



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

---

# Image Processing Project



---

## Aplicación de captura y procesamiento de imágenes tiempo real

---

Autores:

Félix Ozcoz Eraso ( 801108 )

Victor Manuel Marcuello Baquero ( 741278 )

# Índice

<b>Descripción.....</b>	<b>3</b>
<b>Implementación de Filtros.....</b>	<b>4</b>
<b>Filtro de contraste.....</b>	<b>4</b>
<b>Filtro de posterizado.....</b>	<b>5</b>
<b>Filtro de alien.....</b>	<b>5</b>
<b>Filtro de distorsión geométrica.....</b>	<b>7</b>
<b>Opcionales.....</b>	<b>9</b>
<b>Filtro de caleidoscopio.....</b>	<b>9</b>
<b>Vídeo demostración aplicación.....</b>	<b>10</b>
<b>Especificaciones técnicas.....</b>	<b>11</b>

## **Descripción**

El procesamiento de imágenes forma parte de la base para el desarrollo de aplicaciones en el campo de la visión por computador.

En el mismo sentido, el procesamiento de imágenes es el pilar de este proyecto; una aplicación de escritorio que permite capturar imágenes en vivo por cámara y aplicar técnicas de procesamiento de imágenes en tiempo real.

Los usuarios pueden seleccionar entre una variedad de filtros básicos como ajustes de contraste, brillo, desenfoque, posterizado, entre otros. Una vez aplicado un filtro, la imagen se muestra al instante en la pantalla, permitiendo visualizar el efecto del filtro en tiempo real. Los usuarios pueden guardar la imagen final en varios formatos.

Además pueden abrir imágenes, aplicar filtros, visualizar el efecto del filtro en tiempo real y guardar el resultado.

# Implementación de Filtros

## Filtro de contraste

El ajuste de contraste y brillo modifica la distribución de los niveles de intensidad de los píxeles generando una distribución más uniforme de las mismas.

Se le aplica a cada píxel un ajuste de contraste lineal definido por la función

$$g(x) = \alpha * f(x) + \beta$$

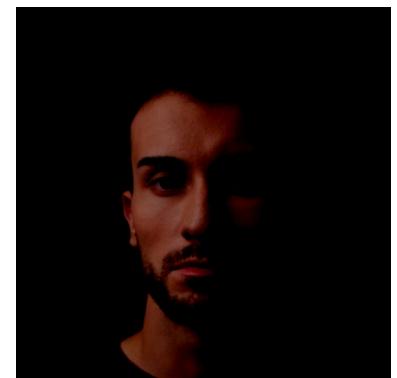
donde  $g(x)$  es el valor del píxel tras el ajuste,  $f(x)$  es el valor original del píxel,  $\alpha$  es el parámetro que controla el contraste y  $\beta$  el brillo.



Original



$\alpha = 0.9$   
 $\beta = -90.0$



$\alpha = 2.0$   
 $\beta = -10.0$

## Filtro de posterizado

El posterizado reduce el número de colores distintos en la imagen; hace que los detalles sutiles se pierdan en favor de grandes áreas de color sólido.

Para cada píxel se divide el valor por un divisor determinado por el usuario que reduce el rango de valores y trunca el resultado a un rango determinado. A continuación multiplica el resultado por el divisor lo que redondea el valor hacia abajo al múltiplo más cercano del divisor. Los valores de píxeles son múltiplos del divisor que resulta en una reducción en el número de colores.



Original



div = 64



div = 110

## Filtro de alien

El filtro de alien es un efecto visual cuya aplicación es principalmente creativa y estética que consiste en modificar la apariencia de un sujeto con el objetivo de parecerse a lo que en la cultura popular se conoce como alien.

Existen varias formas de implementar este filtro; distorsión facial, inserción de elementos alienígenas (p.e. antenas), cambio de colores, etc. Se opta por la última de ellas; cambiar el color de piel puede modificar drásticamente la apariencia del sujeto consiguiendo una imagen más surrealista al utilizar colores poco comunes en la naturaleza.

Esta implementación consiste en determinar un rango de tonalidades de piel, localizar en la imagen original aquellos píxeles cuyo valor pertenece a ese rango y sustituir sus valores por el color seleccionado por el usuario.

Más técnicamente; el primer paso es convertir la imagen a HSV. Este espacio de colores separa la información del color de la información del brillo ofreciendo robustez ante variaciones de iluminación. Además facilita la definición de rangos de color y simplifica la detección de colores mediante umbralización en un solo canal (Hue (tono)).

El siguiente paso es definir una máscara binaria para la piel que umbraliza a 255 (blanco) los píxeles que pertenecen al rango de piel definido. A continuación se crea una imagen con las mismas dimensiones que la original y se colorea con el color especificado; se aplica la máscara al color de piel para obtener la región de la imagen original que corresponde a la piel coloreada. Para colorear con mayor precisión se separa el fondo de la piel y se combina la piel coloreada con el fondo extraído.



Original



Alien  
color = (255, 0, 0)

## Filtro de distorsión geométrica

( barrel distortion and pincushion distortion )

Este filtro genera distorsiones ópticas similares a las fotografías tomadas con lentes de cámaras. La causa de este efecto son las particularidades físicas de las lentes y cómo refractan la luz al entrar en la cámara.

Barrel distortion curva las líneas rectas hacia afuera desde el centro de la imagen dando forma de barril. Esta distorsión tiende a ser notable en las partes exteriores de la imagen, lo que significa que las áreas cercanas a los bordes pueden parecer curvarse hacia afuera.

Pincushion distortion hace que las líneas rectas parecen curvarse hacia adentro desde el centro de la imagen, similar a la forma de un cojín. La distorsión tiende a ser prominente en las áreas exteriores lo que provoca que esas zonas parecen curvarse hacia adentro.

Estas distorsiones alteran la ubicación de los píxeles en una imagen para crear la apariencia de curvatura de líneas rectas sobre un centro de distorsión.

El método de reubicación de los píxeles por el que se opta se describe en términos de cómo cambian los radios al centro de la imagen.

En detalle, se basa en la relación entre la distancia euclídea de cada píxel de la imagen al centro de distorsión en la imagen original, denominada ‘radio’ y las nuevas localizaciones de cada uno de los píxeles que son los radios una vez aplicada la distorsión.

Esta explicación se resume en las siguientes ecuaciones:

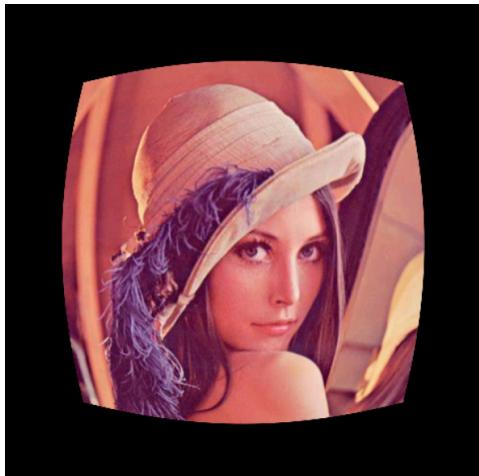
$$1) \quad \text{radio\_dist} = \text{radio} * (1 + k_1 * \text{radio}^2 + k_2 * \text{radio}^2)$$

$$2) \quad x\_dis = x * (\text{radio\_dist} / \text{radio}) + cx$$

**nota:** también se realiza 2) para las coordenadas en y

donde:

- $k_1$  y  $k_2$  son los parámetros del filtro.
- $x$  y  $x_{dis}$  son las coordenadas en el eje x original y las nuevas coordenadas, respectivamente
- $cx$  es la coordenada x del centro de distorsión utilizado para asegurar que los puntos corregidos se desplacen correctamente a lo largo del eje x.



Barrel Distortion

$$k1 = 0.00001$$

$$k2 = 0.0$$



Pincushion Distortion

$$k1 = -0.00001$$

$$k2 = 0.0$$

# Opcionales

## Filtro de caleidoscopio

Este filtro permite simular el funcionamiento de un caleidoscopio que genera una imagen compuesta de imágenes duplicadas que se distribuyen de forma simétrica.

El caleidoscopio se construye a partir de un tubo, una lente y tres espejos inclinados que conforman un prisma triangular cuya parte reflectante es interna. A través de la mirilla del instrumento se pueden ver imágenes que contienen objetos de distintas formas y colores. Cuando se volteá el tubo, dichas imágenes se duplican de forma simétrica.

Existen varias versiones modificadas a través de algunos parámetros como el nivel de aumento de la lente, el número de espejos alojados en el interior del caleidoscopio (así como la distancia angular entre cada espejo, que es el ángulo de rotación), las coordenadas que conforman el centro exacto de enfoque, etc...

Se opta por implementar una versión del caleidoscopio simple, cuyo interior contiene cuatro espejos que se ubican a 90 grados entre sí, por lo que el ángulo de rotación es múltiplo de 90.

En cuanto a la lente del caleidoscopio hay un abanico de modificaciones que permiten distorsionar, aumentar o cambiar el punto central de enfoque. La versión elegida es una lente simple invertible, esto es, se puede alternar entre reflejar la parte interna y externa del espejo, siendo la parte interna la que refleja por defecto.

Cabe destacar que la implementación está adaptada para imágenes cuadradas, si no lo fuesen se redimensionan para aplicar el filtro.

Para implementar este filtro primero se crea una máscara bi-tonal diagonal (esta máscara utiliza los colores blanco y negro), que simula cada uno de los espejos del caleidoscopio, de tal forma que la parte reflectante sea la zona coloreada de negro.

A continuación, se utiliza la máscara para crear una imagen reflejada (para ello, se necesita una imagen traspuesta de la imagen original, cuya operación se realiza al inicio) y se rota dicha imagen en un ángulo determinado (0,90,180,270,360). En último lugar, se duplican las imágenes reflejadas, se concatenan para crear la imagen caleidoscópica y, finalmente, se amplifica esta última imagen en un 50%.



**invert = “no”**  
**rotation = 90°**

**invert = “yes”**  
**rotation = 90°**

## Vídeo demostración aplicación

Este link referencia el vídeo demostración paso a paso comenzando por descargar la aplicación hasta probar todas las funcionalidades que ofrece.

[https://drive.google.com/file/d/1Rc-78bUFRKqRet-bP2yZBiORNTZRy-f4/view?usp=driv\\_e\\_link](https://drive.google.com/file/d/1Rc-78bUFRKqRet-bP2yZBiORNTZRy-f4/view?usp=driv_e_link)

**NOTA:** el acceso al vídeo está restringido a correos de la Universidad de Zaragoza. En caso de no poder acceder al vídeo, por favor envíame un correo a 801108@unizar.es, gracias

# Especificaciones técnicas

## Arquitectura:

Aplicación de escritorio íntegramente desarrollada en python

## Lenguaje de programación:

Python v3.10

## Paquetes:

- OpenCV v4.9.0.8: manipulación de imágenes
- Tkinter : biblioteca estándar de Python para interfaces gráficas de usuario
- Pillow v.10.2: Python Image Library (PIL) fork, librería que ofrece operativa adicional para procesamiento de imágenes

## Hardware:

La carga de trabajo es mínima; la aplicación puede ejecutarse en sistemas capaces de ejecutar aplicaciones ligeras y por tanto no se especifican requisitos de hardware concretos.