


|   |  |                                   |
|---|--|-----------------------------------|
|  <p>UNIVERSIDAD<br/>FRANCISCO DE VITORIA<br/>VINCE IN BONO MALUM</p> | <p>GRADO EN INGENIERÍA INFORMÁTICA</p> <p>Asignatura: Computación de alto rendimiento</p> <p>Enunciado Trabajo 2</p> | <p>Curso: <u>4</u> Grupo: A/B</p> |
|---|--|-----------------------------------|

## Aplicación de la convolución de matrices al filtrado de imágenes

### Contexto

Las operaciones de convolución de matrices aplicadas al filtrado de imagen y la elección de la matriz kernel son interesantes desde un punto de vista del empleo de técnicas de programación concurrente, dado que ofrecen oportunidades de paralelización realizando una descomposición basada en datos.

Existen una gran variedad de procedimientos que permiten, a partir de una imagen, obtener otra modificada (técnicas de filtrado). Se trata de métodos con los que se puede resaltar o suprimir, de forma selectiva, información contenida en una imagen, para destacar algunos elementos de ella, o también para ocultar valores anómalos. Se pueden distinguir entre filtros de paso bajo, de paso alto, direccionales, de detección de bordes, etc. Algunos de ellos son los métodos de sustracción de la media y los filtros basados en derivadas (ver la Tabla 1)

|  |   |
|--|---|
| <p><b>Filtros de paso bajo:</b></p> <ul style="list-style-type: none"> <li>• Suavizado de imagen</li> <li>• Eliminación de ruido</li> <li>• Ejemplo: Desenfoque</li> </ul> | <p><b>Filtros de paso alto:</b></p> <ul style="list-style-type: none"> <li>• Resalto de zonas de mayor variabilidad</li> <li>• Sustracción de la media</li> <li>• Ejemplo: Enfoque</li> </ul> |
| <p><b>Filtros direccionales:</b></p> <ul style="list-style-type: none"> <li>• Resalto de píxeles que determinan direcciones</li> </ul>                                     | <p><b>Filtros para la detección de bordes:</b></p> <ul style="list-style-type: none"> <li>• Aplicación en ingeniería, estudio del terreno</li> <li>• Ejemplo: Detectar bordes</li> </ul>      |

Tabla 1 – Tipos de filtro.

Una imagen puede interpretarse como una función bidimensional  $z = F(x, y)$  donde  $x$  e  $y$  son coordenadas espaciales, y  $z$  es el valor de la intensidad de la imagen en el punto  $(x, y)$ .

Dada una matriz  $A_{m \times n}$  y una matriz  $C_{(2N+1) \times (2N+1)}$  con  $2N + 1 < m, n$  se define la convolución de las matrices  $A$  y  $C$  como una nueva matriz  $D = A * C$  definida a partir de la expresión

$$d_{ij} = \frac{1}{c} \sum_{r=1}^{2N+1} \sum_{s=1}^{2N+1} a_{i-N+r-1, j-N+r-1} c_{rs},$$

Donde

$$c = \sum_{i,j=1}^{2N+1} c_{i,j} \text{ (si } c = 0 \text{ se toma } c = 1).$$

La matriz  $C$  se le denomina núcleo o kernel de la convolución

La matriz C se le denomina núcleo o kernel de la convolución

$$\begin{array}{|c|c|c|c|c|} \hline 35 & 40 & 41 & 45 & 50 \\ \hline 40 & 40 & 42 & 46 & 52 \\ \hline 42 & 46 & 50 & 55 & 55 \\ \hline 48 & 52 & 56 & 58 & 60 \\ \hline 56 & 60 & 65 & 70 & 75 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline -2 & -1 & 0 \\ \hline -1 & 1 & 1 \\ \hline 0 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & 78 & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array}$$

$$35 \cdot (-2) + 40 \cdot (-1) + 41 \cdot 0 + 40 \cdot (-1) + 40 \cdot 1 + 42 \cdot 1 + 42 \cdot 0 + 46 \cdot 1 + 50 \cdot 2 = 78$$
  

$$\begin{array}{|c|c|c|c|c|} \hline 35 & 40 & 41 & 45 & 50 \\ \hline 40 & 40 & 42 & 46 & 52 \\ \hline 42 & 46 & 50 & 55 & 55 \\ \hline 48 & 52 & 56 & 58 & 60 \\ \hline 56 & 60 & 65 & 70 & 75 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline -2 & -1 & 0 \\ \hline -1 & 1 & 1 \\ \hline 0 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & 87 & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array}$$

$$40 \cdot (-2) + 41 \cdot (-1) + 45 \cdot 0 + 40 \cdot (-1) + 42 \cdot 1 + 46 \cdot 1 + 46 \cdot 0 + 50 \cdot 1 + 55 \cdot 2 = 87$$

Para el filtrado de imágenes se usa habitualmente matrices kernel de orden 3x3 o 5x5.

La siguiente tabla muestra algunas de las matrices máscaras más usadas.

|   |   |  |  |
|---|---|--|--|
| Enfoque   | Desenfoque  | Realce de bordes   | Repujado   |
| $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ | $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$     | $\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   | $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ |
| Detección de bordes   | Filtro de tipo Sobel  | Filtro de tipo Sharpen   |  |
| $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$    | $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$  | $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$  |  |
| Filtro Norte  | Filtro Este   | Filtro de tipo Gauss   |  |
| $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ | $\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 2 & 3 & 1 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$ |  |

## Objetivo de la práctica

1. Realización de operación de filtro a elegir por el grupo de alumnos
2. La programación se realizará en dos versiones:
  - a. Algoritmo serie
  - b. Algoritmo paralelo (Numba CUDA)
3. Puesto que la programación se realizará utilizando el lenguaje Python, se recurrirá al uso de alguna librería disponible para la apertura de ficheros imágenes, así como su salvaguarda posterior con el resultado
4. Se utilizarán un conjunto de imágenes de origen con diferentes dimensiones (ejemplo: 10x10, 50x50, 100x100, 500x500, 1000x1000, etc ...) hasta un peso lo suficientemente grande como para resaltar con cambios de comportamiento entre las dos versiones del algoritmo y tomar conclusiones. La imagen puede ser la misma aplicando previamente un redimensionamiento manual a varias medidas con las que se vaya a trabajar. Para ello se podrá utilizar un editor de imágenes, por ejemplo.
5. En la versión paralela del algoritmo, también se dejará parametrizada la opción del número de GPUs a disposición del reparto y cómputo, de modo que, durante la fase de pruebas y mediciones de rendimiento, se juegue con las dos dimensiones del número de GPUs y tamaño de la imagen.

## Infraestructura de desarrollo

1. Google Colab
  2. Python
  3. NumPy
  4. Numba
  5. Librerías Python para apertura y carga de imágenes
- 

## Secciones de la memoria

Según apartados y secciones del documento disponible en Canvas "*template presentación (reunión 3) resultados finales y conclusiones (2).pptx*"

## Plazos y otras consideraciones

Entrega: 27 de noviembre a través tarea CANVAS (entrega uno por equipo)  
Presentación en clase: 29 de noviembre

## Referencias

Contexto: *Modelling in Science Education and Learning* Volume 9(2), 2016 doi: 10.4995/msel.2016.4524.