Nota: Algunas de las imágenes que aparecen en esta presentación provienen del libro:
Visión por Computador: fundamentos y métodos.
Arturo de la Escalera Hueso. Prentice Hall.

Intelligent Systems Lab

Sistemas de Percepción

Visión por Computador

Arturo de la Escalera
José María Armingol
Fernando García
David Martín
Abdulla Al-Kaff

uc3m | Universidad Carlos III de Madrid

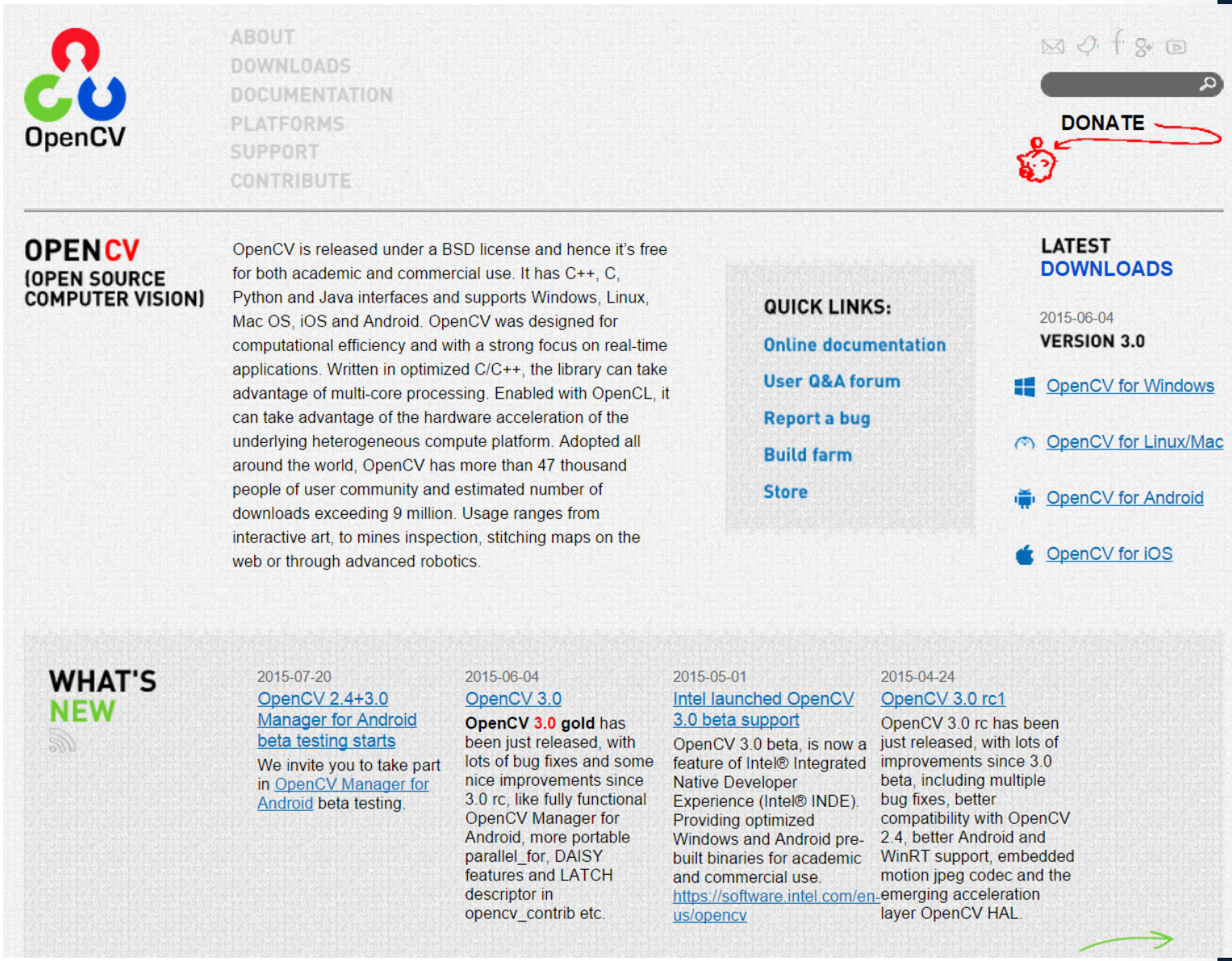
Intelligent Systems Lab

Las Librerías OpenCV

Las librerías OpenCV

- ¿Qué son las librerías OpenCV?
  - Open Source Computer Vision
  - Librerías de visión por computador
  - desarrolladas por Intel
  - 1999 versión alfa, 2015 versión 3.0
  - Licencia BSD . Pueden ser usadas
  - para propósitos comerciales y
  - de investigación.
  - Multiplataforma: Linux, MacOS X y Windows
  - 500 funciones (c, c++, python)
- sourceforge.net/projects/opencvlibrary
- Learning OpenCV. Computer Vision with the OpenCV Library. G. Bradski, A. Kaehler, O'Reilly Media, 2008

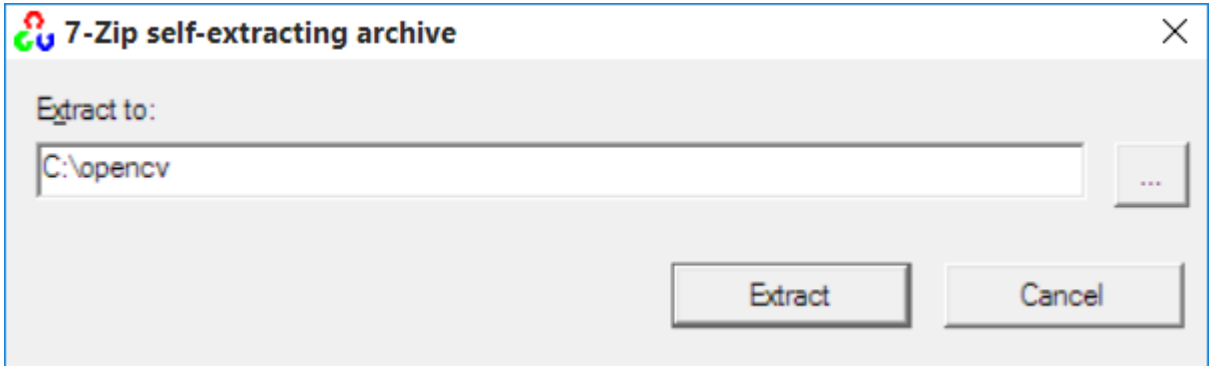
- http://opencv.org



Las librerías OpenCV

- Instalación de las OpenCV
  - Descarga desde: sourceforge.net/projects/opencvlibrary

- Instalar en
  - C:\opencv



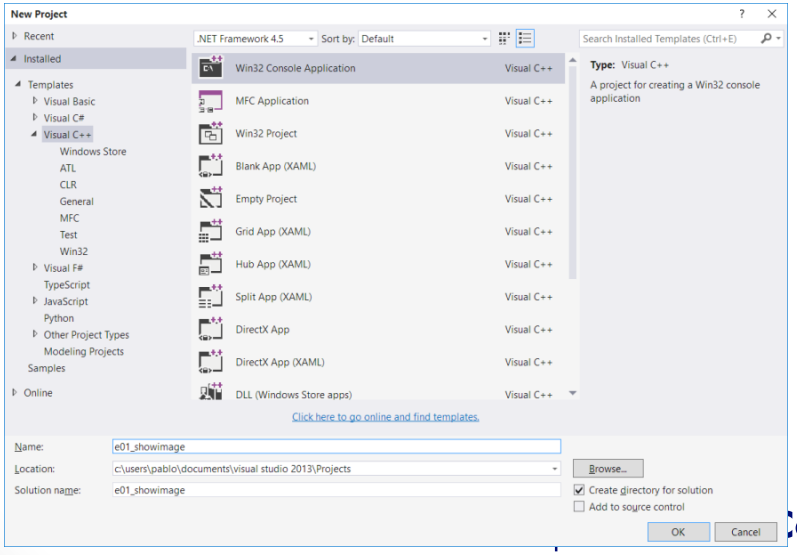
Las librerías OpenCV

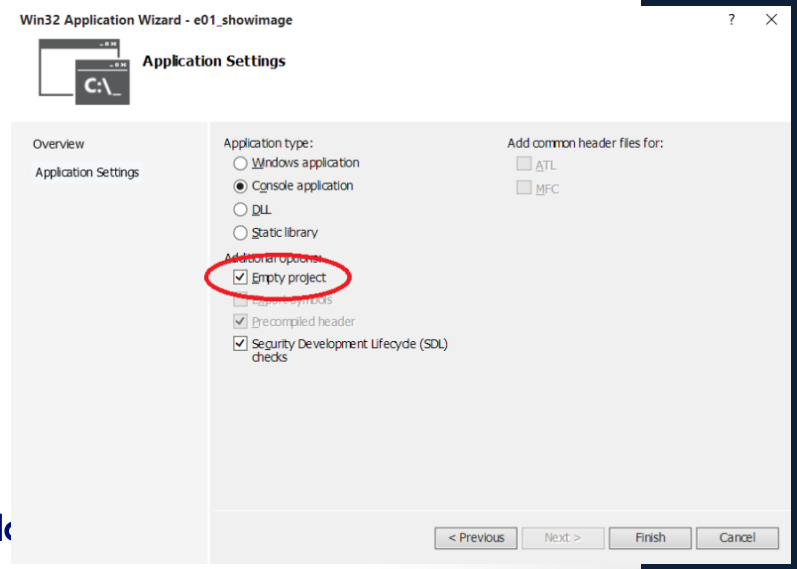
- En Microsoft Visual Studio (todas versiones)
  - Proyecto nuevo
  - Añadir en VisualC++ subdirectorios y la ruta de las librerías.
  - Seleccionar las bibliotecas del proyecto.

¿Cómo?

Las librerías OpenCV

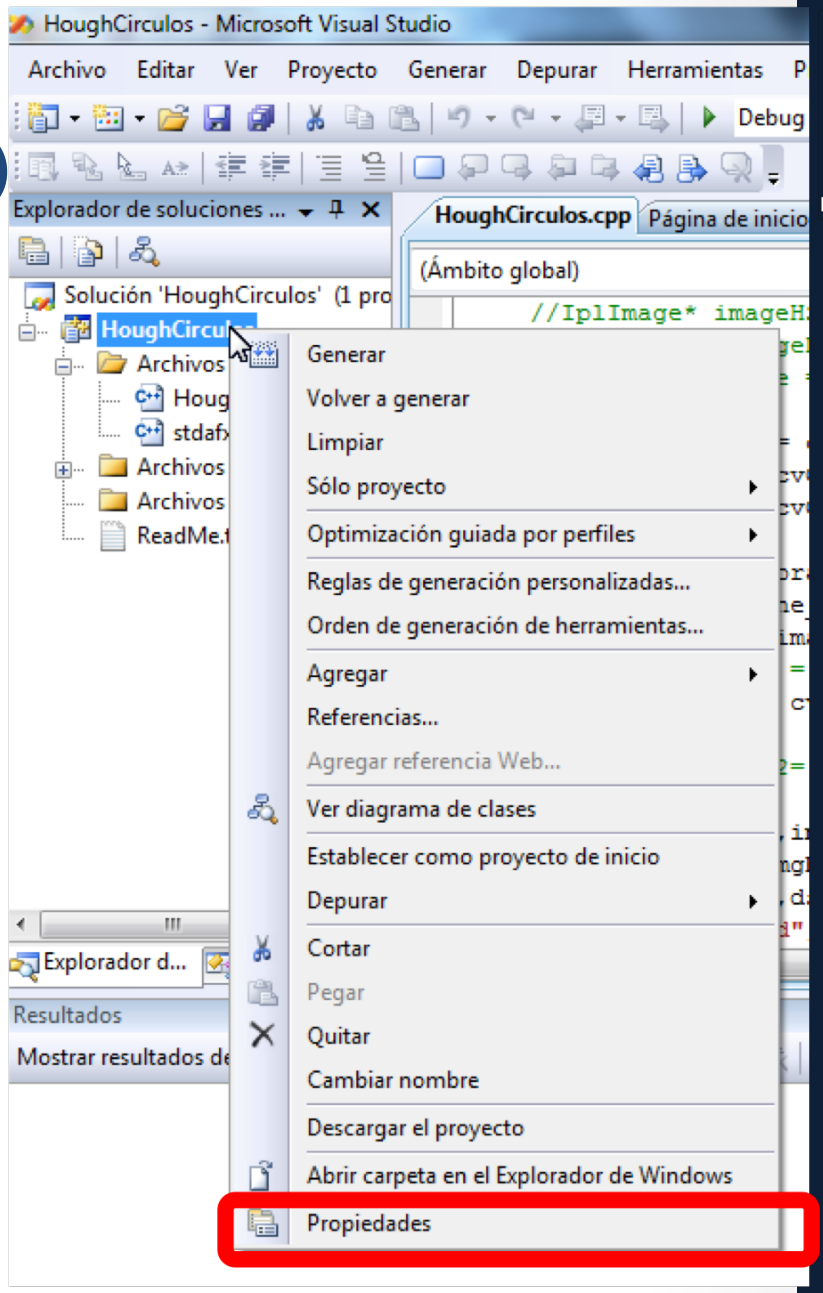
- Iniciar MS Visual C++ Archivo/Nuevo/Proyecto
- Seleccionar Tipo de proyecto:
  - Visual C++ -> Aplicación de consola Win32
- Nombre -> “L01_showimage”





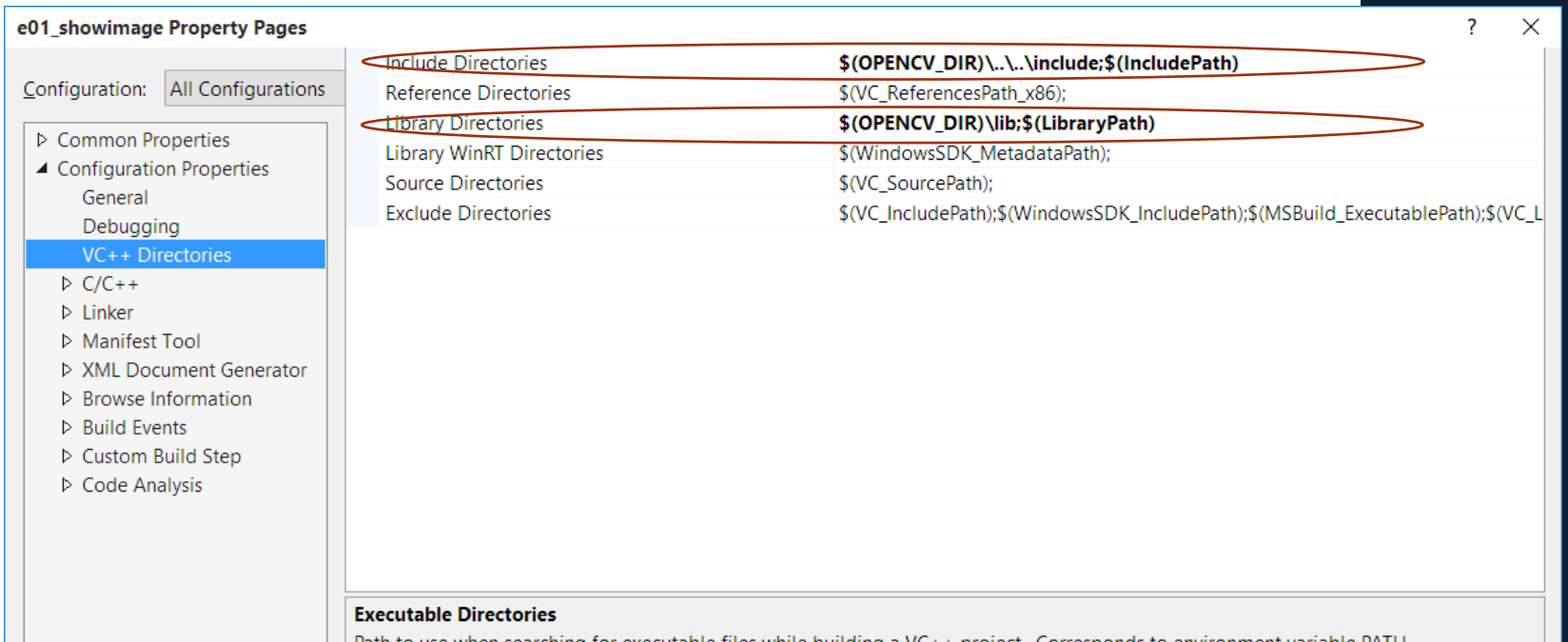
Las librerías O

- Incluir los subdirectorios y la ruta de las librerías
- Seleccionar el nombre del proyecto, pulsar botón derecho y pinchar en propiedades



Las librerías OpenCV

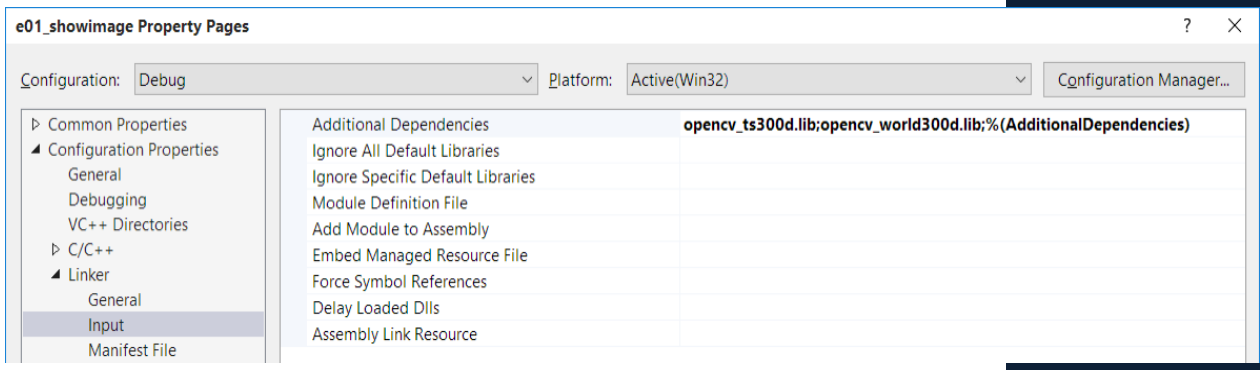
- Configuración: 'All configurations'
- Incluir la ruta:
  - $(OPENCV_DIR)\..\..\include
- Ruta de la librería:
  - $(OPENCV_DIR)\lib



Las librerías OpenCV

- Seleccionar las librerías de las opencv en el proyecto:

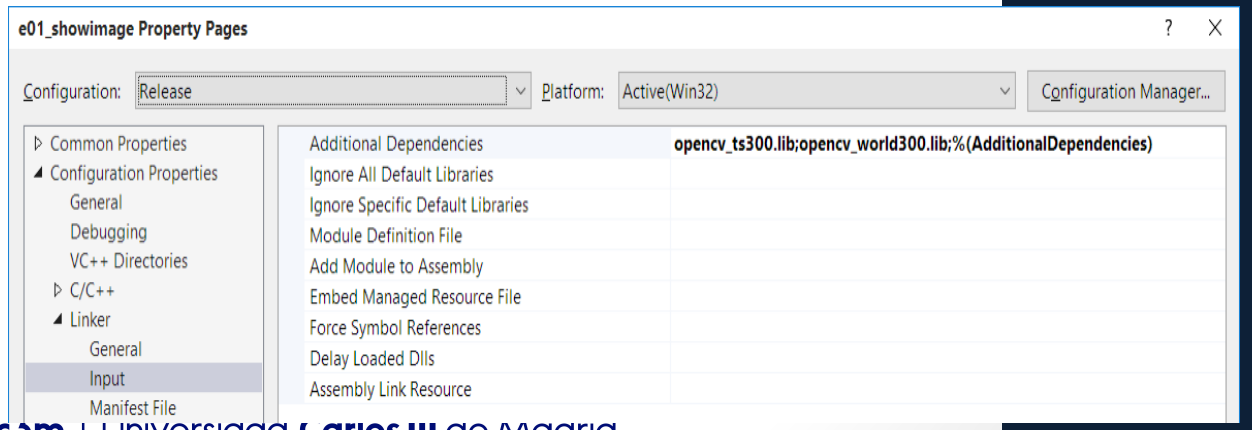
•En Vinculador/entrada (Linker/Input) -> Dependencias adicionales (Aditional dependencies)



•En modo 'debug':
- opencv_ts300d.lib
- opencv_world300d.lib

•En modo 'release':
- opencv_ts300.lib
- opencv_world300.lib

- NOTA:
  - 300 indica que es la versión 3.0
  - Dependiendo de la que sea cambiará

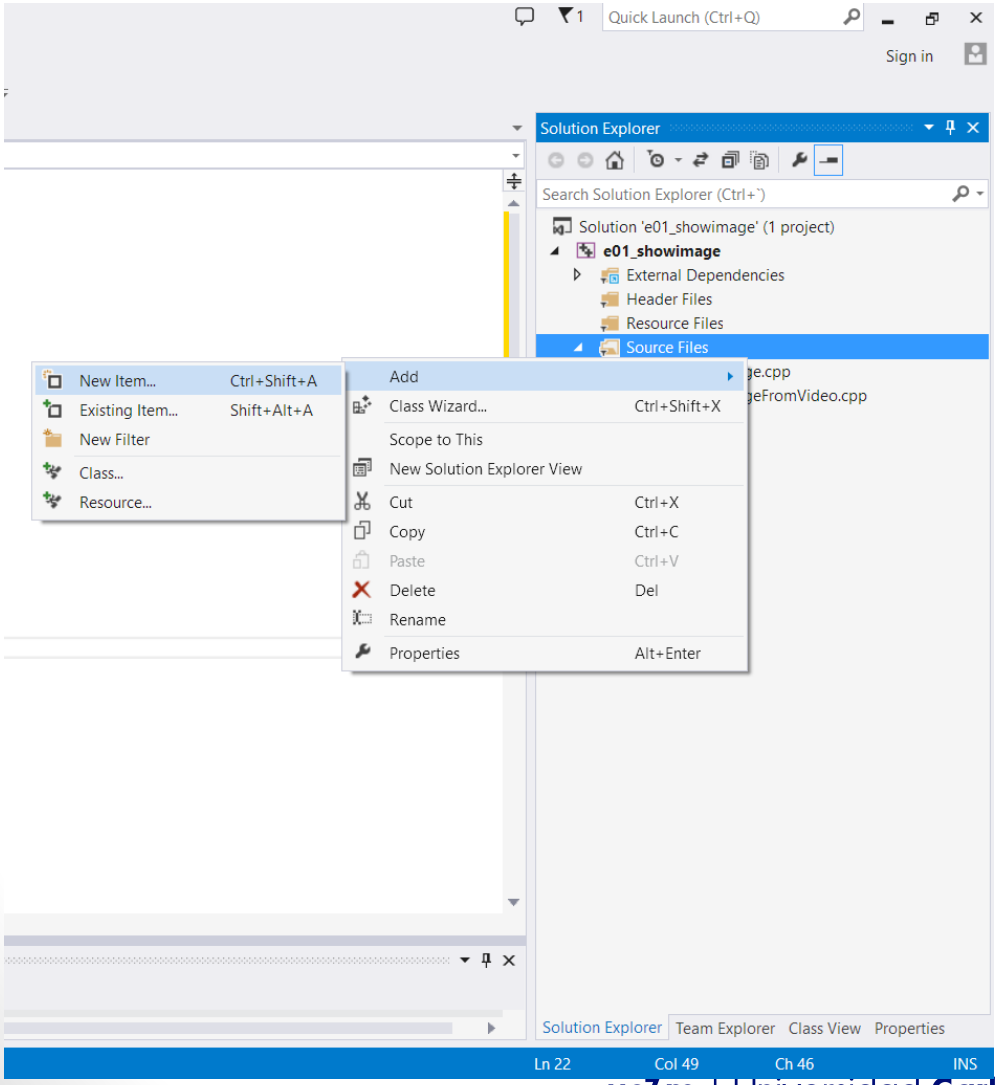


Las librerías OpenCV

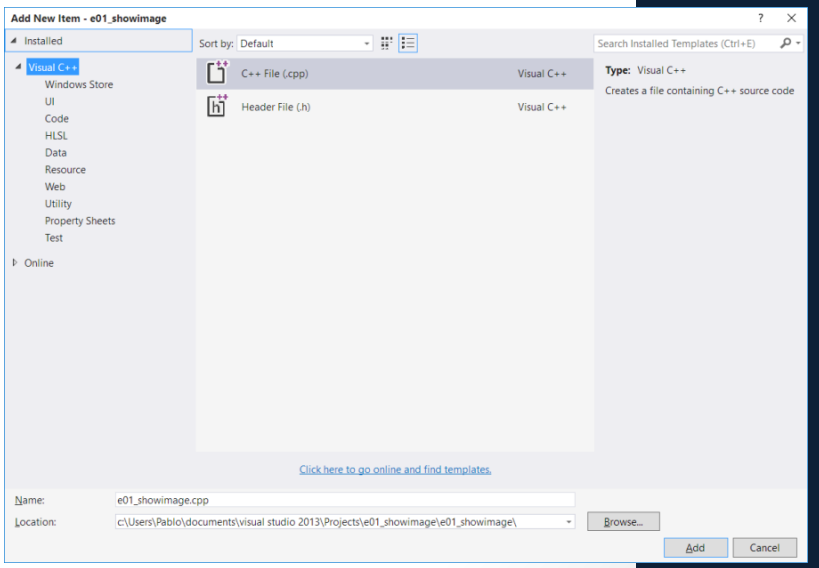
- Complete libraries:

opencv_ts300.lib
opencv_world300.lib
IlmImf.lib
ippicvmt.lib
libjasper.lib
libjpeg.lib
libpng.lib
libtiff.lib
libwebp.lib
opencv_calib3d300.lib
opencv_core300.lib
opencv_features2d300.lib
opencv_flann300.lib
opencv_hal300.lib
opencv_highgui300.lib
opencv_imgcodecs300.lib
opencv_imgproc300.lib
opencv_ml300.lib
opencv_objdetect300.lib
opencv_shape300.lib
opencv_photo300.lib
opencv_stitching300.lib
opencv_superres300.lib
opencv_videostab300.lib
opencv_video300.lib
zlib.lib

Las librerías OpenCV



- Se añaden los códigos fuente:



Las librerías OpenCV

- **Ejemplo 01:**

- Mostrar una imagen de disco:
- #include <opencv/cv.h>
- 1. Nombre de la imagen en el disco
- 2. Cargar la imagen y comprobar
- 3. Mostrar la imagen
- 4. Esperar a la pulsación de cualquier tecla
- 5. Liberar la memoria

e01_showimage.cpp*

(Global Scope)

```
1  // e01_showimage.cpp: Load image from disk and show in window
2  //
3  #include "opencv\cv.hpp"
4  #include <iostream>
5
6  using namespace cv;
7  using namespace std;
8
9  int [illegible] char* argv[])
10 {
11     // Object instantiation
12     Mat img;
13
14     // Load image from disk
15     img = imread("mandril.jpg");
16     if (!img.data){
17         cout << "error loading image" << endl;
18         return 1;
19     }
20
21     // Create window canvas to show image
22     namedWindow("original", CV_WINDOW_AUTOSIZE);
23
24     // Show image in the name of the window
25     imshow("original", img);
26
27     // Function for show the image in ms.
28     // 0 means wait until keyboard is pressed
29     waitKey(0);
30
31     // Free memory
32     destroyWindow("original");
33     // End of the program
34     return 0;
35 }
36
```

100 %

Las librerías OpenCV

- Ejemplo 02: Mostrar una imagen perteneciente a un vídeo.

1. Cargar el archivo del vídeo
2. Comprobar que se ha cargado correctamente
3. Extraer la primera imagen
4. Comprobar que se ha cargado correctamente
5. Mostrar la imagen
6. Presionar una tecla
7. Si ESCAPE, finalizamos el bucle
8. Liberamos memoria
9. Finalizamos el programa

Las librerías OpenCV

- **Ejemplo 02: Mostrar una imagen perteneciente a un vídeo**

```cpp
// e02_showimageFromVideo.cpp: Load image from video stream.

#include "opencv\cv.hpp"
#include <iostream>

#define ESCAPE 27

using namespace cv;
using namespace std;

int main(int argc, char* argv[])
{
    // Name of the video
    char* name = "honda-asimo.avi";

    // "Capture" image extraction
    VideoCapture capture;

    // Object instantiation
    Mat frame;

    // keyboard pressed
    char keypressed = 0;

    // Check the success for image reading
    bool success;

    // Load image from disk
    capture.open(name);
    // if not success, exit program
    if (!capture.isOpened()){
        cout << "error in VideoCapture: check path file" << endl;
        return 1;
    }
```

```cpp
    }

    // Create window canvas to show image
    namedWindow("original", CV_WINDOW_AUTOSIZE);

    while (keypressed != ESCAPE){
        // read frame by frame in a loop
        success = capture.read(frame);

        // if no success exit program
        if (success == false){
            cout << "Cannot read the frame from file" << endl;
            return 1;
        }

        // Show image in window
        imshow("original", frame);

        // save the pres in keypressed
        keypressed = waitKey(0);
    }

    // Free memory
    destroyWindow("original");
    capture.release();
    // End of the program
    return 0;
}
```

Las librerías OpenCV

- Ejemplo 03: Mostrar una imagen desde la cámara

1. Abrir la cámara
2. Comprobar que se ha cargado correctamente
3. Obtenemos una imagen
4. Comprobar la captura
5. Mostrar la imagen
6. Pulsar una tecla
7. Si ESCAPE, finalizar el bucle
8. Liberar memoria
9. Finalizar programa

Las librerías OpenCV

- Ejemplo 03: Mostrar una imagen de una cámara

e03_showimageFromCamera.cpp

(Global Scope)

```cpp
// e02_showimageFromCamera.cpp: Load image from camera.

#include "opencv\cv.hpp"
#include <iostream>

#define ESCAPE 27

using namespace cv;
using namespace

int main(int argc, char* argv[])
{
    // Start "Capture" in default device (0)
    VideoCapture capture(0);
    if (!capture.isOpened()){
        cout << "error in VideoCapture: check device" << endl;
        return 1;
    }

    // Object instantiation
    Mat frame;

    // keyboard pressed
    char keypressed = 0;

    // Check the success for image reading
    bool success;

    // Create window canvas to show image
    namedWindow("original", CV_WINDOW_AUTOSIZE);
```

e03_showimageFromCamera.cpp

(Global Scope)

```cpp

    while (keypressed != ESCAPE){
        // read frame by frame in a loop
        success = capture.read(frame);
        //capture >> frame;
        // if no success exit program
        if (success == false){
            cout << "Cannot read the frame from file" << endl;
            return 1;
        }

        // Show image in window
        imshow("original", frame);

        // save the pres in keypressed
        keypressed = waitKey(0);
    }

    // Free memory
    destroyWindow("original");
    capture.release();
    // End of the program
    return 0;
}
```