

Control del brillo/contraste/gamma de la imagen.

Objetivos de la práctica:

- Aprender a mejorar una imagen aplicando procesamiento puntual de la imagen.
- Aprender a convertir entre tipos de datos para el pixel.
- Aprender a separar/componer canales RGB.
- Aprender a utilizar una GUI con "slider" para ajustar parámetros.

Descripción básica (hasta 6p):

Desarrollar un programa que cargue una imagen (puede ser monocroma o en color) para ajustar el brillo, contraste y gamma de la misma según la siguiente ecuación:

$$I' = cI_g + b$$

donde I es la imagen de entrada e I' es la imagen de salida, ambas normalizadas en el intervalo $[0.0, 1.0]$ para ser procesadas. En la versión básica se deberá codificar una función que haga este proceso accediendo pixel por pixel a la imagen.

Además, el programa tendrá un flag '-i' en la CLI para indicar modo interactivo. En este modo el programa visualizará la imagen en una ventana junto con tres deslizadores para controlar los tres parámetros. El resultado se mostrará de forma interactiva según se modifiquen los deslizadores. Si el usuario pulsa ESC, termina sin guardar el resultado. Si pulsa cualquier otra tecla se guarda el resultado.

La CLI del operador tendrá la forma siguiente:

```
processImg [-i] [-c <float=1.0>] [-b <float=0.0>] [-g <float=1.0>] <input  
img> <output img>
```

el contraste tendrá valores en el intervalo $[0, 2.0]$, el brillo en el intervalo $[-1.0, 1.0]$ y gamma en el intervalo $[0.0, 2.0]$.

Opcional:

- (+1p) Utilizar operaciones vectoriales de OpenCV. (mantener también el código de la versión básica que se supone que es accediendo pixel por pixel a la imagen).
- (+2p) Cuando la imagen es en color RGB, el proceso se aplicaría solo a la "iluminancia" y no al color "croma". Para ello se debe convertir la imagen de RGB al espacio de color HSV, aplicar el proceso en el canal "V" y deshacer el cambio de nuevo al espacio de color RGB. Para activar este modo de proceso añadir un flag '--luma' a la CLI.
- (+1p) Añadir un tercer argumento opcional <img-mask> a la CLI que representa una imagen de mascara 0/255 de forma que el procesamiento sólo afecte a los pixeles emascarados.

Recursos:

- Método `Mat::convertTo` para convertir entre tipos de pixel.
- Función `cv::cvtColor` para conversiones entre espacios de color.
- Funciones `cv::split` and `cv::merge` para desentrelazar/entrelazar canales.
- Operaciones algebraicas con el tipo `cv::Mat`.
- Módulo OpenCV HighGui para la gestión de la interfaz gráfica de usuario.