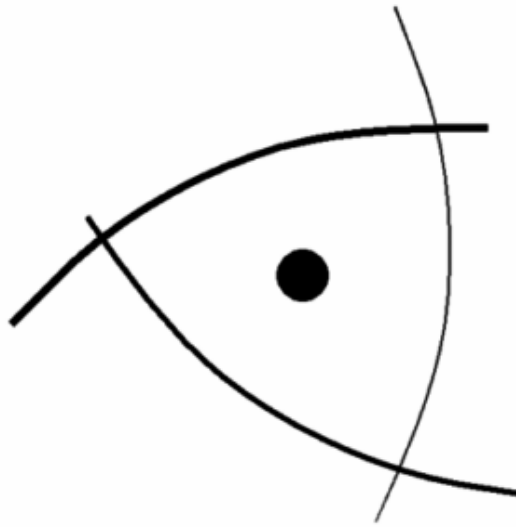


EyesWeb



Multimodal Interfaces

Auteurs : Leventini Simona, Cavadini Diego, Fonjallaz Yannick,
Brühlhart Florian

Superviseurs : Omar Abou Khaled & Jacques Bapst, Denis Lalanne

Département : Technologie de l'Information et de la
Communication

Filière : Informatique

Date : mars 2006 – mai 2006

Table des Matières

1.	Introduction	3
2.	Qu'est-ce que EyesWeb ?	3
3.	Où le trouver ?.....	3
4.	La fenêtre principale	4
5.	La structure de EyesWeb	5
5.1.	Manager.....	5
5.2.	Scheduler.....	5
5.3.	La programmation par blocs.....	5
5.4.	Datatypes	5
6.	Comment l'utiliser?	6
7.	Comment l'installer?	7
8.	Notre petit démo	9
9.	Conclusion.....	12
10.	Références.....	12

1. Introduction

Dans le contexte du cours d'Interfaces MultiModales, l'étude de différentes bibliothèques multimodales nous a été proposée comme travail pratique. Ces différentes bibliothèques utilisées sont centralisées sur une ou plusieurs modalités, comme par exemple le traitement de la voix, du geste, de l'expression du visage, le son, etc.

Pour pouvoir effectuer ce travail la classe a été divisée en groupes dont chacun prend en charge l'étude d'une bibliothèque.

Notre groupe a étudié et testé la bibliothèque EyesWeb. Le rapport suivant en expliquera le but, le fonctionnement, l'usage de notre bibliothèque ainsi qu'une petite démonstration.

2. Qu'est-ce que EyesWeb ?

EyesWeb est un utilitaire de programmation par blocs qui permet de travailler sur certaines modalités, par exemple la vidéo. Il est étudié pour faciliter l'interprétation de ces modalités.

3. Où le trouver ?

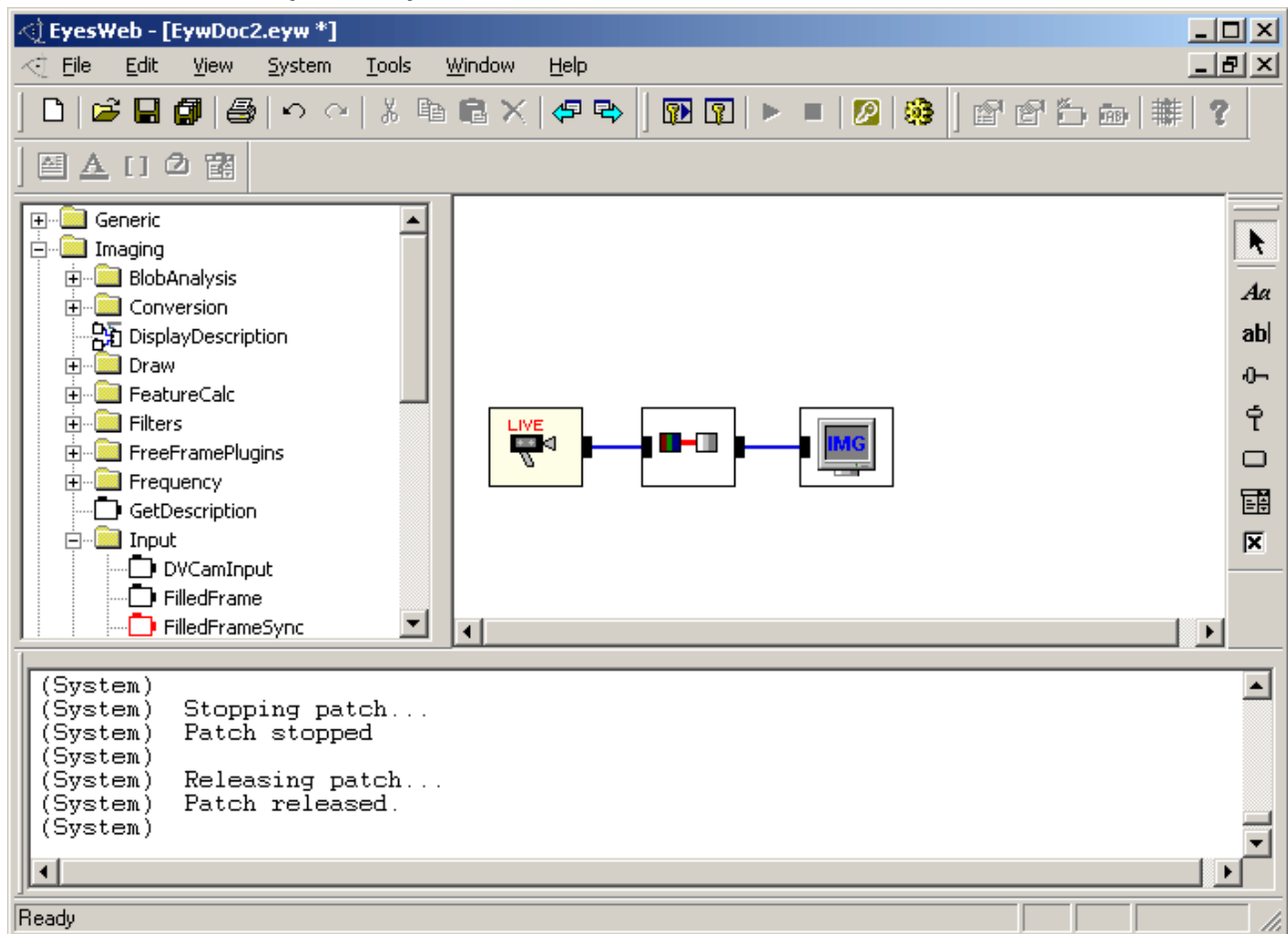
EyesWeb est une plateforme gratuite, il est possible de la télécharger via le site suivant :

<http://www.infomus.dist.unige.it/eywindex.html>

Il existe une version 4.0 en tests, la dernière version stable étant la 3.3.0.

Nous avons préféré utiliser une version stable pour nos tests.

4. La fenêtre principale



La fenêtre est répartie en quatre parties principales.

Tout d'abord, les menus et barres d'outils en haut. Ils comportent les menus et opérations habituels (File, Edit, View, new, open, ...) ainsi que les fonctions principales d'EyesWeb (Locked mode, run, stop, ...). Le bouton avec une fenêtre et une petite clef jaune permet de passer en « locked mode ». C'est une étape obligatoire avant l'exécution du programme, c'est l'équivalent de la compilation. S'il n'y a pas d'erreur, le programme sera prêt à être exécuté par pression sur le bouton en triangle.

Le bouton semblable mais avec une flèche sur la droite permet de verrouiller le programme et passer directement en exécution si il n'y pas d'erreur.

Ensuite, la partie de gauche avec l'arborescence de dossier répertorie les divers blocs qu'il est possible d'ajouter au programme.

C'est dans la partie centrale que l'on programme. C'est ici qu'il faut déposer les blocs et les connecter entre eux (voir chapitre 6). La barre d'outils à droite permet d'insérer divers outils d'interface homme-machine standards, comme des zones de texte, des cases à cocher, etc.

Finalement, la zone du bas est un log des appels systèmes effectués. Les diverses erreurs (telles qu'une entrée non reliée) y seront affichées.

5. La structure de EyesWeb

Les composantes fondamentales de la structure d'EyesWeb sont décrites dans les sous chapitres suivantes.

5.1. *Manager*

Ce composant maintient l'information sur les interconnexions parmi les blocs, et vérifie leur exactitude : quand un lien entre deux blocs existe, il vérifie que les types de données échangés sont homogènes. Le manager est également responsable de la visualisation des blocs, donc des informations sur le nombre d'entrées et sorties de chaque bloc, leur nom, le nombre et le type de paramètres, et chaque autre genre d'information à montrer à l'utilisateur doivent être rendues disponibles à ce composant.

5.2. *Scheduler*

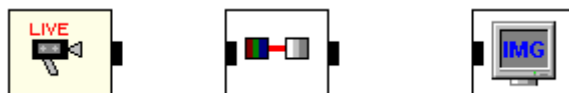
Le scheduler sert essentiellement à gérer l'exécution des différents blocs. C'est-à-dire que c'est lui qui se charge d'activer et de désactiver les blocs pendant l'exécution du programme, ainsi que d'établir un plan d'action lors de la « compilation ».

5.3. *La programmation par blocs*

Les blocs représentent les actions ou captures d'EyesWeb. La programmation dans EyesWeb se fait en connectant des blocs entre eux.

Les blocs génèrent ou travaillent donc les données propres à EyesWeb. Il est possible d'en créer à partir des « programmes » développer, ce qui pourrait être comparé à des fonctions dans d'autres langages.

Les entrées sont généralement à gauche des blocs, et les sorties à droite.



Voici trois exemples de blocs.

Le premier capture les images d'une webcam connectée à l'ordinateur. Il génère une vidéo dans le type de données d'EyesWeb.

Le second bloc transforme la vidéo couleur en noir et blanc.

Le dernier permet de visualiser une vidéo sur une fenêtre lors de l'exécution.

5.4. *Datatypes*

Chaque bloc possède une ou plusieurs entrées ou sorties. Celles-ci ont chacune un type particulier (image, nombre mathématique, graphique, vidéo, ...).

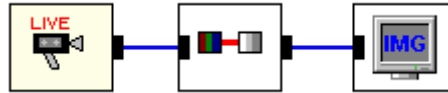
Il est évident que l'on ne peut pas connecter ces blocs n'importe comment (par exemple un nombre en entrée d'un bloc faisant un traitement sur la couleur d'une vidéo n'a pas de sens), pour visualiser le type d'une entrée ou sortie d'un bloc, il faut placer la souris sur cette entrée/sortie et lire dans la zone de notification à quoi elle correspond.

Ceci pose pas mal de problèmes lorsque l'on tente d'expérimenter de nouveaux blocs, en effet c'est uniquement lors de ce que l'on pourrait appeler la compilation du programme qu'une erreur est affichée si un lien est incorrect.

6. Comment l'utiliser?

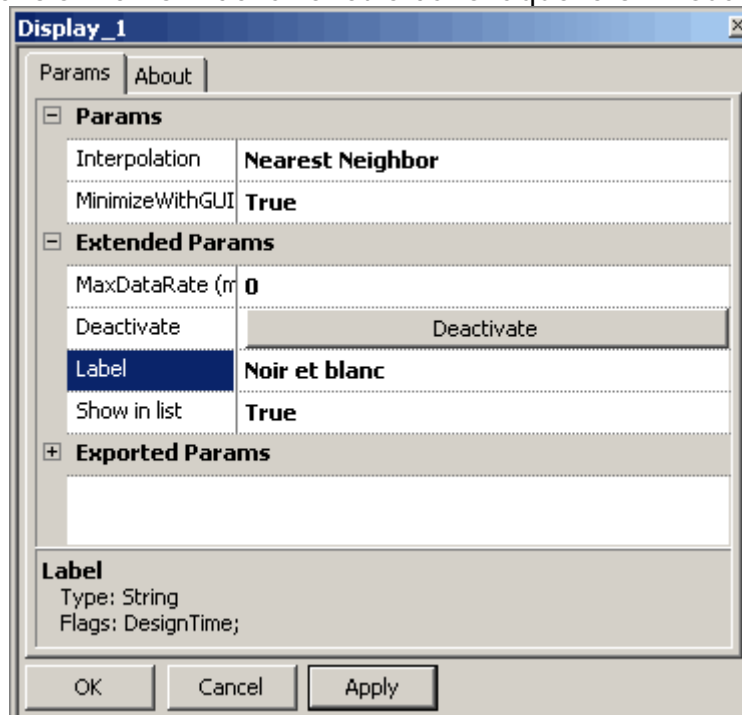
Les blocs doivent être reliés entre eux. Pour ce faire, cliquer avec la souris sur une sortie et maintenir le bouton pressé jusqu'à l'entrée du bloc suivant. Une sortie peut avoir plusieurs connexions, mais une entrée ne peut recevoir qu'une information.

Exemple une fois les trois blocs de l'exemple précédant reliés :



Ceci affichera simplement l'image filmée par la webcam en noir et blanc.

En double cliquant sur un bloc, il est possible de régler ses paramètres. Par exemple, il est possible de choisir le nom de la fenêtre dans laquelle on visualisera la vidéo :



En cliquant sur le bouton « Deactivate », le bloc est désactivé. Pour un afficheur cela aura pour simple effet de ne pas afficher la fenêtre. Pour un bloc tel que celui qui transforme la vidéo en noir et blanc, la sortie sera entièrement noire (et donc désactivée). Ceci empêche donc ce qui suit de fonctionner correctement mais le reste du programme pourra toujours s'exécuter. Par contre, s'il on enlève simplement ce bloc central, le programme ne pourra pas se lancer car une entrée ne reçoit aucune donnée.

Voici le résultat de l'exemple :

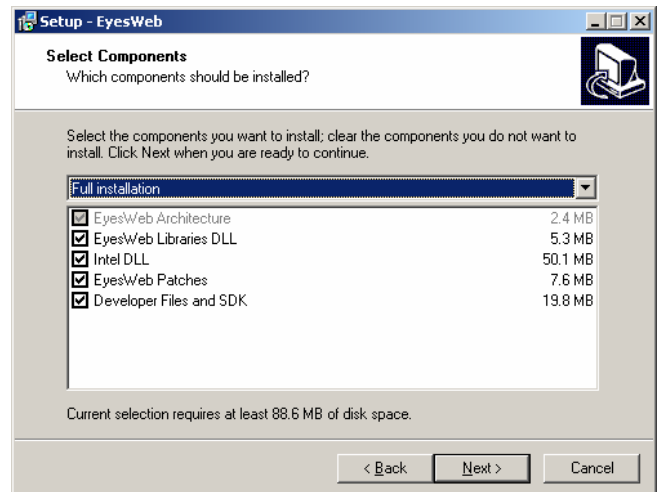
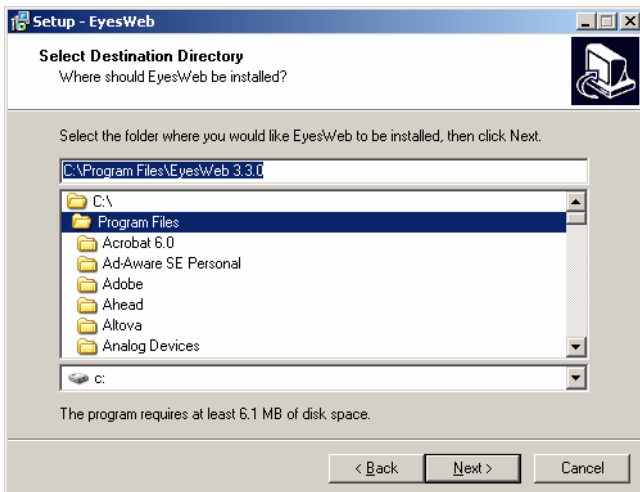
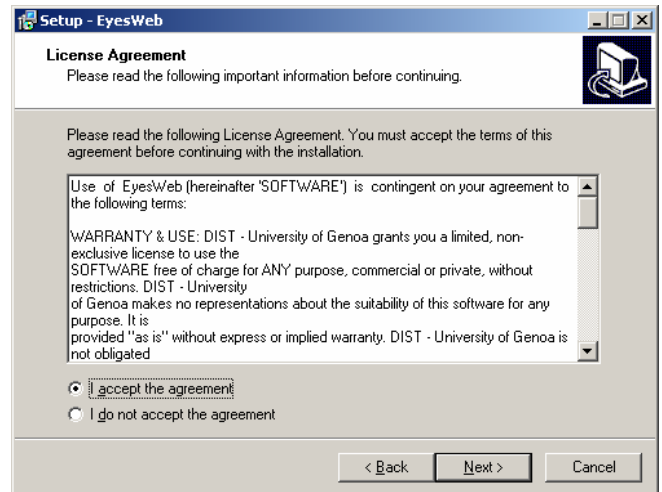
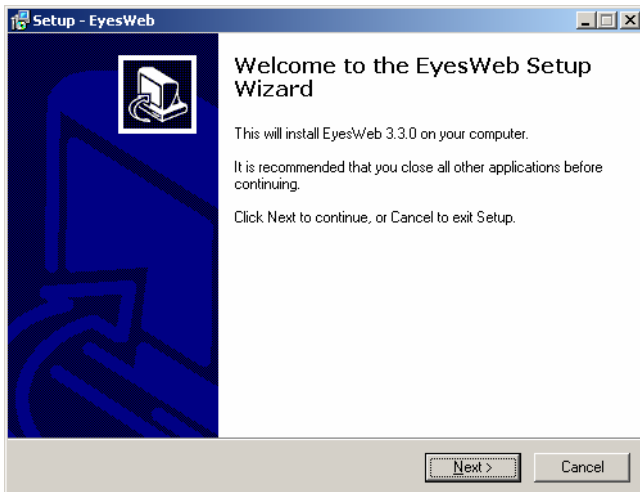


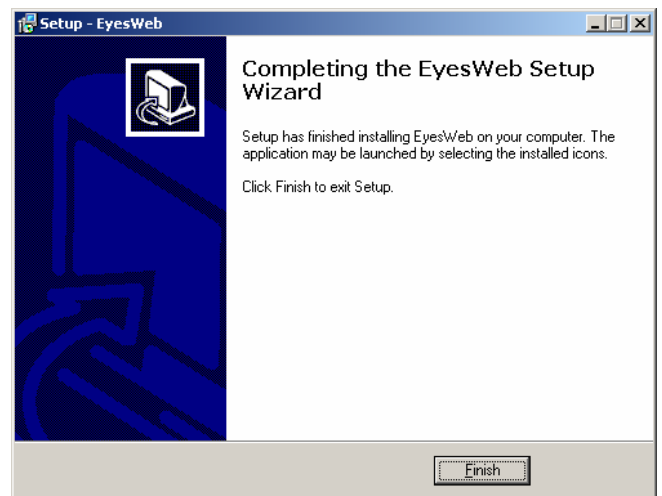
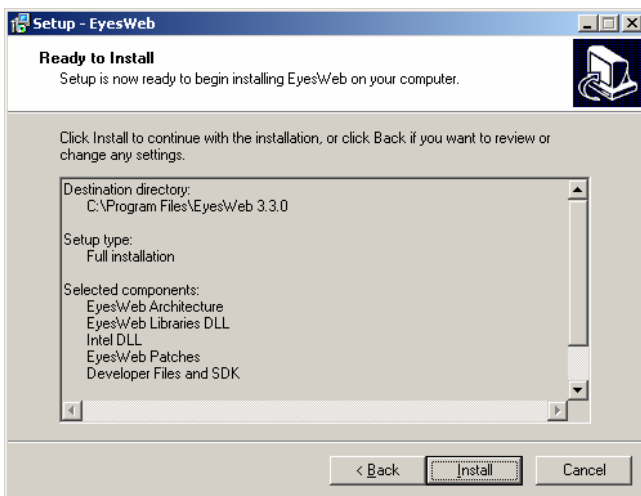
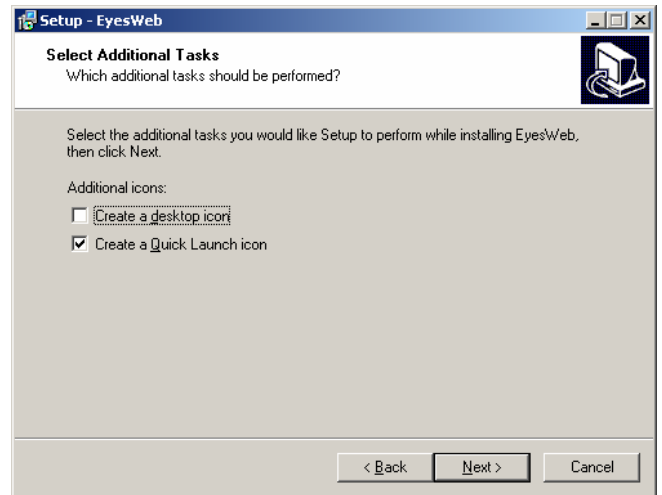
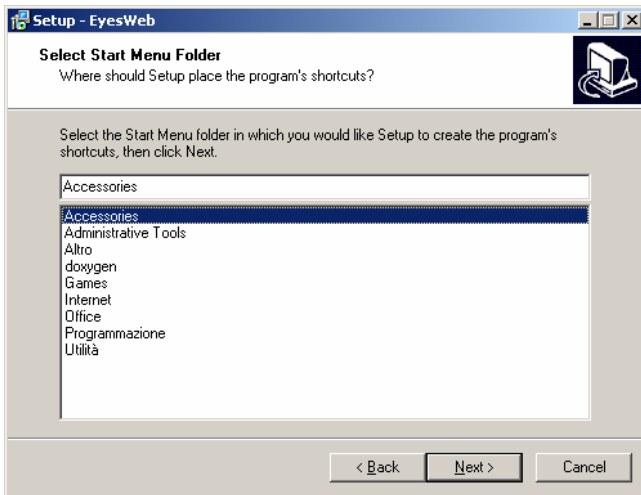
Lors de l'exécution du programme, on peut remarquer que la rapidité de réaction est assez surprenante. En effet les changements se font en temps réel (ce qui est difficile à prouver dans un rapport sur papier).

7. Comment l'installer?

EyesWeb est un programme très facile à installer, il suffit de télécharger l'exécutable (EyesWeb_full_3.3.0.exe) et de l'exécuter.

Ici de suite nous avons mis les pas à suivre pour installer l'application :





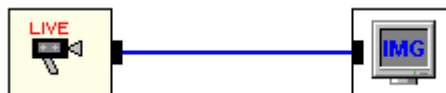
8. Notre petit démo

Les possibilités d'EyesWeb sont très vastes. Nous nous sommes concentrés sur le traitement de la vidéo.

La démo suivante permet d'obtenir divers effets à partir d'une webcam.

Tout d'abord, posons le bloc de capture de la vidéo. Il se trouve dans Imaging / Input / FrameGrabber. Si vous utilisez une caméra DV, vous devez utiliser le bloc DVCamInput.

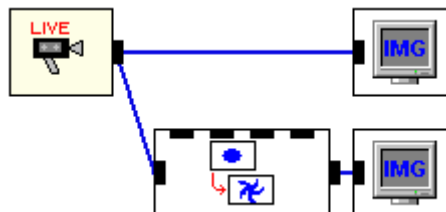
Vous pouvez le relier directement à un bloc d'affichage (Imaging / Output / Display) si vous désirez visualiser la vidéo sans aucun traitement.



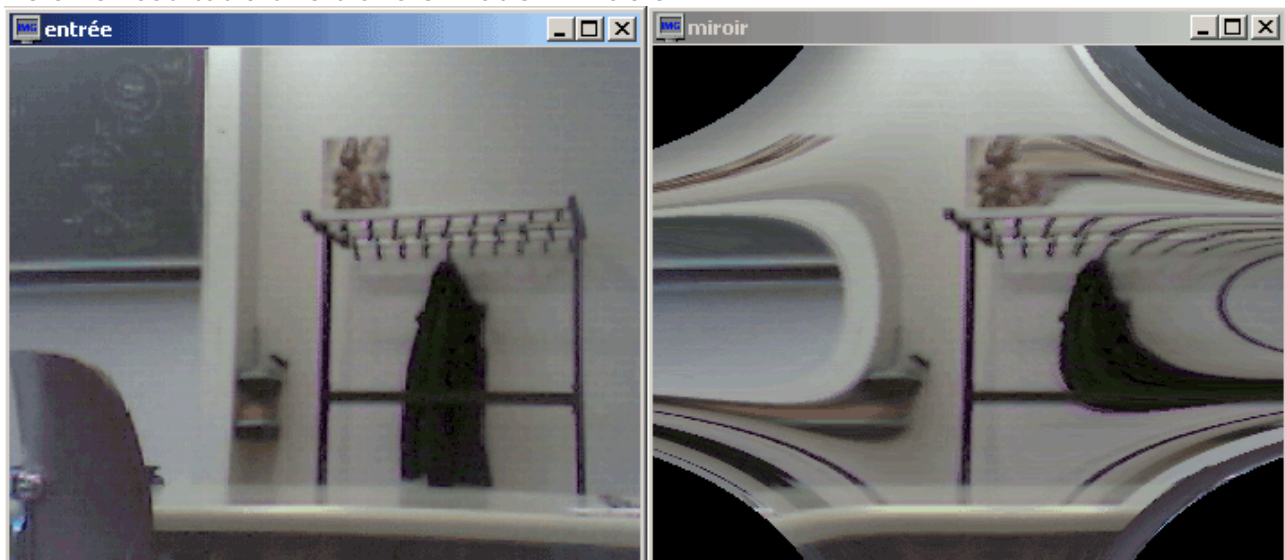
Comme premier exemple de traitement nous allons déformer un peu l'image. Pour ce faire, ajouter un bloc de morphing (Imaging / Operations / MorphMirror).

La vidéo doit être branchée sur l'entrée de gauche. Les quatre entrées sur le côté supérieur du bloc permettent de changer le type de traitement au cours de l'application. Nous ne l'avons pas fait et changeons les paramètres manuellement avant l'exécution. Donc en double cliquant sur le bloc, vous obtenez la fenêtre de paramètres. Choisissez une des transformations (Neutral, FishEye (pour générer une image en œil de poisson), Vortex (tourbillon) ou Diablo (sorte de FishEye inverse)) et mettez sa valeur à 1 pour que la transformation s'applique.

Connectez un bloc d'affichage à la sortie afin de visualiser le résultat. Testez les transformations comme bon vous semble.



Voici le résultat d'une transformation "Diablo" :

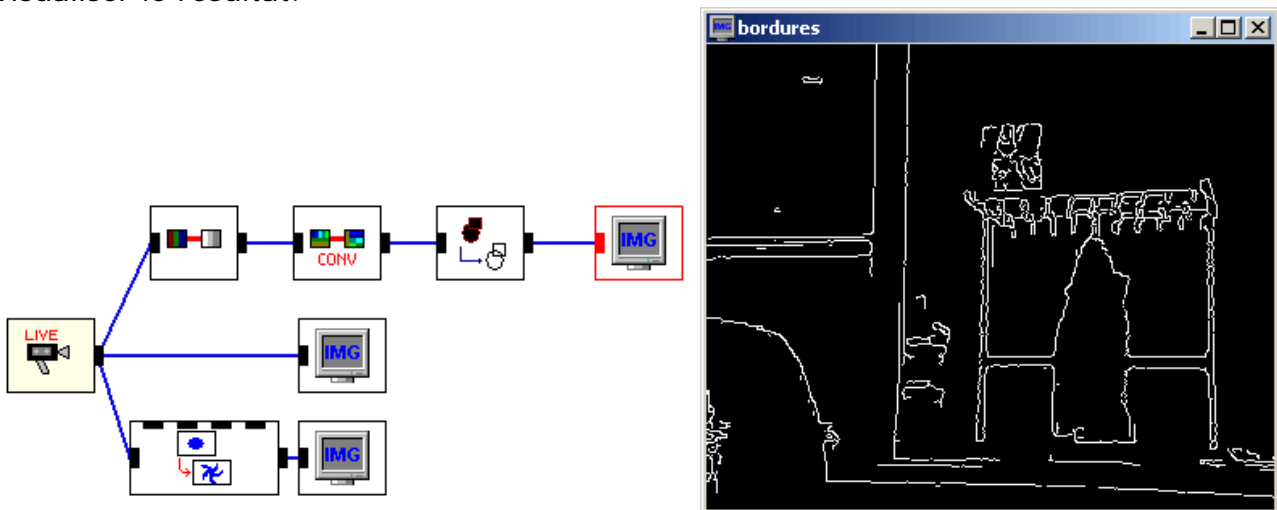


Nous allons maintenant traiter la vidéo pour que les bords des objets seulement ressortent. Cette opération est basée sur le contraste d'une image et donc est traitée depuis une image en noir et blanc.

Ajoutons donc un bloc qui transforme une image couleur en niveaux de gris (Imaging / Conversion / ColorToGray), puis relient son entrée aux images de la webcam. Il faut y ajouter un bloc de conversion (Imaging / Conversion / Convert) avec les paramètres suivants :

- destination depth : same as input ;
- data origin : same as input ;
- destination origin : top left.

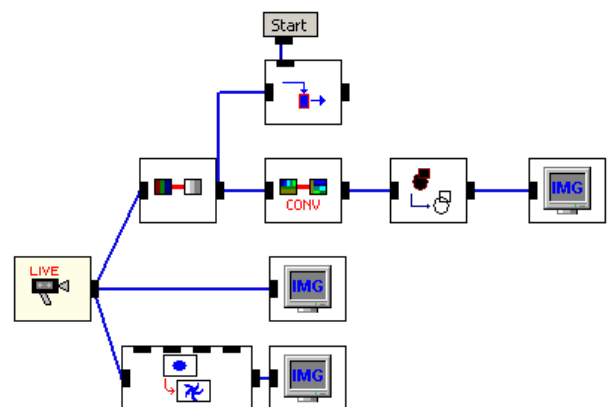
Ce qui aura pour unique effet de changer le système de coordonnées du système, nécessaire pour le bloc suivant. Celui-ci (Imaging / FeatureCalc / CannyCornerDetect) réalise justement le traitement de l'image pour en ressortir les bordures. Les paramètres par défaut conviennent à l'application. Finalement, ajoutez un bloc pour visualiser le résultat.



Le résultat est assez convaincant vu le peu de contrastes entre certains objets.

Avec les prochaines opérations, il vous sera possible (dans des conditions optimum) de faire une publicité pour l'iPod d'Apple (©). C'est un petit peu plus compliqué que ce qui a été fait jusqu'à maintenant.

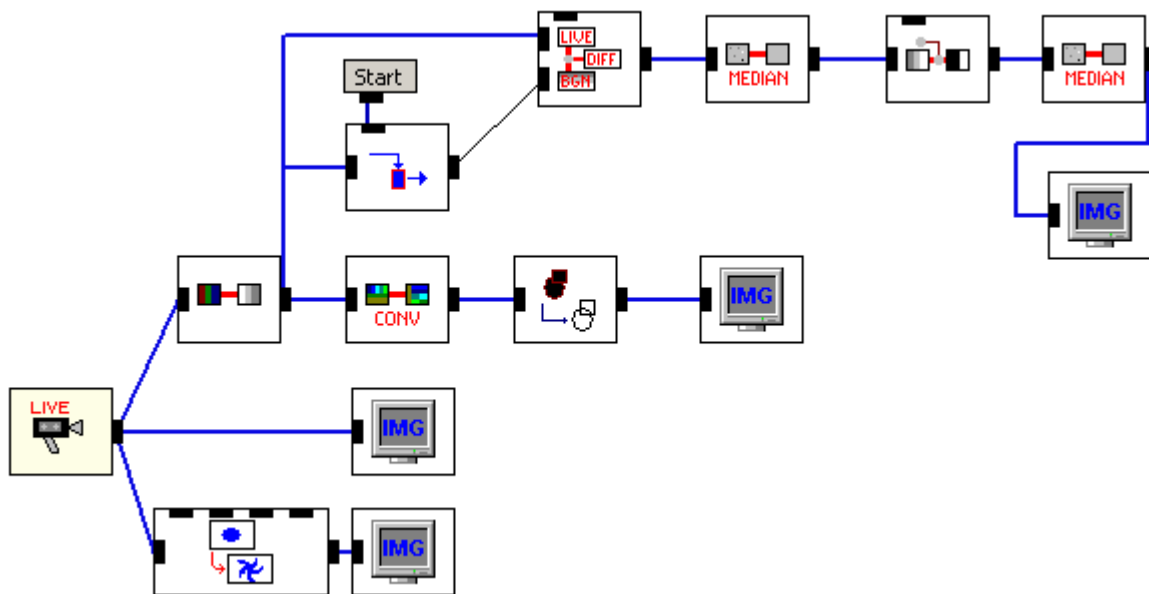
Comme il faut commencer par une conversion en niveaux de gris, il est possible d'utiliser le même bloc que pour la dernière « expérience ». La sortie doit être reliée à un buffer (Generic / Snapshot). Celui-ci doit recevoir un signal pour stocker les informations qu'il a en entrée (comme l'horloge d'un système de bascules). Comme ce signal doit être envoyé au lancement de l'application, nous allons utiliser un bouton qui se déclenchera automatiquement. Dans la palette de droite, choisissez un « command control » et ajoutez-le au schéma. Dans les paramètres, sélectionnez PatchStart comme Event.



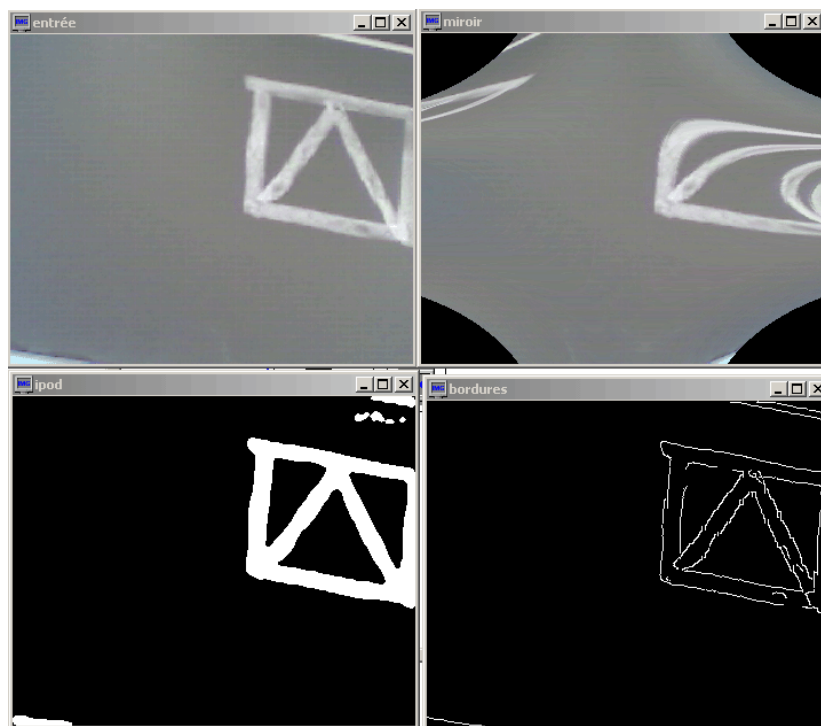
Ce tampon (buffer) nous servira afin d'enlever l'image de fond, ce qui est la prochaine étape. Il suffit d'utiliser un bloc (Imaging / Operations / BgndSubMultThresh) possédant deux entrées, l'une pour l'image de fond (provenant du buffer, entrée du bas) qui sera « enlevée » à l'autre image, provenant de la webcam.

Il faut ensuite appliquer un filtre pour enlever le « grain » de l'image, avant de la transformer en noir et blanc intégral. Il faut encore ajouter quelques blocs :

- Imaging / Filters / NonlinearFilter pour enlever le grain, avec les paramètres suivant : Filter Median comme type de filtre, 5 pour le nombre de colonnes et de lignes ;
- Imaging / Operations / Threshold pour créer l'image désirée ;
- Un second filtre Nonlinear avec les mêmes paramètres (il est possible de copier / coller le précédent) afin de purifier l'image ;
- Un affichage pour la visualisation.



Résultat :



9. Conclusion

Nous avons découvert qu' EyesWeb est un utilitaire assez pratique qui permet de faire de lourds traitements en temps réel et de les développer très peu de temps. Ces traitements peuvent s'appliquer non seulement aux images, mais aussi aux sons. On pourrait donc imaginer une application développée via EyesWeb pour gérer les mouvements et la voix comme interfaces multimodales.

C'est un outil très puissant qui permet de faire beaucoup de choses, nous n'avons malheureusement pu étudier qu'une petite partie. Il nous a malgré tout fallu pas mal de temps pour nous habituer à la programmation par blocs ainsi que comprendre les divers soucis de typages des données.

Il serait intéressant d'étudier les possibilités de gestion du son et de sorties du logiciel, ainsi que les nouveautés disponibles dans la version 4.

10. Références

<http://www.eyesweb.org/> : le site d' EyesWeb

<http://www.infomus.dist.unige.it/EyesWeb/EywTutorials.html> : quelques tutoriaux

<http://www.cs.aue.auc.dk/~rea/eyesweb/> : quelques notes pratiques