

VISIÓN ARTIFICIAL

Práctica 1: Introducción al entorno de desarrollo

El principal objetivo de esta práctica es familiarizarse con el entorno de desarrollo que se utilizará a lo largo de la asignatura. Para ello se proponen varias modificaciones del proyecto software básico que se proporciona como parte del material de la asignatura.

Especificación de objetivos

Incluir en el proyecto básico el control de las nuevas opciones que se muestran en la figura 1. La implementación de la última opción (“Warp + Zoom”) será opcional.

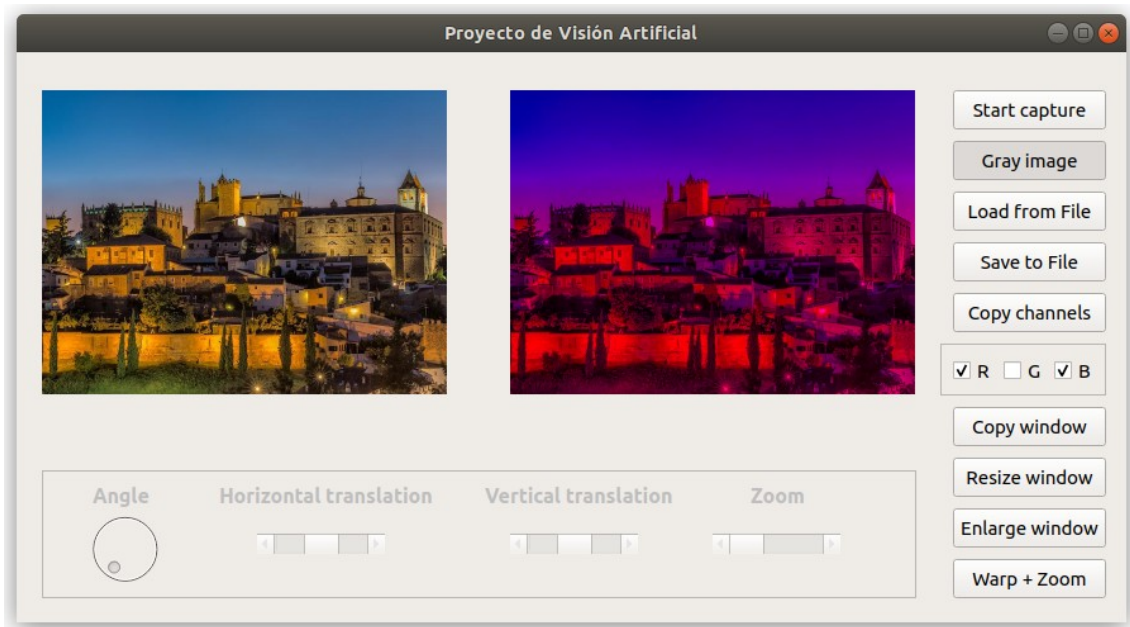


Figura 1: Nuevas opciones del proyecto básico

Cada una de estas opciones realiza la función que se describe a continuación:

- *Load from File* (Cargar imagen desde fichero): permite al usuario trabajar con imágenes de archivo en lugar de utilizar las imágenes capturadas por la cámara. Cuando el usuario seleccione esta opción, deberá mostrarse un diálogo de selección de archivo (*QFileDialog*) para permitir la elección del fichero que contiene la imagen a cargar (ver figura 2). Si el fichero contiene una imagen válida, deberá detenerse la captura de cámara y cargar la imagen del fichero en color (*colorImage*) y en escala de grises (*grayImage*). Si el tamaño de la imagen no coincide con el tamaño utilizado por la aplicación (320x240), deberá redimensionarse a dicho tamaño. Para volver a trabajar con imágenes de cámara, el usuario deberá activar de nuevo la captura pulsando sobre el botón correspondiente.

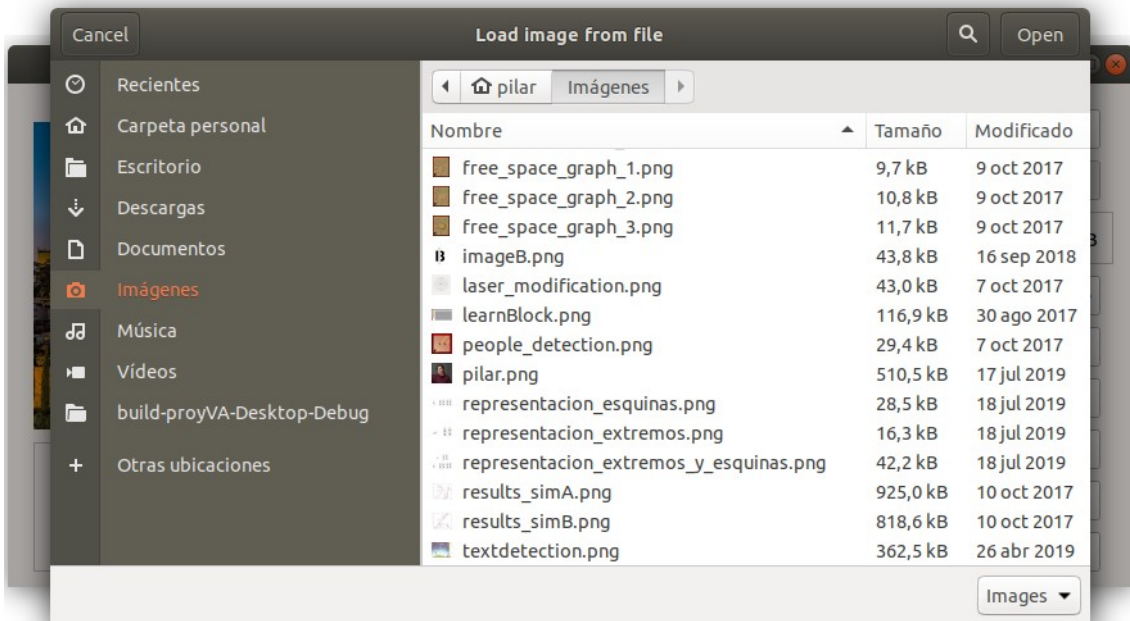


Figura 2: Opción *Load from File*

- *Save to File* (Guardar imagen en fichero): guarda la imagen resultante del procesamiento actualmente visible (*destColorImage* o *destGrayImage*) en el fichero seleccionado por el usuario. La imagen en color deberá guardarse en formato BGR. Como en la opción anterior, el usuario seleccionará el nombre del archivo a través de un diálogo de selección de archivos.
- *Copy channels* (Copiar canales seleccionados): copia a la imagen destino en color (*destColorImage*), los canales activos (casillas de activación “R”, “G” y “B”) de la imagen fuente en color (*colorImage*) (ver figura 1). Para los canales que no se encuentren activos, se considerará un valor de 0.
- *Copy window* (Copiar ventana): si hay alguna ventana seleccionada, esta opción copia el contenido de dicha ventana en las imágenes en color y en escala de grises (*colorImage*, *grayImage*) a las imágenes resultantes del procesamiento (*destColorImage*, *destGrayImage*) centrando sobre ellas la ventana (ver figura 3).

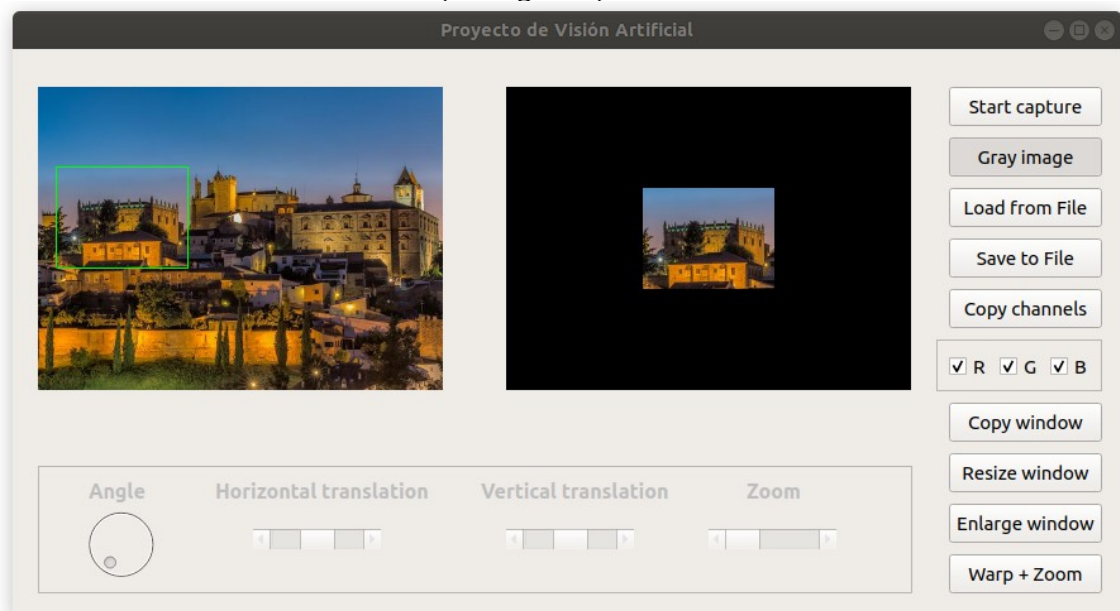


Figura 3: Opción *Copy window*

- *Resize window* (Redimensionar ventana): copia el contenido de la ventana seleccionada en las imágenes en color y en escala de grises a las imágenes resultantes del procesamiento, redimensionando dicha ventana al tamaño máximo de imagen (320x240) (ver figura 4).

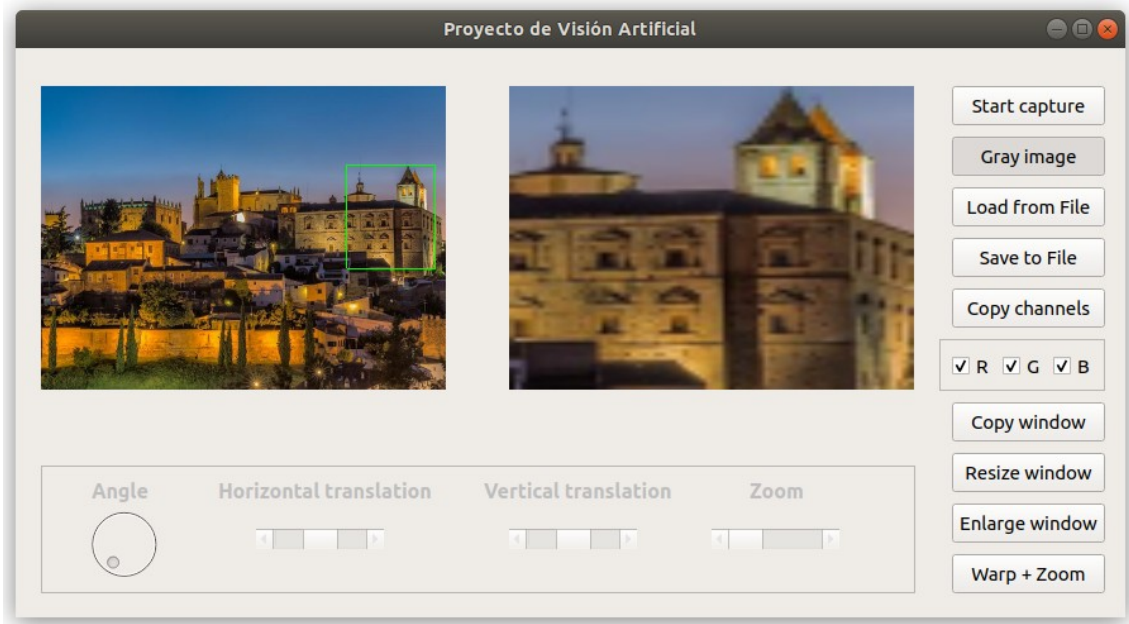


Figura 4: Opción *Resize window*

- *Enlarge window* (Ampliar ventana): copia el contenido de la ventana seleccionada en las imágenes en color y en escala de grises a las imágenes resultantes del procesamiento, ampliando dicha ventana hasta completar el total de filas o columnas de manera que se mantenga la proporción alto/ancho de la imagen (ver figura 5).

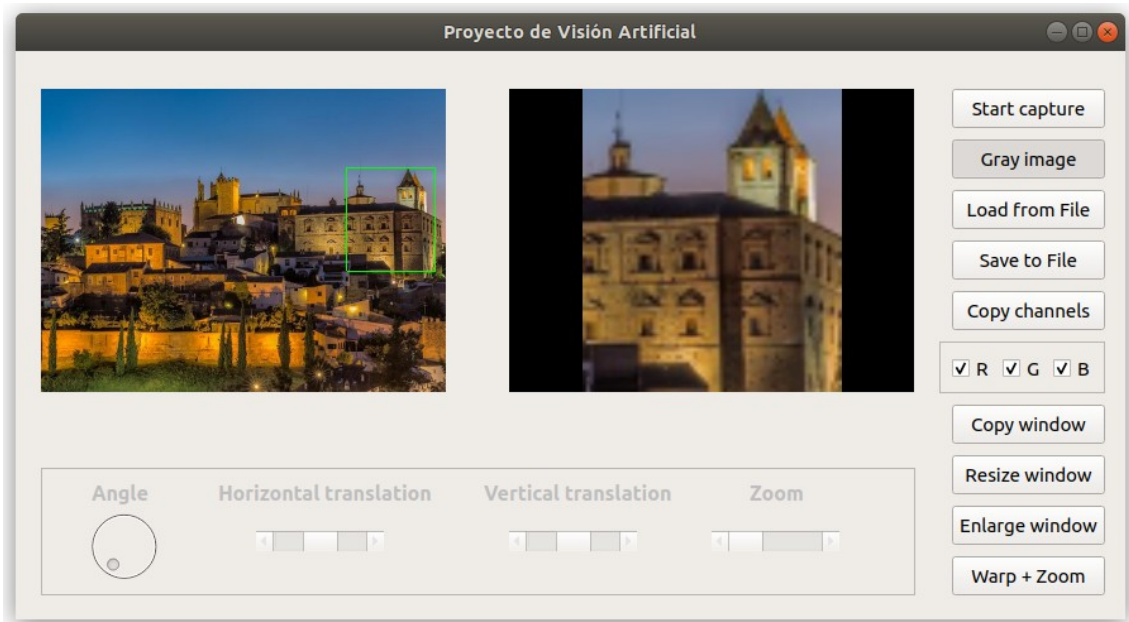


Figura 5: Opción *Enlarge*

Además de estas opciones, se incluirá la posibilidad de visualizar los atributos de un píxel de la imagen origen al hacer clic sobre una posición concreta del visor (ver figura 6). Así, si la imagen visible en el visor es la de grises se mostrará el nivel del gris del píxel indicado y, si es la de color,

el valor numérico de los atributos de rojo, verde y azul. Una posibilidad a la hora de mostrar esta información es utilizar la clase `QToolTip` de Qt para mostrar descripciones emergentes. No obstante, se deja libertad para utilizar otros elementos de Qt (p.e., `QLCDNumber`).

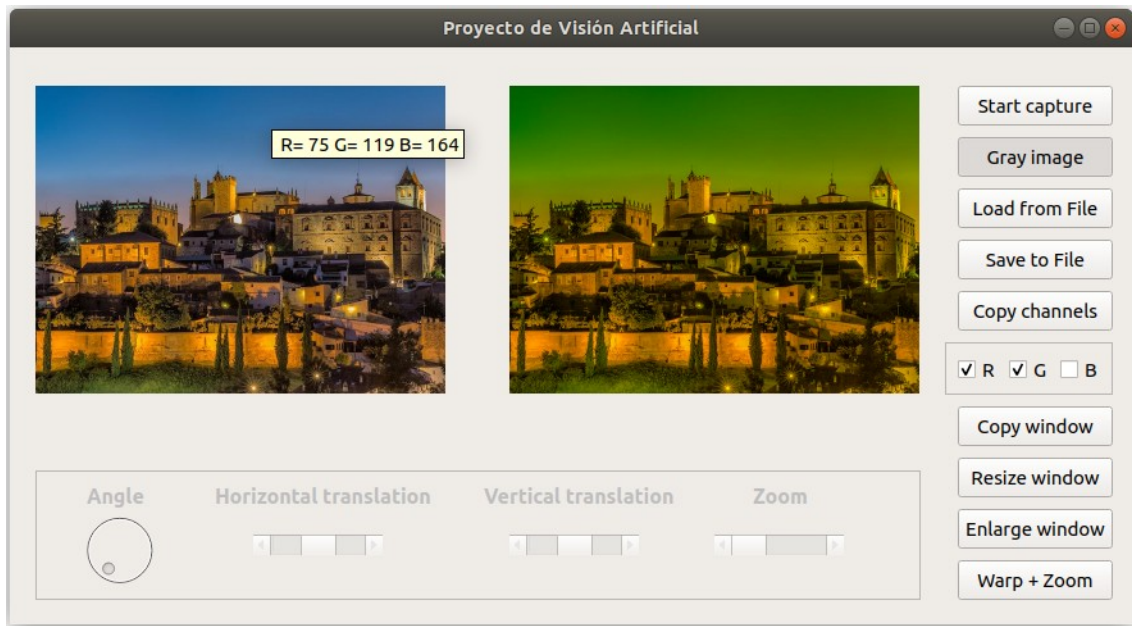


Figura 6: Visualización de los atributos de un píxel

Utilidades

- **Qt:** `QFileDialog`, `QToolTip`
- **OpenCV:**
 - `imread`: carga una imagen desde fichero.
 - `imwrite`: guarda una imagen en fichero.
 - `split`: separa los canales de una imagen en varias imágenes, una por canal.
 - `merge`: forma una imagen única a partir de varias imágenes que contienen canales individuales.
 - `Mat (Mat, Rect)`: constructor de `Mat` para crear una imagen a partir de una ventana de otra imagen. La zona de datos de la nueva imagen es parte de la zona de datos de la imagen original.
 - Para generar una ventana de imagen, alternativamente al constructor anterior, se pueden usar los métodos:
 - `rowRange` (método de `Mat`): crea una imagen (cabecera) con el rango de filas especificado.
 - `colRange` (método de `Mat`): crea una imagen (cabecera) con el rango de columnas especificado.
 - `copyTo` (método de `Mat`): copia el contenido de una imagen en otra.
 - `resize`: redimensiona una imagen.
 - `setTo` (método de `Mat`): pone todos los píxeles de la imagen al valor indicado.
 - `at` (método de `Mat`): acceso a un píxel de la imagen.

Ampliación

Se propone, como ampliación de la práctica, incluir opciones que permitan rotar, trasladar y ampliar la imagen de entrada. El resultado de esta transformación deberá almacenarse en las imágenes resultantes del procesamiento (`destGrayImage` y `destColorImage`) y visualizarse a través del visor correspondiente.

La figura 7 muestra una posible transformación de la imagen de entrada a partir de la aplicación de esta opción. La ejecución de dicha opción se llevará a cabo a través del botón etiquetado como “Warp + Zoom”. Mientras este botón se encuentre activo, el programa deberá aplicar la rotación, traslación y valor de zoom indicados en los elementos de la interfaz “Angle”, “Horizontal translation”, “Vertical translation” y “Zoom”.

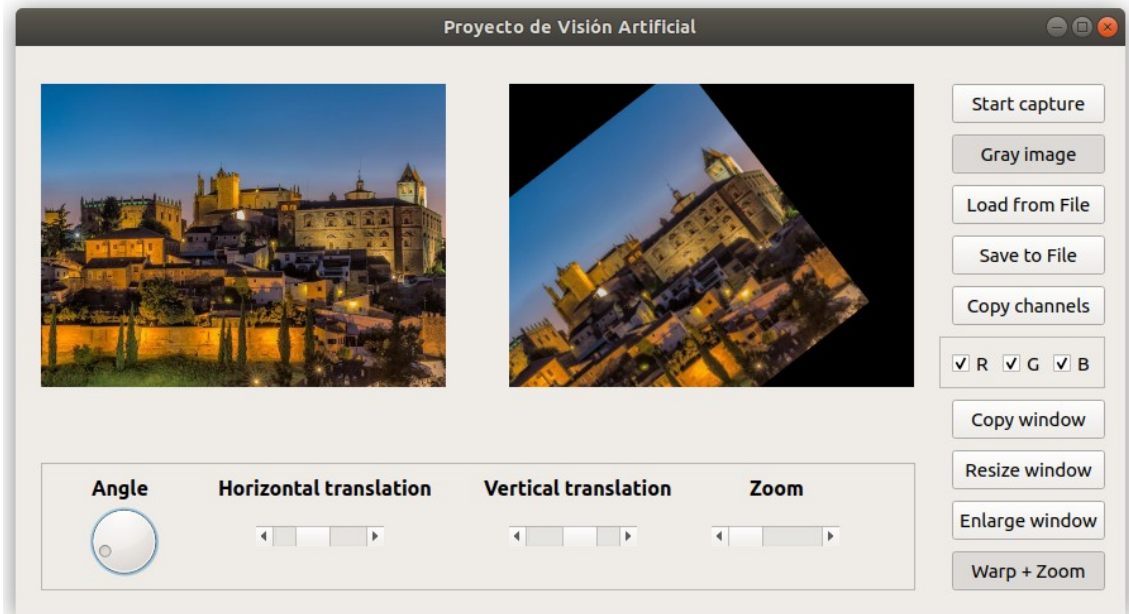


Figura 7: Nueva opción de rotación, traslación y ampliación de la imagen. Resultado tras aplicar una rotación y una traslación.

Para llevar a cabo la rotación y traslación de la imagen se aplicará una transformación afín (función *warpAffine* de OpenCV) a partir de los valores indicados por el usuario. La rotación se aplicará suponiendo que el centro de giro es el centro de imagen. Para generar la parte de la matriz de transformación asociada con la rotación, se recomienda usar la función *getRotationMatrix2D* de OpenCV.

La ampliación (*zoom*) de la imagen se resolverá redimensionando la imagen (una vez transformada) con el factor de escala indicado por el usuario y copiando la ventana central de tamaño 320x240 a la imagen destino. La figura 8 muestra el efecto de aplicar esta operación sobre la imagen transformada de la figura 7.

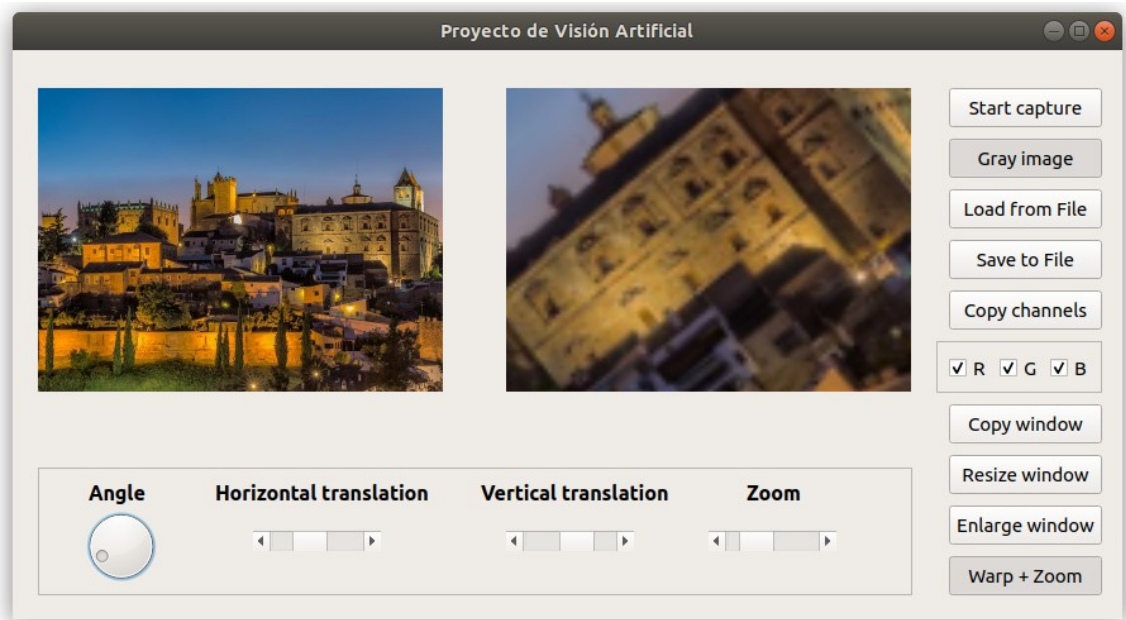


Figura 8: Resultado tras aplicar un zoom sobre la transformación de la imagen de la figura 7.