

# OpenCV Tutorial C++

[Home](#)
[OpenCV Lessons](#)
[Reference Books](#)
[About me](#)

## How to Add Trackbar

Trackbars are very useful in lots of occasions. It enables users to change various parameters while the OpenCV application is running.

( If you have not install and configure OpenCV yet, please refer to [Installing & Configuring with Visual Studio](#) . )

### Simple Use of Trackbars

Whenever you change the position of a trackbar, the value of an integer variable is changed. Using that value, we can change a property of an image or a video. The following example will show you how to do it with OpenCV.

### OpenCV Example of How to Change Brightness and Contrast of an Image with Trackbars

In the following example, I have added two trackbars to change the brightness and contrast of an image. It is iterating in a infinite while loop and applying the brightness and contrast to the image periodically because I want to apply the changes to the image whenever the user changes the position of the trackbar.

```

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

using namespace std;
using namespace cv;

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

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

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

    //Create trackbar to change brightness
    int iSliderValue1 = 50;
    createTrackbar("Brightness", "My Window", &iSliderValue1, 100);

    //Create trackbar to change contrast
    int iSliderValue2 = 50;
    createTrackbar("Contrast", "My Window", &iSliderValue2, 100);

    while (true)
    {
        //Change the brightness and contrast of the image (For more information http://opencv-srf.blogspot.com/2013/07/change-contrast-of-image-or-video.html)
        Mat dst;
        int iBrightness = iSliderValue1 - 50;
        double dContrast = iSliderValue2 / 50.0;
        src.convertTo(dst, -1, dContrast, iBrightness);

        //show the brightness and contrast adjusted image
        imshow("My Window", dst);

        // Wait until user press some key for 50ms
        int iKey = waitKey(50);

        //if user press 'ESC' key
    }
}

```

#### SITE MAP

[Home](#)

#### OpenCV Lessons

[.. What is OpenCV?](#)  
[.. Installing & Configuring v](#)  
[.. Basics of OpenCV API](#)  
[.. Read & Display Image](#)  
[.. Capture Video from File o](#)  
[.. Write Image & Video to Fi](#)  
[.. Filtering Images](#)  
[.....Change Brightness of In](#)  
[.....Change Contrast of Ima](#)  
[.....Histogram Equalization](#)  
[.....Smooth / Blur Images](#)  
[.. How to Add Trackbar](#)  
[.. How to Detect Mouse Clic](#)  
[.. Rotate Image & Video](#)  
[.. Color Detection & Object](#)  
[.. Shape Detection &Trackir](#)

#### Reference Books

#### About Me

#### GOOGLE+ FOLLOWERS

##### OpenCV Tutorials

Follow



639 have us in circles

782

#### FACEBOOK FOLLOWERS

Like Share 2,256 people  
what your fr

#### SEARCH THIS BLOG

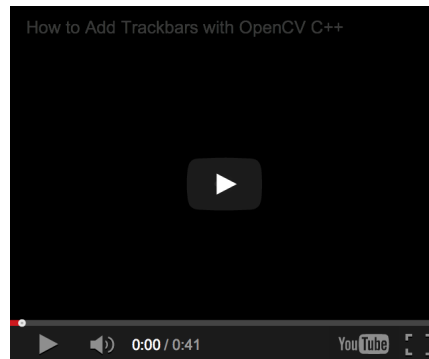
```

    if (iKey == 27)
    {
        break;
    }
}

return 0;
}
////////////////////////////////////
You can download this OpenCV visual c++ example from here.

```

Economizador de Energia do Safari  
Clique para Iniciar o Plug-in do Flash



### Explanation of New OpenCV Functions

- **int** createTrackbar(const string& trackbarname, const string& winname, int\* value, int count, TrackbarCallback onChange = 0, void\* userdata = 0)

This OpenCV function creates a trackbar and attached that trackbar to a specified window

- **trackbarname** - The name of the trackbar
- **winname** - The name of the window to which the trackbar is attached
- **value** - This integer, pointed by this pointer, holds the value associated with the position of the trackbar
- **count** - The maximum value of the trackbar. The minimum value is always zero.
- **onChange** - This function will be called everytime the position of the trackbar is changed. The prototype of this function should be "FunctionName(int, void\*)". The "int" value is the value associate with the position of the trackbar. And "void\*" is any pointer value which you pass as the "userdata" (See the next parameter).
- **userdata** - This pointer variable will be passed as the second parameter of the above function

All other functions have been discussed in the previous lessons. If you have not followed them yet, please visit

- Read & Display Image
- [Change Contrast of Image or Video](#)

which have the all the other OpenCV functions in the above example code.

### Trackbar with Callback Function

In the above example, I have used only 4 parameters for the "createTrackbar" function. But there are 2 more parameters. Here I am going to explain, how to use a callback function using the 5th and 6th parameters of "createTrackbar". The advantage of using the callback function is that it is not required to iterate in a while loop periodically as in the above example.

In the following OpenCV example, I have added two trackbars to change the brightness and contrast of an image. And a callback function is implemented for each trackbar.

```

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

using namespace std;
using namespace cv;

Mat src;

void MyCallbackForBrightness(int iValueForBrightness, void *userData)
{

```

```

Mat dst;
int iValueForContrast = *( static_cast<int*>(userData) );

//Calculating brightness and contrast value
int iBrightness = iValueForBrightness - 50;
double dContrast = iValueForContrast / 50.0;

//Calculated contrast and brightness value
cout << "MyCallbackForBrightness : Contrast=" << dContrast << ", Brightness=" << iBrightness << endl;

//adjust the brightness and contrast
src.convertTo(dst, -1, dContrast, iBrightness);

//show the brightness and contrast adjusted image
imshow("My Window", dst);
}

void MyCallbackForContrast(int iValueForContrast, void *userData)
{
    Mat dst;
    int iValueForBrightness = *( static_cast<int*>(userData) );

    //Calculating brightness and contrast value
    int iBrightness = iValueForBrightness - 50;
    double dContrast = iValueForContrast / 50.0;

    //Calculated contrast and brightness value
    cout << "MyCallbackForContrast : Contrast=" << dContrast << ", Brightness=" << iBrightness << endl;

    //adjust the brightness and contrast
    src.convertTo(dst, -1, dContrast, iBrightness);

    //show the brightness and contrast adjusted image
    imshow("My Window", dst);
}

int main(int argc, char** argv)
{
    // Read original image
    src = imread("MyPic.JPG");

    //If fail to read the image
    if (src.data == false)
    {
        cout << "Error loading the image" << endl;
        return -1;
    }

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

    int iValueForBrightness = 50;
    int iValueForContrast = 50;

    //Create track bar to change brightness
    createTrackbar("Brightness", "My Window", &iValueForBrightness, 100, MyCallbackForBrightness, &iValueForContrast);

    //Create track bar to change contrast
    createTrackbar("Contrast", "My Window", &iValueForContrast, 100, MyCallbackForContrast, &iValueForBrightness);

    imshow("My Window", src);

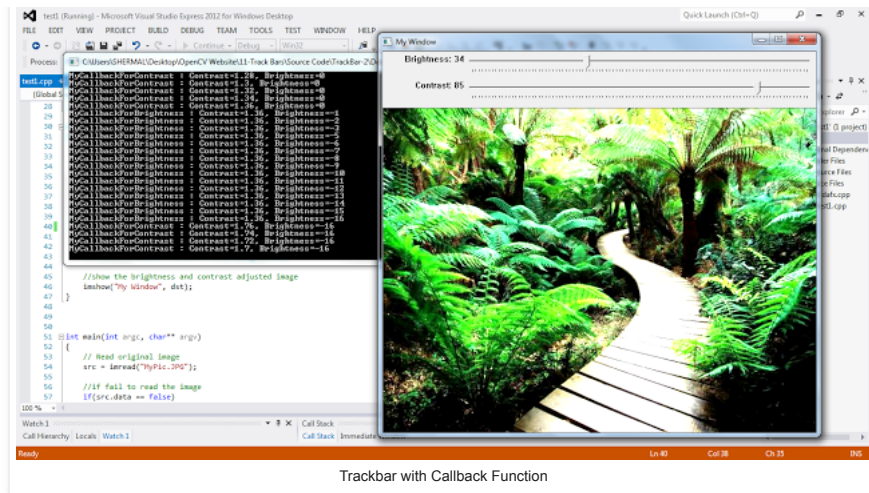
    // Wait until user press some key
    waitKey(0);

    return 0;
}

```

////////////////////////////////////

You can download this OpenCV visual c++ project from [here](http://opencv-srf.blogspot.com.br/2011/11/track-bars.html).



Trackbar with Callback Function

### Explanation

I have used 2 callback functions; **"MyCallbackForBrightness(int, void\*)"** for the **"Brightness"** trackbar and **"MyCallbackForContrast(int, void\*)"** for the **"Contrast"** trackbar.

I have used a global variables, **src** because it should be accessed from all the 3 methods.

Examine closely the 5th and 6th parameters of **"createTrackbar"** method in the **"main"** method.

Whenever the position of the **"Brightness"** trackbar is changed, **"MyCallbackForBrightness(int, void\*)"** will be called. The 1st integer argument holds the value of the position of the **"Brightness"** trackbar. The position of the **"Contrast"** trackbar is passed as the 2nd argument. (Observe the 5th and 6th parameter; **createTrackbar("Brightness", "My Window", &iValueForBrightness, 100, MyCallbackForBrightness, &iValueForContrast);**)

Whenever the position of the **"Contrast"** trackbar is changed, **"MyCallbackForContrast(int, void\*)"** will be called. The 1st integer argument holds the value of the position of the **"Contrast"** trackbar. The position of the **"Brightness"** trackbar is passed as the 2nd argument. (Observe the 5th and 6th parameter; **createTrackbar("Contrast", "My Window", &iValueForContrast, 100, MyCallbackForContrast, &iValueForBrightness);**)

**Next Tutorial : [How to Detect Mouse Clicks and Moves](#)**

**Previous Tutorial : [Filtering Images](#)**

Posted by Shermal Fernando



[+1](#) +3 Recommend this on Google

Is This Helpful : Yes (2) No (0)

## 12 comments:



**Victorz~** April 14, 2013 at 10:50 PM

Your tutorials are so complete and awesome! Thanks a lot. You're the best

[Reply](#)



**Anonymous** May 7, 2013 at 9:54 PM

just what i needed

[Reply](#)



**Ananth Nath** May 24, 2013 at 2:22 PM

Hi your tutorials are awesome. Can i get the tutorial for character recognition from images.my email id is ananthnath@gmail.com

[Reply](#)