

PRAKTIKUM 5

Histogram

Materi:

- Histogram warna
- Histogram equalization
- Contrast stretching

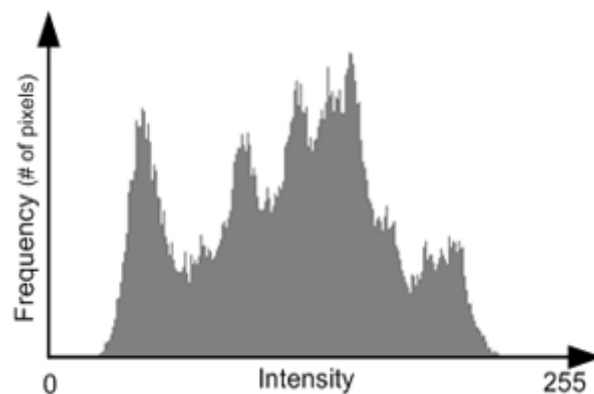
Tujuan Praktikum:

- Mahasiswa dapat menampilkan histogram untuk satu channel grayscale dan setiap channel warna
- Mahasiswa dapat melakukan contrast stretching
- Mahasiswa dapat melakukan enhancement menggunakan histogram equalization
- Mahasiswa dapat menganalisis citra menggunakan histogram

A. PENYAJIAN

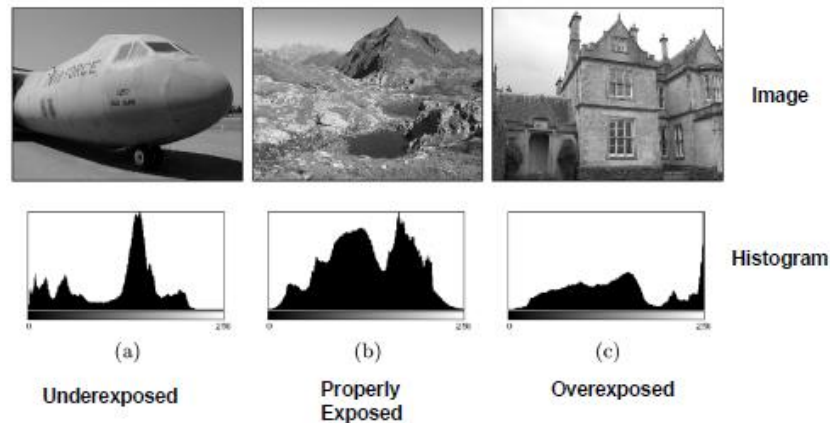
Histogram

Histogram merupakan representasi grafis dari distribusi data. Pada pengolahan citra, data yang ditampilkan pada histogram adalah persebaran frekuensi pixel yang memiliki nilai intensitas yang sama. Nilai intensitas pada histogram biasanya disebut “bin”. Untuk citra grayscale, ukuran bin didalam histogram adalah 256 karena rentang intensitas berkisar 0 – 255.



Gambar 1 Histogram citra grayscale 8-bit

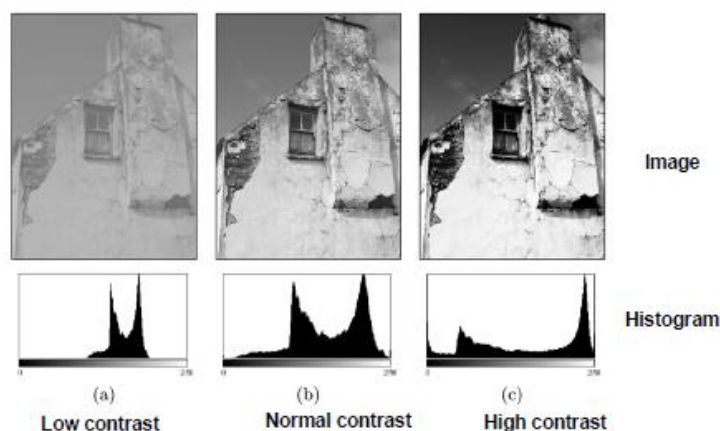
Histogram dapat digunakan untuk melakukan image enhancement atau menentukan nilai threshold. Histogram berguna untuk menganalisis **exposure**, **brightness** dan **kontras**, dan **dynamic range**. Exposure menunjukkan penyebaran nilai intensitas pixel didalam suatu citra. Suatu citra dapat dikatakan ter-expose dengan baik ketika seluruh bin dari ujung kanan ke kiri terisi tanpa ada space yang kosong. Sedangkan untuk brightness, kita dapat menggeser histogram untuk meningkatkan atau menurunkan brightnessnya.



Gambar 2 Exposure pada citra

Contrast Stretching / Histogram Stretching

Tujuan utama dari contrast stretching adalah untuk meningkatkan selisih nilai intensitas maksimal dan minimal pada sebuah citra. Kontras rendah menggunakan sedikit bin, sedangkan kontras tinggi menggunakan banyak bin. Menambah atau mengurangi kontras dapat dilakukan dengan cara meregangkan atau memampatkan histogram.



Gambar 3 Kontras pada citra

Formula untuk histogram stretching yaitu:

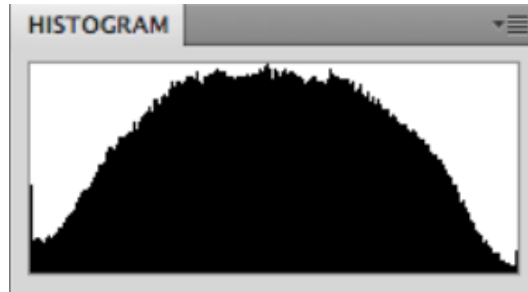
$$g(x, y) = \frac{f(x, y) - f_{min}}{f_{max} - f_{min}} * 255$$

Ket:

$f(x, y)$: nilai pixel di titik (x, y)

f_{min} : nilai pixel terkecil pada citra

f_{max} : nilai pixel terbesar pada citra



Gambar 4 Histogram ideal

Histogram Equalization

Histogram Equalization adalah suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Sifat-sifat dari histogram equalization antara lain:

- Gray level yang sering muncul lebih dijarangkan jaraknya dengan gray level sebelumnya.
- Gray level yang jarang muncul bisa lebih dirapatkan jaraknya dengan gray level sebelumnya.
- Histogram baru pasti mencapai nilai maksimal keabuan (contoh: 255).

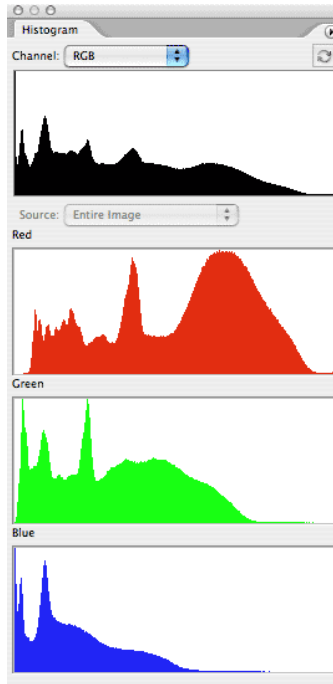
Untuk dapat melakukan histogram equalization ini diperlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari histogram.

Pada OpenCV proses histogram equalization dapat dilakukan dengan fungsi:

```
void equalizeHist(
    Mat src,          /* source image, 8 bit single-channel */
    Mat dst /* destination image, 8 bit single-channel */
);
```

Histogram Warna

Komposisi warna pada citra dapat direpresentasikan dalam bentuk histogram. Histogram warna merepresentasikan distribusi jumlah piksel untuk tiap intensitas warna dalam citra. Untuk mendefinisikan histogram, warna di kuantisasi menjadi beberapa level diskrit, kemudian untuk tiap level tersebut dihitung jumlah piksel yang nilainya sesuai.



Gambar 5 Histogram warna pada channel RGB

B. LATIHAN

Membuat program yang dapat melakukan histogram equalization menggunakan fungsi OpenCV.

equalized.cpp

```
#include <cv.h>
#include <highgui.h>
using namespace cv;

int main()
{
    //membaca citra grayscale
    Mat original = imread("cameraman.jpg",0);
    //deklarasi variable untuk menyimpan citra hasil ekualisasi
    Mat equalized;
    //melakukan fungsi histogram equalization
    equalizeHist(img, equalized);
    //menampilkan citra
    imshow("original", original);
    imshow("equalized", equalized);
    waitKey(0);
    return 0;
}
```

Output equalized.cpp



Membuat program yang dapat menampilkan histogram warna pada citra lily.jpg

HistogramWarna.cpp

```
#include <cv.h>
#include <highgui.h>
#define BIN 255

using namespace cv;

//fungsi untuk menampilkan histogram
void drawHistogram(float data[], string title, int xScale, int yScale)
{
    //find max value
    float yMax = maxVal(data, BIN);
    //set margin
    float yMargin = 1;
    //set y size
    int ySize = 128*yScale;
    //set x size
    int xSize = 256*xScale;
    //create Mat to draw histogram
    Mat histogram = Mat(ySize,xSize,CV_8UC1,double(0));
    //iterating
    for(int x = 0; x < histogram.cols; x++)
    {
        //histogram view scaling
        int histValue = ySize - ceil(data[x/xScale]*(ySize-yMargin)/yMax);
        for(int y = histogram.rows-1; y >= histValue; y--)
        {
            histogram.at<uchar>(y,x) = 255;
        }
    }
    imshow(title+" histogram", histogram);
}

int main()
{
    Mat img = imread("lily.jpg");

    float blue[BIN] = {0.0};
    float green[BIN] = {0.0};
```

```
float red[BIN] = {0.0};

for(int y = 0; y < img.rows; y++)
{
    Vec3b *pixel = img.ptr<Vec3b>(y);
    for(int x = 0; x < img.cols; x++)
    {
        blue[pixel[x][0]]++;
        green[pixel[x][1]]++;
        red[pixel[x][2]]++;
    }
}

drawHistogram(blue, "blue channel ", 1, 1);
drawHistogram(green, "green channel ", 1, 1);
drawHistogram(red, "red channel ", 1, 1);

imshow("original", img);
waitKey(0);
return 0;
}
```

Output HistogramWarna.cpp



Nama :

NRP :

Nama Dosen :

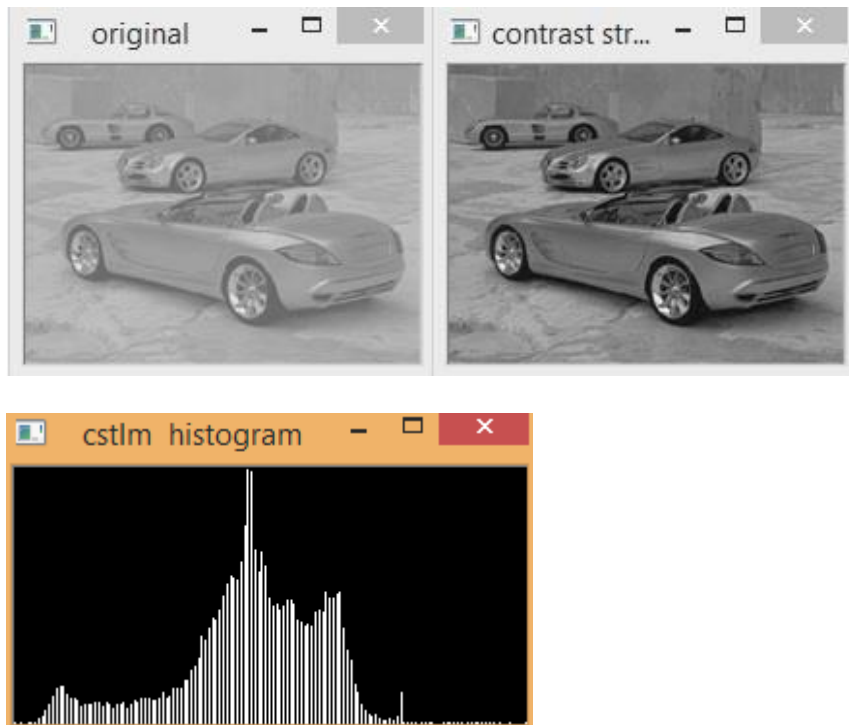
Nama Asisten :

C. Lembar Kerja Praktikum

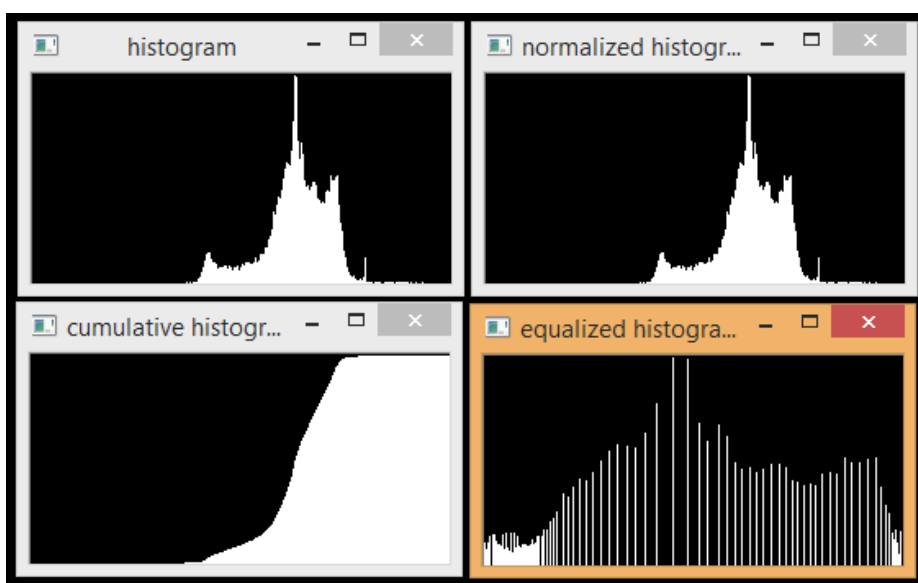
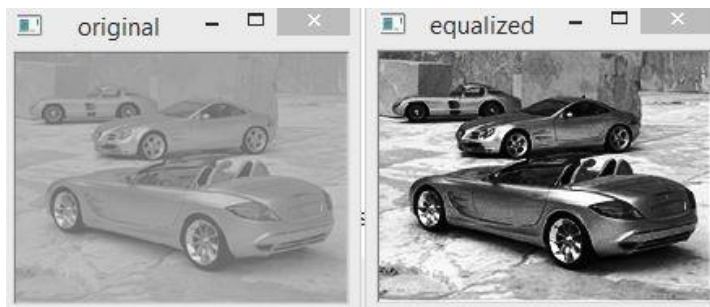
Simpan kode program beserta screenshot citra original, citra hasil ekualisasi, citra hasil contrast stretching, histogram citra original, histogram hasil contrast stretching, normalized histogram, cumulative histogram, dan equalized histogram. File disimpan dengan format LKP5_NIM_Kelas dalam file .pdf.

1. Download citra car.png pada LMS
2. Baca citra car.png dalam format *grayscale*
3. Buatlah sebuah fungsi `contrastStretching` tanpa menggunakan fungsi OpenCV pada car.png (Fungsi `opencv` yang diperbolehkan: `imread`, `waitKey`, `imshow`, dan membaca citra *grayscale*).
4. Buatlah sebuah fungsi `histogramEqualization` tanpa menggunakan fungsi `openCV` dan lakukan proses histogram equalization pada car.png (Fungsi `opencv` yang diperbolehkan: `imread`, `waitKey`, `imshow`, dan membaca citra *grayscale*).
5. Algoritme `histogramEqualization`:
 - Hitung jumlah kemunculan piksel setiap nilai derajat keabuan
 - Hitunglah peluang nilai kemunculan setiap nilai derajat keabuan (normalized histogram)
 - Hitung histogram kumulatif
 - Lakukan histogram equalization
6. Tampilkan histogram hasil contrast stretching, histogram citra original, normalized histogram, cumulative histogram, dan equalized histogram
7. Berikan penjelasan singkat terhadap citra hasil histogram equalization dan bandingkan terhadap hasil contrast stretching.

Hasil yang diharapkan**Contrast stretching**



Histogram equalization



Contoh program


```

#include <cv.h>
#include <highgui.h>
#define BIN 256

using namespace cv;

//fungsi mencari nilai maksimum pada array
float maxVal(float data[], int n)
{
    float outp = 0;
    for(int i = 0; i < n; i++)
    {
        if(data[i] > outp) outp = data[i];
    }
    return outp;
}

//fungsi untuk menampilkan histogram
void drawHistogram(float data[], string title, int xScale, int yScale)
{
    //find max value
    float yMax = maxVal(data, BIN);
    //set margin
    float yMargin = 1;
    //set y size
    int ySize = 128*yScale;
    //set x size
    int xSize = 256*xScale;
    //create Mat to draw histogram
    Mat histogram = Mat(ySize,xSize,CV_8UC1,double(0));
    //iterating
    for(int x = 0; x < histogram.cols; x++)
    {
        //histogram view scaling
        int histValue = ySize - ceil(data[x/xScale]*(ySize-yMargin)/yMax);
        for(int y = histogram.rows-1; y >= histValue; y--)
        {
            histogram.at<uchar>(y,x) = 255;
        }
    }
    imshow(title+" histogram", histogram);
}

//fungsi untuk ekualisasi histogram
void histogramEqualization(Mat image, Mat dst)
{
    //deklarasi array 1 dimensi dengan ukuran BIN
    float hist[BIN] = {0.0};
    //menghitung histogram
    ...
    ...

```

```

...
//draw histogram
drawHistogram(hist, "", 1, 1);

//menghitung normalized histogram
...
...
...
//draw normalized histogram
drawHistogram(hist, "normalized", 1, 1);

//menghitung cumulative histogram
...
...
...
//draw cumulative histogram
drawHistogram(hist, "cumulative", 1, 1);

//menghitung equalized histogram
...
...
...
//draw equalized histogram
drawHistogram(hist, "equalized", 1, 1);

//terapkan pada citra Mat dst
...
...
...
}

Void contrastStretching(Mat img, Mat dst)
{
    ...
    ...
    ...
}

int main()
{
    Mat original;
    // ... baca citra car.png dalam format grayscale
    // ... buat Mat equalized dengan ukuran sama dengan original, tipe data
        CV_8UC1

    ...
    ...
    return 0;
}

```