

First time here? Check out the FAQ!



OpenCV

This forum is disabled, please visit <https://forum.opencv.org>

faq

tags

users

badges

Hi there! Please sign in help



ALL

UNANSWERED

Search or ask your question

ASK YOUR QUESTION

5 How to make auto-adjustments(brightness and contrast) for image Android Opencv Image Correction edit

imagecorrection

brightness

contrast

i'm using OpenCV for **Android**.

I would like to know,how to make **image correction**(auto adjustments of brightness/contrast) for image(bitmap) in android via OpenCV or that can be done by native ColorMatrixFilter from Android!??

I tried to google,but didn't found good tutorials/examples.
So how can i achieve my goal? Any ideas?
Thanks!

asked Nov 6 '15

 VeTaLio
48 ● 2 ● 6 ● 9

updated Nov 30 '15

 sturkmen
6772 ● 3 ● 48 ● 79
<https://github.com/stu...>

Question Tools

Follow

8 followers

[subscribe to rss feed](#)

Comments

- Some easy techniques for auto contrast:
 - [Stretch Contrast](#), with the corresponding code in Gimp: [contrast-stretch.c](#)
 - [Histogram equalization](#)

 Eduardo (Nov 7 '15) edit
[add a comment](#)

Stats

Asked: **Nov 6 '15**Seen: **53,883 times**Last updated: **Nov 17 '15**

Related questions

[How to change contrast/brightness of a CAM using OpenCV 2.4.2](#)
[Simple colour enhancement for images](#)
[android change brightness and contrast of image](#)
[Keeping original image colors](#)
[Changing brightness and contrast cv2.cuda python](#)
[how to estimate the \(weber\) contrast of an image?](#)
[B&W Image brightness- using a Trackbar](#)
[Improve contrast of a 16U image](#)
[contrast-stretch with clipping](#)
[Brightness variation between cameras](#)

3 answers

Sort by » oldest newest most voted

16

Brightness and contrast is linear operator with parameter **alpha** and **beta**

answered Nov 8 '15

 pklab
4284 ● 7 ● 30 ● 60

updated Nov 17 '15

$$O(x,y) = \alpha * I(x,y) + \beta$$

In OpenCV you can do this with [convertTo](#).

The question here is **how to calculate** **alpha** and **beta** automatically ?

Looking at histogram, alpha operates as color range amplifier, beta operates as range shift.

Automatic brightness and contrast optimization calculates alpha and beta so that the output range is 0..255.

```
input_range = max(I) - min(I)
wanted_output_range = 255;
alpha = output_range / input_range = 255 / ( max(I) - min(I) )
```

You can calculate beta so that min(O) = 0;

```
min(O) = alpha * min(I) + beta
beta = -min(I) * alpha
```

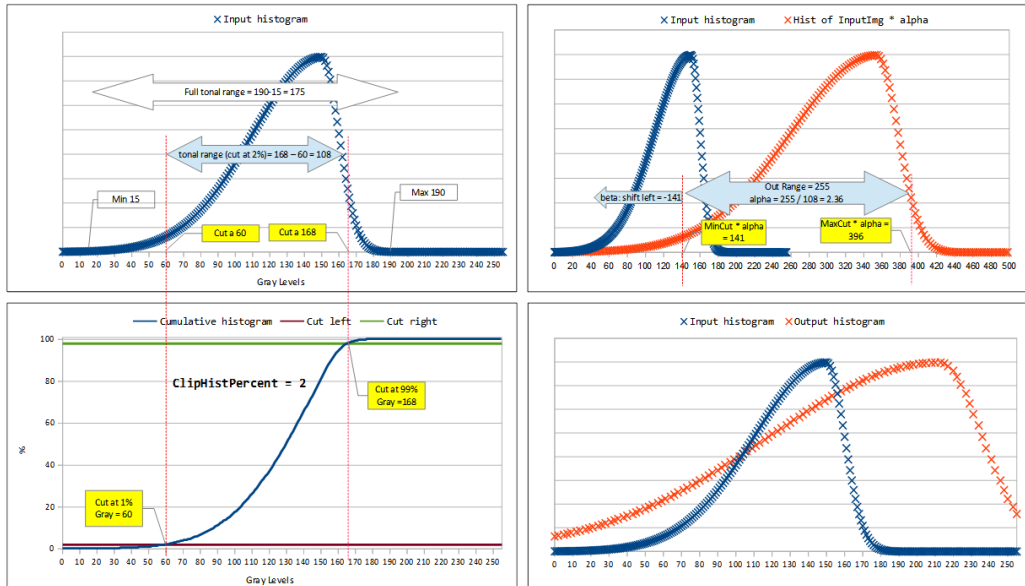
First time here? Check out the FAQ!

To maximize the result of it's useful to cut out some color with few pixels.

This is done cutting left right and wings of histogram where color frequency is less than a value (typically 1%). Calculating cumulative distribution from the histogram you can easily find where to cut.

may be sample chart helps to understand:

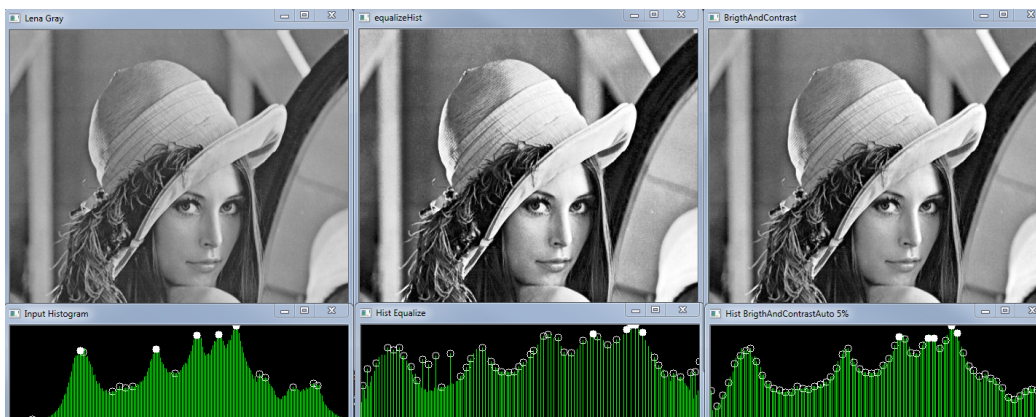
Example - Brightness and Contrast normalization: $\text{OutImg} = \alpha * \text{InputImg} + \beta$



EDIT: Histogram Normalization vs Equalization

By the way `BrightnessAndContrastAuto` could be named `normalizeHist` because it works on BGR and gray images stretching the histogram to the full range without touching bins balance. If input image has range 0..255 `BrightnessAndContrastAuto` will do nothing.

Histogram equalization and CLAE works only on gray images and they change grays level balancing. look at the images below:



EDIT2: Add support for BGRA images too

EDIT3: Correct error in "restore alpha channel from source "

however here is the code:

```
/**
 * \brief Automatic brightness and contrast optimization with optional histogram clipping
 * \param [in]src Input image GRAY or BGR or BGRA
 * \param [out]dst Destination image
 * \param clipHistPercent cut wings of histogram at given percent tipical=>1, 0=>Disabled
 * \note In case of BGRA image, we won't touch the transparency
```

First time here? Check out the FAQ!

```

{
    CV_Assert(clipHistPercent >= 0);
    CV_Assert((src.type() == CV_8UC1) || (src.type() == CV_8UC3) || (src.type() == CV_8UC4));

    int histSize = 256;
    float alpha, beta;
    double minGray = 0, maxGray = 0;

    //to calculate grayscale histogram
    cv::Mat gray;
    if (src.type() == CV_8UC1) gray = src;
    else if (src.type() == CV_8UC3) cvtColor(src, gray, CV_BGR2GRAY);
    else if (src.type() == CV_8UC4) cvtColor(src, gray, CV_BGRA2GRAY);
    if (clipHistPercent == 0)
    {
        // keep full available range
        cv::minMaxLoc(gray, &minGray, &maxGray);
    }
    else
    {
        cv::Mat hist; //the grayscale histogram

        float range[] = { 0, 256 };
        const float* histRange = { range };
        bool uniform = true;
        bool accumulate = false;
        calcHist(&gray, 1, 0, cv::Mat(), hist, 1, &histSize, &histRange, uniform, accumulate);

        // calculate cumulative distribution from the histogram
        std::vector<float> accumulator(histSize);
        accumulator[0] = hist.at<float>(0);
        for (int i = 1; i < histSize; i++)
        {
            accumulator[i] = accumulator[i - 1] + hist.at<float>(i);
        }

        // locate points that cuts at required value
        float max = accumulator.back();
        clipHistPercent *= (max / 100.0); //make percent as absolute
        clipHistPercent /= 2.0; // left and right wings
        // locate left cut
        minGray = 0;
        while (accumulator[minGray] < clipHistPercent)
            minGray++;

        // locate right cut
        maxGray = histSize - 1;
        while (accumulator[maxGray] >= (max - clipHistPercent))
            maxGray--;

        // current range
        float inputRange = maxGray ...
    }
}

```

(more)

Comments


1 Nice answer @pklab. Well done. +1 from me ☺

 theodore (Nov 8 '15) edit

Funny, it seems that the [Gimp white balance](#) command uses the same principle.

I also forgot to mention that the [histogram equalization](#) is a built-in functionality in OpenCV...

There is also the [CLAHE](#) function (Contrast Limited Adaptive Histogram Equalization) that could be used but there is some parameters to tune to use it.

 Eduardo (Nov 9 '15) edit

First time here? Check out the FAQ!

 [VeTaLio](#) (Nov 9 '15) [edit](#)

The code is plain... it's should be easy to translate in Java, may be you could use `ArrayList` instead of `std::vector` [@theodore](#) , [@VeTaLio](#) thanks for appreciation ... if it works you could accept the answer ;=)

 [pklab](#) (Nov 9 '15) [edit](#)


[@pklab](#) yes,that is not a big problem. Ill mark your question as solution,but little bit later,ok:) ? So have some questions about how to reproduce in java,because some stuff doesn't clear for me :(!

 [VeTaLio](#) (Nov 11 '15) [edit](#)


[@VeTaLio](#) you could do it yourself in Java... it's a simple exercise, feel free to ask about *stuffs doesn't clear for you*

 [pklab](#) (Nov 11 '15) [edit](#)

[@pklab](#) as i understand you correct,that wingsCutPercent i need to put manually(as you said,1% => 0.01f value or i need to extract) ? Also,i got a problem,when i'm doing picture,converting via BitmapToMat,my source Mat have type CV_8UC4. So i didn't use your block of code if /else if(where you put CvtColor). I'm doing this directly => CvtColor(source,gray, CV_BGR2GRAY); and the problem occurs here: calcHist - first elemnt,in java there is arraylist of Mat. So via converters i'm trying to convert into ArrayList<mat> and i getting this exception :
java.lang.IllegalArgumentException: CvType.CV_32SC2 != m.type() || m.cols()!=1 I need to convert this for CvType.CV_32SC2?

 [VeTaLio](#) (Nov 13 '15) [edit](#)


[@pklab](#) once i ask this question and didn't get answer => my old post <http://answers.opencv.org/question/74...> is similiar problem with Converters! Thanks!!!

 [VeTaLio](#) (Nov 13 '15) [edit](#)

OK, 2nd edit...check new code, it supports BGRA images too. wingsCutPercent (now clipHistPercent) is a percent than if you want to cut 1% use 1.0. Howhowever it's optional.

 [pklab](#) (Nov 13 '15) [edit](#)

[@pklab](#) so i've translated to java and problem is that i'm getting too much brightness and contrast. I will upload some example to show you.

 [VeTaLio](#) (Nov 16 '15) [edit](#)

[see more comments](#)

0

for python users, here two functions create the same result.

autoAdjustments_with_convertScaleAbs() works very, very fast

answered May 14 '0



zgormez
1 *1

```
def autoAdjustments_with_convertScaleAbs(img):
    alow = img.min()
    ahigh = img.max()
    amax = 255
    amin = 0

    # calculate alpha, beta
    alpha = ((amax - amin) / (ahigh - alow))
    beta = amin - alow * alpha
    # perform the operation g(x,y)= alpha * f(x,y)+ beta
    new_img = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)

    return [new_img, alpha, beta]
```

```
def autoAdjustments(img):
    # create new image with the same size and type as the original image
    new_img = np.zeros(img.shape, img.dtype)

    # calculate stats
    alow = img.min()
    ahigh = img.max()
    amax = 255
    amin = 0

    # access each pixel, and auto adjust
    for x in range(img.shape[1]):
        for y in range(img.shape[0]):
            a = img[x, y]
```

First time here? Check out the FAQ!

```
return new_img
```

add a comment

-1

Does anyone has been able to convert this to C# or VB.NET? I am having a hard time getting this converted.

answered Mar 28 '17

WIVM
1 ● 1 ● 1

add a comment

[about](#) | [faq](#) | [help](#) | [privacy policy](#) | [terms of service](#)
Powered by Askbot version 0.10.2

Copyright OpenCV foundation, 2012-2018. Content on this site is licensed under a [Creative Commons Attribution Share Alike 3.0 license](#).