Práctica 0
Instalación openCV bajo Windows
Ejemplo bajo Microsoft Visual C++ 6.0

Robótica y Visión por Computador
Ing. Telecomunicaciones

Universidad Miguel Hernández

Instalación básica:

Los siguientes pasos deben seguirse para instalar las librerias OpenCV en un sistema Windows.

1. Bajar el ejecutable `OpenCV_b5a.exe` y guardarlo en disco.
2. Ejecutar la instalación. Dejar las opciones por defecto. Dentro de `'Archivos de programa\OpenCV'` se creará una estructura de directorios
3. Ir a `Inicio→ OpenCV→ OpenCV Workspace MSVC 6`, esto abrirá el Visual Studio 6.0 y el proyecto de las OpenCV. Aunque en la instalación se han copiado los ejecutables, es necesario recompilar para generar las `.dll` y `.lib` adecuadas para nuestro procesador. Para ello, vamos a `Build→Batch Build→Build`. (Se recomienda compilar sólo las opciones `Debug` y `Release` de cada proyecto). Cerrar el proyecto.
4. Añadir el directorio `OpenCV\bin` al path del sistema. Para ello, en WinXP iremos a `panel de control→ sistema →(pestaña Opciones Avanzadas) Variables de entorno →` añadiremos `OpenCV\bin` al final del **path** actual. Se recomienda reiniciar Windows para asegurarnos que el cambio tenga efecto.
5. Probar los ejemplos que vienen con las librerías.
6. Bajarse el proyecto ejemplo `practica0.zip` y descomprimirlo en una carpeta nueva.
7. Abrir el proyecto pinchando en `practicas.dsw`. Comprobar que el proyecto activo es `practica0` en la configuración `Win32-Release`.

   NOTA: *Este proyecto está configurado de forma que la instalación básica de OpenCV se ha realizado en* `C:\Archivos de programa\OpenCV`. *Si la instalación se hubiera realizado en otro directorio, se deberá revisar la configuración del proyecto, tal y como se explica en el apartado siguiente, Configuración del proyecto para un proyecto nuevo.*
8. Revisar el código de `practica0.c`
9. Compilar, linkar y ejecutar.

Configuración del proyecto para un proyecto nuevo.

Los siguientes pasos deben seguirse para configurar bajo Microsoft Visual C++ 6.0 un nuevo proyecto de consola que trabaje con las librerías OpenCV. Se recomienda crear un nuevo proyecto e incluir el fichero `practica0.c` en él, de forma que nos familiaricemos con las dependencias.

1. Arrancar Developer Studio y crear una nueva aplicación:
   1. Desde el menu "**File**"->"**New...**"-> en la pestaña "**Projects**". Elegimos "**Win32 console application**"
   2. Nombramos el proyecto y lo guardamos en un directorio
   3. Si es el primer proyecto dentro del workspace, creamos en primer lugar el workspace ("**Create new workspace**"). También se puede

incluir un Nuevo proyecto al workspace actual. Esta opción la podemos utilizar para tener todas las prácticas dentro del mismo workspace. ("Add to current workspace").

4. pulsamos el boton "next"
5. Elegimos **"An empty project"**, click **"Finish"**, **"OK"**.

2. Añadimos los ficheros al proyecto:
   - **"File"->"New..."->**pestaña **"Files"**.
   - elegimos **"C++ Source File"**, y damos un nombre al fichero **"OK"**
   - añadimos los #includes necesarios para trabajar con OpenCV:

```
#include "cv.h"
#include "cvaux.h" // experimental stuff (if need)
#include "highgui.h"
```

3. Configuramos los *settings* del proyecto:
   - Abrimos el cuadro de diálogo para los settings del proyecto **"Project"->"Settings...".**
   - Seleccionamos el proyecto a configurar en el panel derecho.
   - Para todas las configuraciones:
     - Seleccionamos **"Settings For:"->"All Configurations"**
     - Elegimos **"C/C++" tab -> "Preprocessor"** category -> **"Additional Include Directories:"**. Y añadimos los siguientes paths, (separados por comas). (Los paths pueden introducirse como direcciones absolutas o relativas al fichero .dsp)

```
opencv\cxcore\include, opencv\cv\include,
opencv\otherlibs\highgui,opencv\cvaux\include.
```

     - Elegimos **"Link" tab -> "Input"** category -> **"Additional library path:"**. Se añaden el siguiente path para las librerías (cxcore.lib, cv.lib, cvaux.lib, highgui.lib):

```
C:\Archivos de programa\OpenCV\lib
```

   - Para la configuración **"Debug"**
     - Seleccionamos **"Settings For:"->"Win32 Debug"**.
     - Elegimos **"Link" tab -> "General"** category -> **"Object/library modules".** Y añadimos los siguientes paths, (separados por comas). (Los paths pueden introducirse como direcciones absolutas o relativas al fichero .dsp)

```
cvd.lib, highguid.lib, cvauxd.lib
```

   - Para la configuración **"Release"**
     - Seleccionamos **"Settings For:"->"Win32 Release"**.
     - Elegimos **"Link" tab -> "General"** category -> **"Object/library modules"**. Y añadimos los siguientes paths,

(separados por comas). (Los paths pueden introducirse como direcciones absolutas o relativas al fichero .dsp)

```
cv.lib, highgui.lib, cvaux.lib
```

4. Añadimos las dependencias del proyecto dentro del propio workspace:
   - Seleccionamos: **"Project" -> "Insert project into workspace"**.
   - Seleccionamos `opencv\cv\make\cv.dsp`.
   - Hacemos lo mismo para los otros proyectos: `opencv\cvaux\make\cvaux.dsp`, `opencv\otherlibs\highgui\highgui.dsp`.
   - Establecemos las dependencias:
     - Seleccionamos: **"Project" -> "Dependencies..."**
       1. Para `"cv"` elegimos `"cxcore"`,
       2. Para `"cvaux"` elegimos `"cv", "cxcore"`,
       3. Para `"highgui"` elegimos `"cxcore"`,
       4. Para nuestro proyecto elegimos todas: `"cxcore", "cv", "cvaux", "highgui"`.
5. Compilar, linkar y ejecutar.

Practica0.c

```
//
// practica0.c - creacion y visualizacion de una imagen usando OpenCV
//
// Robotica y Vision por Computador

#include "cv.h" // incluye las definiciones de OpenCV
#include "cvaux.h" // Stuff.
#include "highgui.h" // incluye highGUI. Necesario para crear ventanas de visualizacion
#include <stdio.h>

int main()
{
	IplImage *cvImg; // estructura IplImage para crear y manejar imagenes
	CvSize imgSize; // tamanyo de la imagen

	int i = 0, j = 0;
	imgSize.width = 640; // el tamaño de la imagen es
	imgSize.height = 480; // 640x480 pixels

	// se crea una imagen de grises con 8 bits de profundidad
	cvImg = cvCreateImage( imgSize, 8, 1 );

	// para recorrer la imagen, se utiliza un bucle anidado
	// y el acceso al pixel se realiza al igual que para
	// cualquier array unidimensional
	for ( i = 0; i < imgSize.width; i++ )
		for ( j = 0; j < imgSize.height; j++ )
		((uchar*)(cvImg->imageData + cvImg->widthStep*j))[i] =(char) ( ( i * j ) % 256 );

	cvNamedWindow( "RVC. probando OpenCV...", 1 ); // se crea la ventana de visualizacion
	cvShowImage( "RVC. probando OpenCV...", cvImg ); // se visualiza la imagen en dicha ventana

	cvWaitKey( 0 ); // espera la pulsacion de la tecla
	cvDestroyWindow( "image" ); // se cierra la ventana de visualizacion
	cvReleaseImage( &cvImg ); // se libera la memoria

	return( 0 );
}
```