

# Implementación Blur-Effect en una imagen: Método Secuencial

Laura Cerón Martínez, Camilo Dajer Piñerez  
 Universidad Nacional Bogotá, Colombia  
 Email: lpceronm@unal.edu.co, cadajerp@unal.edu.co

**Resumen**—El *Blur-effect* más conocido como *Gaussian blur* es un efecto de difuminado de una imagen logrado gracias a la utilización de funciones Gaussianas. En este documento se presenta la implementación del algoritmo de *Gaussian-Blur* de forma secuencial, haciendo uso de la librería *OpenCV* para el procesamiento de imágenes. Se presentaran, así mismo, los tiempos de ejecución obtenidos de diversas pruebas realizadas.

## I. INTRODUCCIÓN.

### I-A. Gaussian Blur

Conocido también como *Blur-effect* o *Gaussian Smoothing*. Es un tipo de filtro de difuminado de imágenes, el cual, haciendo uso de funciones Gaussianas, busca distorsionar una imagen.

Matemáticamente hablando, aplicar *Gaussian Blur* sobre una imagen, es equivalente a decir que se está realizando una convolución de dicha imagen con un *kernel* (matriz de convolución).

En el campo de procesamiento de imágenes, la convolución es el proceso encargado de calcular la transformación a aplicar a cada pixel de una imagen. En general dicha transformación consiste en dado un pixel *A*, hallar el promedio de valores de un grupo de pixeles colindantes dentro de un radio definido por un *kernel*, dicho valor será remplazado en el pixel *A*.

La figura 1 muestra de forma matricial los pixeles de una imagen. En la matriz inferior de la figura se puede observar los valores que representan los colores de cada pixel de la sección resaltada de la matriz. Suponiendo que esta imagen se desea difuminar haciendo uso del algoritmo de *Blur-effect*, se deberá aplicar a cada uno de sus pixeles la transformación descrita anteriormente. Tomemos el caso particular del pixel que se encuentra en el centro de la matriz de la figura 1 cuyo valor es 106; para realizar la transformación correspondiente a este, se deberá hallar el promedio de los valores de los pixeles que se encuentren a su alrededor. El calculo sería el siguiente:

$$T = AVG(104 + 100 + 108 + 98 + 99 + 95 + 90 + 85)$$

$$T = 97,4$$

(1)

El resultado de esta transformación, será el nuevo valor que tome dicho pixel. Una vez se realice este procedimiento con cada pixel que conforma la imagen, se obtendrá el efecto difuminado deseado.

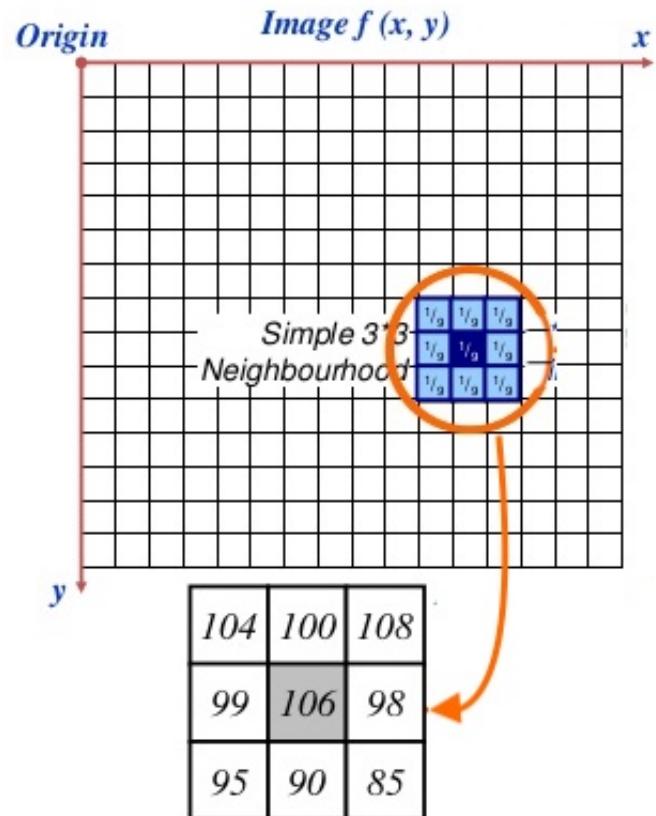


Figura 1: Representación de pixeles de una imagen.

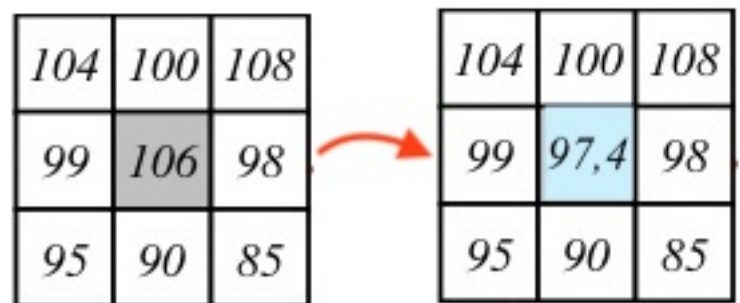


Figura 2: Transformación del pixel

## II. IMPLEMENTACIÓN ALGORITMO BLUR EFFECT

Con el objetivo de ilustrar el efecto producido por el *Gaussian Blur*, se realizó la implementación de dicho algoritmo utilizando el lenguaje de programación C++. La elección de este lenguaje se da gracias a las facilidades que este ofrece para la implementación de programas complejos y su excelente tiempo de ejecución. Así mismo, se hizo uso de la librería OpenCV con el objetivo de realizar el procesamiento y manipulación de imágenes dentro del programa.

### II-A. Implementación secuencial

El programa recibe una imagen y un tamaño de *kernel* como parámetros de entrada. El tamaño del *kernel* definirá el radio de vecindad de los píxeles que posteriormente serán usados para el cálculo del *blur-effect*.

A continuación, haciendo uso de la librería OpenCV, se extraen de la imagen los componentes del modelo de color RGB de cada uno de los píxeles que la componen. Con estos valores, el algoritmo calculará para cada píxel su nuevo valor RGB, para obtener finalmente el efecto difuminado en la imagen.

## III. EXPERIMENTOS Y RESULTADOS.

Con el fin de evidenciar el *blur-effect* y estudiar el comportamiento del algoritmo implementado, se realizó la ejecución del programa, haciendo uso de imágenes de diversas resoluciones (720px, 1080px y 4k) con *kernels* de distintos tamaños (3, 5, 7, 9, 11, 13 y 15).

Para cada uno de los escenarios se tomó el tiempo de ejecución del programa; estos tiempos fueron utilizados para realizar posteriormente el análisis de rendimiento del mismo.

La Figura 3 muestra la transformación sufrida por la imagen, luego de aplicarle el algoritmo **blur-effect**



Figura 3: Transformación con *blur-effect*

### III-A. Resultados

Al realizar la ejecución del programa, con cada uno de los posibles escenarios, se obtuvieron los tiempos de respuesta que se observan en el cuadro I.

Kernel	Tiempo de respuesta (ms)		
Tamaño	1280x720	1920x1080	3840x2160
3	750,425	1651,32	6598,11
5	1595,82	3557,43	14216,2
7	2786,14	6304,78	25279,7
9	4553,48	10321,5	40815,7
11	6603,69	14965,4	59196
13	9109,15	20319,5	80785,7
15	11904,2	26716,2	105535

Cuadro I: Tiempos de respuesta.

En la figura 4 que se muestra a continuación, se puede evidenciar el comportamiento del algoritmo en cada uno de los escenarios. En dichos escenarios se cambiaba el tipo de imagen y el tamaño del *kernel*.

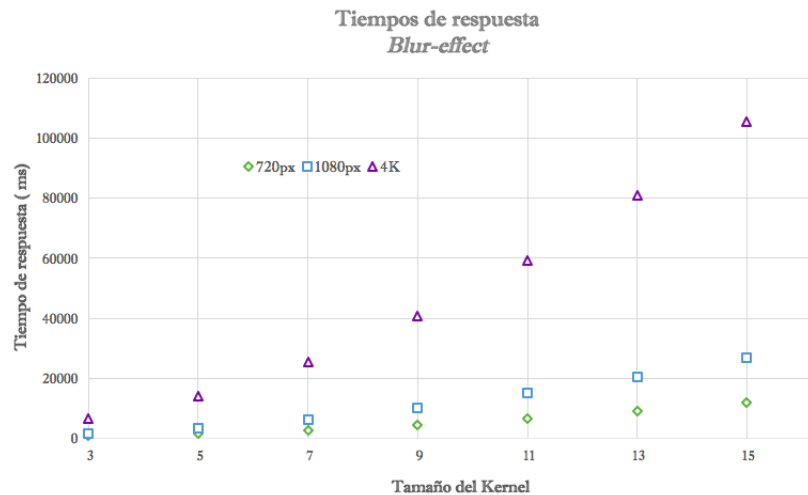


Figura 4: Tiempos de respuesta

Como se puede observar en la gráfica anterior, los tiempos de respuesta se incrementan exponencialmente, a medida que se aumenta el tamaño del *kernel* y la resolución de la imagen. Más específicamente se puede apreciar un cambio drástico en los tiempos de respuesta entre la imagen 4K y la de 1080px cuando se utiliza un *kernel* de tamaño 15.

## CONCLUSIONES.

Después de la realización de las diferentes pruebas, se pudo observar que el proceso de *blur-effect* en una imagen de muy alta resolución como lo son las imágenes 4K, pueden tardar en realizarse dependiendo del *kernel* utilizado, por lo tanto al ser un proceso iterativo no dependiente de procesos anteriores se puede paralelizar para obtener un mejor tiempo de ejecución.

## BIBLIOGRAFÍA.

- <https://www.raywenderlich.com/178486/uivisualeffectview-tutorial-getting-started>

- *Digital Image Processing* <https://www.slideshare.net/OVISOPTOamir/image-processing-42285753>
- *Gaussian blur* [https://en.wikipedia.org/wiki/Gaussian\\_blur](https://en.wikipedia.org/wiki/Gaussian_blur)
- *Kernel (image processing)* [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))
- *Gaussian blur* [https://en.wikipedia.org/wiki/Gaussian\\_blur](https://en.wikipedia.org/wiki/Gaussian_blur)
- *Desenfoque gaussiano* [https://es.wikipedia.org/wiki/Desenfoque\\_gaussiano](https://es.wikipedia.org/wiki/Desenfoque_gaussiano)