

Nº Orden	Apellido y nombre	L.U.	Cantidad de hojas

Organización del Computador 2

Recuperatorio primer parcial — 2 de julio de 2009

Normas Generales

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Está permitido tener los manuales y los apuntes con las listas de instrucciones durante el examen. Está prohibido compartir manuales o apuntes entre alumnos en el examen.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido y LU.
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- La codificación de los ejercicios debe estar debidamente comentada.
- Por favor entregar esta hoja junto al examen.

Ej. 1. (40 puntos)

El *filtro de suavizado de promedio cargado* se utiliza para eliminar ruido en las imágenes. Consiste en aplicar una máscara por convolución a todos los píxeles de la imagen. La máscara que se utiliza es la siguiente:

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Es decir, $b_{i,j} = \frac{a_{i-1,j-1} + 2a_{i-1,j} + a_{i-1,j+1} + 2a_{i,j-1} + 4a_{i,j} + 2a_{i,j+1} + a_{i+1,j-1} + 2a_{i+1,j} + a_{i+1,j+1}}{16}$

Escriba una función en lenguaje ensamblador que, dada una matriz A que contiene los píxeles de una imagen en escala de grises (1 píxel = 1 byte) de dimensión $n \times n$ (n de 16 bits sin signo, múltiplo de 2), aplique el filtro de suavizado y devuelva una nueva matriz B con el resultado de la operación.

El prototipo de la función es: `void suavizar(char* A, char* B, unsigned short n);`

Aclaraciones:

1. Fuera del rango en el cual se puede realizar la convolución, la imagen resultante no se debe modificar.
2. Si bien el resultado del cálculo debe almacenarse en 8 bits, los cálculos intermedios deben realizarse con 16 bits de precisión.
3. Para optimizar el cálculo se pide utilizar las instrucciones SIMD y procesar de a 4 píxeles por ciclo.

Ej. 2. (40 puntos)

En procesamiento de señales (por ejemplo audio) se utilizan *filtros Gaussianos* para no rebasar valores máximo (saturación) y reducir los cambios bruscos de la señal en el tiempo. Un caso particular de filtro Gaussiano es el filtro de Gabor cuya función matemática viene dada por:

$$g(x) = ke^{2\pi \cos(\theta)x - ax^2} \quad \text{con } k, a \in \mathbb{R} \quad y \quad 0 \leq \theta \leq 2\pi$$

Escriba una función en lenguaje ensamblador que dada una señal de audio almacenada en un vector de números de punto flotante de precisión simple de longitud n (n de 16 bits sin signo), aplique el filtro de Gabor para eliminar la saturación. El prototipo de la función es el siguiente:

`void filtroGabor(float* A, float k, float a, float tita, unsigned short n);`

Ej. 3. (20 puntos)

Explique y escriba el código en lenguaje ensamblador correspondiente al siguiente pseodocódigo de un salto condicional: `if (a > 0) then a = b else a = c;`

1. Si los datos `a`, `b` y `c` corresponden a registros de la FPU que contienen números en punto flotante de 32 bits de precisión.
2. Si los datos `a`, `b` y `c` corresponden a registros SSE que contienen 4 números enteros signados de 32 bits de precisión y la comparación `a > 0` se interpreta como que todos los enteros empaquetados del registro `a` deben ser mayores que cero.