

# Práctica 3. Procesamiento de imágenes

## Procesamiento de Imágenes, Audio y Vídeo

Luna Yue Hernández Guerra, Jorge Lorenzo Lorenzo

Octubre 2025

GitHub

### Resumen

En esta práctica se profundiza en diferentes técnicas de procesamiento digital de imágenes mediante el uso de la biblioteca `OpenCV`. A lo largo de los ejercicios se aplican operaciones de inversión, mezclas no lineales, ecualización de histogramas y filtrados. Además, se experimenta con la generación de vídeos basados en secuencias de transformación y la comparación entre distintos tipos de filtros para la eliminación de ruido y la detección de bordes.

## 1. Enunciados

### 1.1. Negativo y mezcla progresiva de imágenes

Se aplicó la operación  $255 - I(i, j)$  sobre la imagen `imagen2.png` para obtener su negativo. A partir de este resultado, se generó un vídeo que combina progresivamente ambas imágenes por columnas, haciendo que la imagen original vaya apareciendo sobre su negativo. El número de iteraciones se corresponde con el ancho de la imagen y en cada una de ellas se desplaza una columna hacia la derecha la parte visible de la imagen original. El resultado final se guardó como `video2.mp4`.

### 1.2. Transformaciones logarítmica y exponencial

Se desarrolló una aplicación interactiva que permite aplicar transformaciones en el rango dinámico (logarítmica y exponencial) mediante *trackbars*, que controlan el grado de compresión o expansión de la intensidad. Se probaron ambas transformaciones sobre imágenes donde los efectos se noten, la logarítmica se prueba sobre una imagen oscura para ver como aumenta su brillo, mientras que la exponencial sobre imágenes muy brillantes para ver como se oscurecen.

### 1.3. Mezcla no lineal de imágenes

En este apartado se mezclaron las imágenes `imagen7_1.jpg` e `imagen7_2.png` utilizando la expresión:

$$I = I_1 \cos^2 \left( \frac{\alpha\pi}{180} \right) + I_2 \sin^2 \left( \frac{\alpha\pi}{180} \right)$$

donde el ángulo  $\alpha$  varía entre 0 y 360°. Se generó un vídeo de 360 fotogramas incrementando el ángulo en 1° por iteración, guardando el resultado como `video7.mp4`.

### 1.4. Ecualización del histograma

Se obtuvo el histograma ecualizado de la imagen `imagen3.png`, mejorando la distribución de intensidades y el contraste general. El resultado se guardó para su posterior filtrado.

### 1.5. Filtrado bilateral

A la imagen ecualizada del apartado anterior se le aplicó un filtro bilateral con un diámetro  $d = 10$  y valores de  $\sigma$  iguales a 11 tanto en color como en espacio. Este filtro permite reducir el ruido preservando los bordes. El resultado final se guardó como `imagen3_salida.png`.

Como el resultado no fue tan notorio, se realizó también el filtrado con valores de  $\sigma$  iguales a 50 y el resultado se guardó como `imagen3_salida_sigma50.png`.

## 1.6. Gradiente mediante Sobel y filtrado Gaussiano

Se calculó la norma del gradiente de la imagen `imagen5.jpg` utilizando los operadores de Sobel. Posteriormente, se umbralizaron los valores obtenidos: los mayores o iguales a 128 se asignaron a 255 y los menores a 0. Finalmente, se aplicó un filtro Gaussiano de radio (7,7) con  $\sigma = -1$ , obteniendo una imagen suavizada guardada como `imagen5_salida.png`.

## 1.7. Filtrado iterativo con mediana

Se aplicó un filtro mediano con un tamaño de ventana  $ksize = 7$  sobre la imagen `imagen4.jpg`, generando un vídeo de 100 fotogramas. En cada iteración, el filtro se aplica sobre el resultado anterior, produciendo un efecto progresivo de suavizado. El vídeo final se guardó con el nombre `video4.mp4`.

## 1.8. Comparativa de métodos de eliminación de ruido

Se procesó la imagen `imagen6.png` aplicando distintos algoritmos de eliminación de ruido: filtros de media, mediana, Gaussiano y bilateral. Cada método se probó individualmente para evaluar su efectividad en la reducción del ruido. Se realizó un estudio comparativo, seleccionando el método con mejor equilibrio entre preservación de bordes y suavizado.

## 1.9. Técnicas adicionales de filtrado

En esta sección, se exploraron y aplicaron tres técnicas de filtrado. En primer lugar, se investigó el método *Non-Local Means* (NLM), un algoritmo avanzado que elimina el ruido preservando las texturas y los bordes al comparar regiones similares dentro de la imagen.

Posteriormente, se diseñaron dos filtros propios. El primero, denominado *Local Histogram Attraction*, ajusta cada píxel hacia el valor más frecuente (la moda) entre sus vecinos. Así, los píxeles que representen el ruido se acercarán al valor verdadero. Esta corrección se realiza mediante un factor de fuerza ('strength') que controla cuánto se desplaza el valor original del píxel hacia la moda local. El segundo, llamado *Smart Median*, combina el filtrado mediano tradicional con un mecanismo adaptativo que sustituye únicamente los píxeles detectados como ruidosos, manteniendo intactos los detalles del resto de la imagen.

Tras aplicar las 3 técnicas, se concluyó que la que da mejores resultados es la *Smart Median*, ya que da una imagen más clara y con menos ruido que las 2 anteriores.

## 2. Dificultades

En primer lugar, encontramos ciertos obstáculos con las transformaciones logarítmicas y exponenciales al no elegir imágenes en las que el resultado se

viese potenciado.

En segundo lugar, los resultados obtenidos en durante el filtrado bilateral no fueron tan notorios, por lo que tuvimos que escoger un valor mayor para  $\sigma$ .

Por último, el principal desafío fue encontrar nuevos métodos de filtrado con los que obtener unos resultados coherentes.

### **3. Conclusión**

La práctica permite conocer diferentes herramientas de OpenCV bastante eficientes para hacer diferentes filtrado sobre las imagenes, como podría ser la eliminación de ruido, o la eliminación de demasiado brillo o el otro extremo, que serían imagenes muy oscuras.