The method and implementation part of the report

Otsu

大津法（OTSU）是一种确定图像二值化分割阈值的算法，由日本学者大津于1979年提出。从大津法的原理上来讲，该方法又称作最大类间方差法，因为按照大津法求得的阈值进行图像二值化分割后，前景与背景图像的类间方差最大。

> Otsu method (OTSU) is an algorithm to determine the threshold value for image binarization segmentation, which was proposed by Japanese scholar Otsu in 1979. In terms of the principle of Otsu's method, the method is also called the maximum interclass variance method because the interclass variance between foreground and background images is maximum after image binarization segmentation according to the threshold value obtained by Otsu's method.

它被认为是图像分割中阈值选取的最佳算法，计算简单，不受图像亮度和对比度的影响，因此在数字图像处理上得到了广泛的应用。它是按图像的灰度特性，将图像分成背景和前景两部分。因方差是灰度分布均匀性的一种度量, 背景和前景之间的类间方差越大, 说明构成图像的两部分的差别越大, 当部分前景错分为背景或部分背景错分为前景都会导致两部分差别变小。因此, 使类间方差最大的分割意味着错分概率最小。

> It is considered as the best algorithm for threshold selection in image segmentation, is simple to compute, and is not affected by image brightness and contrast, so it has been widely used in digital image processing. It is to divide the image into two parts, background and foreground, according to the grayscale characteristics of the image. Because the variance is a measure of the uniformity of grayscale distribution, the larger the interclass variance between the background and foreground, the greater the difference between the two parts of the image, and when part of the foreground is divided into the background or part of the background is divided into the foreground, the difference between the two parts will become smaller. Therefore, the segmentation with the largest interclass variance implies the smallest probability of misclassification.

OTSU算法的假设是存在阈值TH将图像所有像素分为两类C1(小于TH)和C2(大于TH)，则这两类像素各自的均值就为m1、m2，图像全局均值为mG。同时像素被分为C1和C2类的概率分别为p1、p2。因此就有:

> The assumption of the OTSU algorithm is that there exists a threshold TH to classify all pixels of the image into two classes C1 (less than TH) and C2 (greater than TH), then the mean value of each of these two classes of pixels will be m1, m2 and the global mean value of the image will be mG. Also the probability of a pixel being classified into C1 and C2 classes is p1, p2 respectively. thus there is :

$$p_1 = \sum_{i=0}^{Th} p_i$$
$$p_2 = \sum_{i=Th+1}^{255} p_i$$

$$m_1 = 1/p_1 * \sum_{i=0}^{Th} i * p_i$$
$$m_2 = 1/p_2 * \sum_{i=Th+1}^{255} i * p_i$$

$$p_1 * m_1 + p_2 * m_2 = m_G (1)$$
$$p_1 + p_2 = 1 (2)$$

根据方差的概念，类间方差表达式为：

> According to the concept of variance, the expression for the variance between the two classes is:

$$\sigma^2 = p_1(m_1 - m_G)^2 + p_2(m_2 - m_G)^2$$

我们的最终目标就是通过遍历0到255之间的每一个值，找到一个合适的阈值Th，使得类间方差最大，即：

> Our ultimate goal is to find a suitable threshold Th that maximizes the interclass variance by traversing every value between 0 and 255, which is:

$$Th = argmax(\sigma^2)$$

理论上来说，通过这个阈值对图像进行二值化就可以得到理想的阈值分割结果。

> Theoretically, binarizing the image by this threshold will give the ideal threshold segmentation result.

Canny

传统的Canny边缘检测由四个步骤构成：

> The traditional Canny edge detection consists of four steps.

首先是使用高斯算子对图像进行滤波，用来平滑图像，消除噪声。

> First is the filtering of the image using a Gaussian operator, which is used to smooth the image and eliminate noise.

然后使用合适的梯度算子（一般是Sobel）对滤波后的图像进行卷积运算，目的是求出每个像素与周围8个像素点之间的梯度和方向的关系。

> The filtered image is then convolved using a suitable gradient operator (typically Sobel), with the aim of finding the relationship between the gradient and orientation of each pixel and the surrounding 8 pixel points.

可以看到Sobel算子的表达式为：

> The Sobel operator expression is:

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

第三步是非极大值抑制，也就是检测该像素点是否是在某个方向上的梯度值最大点，如果是则保留。

> The third step is non-maximum suppression, which is to detect whether the pixel point is the point with the maximum gradient value in a certain direction and keep it if it really is.

对于某个像素点，用上述算子卷积得到的结果分别为Gx和Gy，其梯度和方向的计算公式为：

> For each specific pixel point, the result of the convolution with the Sobel operator is Gx and Gy, and the gradient and orientation calculation formula is:

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

$$\theta = arctan(G_y/G_x)$$

为了简化梯度方向的计算，我们通常将梯度方向分割为4个方向，分别是：0度、45度、90度和135度。例如，原始梯度方向在0度和22.5度之间的像素点，我们认为其梯度方向为0度。原始梯度方向在22.5度和67.5度之间的像素点，我们认为其梯度方向为45度。剩余角度同理。

> For simplicity, we usually divide the gradient direction into 4 directions, namely 0 degrees, 45 degrees, 90 degrees and 135 degrees. For example, the original gradient direction in 0 degrees and 22.5 degrees, we think it is 0 degrees. The original gradient direction in 22.5 degrees and 67.5 degrees, we think it is 45 degrees. The remaining angle is operated with the same logic.

最后一步是双阈值分割，在这里我们高低两种阈值对每一个像素点进行判断，如果该像素点的梯度值大于高阈值，则认为该像素点为边缘点，如果该像素点的梯度值小于低阈值，则认为该像素点不是边缘点。若在高阈值和低阈值之间，则观察此点周围的八个点中是否出现梯度值大于高阈值的点，如果有，则认为该像素点为边缘点，如果没有，则认为该像素点不是边缘点。

> The last step is double threshold segmentation, where we high and low threshold for each pixel point, if the gradient value of the pixel point is greater than the high threshold, the pixel point is considered as an edge point, if the gradient value of the pixel point is less than the low threshold, the pixel point is considered as not an edge point. If it is between the high threshold and low threshold, then observe whether there is a point with a gradient value greater than the high threshold among the eight points around this point, and if there is, the pixel point is considered as an edge point, and if not, the pixel point is considered as not an edge point.

对于双阈值分割，传统的Canny法需要人为设定高阈值和低阈值。二者数值的比例通常为2:1或3:1

> For the double thresholds segmentation part, the traditional Canny method requires user manually setting high and low thresholds. The ratio of the two values is usually 2:1 or 3:1.

Twice thresholding

传统的Otsu法简单，快捷，有效。但是它有两个致命的缺点。首先传统Otsu法对噪声非常敏感，尤其是光照不均产生的影响。其次，由原理可知，传统Otsu法只能对一个事物进行分割，这也就意味着一旦图形变得复杂，或当目标和背景大小比例悬殊、类间方差函数可能呈现双峰或者多峰，这个时候分割效果就会很差。

> The traditional Otsu method is simple, fast and effective. But it has two fatal drawbacks. Firstly, the traditional Otsu method is very sensitive to noise, especially the effect of uneven illumination. Secondly, it is known from the principle that the traditional Otsu method can only segment one thing, which means that once the graph becomes complex, or when the target and background size ratio is disparate, the inter-class variance function may present double or multiple peaks, this time the segmentation effect will be very poor.

在本项目中尤为重要的是，待检测的路面进场会有白色的road mark。这些标记在直方图中会大量分布在高亮度区域，而带检测的裂缝由于其自身特点，占比远小于road mark。因此，对这种图进行传统Ostu法阈值分割一般会直接将road mark提取出来，忽律重要的路面缺陷。

为了改善这些缺点，我们可以采用二次阈值分割的方法。

> To improve these drawbacks, we introduce the method called twice thresholding.

Otsu threshold segmentation

第一次阈值分割，目的是为了消除road mark，以及噪声和光照的影响。传统Ostu法会使用0来替代大于阈值的像素点，255来代替小于阈值的像素点。但在本方法中，大于阈值的点会被设置成0，而小于等于阈值的点则保持不变。

> The purpose of the first threshold segmentation is to eliminate the road mark, as well as the effects of noise and illumination. The traditional Ostu method will use 0 to replace the pixel points greater than the threshold and 255 to replace the pixel points less than the threshold. However, in this method, points greater than the threshold will be set to 0, while points less than or equal to the threshold will remain unchanged.

$$G(x,y)=\begin{cases} 0, & f(x,y)\geq T \\ f(x,y), & f(x,y)<T \end{cases}$$

Iterative threshold segmentation

具体公式的出处

第二次阈值分割采用了最小误判概率法，也叫做迭代法。其原理是从0到255，遍历每一个像素值，找到使阈值错误划分前景与背景的概率最小的阈值。

> The second threshold segmentation uses the minimum probability of misclassification method, also called the iterative method. The principle is to iterate through each pixel value from 0 to 255 to find the threshold that minimizes the probability that the threshold misclassifies the foreground from the background.

我们假设分割阈值为T，则前景被错误分类为背景的概率是：

> We assume that the segmentation threshold is T. The probability that the foreground is misclassified as background is:

$$E_1(T) = \theta_1 \int_T^\infty p(x)dx$$

背景被错误分类为前景的概率是：

> The probability that the background is misclassified as foreground is:

$$E_2(T) = \theta_2 \int_{-\infty}^T q(x)dx$$

因此，阈值T造成错误分割的概率是：

> Therefore, the probability of incorrect segmentation due to the threshold T is:

$$E(T) = E_1(T) + E_2(T)$$

当E(T)取得最小值时，此函数在该点其导数为0。

> When E(T) obtains the minimum value, this function has a derivative of 0 at that point.

$$\frac{\partial E}{\partial T} = \theta_2 q(T) - \theta_1 p(T) = 0$$

$$\theta_2 q(T) = \theta_1 p(T)$$

我们在这里假设图像中的前景和背景像素灰度呈正态分布，均值和方差分别为μ_1和${\sigma_1}^2$和μ_2和${\sigma_2}^2$，则有：

> We assume here that the foreground and background pixel gray levels in the image are normally distributed with means and variances of μ_1 and ${\sigma_1}^2$ and μ_2 and ${\sigma_2}^2$, respectively, then we have.

$$\theta_2 e^{\frac{-(\mu_2-T)^2}{2\sigma_2^2}} = \theta_1 e^{\frac{-(\mu_1-T)^2}{2\sigma_1^2}}$$

$$T = \frac{\mu_1+\mu_2}{2}$$

现在，我们就得到开始迭代所需要的核心公式了。

> Now we get the core formula we need to start the iteration.

开始迭代前，首先我们要找到一个合适的初始阈值，这个初始阈值可以通过对图像的灰度分布进行分析来得到。这里，我们用直方图的中位数来作为初始阈值。

> Before starting the iteration, first we have to find a suitable initial threshold, which can be obtained by analyzing the grayscale distribution of the image. Here, we use the median of the histogram as the initial threshold.

第二步，我们从初始阈值到255，遍历所有灰度，计算该灰度分割出的前景和背景分别的平均灰度值，记为μ_1和μ_2

> In the second step, we iterate through all the grayscales from the initial threshold to 255, and calculate the average grayscale value of the foreground and background segmented by that grayscale, respectively, noted as μ_1 and μ_2

第三步，从这两个平均灰度值中计算新的阈值：

> In the third step, a new threshold is calculated from these two average grayscale values.

$T_{k+1} = \frac{\mu_1+\mu_2}{2}$

如果此时$T_k = T_{k+1}$则停止迭代，此时的阈值即为我们想要的结果。若二者的值不相等，则将阈值设定为T_{k+1}，继续迭代。

> If at this point $T_k = T_{k+1}$ then stop the iteration and the threshold value at this point is the result we want. If the two values are not equal, then set the threshold to T_{k+1} and continue iteration.

最后，我们利用得到的阈值T_k对图像进行分割：

> Finally, we segment the image using the obtained thresholds T_k.

$$D(x,y) = \begin{cases} 0, & G(x,y) \geq T_k \; or \; G(x,y) == 0 \\ 255, & G(x,y) < T_k \end{cases}$$

Improved Canny

传统的Canny算法被证明是最好的边缘检测算法。他能尽可能地找到图像中存在的实际边缘，且标识出的边缘和实际边缘的差异很小。

> The traditional Canny algorithm is proven to be the best edge detection algorithm. He can find the actual edges present in the image as much as possible and the difference between the identified edges and the actual edges is small.

但它的缺点也很明显。首先Canny算法对噪声非常敏感，传统的高斯滤波不足以去掉所有的噪声。其次Sobel算子对灰度渐变的图像处理效果较好，而我们的项目中需要处理的道路裂缝通常都是灰度突变的。然后，Sobel算子对横向和纵向的灰度变化处理效果较好，而道路裂缝通常是斜向或网纹状的。

> But it also has obvious disadvantages. Firstly the Canny algorithm is very sensitive to noise and the traditional Gaussian filtering is not enough to remove all the noise. Secondly the Sobel operator is better for images with gradual grayscale changes, while the road cracks we need to deal with in our project are usually abruptly changing in grayscale. Then, the Sobel operator is better for processing grayscale variations in horizontal and vertical directions, while road cracks are usually oblique or reticulated.

为了改良这些缺点，我们采用了改进Canny算法。

> To improve these drawbacks, we used the improved Canny algorithm.

Neo gradient operator

在这里，我们引入了一个改良的的梯度算子obilque，将它和Sobel一起使用，从而获得更好的结果。其表达式为：

$$Oblique_1 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

$$Oblique_2 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$G_1(x,y) = f(x,y-1) + 2f(x+1,y-1) - f(x-1,y) + f(x+1,y) - f(x,y+1) - 2f(x-1,y+1)$$

$$G_2(x,y) = -2f(x-1,y-1) - f(x,y-1) - f(x-1,y) + f(x+1,y) + f(x,y+1) + 2f(x+1,y+1)$$

$$G(x,y) = \sqrt{G_1(x,y)^2 + G_2(x,y)^2}$$

可以看到，新引入的梯度算子着力于增强对斜向梯度变化的检测。

> It can be seen that the newly introduced gradient operator focuses on enhancing the detection of oblique gradient changes.

The improved way to calculate gradient

我们修改了Canny算法的步骤，在非极大值抑制步骤后加入了一个综合的操作。综合Sobel算子和Oblique算子的梯度。遍历每一个像素点，取两个算子中较大的梯度为最终结果。

> We modified the steps of the Canny algorithm by adding a synthesis operation after the non-maximum suppression step. The gradient of the combined Sobel operator and Oblique operator. The gradient of the larger of the two operators is taken as the final result by traversing each pixel point.

The half-automatic double thresholding

在调用传统的Canny算法，如python的OpenCV库时，往往需要人为指定两个阈值进行双阈值选择。它的缺点在于，对于不同的路面，这个阈值的选择往往是不同的，对每一张图片都进行人为调整会带来很多不便。

> When calling traditional Canny algorithms in the built-in library, such as python's OpenCV library, it is often necessary to artificially specify two thresholds for double-threshold selection. Its drawback is that this threshold selection is often different for different pavements, and human adjustment for each image can cause a lot of inconvenience.

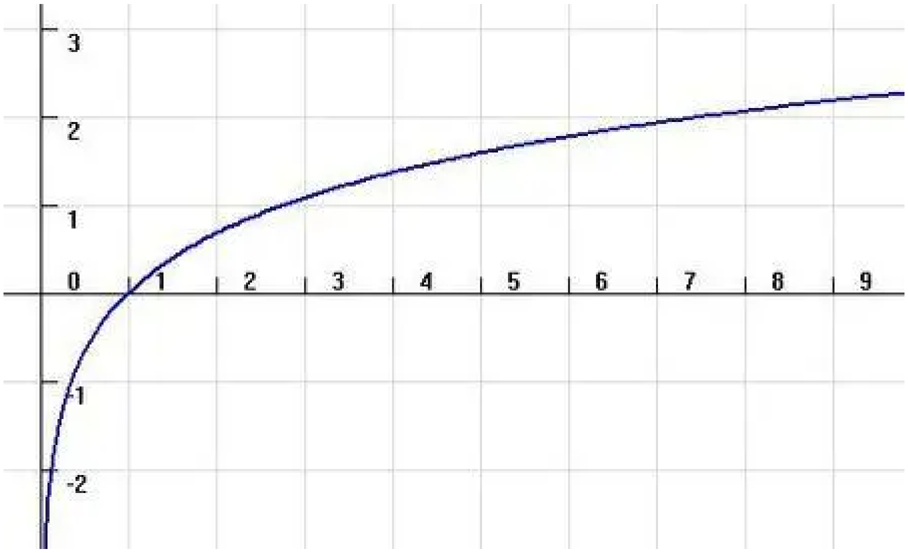
在我们的改进Canny算法中，我们采用了直方图统计的方式，找到了图像中的最大灰度值，将其乘以一个权重便可以得到高阈值，对于水泥路面这个权重被设置为0.175，对于沥青路面这个权重被设置为0.2。而低阈值通常为高阈值的二分之一或三分之一。

> In our modified Canny algorithm, we use histogram statistics to find the maximum gray value in the image and multiply it by a weight to obtain the high threshold, which is set to 0.175 for concrete pavements and 0.2 for asphalt pavements, while the low threshold is usually one-half or one-third of the high threshold.

Log transformation

图像的LOG变换是一种常用的图像处理方法，它可以将图像的亮度值转换为对数值，从而使得图像的亮度值变化更加平缓，这样可以更好地提取图像的边缘。它利用对数曲线在像素值较低的区域斜率大，在像素值较高的区域斜率较小的特点，所以图像经过对数变换后，较暗区域的对比度将有所提升，所以就可以增强图像的暗部细节。而我们要提取的裂缝信息恰好就是图片中较暗的区域。

> The log transformation is a common image processing method, which can transform the intensity of the image into logarithm value, so that the brightness of the image changes more smoothly, and this can make the edge information more clear. It takes advantage of the fact that the logarithmic curve has a large slope in regions with lower pixel values and a smaller slope in regions with higher pixel values, so the image will have an increased contrast in the darker regions after the logarithmic transformation, thus, the dark details of the image can be enhanced. We want to extract the information of road cracks which is exactly in the dark region of the image.



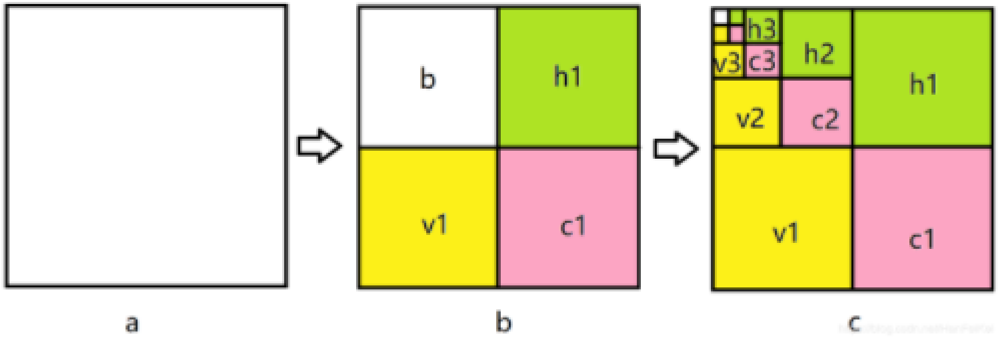
Wavelet filtering

和傅立叶变换很相似，小波变换使用一系列不同尺度的小波去分解原函数，变换得到的结果是原函数在不同尺度小波下的系数，这些系数可以用来描述原函数的不同特征。小波通过平移卷积得到原函数的时间特性。小波通过尺度变换来得到原函数的频率特性。

> Much like the Fourier transform, the wavelet transform uses a series of wavelets at different scales to decompose the original function. The result of the transform is the coefficients of the original function at different scales of wavelets, and these coefficients can be used to describe different characteristics of the original function. The wavelets are convolved by translation to obtain the time characteristics of the original function. The wavelet is transformed by scale to get the frequency characteristics of the original function.

我们将图像的像素点压缩成一个一维序列，这就是我们的原函数。然后对这个原函数使用一次小波变换，得到四个系数，分别是低频分量和三个不同方向的高频分量。这里，我们将高频分量视为噪声，对其进行硬阈值滤波操作。最后，对滤波后的四个系数进行逆小波变化，得到小波变换后的图像。

> We compress the pixel points of the image into a one-dimensional sequence, which is seemed as the original function. Then we use one wavelet transform on this original function to obtain four coefficients, which are the low-frequency components and the high-frequency components in three different directions. Here, we consider the high-frequency components as noise and perform a hard threshold filtering operation on them. Finally, the filtered four coefficients are subjected to an inverse wavelet change to obtain the wavelet transformed image.



Reference

1. Research on crack detection method of airport runway based on twice-threshold segmentation
2. Image Edge Detection Algorithm Based on Improved Canny Operator