ID student: 0780828

Name: Alisher Mukashev

Class: Digital Image Processing

Introduction

This overview aims to introduce the report for the project 2 in the course 5051 Digital Image Processing. This report covers the solution for three assigned tasks. These tasks are described in the document ***reportProject2.pdf*** published on the website e3new for the course. This report has five separate sections which were written as standalone. Each section describes its assigned task and presents a theoretical framework for its solution.

Section 1.a

The task: For the RGB color image, *dew on roses (color).tif*. Determine and plot the R, G and B-component images.

Figure 1-1.

Background

This section refers to a Color Model problem. For such purpose given image *dew on roses (color).tif* (Figure 1-1).

Here we consider RGB color model. In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Figure 1-2, in which RGB primary values are at three corners; the secondary colors cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale (points of equal RGB values) extends from black to white along the line joining these two points.

The different colors in this model are points on or inside the cube, and are defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so that the cube shown in Figure 1-2 is the unit cube. That is, all values of R, G, and B are assumed to be in the range [0,1].

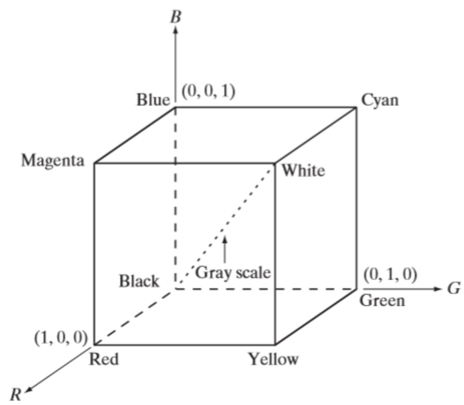

Figure 1-2.

Let's consider RGB image. It's actually a matrix (MxNx3), where M,N-shape of input image:

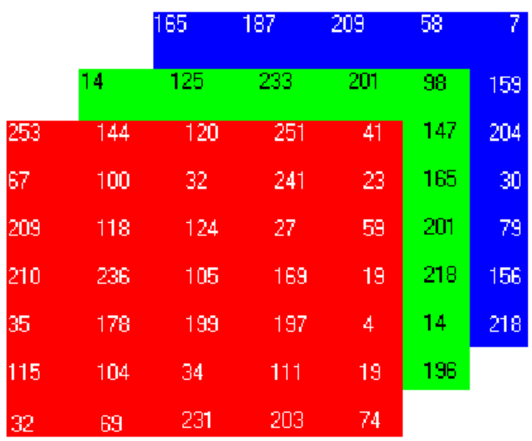

Figure 1-3

We can easily get R or G or B component by splitting the input image into three channels. For example there is a matrix from Figure 1-3 = {X11r (253), X11g (14), X11b (165); X21r (144), X21g (125), X21b (187); X31r (120), X31g (233), X31b (209); …}. In this case R-component will be {X11r (253), X21r (144), X31r (120), …}, G-component {X11g (14), X21g (125), X31g (233), …} and B-component {X11b (165), X21b (187), X31b (209), …}

Algorithm, Flow-chart

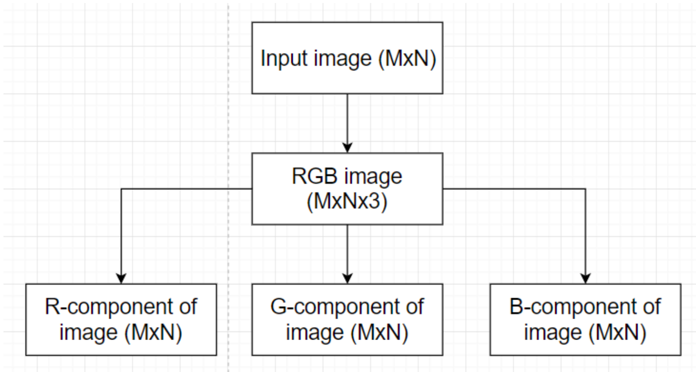

Figure 1-5.

Results

R-component

Figure 1-6.

G-component

Figure 1-7.

B-component

Figure 1-8.

Discussions

In this section we studied RGB color model. As we can see we consider image as RGB-image and split into three different components.

Section 1.b

The task: Obtain and plot the color-transformed image by modifying the B-component image using histogram equalization scheme.

Background

This section refers to Color Transformations problem. For such purpose given image *dew on roses (color).tif* (Figure 1-1).

Consider an image whose pixel values are confined to some specific range of values only. For eg, brighter image will have all pixels confined to high values. But a good image will have pixels from all regions of the image. So you need to stretch this histogram to either ends (as given in below image, from wikipedia) and that is what Histogram Equalization does (in simple words). This normally improves the contrast of the image.

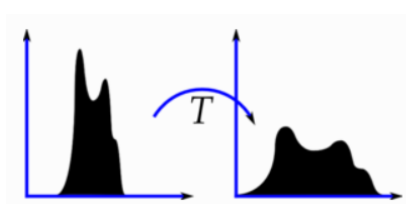

Figure 1-9.

We need a transformation function which maps the input pixels in one region to output pixels in full region. That is what histogram equalization does.

In this task we work with B-component. It means that firstly, we need to do determine B-component from input image. We did it in the previous task. Then we compute histogram and cumulative distribution function of this histogram from obtained B-component image. After that we can create a transformation function to produce a new image with a flat histogram. In the end we merge R, G and B component to get final image.

Algorithm, Flow chart

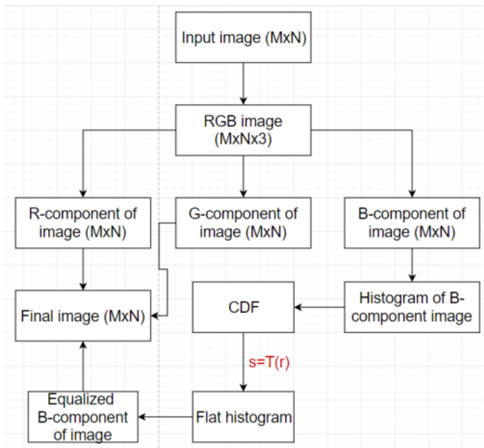

Figure 1-10

Results

Figure 1-11

Discussion

After applying the equalization, we get the image which has certainly more contrast. Thus, we can conclude that we can use an image's histogram for contrast adjustment.

Section 2.a

The task: Consider the image *airplane in the sky.tif*. Preform the image segmentation by edge-pixel detection + edge linking. **Edge-pixel detection**: apply the Marr-Hildreth edge detection algorithm to obtain the edge image. Plot all the images generated during the entire step-by-step procedure of applying the algorithm. Assume two thresholds: 0% and 4% the maximum gray level of the LoG image (refer to Example 10.7).

Figure 2-1

Background

This section refers to Edge Detection problem. For such purpose given image *airplane in the sky.tif* (Figure 2-1).

The edge-detection methods are based simply on filtering an image with one or more masks, with no provisions being made for edge characteristics and noise content. In this section, we discuss Marr-Hildreth technique that make an attempt to improve on simple edge-detection methods by taking into account factors such as image noise and the nature of edges themselves.

The Marr-Hildreth edge-detection algorithm may be summarized as follows:
1. Filter the input image, *f(x, y),* with an Gaussian lowpass filter obtained by sampling Eq. (2-1).

2. Compute the Laplacian of the image resulting from Step 1 using, for example, the mask in Figure 2-2. [Steps 1 and 2 implement Eq. (2-2).], Figure 2-4.

3. Find the zero crossings of the image from Step 2, Figure 2-5, 2-6.

$$G(x, y) = e^{-\frac{x^2+y^2}{2*\sigma^2}} \qquad (2\text{-}1)$$

1	1	1
1	−8	1
1	1	1

Figure 2-2

$$g(x,y) = [\nabla^2 G(x,y)] * f(x,y) = \nabla^2[G(x,y) * f(x,y)] \quad (2\text{-}2)$$

Zero crossing: 3x3 mask, at least 2 opposing pixels of center pixel p:

1. Must have opposite signs.

2. Absolute difference between pixels with opposite sign >= **threshold**.

Algorithm, Flow chart

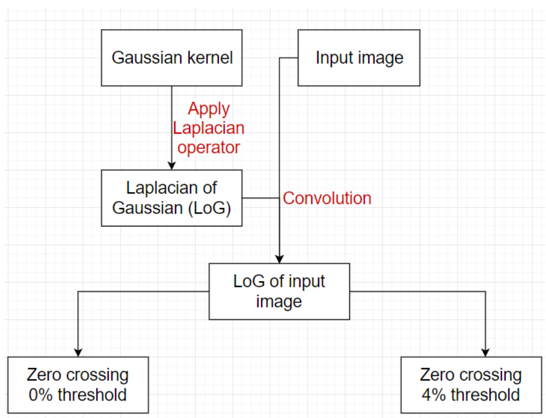

Figure 2-3

Results

Figure 2-4

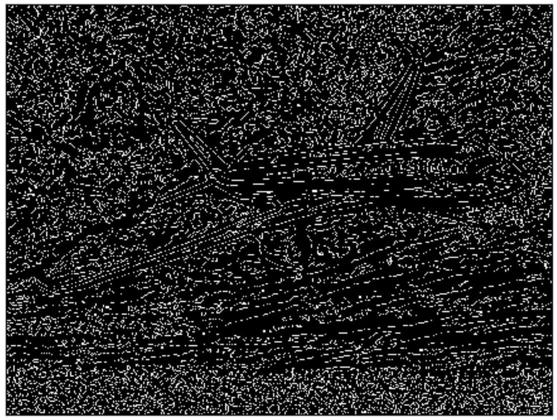

Figure 2-5

Figure 2-6

Discussions

In this section we applied the Marr0Hildreth edge detection algorithm to obtain the edge image. We can see that after zero crossing with threshold of zero we got 'spaghetti effect'. This is a serious disadvantage of this method. But we can avoid this problem by using threshold of 4%. This threshold filtered out all noises around the aircraft body. Another important consequence of using zero crossings for edge detection is that the resulting edges are 1 pixel thick. This property simplifies subsequent stages of processing, such as edge linking, which we discuss further.

Section 2.b

The task: Consider the image *airplane in the sky.tif*. Preform the image segmentation by edge-pixel detection + edge linking. **Edge linking by Hough transform**: based on the edge maps obtained in a), use the Hough transform to perform *edge linking*.

• Firstly, determine the Hough parameter space using 2º increments for θ and 5 pixels increments for ρ. Make the same plot as Figure 10.31(c) (Example 10.12).

• From your results of Hough parameter space, determine the cells containing the largest 20 counts to make the same plots as Figures 10.31(d)-10.31(e).

• Determine the possible cells for the aircraft body.

Background

This section refers to Edge Linking problem. For such purpose given image *airplane in the sky.tif* (Figure 2-1).

Steps to ***determine the Hough parameter space***:

1. Compute the gradient of image. We already did it in the previous task (Edge-pixel detection). We will use the image which we obtained by using Zero crossing with a threshold equal to 4% of the maximum value of the image (Figure 2-6).

2. Identify edge pixels and their coordinates (x_i, y_i).

3. Construct a parameter plane ($\rho\theta$-plane) with specifying divisions (2º increments for θ and 5 pixels increments for ρ). The process of creating a parameter plane is shown on Figure 2-7.

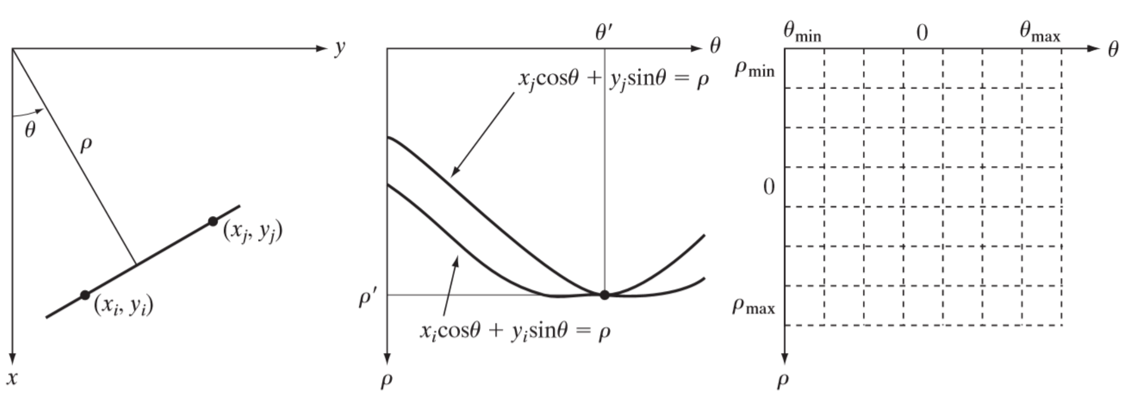

Figure 2-7.

Algorithm, Flow chart

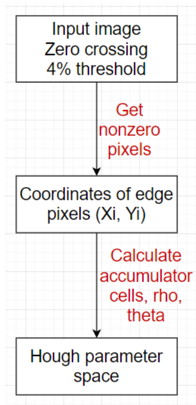

Figure 2-8

Results

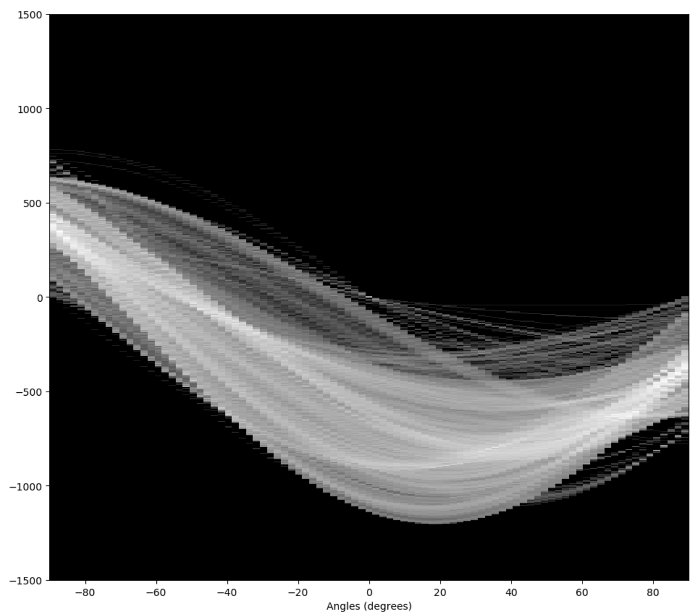

Figure 2-9

Steps to ***determine the cells containing the largest 20 counts***:

1. Consider the Hough parameter space which we got in the previous task (Figure 2-9).

2. Find 20 highest count in the Hough parameter space. In that cells we can find ρ and θ.

3. Draw the lines, corresponding to this points, Figure 2-11

Algorithm, Flow chart

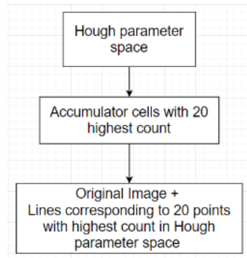

Figure 2-10

Results

Figure 2-11

Steps to ***determine the possible cells for the aircraft body***:

1. Consider the lines which we got in the previous task (Figure 2-11).

2. Using coordinates of these lines we can determine possible cells for aircraft body, Figure 2-22.

Results

Figure 2-22

Discussions

In this section we applied the Hough transform to perform edge linking. The result is that we can define the aircraft body.

Section 3

The task: For the grayscale image, *dew on roses.tif*. Use the multiple global thresholding based on Otsu's algorithm partition the image into three segments.

Figure 3-1

Background

This section refers to Image Thresholding problem. For such purpose given image *dew on roses.tif* (Figure 3-1).

The approach discussed in this section, called Otsu's method (Otsu [1979]), is an attractive alternative. The method is optimum in the sense that it maximizes the between-class variance, a well-known measure used in statistical discriminant analysis. The basic idea is that well-thresholded classes should be distinct with respect to the intensity values of their pixels and, conversely, that a threshold giving the best separation between classes in terms of their intensity values would be the best (optimum) threshold.

The Otsu's algorithm for multiple thresholds may be summarized as follows:

1. Compute the histogram of input image $p_i = \frac{n_i}{MN}$, Figure 3-3.

2. Find cumulative sum $P_k = \sum_{i \in C_k} p_i$, where C_k is the k-class.

3. Find cumulative means $m_k = \frac{1}{P_k} * \sum_{i \in C_k} i * p_i$.

4. Find global mean $m_G = \sum_{i=0}^{L-1} i * p_i$, where L – intensity levels.

5. Compute the between-class variance $\sigma_B^2 = \sum_{k=1}^{K} P_k * (m_k - m_G)^2$.

6. Find k_1 and k_2, which maximize the between-class variance (find optimum thresholds), Eq. (3-1).

7. Get the thresholded image, *g(x, y)*, Eq. (3-2), Figure 3-4.

$$\sigma_B^2(k_1^*,\ k_2^*) = \max_{0<k_1<k_2<L-1} \sigma_B^2(k_1, k_2) \tag{3-1}$$

$$g(x,y) = \begin{cases} a,\ if\ f(x,y) \le k_1^* \\ b, if\ k_1^* < (x,y) \le k_2^* \\ c, if\ f(x,y) > k_2^* \end{cases} \tag{3-2}$$

Where *a*, *b* and *c* are three distinct intensity values (*a*=0, *b*= 127, *c*=255).

In the table you can see obtained results:

Parameter\Figure	Intensity value
k_1	110
k_2	160

Algorithm, Flow chart

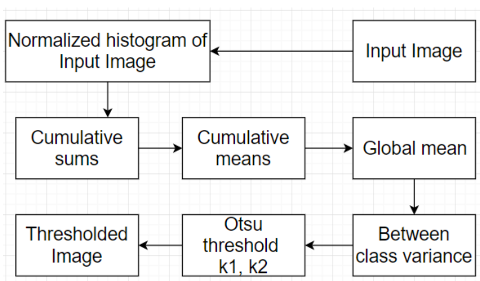

Figure 3-2

Results

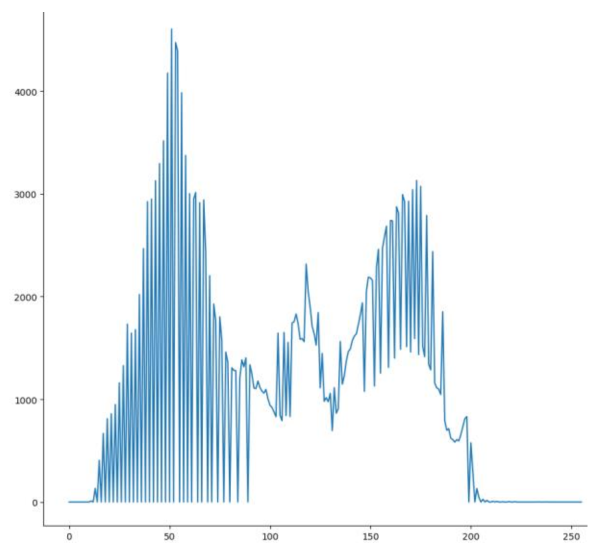

Figure 3-3

Figure 3-4

Discussions

The objective of task in this section is to segment the image into three regions: the dark background, the illuminated area of the rose, and the area in shadows. The principal reason this example worked out so well can be traced to the histogram in the Figure 3-3 having three distinct modes.