

[首頁](#)[MySQL教學](#)[網站技巧](#)[網路程式設計](#)[軟體程式設計](#)[資料庫](#)[作業系統](#)[其它](#)[其他](#)[首頁](#)[科技](#)[程式語言](#)

彩色影像的直方圖均衡化--基於OpenCV中EqualizeHist_Demo實現

其他 · 發表 2019-01-11

本文給出基於彩色影像直方圖均衡化的OpenCV程式碼與結果示例！

具體內容包含：

1. 灰度影像直方圖均衡化
2. 對RGB三通道各自均衡化後，再組合輸出結果
3. RGB影像轉化為HSI，YUV，YCbCr顏色空間後，對亮度通道進行均衡化運算後再轉回RGB空間

[cpp] [view plain](#) [copy](#) [print](#) [?](#) [?](#)

```
1. /**
2.  * @function EqualizeHist_Demo.cpp
3.  * @brief Demo code for equalizeHist function
4.  * @author OpenCV team
5.  */
6. #include "opencv2/imgcodecs.hpp"
7. #include "opencv2/highgui/highgui.hpp"
8. #include "opencv2/imgproc/imgproc.hpp"
9. #include <iostream>
10. #include <stdio.h>
11. using namespace cv;
12. using namespace std;
13. // add by frank, 2014-09-25
14. Mat equalizeChannelHist(const Mat & inputImage)
15. {
16.     if( inputImage.channels() >= 3 )
17.     {
18.         vector<Mat> channels;
19.         split(inputImage, channels);
20.         Mat B,G,R;
21.         equalizeHist( channels[0], B );
22.         equalizeHist( channels[1], G );
23.         equalizeHist( channels[2], R );
24.         vector<Mat> combined;
25.         combined.push_back(B);
26.         combined.push_back(G);
27.         combined.push_back(R);
28.         Mat result;
29.         merge(combined, result);
30.         return result;
31.     }
32.     return Mat();
33. }
34. Mat equalizeIntensityHist(const Mat & inputImage)
```

```
35. {
36.   if(inputImage.channels() >= 3)
37.   {
38.     Mat ycrb;
39.     cvtColor(inputImage, ycrb, COLOR_BGR2YCrCb);
40.     vector<Mat> channels;
41.     split(ycrb, channels);
42.     equalizeHist(channels[0], channels[0]);
43.     Mat result;
44.     merge(channels,ycrb);
45.     cvtColor(ycrb, result, COLOR_YCrCb2BGR);
46.     return result;
47.   }
48.   return Mat();
49. }
50. void getGrayImageHistImage(const Mat & src, Mat & histImage)
51. {
52.   Mat hist;
53.   int histSize = 256;
54.   calcHist(&src, 1, 0, Mat(), hist, 1, &histSize, 0);
55.   normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, CV_32F);
56.   histImage = Scalar::all(255);
57.   int binW = cvRound((double)histImage.cols/histSize);
58.   for( int i = 0; i < histSize; i++ )
59.     rectangle( histImage, Point(i*binW, histImage.rows),
60.       Point((i+1)*binW, histImage.rows - cvRound(hist.at<float>(i))),
61.       Scalar::all(0), -1, 8, 0 );
62. }
63. int main( int, char** argv )
64. {
65.   Mat src, dst;
66.   Mat intensity_color_dst;
67.   Mat channel_color_dst;
68.   constchar* source_gray_window = "Source Gray Image";
69.   constchar* equalized_gray_window = "Equalized Gray Image";
```

```
70.  constchar* source_color_window = "Source Color Image";
71.  constchar* equalized_intensity_color_window = "Equalized Intensity Color Image";
72.  constchar* equalized_channels_color_window = "Equalized Channels Color Image";
73.  /// Load image
74.  src = imread( argv[1], 1 );
75.  if( src.empty() )
76.  {
77.      cout<<"Usage: ./Histogram_Demo <path_to_image>"<<endl;
78.      return -1;
79.  }
80.  /// color image intensity equalization
81.  {
82.      intensity_color_dst = equalizeIntensityHist(src);
83.      namedWindow( source_color_window, WINDOW_AUTOSIZE );
84.      namedWindow( equalized_intensity_color_window, WINDOW_AUTOSIZE );
85.      imshow( source_color_window, src );
86.      imshow( equalized_intensity_color_window, intensity_color_dst );
87.  }
88.  /// color image each channel equalization
89.  {
90.      channel_color_dst = equalizeChannelHist(src);
91.      namedWindow( equalized_channels_color_window, WINDOW_AUTOSIZE );
92.      imshow( equalized_channels_color_window, channel_color_dst );
93.  }
94.  /// gray image equalization
95.  {
96.      cvtColor( src, src, COLOR_BGR2GRAY );
97.      equalizeHist( src, dst );
98.      namedWindow( source_gray_window, WINDOW_AUTOSIZE );
99.      namedWindow( equalized_gray_window, WINDOW_AUTOSIZE );
00.      imshow( source_gray_window, src );
01.      imshow( equalized_gray_window, dst );
02.  /// get source gray image Histogram
03.  Mat graySrc_histImage = Mat::ones(200, 260, CV_8U)*255;
04.  getGrayImageHistImage(src, graySrc_histImage);
```

```
05.    imshow("source gray image histogram", graySrc_histImage);
06.    /// get equalized gray image Histogram
07.    Mat grayDst_histImage = Mat::ones(200, 260, CV_8U)*255;
08.    getGrayImageHistImage(dst, grayDst_histImage);
09.    imshow("Equalized gray image histogram", grayDst_histImage);
10. }
11. /// Wait until user exits the program
12. waitKey(0);
13. return 0;
14. }
```

```
/**
 * @function EqualizeHist_Demo.cpp
 * @brief Demo code for equalizeHist function
 * @author OpenCV team
 */

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

using namespace cv;
using namespace std;

// add by frank, 2014-09-25
Mat equalizeChannelHist(const Mat & inputImage)
{
    if( inputImage.channels() >= 3 )
    {
        vector<Mat> channels;
        split(inputImage, channels);

        Mat B,G,R;
```

```
        equalizeHist( channels[0], B );
        equalizeHist( channels[1], G );
        equalizeHist( channels[2], R );

        vector<Mat> combined;
        combined.push_back(B);
        combined.push_back(G);
        combined.push_back(R);

        Mat result;
        merge(combined, result);

        return result;
    }

    return Mat();
}

Mat equalizeIntensityHist(const Mat & inputImage)
{
    if(inputImage.channels() >= 3)
    {
        Mat ycrCb;

        cvtColor(inputImage, ycrCb, COLOR_BGR2YCrCb);

        vector<Mat> channels;
        split(ycrCb, channels);

        equalizeHist(channels[0], channels[0]);

        Mat result;
        merge(channels, ycrCb);
```

```
        cvtColor(ycrCb, result, COLOR_YCrCb2BGR);

        return result;
    }

    return Mat();
}

void getGrayImageHistImage(const Mat & src, Mat & histImage)
{
    Mat hist;
    int histSize = 256;

    calcHist(&src, 1, 0, Mat(), hist, 1, &histSize, 0);
    normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, CV_32F);

    histImage = Scalar::all(255);
    int binW = cvRound((double)histImage.cols/histSize);

    for( int i = 0; i < histSize; i++ )
        rectangle( histImage, Point(i*binW, histImage.rows),
            Point((i+1)*binW, histImage.rows - cvRound(hist.at<float>(i))),
            Scalar::all(0), -1, 8, 0 );
}

int main( int, char** argv )
{
    Mat src, dst;
    Mat intensity_color_dst;
    Mat channel_color_dst;

    const char* source_gray_window = "Source Gray Image";
    const char* equalized_gray_window = "Equalized Gray Image";
    const char* source_color_window = "Source Color Image";
```

```
const char* equalized_intensity_color_window = "Equalized Intensity Color In
const char* equalized_channels_color_window = "Equalized Channels Color Imag

/// Load image
src = imread( argv[1], 1 );

if( src.empty() )
{
    cout<<"Usage: ./Histogram_Demo <path_to_image>"<<endl;
    return -1;
}

/// color image intensity equalization
{
    intensity_color_dst = equalizeIntensityHist(src);

    namedWindow( source_color_window, WINDOW_AUTOSIZE );
    namedWindow( equalized_intensity_color_window, WINDOW_AUTOSIZE );

    imshow( source_color_window, src );
    imshow( equalized_intensity_color_window, intensity_color_dst );
}

/// color image each channel equalization
{
    channel_color_dst = equalizeChannelHist(src);
    namedWindow( equalized_channels_color_window, WINDOW_AUTOSIZE );
    imshow( equalized_channels_color_window, channel_color_dst );
}

/// gray image equalization
{
    cvtColor( src, src, COLOR_BGR2GRAY );
    equalizeHist( src, dst );
}
```



```
namedWindow( source_gray_window, WINDOW_AUTOSIZE );
namedWindow( equalized_gray_window, WINDOW_AUTOSIZE );

imshow( source_gray_window, src );
imshow( equalized_gray_window, dst );

/// get source gray image Histogram
Mat graySrc_histImage = Mat::ones(200, 260, CV_8U)*255;
getGrayImageHistImage(src, graySrc_histImage);
imshow("source gray image histogram", graySrc_histImage);

/// get equalized gray image Histogram
Mat grayDst_histImage = Mat::ones(200, 260, CV_8U)*255;
getGrayImageHistImage(dst, grayDst_histImage);
imshow("Equalized gray image histogram", grayDst_histImage);
}

/// Wait until user exits the program
waitKey(0);

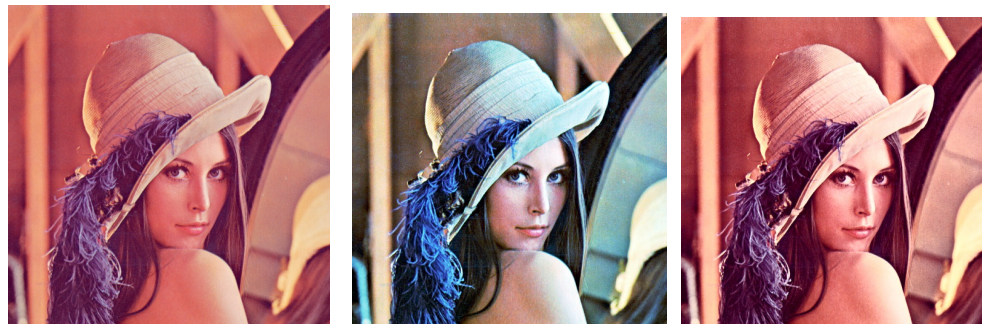
return 0;
}
```

執行結果，如下圖所示：



原灰度影像

直方圖均衡化增強後圖像



原彩色影像

RGB各通道直方圖均衡化後圖像

YCbCr 亮度通道Y直方圖均

衡化後圖像

分RGB通道均衡化後圖像顏色有失真情況，而亮度通道均衡化結果不會，主要原因：

Histogram equalization is a non-linear process. Channel splitting and equalizing each channel separately is not the proper way for equalization of contrast. Equalization involves Intensity values of the image not the color components. So for a simple RGB color image, HE should not be applied individually on each channel. Rather, it should be applied such that intensity values are equalized without disturbing the color balance of the image. So, the first step is to convert the color space of the image from RGB into one of the color space which separates intensity values from color components. Some of these are:

HSV/HLS

YUV

YCbCr

Convert the image from RGB to one of the above mentioned color spaces. YCbCr is preferred as it is designed for digital images. Perform HE of the intensity plane Y. Convert the image back to RGB.

標籤：

👍 您可能也會喜歡...

彩色影像的直方圖均衡化--基於OpenCV中
EqualizeHist_Demo實現

灰度圖的直方圖均衡化的原理及實現

OpenCV—python 色階調整 (直方圖均衡化)

使用C++實現彩色影像直方圖均衡化的三種方法

[影像去霧] (一) 限制對比度的自適應直方圖均衡化

(CLAHE)的學習體會——影像分塊

影像處理基礎02直方圖均衡化的推導和程式設計實現

【數字影像處理】灰度直方圖、直方圖均衡化、直方圖規
定化

0016-在OpenCV環境下進行影像的直方圖均衡化

灰度圖的直方圖均衡化(Histogram Equalization)原理與

Python 實現

MATLAB的直方圖均衡化

我理解的直方圖均衡化

C# EMGU 3.4.1學習筆記 (十二) 示例程式：直方圖均衡
化 (彩色影像)

直方圖均衡化的作用是影像增強原理

OpenCV數字影像處理 (一) ——直方圖均衡化的實現

MATLAB：虹膜識別的影像灰度化處理·直方圖均衡化

首頁

Python教學

