VISIÓN ARTIFICIAL
Práctica 2: Procesamiento de imágenes

Construir una aplicación de procesamiento de imágenes que incluya opciones para aplicar las diferentes técnicas de procesamiento revisadas en el tema 2 de la asignatura.

Especificación de objetivos

Las operaciones de procesamiento que deberá proporcionar la aplicación son: transformación de píxel (nivel de gris), umbralización, ecualización del histograma, suavizado gaussiano, filtro mediana, filtro lineal, dilatación y erosión. Además de estas operaciones, se incluirá una opción adicional que permitirá realizar hasta 4 operaciones de procesamiento en secuencia sobre la misma imagen. En la parte obligatoria de la práctica, las operaciones se llevarán a cabo exclusivamente sobre la imagen de grises.

A continuación se detallan las especificaciones concretas de cada operación:

- Transformación de píxel: aplicación de una función de transformación de niveles de gris definida por tramos. El usuario especificará la transformación de 4 niveles de gris, 2 de ellos fijos (0 y 255) y dos variables. El programa deberá construir la tabla LUT que representa la función de transformación, definiendo un segmento recto entre cada par de puntos.
- Umbralización: binarización de la imagen a partir de un valor umbral fijado por el usuario.
- Ecualización del histograma: ampliación de contraste de la imagen mediante una ecualización de su histograma.
- Suavizado gaussiano: aplicación de un filtro cuadrado de suavizado gaussiano del tamaño indicado por el usuario. La anchura del filtro (w) determinará el valor de la desviación típica de la función gaussiana (σ) a partir de la relación $\sigma = w/5$.
- Filtro mediana: suavizado de la imagen mediante un filtro mediana considerando entornos de píxel de tamaño 3x3.
- Filtro lineal: aplicación de un filtro lineal genérico utilizando un *kernel* de tamaño 3x3 especificado por el usuario. Tras la aplicación del filtro cada píxel podrá ser modificado sumándole un valor adicional indicado también por el usuario.
- Dilatación: aplicación de una operación morfológica de dilatación tras binarizar la imagen considerando el mismo valor umbral que en la opción "*Umbralización*". El elemento estructural utilizado para la operación será un cuadrado de tamaño 3x3.
- Erosión: aplicación de una operación morfológica de erosión tras binarizar la imagen considerando el mismo valor umbral que en la opción "*Umbralización*". El elemento estructural utilizado para la operación será un cuadrado de tamaño 3x3.

Junto con la implementación de estas operaciones, se plantea también como objetivo llevar a cabo una serie de operaciones de procesamiento sobre ciertas imágenes para obtener determinados resultados. Para ello se proporciona un fichero (*imagenes.zip*) que incluye 8 imágenes. Cada fichero *imgX_ini.png* contiene la imagen de partida. Sobre esta imagen se debe determinar la operación o secuencia de operaciones de procesamiento que permiten generar la imagen *imgX_obj.png*. Para cubrir este objetivo, deberá entregarse un documento (formato PDF) donde se describa, por cada imagen inicial, las distintas operaciones que deben realizarse para obtener la imagen objetivo.

Interfaces de usuario

La figura 1 muestra el aspecto de la interfaz de usuario principal de la aplicación. En la parte superior se muestran la imagen original (izquierda) y la resultante de aplicar una operación de procesamiento (derecha). Debajo de cada imagen, se muestra su histograma. En la parte inferior se proporcionan diferente elementos para la ejecución de cada operación de procesamiento.

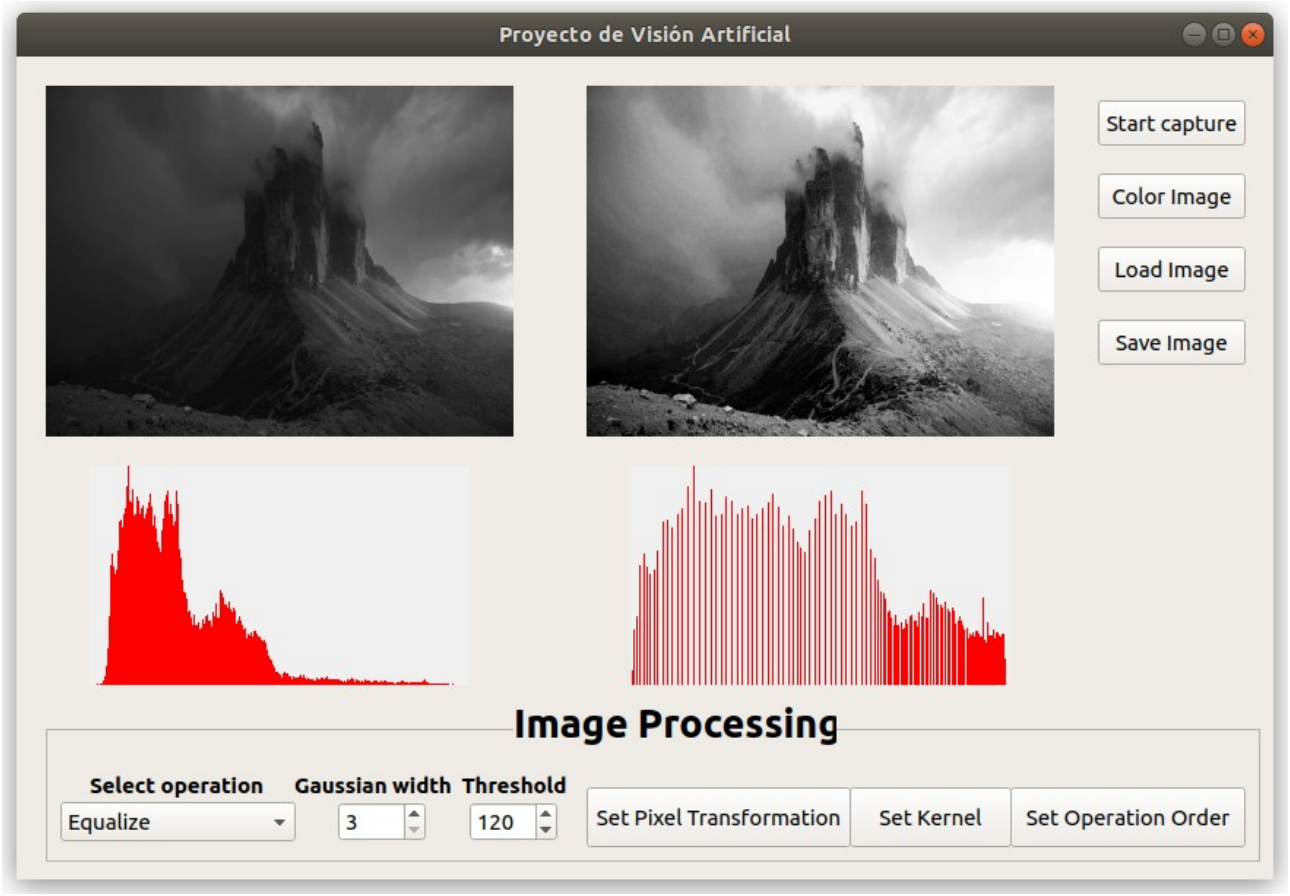

Figura 1: Interfaz de la aplicación.

El *cuadro combinado* de la izquierda permite seleccionar una operación concreta de procesamiento. Al pulsar sobre este elemento se despliega una lista con las posibles operaciones proporcionadas por la aplicación (ver figura 2).

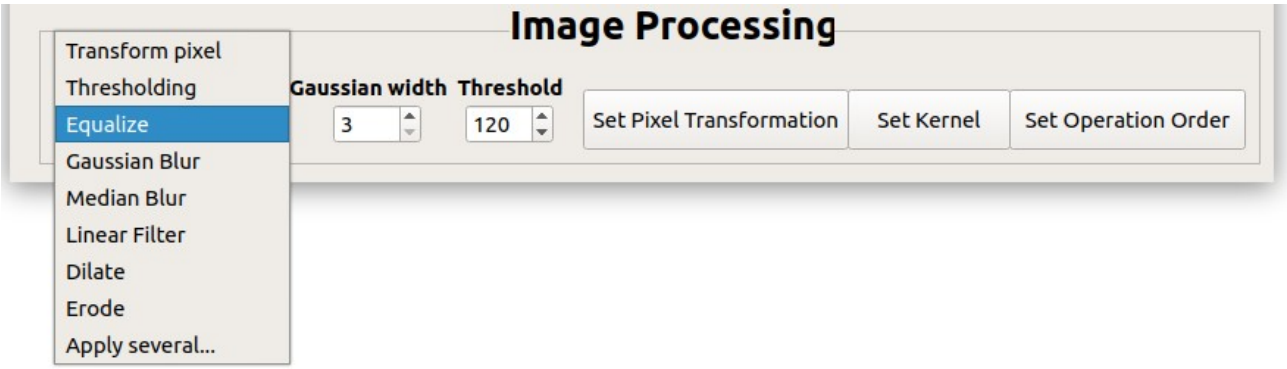

Figura 2: Listado de opciones

El selector *Gaussian width* permite al usuario indicar la anchura del kernel en la opción *Gaussian Blur* (suavizado gaussiano). El siguiente selector (*Threshold*) contiene el valor umbral utilizado en la binarización de la imagen (opción *Thresholding*). Por último, los botones "*Set Pixel Transformation*", "*Set Kernel*" y "*Set Operation Order*" despliegan diferentes diálogos para fijar, respectivamente, la transformación de píxel asociada con la opción *Transform Pixel*, el contenido del kernel de la opción *Linear Filter* y el grupo de operaciones a secuenciar en la opción *Apply several*. Las figuras de la 3 a la 5 muestran el aspecto de estos 3 diálogos.

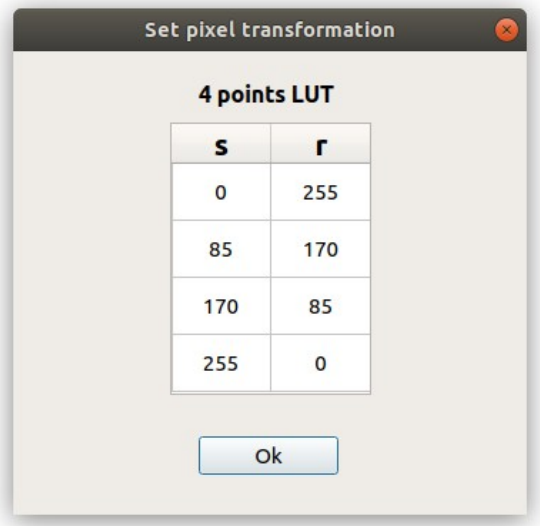

Figura 3: Diálogo asociado con el botón *Set Pixel Transformation*

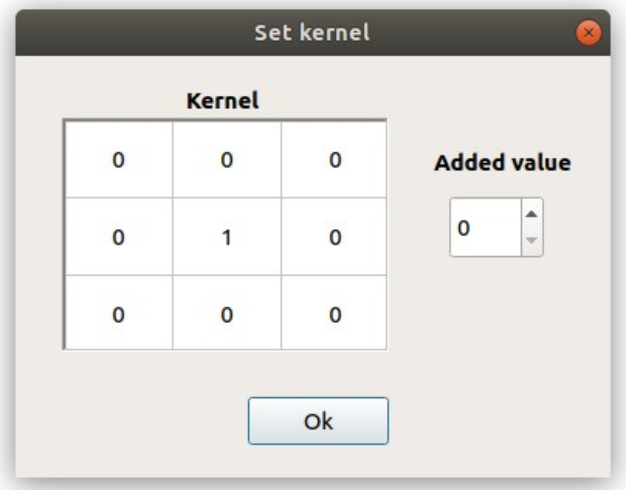

Figura 4: Diálogo asociado con el botón *Set Kernel*

Figura 5: Diálogo asociado con el botón *Set Operation Order*

Utilidades

Proyecto software de partida

Se proporciona una nueva versión del proyecto básico en la que se ha incluido una nueva interfaz principal para la aplicación con los elementos que se muestran en la figura 1. Asimismo, como parte de los archivos del proyecto, se han incluido los formularios de Qt que contienen la descripción de los diálogos de las figuras 3, 4 y 5. En la clase "MainWindow", se han añadido 3 atributos de clase asociados con estos diálogos: *pixelTDialog*, *lFilterDialog* y *operOrderDialog*. A través de estos 3 objetos se realizará el control de los 3 diálogos.

Además de estos nuevos elementos, la clase cuenta con un nuevo método (*updateHistogram*) que se encarga de calcular y mostrar en pantalla el histograma de una imagen. Para ello, recibe por parámetro la imagen (objeto de tipo Mat), que debe ser de tipo CV_8UC1, y un puntero al visor (objeto de tipo *imgViewer*) donde se dibujará la representación del histograma. El método es invocado en el método *compute* para mostrar el histograma de las imágenes origen y destino en cada captura de imagen.

OpenCV

- *cv::LUT()*: transformación a nivel de píxel.
- *cv::threshold()*: umbralización.
- *cv::equalizeHist()*: ecualización del histograma.
- *cv::GaussianBlur()*: suavizado gaussiano.
- *cv::medianBlur()*: suavizado mediana.
- *cv::filter2D()*: aplicación de filtro lineal.
- *cv::dilate()*: aplicación de filtro morfológico de dilatación.
- *cv::erode()*: aplicación de filtro morfológico de erosión.

Ampliaciones

Una de las ampliaciones propuestas es poder aplicar las diferentes opciones de procesamiento a imágenes en color (ver figura 6). Para que el procesamiento afecte solo a la información de intensidad y no a la de color, será necesario transformar la representación del espacio RGB a un espacio que incluya información separada de intensidad antes del procesamiento y volver al espacio

RGB una vez procesada la imagen. Concretamente, se recomienda usar el espacio de color YUV, que separa la información de intensidad (canal Y) de la de color (canales U y V). Para la transformación del espacio de color puede usarse la función de OpenCV *cvtColor*. La obtención del canal Y para su procesamiento puede llevarse a cabo con la función *split* y la composición de la imagen tras procesar el canal Y con la función *merge*. Cuando las imágenes visibles sean las de color, los histogramas que deben visualizarse estarán asociados con el canal Y de la imagen origen y destino. Para visualizar ambos histogramas, deben realizarse las llamadas correspondientes al método *updateHistogram* indicando como primer parámetro el objeto *Mat* que contienen el canal de intensidad de cada imagen.

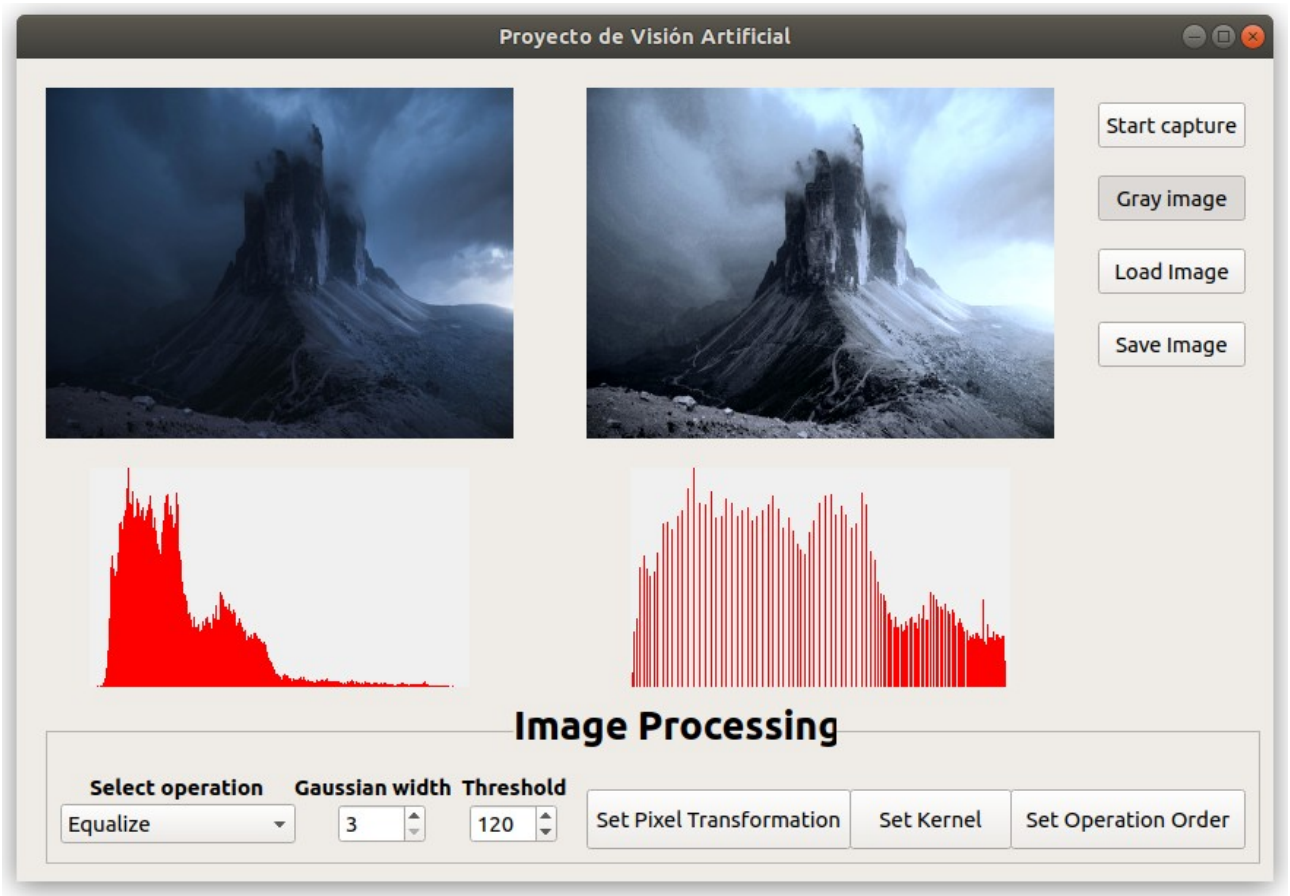

Figura 6: Operación de ecualización aplicada a una imagen en color

Además de esta ampliación, se proponen las siguientes :

- Incluir transformaciones de píxel predefinidas (p.e. función negativo).
- Permitir al usuario seleccionar distintos tamaños del kernel (p.e. 3, 5 ó 7) en el filtrado lineal de imágenes.
- Permitir al usuario seleccionar diferentes elementos estructurales en la aplicación de las operaciones morfológicas de dilatación y erosión.