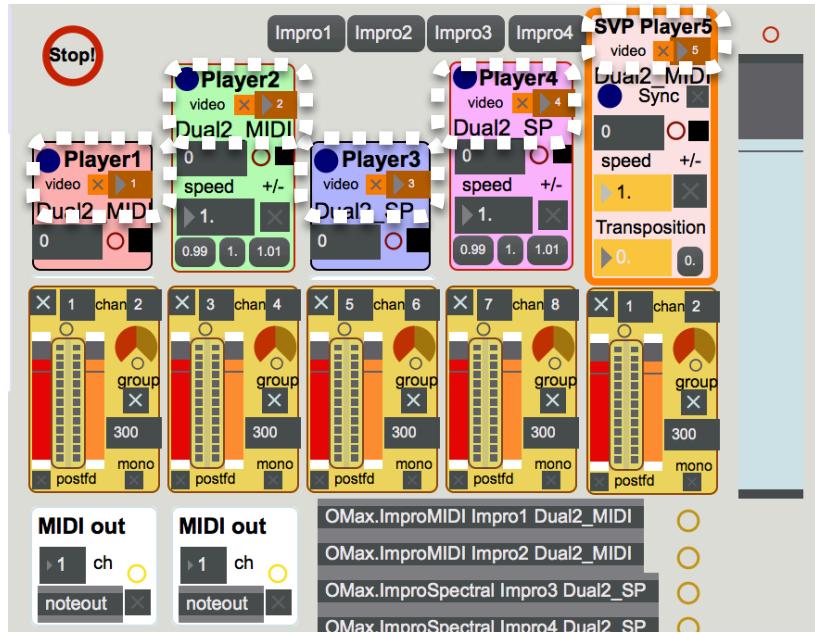
VII- Video tracks

VII-1 A Video Track and its Corresponding Audio Track

On the general setup, each of the video players is assigned to an audio track. By default, the five audio player are directed toward video players of the same number. The default version of OMaxvideo comes with four players – or even three and a PisteTriche for projecting films without any sound index. It is relatively easy (max programing at medium level) to add two missing tracks and, if you use two computers, it is really feasible. Very often, however, no more than three video channel are required, and each OMax track can easily be switched to any video track. With one single computer, limiting oneself to five video streams (one in and four out) seems more reasonable.

Enabling Video Tracks

Actually, in no way is it required to have each audio player actually connected to a video channel. It can be unnecessarily expensive, especially for audio tracks which are never or rarely used. Another case is, for example, when the SVP track of OMax is constantly used in sync with another track for harmonizing effects: the resulting image will be the same, and calculating it twice is really unnecessary. More to the point, rarely are so many video tracks desirable, even if for multi-screen or divided screen effects.



The video connection can be disabled from Computer_Audio; from the start or on the fly, it can be decided to deal with much less video players than audio players.

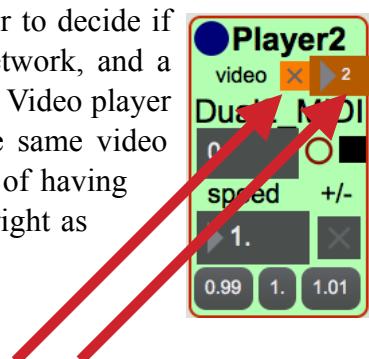
On OMax [Computer_Audio], each player contains an associated program in charge of sending the audio dates to OMaxVideo. There is a toggle (orange_brown) in order to decide if the information of the player is actually going to be sent on the network, and a number box to decide to which video player it is going to be sent. The Video player number is free to be chosen. In fact, if two audio players share the same video player, it will result in strange video results, but the possibility is left of having two audio players with the same video channel, and it will work all right as long as they do not play together.

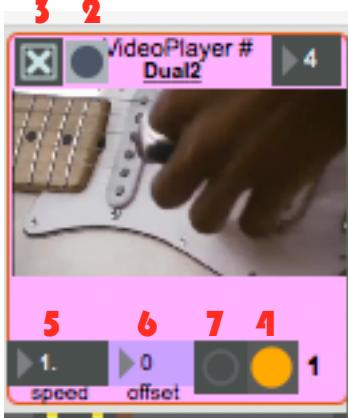
VII-2 Videoplayer reference

All video players have the same features.

This is true whether or not it is linked to a variable speed player (Player2 or SVP type):

1. Index of the corresponding audio track number. Normally this index is unique for each video player. (It is allowed to have the same index for several video players, although it is perfectly stupid, since it means several loads and edits of the same film for the same result).. These are initialized as the #1 (first argument) of the *VideoPlayer* bpatcher. When opening the unmodified patch, they correspond to Players 1, 2, 3 and possibly 4. However, for all normal video players, the number can be changed on the fly. Only the highest numbered player must be the special BLACK player: its value (10 by default) should *not* be modified.





2. Orange diode (when ON) showing if the player is currently playing on the rendering window (see Tutorials 5 and 6)
3. Receive enable toggle (check to receive video). Generally ON. Uncheck if the video stream is too heavy on a track on which one does not want video.
4. Video-playing light (yellow). Is lit if the video (and corresponding audio) is actually playing
5. Speed of lecture; the default value 1. is automatically set by the corresponding audio player., in order to correspond to the audio speed rendering
6. Offset, to fine tune the offset between sound and video. Normally on zero (if everything has been done correctly)

7. Force mixing action on this particular video track: currently unused.

IMPORTANT: When you modify the video player number on the fly, the speed is reset to 1. So changing from a fixed-speed player to a variable-speed player (like SVP or Player2 type) does not change the speed of the video unless an order of modification comes from the audio program. The cutting points are always correct, but the reading speed is not. As soon as the speed changes on the OMax [more generally, audio] side, the change of speed is reflected on the video; but getting from a «normal» player to, say, a SVP player playing twice faster will only reflect the change when the speed will be sent again to the Video Part (this is one of the drawbacks of the two-computer setup). Of course, the video operator can try to set it «by ear», but this can be quite a complex task (especially with SVP)!

Like for the acquisition window, the name of your session (**Dual2** on the picture) is here to indicate that there is a correct connection. It also allows for several OMaxVideo running from different Audio programs (with different names, of course).

IX- Rendering, Matrix and Effects

XI- Tutorial 6: Cheating

XXX OBSOLETE BUT TO BE KEPT AND UPDATED

Appendix

Using an Existing Musical Film

Existing music is used in many instances (as the the « Three Ladies project »). Sometimes, this existing music has been filmed, and the images can also be used.

Musical films can easily be used, following these steps:

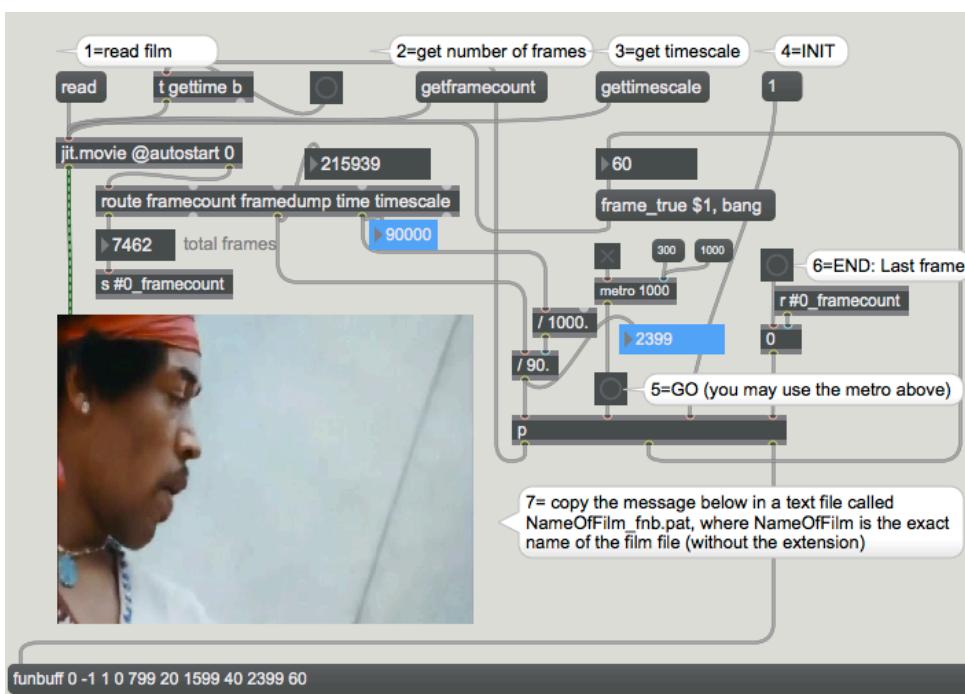
1. Extract the sound file in a Max-compatible format (free software as Audacity do just that, and can save in diff or wave format).
 2. [BETTER, BUT NOT REQUIRED] If you want to be picky, you can reassemble the film removing the sound: this results in a smaller file, better handled by the program and saving unneeded memory space. You can take advantage of this step to reconstruct the film with a constant framerate (softwares as Handbrake or Quicktime can handle this)
 3. You should get a sound file with a decent quality, and a film. You now need to construct the table making the correspondance between the time of the sound file and the corresponding frame in the film. This is done by the utility Make_funbuff_OK.

If the name of the film is NameOfFilm.mov (or .avi), the table should be named NameOfFilm_fnb.pat (and actually be in a text format).

1- Constructing a Reference Table (funbuff)

Open the program Make funbuff OK, and follow the seven steps indicated.

1. Read the film
 2. Get the total number of frame, which will appear in the left black number box
 3. Get the timescale, which will appear in the blue number box
 4. Press the 1 (INIT). The beginning of the funbuff is displayed (conventionally time 0 correspond to -1 and time 1 to 0, that is the first frame)



5.Then, press the Bang GO: the time corresponding to frame #20 appears, followed by the frame number. By pressing again, you'll have the time for frame#40, #60 etc. As it could become tedious to press the button until frame 7462, you can use the metro above, which will fill up the message box automatically and stop when the number is above the total number of frames (in this case it will stop at

7460). For each bang, the corresponding frame is displayed.

6. If the total number of frames is not a multiple of 20, press the bang END. It adds an entry for the last frame. The end result should look like this :
7. You just have to cut and paste the entire message in a text file called NameOfFilm_fnb.pat, where NameOfFilm.mov (or .avi) is the name of the original film. This tabla will be automatically loaded into OMaxVideo when you load the film (both should be in the same repertory).

