OpenCV Tutorial C++

Home | OpenCV Lessons | Reference Books | About me

How to Detect Mouse Clicks and Moves

OpenCV supports for detecting mouse events. Mouse events include mouse clicks and movements over an attached OpenCV window.

OpenCV Example Code

It is very simple to do that. All you have do is to define a callback function in the OpenCV C++ code attaching to the OpenCV window. That callback function will be called every time, mouse events occur. That callback function will also give the coordinates of the mouse events. (e.g - (x, y) coordinate of a mouse click).

Here is the simple OpenCV code to detect left, right and middle mouse clicks and mouse movements with its coordinates

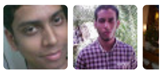
```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace std;
using namespace cv;

void CallBackFunc(int event, int x, int y, int flags, void* userdata)
{
    if  ( event == EVENT_LBUTTONDOWN )
    {
        cout << "Left button of the mouse is clicked - position (" << x << ", " << y << ")" << endl;
    }
    else if  ( event == EVENT_RBUTTONDOWN )
    {
        cout << "Right button of the mouse is clicked - position (" << x << ", " << y << ")" << endl;
    }
    else if  ( event == EVENT_MBUTTONDOWN )
    {
        cout << "Middle button of the mouse is clicked - position (" << x << ", " << y << ")" << endl;
    }
    else if ( event == EVENT_MOUSEMOVE )
    {
        cout << "Mouse move over the window - position (" << x << ", " << y << ")" << endl;

    }
}

int main(int argc, char** argv)
{
    // Read image from file
    Mat img = imread("MyPic.JPG");

    //if fail to read the image
    if ( img.empty() )
    {
        cout << "Error loading the image" << endl;
        return -1;
    }

    //Create a window
    namedWindow("My Window", 1);

    //set the callback function for any mouse event
    setMouseCallback("My Window", CallBackFunc, NULL);

    //show the image
    imshow("My Window", img);

    // Wait until user press some key
    waitKey(0);

    return 0;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

SITE MAP
Home

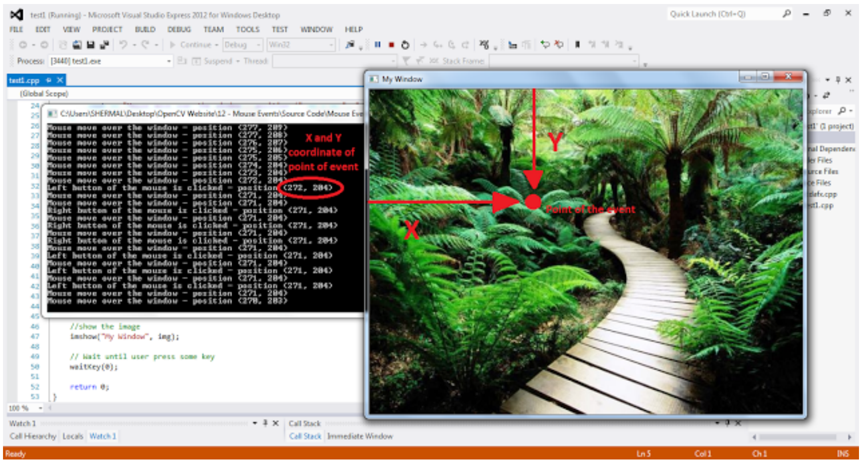
OpenCV Lessons
.. What is OpenCV?
.. Installing & Configu
.. Basics of OpenCV
.. Read & Display Ima
.. Capture Video from
.. Write Image & Vide
.. Filtering Images
.....Change Brightnes
.....Change Contrast
.....Historgram Equal
.....Smooth / Blur Ima
.. How to Add Trackb
.. How to Detect Mou
.. Rotate Image & Vid
.. Color Detection & C
.. Shape Detection &

Reference Books

About Me



You can download this OpenCV visual C++ project from here



test1 (Running) - Microsoft Visual Studio Express 2012 for Windows Desktop
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Process: [3440] test1.exe
test1.cpp
(Global Scope)
C:\Users\SHERMAL\Desktop\OpenCV Website\12 - Mouse Events\Source Code\Mouse Eve
X and Y coordinate of point of event
My Window
Y
X
Point of the event
Watch 1
Call Stack
Ready

```
Mouse move over the window - position (277, 209)
Mouse move over the window - position (277, 208)
Mouse move over the window - position (276, 207)
Mouse move over the window - position (275, 206)
Mouse move over the window - position (275, 205)
Mouse move over the window - position (274, 204)
Mouse move over the window - position (273, 204)
Mouse move over the window - position (272, 204)
Left button of the mouse is clicked - position (272, 204)
Mouse move over the window - position (271, 204)
Mouse move over the window - position (271, 204)
Right button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Right button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Right button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Left button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Left button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Left button of the mouse is clicked - position (271, 204)
Mouse move over the window - position (271, 204)
Mouse move over the window - position (270, 203)
```

```
     //show the image
     imshow("My Window", img);

     // Wait until user press some key
     waitKey(0);

     return 0;
}
```

Detect Mouse Clicks and Moves Over the OpenCV Window

Summary

In the above OpenCV sample code, "CallbackFunc" function will be called on any mouse event (Moving a mouse over the attached OpenCV window is also a mouse event). By using suitable if - else blocks, I printed only left, right and middle mouse clicks and mouse movements over the window.

Here are new OpenCV functions, found in the above example code. If you are not familiar with the other OpenCV functions as well, please go through the other lessons in this tutorial.

- **void setMouseCallback(const string& winname, MouseCallback onMouse, void* userdata = 0)**

This function sets a callback function to be called every time any mouse events occurs in the specified window. Here is the detailed explanation of the each parameters of the above OpenCV function.

  - **winname** - Name of the OpenCV window. All mouse events related to this window will be registered
  - **onMouse** - Name of the callback function. Whenever mouse events related to the above window occur, this callback function will be called. This function should have the signature like the following
    - **void FunctionName(int event, int x, int y, int flags, void* userdata)**
      - **event** - Type of the mouse event. These are the entire list of mouse events
        - EVENT_MOUSEMOVE
        - EVENT_LBUTTONDOWN
        - EVENT_RBUTTONDOWN
        - EVENT_MBUTTONDOWN
        - EVENT_LBUTTONUP
        - EVENT_RBUTTONUP
        - EVENT_MBUTTONUP
        - EVENT_LBUTTONDBLCLK
        - EVENT_RBUTTONDBLCLK
        - EVENT_MBUTTONDBLCLK
      - **x** - x coordinate of the mouse event
      - **y** - y coordinate of the mouse event
      - **flags** - Specific condition whenever a mouse event occurs. See the next OpenCV example code for the usage of this parameter. Here is the entire list of enum values which will be possesed by "flags"
        - EVENT_FLAG_LBUTTON
        - EVENT_FLAG_RBUTTON
        - EVENT_FLAG_MBUTTON

        - EVENT_FLAG_CTRLKEY
        - EVENT_FLAG_SHIFTKEY
        - EVENT_FLAG_ALTKEY
    - **userdata** - Any pointer passes to the **"setMouseCallback"** function as the 3rd parameter (see below)
- **userdata** - This pointer will be passed to the callback function

OpenCV Example to Detect Mouse Clicks While Pressing a Key

I am going to explain you how to detect a mouse event while pressing a key of the keyboard.

The following OpenCV example code will detect left mouse clicks while pressing the "CTRL" key , right mouse clicks while pressing the "SHIFT" key and movements of the mouse over the OpenCV window while pressing the "ALT" key.

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include "opencv2/highgui/highgui.hpp"
#include <iostream>

using namespace std;
using namespace cv;

void CallBackFunc(int event, int x, int y, int flags, void* userdata)
{
    if ( flags == (EVENT_FLAG_CTRLKEY + EVENT_FLAG_LBUTTON) )
    {
        cout << "Left mouse button is clicked while pressing CTRL key - position (" << x << ", " << y << ")" << endl;
    }
    else if ( flags == (EVENT_FLAG_RBUTTON + EVENT_FLAG_SHIFTKEY) )
    {
        cout << "Right mouse button is clicked while pressing SHIFT key - position (" << x << ", " << y << ")" << endl;
    }
    else if ( event == EVENT_MOUSEMOVE && flags == EVENT_FLAG_ALTKEY)
    {
        cout << "Mouse is moved over the window while pressing ALT key - position (" << x << ", " << y << ")" << endl;
    }
}

int main(int argc, char** argv)
{
    // Read image from file
    Mat img = imread("MyPic.JPG");

    //if fail to read the image
    if ( img.empty() )
    {
        cout << "Error loading the image" << endl;
        return -1;
    }

    //Create a window
    namedWindow("My Window", 1);

    //set the callback function for any mouse event
    setMouseCallback("My Window", CallBackFunc, NULL);

    //show the image
    imshow("My Window", img);

    // Wait until user press some key
    waitKey(0);

    return 0;
}
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

You can download this visual C++ OpenCV project from here.

No new OpenCV functions in the above example. If you have some doubt about any OpenCV functions in the above example, please refer to previous lessons.

Next Tutorial : Rotate Image & Video

Previous Tutorial : How to Add Trackbar

Posted by Shermal Fernando

g+1 +3 Recommend this on Google

Is This Helpful : □ Yes (1) □ No (0)

16 comments:

Anonymous February 22, 2013 at 5:58 PM

Exactly what I wanted!! Thanks a lot!!

Reply

Anonymous February 28, 2013 at 4:14 PM

Thanku so much.it helped me a lot

Reply

Anonymous April 22, 2013 at 5:51 PM

super mama

Reply

Anonymous May 29, 2013 at 6:18 PM

hello friends
when I compile the code, I have finaly an error,"error:invalid preprocessing directive #Include

please help me
thanks

Reply

Replies

Shermal Fernando May 29, 2013 at 9:23 PM

You haven't properly configured your IDE or you are including a wrong header file. Please refer to
http://opencv-srf.blogspot.com/2013/05/installing-configuring-opencv-with-vs.html
or
http://opencv-srf.blogspot.com/2011/09/getting-started-with-opencv_16.html

Reply

Anonymous May 30, 2013 at 6:33 PM

in fact, I use the version 12.11,

Reply

Ty Nguyen August 13, 2013 at 1:23 PM

Hi, pro.Thanks for your sharing.
I have an issue that it does not work well when i change the program by changing only mouseEvent() function as following:

```
if(evt==CV_EVENT_LBUTTONDOWN && flags==CV_EVENT_FLAG_CTRLKEY){
printf("%d %d\n",x,y);
}
```

After running, i tried to click left-mouse and press CONTROLKEY simultaneously but nothing happened on the command window as it should be.
Plz help me to check it out.
Thanks.

Reply

Replies

Piotr Pieczywek September 11, 2013 at 4:10 PM

Hi,
As it is stated in OpenCV documentation, the parameter "flags" is a combination of CV_EVENT_FLAG. In Your's case, when You are pressing left mouse button and ctrl key simultaneously flags = CV_EVENT_FLAG_CTRLKEY + CV_EVENT_FLAG_LBUTTON and this gives 9. You can catch this event as flags== CV_EVENT_FLAG_CTRLKEY + CV_EVENT_FLAG_LBUTTON or define Your own event like: #define CV_EVENT_FLAG_CTRLKEY_WITH_LBUTTON 9. Your code should look like this:

```
if(evt==CV_EVENT_LBUTTONDOWN && flags==CV_EVENT_FLAG_CTRLKEY_WITH_LBUTTON){
printf("%d %d\n",x,y);
}
```

Reply

mouse tracking October 1, 2013 at 5:44 PM

You got a definitely helpful blog I've been right here reading for about an hour. I'm a newbie and your accomplishment is quite a lot an inspiration for me.

Reply

Rahul Singh October 8, 2013 at 12:30 PM

@Blog writer : First I told you that your blog is really helpfull and i learn a lot from your blog in short time . And I also want to ask you that i want to simulate cursor by my tracking object .(mean when i move my object the cursor is also moving according to object) can you tell me how the required functions and how to approach this ?

Reply

Replies

Shermal Fernando October 8, 2013 at 1:33 PM

First you have to find the location of a object. For this purpose, refer to
http://opencv-srf.blogspot.com/2010/09/object-detection-using-color-seperation.html
http://opencv-srf.blogspot.com/2011/09/object-detection-tracking-using-contours.html

Then you have to move the cursor to the location of the object. But I am not sure what you mean by cursor? Is it the mouse pointer?

Reply

Jim Pantelis October 21, 2013 at 4:06 AM

After some tutorial (I think Track Bars) you started using C API and not C++ API. For example using IplImage* with needs memory deallocates instead of using Mat that does it automatically from CV2 and on. Also using like cvNamedWindow (C) instead of namedWindow (C++).

Why is that, do you have a reason?

Reply

Replies

Shermal Fernando October 21, 2013 at 9:02 AM

This tutorial is written in C initially. Now I am converting this whole tutorial to C++. This post will also be updated surely in the next week.

Crystopher Vega January 12, 2014 at 2:56 AM

First of all Fernando thank you very much for this tutorial its by far the best I've found. but i was wondering when will you be updating the rest of the tutorials to C++ I got very confused when I got to color detection. one more thing if you have the time and want to, if you could do a small tutorial on how to get video from an ip camera. Once more thank you very much for all this it has been very very helpful

Reply

mouse tracking October 23, 2013 at 4:54 PM

You got a definitely helpful blog I've been right here reading for about an hour. I'm a newbie and your accomplishment is quite a lot an inspiration for me.

Reply

Unknown February 15, 2014 at 8:16 AM

can you suggest me function or macros that can control moment of cursor i.e. I want to control cursor position with my application not by mouse?Your help would be appreciated
Thank you
-Jay Kothari

Reply

Enter your comment...

Comment as: Google Accour ▼

Publish **Preview**

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Template images by 5ugarless. Powered by Blogger.