



HighGUI: Creating Interfaces

By [Utkarsh](#) | Published: February 16, 2010

Suka 2.575 orang menyukai ini.

Introduction

In this tutorial, you'll learn how to add trackbars to windows. And also how to detect mouse click events within a window. An application of these flexibilities would be being able to dynamically control things within your program... like changing the amount of erode without recompiling the code.

In this tutorial, we'll create an application which grabs images from your camera and lets you adjust the brightness and contrast of the image... on the fly. And on clicking, displays the X and Y coordinates of that point.

On the fly brightness and contrast, courtesy OpenCV

We'll start off by creating a program that will constantly input images from a camera. If you want, you can read about that in detail in [Capturing Images](#). For now, create a new project of Win32 Console Application. Name it whatever you want. I named it Interface. Click OK and accept the default options by clicking Finish.

We start off by including the OpenCV header files and creating a camera capture object:

```
#include <cv.h>
#include <highgui.h>
int main()
{
    CvCapture* capture = 0;
    capture = cvCaptureFromCAM(0);
    if(!capture)
    {
```

```
    printf("Could not initialize capturing...\n");  
    return -1;  
}  
cvNamedWindow("video");
```

The code above creates a capture structure pointing to camera #0 and creates a window named "video".

All this code has been explained in detail in the previous tutorial, Capturing Images. So if you have any doubts, please refer to it.

Now, we create two variables. These variables will hold the value of the trackbars we'll create in a moment:

```
int bright=128, contrast=26;
```

And now we actually create the trackbars:

```
cvCreateTrackbar("brightness", "video", &bright, 255, NULL);  
cvCreateTrackbar("contrast", "video", &contrast, 50, NULL);
```

The cvCreateTrackbar takes in five parameters:

1. The name of the trackbar to be created
2. The name of the window in which it will be placed
3. A pointer to a variable (this will hold the value of the trackbar)
4. The maximum value of the trackbar (the minimum is always 0)
5. A callback function (which is called whenever the position of the trackbar is changed)

We create two trackbars, one named "brightness" and the other named "contrast". Both are placed in the "video" window. We send the addresses of brightness and contrast as pointers too. So, whenever the user moves the trackbar, these variables are updated.

The maximum possible value for brightness is 255 and for contrast is 50. And because we don't use the callback functions, we send NULL.

Now we get to the infinite loop. Within this infinite loop, we request for frames:

```
while(true)  
{  
    IplImage* frame = 0;  
    frame = cvQueryFrame(capture);  
    if(!frame)  
        break;
```

Again, if you have any doubts with whats going on, have a look at Capturing Images.

Now, frame contains the image captured from the camera. We need to modify the image, based on the value stored in brightness and contrast. And we do that using code developed in the Processing and Filtering Images tutorial.

For brightness, we could either reduce or increase brightness. Since the minimum value of a trackbar is always 0, subtracting 128 from brightness would give us values from -128 to 127... which is good enough for our purposes (reduce to increase brightness). Here's the code:

```
cvAddS(frame, cvScalar(bright-128,bright-128,bright-128), frame);
```

And then, we modify the contrast using the `cvScale` command. Since this trackbar has a from from 0 to 50... subtracting 25 from contrast will give us a range of -25 to 25. Now, multiplying by a negative number makes no sense (you can't have negative intensity of pixel... atleast not on a monitor/LCD).

Positive value of the contrast trackbar physically means increasing contrast. For that, we multiply. Negative physically means is decreasing contrast. And that means dividing by a constant factor. To do this, you use the following code:

```
if(contrast>25)
    cvScale(frame, frame, contrast-25);
else if(contrast<25)
    cvScale(frame, frame, 1/(double)(25-contrast));
```

If you noticed the code well enough.. you'll see that nothing happens when `contrast = 0` (or, value 25). That's because, you can't do either thing... you can't divide by 0... nor will multiplying by 0 give any meaningful result.

Finally, we display the image in the window "video" and wait for key presses. If the Esc key (ASCII = 27) isn't pressed you continue with the loop. Else, you exit.

```
cvShowImage("video", frame);
int c = cvWaitKey(20);
if((char)c==27)
    break;
}

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

And this finishes the first part of the tutorial. You can now compile and run the program. You should see the webcam's video feed on your screen... and two trackbars which you can use to modify the brightness and contrast of the image.

Detecting mouse clicks

Now for something different. We'll print out the coordinates of the point the use clicks. Once you have the coordinates, you can do almost anything... make a sribble type application (with live video going on in the background!!), or something even more wierd.

To accomplish this task, we'll use a callback function. A callback function is nothing special... its just that another function calls it. In our case, you'll pass this function (actually, the address of this function) to HighGUI. And HighGUI will call this function whenever any event (like mouse click) occurs.

We'll first write this function. Add it before the main function:

```
void on_mouse(int evt, int x, int y, int flags, void* param)
{
```

The parameters of this functions are provided by the HighGUI library itself. The first parameter is the type of event that occured (left mouse button down/up, etc), followed by the X and Y coordinates. The fourth parameter indicates which buttons were pressed (left/middle/right mouse button, the Ctrl/Alt,Shift button).

And finally some user defined parameter.

For now, we'll just add some printing code:

```
if(evt==CV_EVENT_LBUTTONDOWN)
    printf("Clicked at (%d,%d)\n", x, y);
}
```

So, whenever this function is called, and the event that occurred is "left mouse button was pressed", it will print out the coordinates of the point where the mouse was clicked.

Now, we need to tell the HighGUI window to call this function whenever there is any mouse related event. To do that, add this line:

```
cvNamedWindow("video");
cvSetMouseCallback("video", on_mouse, 0); // ADD this line
int bright=128, contrast=26;
```

We're setting the mouse callback, for the window "video". The callback function is on_mouse (note that there are no parentheses after on_mouse). And finally, because we're not using and user defined parameter, we send a 0.

And we're done! Compile and execute this program and try clicking on the webcam image you see!!

Wrap up

In this tutorial, you learned a bit more about HighGUI and how you can make use of it. You learned about trackbars and using them to affect calculations in your program. You also learned how you can use your mouse.

If you have any suggestions or criticism regarding this tutorial, please do contact me or **leave a comment below!**

Other 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

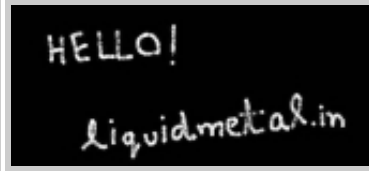
Suka 2.575 orang menyukai ini.



[Capturing images](#)



[Filtering images](#)



[Hello World! with
Images](#)

20 Comments



Arnaud

Posted July 5, 2010 at 1:48 pm | [Permalink](#)

Nice tutorial but I don't understand while most of the people are still using the C syntax with OpenCV 2.0...

[Reply](#)



Utkarsh

Posted July 6, 2010 at 6:16 am | [Permalink](#)

Hi Arnaud! I think it's because almost every resource available, on the internet or as books, uses the C syntax. I have seen a quite a people use the C++ syntax too. And thanks for the nudge, I think i'll put up a tut on using the C++ interface now 😊

[Reply](#)



Brian Lok

Posted October 7, 2010 at 10:47 am | [Permalink](#)

Hi Utkarsh, can you give me some guidance on using OpenCv and OpenGL for my my final year project.

We planned to achieve motion capture with just one camera, but we don't know what functions to use in OpenCv.

One video from Youtube was what we liked to achieve, could you try to identify what the software and functions the video used?

<http://www.youtube.com/watch?v=fYZtmkfWh5g>

Thanks =)

[Reply](#)



Utkarsh

Posted October 9, 2010 at 7:20 pm | [Permalink](#)

That was an impressive video! I think it uses optical flow to estimate which parts of the body were moved. You can read up about optical flow and its functions in OpenCV in the [Learning OpenCV](#) book... it has a detailed explanation for optical flow.

[Reply](#)



Josh

Posted October 31, 2010 at 10:07 am | [Permalink](#)

How would you detect a mouse leaving the window? Is there a mouse exit or mouse out event that can be listened for?

[Reply](#)



Utkarsh

Posted October 31, 2010 at 10:40 am | [Permalink](#)

No. A mouse leave event isn't support. At least yet. You can have a look at possible events here:

http://opencv.willowgarage.com/documentation/user_interface.html#setmousecallback

[Reply](#)



Helen

Posted November 29, 2010 at 10:05 am | [Permalink](#)

hey, it's the best explanation i found.... I've been all day looking for an interface which has a video setting, I didn't know about the trackbar function... it was so useful for my project... If you have a friendly interface for set the configuration of a camera, it would be great...



[Reply](#)



Utkarsh

Posted December 9, 2010 at 12:25 pm | [Permalink](#)

Glad you liked it! What do you mean... an interface to set the configuration?

[Reply](#)



arda

Posted February 11, 2011 at 1:45 pm | [Permalink](#)

This is really informative. Thanks. Can you help me with a small problem about this though? I would like to use the coordinates i get (x,y) in the main program.

```
void on_mouse(int evt, int x, int y, int flags, void* param)
{
    if(evt==CV_EVENT_LBUTTONDOWN)
        printf("Clicked at (%d,%d)\n", x, y);
}

....
cvNamedWindow("video");
cvSetMouseCallback("video", on_mouse, 0);

//FOR EXAMPLE in this line i would like to add two coordinates;
int total=x+y ;

....
```

But as you know x and y seems undeclared.

Thanks again.

[Reply](#)



Utkarsh

Posted February 11, 2011 at 1:48 pm | [Permalink](#)

Make some global variables. Store the X and Y coordinates in those. That might solve your problem.

[Reply](#)



arda

Posted February 11, 2011 at 2:13 pm | [Permalink](#)

YES done. in case someone bumps into the same problem

```
int x,xcoor,y,ycoor,total; //global variables
void on_mouse(int evt, int x, int y, int flags, void* param)
{
    if(evt==CV_EVENT_LBUTTONDOWN)
        printf("Clicked at (%d,%d)\n", x, y);
    xcoor=x;
    ycoor=y;
}

....
cvNamedWindow("video");
```

```
cvSetMouseCallback("video", on_mouse, 0); // ADD this line  
  
int total=xcoor+ycoor ;  
....
```

thanks again . glad to meet you

[Reply](#)



Sreevathsan

Posted March 3, 2011 at 12:45 pm | [Permalink](#)

hi dude,
Really impressed by your website.
we successfully created a GUI using Visual Studio.
our dialog box has a button to open the web cam. We want the webcam window (that is displaying the video) to be integrated with the dialog box.
Thanks in Advance 😊
Cheers 😊

[Reply](#)



Utkarsh

Posted March 3, 2011 at 12:56 pm | [Permalink](#)

It depends on what you're using – Mac, .net, gtk, etc. This is highly platform dependent.

[Reply](#)



Sreevathsan

Posted March 8, 2011 at 7:33 pm | [Permalink](#)

fine thanks for the reply.
Am using windows 7 and Visual Studio 2007 with OpenCV2.1
Can u provide me a code snipet for my problem or any other suggestions.

[Reply](#)



Utkarsh

Posted March 9, 2011 at 9:42 am | [Permalink](#)

Sorry mate, haven't figured that out myself yet 😊

[Reply](#)



Bongani

Posted June 4, 2011 at 10:35 pm | [Permalink](#)

Hi, thnx for a gr8 website and the tutorials, they've really helped me on my retinal recognition mini project. We have successfully managed to create the console app and it works like a charm (thanks to you) but now the problem is integrating it with Windows forms. When I try to include OpenCV in a windows forms project I get errors about unmanaged and managed code,,,when i switch the compliler to /clr as a lot of forums have suggested i get linker errors. Please please help, this is very urgent. Thnx.

[Reply](#)



Utkarsh

Posted June 17, 2011 at 4:23 pm | [Permalink](#)

Try using EmguCV for it. It's a wrapper for OpenCV. It comes with a few controls too (to display an IplImage on a form, etc).

[Reply](#)



Luiza

Posted August 2, 2011 at 8:35 pm | [Permalink](#)

Hello,

I really like your tutorial, but I don't get this syntax:

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

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

I've tried to find about it in the web, but I didn't found much. Thanks.

[Reply](#)



Luiza

Posted August 2, 2011 at 8:35 pm | [Permalink](#)

Hello,

I really like your tutorial, but I don't get this syntax:

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

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

I've tried to find about it in the web, but I didn't find much. Thanks.

[Reply](#)



Utkarsh

Posted August 9, 2011 at 6:58 pm | [Permalink](#)

Well, that's standard C/C++ syntax. You're telling the compiler how the functions in the opencv DLL files want their parameters, what they return, etc.

[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="">` `<code>` `<del datetime="">` `` `<i>` `<q cite="">` `<strike>` ``

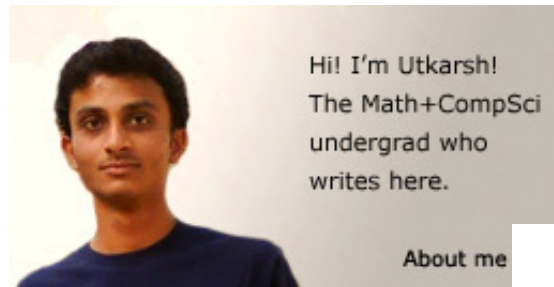
Post Comment

Free updates

Email Address

[More info](#)

1000+ subscribers!



Popular Posts

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

More on AI SHack

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

Scanning QR Codes

The Canny Edge Detector

Implementing Canny Edges from
scratch

Image Moments

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](#)