

# Caméra et flux vidéo.

Pour les PC de CPE, le TP se déroulera sous **Windows**.

Il est possible d'utiliser deux logiciels différents:

**VLC**, un logiciel de capture et de manipulation de flux vidéo.

<http://www.videolan.org/vlc/index.html>

→ Lancez le logiciel VLC.

**VirtualDub**, un autre logiciel fonctionnant sous Windows uniquement.

<http://www.virtualdub.org>

→ Téléchargez VirtualDub, décompressez le. Vous pouvez le lancer sans installation.

## Capture de flux vidéo

Démarche pour réaliser la capture du flux de la caméra:

**Pour VLC:**

- Media
  - Open Capture Device / Ouvrir un périphérique de capture  
*Capture mode (mode de capture) doit être sur **Video camera** ou **DirectShow***  
*Lancez **Play** (/Lire) en bas à gauche*

→ Observez le flux vidéo de la caméra.

**Pour VirtualDub:**

- File
  - Capture AVI

## Analyse du flux vidéo

Pour obtenir différentes informations sur le flux vidéo:

### Sous VLC:

- Tools / Outils
  - Codec Information / Informations sur les codecs
  
- ➔ Trouvez la dimension de la prise de vue en nombre de pixels (résolution).
  
- ➔ Réalisez une capture d'image (Video; Take Snapshot).
- ➔ Vérifiez que la dimension de votre image (le fichier sur le disque dur) soit bien celle indiquée par VLC.
  
- ➔ Quel est la fréquence de mise à jour des images de la capture vidéo (Codec Information de VLC) ? Déduisez l'intervalle de temps (en secondes) entre deux images consécutives.

### Taille mémoire

Un pixel contient l'information des canaux rouge, vert et bleu. Chaque canal est codé sur 256 niveaux. La couleur d'un pixel peut donc être encodé sur 3 octets en mémoire.

- ➔ Calculez la taille prise en mémoire (en Mo) par une image (non compressée).
- ➔ Calculez la bande passante théorique nécessaire pour transmettre l'image de cette webcam.

Pour information, un débit de communication par Skype en "haute qualité" est de 500kb/s (soit 62,5Ko/s).

<https://support.skype.com/fr/faq/FA1417/quelle-est-la-quantite-de-bande-passante-necessaire-a-skype>

- ➔ Calculez le taux de compression nécessaire sur ce flux vidéo pour qu'il puisse être utilisé sur internet.

### Capture d'image et conversion

- Capturez une image de votre choix par la caméra. (Si l'image est enregistrée en png, convertissez la en .jpg à l'aide de Gimp).
- Chargez l'image sous Matlab (fonction `imread()`). Observez qu'une image couleur est stockée sur une matrice de taille  $N_x \times N_y \times 3$ .
- Utilisez la fonction `imshow()` pour afficher l'image couleur.
- Utilisez cette même fonction pour afficher séparément la composante rouge, verte, et bleue de l'image (affichage en niveau de gris).
- Sauvez votre image au format .ppm (format d'image non compressé) en utilisant la fonction `imwrite()`.
- Comparez les tailles des fichiers jpg de la capture de la caméra, et de l'image ppm. En déduire le taux de compression entre l'image .jpg et le fichier .ppm.

La fonction `imfinfo` de Matlab permet d'obtenir des informations sur le fichier d'une image.

- Obtenir des informations sur votre fichier
- Calculez en octets et kilo-octets la taille du fichier à partir: du nombre de pixels de l'image et de la taille de l'entête.

### Encodage couleur

Les paramètres du flux vidéo (sous VLC) indiquent également l'encodage des couleurs, ici *Packed YUV 4:2:2*

Premièrement, YUV signifie que l'information de couleur n'est pas transmise pour le canal, rouge, vert et bleu, mais dans un autre **espace couleur** appelé **luminance chrominance**.

### Généralités sur les espaces couleurs

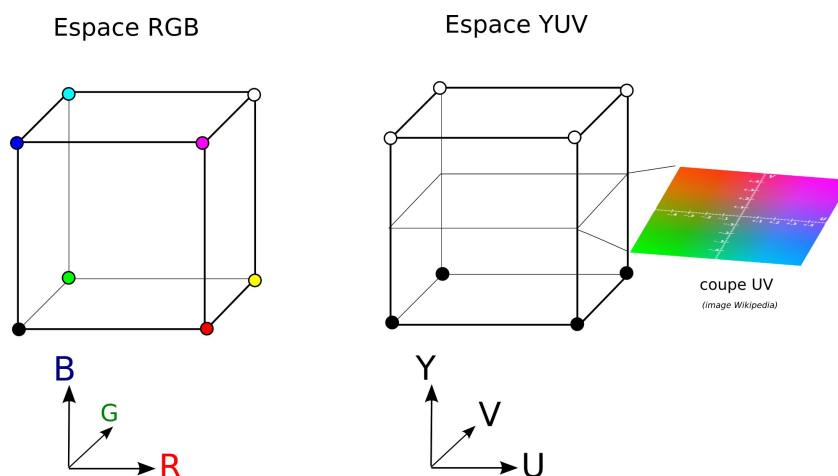
En effet, l'espace (r,g,b) n'est pas l'unique moyen d'encoder une information de couleur. On pourra retrouver classiquement l'espace *CMYK* (cyan, magenta, yellow, black) utilisé pour les imprimantes afin de modéliser plus aisément la synthèse soustractive, et l'espace *YUV* de luminance chrominance pour le transfert d'information. On pourra également trouver les espaces *HSV* (couleur, saturation, valeur) pour les logiciels d'arts et les espaces *Pantones* pour les espaces colorimétriques dites à haute fidélité.

Tous comme un vecteur spatial, la couleur d'un pixel peut être exprimée dans l'un de ces espaces à l'aide d'un changement de base lorsque celui-ci s'exprime de manière linéaire.

**Espace YUV** (facultatif, réalisez les questions de cette partie à la fin si il vous reste du temps)

La luminance correspond à l'intensité lumineuse, c'est à dire aux zones clairs et sombres d'une image. Intuitivement, il correspond à la vision "noir et blanc" d'une image couleur. La luminance est encodé par le premier paramètre dans l'espace YUV.

La chrominance correspond à l'information de couleur et de saturation, indépendamment de sa luminosité. Il correspond aux deux autres paramètres U et V. U représente principalement la différence entre le bleu et la valeur de luminance, alors que V représente la différence entre le rouge et la valeur de luminance.



Représentation des espaces RGB et YUV.

Le passage d'une couleur de l'espace (Y,U,V) vers (R,G,B) correspond à une transformation linéaire défini par la matrice suivante:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

- (facultatif) Exprimez la valeur de la luminance (Y) en fonction des canaux R,G, et B.
- (facultatif) Pourquoi la luminance n'est-elle pas calculée comme étant égale à  $1/3 \times (R+G+B)$  ?

Notre perception des détails est généralement plus sensible aux informations de luminance que celles de chrominance. Il est alors possible de perdre une partie des informations de chrominances sans que nous ne n'y soyons sensible. Cette approche est largement utilisé dans les principes de compressions.

Dans notre cas, l'information YUV 4:2:2, indique que l'on sous échantillonne l'information de chrominance par 2 dans la direction horizontale par rapport à l'information de luminance.

(Pour une ligne de 4 pixels, on ne garde que 2 informations de chrominances).

Une compression par un facteur 4 est alors obtenue immédiatement sans perte notable de la qualité de l'image.

→ (facultatif) Chargez une image couleur sous Matlab.

Le programme `color_convert.m` permet de sous échantillonner une image dans l'espace YUV. L'image 1 correspond à un sous-échantillonnage de la chrominance par un facteur 4 sur chacun des canaux. L'image 2 correspond à un sous-échantillonnage spatial simple. L'image 3 correspond à l'image d'origine.

→ (facultatif) Observez le résultat pour différentes valeurs de sous-échantillonnage (2, 4, 8, etc). Observez que l'image sous échantillonnée dans l'espace de chrominance semble moins pixelisée que l'image sous-échantillonnée spatialement.

La fonction `colorspace_transform` permet d'appliquer un changement de base sur la couleur des pixels.

→ (facultatif) Appliquez cette fonction sur votre image pour la transformer de l'espace RGB vers YUV, puis observez chaque canal de l'image. Pour cela, on affichera en noir et blanc chaque canal de l'image encodé dans l'espace YUV.

## **Enregistrement d'une vidéo**

Pour enregistrer une vidéo dans un fichier à partir de VLC:

### **Sous VLC**

- Open Media
  - Capture Device
    - Changez Play en Convert

→ Observez les différents formats d'enregistrement vidéo.

Note. Les formats sont définis par la syntaxe suivante:

[type de compression video] + [type de compression audio] (Conteneur)

### **Type de compression video**

Les formats de compressions vidéos sont souvent désignés par le terme codec (**co**mpression - **de**compression), et correspondent à différents encodages de compressions d'un flux vidéo.

On pourra citer entre autre:

*MPEG-2*: format de compression vidéo ancien

*H.264*: Format plus récent très répandu.

*Theora*: Format de compression libre au contraire du H.264 associé à de nombreux brevets.

Ces formats définissent comment le flux vidéo est compressé. On trouvera dans ces approches plusieurs étapes de compressions.

Un encodage du mouvement d'une frame (image à un instant donné) à l'autre par une recherche de blocs similaires ou par une détection de mouvement, une analyse prédictive de frames non enregistrés, un encodage de l'erreur pour les frames prédites.

### **Conteneurs**

Les extensions (MPEG4, TS, OGG, AVI, etc) ne sont pas des formats de compressions mais des conteneurs. Ces formats correspondent à une norme d'organisation des données vidéos, audio, et des méta données. Ils définissent par exemple les types de résolutions admissibles, les fréquences possibles et organisent les méta-données tels que les chapitres et sous-titres associées à une vidéo.

Il ne définissent cependant pas la manière dont les données qu'ils contiennent sont compressées qui reste à la charge du codec.

→ Enregistrez une vidéo dans un fichier sur le disque.

### **Transfert analogique**

Lorsque vous capturez une vidéo, observez les différents standards possibles:

On retrouve généralement PAL, SECAM et NTSC

Ce sont des formats de transferts de flux vidéos, ils définissent les formats de communications analogiques, historiquement pour être lues par la télé.

**NTSC**: National Television System Committee

Résolution: 525 Lignes

Fréquence: 30 frames par secondes

Espace couleur: YUV

Format USA, Japon

**PAL**: Phase Alternating Line

Résolution: 625 Lignes

Fréquence: 25 frames par secondes

Espace couleur: YUV

**SECAM:** Séquentiel couleur à mémoire

Résolution: 625 Lignes

Fréquence: 25 frames par secondes

Espace couleur: YDbDr (espace Luminance, chrominance légèrement proche de YUV)

Format Français

### **Qualification d'une caméra à l'aide d'une mire: résolution, linéarité.**

Qu'il s'agisse d'une caméra, d'un détecteur de rayons X... de nombreuses applications requièrent de connaître la réponse du détecteur à un signal d'entrée. Entre autre on s'intéressera à la résolution effective du capteur (jusqu'à combien de paires de lignes par millimètres peut-on discriminer sur l'image caméra ? une scène prise sous un éclairage uniforme se traduit-elle par un fond uniforme sur l'image ? le capteur répond-il linéairement à un dégradé « linéaire » des gris ? quelle gamme de contrastes peut-on distinguer sur l'image caméra et le contraste le plus petit qu'on peut discriminer est-il le même dans les blancs que dans les noirs ?)

Pour quantifier ces différents paramètres on utilise des mires c'est-à-dire des motifs bien connus dont on évalue la réponse sur le capteur.

#### **Analyse qualitative**

- ➔ Prendre une image de la mire. On réglera la distance de sorte que le champ de vision contienne essentiellement la mire et que celle-ci soit éclairée de manière homogène. Déduire à l'œil la résolution de la caméra c'est-à-dire le motif le plus fin, distinguable. On veillera à la mise au point de la focale.
- ➔ La mire montre un dégradé de couleurs. Ce dégradé vous semble-t-il semblable sur l'image acquise ? Qu'en déduisez-vous ?
- ➔ Le contraste entre deux niveaux de gris que vous pouvez distinguer sur l'image est-il le même dans le blanc que dans les noirs ?

#### **Analyse quantitative**

- ➔ Chargez votre image sous Matlab et convertissez celle-ci en niveau de gris.
- ➔ Tracez l'évolution des niveau de gris de cette échelle de niveau de gris en extrayant une coupe dans l'image.
- ➔ A l'aide de Matlab ou d'un logiciel externe tel que Gimp, échantillonnez la valeur du niveau

de gris de l'image pour chaque niveau de gris de la mire.

- Tracez la courbe correspondant au niveau de gris de l'image obtenue en fonction des niveaux de gris de la mire en entrée. Quelle courbe attendriez vous si l'image capturée correspondait à une estimation fidèle de l'intensité de gris de la mire réelle?

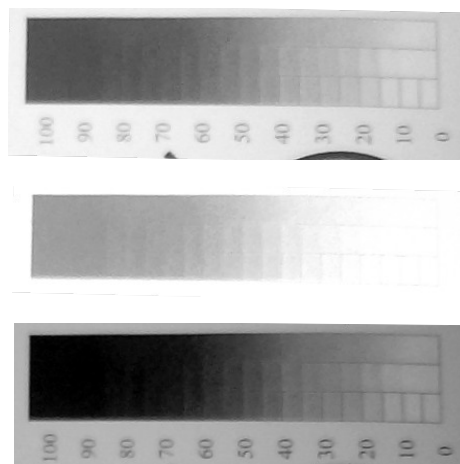
### Outils de modification d'images

VLC dispose de différents paramètres permettant de modifier certaines propriétés de l'image

- Tools
  - Effects and Filters
    - Video Effects

La barre d'édition permet de faire varier différents paramètres de l'image tel que *l'illumination* (*brillance*), le coefficient *gamma*, le *contraste* etc.

- Observez l'influence des différents paramètres.
- Prenez trois captures d'écran d'une même vue de la mire pour lesquels seul le paramètre d'illumination est modifié. La première capture possédera un paramètre d'illumination correct. La seconde sera réalisé avec une forte illumination, et la troisième, avec une faible illumination.



*Exemples de modification d'illumination: Haut: défaut, Milieu: Ajout d'illumination, Bas: Suppression d'illumination.*

- Tracez les 3 courbes des niveaux de gris correspondant à ces 3 niveaux d'illuminations. Quelle transformation de la courbe peut-on associer à un changement d'illumination?



- (facultatif) Réalisez le même test pour le paramètre de contraste, et celui de gamma. Pour chaque cas, déduisez l'influence du paramètre sur le niveau de gris.
- (facultatif) Pour chacun de ces 3 paramètres, déduisez un modèle de fonction  $f$  telle que:  
 $\text{Intensité\_écran} = f(\text{Intensité\_vraie}, \text{paramètre})$   
( $f$  pouvant être un offset, une homothétie, une déformation linéaire, ou une fonction puissance)
- (facultatif) Qu'effectue la fonction `postérisation` de VLC ?

### Traitements supplémentaires sous Matlab (facultatif)

#### Visualisation

Considérons une image de votre choix chargée sous Matlab.

- (facultatif) Modifier la dynamique de l'image pour afficher cette même image entre 0 et 0.5

#### Calcul signal à bruit

- (facultatif) En sélectionnant une partie de l'image de la mire ne contenant pas de motif, calculer le signal sur bruit. On pourra prendre comme estimateur le rapport de la moyenne de l'image sur l'écart-type.

#### Correction d'une dérive d'éclairement

- (facultatif) Dans les mêmes conditions que pour la mire, prendre une image d'une feuille blanche. Vous obtenez ainsi une image de la dérive d'éclairement (`derive.jpg`).
- (facultatif) A partir de l'image '`derive.jpg`' essayer de corriger l'image de la mire par division termes à termes de `mire.jpg./derive.jpg`.

#### Contraste d'une image

- (facultatif) Choisir une ligne (ou une colonne) de l'image coupant un motif « lignes/millimètres » et afficher en 1D cette coupe (niveau de gris en fonction des coordonnées lignes/colonnes). Calculer le contraste (maximum/minimum).

Quelques fonctions utiles sous matlab

- `rgb2gray` : transformation d'une image couleur en une image à niveau de gris
- `size(I)` taille de I (y compris le nombre de canaux de couleurs)
- `imread('fic')` : lecture d'une image
- `imshow(I)` : affichage d'une image