

OpenCV Tutorial C++

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

Change Contrast of Image or Video

Changing the contrast is also a point operation on each pixel. The easiest way to increase the contrast of an image is, multiplying each pixel value by a number larger than 1.

$$\text{new_img}(i, j) = \text{img}(i, j) * c \quad c > 1$$

The easiest way to decrease the contrast is, multiplying each pixel value by a number smaller than 1.

$$\text{new_img}(i, j) = \text{img}(i, j) * c \quad c < 1$$

There are more advance methods to adjust contrast of an image such as histogram equalization. Such method adjust the contrast of an image such that color distribution is balanced equally. We will discuss the histogram equalization in the next lesson.

e.g- Say, this is your original image

144	245	132	54
10	62	81	84
99	106	29	7

Example Image

By multiplying each pixel value by 2, you can effectively double the contrast of an image. Here is the image of which the contrast is increased.

I have considered this image as a 8 bit unsigned image. So, any pixel value should be from 0 to 255. If the resulting image has values more than 255, it should be rounded off to 255.

144 * 2	245 * 2	132 * 2	54 * 2
10 * 2	62 * 2	81 * 2	84 * 2
99 * 2	106 * 2	29 * 2	7 * 2

=

255	255	255	108
20	124	162	168
198	212	58	14

Contrast Increased

By multiplying each pixel value by 0.5, you can effectively halve the contrast of an image. Here is the image of which contrast is decreased.

144 * 0.5	245 * 0.5	132 * 0.5	54 * 0.5
10 * 0.5	62 * 0.5	81 * 0.5	84 * 0.5
99 * 0.5	106 * 0.5	29 * 0.5	7 * 0.5

=

72	123	66	27
5	31	41	42
50	53	15	4

Contrast Decreased

Change Contrast of an Image

How to increase or decrease the contrast of an image is demonstrated in the following OpenCV C++ example. Keep in mind that this is the very basic way of changing contrast. In the next lesson, I'll show you how to change the contrast using histogram equalization.

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

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](#)
[.....Histogram Equal](#)
[.....Smooth / Blur Ima](#)
[.. How to Add Trackb](#)
[.. How to Detect Mou](#)
[.. Rotate Image & Vid](#)
[.. Color Detection & C](#)
[.. Shape Detection &](#)

[Reference Books](#)

[About Me](#)

GOOGLE+ FOLLOWER

OpenCV Tutor

Follow



455 have us in circle

578

FACEBOOK FOLLOW

Like Share 1,12

SEARCH THIS BLOG

```

using namespace cv;

using namespace std;

int main( int argc, const char** argv )
{
    Mat img = imread("MyPic.JPG", CV_LOAD_IMAGE_COLOR); //open and read the image

    if (img.empty())
    {
        cout << "Image cannot be loaded..!!" << endl;
        return -1;
    }

    Mat imgH;
    img.convertTo(imgH, -1, 2, 0); //increase the contrast (double)

    Mat imgL;
    img.convertTo(imgL, -1, 0.5, 0); //decrease the contrast (halve)

    //create windows
    namedWindow("Original Image", CV_WINDOW_AUTOSIZE);
    namedWindow("High Contrast", CV_WINDOW_AUTOSIZE);
    namedWindow("Low Contrast", CV_WINDOW_AUTOSIZE);

    //show the image
    imshow("Original Image", img);
    imshow("High Contrast", imgH);
    imshow("Low Contrast", imgL);

    waitKey(0); //wait for key press

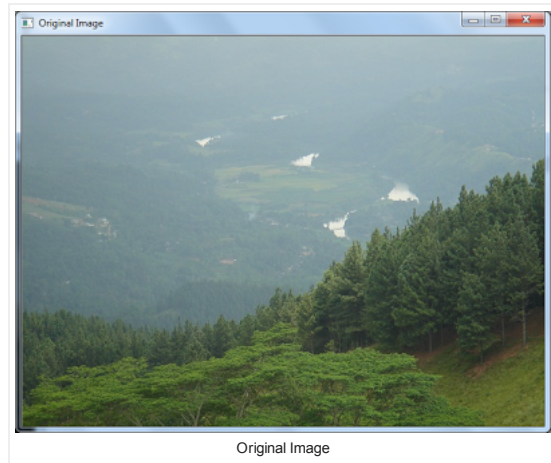
    destroyAllWindows(); //destroy all open windows

    return 0;
}

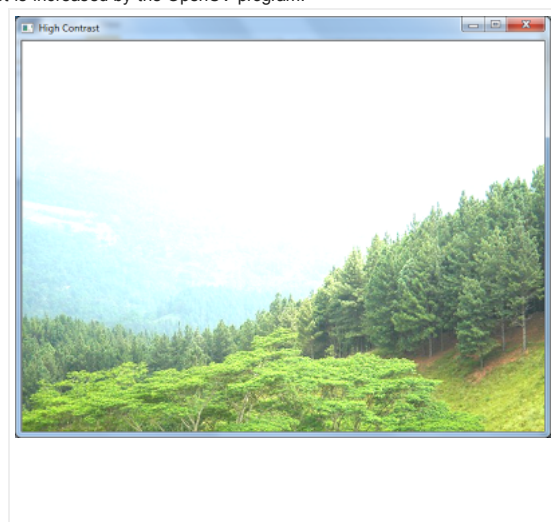
```

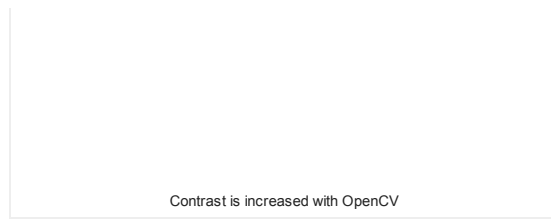
You can download this OpenCV visual c++ project from [here](#). (The downloaded file is a compressed .rar folder. So, you have to extract it using Winrar or other suitable software)

Here is the original image.

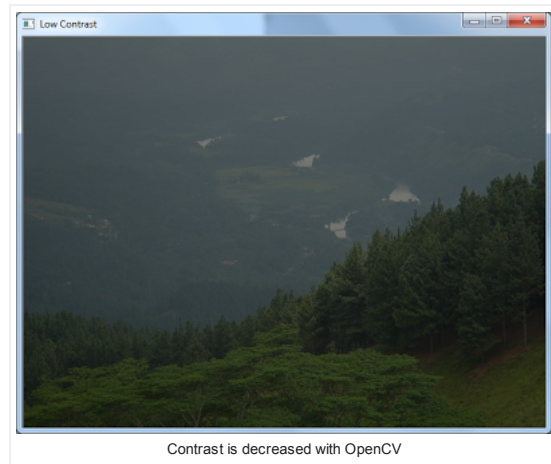


Here is the image of which contrast is increased by the OpenCV program.





Here is the image of which contrast is decreased by the OpenCV program.



New OpenCV methods

- **void** `convertTo(OutputArray m, int rtype, double alpha=1, double beta=0)`

This OpenCV function converts image into another format with scaling. Scaling is done according to the following formula.

$$m[i,j] = \text{alfa} * \text{img}[i,j] + \text{beta}$$

Here is the parameters of this function

- **OutputArray m** - Store the converted image
- **int rtype** - Depth of the output image. If the value of **rtype** is negative, output type is same as the input image. I have used a negative value in the above program because I don't want to change the depth of the original image. Possible inputs to this parameter
 - CV_8U
 - CV_32S
 - CV_64F
 - -1

Complete list of depths can be found in [Basics of OpenCV API](#)

- **double alpha** - Multiplication factor; Every pixel will be multiplied by this value
- **double beta** - This value will be added to every pixels after multiplying with the above value.

Here is the formula again. Here $m[i, j]$ means a pixel at i^{th} row and j^{th} column.

$$m[i,j] = \text{alfa} * \text{img}[i,j] + \text{beta}$$

Change the Contrast of a Video

It is similar to the above program except that you have to change the contrast for each and every frame of the video. Here is the example OpenCV program.

```

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

using namespace cv;
using namespace std;

int main(int argc, char* argv[])
{
    VideoCapture cap("C:/Users/SHERMAL/Desktop/SampleVideo.wmv"); // open the video file for reading

    if ( !cap.isOpened() ) // if not success, exit program
    {
        cout << "Cannot open the video file" << endl;
        return -1;
    }

    //create windows
    namedWindow("Original Video",CV_WINDOW_AUTOSIZE);
    namedWindow("Contrast Increased",CV_WINDOW_AUTOSIZE);
    namedWindow("Contrast Decreased",CV_WINDOW_AUTOSIZE);

    while (1)

```

```

{
    Mat frame;

    bool bSuccess = cap.read(frame); // read a new frame from video

    if (!bSuccess) //if not success, break loop
    {
        cout << "Cannot read the frame from video file" << endl;
        break;
    }

    Mat imgH;
    frame.convertTo(imgH, -1, 2, 0); //increase the contrast (double)

    Mat imgL;
    frame.convertTo(imgL, -1, 0.5, 0); //decrease the contrast (halve)

    //show the image
    imshow("Original Video", frame);
    imshow("Contrast Increased", imgH);
    imshow("Contrast Decreased", imgL);

    if (waitKey(30) == 27) //wait for 'esc' key press for 30 ms. If 'esc' key is pressed, break loop
    {
        cout << "esc key is pressed by user" << endl;
        break;
    }
}
return 0;
}

```

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

You can download this OpenCV visual c++ project from [here](#). (The downloaded file is a compressed .rar folder. So, you have to extract it using Winrar or other suitable software)

All the OpenCV methods in the above example program have been discussed in previous lessons. So, I am not going to repeat them again.

Next Lesson: Histogram Equalization

Previous Lesson: Change Brightness of Image or Video

Posted by Shermal Fernando      Recommend this on Google

Is This Helpful : ☐ Yes (1) ☐ No (0)

4 comments:



Anonymous October 28, 2013 at 3:20 AM

This has been extremely helpful. All other programs that I encountered to change contrast of the image involved accessing each pixel of the image and applying the formula $\text{new_img}(i, j) = \text{img}(i, j) * c$. Even in opencv docs uses the formula mentioned earlier. The formula approach didn't work for me, but this did and I'm delighted. Thanks!

[Reply](#)



stephan verbeeck November 7, 2013 at 3:16 PM

This comment has been removed by the author.

[Reply](#)

▼ [Replies](#)



stephan verbeeck November 7, 2013 at 3:30 PM

(you forgot to adjust the brightness to compensate for the multiplication)

```

#define NA -1 // indicator that value is [N]ot [A]vailable or [N]ot [A]pplicable
#define MIX(v1,v2,ratio) (v1+(v2-v1)*ratio) // Mix two numbers according to a floating point ratio 0..1 (being 0%..100%)
#define MINMAX(min,max,v) ((v)<(min) ? (min) : (v)>(max) ? (max) : (v)) // Limit value inside given valid range

double multiplier = (double)MINMAX(-100,100,contrast)/100.; // convert range -100..100 -> -1..1
double alpha = multiplier>0 ? MIX(1.0, 2.0, multiplier) : MIX(1.0, 0.5, -multiplier); // 0.5 will halve the contrast while 2.0 will double the contrast
double beta = multiplier>0 ? MIX(0.0, -0.5, multiplier) : MIX(0.0, 0.25, -multiplier); // compensate brightness accordingly to get the original average
cv::Mat(src).convertTo(cv::Mat(dst), NA, alpha, beta);

```

[Reply](#)

musafir March 14, 2014 at 11:00 AM



Its simply great tutorial
Can you right a tutorial about depth calculation using single camera. using simple pinhole camera model.
it will be helpful for me.

[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](#).