



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Operaciones con SIMD

Organización del Computador II
Segundo Cuatrimestre de 2016

Integrante	LU	Correo electrónico
Núñez Morales, Carlos	732/08	cdani.nm@gmail.com
Martinez Quispe, Franco	025/14	francogm01@gmail.com
Mirko Yves Bahoz Torrico	28/10	mirko.torrico@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

1. Implementacion

1.1. Explicacion general

En este trabajo practico se busca desarrollar ciertas funciones usando las instrucciones de SSE. Se utilizaron estas instrucciones para la implementacion de una serie de filtros que procesan imagenes en formato BMP, haciendo el desarrollo en lenguaje ASM. Adicionalmente se tuvo que implementar versiones en lenguaje C, para poder hacer comparaciones con las anteriores.

1.2. Detalles de implementacion

1.2.1. convert

Este filtro busca tomar la imagen compuesta de formato de colores RGB y realizar una transformacion lineal que corresponde al formato YUV, y ademas realizar el mismo trabajo para convertir imagenes YUV a RGB.

En la version C, se trabajo recorriendo cada pixel y aplicando las operaciones descritas en el enunciado de modo que, estas formen el conjunto de colores YUV, o RGB dependiendo de la conversion, que constituyen los colores de cada pixel en la imagen de salida. Siendo C, las operaciones se aplicaron linealmente, teniendo en cuenta la saturacion de colores, lo que se cuida mas en la conversion RGBaYUV.

En la version ASM, se trabajo de la misma manera, pixel a pixel. Usando los registros de SSE, se tuvo que extender los valores de los colores del pixel a trabajar para poder realizar los calculos de conversion de colores. Si bien la idea parecia extender los colores del pixel de byte a word, en nuestro caso se extendieron los colores a dword para poder tener una mejor precision y resultado en las operaciones de conversion. Al hacer el filtro, se noto que las operaciones tenian mas caracter de precision aritmetica que logica, la cual esta ultima fue la primera impresion al empezar a trabajar. Por esta razon se tuvo que realizar dicha extension a dword.

1.2.2. fourCombine

En este filtro se tomaron pixeles de la imagen de modo que se ordenaran en distintos cuadrantes de la imagen de salida.

En la version C, se itera tomando un cuadrante de 2x2 pixeles, y se mueve el pixel correspondiente a los punteros que corresponden a los cuadrantes de la imagen de salida.

En la version en ASM, se trabajo de la misma manera, pero se usaron 2 registros xmm para guardar 2 pixeles de cada fila del cuadrante. Se tuvo que calcular el offset correspondiente a cada cuadrante de la imagen de salida antes de la iteracion de los pixeles. Si bien el corrector puede notar que se uso `pshufd` en vez de una operacion de shifteo, se aclara que el que desarrollo este filtro tuvo algunos problemas al notar diferencias entre las operaciones de shifteo y el buscar una en el manual.

1.2.3. linearZoom

1.2.4. maxClose

2. Analisis preliminar

2.1. Comparacion C vs ASM

En esta seccion vamos a mostrar las diferencias de rendimiento de los filtros entre sus 2 versiones implementadas. Se mostrarán estas diferencias aplicando cada filtro a imágenes de distinto tamaño.

3. Hipotesis de trabajo

3.0.1. Conjunto de ideas de experimentos

Para la experimentación, se decidió hacer versiones distintas de los filtros en ASM. Estas nuevas versiones resuelven el mismo problema mediante un acercamiento distinto, ya sea utilizar instrucciones diferentes, trabajar las operaciones numéricas de alguna manera distinta, o simplemente resolver los algoritmos con instrucciones menos complicadas (aunque hagan al algoritmo más largo).

4. Diseño experimental

5. Resultados y Analisis

6. Conclusiones