1. **Otus单阙值算法**
2. Otus 算法原理

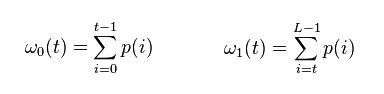
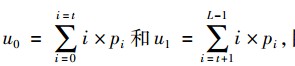
根据图像的灰度特性，将图像分成2个类，这两个类之间的类间方差越大，这些类之间的差别就越大，此时分割的错分概率也为最小。因方差是灰度分布均匀性的一种度量,方差值越大,说明构成图像的两部分差别越大,当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小，因此使类间方差最大的分割意味着错分概率最小。

于是我们寻找一个阈值t，使得类内方差最小，等价于类间方差最大：

C:\Users\qian\AppData\Roaming\Tencent\Users\563174213\QQ\WinTemp\RichOle\9U$LS$IT8ZZF3RUOPPJGAE5.jpg

C:\Users\qian\AppData\Roaming\Tencent\Users\563174213\QQ\WinTemp\RichOle\H19@A391D`S}1355FBO@BY6.jpg

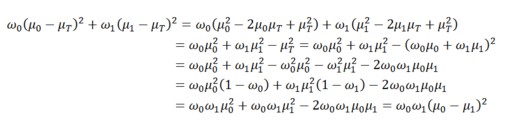
分割出的前景和背景两部分构成了整幅图像，而前景取值u0，概率为w0，背景取值u1，概率为w1，总均值为u，

因为

C:\Users\qian\AppData\Roaming\Tencent\Users\563174213\QQ\WinTemp\RichOle\BUWDWELG_[%9147}SRQL%JR.jpg

可以推导出：



2. 算法实现步骤

1. 计算图像的直方图，并保存每一个像素灰度值出现的概率；
2. 做遍历，使阙值T从图像的最小灰度值开始遍历，直到图像的最大灰度值；
3. 做循环并计算，将所有I>= T的为class1，I<T的为class2。 分别算出两个类在整个图像中所占百分比w1，w2 和平均灰度值 u1，u2；
4. 计算整个图像的平均灰度值Imean和方差var；
5. T遍历了后，得到最佳seuil，使得类间方差最大值 sigma2B。
6. **Otus多阙值算法**

1. Otus 多阙值分割原理

假设有N个阙值把图像分为N+1个类，那么阙值区间被分为[0,1,…,T1],[T1+1,…,T2],…,[Tn+1,L-1].在整个阙值空间上，我们找到最优的阙值组合，是类间的方差最大：

C:\Users\qian\AppData\Roaming\Tencent\Users\563174213\QQ\WinTemp\RichOle\EHGK1`O8J~QJH~MFR6S)TRI.jpg

C:\Users\qian\AppData\Roaming\Tencent\Users\563174213\QQ\WinTemp\RichOle\O{)9)({C@F4P45JGQC8R[UN.jpg

在这次实验中，我们测验了2阙值，3阙值的分割。

2. 算法思路及步骤

1. 利用otus经典算法，计算出第一个阙值点T1，该阙值把区间分为两类c1和c2；
2. 分别计算两个区间的类间最大方差，var1和var2，
3. 比价var1和var2，在方差大的那个区间内部再进行下一次分割，得到第二个阙值T2，同时把剩下的较小方差rest\_var保存下来，用于下一次阙值计算和方差比较；
4. 阙值T2将该区间再次分为两类c3和c4，计算其内部的最大方差var3和var4,比较并找到最大方差max(var3,var4,rest\_var)对于的阙值T3；
5. 把T3加入阙值组中。计算T3得到的两个子区间求两个最大方差，并把他们与之前剩下的两个方差进行比较，找到最大的方差对应的阙值。
6. 反复循环执行以上步骤，直至达到所预想的类别数目。得到最优阙值组合。

**III. 算法实现以及结果**

中心思想：

1. 将图像分为N个block，利用otus多阙值算法在每一个block里分别计算每个block对应的阙值组。

2. 在每一个block里，计算每一列中像素灰度值出现在分类中的概率。比较并找出最大的概率以及其所对应的类，将该列的所有值赋以我们自定义的该类灰度值。

3，为了更好的进行阙值分割，在进行分割前，对图像进行预处理，使消除图像的噪音，增强对比度等。