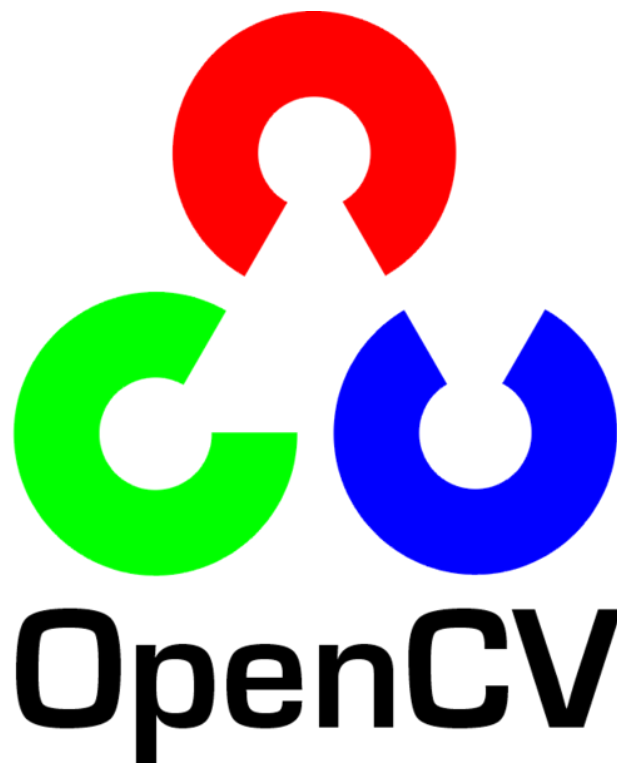


[CTT451] – [Nhập môn Thị giác Máy tính]

Tháng 3/2013

## SỬ DỤNG OPENCV



# MỤC LỤC

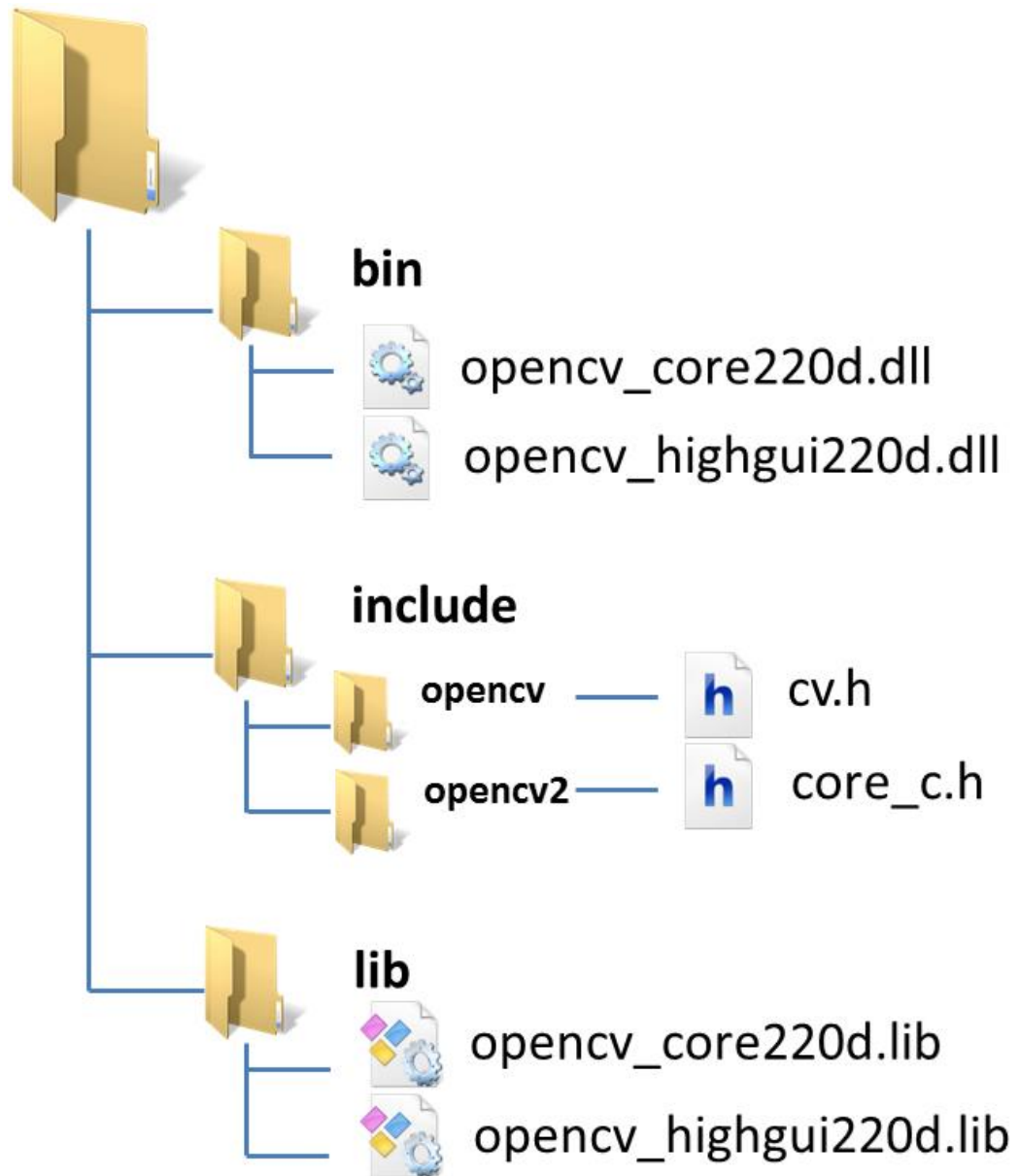
1.	Chuẩn bị:.....	2
2.	Cấu hình OpenCV với Visual Studio C++ .....	2
3.	Chương trình đầu tiên .....	6
4.	Chương trình thứ hai: .....	7
5.	Bài tập .....	8

# CẤU HÌNH OPENCV VỚI VS C++

## 1. Chuẩn bị:

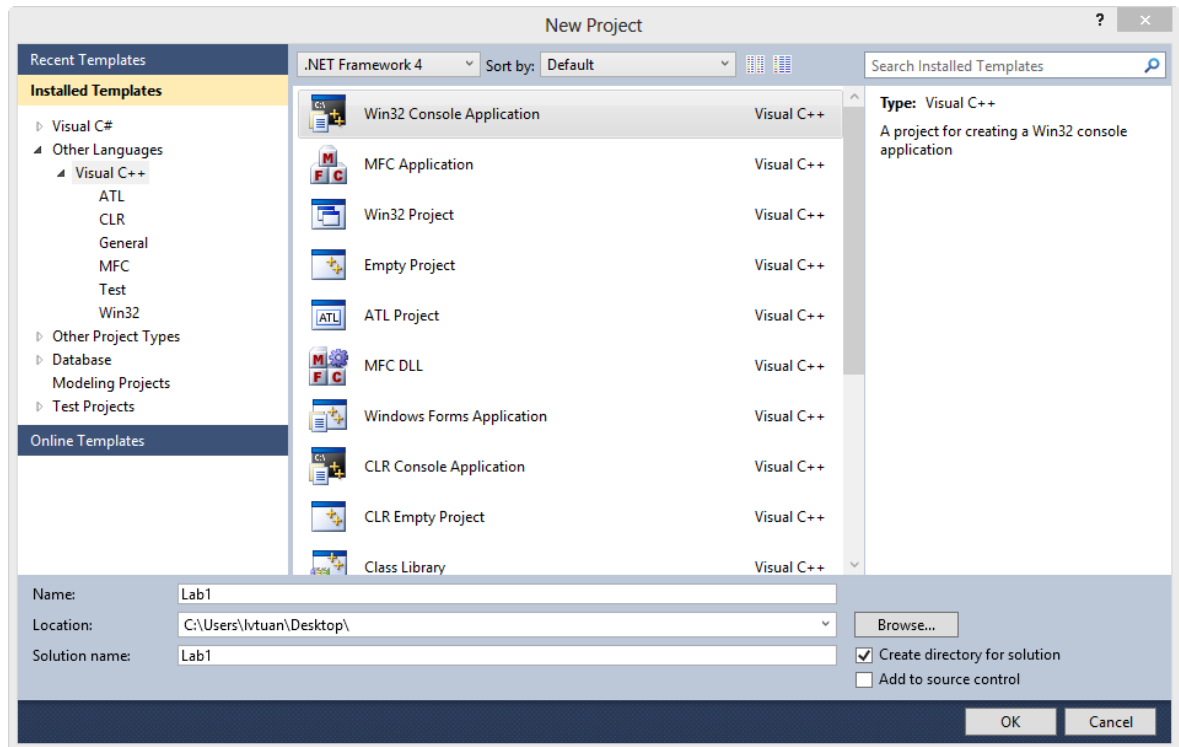
Download OpenCV tại: <http://opencv.org/downloads.html>, hoặc download thư viện OpenCV đã được biên dịch sẵn: <http://www.mediafire.com/?gda0fgqi5u8h7>.

Gồm 3 thư mục như sau:

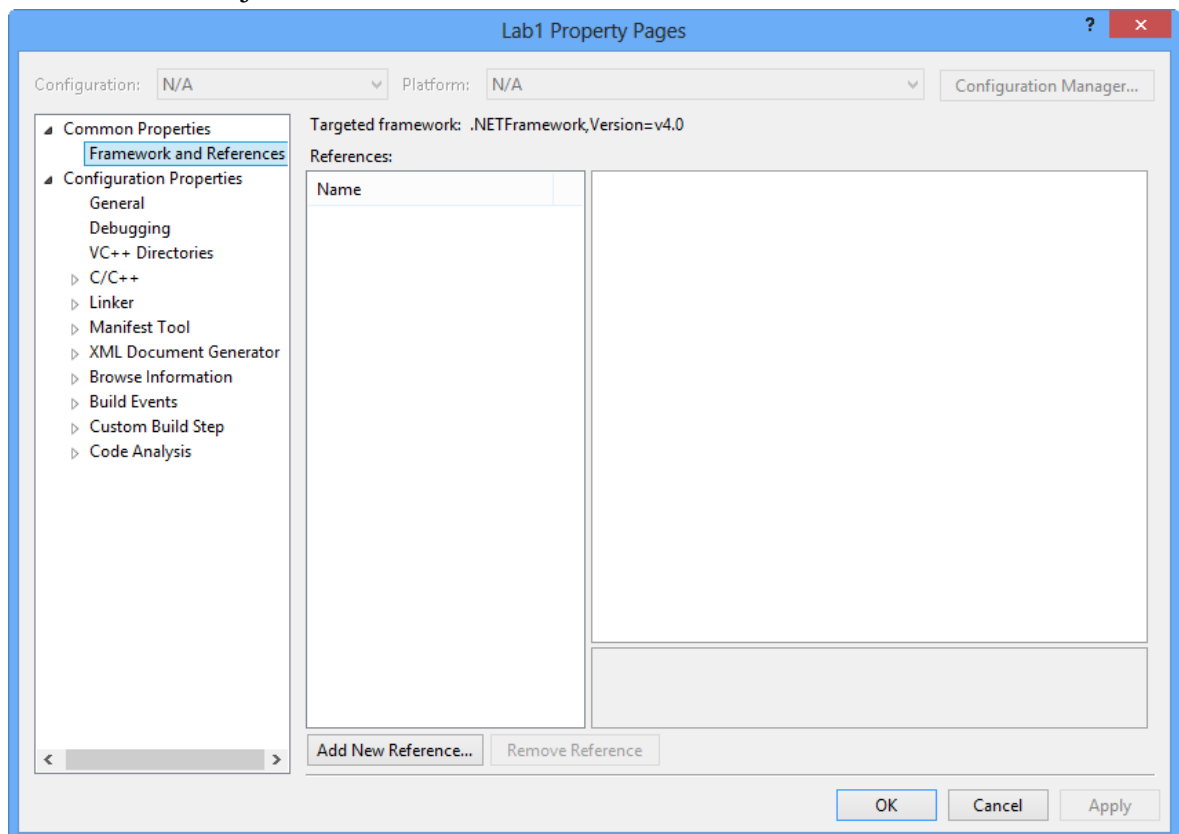


## 2. Cấu hình OpenCV với Visual Studio C++

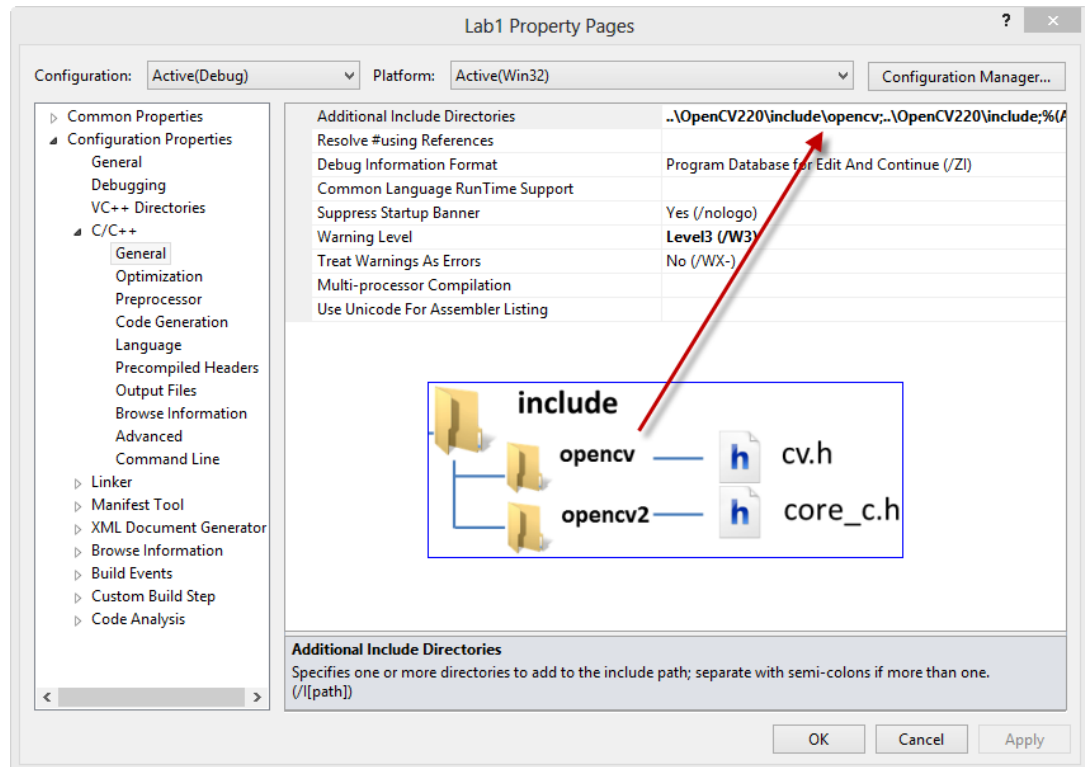
Tạo project Lab1:



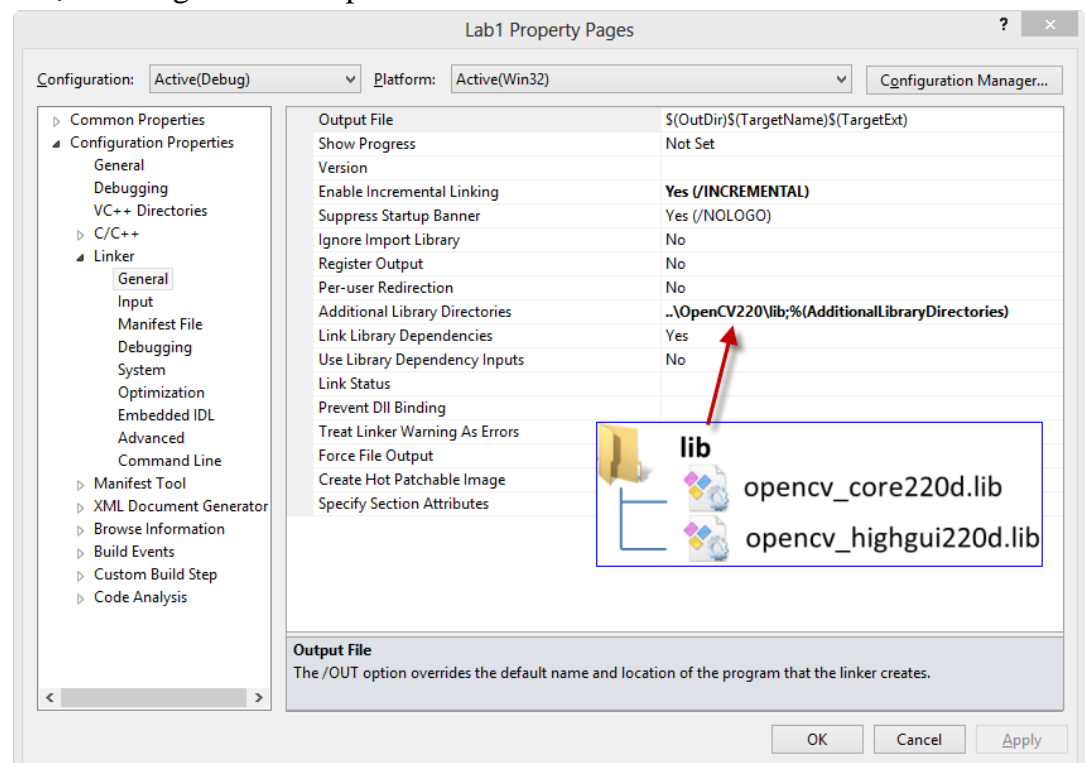
Từ Menu Project → References...



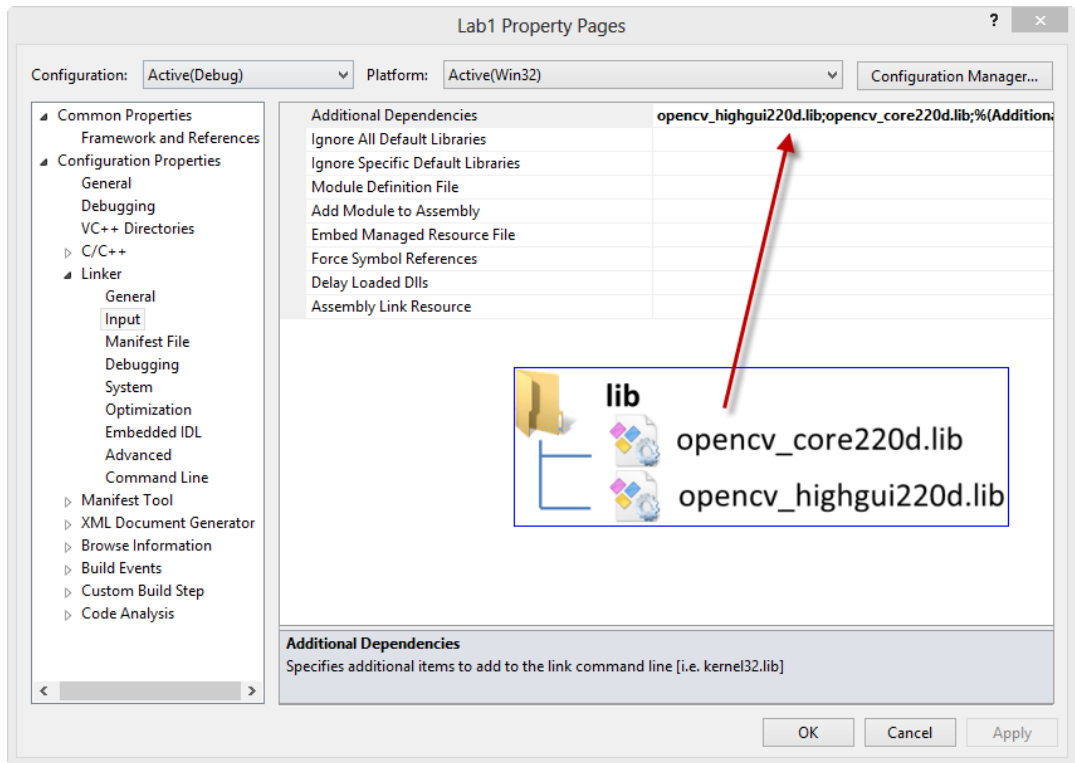
## Chọn Configuration Properties → C/C++ → General



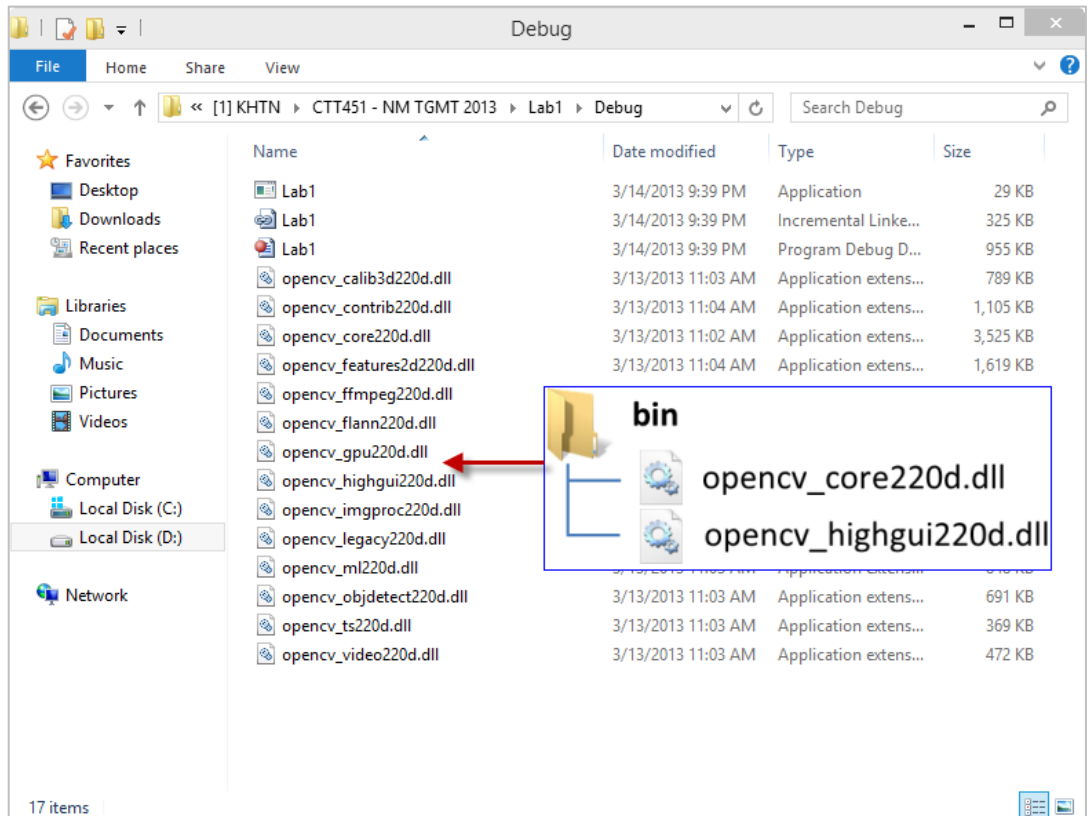
## Chọn Configuration Properties → Linker → General



Chọn Configuration Properties → Linker → Input



Sao chép các file đuôi \*.dll vào thư mục debug của Project



### 3. Chương trình đầu tiên

Viết chương trình tải và hiện một tập tin ảnh cho trước lên màn hình.

Sau khi tạo project, bổ sung đoạn chương trình sau:

```
#include "stdafx.h"
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>

int _tmain(int argc, _TCHAR* argv[])
{
    IplImage * img = cvLoadImage("C:\\\\hinh1.jpg");
    if ( img != NULL )
    {
        cvNamedWindow( "My window" );
        cvShowImage( "My window", img );
        cvWaitKey(); //Đợi người dùng nhấn 1 phím bất kỳ
        cvReleaseImage( &img ); //Giải phóng vùng nhớ
        cvDestroyWindow( "My window" ); //Đóng cửa sổ
    }
    return 0;
}
```



Đoạn chương trình trên sẽ tải một ảnh lên bộ nhớ và hiển thị ra màn hình. Ta xem xét một số dòng lệnh chính:

```
IplImage * img = cvLoadImage("C:\\hinh1.jpg");
```

Hàm `cvLoadImage()` thực hiện tải ảnh dựa vào tên file được truyền vào, đồng thời cấp phát một vùng nhớ cần thiết cho cấu trúc dữ liệu ảnh. Hàm này trả về một con trỏ tới vùng nhớ được cấp phát trên.

```
cvNamedWindow( "My window" );
```

Hàm `cvNamedWindow` tạo một cửa sổ trên màn hình để chứa và hiển thị ảnh.

```
cvShowImage( "My window", img );
```

Chúng ta đã có một ảnh dưới dạng một con trỏ `IplImage *`, và hiển thị nó lên cửa sổ vừa tạo thông qua hàm `cvShowImage()`.

#### 4. Chương trình thứ hai:

Viết chương trình phát một đoạn video:

```
#include <highgui.h>
int main( int argc, char** argv )
{
    cvNamedWindow( "Example2", CV_WINDOW_AUTOSIZE );
    CvCapture* capture =
    cvCreateFileCapture("C:\\video1.avi");
    IplImage* frame;
    while(1) {
        frame = cvQueryFrame( capture );
        if( !frame ) break;
        cvShowImage( "Example2", frame );
        char c = cvWaitKey(33);
        if( c == 27 ) break;
    }
    cvReleaseCapture( &capture );
    cvDestroyWindow( "Example2" );
}
```

```
CvCapture* capture = cvCreateFileCapture("video1.avi");
```



Hàm `cvCreateFileCapture()` với tham số truyền vào là tên đoạn video cần load đoạn video và sẽ trả về con trỏ tới cấu trúc `CvCapture`. Cấu trúc này sẽ chứa toàn bộ thông tin đoạn video đã được đọc.

```
frame = cvQueryFrame( capture );
```

Bên trong vòng lặp `while`, bắt đầu đọc đoạn video. Hàm `cvQueryFrame()` bắt frame kế tiếp trong đoạn video trên vào vùng nhớ (vùng nhớ này là một phần của cấu trúc `CvCapture` đã được cấp phát trước đó). Một con trỏ được trả về cho frame được bắt giữ này. Vì `cvQueryFrame` sử dụng vùng nhớ đã được cấp phát cho cấu trúc nên không cần gọi hàm `cvReleaseImage()` cho frame này.



## 5. Bài tập

Viết chương trình chuyển ảnh sang ảnh mức xám (grayscale).