

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 :

1. optimisation algorithmique,
2. optimisation pour les processeurs RISC et VLIW,
3. utilisation efficace des ressources matérielles (par exemple DMA).

Réalisation :

1. Organisation et évaluation

Le travail est à réaliser par équipe de 2 personnes. L'implantation sera validée par les enseignants lors de la dernière séance de projet. Rédigez un rapport qui va présenter :

- a. Les algorithmes et les implantations réalisées,
- b. Les explications et justifications des choix algorithmiques et d'implantation
- c. Les résultats obtenus pendant toutes les étapes du développement du projet,
- d. Les graphes, courbes et tableaux montrant les gains de temps d'exécutions en % pour chaque opérateur,
- e. Le code des opérateurs avec les commentaires.

Une soutenance de 15 minutes (stricte) par binôme sera organisée après la fin des projets (présentation en ppt devant l'ensemble des élèves).

2. Chaîne de traitement d'image à implanter

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

- a. Filtre de Deriche, version optimisée par Garcia Lorca
- b. Binarisation de l'image
- c. Transformée de Hough
- d. Détection des maxima par binarisation de l'espace accumulateur
- e. Calcul et affichage des lignes détectées

3. Étapes du développement à respecter

a. Préparation théorique

- i. Étudiez les ressources et caractéristiques matérielles 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).
- ii. Pour chaque opérateur estimez le nombre d'opérations théorique par pixel et le nombre d'opérations par seconde. Estimez 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.

b. Première implantation « basique »

- i. Réalisez une première version sans optimisation des opérateurs qui composent la chaîne de traitement définie. Cette première version va utiliser :
 1. les calculs en virgule flottante pour les opérateurs qui en ont besoin.

2. les fonctions trigonométriques appelées dans la librairie mathématique
3. la transformé de Hough qui sera calculée selon l'algorithme exhaustif.
- ii. Effectuez les mesures des temps d'exécution de chaque opérateur ; 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.
- c. Deuxième version optimisée
 - i. Appliquez des techniques d'optimisation sur la deuxième version, appliquez l'optimisation algorithmique pour réduire le nombre d'opérations et minimiser la mémoire allouée ; Pour cela, lisez les articles cités dans les références et mettez en œuvre les optimisations qui y sont proposées :
 - ii. Optimisation pour des processeurs RISC et VLIW (déroulage de boucle, rotation des registres, etc ...)
 - iii. Pour la version optimisée, évaluez le nombre d'opération par pixel et par seconde, évaluez le nombre d'accès aux données.
 - iv. Effectuez les mesures des temps d'exécution de chaque opérateur ; 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.
- d. Troisième version avec optimisation de précision
 - i. Cette version doit contenir les calculs en entier (virgule fixe) qui remplacera les calculs en virgule flottante
 - ii. Calculs trigonométriques pré calculés dans une LUT
 - iii. Utilisez les DMA pour retourner les images
 - iv. Effectuez les mesures des temps d'exécution de chaque opérateur ; 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.

Références :

T. Ea, L. Lacassagne, P. Garda, Execution temps reel des detecteurs de contours de Deriche par des processeurs RISC, Congrès Adéquation Algorithme Architecture, 1998, France (<http://www.ief.upsud.fr/~lacas/Publications/AAA98.pdf>)

F. Lohier, L. Lacassagne, P. Garda, Programming techniques for real time software implementation of optimal edge detectors: a comparison between state of the Art DSP and RISC architectures", DSP World, 1998 (<http://www.ief.upsud.fr/~lacas/Publications/DSPWorld98.pdf>)

D. Demigny, F. G. Lorca, L. Kemal et J.P. Cocquerez, Conception nouvelle du détecteur de contours de Deriche, Symposium on signal and Image Processing, GRETSI, 1995 (<http://documents.irevues.inist.fr/bitstream/handle/2042/2030/006.PDF%20TEXTE.pdf?sequence=>)

F O'Gorman, M Clowes, Finding picture edges through collinearity of feature points (1973), Third International Joint Conference on Artificial Intelligence (<http://www.ijcai.org/Past%20Proceedings/IJCAI-73/PDF/058.pdf>)