



Capturing images

By *Utkarsh* | Published: February 13, 2010

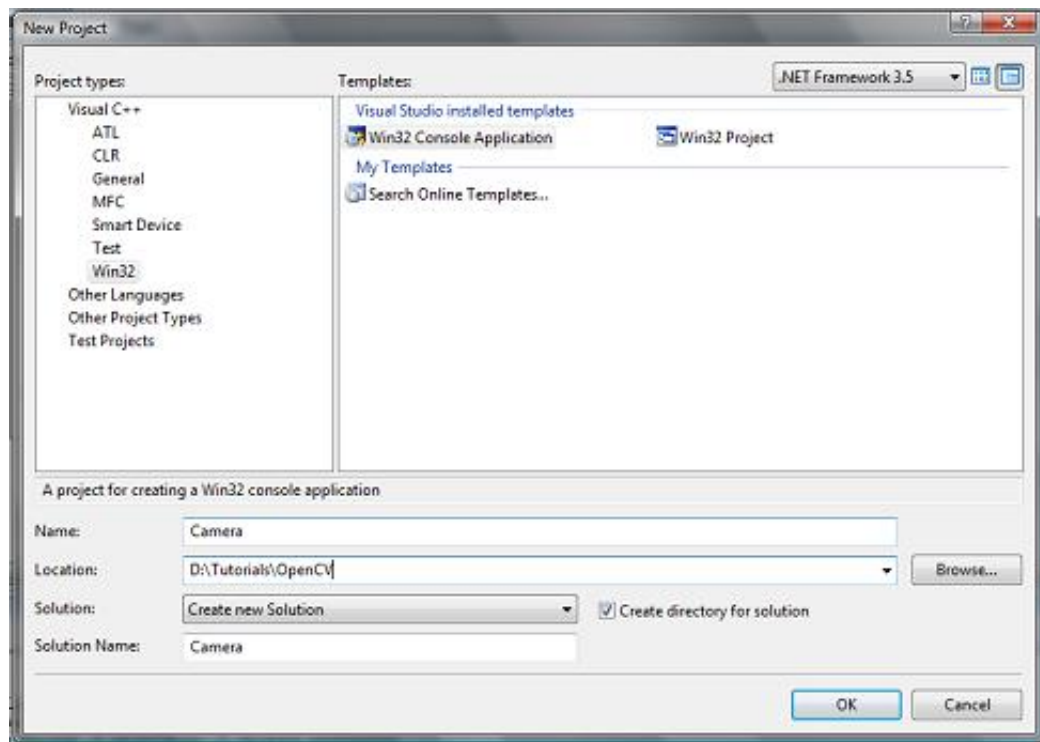
Suka 2.575 orang menyukai ini.

Introduction

Uptill now, we've used static images – images loaded from disk, and we did things with it. Now, you'll see how to capture images from a camera using the functions within the CV library itself. After we've got a working application, we'll add a filter to modify the image, (you can read up about filters in the Processing and filtering images tutorial).

Part A: Capturing images

Create a new Win32 Console Application project... name it whatever you want. I named it Camera.



Click OK. As always, accept the defaults and click Finish. Again, this creates a file called Camera.cpp. Open it up. We'll be modifying it.

```
// Camera.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
int main()
{
    return 0;
}
```

Above is the original Camera.cpp. We'll start off by including OpenCV headers. Add these lines at the top of Camera.cpp:

```
#include <cv.h>
#include <highgui.h>
```

Also, go to Project > Camera Properties > Linker > Input, and put the following piece of text in the Additional Dependencies: "cv.lib cvaux.lib cxcore.lib highgui.lib", without the quotes of course.

With all this done, we'll start off with the actual capturing part. In OpenCV, you create a CvCapture structure. The structure is defined within the OpenCV headers and it "represents" a camera. If you have more than one camera connected to your computer, you can have this object point to any one camera.

After the structure is created (successfully that is), you can ask it give you the current frame. If you repeatedly ask for the current frame, you can get images in real time. And that is exactly what we'll be doing: an infinite loop in which you request for the current image.

We'll start of by creating the capture structure. Add these lines to the main function:

```
CvCapture* capture = 0;
capture = cvCaptureFromCAM(0);
```

capture is the pointer to a CvCapture structure. We initially set it to nothing (the zero). Then you initialize

capture to point to the very first camera on your system (camera indices start from 0).

Once we've tried initializing capture, we need to make sure that it initialized properly. To do that, just check if capture is non zero. If it is, then it points to a camera properly. Add this code:

```
if(!capture)
{
    printf("Could not initialize capturing...\n");
    return -1;
}

cvNamedWindow("video");
```

If capture is nonzero, it got initialized. If not, we print out a message and exit the program these itself. And we also create a new window, with the title "video".

Now that we have a CvCapture structure, we can request OpenCV to give us frames captured from the camera whenever we want. We'll start off with an infinite loop. Add this to the main function:

```
while(true)
{
```

Next, we create an IplImage structure that will store the image captured from the camera:

```
IplImage* frame = 0;
```

Then we request OpenCV to give us the latest frame using the cvQueryFrame function:

```
frame = cvQueryFrame(capture);

if(!frame)
    break;
```

The cvQueryFrame function takes capture as a parameter. If you had multiple cameras attached, you could take images from the ones you wanted, and ignore the rest.

Then we check if the image we got is valid or not. If its zero, its an invalid image and we immediately get out of the infinite loop.

Finally, you display the image:

```
cvShowImage("video", frame);
```

Since this is an infinite loop, you need to have a mechanism to get out of it. We'll use the cvWaitKey() function I told you about earlier. We'll use it for two purposes: 1) Create a delay between capturing images (to restrict the number of images taken per second) 2) Check which key was pressed (to be able to decide when to quit).

First, to create a delay:

```
int c = cvWaitKey(20);
```

This makes sure that the program halts for 20 milliseconds. And in case there is a key press during this period... the ASCII value of the key goes into the variable c.

We can then check the value of c to decide when to get out of the infinite loop. And we do that using the following:

```
        if((char)c==27 )
            break;
    }
```

27 is the ASCII code for the Escape key. So whenever you press the Escape key, the infinite loop ends.

Once the loop ends, we need to do a bit of cleanup. We need to release the camera so that other applications can use it. And we do that like this:

```
    cvReleaseCapture(&capture);
    return 0;
}
```

And that finishes up our program! As a checkpoint, here's the entire program:

Compile your program and execute it. You should see a windows with some video in it.

Live effects!

Now that we have a working camera capturing application, you can use OpenCV to process images in real time!

You can really do anything – search for circles, or detect patches of colour, or anything else you can think of. For the sake of explanation, I'll show how to erode the image.

Just before the line `cvShowImage("video", frame);` in the code, add this line:

```
if(!frame)
    break;

cvErode(frame, frame, 0, 2); // ADD this line
cvShowImage("video", frame);
```

Whenever a frame is taken in from the camera, it is eroded and then displayed. So what you ultimately get is an eroded video of yours

Conclusion

In this tutorial you learned how to use OpenCV to access all the camera on your computer and also do things with the images you get. Next you'll learn how to create a bit more advanced user interfaces.

Got questions or suggestions? Or criticism? Let me know! **Leave a comment below!**

Next Parts

This post is a part of an article series on OpenCV for Beginners

1. [Why OpenCV?](#)

2. [Installing and getting OpenCV running](#)
3. [Hello, World! With Images!](#)
4. [Filtering Images](#)
5. [Capturing Images](#)
6. [HighGUI: Creating Interfaces](#)

Issues? Suggestions? Visit the [Github issue tracker for AI Shack](#)

Back to top

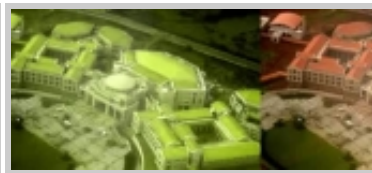
Suka 2.575 orang menyukai ini.



[Capturing images with DirectX](#)



[HighGUI: Creating Interfaces](#)



[Filtering images](#)

25 Comments



Apex

Posted September 8, 2010 at 9:26 pm | [Permalink](#)

Trying to capture an image using the inbuilt webcam in my laptop..
However i always end up getting a black window with the following error:
The program '[5300] camtry.exe: Native' has exited with code -1 (0xffffffff).
OR
the program has exited with code 0.

What seems to be the problem??

[Reply](#)



Utkarsh

Posted September 19, 2010 at 11:59 pm | [Permalink](#)

Hmm... weird. What webcam do you have? Maybe your webcam isn't supported.
Tried using videoInput?

[Reply](#)



Uni

Posted October 24, 2010 at 3:16 am | [Permalink](#)

Hey

I tried this out. Code compiled and all. But at execution, my webcam switches on, but the window titled "video" is all black. I don't see any video!

And there is no error/warning too.
Any idea what's wrong? :S

[Reply](#)



Utkarsh

Posted October 25, 2010 at 7:08 am | [Permalink](#)

Are you sure the code you wrote is correct? Also, try using videoInput if you're using Windows: <http://www.aishack.in/2010/03/capturing-images-with-directx/>

[Reply](#)



pato wlm

Posted December 14, 2010 at 5:38 am | [Permalink](#)

Hello, I just have a little question. I noticed that in this code your header files appear as:

```
#include <cv.h>
#include <highgui.h>
```

But on your others post (So far) they appear as:

```
#include "cv.h"
#include "highgui.h"
```

Is that the same, or is there a difference? Sorry if it's a noob question, but I'm really new to programming

[Reply](#)



Utkarsh

Posted December 15, 2010 at 12:44 am | [Permalink](#)

There is a slight difference, but doesn't matter much. The <> includes headers from the standard include directories. The "" includes headers from the current directory. If not found in the current directory, it checks the standard directories.

Usually, <> is used for standard files and "" for custom headers. In our case, we don't modify the headers, so it ultimately checks the standard directory... effectively the same as <>

[Reply](#)



Filippo

Posted December 31, 2010 at 12:00 am | [Permalink](#)

Hi! This article is really good! Thanks!

I have a question for you... I can get images from my built-in webcam or from an usb cam with no problems. The only thing I noticed is that using a resolution of 640×480, when I do face detection with haarcascade_frontalface_default.xml the images are shown very slowly. With 320×240 they are very fluid. Is that normal or something is wrong?

Thank you very much for your help.

Filippo

[Reply](#)



Utkarsh

Posted January 11, 2011 at 8:00 pm | [Permalink](#)

Hmm... I think this should be expected. You've increased the image size by 4. How bad is the slow down?

[Reply](#)



roseworld

Posted January 9, 2011 at 3:18 am | [Permalink](#)

Hi Utkarsh,

Thank you soooo much for your help and good code,

kindly please, if you can give us example for capturing faces from two Webcam working in synchronous

[Reply](#)



Utkarsh

Posted January 11, 2011 at 7:51 pm | [Permalink](#)

Just create a new CvCapture structure and make it point to the second device (cvCaptureFromCAM). I think this should work.

[Reply](#)



SAS

Posted January 17, 2011 at 6:50 pm | [Permalink](#)

hi i'm a beginner to opencv.. how to read an image in rgb matix form(i.e to manipulate the R,G and B matrices)??

[Reply](#)



Utkarsh

Posted January 19, 2011 at 9:39 pm | [Permalink](#)

Did you try using the cvSplit function? It splits a BGR image into individual channels. Then you can work with individual channels. Once you're done manipulating those matrices, you can use cvMerge to combine the channels into a BGR image.

[Reply](#)



ferros

Posted February 14, 2011 at 9:16 pm | [Permalink](#)

Hi Utkarsh

Wanna ask how can i display the images captured in a picture box? Is it possible?

[Reply](#)



Utkarsh

Posted February 14, 2011 at 9:20 pm | [Permalink](#)

Yes, it's possible. If you're using EmguCV, it comes with a control that displays OpenCV's images. If not, you'll have to work with the operating system API and figure out how to paint an image.

[Reply](#)



ferros

Posted February 14, 2011 at 9:27 pm | [Permalink](#)

Sorry, is it the EmguCV is another library like OpenCV? How i going to use it, is it need to install EmguCV?

[Reply](#)



Utkarsh

Posted February 14, 2011 at 9:30 pm | [Permalink](#)

It's the .NET version of OpenCV. A wrapper to interface .NET and the opencv DLLs.

[Reply](#)



Eugene

Posted March 10, 2011 at 4:38 am | [Permalink](#)

Hey there.

Your website is brilliant. It has provided me with lots of help.

Just one favor. Is it possible for you to upload capturing images code for matlab because i have to build a source code using matlab.

Thanks in advance

[Reply](#)



Utkarsh

Posted March 11, 2011 at 4:13 pm | [Permalink](#)

Capturing images in matlab is super simple as far as I remember. There's a tutorial in the image acquisition toolbox documentation.

[Reply](#)



Eugene

Posted March 10, 2011 at 6:59 pm | [Permalink](#)

Hey there.

Thanks for the code! it's awesome.

There is a question for you. As i can see, this code is for us to capture image continuously right. so it does not actually detect a line and only then capture the image of it?

For example, if i have a flat surface running under the camera, and there is a scratch at the end of the surface, how can i make the camera to just take the picture of the scratch rather than every image of the surface.

Thanks

[Reply](#)



Utkarsh

Posted March 11, 2011 at 4:14 pm | [Permalink](#)

Not sure. Are there any distinctive features about the scratch?

[Reply](#)



Eugene

Posted March 11, 2011 at 6:58 pm | [Permalink](#)

Hey thanks for the reply.

I have looked into the image toolbox. And I found out smtg related to them such as kernel conv2 functions. But I have no idea how to start or what should I do with the functions to build an image capturing code. So would be really glad if u can help me with it.

Thanks a lot

[Reply](#)



Utkarsh

Posted March 11, 2011 at 9:20 pm | [Permalink](#)

Check the image acquisition toolbox's documentation. There's a tutorial in there. Or look around on the internet – you'll definitely find info about this.

[Reply](#)



Aaron

Posted April 1, 2011 at 12:49 pm | [Permalink](#)

i am getting 10 undeclared identifier .. Could you help me out with this . URGENT . Thank you

[Reply](#)



Arun

Posted June 30, 2011 at 4:14 pm | [Permalink](#)

Hi,

I was working on capturing jpeg images from a file and saving it.

Since there will be multiple frames I was thinking of saving it as numbers. But I am facing a problem while saving.

The function `cvSaveImage("text.jpg",frame);` takes in only a name along with file format within a ". How can i make it a varying number from 0.jpg – n.jpg .

I tried using strings and appending strings to create file names but they cannot be used as input parameter for `cvSaveImage`, Please help

Thank you

[Reply](#)



Utkarsh

Posted July 13, 2011 at 9:22 am | [Permalink](#)

Hi! Well, this is a C/C++ problem. You need to use `sprintf`

[Reply](#)

Post a Comment

Your email is *never* published nor shared. Required fields are marked *

Name *

Email *

Website

Comment

You may use these HTML tags and attributes <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

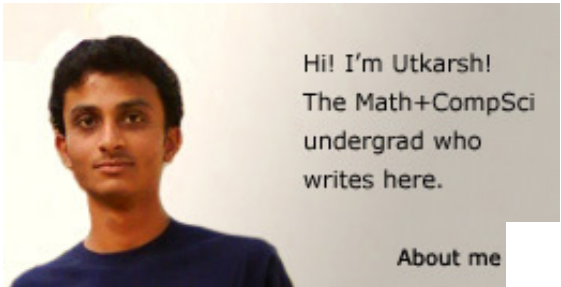
Post Comment

Free updates

Email Address

[More info](#)

1000+ subscribers!



Popular Posts

- [The Canny Edge Detector](#)
- [Implementing Canny Edges from scratch](#)
- [Image Moments](#)
- [Predator: Tracking + Learning](#)
- [Predator: Tracking + Learning](#)

More on AI SHack

Popular **Beginners** **Utkarsh's favs**

The Canny Edge Detector

Implementing Canny Edges from
scratch

Image Moments

Predator: Tracking + Learning

Predator: Tracking + Learning

About me



My name is Utkarsh Sinha, and I'm an undergraduate student, pursuing B.E. Computer Science + M.Sc. Mathematics. Here, I help you understand ideas in Artificial Intelligence, using a not so techy and

mathematical language. And in the process, learn more about Artificial Intelligence myself.

[Read more at the about page](#)

Created by Utkarsh Sinha - d312a67 - [Linode](#)