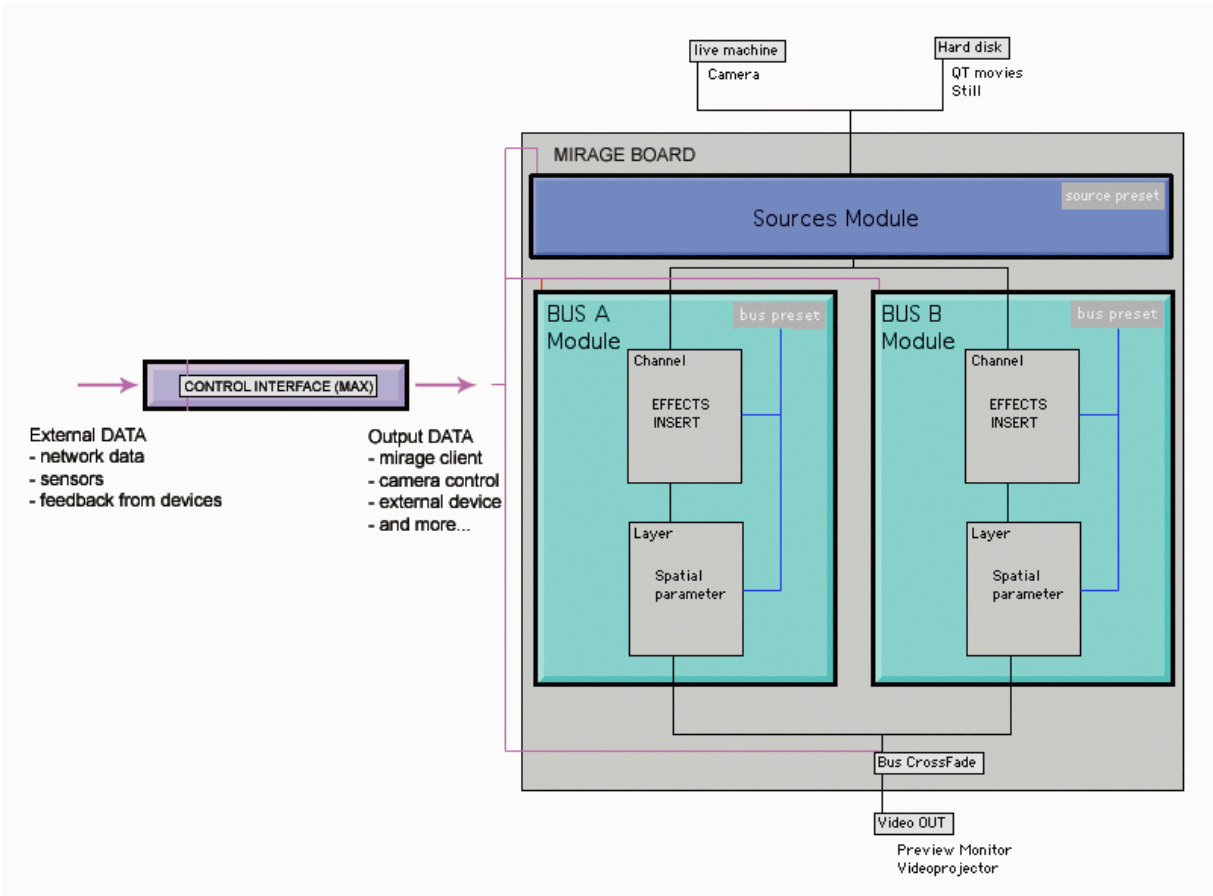
Mirage. Présentation et tutoriel de prise en main

Guy Levesque, Georges Gagneré

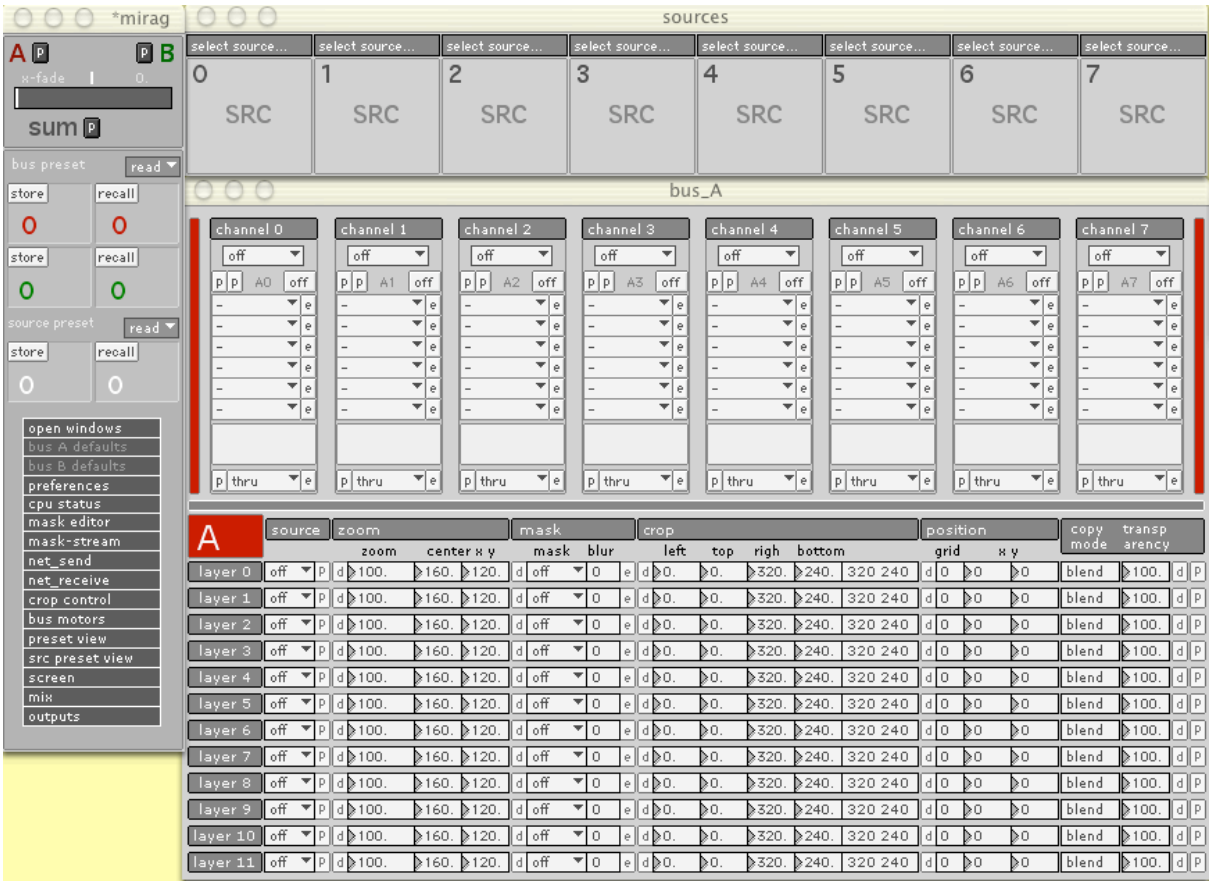
Exemple de moteur vidéo pour le temps réel : Mirage.

Mirage a été développé en 2003 par Jan Schacher, Pedro Soler et Georges Gagneré afin de réaliser une régie vidéo temps réel pour un spectacle de théâtre. Johnny Dekam et Jonathan Marcus ont poursuivi le développement en 2004.
Mirage utilise 'Max' et la bibliothèque d'objets 'Soft VNS' sur le système d'exploitation 'Mac OSX'. En résumé, 'Max' est un langage graphique qui permet de manipuler des « objets » (c'est-à-dire des petits programmes qui réalisent des fonctions élémentaires) afin de réaliser des programmes (appelés « patches »). Avec une bibliothèque d'« objets » sonores comme 'MSP', 'Max' permet de fabriquer et manipuler des sons, et avec des bibliothèque vidéo comme 'Jitter' ou 'Soft VNS', des images. Il ressemble à un mécano graphique qui permet d'assembler des morceaux de programmes entre eux afin de faire des programmes plus complexes. Des développeurs expérimentés peuvent développer leurs propres objets afin de répondre à leurs besoins personnels.
Le choix de 'Max' est lié à l'existence d'une communauté d'artistes vidéo et de régisseurs, déjà familiarisée à ce langage objet par le biais de leurs réalisations sonores. Le choix de 'Soft VNS' est lié à la rapidité de traitement vidéo qu'offre ce logiciel pour les images en direct.

Les choix qui caractérisent le compromis entre manipulation et fabrication sur lesquels repose Mirage sont les suivants :
1) commander facilement des sources d'images (séquences vidéo, directes ou capturées)
2) appliquer des effets sur les flux d'images sélectionnés dans les sources (channels)
3) agencer les images dans l'espace en proposant un système fixe de calques (layers)



Le moteur présente un module de sources et deux bus qui regroupent un choix d'effets et d'organisation spatiale des sources. Il est commandé par une interface de contrôle.
Le détail du tableau de contrôle (*board*) se présente ainsi :



En haut à droite se trouve le module des sources, qui contient huit boîtes présentant le choix suivant : caméra, movie (séquence vidéo), still (photo), buffer, external patch.
Puis il y a deux bus A et B qui comportent des channels et des layers (n'est visible sur l'image ci-dessus que le bus A en rouge)
On applique des effets sur les sources via les cases 'channels'. On peut traiter huit sources, et leur appliquer 7 effets simultanés dans chaque channel.
Ensuite, on place la source traitée dans l'espace via un système de 12 couches (layers) qui permettent de zoomer, de rogner l'image, de lui appliquer des masques, de la positionner et de la mélanger à d'autres images.
On peut réaliser simultanément une autre combinaison d'effets et de positionnements dans le bus B, à partir du même choix de sources. Les bus et les sources sont indépendants.

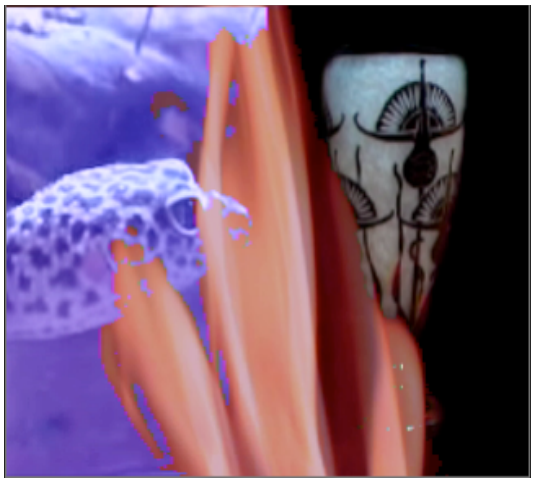
Mirage est gratuit et téléchargeable sur http://sourceforge.net/projects/mirage/. Il est contrôlable à partir de Max ou Pure Data (l'équivalent de Max en logiciel libre).

La documentation du logiciel est disposnible sur www.didascalie.net rubrique 'tech'.

Nous allons maintenant illustrer l'utilisation du moteur en construisant un état vidéo simple, puis nous manipulerons en temps réel le moteur par une interface de contrôle.

Edition

Voici l'état à construire (*compositing*) :



BUS A

Pour cet exemple, nous allons utiliser 3 médias (2 vidéos, 1 image).

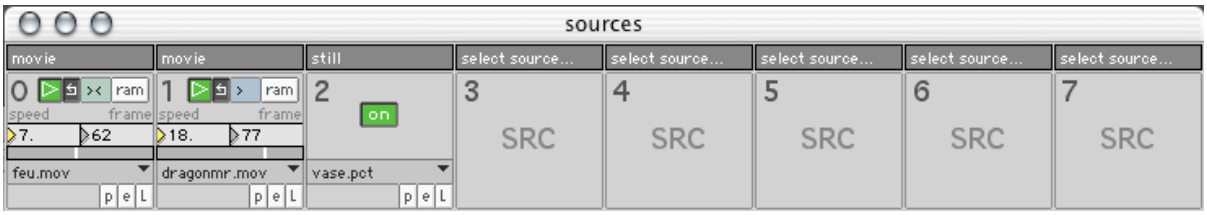
dragonmr.mov

vase.pct

feu.mov

Paramétrage des sources

On sélectionne donc trois sources dans le moteur :

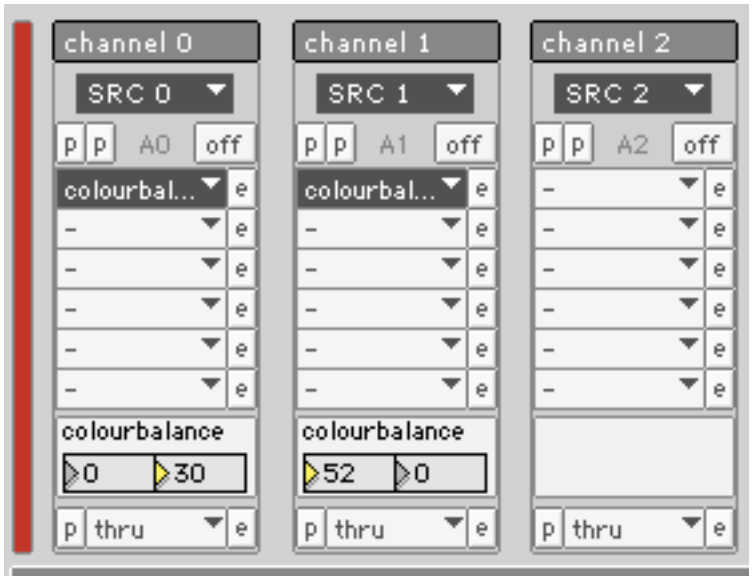


Chaque source présente plusieurs paramètres de réglages. Les séquences vidéo sont des fichiers qui sont sur le disque dur de l'ordinateur sur lequel le moteur est installé (en l'occurrence feu.mov et dragonmr.mov). De la même manière, l'image est un fichier informatique (vase.pct).

Paramétrage du bus A

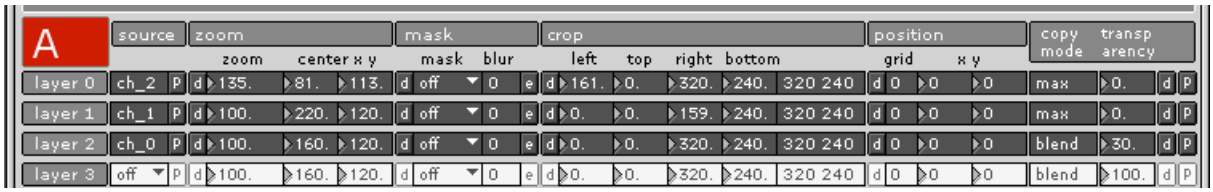
Le bus A est la combinaison d'une vidéo (dragonmr.mov) et d'une image (vase.pct) utilisant chacune la moitié de l'image vidéo finale. Le dragon est à gauche, le vase à droite. Les flammes (feu.mov) se superposent aux deux autres médias.

- BUS A channel (traitement des médias)

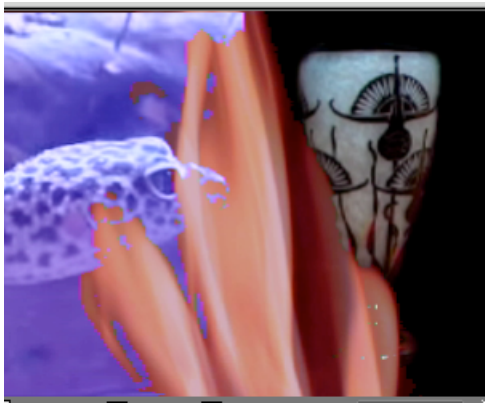


On applique à chaque source les effets dont on a besoin : nous utilisons en l'occurrence l'effet *colourbalance* qui permet de modifier les couleurs d'une image. On peut n'appliquer aucun effet (comme dans le cas de la source 2) ou plusieurs (jusqu'à 7 simultanément).

- BUS A layer (construction de l'état du bus)



Par exemple sur le layer 0 : nous devons centrer le vase dans la moitié droite, l'agrandir et couper le bord gauche. Nous faisons alors subir en temps réel les transformations suivantes à l'image :



- ZOOM : 135 (agrandissement de l'image d'un facteur 1,35) ;
- CENTER X,Y : 81, 113 (déplacement du centre de l'image de 79 pixels vers la droite et de 7 pixels vers le haut) ;
- CROP : LEFT : 161 (on coupe la gauche de l'image sur une longueur de 161 pixels, pour ne garder qu'une moitié d'image).

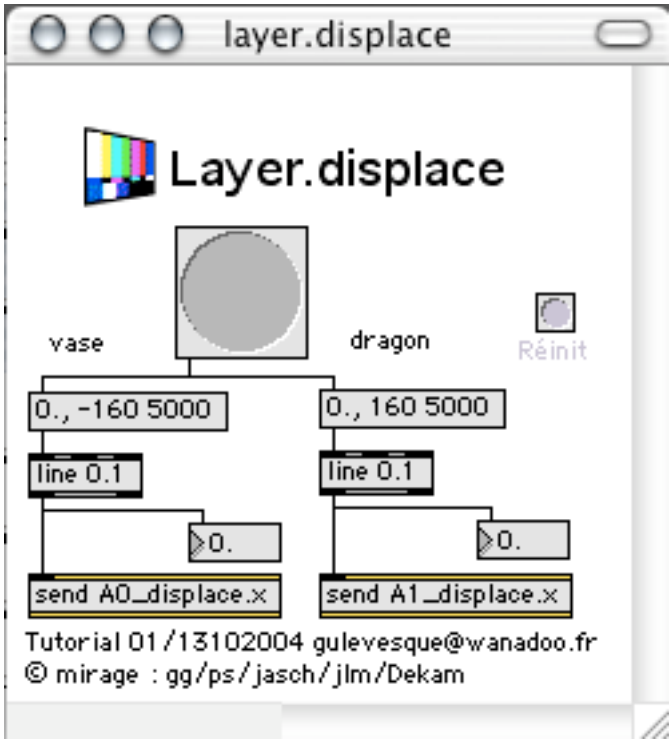
Compositing final

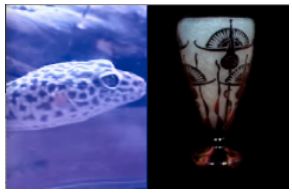
Contrôle

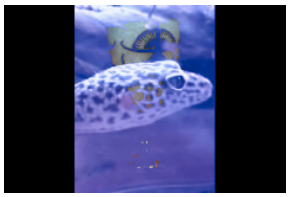
A partir de l'état vidéo que nous avons édité avec le moteur, nous pouvons désormais envisager une phase de manipulation. Nous allons mettre en œuvre un langage de programmation (nous utiliserons 'Max') pour envoyer des instructions à Mirage et transformer l'état vidéo précédemment construit. Il faut se représenter que tous les paramètres présents sur le tableau de bord de Mirage sont contrôlables en temps réel.

La manipulation par l'interface de contrôle que nous allons décrire correspond à un changement de l'agencement des layers (permutation entre le dragon et le vase)

Nous souhaitons permuter le dragon et le vase en 5 secondes (5000 ms). Le dragon va de gauche à droite (de 0 à 160) pendant que le vase va de droite à gauche (0 à –160).




départ


intermédiaire


final

Nous utilisons les commandes Mirage A0_displace.x et A1_displace.x qui jouent sur la position de l'image dans le layer correspondant, et que nous pilotons avec l'objet élémentaire « line » de Max.

Nous avons donc écrit un patch élémentaire de manipulation d'images.

Le regroupement des opérations nécessaires à un spectacle dans un patch forme la conduite.

Le comédien, via des capteurs, et le régisseur, via des potentiomètres midi, peuvent eux-mêmes contrôler les paramètres.

Un des enjeux importants de développement est actuellement d'écrire des patchs de conduite très facilement transformables en répétition.