

Trabajo práctico 2: “SIMD”

Normativa

Normativa de entrega:

Fecha de entrega: 28/05 hasta las 23:59

Los grupos *DEBEN* ser de a 2 o 3 personas maximo.

Realizar un informe (pdf) o README.md Markdown en git que contenga referencias al código realizado. Mostrar mediante diagramas las pruebas realizadas. Mostrar pruebas que muestren que su código funciona correctamente

Entregar todo el código C y ASM pedido.

El entregable será un link a un proyecto en git que contenga el informe y una carpeta src con todas las fuentes. Además de esto se deberá entregar un archivo bash .sh que compile y ejecute las pruebas que se entregan en el informe. Estas deberán probar imágenes a distintos tamaños (varias veces por tamaño) para los dos algoritmos de Blend así se puede observar la diferencia en velocidad de usar SIMD vs no usarlo a distintas resoluciones. Las pruebas deben guardar en el csv los resultados de cada una así después se puede graficar fácilmente.

El entregable será enviado por mail a la dirección oc2.ungs@gmail.com

El asunto del mail deberá ser: < APELLIDO_ALGUN_INTEGRANTE > -TP2 - OC2

El cuerpo del mail deberá contener los nombres, apellidos y DNI de cada integrante del grupo.

El link al repo

TP2: SIMD

Se desea realizar un sistema para procesar imágenes de tipo BMP de 24 bits El mismo se basa en un header y luego las matrices de colores *RGB*.

https://en.wikipedia.org/wiki/BMP_file_format

Se provee el archivos de ejemplo

1. Realice una función, con la siguiente aridad, void aclarar(unsigned char** red, unsigned char** green, unsigned char** blue, int n) en assembler sin utilizar instrucciones multimedia que le sume n a cada una de las matrices. Deberá saturar correctamente.
2. Realice una función, con la siguiente aridad, void medianFilter(unsigned char** red, unsigned char** green, unsigned char** blue, window) en assembler sin utilizar instrucciones multimedia que genere el filtro de medianas de las matrices.

Para esto lo que hará es, para cada pixel de la imagen, obtendrá la mediana del cuadrado definido por el tamaño *windowxwindow* en ese punto y pondrá dicho color en ese pixel. Podrá no tratar los bordes, es decir que sólo lo aplicará en píxeles que sean interiores a la ventana dados (o sea que si le sumamos el valor de la ventana/2 o se lo borramos, no nos salimos de la imagen)

https://en.wikipedia.org/wiki/Median_filter

<http://www.librow.com/articles/article-1>



3. Realice una función, con la siguiente aridad, void multiplyBlend(unsigned char** red1, unsigned char** green1, unsigned char** blue1, unsigned char** red2, unsigned char** green2, unsigned char** blue2)

en assembler sin utilizar instrucciones multimedia que genere el multiply blending de las 2 imágenes. El resultado lo dejará en la imagen 1. Para esto sólo bastará con hacer la cuenta (pixel a pixel) $f(a, b) = (a * b)$

https://en.wikipedia.org/wiki/Blend_modes



Generará



4. Para el algoritmo del ejercicio 3 (blend) realice su contraparte utilizando operaciones multimedia en ASM
5. Tome medidas de tiempos de comparación entre unos a distintas resoluciones y vea como escalan los distintos algoritmos usando SIMD y SISD. Recuerde que deberá tomar un promedio de muchas mediciones para que el resultado tenga sentido.
6. Bonus: hacer mas filtros, que tal negativo, cambiar contraste etc