Application à un système vidéo

Romain Raveaux

Sommaire :

But du TP

1°) Visualiser les images issues de la camera.
2°) Effectuer un traitement d'image (filtrage) en avec une implémentation en Java
3°) Monitorer l'exécution en calculant le nombre de Frame Per Second

Introduction

Ce TP se fera avec Eclipse+ADT car l'appel à du code natif (c/c++) n'est pas complétement supporté avec Android Studio (c'est possible mais il faut bidouiller un peu).

Il existe plusieurs bibliothèques permettant de capturer le flux vidéo de la camera. On citera les 2 API natives d'Android : 1°) l'API camera1 disponible depuis Froyo mais peu performante et 2°) l'API Camera2 disponible depuis Lollipop paramétrable à souhait mais complexe à prendre en main. Dans ce TP, nous opterons pour une bibliothèque externe appelée OpenCV offrant un bon compromis entre performance et rapidité de mise en œuvre.

Installation de OpenCV manager

Dans un premier dans temps, il faut installer OpenCV Manager sur votre tablette. OpenCV Manager permet aux applications installées sur votre tablette de bénéficier des services de traitements d'images offerts par OpenCV.
Dans votre répertoire ' « /home/polytech/dev/OpenCV-android-sdk/apk/ » , vous trouverez l'APK convenant à la tablette : « OpenCV_3.1.0_Manager_3.10_armeabi-v7a.apk ».

Vous utiliserez ADB pour faire l'installation cet APK.
Donner la ligne de commande permettant de faire cette installation.
PS : « L'installation peut prendre quelques minutes »

Démarrage avec Eclipse

Ouvrez le projet OpenCV Library, il s'agit du projet de la bibliothèque OpenCV.
Ouvrez le projet PolytechVideoProcessing
Exécuter sur la tablette l'application.

Mesurer le nombre de frame par seconde.
Expliquer le cycle de vie de l'application ?
Expliquer le rôle des méthodes onCameraViewStarted et onCameraFrame ?

Traitement d'image en JAVA

Le tableau outarray contient les pixels de l'image dans une structure à une dimension.
Pour obtenir le pixel aux coordonnées x, y, il faut calculer un index dans le tableau :
int index = y*w+x
Où w est la larheur de l'image.
Dans la suite du TP, il est impératif d'utiliser la structure outarray et non pas la structure Mat d'OpenCV. Il est aussi impératif d'implémenter les traitements d'image suivants :

Premier Traitement : Gradient
Le gradient est un traitement simple. Il consiste à faire la différence entre de 2 pixels proches dans l'image (I).
Le gradient horizontal du pixel x,y peut s'écrire : GradH(x,y) = I(x-1,y)-I(x+1,y)
Le gradient vertical du pixel x,y peut s'écrire : GradV(x,y) = I(x,y-1)-I(x,y+1)
Le gradient combine GradH et GradV : Grad(x,y) = GradH(x,y)+ GradV(x,y)
A votre avis que doit on visualiser si l'on applique le gradient sur une image ?
A faire : Implémenter le gradient et visualiser le résultat sur le flux vidéo
Astuce d'implémentation : N'oubliez pas qu'un pixel est compris entre 0 et 255. Les valeurs négatives devront être changées par leurs valeurs absolues.
Astuce d'implémentation numéro 2: Le gradient doit être calculée sur l'image originale. Il faudra donc deux images pour chaque frame reçue: L'image d'entrée et l'image de sortie calculée.

Dans la console (logcat) et dans à l'écran, vous verrez le nombre de frame par seconde s'afficher. Mesurer le nombre de frame par seconde avant et après l'application du gradient.

Deuxième Traitement : Filtre de Sobel
Implémenter le gradient de Sobel
Vous avez plus de détail sur https://fr.wikipedia.org/wiki/Filtre_de_Sobel
Pour l'implémentation vous allez avoir besoin de la notion de produit de convolution.
Un produit de convolution peut être vu comme une somme de produits scalaires.

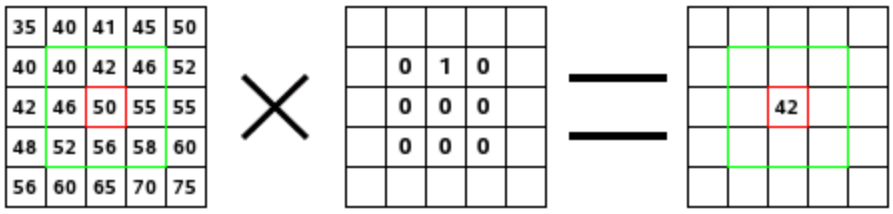

Figure 1 source : https://docs.gimp.org/fr/plug-in-convmatrix.html

Le pixel initial a pris la valeur 42 : (40*0)+(42*1)+(46*0) + (46*0)+(50*0)+(55*0) + (52*0)+(56*0)+(58*0) = 42
A faire : Implémenter le gradient de Sobel
Astuce d'implémentation : N'oubliez pas qu'un pixel ne peut pas faire plus de 255. Il faudra normaliser le résultat final.
Mesurer le nombre de frame par seconde

Conclure sur la charge que représente chaque traitement ?