

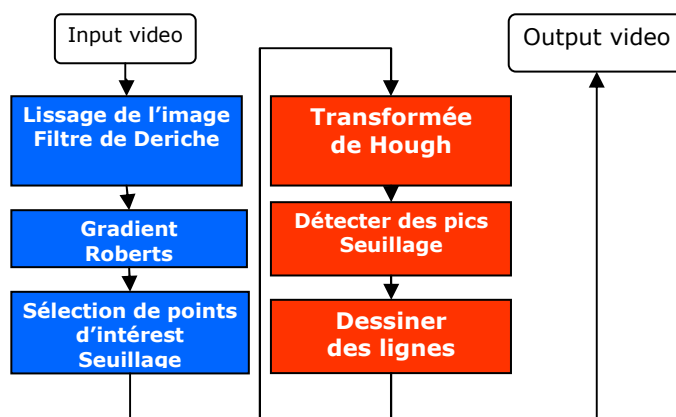
Projet IF4-ARCH

Méthodologie d'optimisation

Objectif : Implémenter et optimiser une chaîne de traitement d'image sur DSP TMS320C6437 en mettant en œuvre des techniques d'optimisations présentées en cours : optimisation algorithmique, optimisation pour les processeurs RISC et VLIW, utilisation des DMA.

Présentation de la chaîne de traitement d'image : Il s'agit d'une suite des opérateurs de traitement d'image qui réalise l'application de détection des lignes par la transformée de Hough. Cette application a été présentée en cours IF4ARCH : Méthodologie d'optimisation. Elle est composée des opérateurs suivants :

1. Filtre de Deriche et le gradient de Roberts
2. Binarisation de l'image
3. Transformée de Hough
4. Détection des « pics » par binarisation de l'espace accumulateur
5. Calcul et affichage des lignes détectées



Travail à effectuer :

Le travail sera réalisé par équipes de 4 personnes. La répartition des tâches en équipe sera la suivante :

- Opérateurs 1-2 : 2 personnes
- Opérateurs 3-5 : 2 personnes

Étapes à respecter :

1. Donnez les caractéristiques du système que vous utilisez :
 - Fréquence de fonctionnement du processeur
 - Caractéristiques VLIW (nombre de voies, nombre des unités de calculs et leurs types)
 - Système de mémoire (taille, hiérarchie)

Pour chaque opérateur estimez le nombre d'opérations théorique par pixel et le nombre d'opérations par seconde. Estimer le nombre d'accès aux données par pixel et par seconde, en fonction de la cadence souhaitée.

Calculez le nombre de pixels et le nombre d'accès à la mémoire par seconde pour l'ensemble d'opérateurs et comparez-les avec les performances théoriques du DSP.

2. Réalisez une première version sans optimisation des opérateurs et
 - a. Cette première version doit contenir
 - i. Version Garcia Lorca du filtre de Deriche avec les calculs en virgule flottante
 - ii. Transformée de Hough utilisant le gradient avec les fonctions trigonométriques appelées dans la librairie mathématique et faire les calculs en virgule flottante
 - b. affichez le résultat en sortie de chaque opérateur
 - c. mesurer les temps d'exécution de chaque opérateur et l'application entière. Créez un tableau de profiling qui mettra en évidence le temps de chaque opérateur et le pourcentage que cet opérateur représente dans toute l'application.
3. Réalisez une deuxième version sans optimisation des opérateurs et
 - a. Cette première version doit contenir
 - i. Version Garcia Lorca du filtre de Deriche avec les calculs en entier
 - ii. Transformée de Hough utilisant le gradient avec les fonctions trigonométriques précalculés dans un LUT
 - b. affichez le résultat en sortie de chaque opérateur
 - c. mesurer les temps d'exécution de chaque opérateur et l'application entière. Créez un tableau de profiling qui mettra en évidence le temps de chaque opérateur et le pourcentage que cet opérateur représente dans toute l'application.
4. Appliquez des techniques d'optimisation sur la deuxième version
 - a. Optimisation algorithmique pour réduire le nombre d'opérations et minimiser la mémoire allouée
 - b. Optimisation pour des processeurs RISC et VLIW (déroulage de boucle, rotation des registres, etc ...)
 - c. Utiliser les DMA pour retourner les images
5. Pour la version optimisée, évaluez le nombre d'opération par pixel et par seconde, évaluez le nombre d'accès aux données.
6. L'implantation sera validée par les enseignants en dernier TP du projet.
7. Rédigez un rapport qui va présenter :
 - a. Les algorithmes et les implantations réalisées
 - b. Des informations demandées dans les points 1-3
 - c. Dessiner des courbes montrant les gains de temps d'exécutions en % pour chaque opérateur en fonction des tailles des images et le gain sur la chaîne complète
 - d. Explication et justification des choix algorithmiques et d'implantation
 - e. Code des opérateurs avec commentaires
8. Soutenance : 15 minutes par groupe, présentation en ppt devant l'ensemble des élèves.