



## Mirage: Example of a Video Engine for Performing Arts

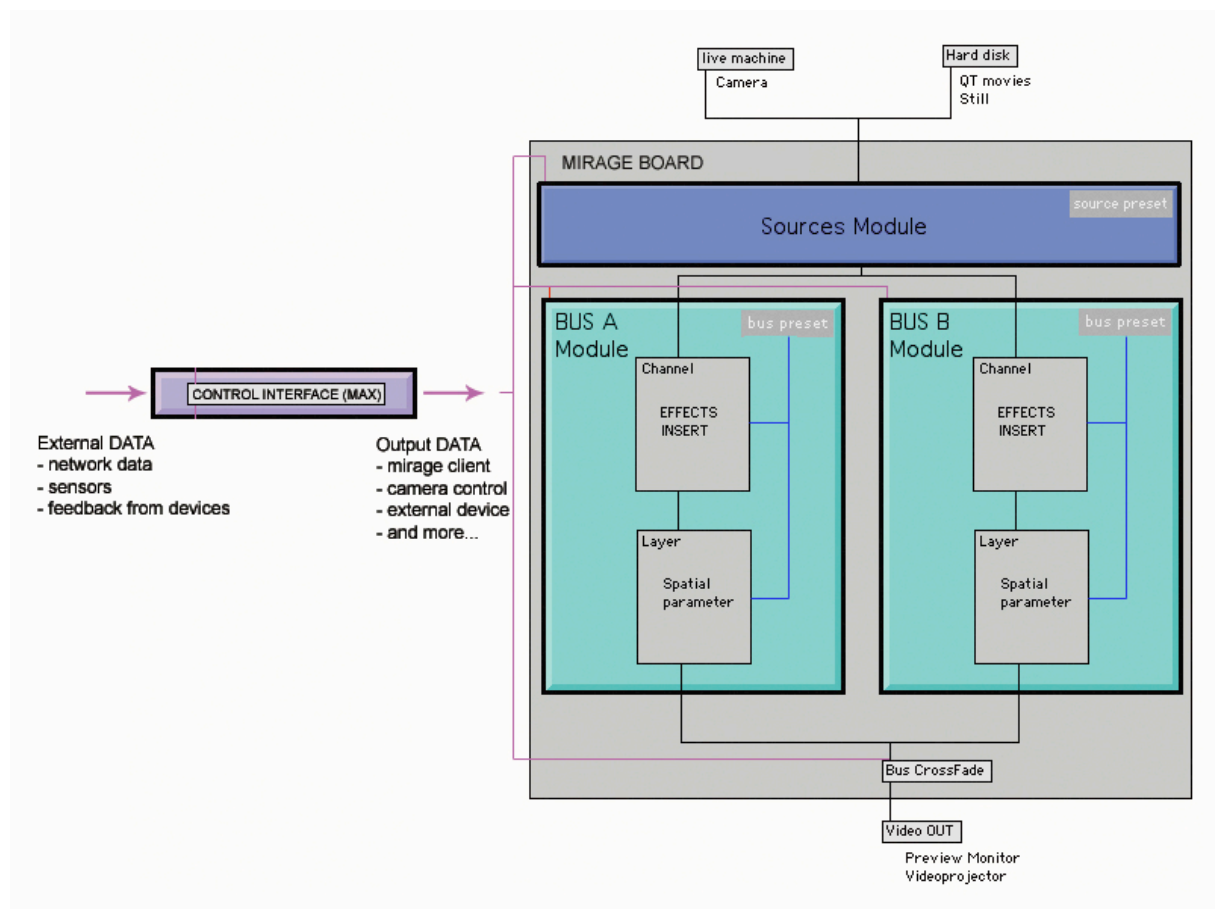
Mirage software deals with A/V broadcasting and control of real time effects on the theatre scene. It is one of the possible answer to the needs of stage productions as well as theatre workshops using multimedia technologies. Mirage has been developed with 'Max' and 'Soft VNS', library of objects on MacOSX operating system.

Shortly speaking, 'Max' is a graphic language for manipulating « objects » - i.e. small programmes that carry out elementary functions – in order to make programmes called patches. With a sound-objects library like MSP, Max can produce sounds in real-time environments ; and with video libraries like Jitter or Soft VNS, pictures. It looks like a graphic mechanical game : it arranges existing programming elements in order to make more complex ones. High-degree developers may invent their own objects in order to answer their personal needs.

The choice of 'Max' is linked to an existing community of video artists and stage managers that were accustomed with this language object through their own sound performances. The choice of 'Soft VNS' is related to its speed of live video treatment.

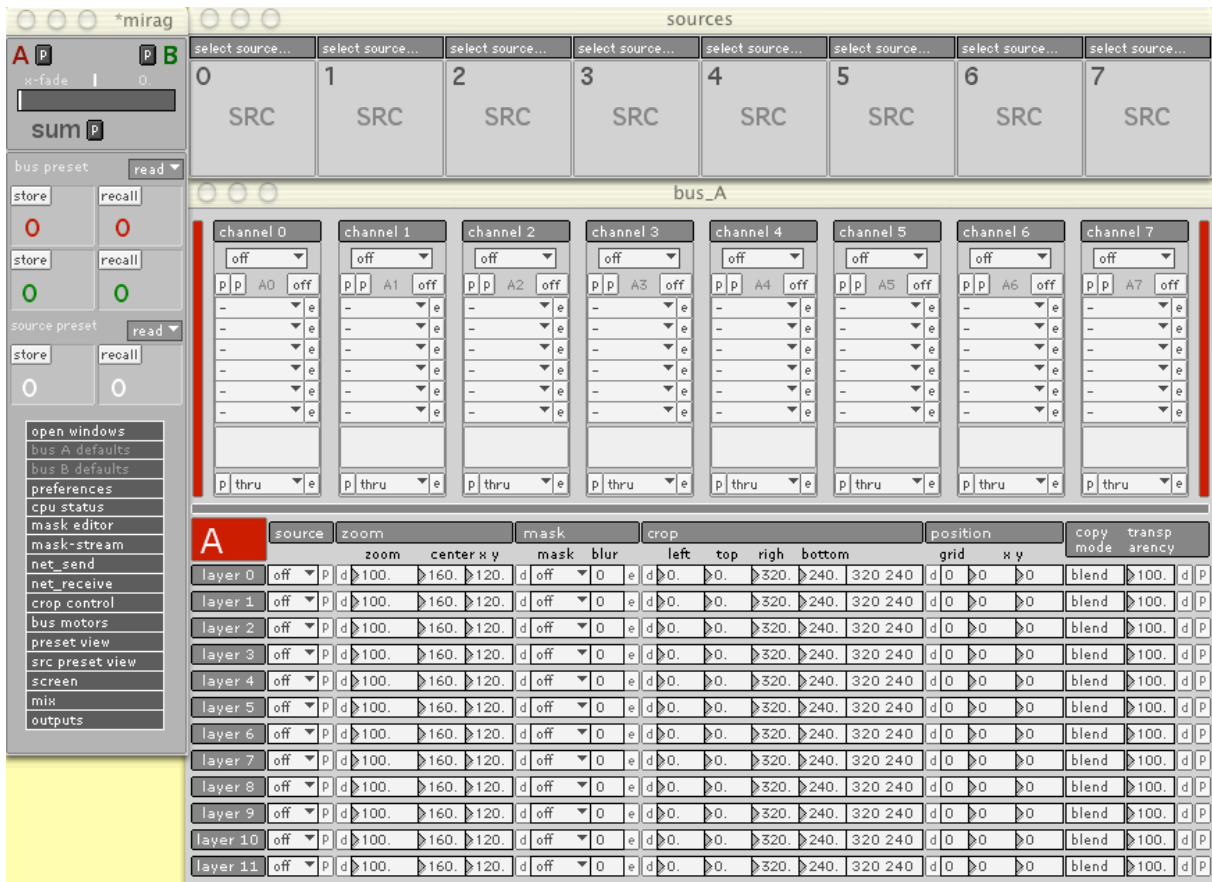
Several choices characterize the compromise between manipulating and making and constitute the principles of Mirage :

- 1) to easily command video *sources* (live or recorded video sequences) ;
- 2) to apply effects on image flows that are selected through *channels* ;
- 3) to arrange videos in space thanks to a fixed system of transfers (*layers*).



So, the Engine presents a module of sources and two bus that gather numerous effects and the spatial organization of the sources. The Control Interface controls them.

The control board looks like the following schema :



At the top, on the right, there is the module of sources that contains eight boxes : in the boxes, one can select a camera or a movie or a fixed image or a buffer or an external patch.

Then, there are two bus, A and B, that contain channels and layers. (On the image, we can only see the red bus A).

We can apply effects on the sources *via* the channels. One can treat eight sources by applying on them seven simultaneous effects on each channel.

After, one can put the treated source in space via a system of twelve layers (zoom, crop, masking, mixing, positioning functions).

Simultaneously, one can realize another combination of effects and positions in the bus B, from the same choice of sources. Sources and bus are independent each other.

Let's resume now simply the steps necessary to build two elementary video states and to manipulate them in real time, thanks to a control interface.

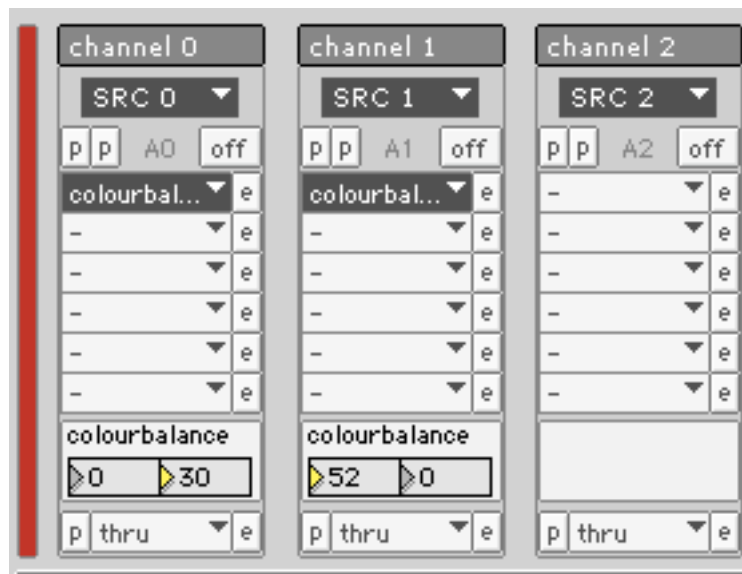
Mirage's detailed tutorials can be downloaded on : [www.didascalie.net](http://www.didascalie.net) tech section.

Mirage is a freeware. To download it and to test it : <http://sourceforge.net/projects/mirage/>.

One can control Mirage thanks to Max or Pure Data (equivalent to Max but freeware).



- BUS A channel (media treatment)

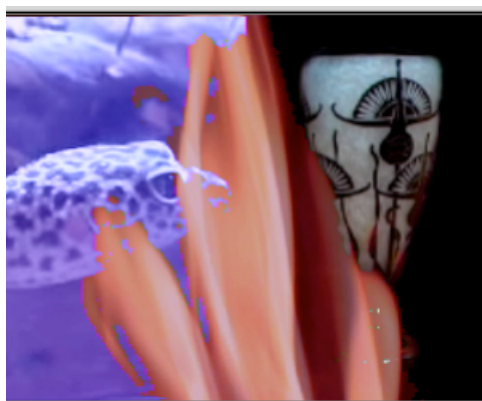


One can apply effects on each needed source. In the present case, we use a colourbalance effect, which modifies the colours of the image. One can decide to apply several effects (7 maximum simultaneously) or no effect (as in the source 2).

- BUS A layer (building the bus state)

A	source	zoom	mask		crop				position		copy	transp					
		zoom	center x	y	mask	blur	left	top	right	bottom	grid	x	y	mode	arency		
layer 0	ch_2	P d >135.	>81.	>113.	d off	0 e	d >161.	>0.	>320.	>240.	320 240	d 0	>0	>0	max	>0.	d P
layer 1	ch_1	P d >100.	>220.	>120.	d off	0 e	d >0.	>0.	>159.	>240.	320 240	d 0	>0	>0	max	>0.	d P
layer 2	ch_0	P d >100.	>160.	>120.	d off	0 e	d >0.	>0.	>320.	>240.	320 240	d 0	>0	>0	blend	>30.	d P
layer 3	off	P d >100.	>160.	>120.	d off	0 e	d >0.	>0.	>320.	>240.	320 240	d 0	>0	>0	blend	>100.	d P

For instance, on the layer 0 : we have to center the vase on the half right, to make it bigger and to cut the border on the left. In a real time process, the image experiences the following transformations :

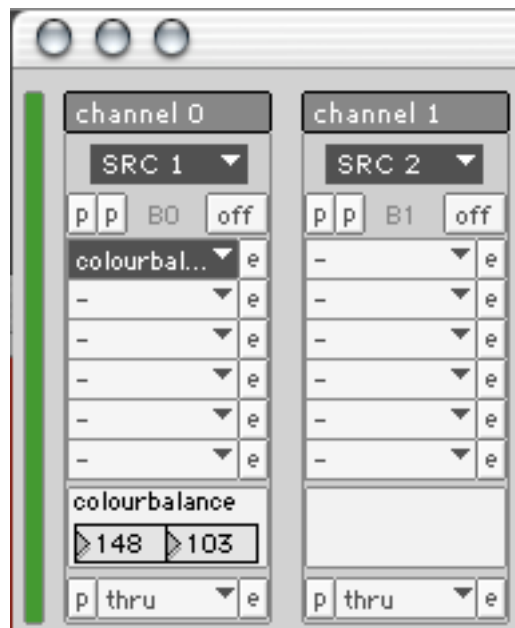


- ZOOM : 135 (enlargement of the image with a 1.35 factor) ;
- CENTER X,Y : 81, 113 (shifting of the image center : 79 pixels towards the right and 7 pixels towards the top) ;
- CROP : LEFT : 161 (the left side of the image is cut (161 pixels lengthwise) in order to keep half of the image.

Final Compositing

On bus B, we are starting from the same sources and the settlements are the following :

- BUS B (channel)



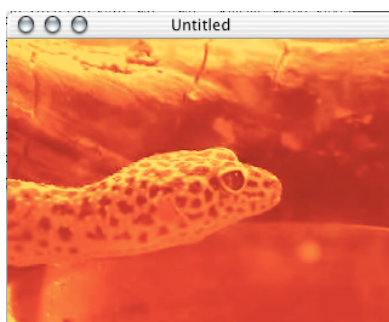
We only use two media for the new composition on channel 0 and channel 1.

- BUS B (layer)

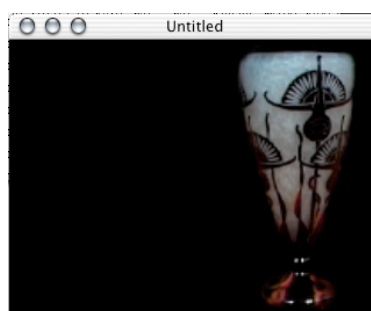
B	source	zoom	mask	crop	position	copy	transp												
	mode	center	mask	blur	left	top	right	bottom	grid	x	y	mode	arency						
layer 0	ch_1	P	d	>153. >81. >114.	d	off	▼	0	e	d	>161. >0. >320. >240.	320	240	d	0	>0	d	P	
layer 1	ch_0	P	d	>100. >160. >120.	d	off	▼	0	e	d	>0. >0. >320. >240.	320	240	d	0	>0	>0.	d	P

We only use two layers.

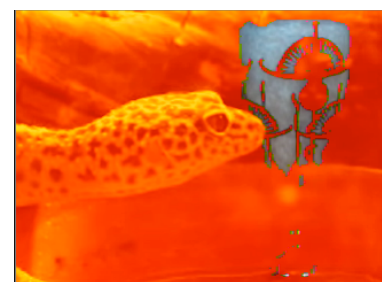
And the result is the following :



Layer 0



Layer 1



Compositing final

## Savings



Every composition of bus and sources can be saved under a number (store) and be recalled (recall) at a later date.

The files are saved on the hard drive and can be recalled whenever (read & write).

**Attention : this function is not stable.**

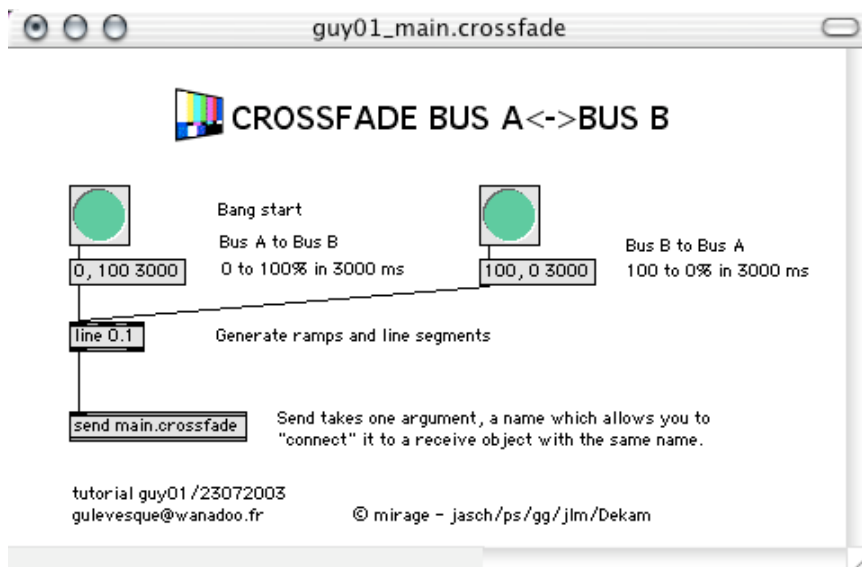
## **Control**

We have just edited two video states with the engine. Now, we can manipulate these material. We implement a programming language (we will use 'Max') to send instructions to Mirage and transform the two video states. All the parameters on Mirage board can be handled in a real-time process.

We are going to successively describe three manipulations with the control interface :

- shifting from one bus to another one ;
- changing an effect (vertical flip of the vase) ;
- changing the arrangement of layers (permuting dragon/vase).

### Shifting from Bus A to Bus B



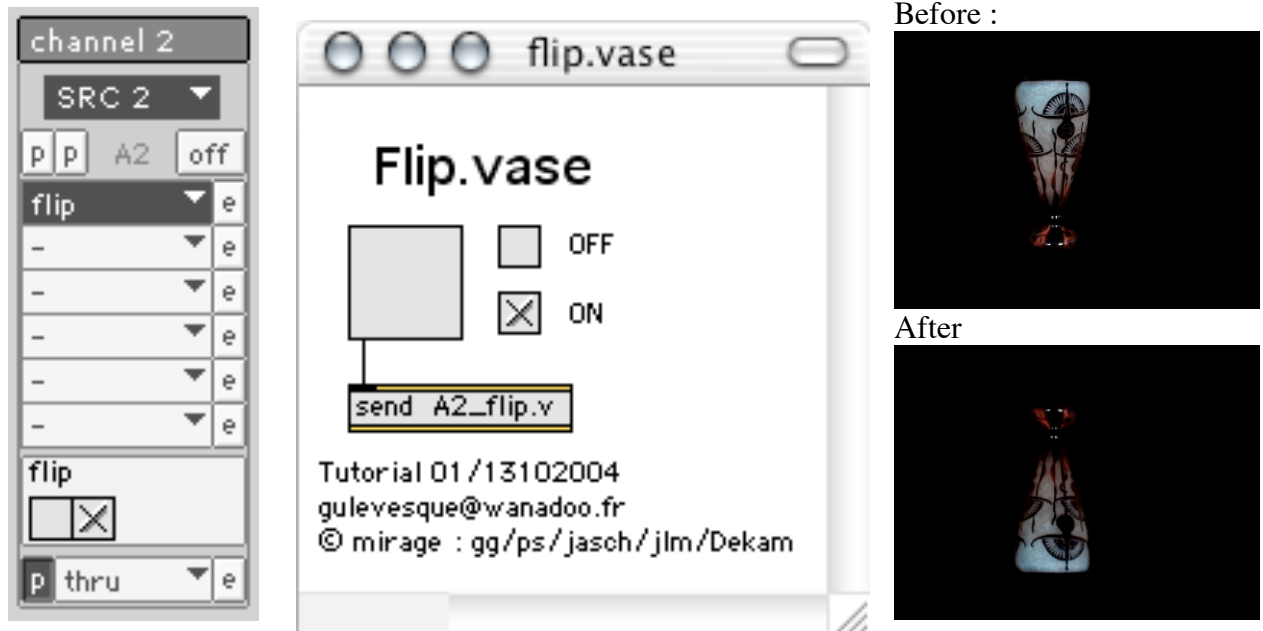
We want to shift from bus A to bus B in a definite time.

A click in the green circle on the left (BANG) launches the following instruction : a linear increase from 0 to 100 in 3 seconds.

The result of the increase is sent to the Mirage address main.crossfade : it pilots the shift from one bus to another (send main.crossfade). 0

corresponds to the bus A while 100 corresponds to the bus B. Clicking to the second bang involves to return to bus A in 3 seconds.

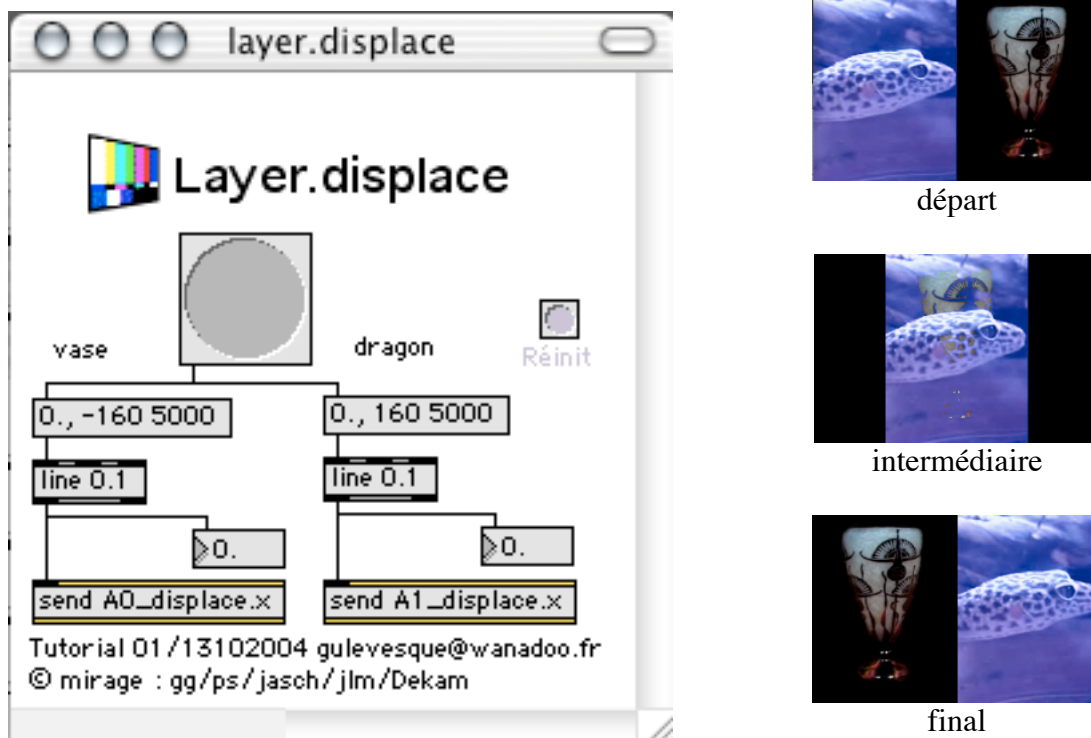
### Changing an Effect



We apply a FLIP effect on the vase on Channel 2 in bus A. The corresponding Mirage address is A2\_flip.v.

### Changes in layer compositing

We wish to permute the dragon and the vase in 5 seconds. The dragon is moving from the left to the right towards the vase is moving from the right to the left.



We use the following commands : Mirage A0\_displace.x and A1\_displace.x : they change the position of the image in the corresponding layer.

### Cue list

So, we have written three elementary patches to manipulate images.

Gathering the necessary operations for a performance in a patch constitutes a cue list.

The actor, thanks to sensors, and the stage manager, thanks to midi potentiometers, can themselves control parameters.

One of the challenges is now to manage to write conduct patches which are easily and fastly malleable during rehearsals.