
PRÁCTICA 1
CARGA DESDE .BMP Y .JPG DE UNA IMAGEN
OPERACIONES PUNTUALES

Robótica y Visión por Computador
Ing. Telecomunicaciones

Universidad Miguel Hernández

Realización de la práctica:

La presente practica se realiza en modo tutorial para comprender el funcionamiento básico de las librerías OpenCV. En primer lugar se realiza la lectura y carga de dos imágenes (un .bmp y un .jpg), para posteriormente realizar una operación puntual con ellas. Concretamente se realiza la diferencia entre ambas y un escalado de la imagen resultante. Con este ejemplo se observa la diferencia entre ambos formatos debido a la pérdida de información en la compresión de imágenes.

1. En primer lugar se debe abrir el workspace creado en la práctica 0 o bien generar uno nuevo tal y como se explicó.
2. Añadimos un nuevo proyecto al workspace, que llamaremos **practical1**
 - a. Si estamos trabajando sobre el workspace anterior, los proyectos `cv.dsp`, `cvaux.dsp` y `highgui.dsp` ya están añadidos, y lo único que debemos hacer es establecer las dependencias y los settings del nuevo proyecto creado.
 - b. Si hemos creado un proyecto nuevo, se deben añadir los proyectos `cv.dsp`, `cvaux.dsp` y `highgui.dsp` tal y como se explicó en la práctica 0 y luego establecer la configuración adecuada.
3. Desde el explorador de Windows, creamos un directorio `\images` dentro de la carpeta `\practical1` y colocamos las imágenes de la práctica, `example.bmp` y `example.jpg`.
4. Escribir el código siguiente:

```
// el fichero example.bmp debe estar en el directorio images
char name0[] = "images/example.bmp";
// el fichero example.jpg debe estar en el directorio images
char name1[] = "images/example.jpg";

int main()
{

    IplImage* img0 = NULL; // inicializo img0
    IplImage* img1 = NULL; // inicializo img1

    img0 = cvLoadImage( name0, -1 );// cargamos example.bmp en img0
    img1 = cvLoadImage( name1, -1 );// cargamos example.jpg en img1

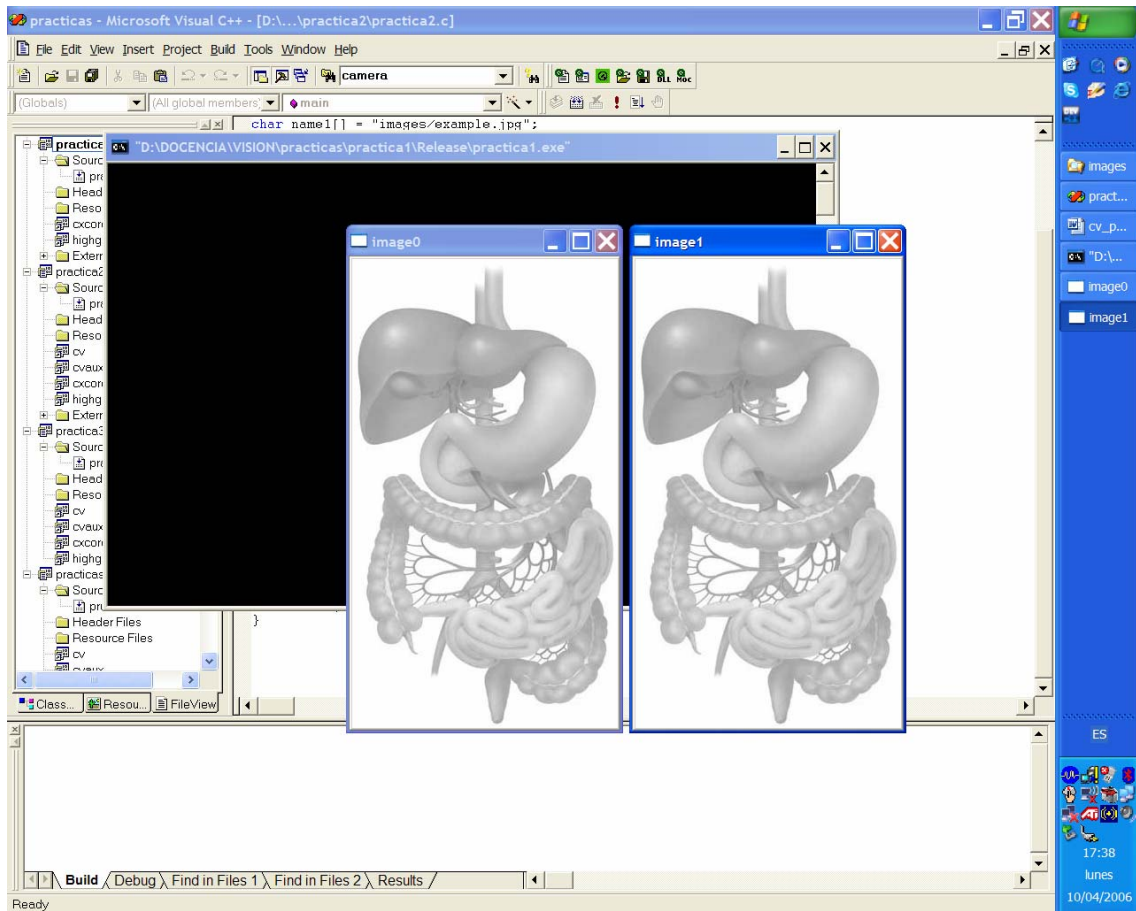
    cvNamedWindow( "image0", 1 );
    cvNamedWindow( "image1", 1 );

    // se muestran las imagenes en dichas ventanas
    cvShowImage( "image0", img0 );
    cvShowImage( "image1", img1 );

    cvWaitKey(0); // espera la pulsacion de la tecla
    cvReleaseImage( &img0 );// se libera la memoria
    cvReleaseImage( &img1 );// se libera la memoria

    return(0);
}
```

El resultado debe ser el observado en la figura



5. Modificamos el código para realizar la resta de ambas imágenes. Para ello se debe crear una imagen resultante de la misma dimensión que las imágenes originales. Se utilizará la función `cvSub` para restar las imágenes y `cvConvertScale` para cambiar la escala y poder visualizar mejor el resultado

```
IplImage* res = NULL; // inicializo una imagen resultado
CvSize imgSize; // tamaño de la imagen

img0 = cvLoadImage( name0, -1 );// cargamos example.bmp en img0
img1 = cvLoadImage( name1, -1 );// cargamos example.jpg en img1

// asignamos el tamaño de img0 a la estructura imgSize
imgSize.width = img0->width;
imgSize.height = img0->height;

// creacion de la imagen resultante
res = cvCreateImage( imgSize, 8, 3 );

// se resta img0 de img1 y el resultado se almacena en res
cvSub( img0, img1, res, 0 );
// se amplifica el resultado para poder visualizarlo
cvConvertScale( res, res, 70, 0 );
```

```
cvNamedWindow( "imagen", 1 );  
cvShowImage( "imagen", res );// se muestra res
```

El resultado debe ser el observado en la figura



6. Se recomienda utilizar otras funciones de las OpenCV para operaciones puntuales, como puedan ser la inversa o una umbralización.