

Rappels

Quelques raccourcis intéressants :

Ctrl + R : Exécuter le programme

Ctrl + Maj + T : "Tweak" : permet de modifier la valeur des variables et voir le résultat en direct

Ctrl + Espace : complétion intelligente

Ctrl + T : indentation automatique du code

Documentation : <https://processing.org/reference>

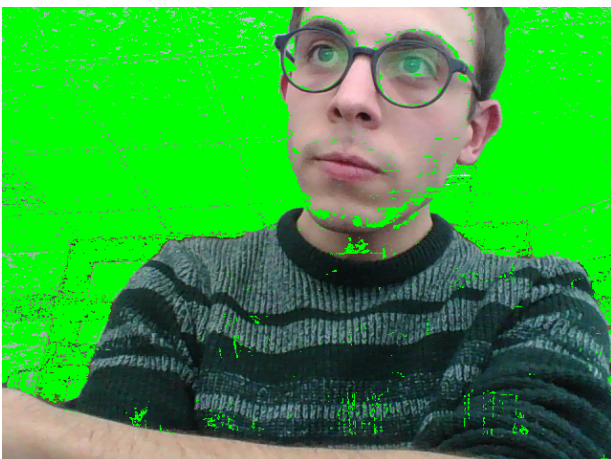
Documentation OpenCV : <http://atduskgreg.github.io/opencv-processing/reference/>

Avant de commencer ce TD, assurez-vous que les bibliothèques Video et OpenCV soient bien installées dans Processing !

Analyse de vidéos

Exercice 1 : Soustraction de fond

Nous souhaitons insérer une image de fond dans un flux vidéo (webcam). L'idée est que l'objet a une couleur différente du fond, et qu'il est donc possible d'identifier les pixels ayant apparus dans la vidéo par une différence avec l'image de fond. Nous allons pour cela sauvegarder une image de fond sans l'objet d'intérêt (votre visage). Lorsque la couleur d'un pixel des images suivantes de la vidéo est éloignée de celle du pixel de fond, il s'agit d'un pixel objet, sinon, il s'agit d'un pixel de fond.



Lisez attentivement le fichier **suppression_fond_diff.pde**. Exécutez le programme. Pour l'instant, l'enregistrement de l'image de fond n'est pas pris en charge.

1. La fonction `mousePressed()` est définie par Processing et est appelée à chaque fois que l'utilisateur clique sur un bouton de la souris. Complétez la fonction `mousePressed()` à l'endroit marqué d'un commentaire **TODO** de façon à ce que l'image visualisée au moment du clic soit enregistrée dans la variable `imageFond`. Pour cela, vous pourrez utiliser la fonction `copy()` sur la vidéo.
2. Testez votre programme : enregistrez l'image lorsque votre visage est hors-champ. Puis revenez dans le cadre : un fond vert doit apparaître à la place des pixels de fond.
3. Remplacez le fond vert par une image de remplacement. Dans la fonction `suppressionFond()`, modifiez les instructions dans la condition `if (diff > seuil)` de façon à attribuer la couleur du pixel de l'image de remplacement au lieu de la couleur verte.
NB : une image de remplacement est déjà chargée dans la fonction `setup` (variable `imageRemplacement`).
4. Remplacez le fond avec d'autres images de votre choix.

Exercice 2 : Soustraction de fond avec OpenCV

OpenCV est une bibliothèque contenant de nombreux algorithmes de traitement et d'analyse d'images et de vidéos. Elle est initialement écrite en C++, puis adaptée en module pour Processing.

Le but de cet exercice est de vous familiariser avec OpenCV, en utilisant l'algorithme de soustraction de fond basé sur un mélange de gaussiennes.

Utilisez la documentation et les exemples disponibles à cette adresse.



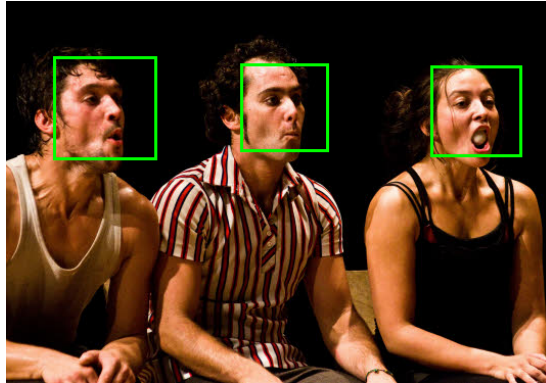
Lisez attentivement le programme **suppression_fond_mog.pde**. Ce programme affiche les contours des objets en mouvement en rouge.

Quelle est la différence entre cet algorithme de suppression de fond et celui implémenté dans l'exercice 1 ? Pour quelle application cet algorithme est-il plus adapté ?

Exercice 3 : Détection de visage

Le but de cet exercice est d'utiliser l'algorithme de détection de visage implémenté dans le module OpenCV de Processing pour détecter les visages présents dans un flux vidéo.

Lisez attentivement le fichier **detection_visage.pde**. Le programme permet de détecter les visages dans une image statique.



1. Modifiez le programme afin de détecter les visages à partir du flux vidéo (webcam).
2. Consultez la documentation du module OpenCV pour modifier la cascade utilisée (CASCADE_FRONTALFACE) afin de détecter :
 - les oreilles
 - les yeux
 - le nez

Exercice 4 : Extraction de caractéristiques

Le but de cet exercice est d'extraire des caractéristiques sur une image segmentée, à l'aide d'OpenCV.



Lisez attentivement le fichier **extraction_caracteristiques.pde**. A l'exécution, une image de feuille apparaît, et l'aire (en nombre de pixels) s'affiche dans la console.

À l'aide de la documentation OpenCV, et du code de l'exercice précédent :

1. Extrayez les contours à l'aide de la fonction `findContours()` de OpenCV.
2. Extrayez la boîte englobante de la feuille à l'aide de la fonction `getBoundingBox()` du contour.
3. Affichez la boîte englobante à l'aide de la fonction `rect(x,y,w,h)`, ainsi que son centre à l'aide de la fonction `ellipse(x,y,r,r)`.
4. Calculez et affichez le périmètre de l'objet à l'aide de la fonction `numPoints()` du contour (nombre de points dans le contour).

5. Calculez et affichez la compacité de l'objet à l'aide la formule suivante : $\frac{4\pi \text{ Aire}}{\text{Périmètre}^2}$. π s'écrit PI en Processing.
6. Calculez et affichez l'élongation de l'objet : $\frac{\text{longueur boîte englobante}}{\text{largeur boîte englobante}}$
7. Testez votre programme sur d'autres images (présentes dans le dossier data).