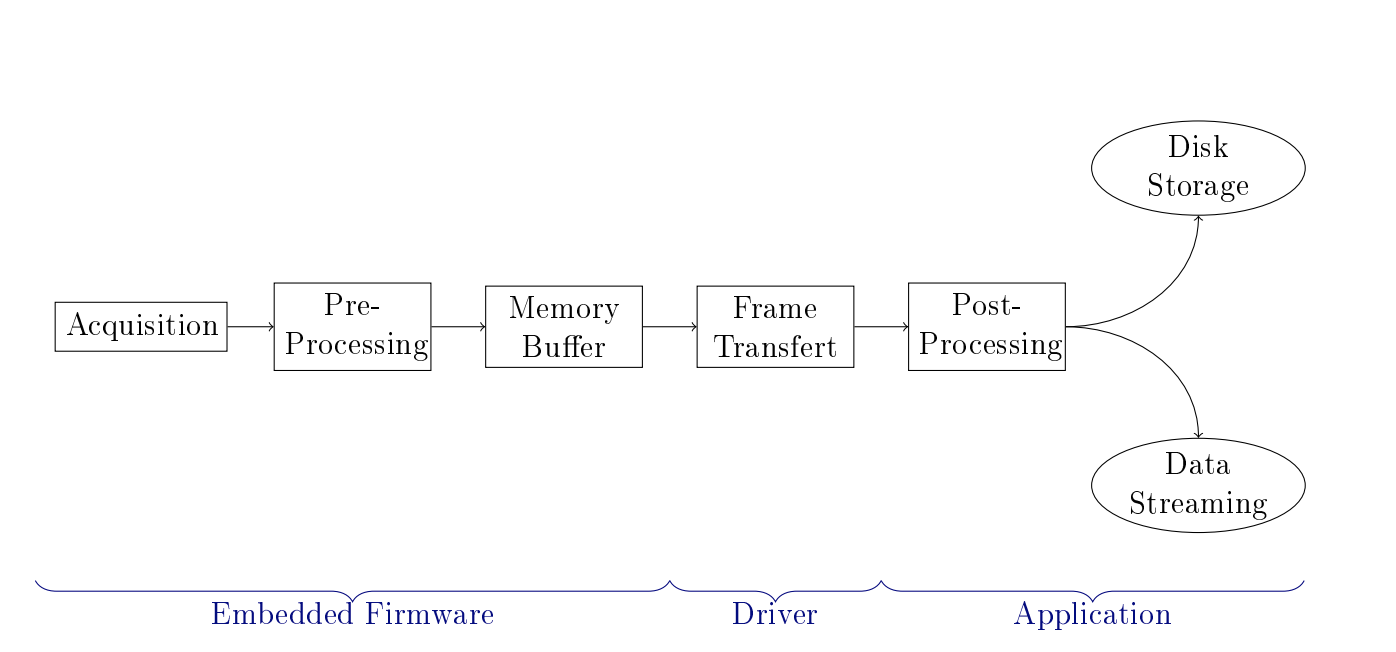
RAPPORT TP FLUX VIDEO

**2 Acquisition et gestion des données**

**2.1 Espace mémoire et débit**



*2. Trouvez la résolution (nombre de pixels) de la prise de vue.*

Dans les propriétés de l’image nous trouvons une résolution de 752 x 416 ce qui correspond au nombre de pixels.

*3. Dans un explorateur de fichier, retrouvez votre image et regardez la résolution et*

*la taille du fichier sauvegardé sur le disque. Comparez les informations avec celles*

*fournies par VLC.*

Nous trouvons une résolution pour l’image enregistré de 8, et la taille du fichier enregistré dans le disque dur est de 380 Ko

/// à vérifier avec VLC ///

*4. Quel est la fréquence de mise à jour des images de la capture vidéo (Codec Information*

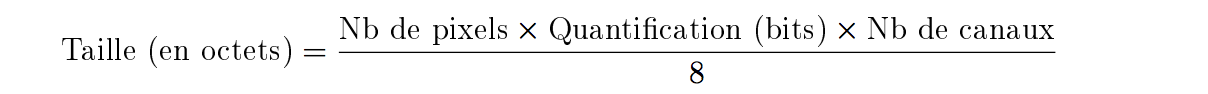
*De VLC) ? Déduisez l'intervalle de temps (en secondes) entre deux images*

*Consécutives.*

D’après le codec VLC, a fréquence de mise à jour des images est de 25 images par secondes ainsi on en déduit l’intervalle de temps entre deux images consécutives : = 0,04 secondes entre deux images.

*5. Calculez la taille prise en mémoire (en Mo) par une image (non compressée).*

Nous savons que la taille d’une image est définie par :

**

Le nombre de canaux est de trois car nous avons un canal par couleur et l‘image est en RGB. De plus la quantification est la discrétisation de la valeur que pourrait prendre un pixel, celui-ci est codé sur 256 niveaux soit un octet par canal.

Ainsi nous avons : soit un petit Méga Octet.

*6. Calculez la bande passante théorique nécessaire pour transmettre l'image de cette*

*webcam sachant que le flux vidéo nécessaire pour avoir une image fluide est d'au*

*moins 25fps. Pour information, un débit de communication par Skype en "haute*

*qualité" est de 500kb/s (soit 62,5Ko/s).*

La bande passante représente l’unité mémoire par seconde. On en déduit donc que la bande passante correspond au nombre d’image par seconde multiplié par la taille de l’image. On obtient donc soit 23Mo/s. Ce résultat est extrêmement élevé, c’est pourquoi il sera nécessaire de compresser ce flux vidéo afin de l’utiliser sur internet.

*7. Calculez le taux de compression nécessaire sur ce ux vidéo pour qu'il puisse être*

*utilisé sur internet.*

Afin d’utiliser ce flux vidéo sur internet il faut donc le compresser. Le taux de compression est de = 375.

*8. Concluez quant au mode de gestion des images par les logiciels d'acquisitions pour*

*pouvoir streamer les images à 25fps*.

Concernant le mode de gestion des images par les logiciels d’acquisitions pour pouvoir streamer les images à 25fps, on conclue que la compression d’une vidéo est différentes de celle d’une image. En effet elle est basée sur une redondance temporelle qui peuvent être supprimer via un pré processing et des algorithmes de prédiction.

**2.2 Compression d'images**

*2. Chargez l'image sous Matlab (fonction imread()). Observez* qu'une *image couleur*

*est stockée sur une matrice de taille Nx × Ny × 3.*

Après avoir exécuter la ligne de code faisant appelle a la fonction imread(), on observe que notre image est stocké dans une matrice de taille 3x3 tel que : 416x752x3

*4. 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 chier .ppm.*

= 70,2 KO

= 938,5 KO

En comparant la taille des fichiers jpg de la capture de la caméra et de l’image préalablement convertit au format ppm, on observe que le taux de compression entre ces deux images est de : = 13,36

*5. Utilisez la fonction imshow() pour afficher l'image couleur.*

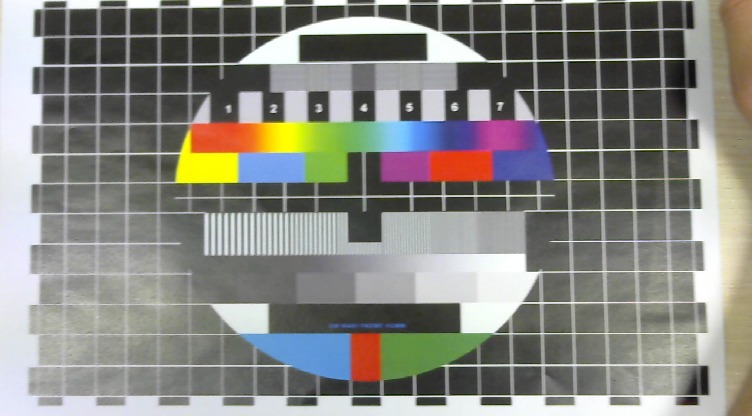
**

Figure - affichage de l'image couleur

*6. Utilisez cette même fonction pour afficher séparément la composante rouge, verte,*

*et bleue de l'image (affichage en niveau de gris).*



Figure - code afin d'afficher séparemment les 3 composantes

A la suite de ce code, on obtient la figure suivante :



Figure - Image en niveau de gris selon les 3 composante rouge vert bleu

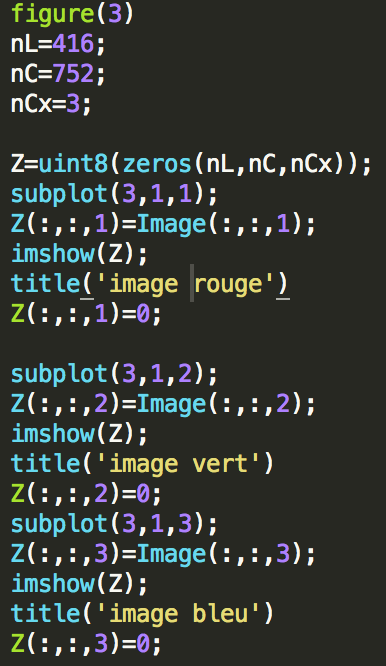
On obtient bien les trois images en niveau de gris selon un canal différent.

On peut alors vérifier que le niveau de gris de la composante verte n’est pas exactement celle attendue. En effet la couleur verte est en réalité un mélange de rouge bleu et de vert.

*7. affichez maintenant les 3 canaux dans leur couleur respective (le canal rouge en*

*rouge, le canal bleu en bleu, . . .).*

Afin d’afficher les trois canaux dans leur couleur on utilise le code suivant



Après ce code on obtient bien la figure correspondante

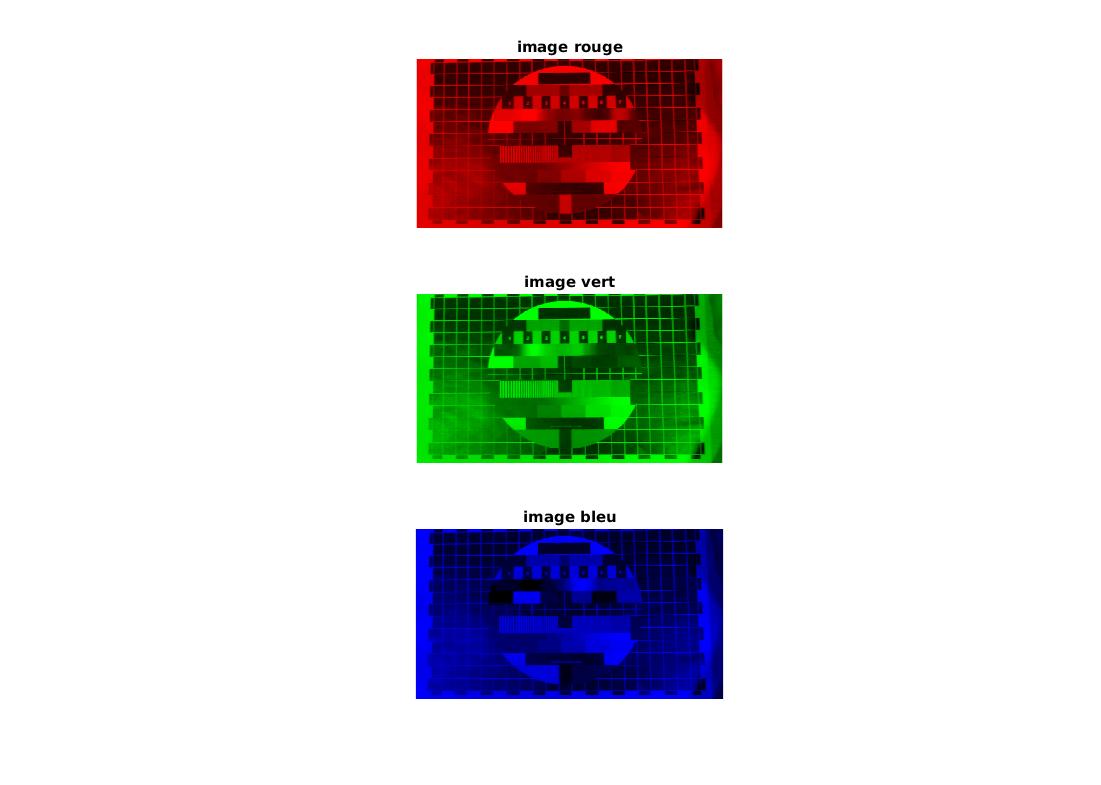
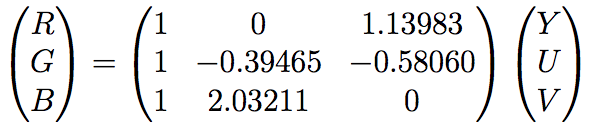


Figure - image représentant les 3 canaux dans leur couleur respective

*8. Exprimez la valeur de la luminance (Y) en fonction des canaux R, G et B.*

D’après l’énoncé, nous avons la relation suivante liant le repère de l’espace RGB et celui de YUV ( Y : luminance, U et V la couleur ) :



En inversant simplement la matrice ci-dessus, on obtient bien la valeur de la luminance (Y) :

Y = 0,299\*R + 0,587\*G + 0,114\*V

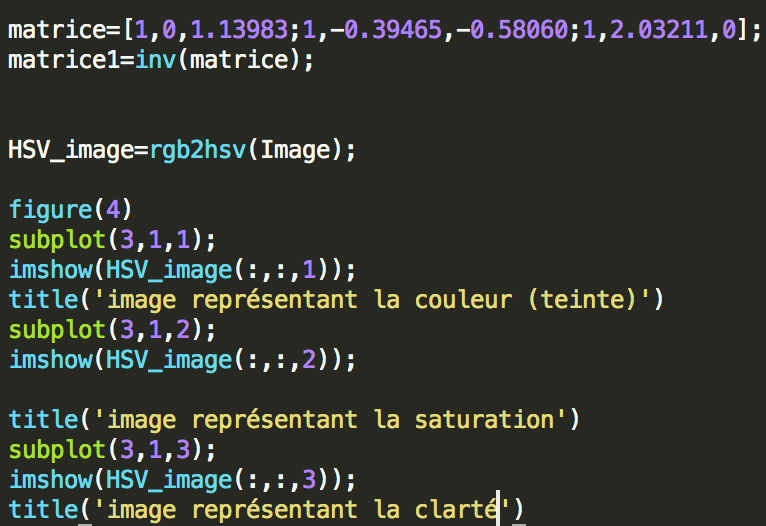
*9. Pourquoi la luminance n'est-elle pas calculée comme étant égale à 1/3 ×(R+G+B) ?*

La luminance n’est pas calculé comme étant égale à 1/3 \* (R+G+B) car la matrice (Y U V) contient plus de détails que la matrice (R G B) et qu’elle n’est pas égale dans tous ces termes .

*10. À l'aide de la fonction Matlab rgb2hsv(), convertissez l'image RGB en HSV.*

*Affichez chacun des canaux en niveaux de gris, et interprétez les images obtenues.*

On utilise le code suivant :



Après ce code, on obtient bien la figure représentant l’image en HSV :

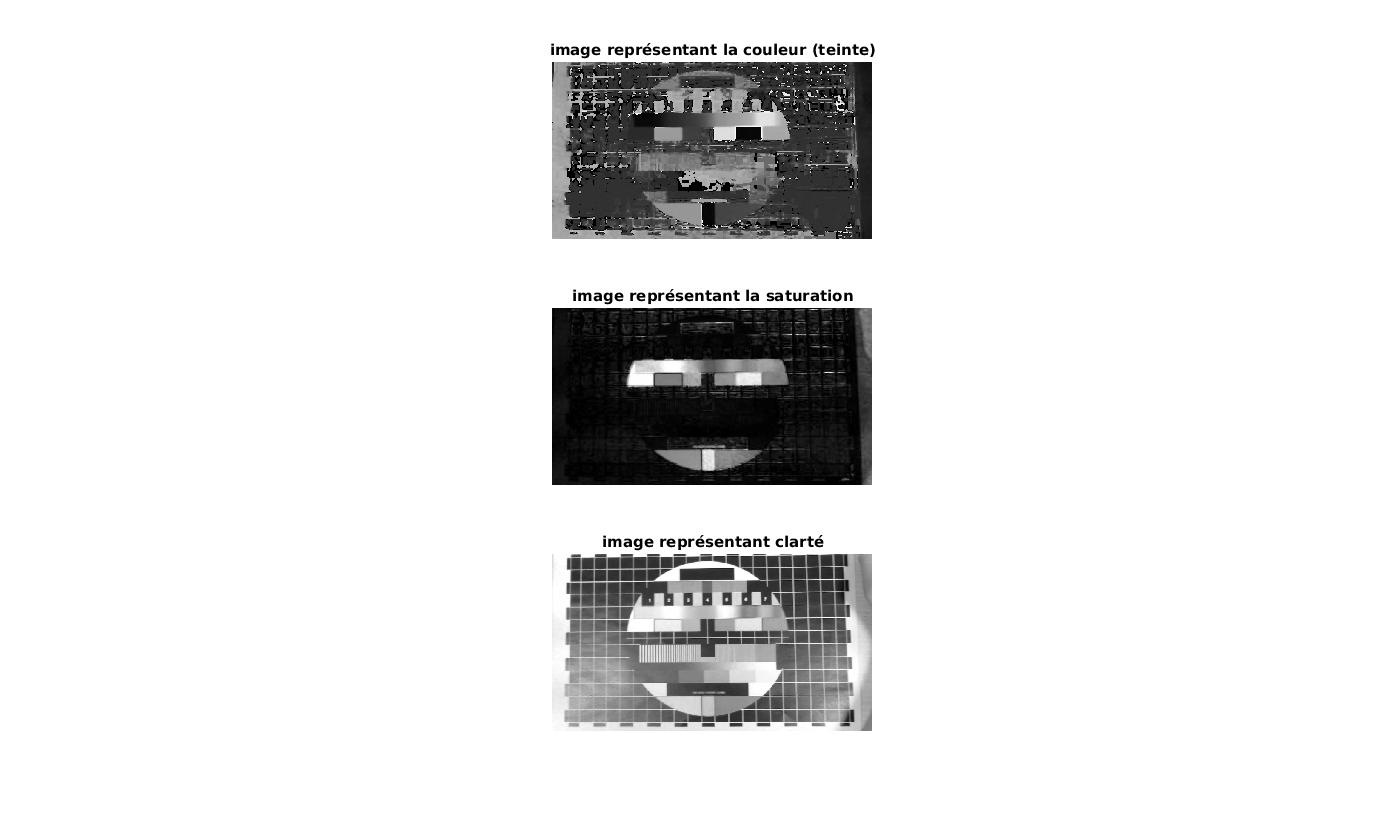


Figure - image en HSV

On observe sur cette image les trois différentes caractéristiques à savoir la couleur, la saturation ainsi que la clarté. Sur la première image on observe que le rouge est représenté comme le plus bas niveau de gris alors que le violet comme le plus haut. Concernant la saturation, on observe que plus la couleur est vive plus la saturation est élevé. Par exemple le rouge est très saturé car il est extrêmement vif et inversement, il y a beaucoup de noir car les couleurs pales et sombre sont peu saturé.