Homework 7

Image Segmentation and Edge Detection

2020311381 Sanghyeok Choi

**Goal of Project** : Understanding on Preprocessing for Computer Vision (Object Boundary Detection)

**Tools :** Watershed Algorithm, Canny Edge Detection.

1. Watershed Algorithm :

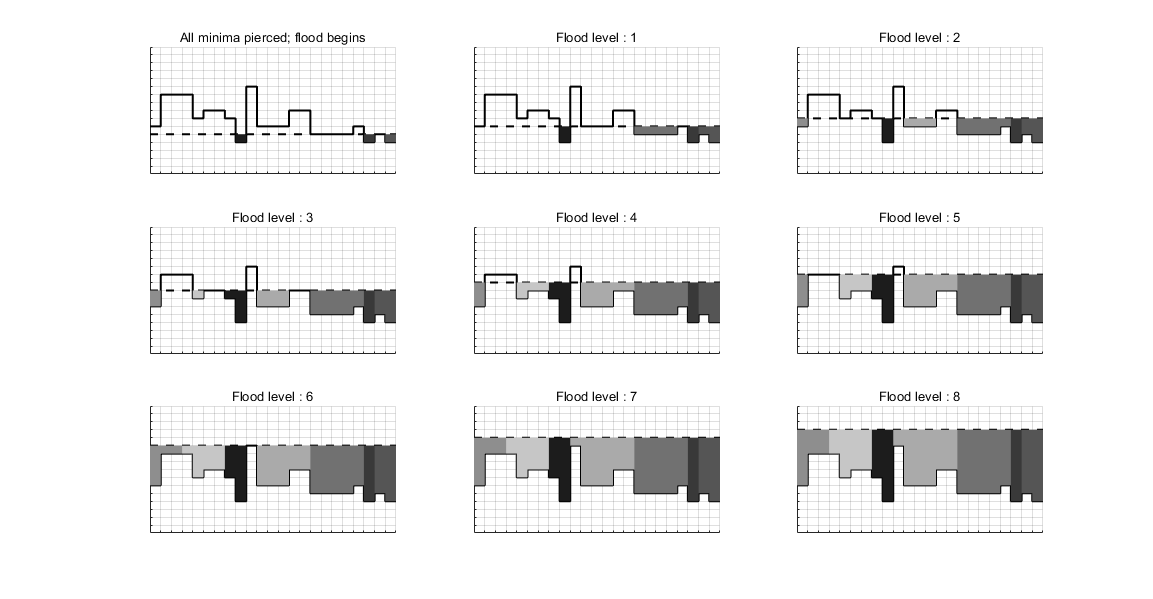
It called watershed because algorithm seems like flood.

First find each minimum points. And fill by other color for distinguish them. Next process, find each points that have slightly larger than previous one. Check neighbor value; if all neighbor values are zero, fill another color for distinguish, Unless all neighbor values are zero, fill non-zero neighbor values. Iterate these steps.

In my case, I match color by some numbers. It starts in ‘1’ and add 1 whenever color need more. If there are no color, it is 0. When some points are select by each condition of floods, get maximum value of neighbor points. If there is non-zero, it will take that value and if not, it will take 0. Next if it still 0, match color. Let’s see one example case of 1-dimension.

You are given the following sequence numbers;

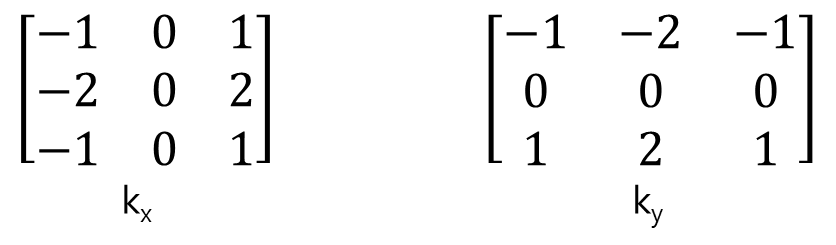
f = [6*,* 10*,* 10*,* 10*,* 7*,* 8*,* 8*,* 7*,* 4*,* 11*,* 6*,* 6*,* 6*,* 8*,* 8*,* 5*,* 5*,* 5*,* 5*,* 6*,* 4*,* 5*,* 4].



< Fig 1 >

As you see < Fig 1 >, Each region are distinguish by different color.

2. Canny Edge Detection:



< Fig 2 >

< Fig 2 > is filter for get gradient image. Apply them and get Gx, Gy.

G = , . For each pixel of G, check whether G is local maximum of direction. (I,j)th theta = then

If (I,j)th G is larger than direction G, (I,j)th output is correspond G value and if not, (I,j)th of output is 0. Now apply hysteresis edge linking. It use two threshold, if gradient is greater than maxval, it is considered as edge, and if it is less than minival, it is not considered. Between two threshold, if at least one pixel of edge string which it is included is greater than maxval, it can be considered as edge.

**Process :**



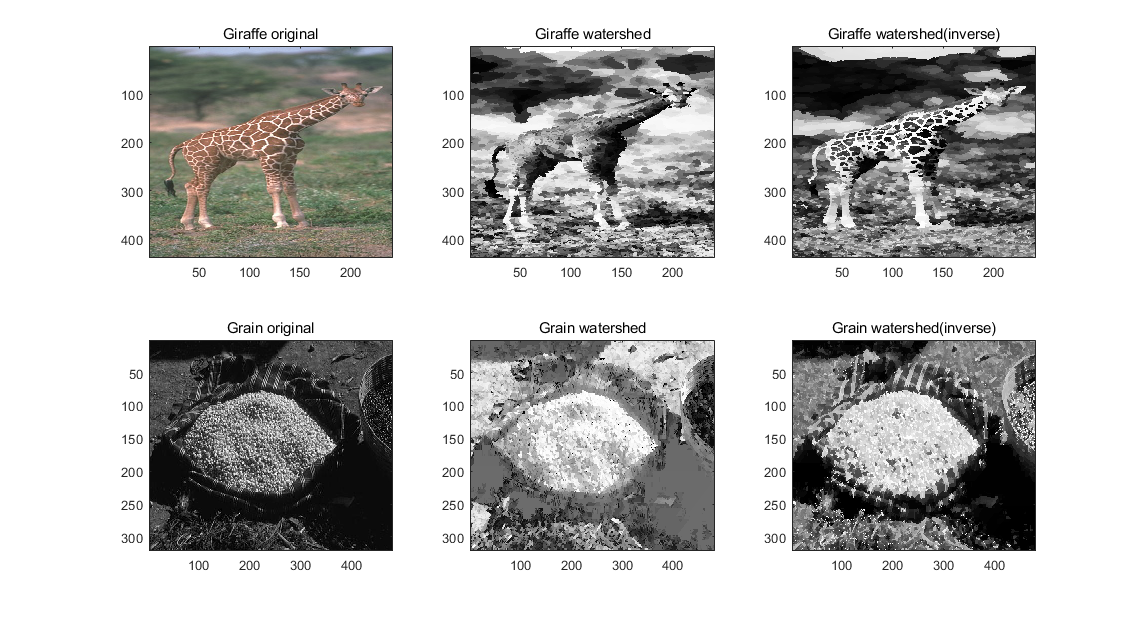
< Fig 3 >

There are two pictures that 8bits per pixels. Now I’ll apply two function; image segmenation & edge detection. These methods are extracting information from an image much less information than the original one, but it contains much more relevant to the other modules of an automatic vision system than the discarded information. There are a lots of correspond methods, but in this case I use “Water shed” for segmentation and “Canny edge detection” for edge detection. I wonder the it does not need for threshold for segmentation. So I use water shed for segmentation. In edge detection, I already know canny detection when I solve boundary problem for other study. But I don’t know how it works. So I use it to understand by this assignment.

I apply this method by “Matlab” tool.

**Result :**

Prob 1. Image Segmentation : Watershed Algorithm

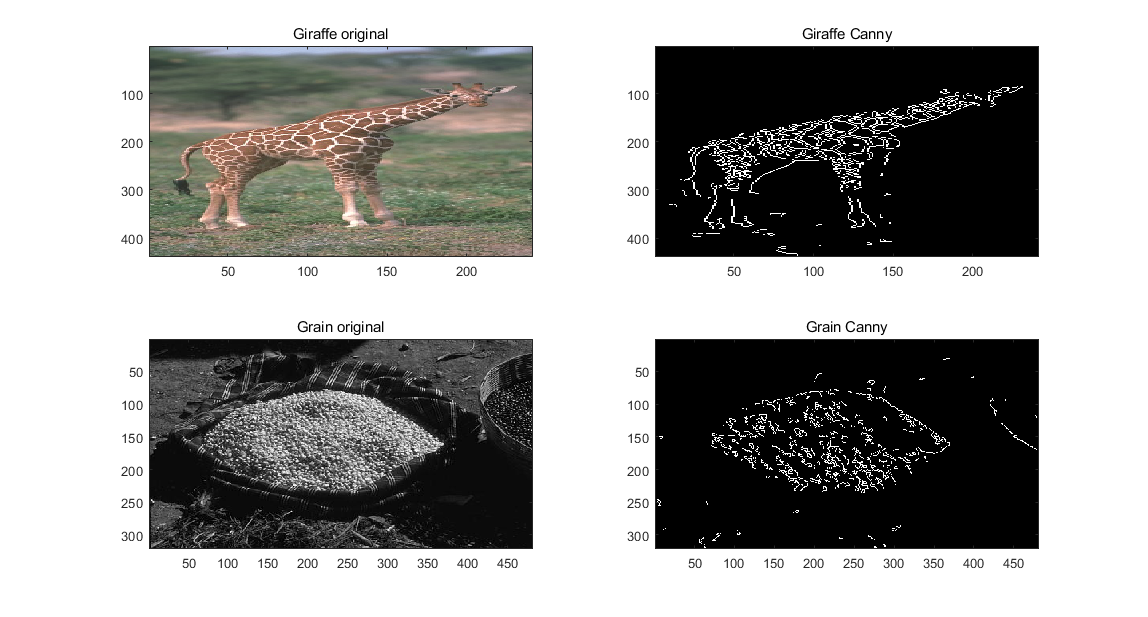


< Fig 4 >

When you see < Fig 1 >, it fills down to up. But it can divide region by up to down. < Fig 4 > is images that first column is original images, second column is apply watershed and final third column is apply watershed by inverse. Watershed by inverse mean apply not intensity but -1 x intensity.

It seems like painting on canvas. Also it seems like emphasize each object.

Prob 2. Edge Detection : Canny Edge Detection



< Fig 5 >

In canny method, there are two threshold; Low threshold and high threshold. High threshold for threshold that remain sure. Low threshold for edge link more or not. As you can see it removes much noise and show objects that I want. But still it does not show only boundary of objects. I don’t know how to do by methods that I learned so I use until canny method.

**Discussion :**

It still cannot show only object’s boundary. Actually I wanted to know that jobs when I study other similar cases. There are a lot of methods for edge detection. Maybe someday, I can find proper methods. But I think still perfect edge detection method is not exists. So I think previous information is most important.

**Reference :**

Chapter6 of TextbookFile

<https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>