

Optimisation des temps de calcul et processeur RISC
Projet "Vision"

→ **Objectifs**

L'objectif de ce projet est d'expérimenter des techniques d'optimisation dans le contexte d'une chaîne de traitement d'image embarquée sur la carte SABRE/freescale IMX6 (ARM_Cortex-A9, quad-core) :

1. optimisation algorithmique (algorithme de complexité inférieure au sens mathématique, structures de données adaptées)
2. optimisation pour les processeurs RISC (déroulage de boucle),
3. optimisation pour une utilisation efficace des ressources matérielles : accès aux données, gestion de la mémoire cache, multi-thread.

→ **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.

Voici quelques conseils supplémentaires concernant des éléments à mettre dans le rapport :

- L'architecture des systèmes utilisés (processeur, fréquence, hiérarchie mémoire, systèmes d'exploitation etc...),
- Les algorithmes et les implantations réalisées, les explications et justifications des choix algorithmiques et d'implantation,
- Les résultats obtenus pendant toutes les étapes du développement du projet :
 - Les graphes, courbes et tableaux montrant les résultats de profilage
 - Tableaux des gains de temps d'exécutions en % pour chaque opérateur réalisé
 - Le code des opérateurs avec les commentaires (annexe)

Une présentation de 2 slides par binôme sera organisée à la fin des projets, derrière le PC, pendant démonstration.

→ **Prise en main de l'environnement**

Vous allez d'abord les PC pour développer le programme et réaliser les optimisations. Puis nous allons utiliser les cartes SABRE/freescale IMX6 (et/ou Rpi), connectées à votre PC par une connexion réseaux, télécharger des fichiers, compiler les fichiers à distance et exécuter le programme sur les cartes pour mesurer les temps d'exécution non optimisés et optimisés sur des cibles embarquées.

1. Télécharger le fichier source « projet_base_inf.cpp » de la version de base du projet, ce projet est déjà paramétré pour utilisation de la librairie OpenCV
2. Pour exécuter le programme il faut copier vos fichiers sources dans le répertoire de travail

→ **Travail à réaliser en première partie du projet**

Le projet de base contient les étapes d'acquisition, filtre médian, filtre de sobel et d'affichage des images qui sont assurées avec les fonctions OpenCV. Mésurez les performances du projet original sur le PC. Puis remplacez le médian et sobel par vos propres fonctions du median et sobel.

Méthodologie :

- 1. Réaliser une évaluation de temps d'exécution des filtres en version "naïves"** produisez le graphe pour différentes valeur de taille de filtre median. Représentez les résultats obtenus à l'aide d'un graphique.
- 2. Optimisez filtre médiane**
 - a. essayez appliquer les techniques d'optimisations apprises en cours pour accélérer les temps d'exécutions.
 - i. à chaque fois, mesurez le temps d'exécution de cette fonction pour différents **k**, **tracer la courbe dans le même graphique, créez des tableaux comparatifs.**
3. Une fois que vous avez épuisé toutes les possibilités d'optimisation comme déroulage de boucles, réorganisation de code etc... Vous pouvez tester les options d'optimisation du compilateur GCC pour la version la plus rapide.

Note :

- 1. Pour chacune des étapes du développement demandé, ne pas oublier de vérifier que le résultat obtenu est correct.**

→ **Travail à réaliser en deuxième partie du projet**

- **effectuez des mesures des temps d'exécution pour différentes cibles fournies**
- **réalisez les tableaux comparatifs des caractéristiques des architectures et des speedup obtenus pour chaque architecture**

Rédiger un rapport pour présenter votre travail et résultats obtenus.