

# Traitement d'images avec OpenCV sous Visual Studio

Ce TP concerne un traitement d'images simple à l'aide de la librairie OpenCV. D'autres TP suivront et permettront d'effectuer des traitements plus approfondis.

## 1. Traitement d'images fixes

### Affichage d'une image dans une fenêtre

OpenCV est une librairie de classes C++ (il existe des versions Python et Java). La classe, sans doute, la plus importante est la classe : « Mat » pour matrice.

La plupart des classes se situent dans l'espace de noms : cv

Si vous ne précisez pas que vous utilisez l'espace de noms cv alors il faudra faire précéder le nom des classes et des fonctions de cv ::

Il est conseillé de mettre en début de programme la directive : *using namespace cv;* pour ne pas à avoir à rajouter systématiquement cv :: devant les classes et fonctions de l'espace de noms cv

### Traitement d'images

En fait, tout traitement sera effectué à partir de la représentation d'une image sous forme de matrice.

La première opération à effectuer est de copier l'image que l'on veut traiter dans une matrice :

```
Mat img = imread("imagetest.jpg", IMREAD_COLOR);
```

*On instancie un objet img de la classe Mat. On l'initialise avec le résultat de l'appel de la fonction imread()*

*imread() ouvre le fichier image « imagetest.jpg »*

*Le second paramètre est un flag : il indique la manière de lire l'image. Ici l'image sera chargée en version RVB (Rouge Vert Bleu)*

Remarque :

La version que nous utilisons est la version 3 d'OpenCV. Il y a de grosses différences entre les versions 2 et 3. Par exemple, ici, les valeurs possibles pour le second paramètre ne sont pas compatibles entre les deux versions.

Après avoir ouvert le fichier image, il faut l'afficher dans une fenêtre. C'est la fonction *imshow()* qui s'en occupe. Il faut systématiquement faire suivre l'appel à *imshow()* par un appel à *waitkey()*.

```
imshow("image", img);  
waitKey(1);
```

### waitkey()

Si on traduit : attendre une touche. Mais en fait, c'est bien plus que cela ! Le paramètre passé à *waitkey()* est un temps en ms. Ici on attend 1ms.

*waitkey()* fait en sorte de rendre la main à la boucle des événements de visual.

## Accès à un élément de la matrice

Si on veut accéder au pixel (i,j) de l'image, il faut accéder au pixel (i,j) de la matrice. Seulement, un pixel est formé de trois valeurs sur 8 bits : la valeur de rouge, la valeur de vert et la valeur de bleu.

Si on effectue un accès à l'aide de la méthode `at()` de « `img` », on obtient les trois valeurs d'un coup. En fait `at` est une fonction template. Autrement dit, il faut lui dire qu'on veut par exemple, récupérer un vecteur de 3 octets non signés : `Vec3b`

Ça s'écrit ainsi :

```
Vec3b unPixel = img.at<Vec3b>(i, j);
```

*On instancie un objet `Vec3b` qu'on appelle `unPixel`. `unPixel` est initialisé avec le vecteur composé de trois octets trouvé à la position (i,j) de l'image.*

Une matrice `Mat` représentant une image :

	i-2	i-1	i	i+1	i+2
j-1	R : 250 V : 125 B : 45	R : 20 V : 145 B : 115	R : 250 V : 125 B : 45	R : 250 V : 125 B : 45	R : 250 V : 125 B : 45
j	R : 250 V : 125 B : 45	R : 28 V : 210 B : 78	R : 24 V : 212 B : 77	R : 30 V : 218 B : 77	R : 25 V : 209 B : 76
j+1	R : 250 V : 125 B : 45				

Vec3b à la position (i,j)

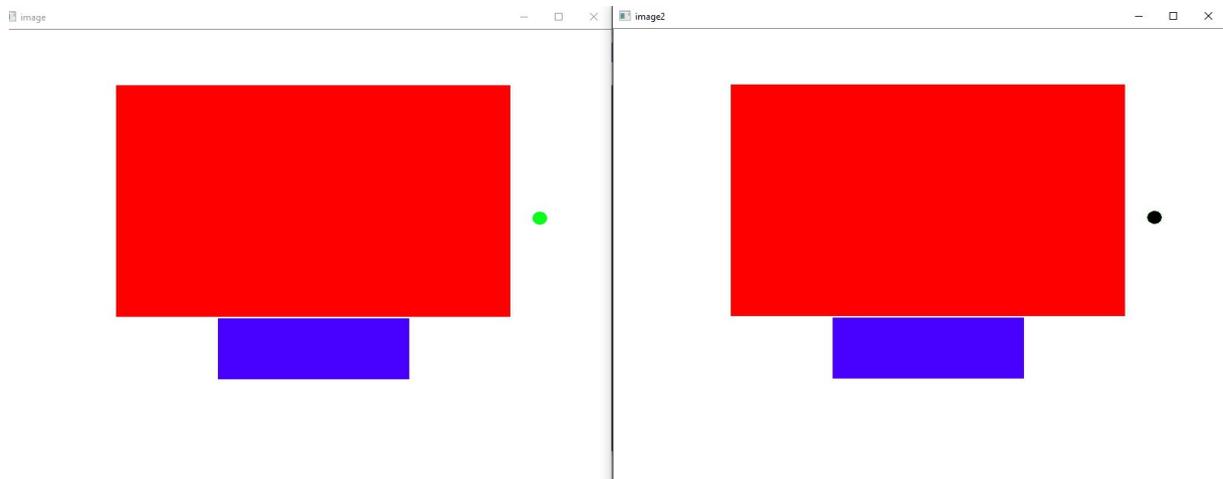
On écrit donc entre `< >` le type que l'on veut en retour.

Attention : sous OpenCV les trois composantes sont dans l'ordre inverse : BVR au lieu de RVB

Par conséquent, si on accède à `unPixel[0]` on accède à la valeur du bleu.

## Travail demandé

Afficher à gauche l'image test originelle et à droite l'image dont vous aurez modifié le petit cercle vert en un cercle noir.



Il est conseillé de balayer l'image de haut en bas (connaître la taille de l'image ?) et de tester si on a des pixels de couleur verte (qu'est-ce que la couleur verte ?)

### Transformer une image couleur en noir et blanc

Lorsqu'on effectue une reconnaissance d'objets dans une image et ou reconnaissance de formes, bien souvent il est plus facile de travailler sur une image nettoyée (filtrée) et convertie en noir et blanc de façon à obtenir des images avec beaucoup de contraste.

Prenez une photo de votre choix et affichez sa version noir et blanc : pour cela, rien de très compliqué ; il suffit de détecter tout ce qui n'est pas blanc (qu'est-ce que le blanc ?)

### Utilisation de fonctions de traitements d'images d'OpenCV

OpenCV contient un certain nombre de fonctionnalités de traitements :

Essayer la fonction *GaussianBlur()* qui permet d'éliminer le bruit dans l'image. C'est-à-dire que la fonction réhausse les pixels et éliminent les pixels dus au bruit.

## 2. Traitement d'images animées (vidéo)

### Affichage d'une vidéo

La classe s'occupant de gérer les flux vidéo est VideoCapture de l'espace de noms cv.

Choisir une vidéo quelconque (.mp4) et développer une application ouvrant une fenêtre et affichant la vidéo.

Dans le code, vous allez devoir afficher la vidéo image par image. Evaluer le temps nécessaire à attendre entre chaque image pour arriver à 24 images/s (cinéma)

### Affichage le flux vidéo à partir d'une caméra USB

Utilisez la classe VideoCapture afin d'afficher cette fois-ci le flux vidéo provenant d'une caméra USB.

Réutilisez le traitement effectué sur les images fixes afin de modifier « en temps réel » le flux vidéo de la caméra : ce qui apparaît en vert est transformé en noir.

Vérifier le filtrage en présentant devant la caméra, un objet de couleur verte.

### Affichage d'incrustations

Après avoir récupéré l'heure et la date, afficher ces informations en haut à gauche de la fenêtre où la vidéo est affichée (incrustations)