

Computación paralela y distribuida

Práctica 1

1. Implementación de un efecto de difuminado de rostros en video

Con estas prácticas se va a comparar el rendimiento de un algoritmo para la difuminación de un rostro en un video, bajo las siguientes condiciones de paralelización:

1. Práctica 1: CPU - secuencial
2. Práctica 2: CPU - OMP (2, 4, 8 y 16 hilos)
3. Práctica 3: GPU - CUDA
4. Práctica 4: Cluster - MPI

El video puede tener cualquier resolución superior a 640x480. Puede hacer uso de una librería para manipulación de imágenes y video como OpenCV. El algoritmo para efecto borroso debe elaborarse. El algoritmo de detección de rostros no es necesario elaborarlo. Se sugiere usar el algoritmo de características HAAR de OpenCV.

Se debe pasar como argumentos del programa el nombre del video de entrada, el nombre del video de salida y si es el caso y el número de hilos. Por ejemplo:

```
cp@desktop:~$time videoFaceBlur videoIn videoOut 4
```

El comando anterior ejecuta la implementación en CPU para el videoIn con 4 hilos. El video resultante se guarda en el archivo videoOut

1.1. Script para ejecución.

Para la ejecución de los programas, elaborar un *script* que permita ejecutar uno a uno los casos y almacenar los resultados de tiempo de respuesta en un archivo de texto plano.

1.2. Consideraciones.

- Especificar la forma en que se hace el balanceo de carga, especificando el método con el que fue particionado el problema (block, block-wise). Hacer dibujos explicativos.
- Hacer uso de punteros y de memoria dinámica (malloc() - free() en *host*).
- Elaborar gráficas donde se observe:
 - El tiempo de respuesta y el número de hilos
 - El *speedup* alcanzado con el número de hilos
- Entregar los siguientes archivos:

```
videoFaceBlur.c  
script_ejecutar_todo  
reporte.pdf  
LEEME
```

Archivo LEEME con observaciones, todos dentro de una carpeta con los nombres que aparecen en el correo para cada integrante. Pej la carpeta podría llamarse: **capedrazab-capedrazab**. Entregar el archivo tar con el mismo nombre de la carpeta.

1.3. Calificación.

Se tendrán en cuenta los siguientes aspectos para la evaluación:

- Funcionamiento pleno del programa (requisito para entrega) 40
- Script de ejecución para todos los programas almacenando el tiempo de respuesta para cada caso. 10 %
- Código limpio. 10 % (modular, tabulaciones, comentarios - básicos, declaración de constantes, buenos nombres para variables, etc.)
- Sustentación por parte de cualquiera de los integrantes del grupo de trabajo. 20
- Reporte. Archivo pdf con formato IEEE puesto en la página donde se hagan explicaciones a cerca del diseño (particionamiento, etc), experimentos, resultados (tiempos de respuesta, speedup) y conclusiones. 20 %