Lab 5 – Edge and line detection Morselli Alberto (2102413)

-short report-

≥ ./approach

After having tried multiple “standard” approaches, like OTSU, watershed, and so on, without getting any promising result, I realized the segmentation task we wanted to achieve had to be solved in a different way. Segmenting cracks on asphalts is a pretty hard task, that involves solving the problem of having basically just 1 channel available to exploit for the segmentation (the grayscale value) being our target images asphalt images. And secondly, the ideal image with no noise involved is itself a noisy image, due to the asphalt pattern. These two elements combined require a pretty hard pre-processing before hoping to implement any standard segmentation algorithm.

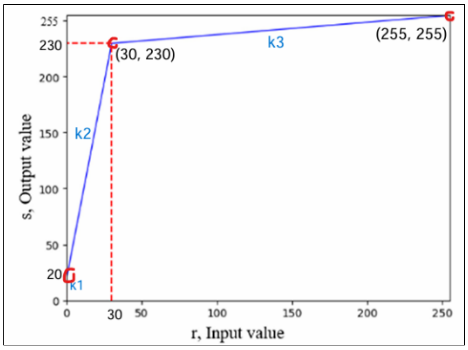
Looking on the web for existing working techniques i found the paper Crack Segmentation Extraction and Parameter Calculation of Asphalt Pavement Based on Image Processing by Zhongbo Li, Chao Yin and Xixuan Zhang, using a multi-stage detector composed by histogram equalization, segmented linear transformation, median filtering, Sauvola binarization, and the connected domain threshold method. I decided to follow their approach, tuning the process based on trial-and-error with trackbars.

≥ ./code details

The code itself is just a sequence of application of the different stages, among which i added a morphological closing operation stage between Sauvola binarization and connected domain final stage. I will provide here just a short explanation of the most relevant stages. The adopted sequence is:

1. grayscale color conversion
2. histogram equalization
3. segmented linear transformation / contrast stretching
4. median filtering
5. binarization trough Sauvola algorithm
6. morphological closing operation
7. connected domain treshold

Segmented linear treshold method (3), shown on the left, is a direct application of the law depicted here below.




```
void contrastStretching(const Mat& src, Mat& dst, Point p1, Point p2){
    Mat table(1, 256, CV_8UC1);

    float k1 = (p1.x == 0)              ? 0 : (p1.y)/(p1.x);
    float k2 = ((p2.x - p1.x) == 0)     ? 0 : (p2.y - p1.y)/(p2.x - p1.x);
    float k3 = ((255 - p2.x) == 0)      ? 0 : (255.0f - p2.y)/(255.0f - p2.x);

    for(int r=0; r<256; ++r){
        if (r <= p1.x)                  {table.at<uchar>(r) = k1 * r;}
        else if (r > p1.x && r <= p2.x) {table.at<uchar>(r) = k2 * (r-p1.x) + p1.y;}
        else                            {table.at<uchar>(r) = k3 * (r-p2.x) + p2.y;}
    }

    LUT(src, table, dst);
}
```

$$S = \begin{cases} K_1 \times r, & 0 \le r \le r_1 \\ K_2 \times (r - r_1) + s_1, & r_1 \le r \le r_2 \\ K_3 \times (r - r_2) + s_2, & r_2 \le r \le 255 \end{cases}$$

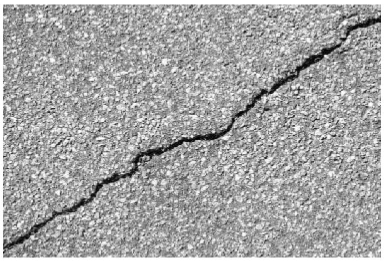
The points used for the application are p1 = (0,20) and p2 = (30,230). The transformation is applied with a LUT for a faster computation.

The standard values of median filtering kernel size (default=3), Sauvola algorithm parameters k (default=0.5) and n size (default=33), and structuring element size (default=3) of the closing operation had been varied from these default values due necessities on the images to obtain better results and the reason is explained after case by case. Lastly, the connected domain method perform the spatial segmentation, onto which i applied a treshold on area sizes of the different components obtained to extract just the n-th biggest ones (default=10).

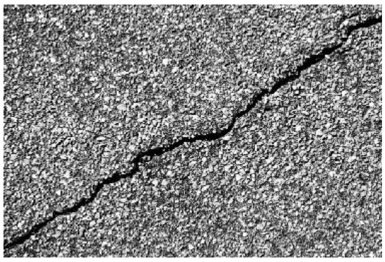
≥ ./experiment 1

The first image was the “easiest”, with highest crack to background contrast ratio and less “pattern noise” involved. I used all the standard parameters detailed above.

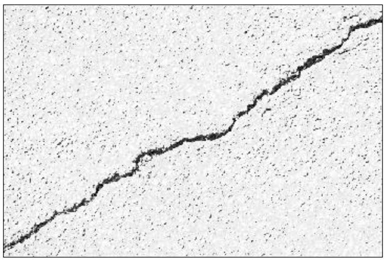
0.original



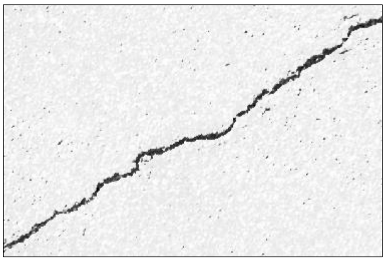
1.grayscale



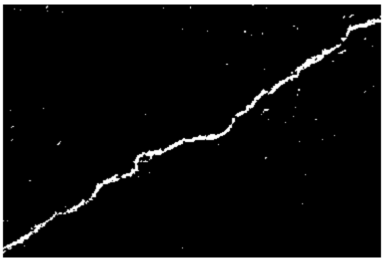
2.histogram equalization



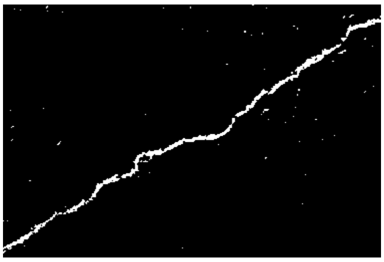
3. segmented contrast stretching



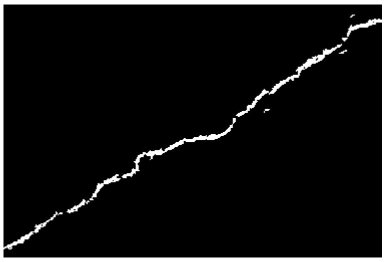
4. median filter (3x3)



5. sauvola binarization k=0.5 (33x33)



6. morphological closing (3x3)



7. connected domain treshold, c=10

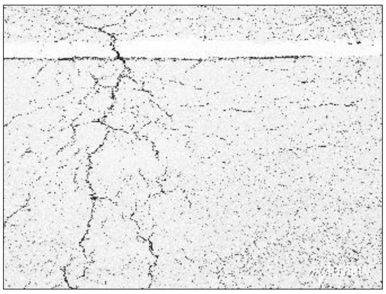
≥ ./experiment 2

This second image was the “hardest”, showing a very branched crack, very weak respect the surrounding. I then had to remove the median filtering application because it was filtering out too much, and i raised a bit the closing operation structuring element from a 3x3 to a 7x7.

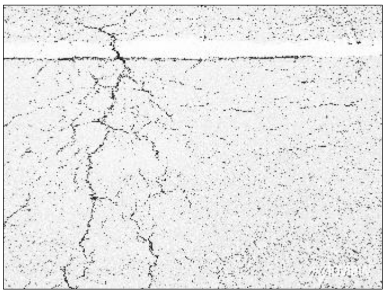
0.original

1.grayscale

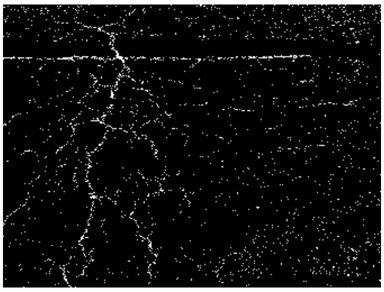
2.histogram equalization



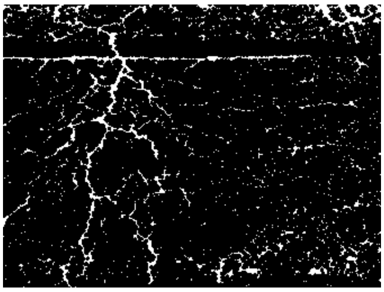
3. segmented contrast stretching



4. median filter (---)



5. sauvola binarization k=0.5 (33x33)

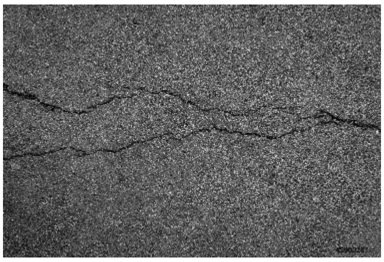


6. morphological closing (7x7)

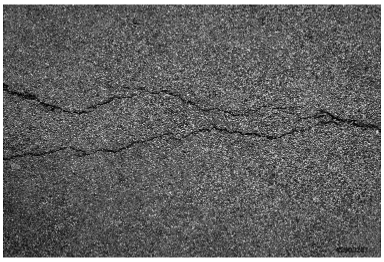
7. connected domain treshold, c=10

≥ ./experiment 3

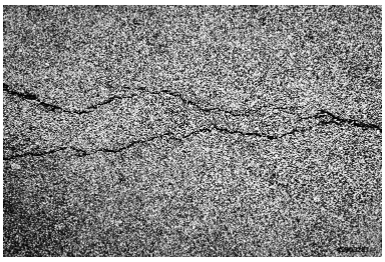
This third one was the noisiest. Median filtering played a very important role here, but the extracted profiles were too disconnected each other i had to raise the final treshold from showing 30 instead of 10 components. The image was also bigger in resolution terms, and i had to enlarge the Sauvola window size to keep the same "action" as the other two. I also raised a little the closing operation element to an 11x11 for the same reason.



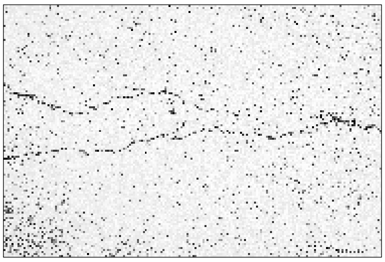
0.original



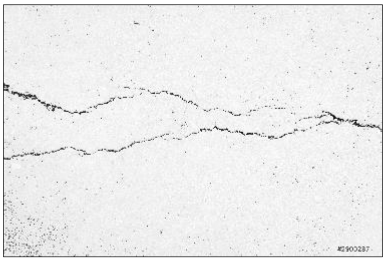
1.grayscale



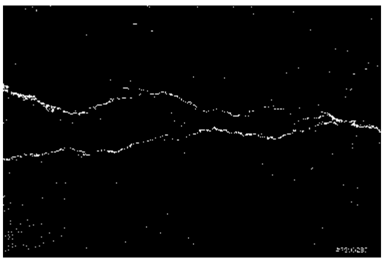
2.histogram equalization



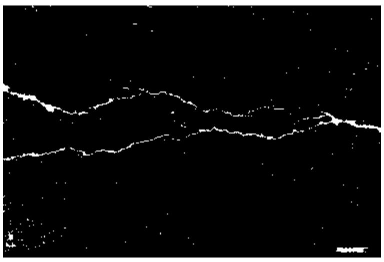
3. segmented contrast stretching



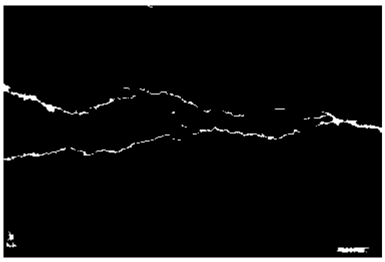
4. median filter (3x3)



sauvola binarization k=0.5 (99x99)



6. morphological closing (11x11)



7. connected domain treshold c=30