

Trabajo Práctico 2

Procesamiento de imágenes para la detección de bordes

Optimización utilizando instrucciones SIMD

Organización del Computador 2

2do. Cuatrimestre 2009

1. Enunciado

Optimizar las funciones de procesamiento de imágenes desarrolladas en el primer trabajo práctico. Para eso se debe utilizar el modelo de programación SIMD (*Single Instruction Multiple Data*) y el set de instrucciones SSE de la arquitectura IA-32 de Intel. Además de los operadores de derivación ya vistos (Roberts, Prewitt y Sobel) para esta nueva entrega se pide implementar también el filtro de Frei-Chen (ver Introducción Teórica del TP1) que requiere un procesamiento en punto flotante.

Al igual que en la entrega anterior, se debe escribir la parte de interacción con el usuario y manejo de archivos en lenguaje C, utilizando la librería **openCV**. Las funciones de procesamiento de imágenes deben estar escritas en lenguaje ensamblador optimizado para instrucciones SIMD. El programa debe recibir en la línea de comandos el nombre del archivo de entrada y la operación:

- r1 = realzar bordes con el operador de Roberts.
- r2 = realzar bordes con el operador de Prewitt.
- r3 = realzar bordes con el operador de Sobel, derivación sólo en x.
- r4 = realzar bordes con el operador de Sobel, derivación sólo en y.
- r5 = realzar bordes con el operador de Sobel, derivación en x y en y.
- r6 = realzar bordes con el operador de Frei-Chen.

Nota

Se puede asumir que las imágenes con las que vamos a trabajar tienen un ancho de línea múltiplo de 16 pixeles, para poder realizar el procesamiento integramente con registros XMM.

Comparación de performance

Como parte de este trabajo práctico debe compararse el tiempo de ejecución (medidos en ciclos de reloj) de las dos implementaciones (la correspondiente al TP1 y al TP2) de los diferentes filtros. También se deben comparar los tiempos de ejecución entre el operador de Sobel

implementado con instrucciones SIMD y el operador de Sobel implementado en la librería `openCV`.

Informe

Debe reflejar el trabajo hecho para obtener el resultado, las decisiones tomadas (con el estudio de sus alternativas), las estructuras de datos usadas (con gráficos y/o dibujos si ayudan a clarificar), las pruebas que hayan hecho para tomar decisiones o al final para buscar errores en el producto final, etcétera. También se debe incluir los datos de la comparación de tiempos, realizada de forma estadística (el menor tiempo de n corridas, el promedio, etc.). Debe contar como mínimo con los siguientes capítulos: introducción, desarrollo, discusión y conclusiones. Estar estructurado top-down o sea leyendo la introducción se debe saber qué se hizo y cuáles son las partes más importantes. Después de leer los primeros capítulos se debe saber cada cosa que se hizo y como se hicieron las más importantes. En el resto de los capítulos se debe poder leer el detalle de todo lo hecho. Además, el informe debe incluir:

- Carátula con número del grupo y los nombres de los integrantes con número de libreta y email
- Manual del usuario
- Algunas imágenes con las que hayan probado y su correspondiente procesamiento.
- Instrucciones para el corrector, por ejemplo como ensamblar los archivos fuente para obtener el ejecutable (Makefile).
- Lista de todos los archivos entregados.

El informe se evalúa de manera independiente del código. Puede reprobarse el informe y en tal caso deberá ser reentregado para aprobar la materia.

Entrega

La fecha de entrega de este trabajo es martes 27 de octubre, en el horario de clase (de 17 a 21 hs). La entrega se realizará en un CD que debe incluir, los ejecutables, todos los archivos fuentes necesarios para crearlos, imágenes de prueba con su correspondiente procesamiento y el informe en **pdf**. Si desarrollaron prototipos en lenguaje C para resolver el trabajo primero en alto nivel, deben entregarlos e incluir los resultados obtenidos en el informe. Para ordenar la entrega se deben crear las siguientes carpetas en el CD: **src**, **exe**, **enunciado**, **informe**, **resultados**.