第5章

图像分割

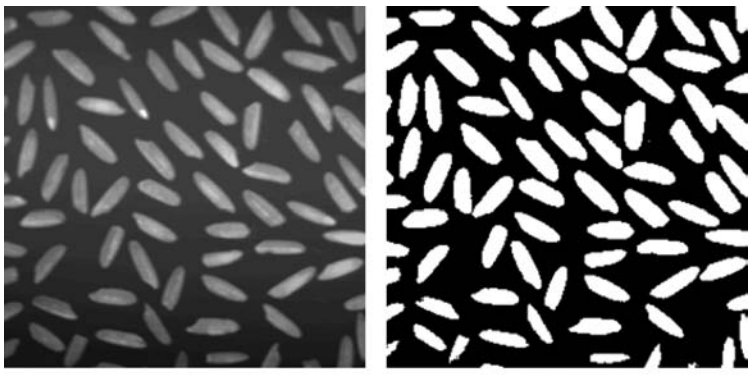
本章讲述图像分割的基本概念，包括图像分割的定义、图像分割和边缘检测的区别、阈值化、二值化、半调阈值化、多阈值化等；讲述基于直方图的阈值选取方法，包括最小误差法和最大差距法；讲述多次分割法及全局阈值、局部阈值和自适应阈值等应用策略；讲述二维直方图、边缘强度加权直方图和等量像素法直方图的构造方法；讲述聚类分割、区域增长与分裂合并算法和基于某种稳定性的图像分割方法；最后以光照不均的消除与图像分割为例，讲述图像分割和前面所学内容的灵活运用。

5.1 基本概念

5.1.1 什么是图像分割

图像分割(image segmentation)就是按照一定的规则将图像划分成若干有意义的区域。即各区域的并集是整个图像，各区域的交集为空。

例如，图 5-1(a)所示的米粒图像被分割成如图 5-1(b)所示的黑色区域和若干白色区域，它们的并集是整幅图像，黑色区域和白色区域无交集。



(a) 原始图像　　(b) 分割结果

图 5-1　米粒图像分割

从图像分割的定义来看，它并没有说明按什么样的规则和实现什么目的才有意义，这也从侧面说明了图像分割的复杂性。一般来说，若分割结果符合特

定场合的应用,这样的分割才称为有意义。一种图像分割方法有可能非常符合甲的需求,但与乙的需求矛盾。规则和需求的多样性决定了图像分割方法的多样性,目前为止图像分割有近 2000 种方法,可见其方法之多,同时也说明不存在通用的图像分割方法。

图像分割经常根据区域颜色值、灰度值或纹理等特征的差异来划分不同的区域。由于光照不足、光照不均、逆光、目标占空比等因素对图像分割的影响较大,因此它们是考查一种图像分割方法是否有效、是否稳健时不可或缺的条件。

5.1.2 图像分割与边缘检测的区别

首先,边缘检测不是图像分割。虽然边缘检测的最终结果也是二值图像,比如白色的像素是边缘点,黑色的像素是内部点,但是根据图像分割的定义,图像分割的结果是区域,而不是稀疏的边缘点,所以边缘检测不能被称为图像分割。

其次,图像分割的结果是若干个区域,每个区域内部的像素是互相连通的,因此区域是有轮廓的。区域的轮廓肯定是闭合曲线,这非常有利于目标面积和目标形状参数的测量。相比之下边缘检测想要得到闭合曲线是非常难的,对于具体的应用而言,在无法确保任何情况下阈值都合理的前提下,就无法保证边缘点一一相连。虽然 4.5.1 节对图 4-31(a)采用二阶微分算子和一阶微分算子相结合的方法,得到了米粒边缘的闭合曲线,但参数 a_0 和 t 是对图 4-31(a)进行精心设计的,如果把图 4-31(a)更换成相同场景米粒的其他图像或者光照稍暗一些,就有可能无法保证边缘闭合了。

5.1.3 阈值化

阈值化(thresholding)是经典的图像分割方法。把阈值(threshold)作为区分目标像素与背景像素的灰度门限,灰度值大于或等于阈值的像素属于物体,而其他的像素属于背景,如式(5-1)所示。使用这种方法,可以有效地分割目标与背景之间存在明显灰度差别的图像,且实现过程非常简单,计算量极低。

$$G(x,y)=\begin{cases}1, & g(x,y)\geqslant \text{threshold}\\ 0, & \text{其他}\end{cases} \tag{5-1}$$

在计算机中灰度值 0 和 1 都会显示成黑色,为了方便观察,一般采用式(5-2)的做法:

$$G(x,y)=\begin{cases}255, & g(x,y)\geqslant \text{threshold}\\ 0, & \text{其他}\end{cases} \tag{5-2}$$

把图像分割成两种灰度值的过程,常被称为图像二值化(image binarization)。

【算法 5-1】 图像二值化

```
void RmwThreshold(BYTE * pGryImg, int width, int height, int thre)
{
    BYTE * pCur = pGryImg;

    for (; pCur < pGryImg + width * height;) * (pCur++) = ( * pCur >= thre) * 255;
    return;
}
```

阈值有时也被用来屏蔽掉图像中背景部分而保留目标部分的灰度信息，如式(5-3)所示，这种分割方法称为半调阈值化(semi-thresholding)。

$$G(x,y)=\begin{cases} g(x,y), & g(x,y)\geqslant \text{threshold} \\ 0, & \text{其他} \end{cases} \tag{5-3}$$

有时也采用多阈值化(multi-thresholding)将图像的灰度范围划分成有限的灰度种类，其处理后的结果不再是二值的，而是结果图像中只有几种灰度值。

实际上，图像分析都会直接或间接地使用阈值化技术，例如第 4 章中通过设定阈值实现从边缘强度得到边缘点。那么如何选取阈值呢？

5.2　基于直方图的阈值选取

第 2 章中介绍了直方图的特点，比如从直方图中很容易看出图像中有几类目标和各类目标的灰度特征，这意味着可以通过直方图数据来选择合适的阈值。下面重点讲述两个典型的基于直方图的阈值选取方法：一个是基于解析描述的最小误差法求解阈值；另一个是基于阈值枚举的最大差距法选取阈值。

另外，还有很多基于直方图选取阈值的方法，比如直方图峰谷分析法以及从直方图峰谷分析法引申来的最大熵阈值选取法等，由于它们缺乏一定的理论基础，实用性也较差，因此不再讲述。

需要注意的是，基于直方图选取阈值是阈值选取的主要方法，直方图并不一定限于灰度值的直方图，也可以是其他物理量的直方图。

5.2.1　最小误差法

最小误差法(minimum error thresholding)源于 Bayes 最小误差分类方法。假设目标灰度和背景灰度是独立分布的随机变量，并且各自的概率密度(直方图)服从一定的正态分布(高斯分布)，由目标和背景组成的图像的灰度密度分布服从混合正态分布，分别如图 5-2(a)、图 5-2(b)和图 5-2(c)所示。

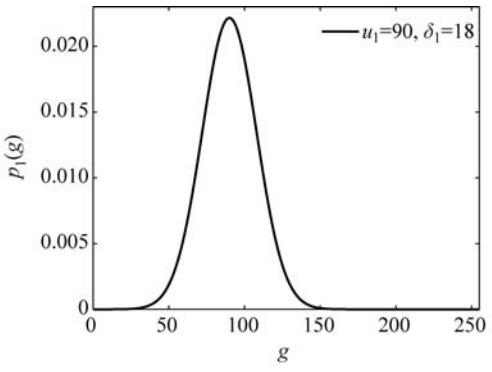
在图 5-2(c)中，可以直观地发现：当阈值取灰度值 90(u_1)时，会把很多背景像素划到目标中，背景像素划到目标中的概率很大；当阈值取灰度值 155(u_2)时，会把很多目标像素划到背景中，目标像素划到背景中的概率很大；当阈值取灰度值 125 左右时，只有少量的背景像素划到目标中和少量的目标像素划到背景中。因此，最优阈值就是使总的错误概率最小的值，定义一个评价函数如式(5-4)所示。

$$E_{\text{all}}=E_{12}(t)\times w_1+E_{21}(t)\times w_2 \tag{5-4}$$

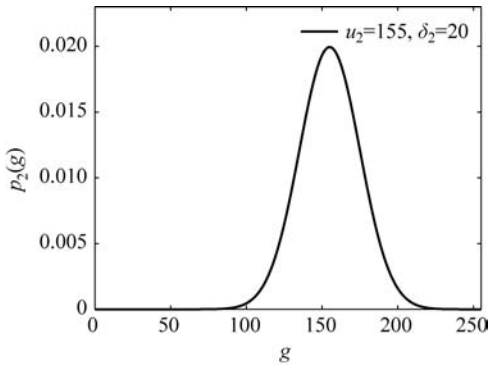
$$E_{12}(t)=\int_t^{255} p_1(g)\mathrm{d}g \tag{5-5}$$

$$E_{21}(t)=\int_0^{t} p_2(g)\mathrm{d}g \tag{5-6}$$

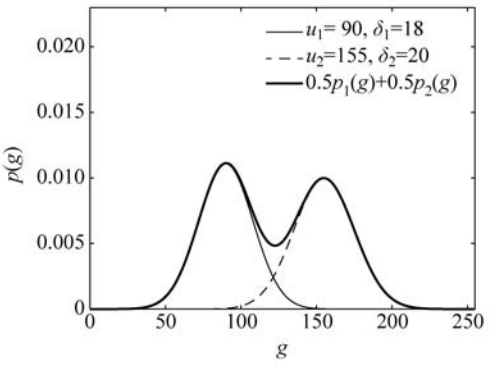
其中，$E_{12}(t)$是阈值为 t 时背景被错误分类成目标的概率，$E_{21}(t)$是阈值为 t 时目标被错误分类成背景的概率；w_1 是图像中背景像素的所占比例，w_2 是图像中目标像素的所占



(a) 背景像素的灰度直方图



(b) 目标像素的灰度直方图



(c) 既有背景又有目标的直方图

图 5-2 概率密度函数

比例，$w_1+w_2=1$；$p_1(g)$是背景像素的概率密度，$p_2(g)$是目标像素的概率密度；E_{all}是阈值为 g 时的总错误率。

使 E_{all} 取最小值时的 t 即为最佳阈值，即式(5-4)的一阶导数等于 0 时的 t。当 $p_1(g)$和 $p_2(g)$的方差相等，即 $\sigma_1=\sigma_2=\sigma_c$ 时，得到如下结论(中间求导和求解的过程忽略)：

$$t=\frac{u_1+u_2}{2}-\frac{\sigma_c^2}{u_1-u_2}\ln\left(\frac{w_1}{w_2}\right) \tag{5-7}$$

若再有 $w_1=w_2$，则 $t=\dfrac{u_1+u_2}{2}$，即阈值是直方图中 2 个波峰对应的灰度值的均值。在一般情况下，同时满足 $\sigma_1=\sigma_2$ 和 $w_1=w_2$ 是很难的，如果不加条件地把阈值取为直方图中 2 个波峰对应的灰度值的均值是不严谨的。仔细分析式(5-7)可知，当 $w_1>w_2$ 时，阈值变大，向 u_2 移动；当 $w_2>w_1$ 时，阈值变小，向 u_1 移动。这可以理解为，若选取的阈值远离占空比大的类别，就能够减小分类错误概率。

由上可知，最小误差法求解阈值需要知道 6 个参数，即 u_1、σ_1、u_2、σ_2、w_1、w_2。在一般情况下，上述参数是未知的，因此需要根据图像的直方图来估计它们。在 2 个正态分布下的混合密度函数为：

$$p(g)=w_1\frac{1}{\sqrt{2\pi\sigma_1^2}}e^{-\frac{(g-u_1)^2}{2\sigma_1^2}}+w_2\frac{1}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{(g-u_2)^2}{2\sigma_2^2}} \tag{5-8}$$

假设实际的直方图为 $h(g)$，因此 $h(g)$ 和 $p(g)$ 的均方误差（实际值与估计值之间的误差）为：

$$\mathrm{Err}=\frac{1}{256}\sum_{g=0}^{255}[h(g)-p(g)]^2 \tag{5-9}$$

采用最小二乘法求解出 Err 取最小值时的 u_1、σ_1、u_2、σ_2、w_1、w_2，代入式(5-4)即可求解最优阈值。

最小误差法的缺点是，若目标和背景的分布不满足正态分布，或者图像中除了背景和目标外还存在其他类别的区域时，其有效性就会大大降低。

5.2.2　最大差距法

最大差距法是日本学者 Otsu 在 1979 年提出的阈值求取方法（A threshold selection method from gray-level histogram[J]. IEEE on SMC-9, 1979(3): 62-66），常被称为 Otsu 或者大津阈值化。其动机是使用阈值把图像分割为背景和目标时，最佳阈值应该使得背景和目标之间的距离最大，即背景和目标的类间距最大。

假设阈值为 t，把图像分割为背景和目标 2 类区域。此时，背景部分的灰度均值为 $u_1(t)$，目标部分的灰度均值为 $u_2(t)$，背景部分在图像中的所占比例为 $w_1(t)$，目标部分在图像中的所占比例为 $w_2(t)$。有：

背景像素到目标像素的距离为：$d_{12}(t)=|u_1(t)-u_2(t)|\times w_1(t)$。

目标像素到背景像素的距离为：$d_{21}(t)=|u_1(t)-u_2(t)|\times w_2(t)$。

目标像素和背景像素的类间距定义为：$d(t)=d_{12}(t)\times d_{21}(t)$。

那么，最佳阈值应该为：

$$t=\arg\max_{g_{\min}\leqslant t<g_{\max}} d(t) \tag{5-10}$$

式(5-10)的含义是在灰度最小值 $g_{\min}$ 到灰度最大值 $g_{\max}$ 之间枚举 t，$d(t)$ 取最大值时的 t 即为所求。

【算法 5-2】　Otsu 阈值求取

```
int RmwGetOtsuThreshold(int *histogram,int nSize)
{
    int thre;
    int i, gmin, gmax;
    double dist, f, max;
    int s1, s2, n1, n2, n;

    //step.1 ----- 确定搜索范围:最小值 ---------------------------- //
    gmin = 0;
    while (histogram[gmin] == 0) ++gmin;
    //step.2 ----- 确定搜索范围:最大值 ---------------------------- //
    gmax = nSize - 1;
    while (histogram[gmax] == 0) -- gmax;
    //step.3 ----- 搜索最佳阈值 ---------------------------------- //
    if (gmin == gmax) return gmin;                    //不满足 2 类分布
    max = 0;
```

```
    thre = 0;
    //初始化 u1
    s1 = n1 = 0;
    //初始化 u2
    for (s2 = n2 = 0, i = gmin; i <= gmax; i++)
    {
        s2 += histogram[i] * i;
        n2 += histogram[i];
    }
    //搜索
    for (i = gmin, n = n2; i < gmax; i++)
    {
        if (!histogram[i]) continue;                    //加速
        //更新 s1s2
        s1 += histogram[i] * i;
        s2 -= histogram[i] * i;
        //更新 n1n2
        n1 += histogram[i];
        n2 -= histogram[i];
        //评价函数
        dist = (s1 * 1.0/n1 - s2 * 1.0/n2);
        f = dist * dist * (n1 * 1.0/n) * (n2 * 1.0/n);
        if (f > max)
        {
            max = f;
            thre = i;
        }
    }
    // step.4 ----- 返回 ---------------------------------------------- //
    return thre + 1;                                    //二值化时是用>= thre,所以要 + 1
}
```

Otsu 方法有两个优点：一是不要求背景和目标的灰度值统计满足正态分布；二是使用了背景和目标的灰度均值，具有很好的抗噪能力。这两个优点决定了 Otsu 是应用最广的阈值求取方法。另外，在计算方法上，最小误差法是解析法，而 Otsu 是枚举法。

5.2.3 多次分割法

在很多实际应用中，面对复杂多变的动态场景，要保证场景中只有 2 类目标是无法做到的，比如红外图像中非结构化道路的道边检测，要在有阴影、水迹、车辆、行人、植被等的图像中找到道边。但是，若处理的图像中区域类别数大于 2，最小误差法和最大差距法就都失去了理论基础，它们的分割效果就无法保证。

随着计算机内存访问速度越来越快，现在已经有 DDR6 内存了，执行一次图像二值化的时间花费非常少，因此多次分割法在实际应用中也经常被使用到。

所谓多次分割法就是对阈值的选择不给出任何准则，它把灰度最小值 $g_{\min}$ 到灰度最大值 $g_{\max}$ 之间的每个或者多个灰度值当作阈值，把图像最多进行$(g_{\max}-g_{\min}-2)$次二值化，最多得到$(g_{\max}-g_{\min}-2)$个二值图像，然后在每个二值图像中，根据目标形状、轮廓像素对应的边缘强度、区域对应的灰度方差等特征，选择出更佳的目标区域。实质上，它是把评价函数放到了图像二值化后对图像分割结果进行评价，而不是在图像二值化前

对阈值进行评价。

多次分割法对阈值的选择不进行评判，这是它的优点；缺点是它需要执行多次图像二值化和对每个二值图像进行分割结果的评价，速度较慢。

5.2.4　全局阈值、局部阈值与自适应阈值

阈值的使用可分为全局阈值、局部阈值和自适应阈值。对整幅图像的所有像素都使用同一个阈值，则该方法被称为全局阈值(global thresholding)；若把图像分成若干个块，每块使用不同的阈值，则该方法被称为局部阈值(local thresholding)；若对每像素都使用不同的阈值，则该方法被称为自适应阈值(adaptive thresholding)。和第 2 章讲过的图像增强以及第 3 章讲过的图像平滑一样，阈值也可以分别在帧级、块级和像素级上使用。

全局阈值的具体实现可以是对整幅图像构造一个直方图，根据该直方图得到一个阈值，使用该阈值对所有像素进行分类。局部阈值的具体实现可以是把整幅图像分成若干个块，每块构造一个直方图，根据该直方图得到一个阈值，使用该阈值对该块内的所有像素进行分类。自适应阈值的具体实现可以是对整幅图像进行重叠分块，每块构造一个直方图，根据该直方图得到一个阈值，这样就得到了多个阈值，这些阈值分别来自每一个块；把这些阈值顺序排列起来，就构成了一个小图像 A，A 中每个像素的值都是各个块的阈值，把 A 图像通过双线性插值放大到和原图像等大，得到 B 图像，则 B 图像中每个像素的值就是原图像中每个像素的阈值。

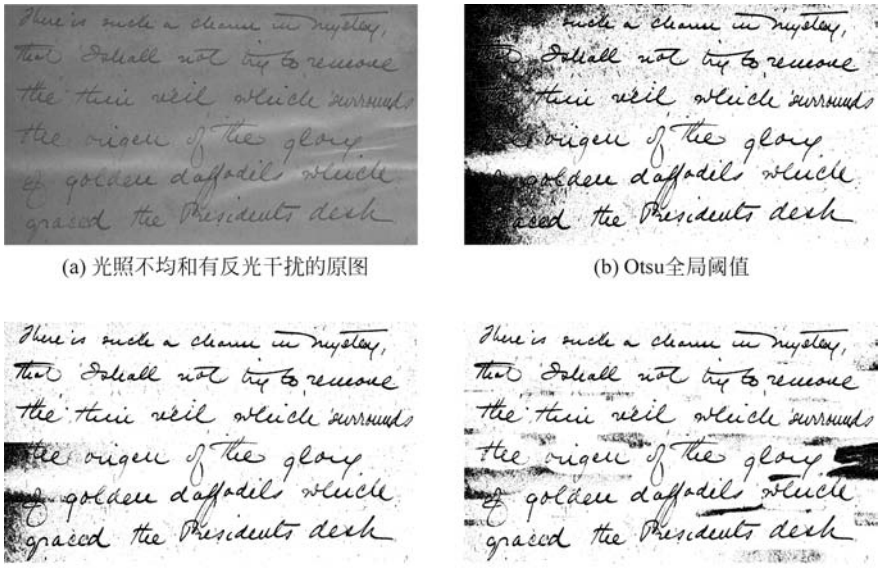
图 5-3(a)是光照不均的原始图像，对该图像构造直方图采用 Otsu 求取阈值，使用该阈值对所有像素进行分割，得到结果图像如图 5-3(b)所示。从图 5-3(b)可以看到左下角的米粒分割错误，这是由于原始图像的左下角较暗所致。为了尽可能地消除光照的不均，把原始图像在水平方向和垂直方向都分成 4 条，共得到 16 个图像块，对每块都采用 Otsu 单独分割，得到的分割结果如图 5-3(c)所示。可以看出分割结果良好，这是因为图像块相对较小，可以认为光照在这个小的局部区域内是均匀的。

(a) 光照不均的原始图像　(b) Otsu全局阈值　(c) 4×4分块的Otsu阈值

图 5-3　全局阈值与局部阈值

图 5-4(a)是光照不均的文档图像，并且由于纸面有皱褶，形成了几条高亮的反光区域，其全局阈值的结果如图 5-4(b)所示，其分块阈值得到的结果如图 5-4(c)所示，其自适应阈值得到的结果如图 5-4(d)所示。由于第三类区域“反光”的存在，虽然图 5-4(c)和图 5-4(d)所

示的分割结果比图 5-4(b)有所改善,但仍存在少量像素分割错误(在 5.7 节中会得到解决)。



(a) 光照不均和有反光干扰的原图　(b) Otsu全局阈值

(c) 4×4分块的Otsu阈值　(d) 自适应阈值

图 5-4　文档图像的二值化

5.3　面向阈值选取的直方图构造

阈值化是一种广泛使用的图像分割手段,基于直方图阈值选取的图像分割主要研究直方图构造和阈值选取两个问题。基于直方图的阈值选取是非常依赖于直方图数据本身的,直方图数据的优劣影响阈值的精度,影响阈值选取算法的稳健性。除了在第 2 章中讲过的常规方法外,直方图的构造还有很多方法。在实际应用中,需要根据图像的不同特点使用合适的直方图构造方法。本节讲述二维直方图、边缘强度加权直方图和等量像素法直方图的构造方法。

5.3.1　二维直方图

在常规的直方图构造中,直方图数据只是考虑了像素自身的灰度值,没有考虑像素所处的空间位置。如果把一幅图像 A 的像素打乱空间位置后重新排列得到图像 B,尽管 A 图像和 B 图像已经发生了巨大的变化,但它们的直方图还是一模一样的。因此,提出了二维直方图的概念。

所谓二维直方图就是像素的灰度值 g 和其邻域均值 u 的联合分布直方图 $h(g,u)$,其含义是当像素的灰度值为 g 时,其邻域均值为 u 的概率。在灰度图像中,因为 g 和 u 都是 256 级灰度,所以二维直方图是一个 256×256 的矩阵。其构造方法如下:

【方法 5-1】　二维直方图的构造

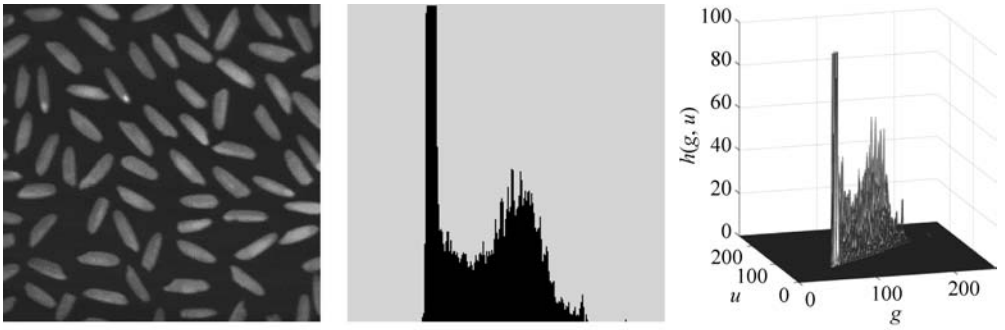
step. 1　定义直方图,int histogram[256][256];

step. 2　对原始图像 pGryImg 进行均值滤波，得到均值滤波后的图像 pAvrImg；

step. 3　初始化直方图，memset(histogram，0，sizeof(int) * 256 * 256)；

step. 4　统计直方图，for(i=0；i<width * height；i++) histogram[pGryImg[i]] [pAvrImg[i]]++；

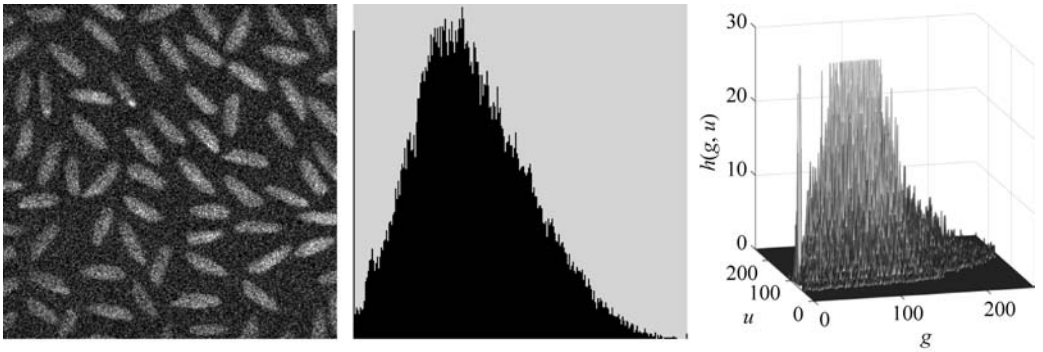
基于“图像中目标区域和背景区域内部的像素灰度值比较均匀”这个通常可以满足的假定，像素的灰度值 g 与其邻域的灰度均值 u 相差不大，所以像素集中在二维直方图的对角线附近。在偏离对角线的地方，直方图波峰的高度急剧下降。图 5-5(a)是图 5-3(a)消除了光照不均的影响后的结果，图 5-5(b)是它的常规一维直方图，图 5-5(c)是它的二维直方图，可以看出像素基本分布在对角线附近。



(a) 原始图像　(b) 常规一维直方图　(c) 二维直方图

图 5-5　正常图像的常规一维直方图和二维直方图

在常规一维直方图构造中，噪声的灰度值也被统计到了直方图中。分析可知，噪声的灰度值和其邻域的均值相差较大，它们的联合分布不会出现在二维直方图的对角线附近。因此，若仅使用二维直方图对角线附近的数据构造一个一维直方图，则该直方图能较好地消除噪声，使用该一维直方图进行图像分割会具有更好的抗噪能力。图 5-6(a)是加入了噪声的原始图像，图 5-6(b)是常规的一维直方图，由于噪声干扰，从中已经看不到如图 5-5(b)所示的波峰和波谷了，但在图 5-6(c)所示二维直方图的对角线附近仍能看到明显的峰谷，说明了其对噪声的抑制能力，对角线上数据分布的峰谷鲜明。



(a) 原始图像　(b) 常规一维直方图　(c) 二维直方图

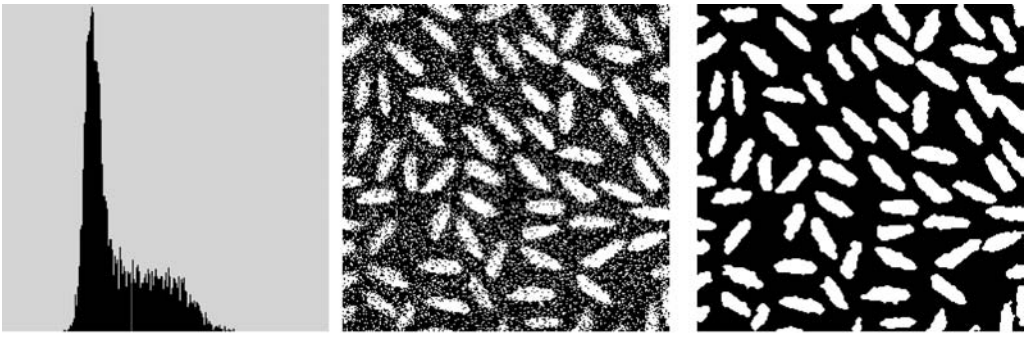
图 5-6　含噪图像的常规一维直方图和二维直方图

算法 5-3 给出了使用二维直方图对角线附近数据构造一维直方图的方法。使用该方法,令 dist=4,得到的一维直方图如图 5-7(a)所示,可以看到明显的峰谷;对该直方图求取 Otsu 阈值进行二值化,得到的图像分割结果如图 5-7(b)所示;采用中值滤波消除图 5-7(b)中噪声得到的结果如图 5-7(c)所示。

【算法 5-3】 使用二维直方图对角线附近数据构造一维直方图

```
void RmwHistogramBy2D( BYTE * pGryImg, BYTE * pAvrImg,
                       int width, int height,
                       int dist,                          //与均值的容差
                       int * histogram
                      )
{
    int * pos;
    int g, u;

    memset(histogram, 0, sizeof(int) * 256);
    for (int i = 0; i < width * height; i++)
    {
        u = pAvrImg[i];
        g = pGryImg[i];
        if (abs(g - u) <= dist)                           //仅统计在均值附近者
        {
            histogram[g]++;
        }
    }
    return;
}
```



(a) 对角线附近的一维直方图　(b) 图像分割结果　(c) 中值滤波后的结果

图 5-7　图像分割验证

图 5-7 说明了采用二维直方图对角线附近数据构造一维直方图的方法是有效的(但在实际应用中,噪声图像并不一定要如此处理,就图 5-6(a)而言,它也可以在图像平滑后直接进行图像分割)。

5.3.2　边缘强度加权直方图

算法 5-3 实质上是仅使用图像的部分像素建立直方图。图 5-6(a)中满足算法 5-3 直方图统计的像素如图 5-8 中黑点位置所示，这些像素约占整幅图像的 11%。它们的灰度值 g 与其邻域的灰度均值 u 基本相等，可以认为它们是背景区域或者目标区域的内部像素，而不是噪声像素或边缘点。

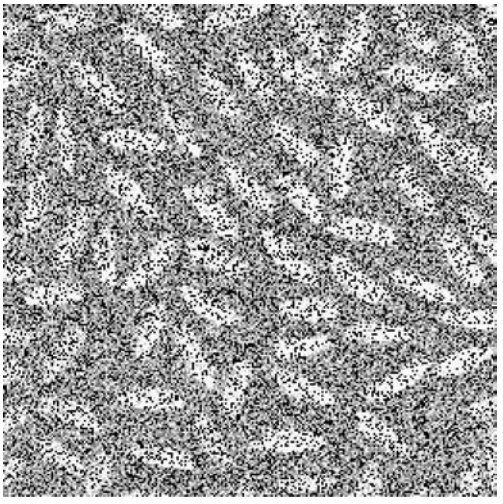


图 5-8　满足 $g \approx u$ 的像素

这就提出了使用哪些像素建立直方图或者像素对直方图的贡献度问题。算法 5-3 可以理解成仅使用 $g \approx u$ 的像素建立直方图；还可以理解成像素对直方图的贡献度问题，只不过这个贡献度非常武断，即要么为 0，要么为 1，因此可以设计一个贡献度函数，如：

$$\Delta = \frac{1}{\max\{|g-u|,1\}} \tag{5-11}$$

其中，$|g-u|$ 近似于边缘强度。

基于以上思想，下面讲述边缘强度反比加权和边缘强度正比加权的灰度直方图。

1. 边缘强度反比加权的灰度直方图

边缘强度反比加权直方图是指当像素的边缘强度大时，该像素灰度值对直方图的贡献就小，就是在构造直方图时把 histogram[$g(x,y)$]+=1 改为 histogram[$g(x,y)$]+=$f(\text{grd}(x,y))$，grd(x,y)代表(x,y)处的边缘强度，用 v 代表。grd(x,y)越大，$f(\text{grd}(x,y))$就越小，如：

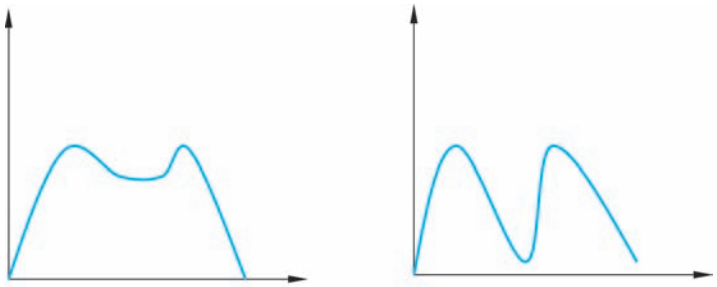
$$f(v)=\begin{cases}1.0, & v=0\\ \dfrac{1.0}{k\times v}, & \text{其他}\end{cases} \tag{5-12}$$

其中，k 为大于或等于 1 的正数。在极端情况下，甚至可以：

$$f(v)=\begin{cases}1, & v\approx 0\\ 0, & \text{其他}\end{cases} \tag{5-13}$$

采用式(5-13)时,实际上就是内部像素灰度直方图,即该直方图仅使用了目标和背景内部的像素,而忽略了目标和背景之间的过渡像素(因为过渡像素的边缘强度不等于0)。

边缘强度反比加权直方图在保持直方图的峰值不变的情况下,可以使得直方图的谷点更低,更有利于阈值的选取,从而提高阈值选取算法的稳健性。当图像模糊或者光照不均时,目标和背景之间存在大量的过渡像素,可以采用边缘强度反比加权直方图。图5-9(a)是一幅模糊图像的常规直方图,图5-9(b)是其边缘强度反比直方图,可以看出图5-9(b)中的谷点更低。



(a) 常规直方图　　(b) 边缘强度反比加权的直方图

图5-9　常规直方图与边缘强度反比加权的直方图比较

2. 边缘强度正比加权的灰度直方图

边缘强度正比加权直方图是当像素的边缘强度大时则该像素灰度值对直方图的贡献就大,即把 histogram[$g(x,y)$]+=1 改为 histogram[$g(x,y)$]+=f(grd(x,y)),grd(x,y)代表(x,y)处的边缘强度,用 v 代表。grd(x,y)越大,f(grd(x,y))就越大,如:

$$f(v)=k*v \tag{5-14}$$

其中,k 为大于或等于1的正数。在极端情况下,甚至可以:

$$f(v)=\begin{cases}1, & v\geqslant \text{threshold}\\ 0, & \text{其他}\end{cases} \tag{5-15}$$

其中,threshold 是边缘强度的阈值。式(5-15)实际上就是边缘像素灰度直方图,即该直方图仅使用了边缘像素,而忽略了目标和背景的内部像素(因为区域内部的像素边缘强度等于0)。

如果一幅图像中,在目标和背景之间的像素具有较大的边缘强度,而在其他位置的像素的边缘强度较小,则在边缘强度正比加权直方图中,目标和背景的灰度之间会出现一个波峰,此波峰所对应的灰度值即是阈值。当目标的占空比小时,若光照均匀,则可以采用边缘像素灰度直方图。图5-10(a)示意了一幅目标占空比很小的图像,其常规直方图如图5-10(b)所示,图5-10(c)是其边缘强度图像,图5-10(d)是其边缘强度正比加权的灰度直方图,其波峰对应的灰度值就是阈值。

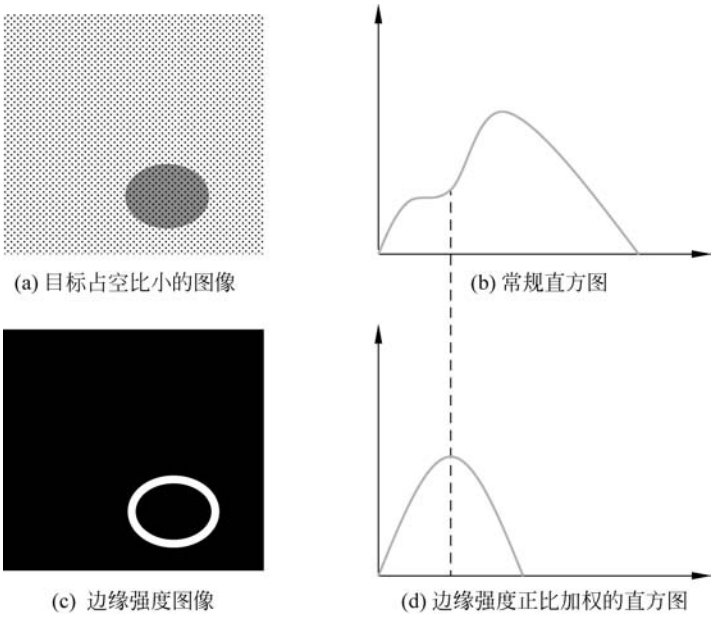


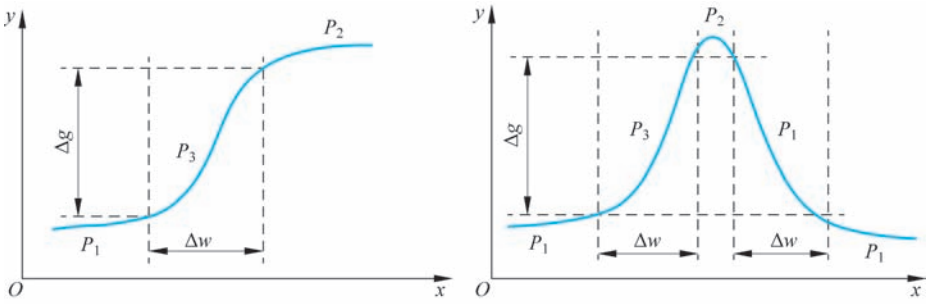
图 5-10　常规直方图与边缘强度正比加权的直方图比较

5.3.3　等量像素法直方图

目标和背景像素数目的不均会严重影响直方图的峰谷鲜明程度，降低直方图阈值选取算法的稳健性，尤其是当目标和背景的占空比相差很大时。当目标占空比很小时，即使在二维直方图对角线附近的数据中，也看不到明显的峰谷分布，目标像素被淹没在大量的背景像素中。边缘强度反比加权直方图上也因为位于目标内部的像素很少，而位于背景内部的像素很多，当背景的灰度分布范围较宽时，直方图上常常不存在对应于目标灰度的波峰，所以不适合目标占空比很小时的情况。受到边缘类型和边缘检测算子的影响，在目标占空比很小时，边缘点就更少，而且光照不均时边缘强度的离散性较大，边缘强度正比加权直方图也很难形成显著的波峰。本书作者在 2001 年提出了等量像素构造直方图的方法(一种基于图像边缘模式的直方图构造新方法[J]. 计算机研究与发展，2001，38(8)：972-976)，较好地解决了这个问题。

根据图像过渡区理论，目标(设为 P_2)和背景(设为 P_1)之间肯定是存在边界过渡区的，则边界过渡区(设为 P_3)的灰度值变化如图 5-11(a)或图 5-11(b)所示，像素的灰度值不可能从背景灰度值直接变化到目标灰度值。设图像中目标和背景的灰度差为 Δg，目标和背景区域的边界宽度为 Δw，则边界上的像素肯定满足其宽度为 Δw 的邻域内的最大灰度和最小灰度之差大于 Δg，由此可以得到边界像素的集合为 φ。将 φ 中分别在水平方向和垂直方向连续的像素表示为水平线段和垂直线段形式，从而在距离线段中点 $\Delta w/2$ 的两侧选取连续的满足灰度之差大于 Δg 的像素对(P_2 和 P_1)来构造直方图，如图 5-11 所示。

极少数情况下，由于区域边界的多样性，图像中某些位置的边界宽度可能小于 Δw 或者目标尺寸小于 Δw，按上述方法则找不到目标像素和对应的背景像素。此时，取线段中点(P_2)和位于线段一侧(左侧或右侧)且与线段中点的灰度之差大于 Δg 的像素点(P_1)作为构造直方图的像素对，如图 5-11(b)所示。

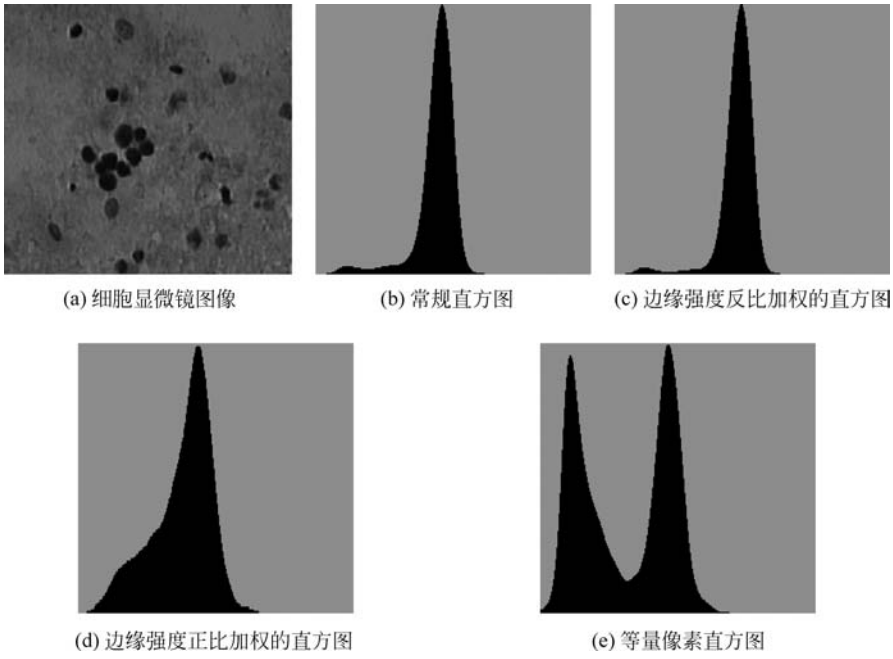


(a) 阶跃边缘的过渡区　　(b) 屋顶状边缘的过渡区

图 5-11　边界过渡区

该种直方图实质上是从背景和目标上选择了等量的像素来构造直方图(有些像素会被重复选用多次),在一定程度上能够解决占空比不均的问题。其建立需要 Δw 和 Δg 2 个参数。Δg 是目标和背景的灰度差,直观易用。Δw 是边界区域的宽度,即使大于或小于真实的边界宽度,也不影响直方图的效果。该种直方图虽然由 Δw 和 Δg 得到,但并不依赖于它们的精度,因此具有很好的通用性和实用性。

图 5-12(a)是一幅细胞显微镜图像。图 5-12(b)是其常规灰度直方图,由于细胞的占空比很小,因此峰谷不够明显,与细胞对应的波峰非常矮。图 5-12(c)是边缘强度反比加权的直方图,相当于内部像素直方图,同样因为细胞占空比小的原因,其峰谷不够明显,但比图 5-12(b)略有改善。图 5-12(d)是边缘强度正比加权的直方图,相当于边缘像素灰度直方图,由于背景的灰度不均匀,其最大波峰对应的灰度值是背景的灰度均值,不是细胞和背景的分割阈值。图 5-12(e)是等量像素灰度直方图,可以看到峰谷鲜明,且基本上使用了相同数量的细胞像素和背景像素。



(a) 细胞显微镜图像　(b) 常规直方图　(c) 边缘强度反比加权的直方图

(d) 边缘强度正比加权的直方图　(e) 等量像素直方图

图 5-12　等量像素法直方图的比较

5.4　聚类分割

如果把目标和背景的分割看作是像素的分类问题，即把一部分像素分类成目标类别，一部分像素分类成背景类别，那么就可以采用模式识别中的聚类方法来实现图像分割。模式识别有众多的聚类方法，其中 k-均值聚类(k-means clustering)是一个简单实用的方法。

k-均值聚类分割的基本思想是设定类别数 k 的值，比如 $k=2$ 或者 $k=3$ 等；给每个类别初始化一个类别中心 u_i，$i=0,1,\cdots,k-1$，即对每个中心赋一个不同的灰度值；然后根据像素灰度值到类别中心的距离最小原则，把像素分到各个类别；全部像素分类完成后，计算每类像素的灰度值均值，作为各类别新的中心值。如此迭代执行，直到各类别的中心值不变为止。

例 5-1 示例了 8 个数据的聚类过程，设 $k=2$，$u_0=2$，$u_1=3$，执行了 3 次迭代后，各类别的中心值不再变化，聚类结束。

例 5-1：k-均值聚类

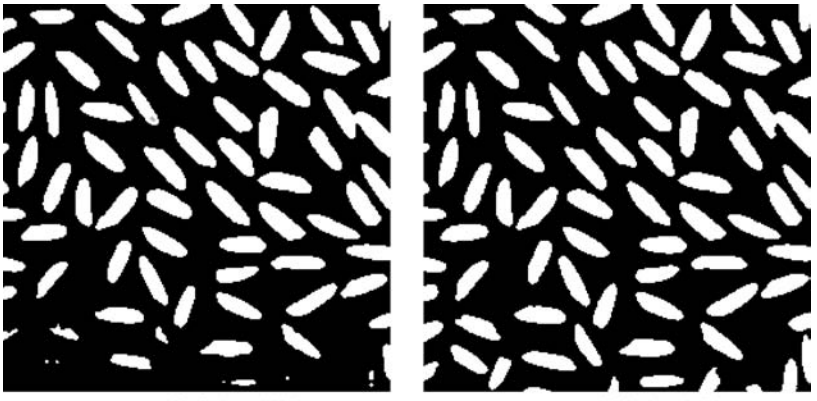
数据：　2　2　3　3　7　6　7　9

第 1 次迭代：　2　2　3　3　7　6　7　9　得到 $u_0=4/2$，$u_1=35/6$

第 2 次迭代：　2　2　3　3　7　6　7　9　得到 $u_0=10/4$，$u_1=29/4$

第 3 次迭代：　2　2　3　3　7　6　7　9　得到 $u_0=10/4$，$u_1=29/4$

由于在聚类过程中只是使用了像素的灰度值，没有使用像素的空间位置信息，因此在具体实现时应该先统计灰度直方图，借助直方图能够提高聚类的速度。另外，因为类别的中心值是浮点数，如果前后 2 次迭代的中心值最大差异足够小，即可判定为类别的中心值不变。图 5-13(a)是对存在光照不均的图 5-3(a)的聚类分割结果，图 5-13(b)是消除了光照不均的图 5-5(a)的聚类分割结果，它们的图像分割效果与 Otsu 阈值化非常接近。



(a) 图5-3(a)的聚类分割结果　　(b) 图5-5(a)的聚类分割结果

图 5-13　k-均值聚类分割($k=2$)

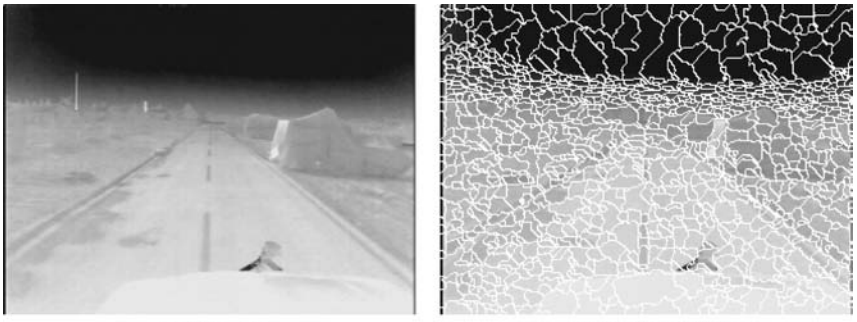
聚类分割所需要的迭代次数依赖于类别中心的初始值。初始值选择得当,有可能很快就收敛;选择不当,有可能需要特别长的时间才收敛。在实际应用中,由于聚类分割的时间花费难以预知,有可能多帧图像的平均聚类时间花费很小,但偶尔一帧图像的聚类时间特别长。由于时间花费不是常数,因此聚类分割在实时图像处理中一般不被采用,多被用在人机交互式处理中,当用户观察到分割结果基本满意时,就进行人工干预,强制迭代终止。

5.5 区域增长与分裂合并算法

聚类分割方法需要事先设定类别数的值,但在很多情况下,类别数是难以预先确定的。因此可以从目标和背景的性质出发,属于同一类型区域的像素应该具有相近的灰度值或其他性质(彩色、纹理、梯度等),这类方法主要有区域增长算法、分水岭算法、区域分裂合并算法等。

区域增长算法(region growing)是随机或者以人工交互的方式给出一个或一组种子(seeds),比如计算机自动选择在一定范围内最亮或最暗的点,或者是人工指定位于目标中心的点;按照某种增长规则,将邻居中与种子像素有相近灰度值或相似性质的像素合并到种子像素所在的区域中;然后将这些像素当作新的种子点重复以上过程,直到没有满足增长规则的新像素被合并进来为止,结束区域增长。

图 5-14(a)是一幅红外图像,图 5-14(b)是采用分水岭区域增长算法(A fast watershed algorithm based on chain code and its application in image segmentation[J]. Patter Recognition Letters, 2005, 26: 1266-1274)得到的结果。分水岭区域增长算法把局域内灰度值最小的像素作为种子。



(a) 原始图像　　(b) 区域增长结果

图 5-14　区域增长算法(分水岭)

区域分裂合并算法(split-and-merge)也是利用了这一原理(同一类型区域的像素具有相近的灰度值或其他性质)进行图像分割。它与区域增长算法有相似之处,但它无须预先指定种子点,而是按某种一致性准则对图像进行区域分裂或者合并。比如可以先把图像分成 4 块,若这其中的一块符合分裂条件,那么这一块又分裂成 4 块,就这样一直分

裂，这个过程被称为区域分裂；当分裂出一定数量的区域时，以每块为中心，检查与之相邻的各块，若满足一定的合并条件就将它们进行合并，这个过程称为区域合并；如此往复，直到再也没有一定大小的新块产生，最后把小块按照合并条件合并到其相邻的大块里。

5.6　基于某种稳定性的图像分割

具有不同灰度的多类目标分割一直是传统图像分割的难题。近些年来随着机器学习的发展，对于已知场景，语义分割能够较好地解决这个问题。但是在很多应用和未知场景中，传统的图像分割方法仍然具有其独特优势。对于多类不同灰度的目标分割，显然无法通过使用单个阈值来实现。本节讲述对图像执行序贯二值化(consecutive thresholding)并在二值化过程中基于提取的某种稳定性来实现图像分割的方法。

基于某种稳定性的图像分割的出发点是假定目标会在一定的灰度范围内稳定存在；其手段是采用序贯二值化得到若干个二值图像，并从一系列二值图像中寻找某种稳定的特征。

5.6.1　基于目标个数稳定性的图像分割

在分割灰度值不同的多类目标时，如果目标和背景有明显的灰度差异，并且目标的尺寸不是特别小，就可以使用基于目标个数稳定性的图像分割。

基于目标个数稳定性的图像分割是把一个灰度区间内的每个灰度值作为阈值对图像执行二值化，得到一系列的二值图像，统计每个二值图像中得到的目标个数，从而构造出阈值和目标个数的直方图。在该直方图中会存在若干个目标个数不变的阈值区间，一般认为有几个目标个数不变的阈值区间，有几个阀值区间，就有几类目标，且每个阈值区间的中心值就是此类目标的分割阈值。

图 5-15(a)是一种包含字符和数字号码的图像，由 10 个字符组成，前 4 个字符是红色的，后 6 个字符是黑色的，变成灰度图像后就是前 4 个字符的灰度值比后 6 个字符的灰度值大。图 5-15(b)是它的直方图，从中能够较明显看到有 3 个波峰，从小到大分别代表后 6 个黑色字符、前 4 个红色字符和白色背景。图 5-15(c)是它的 Otsu 阈值比分割结果(阈值＝149)，从中可以看出对第 1 个和第 2 个字符的分割是不准确的，对第 3 个和第 4 个字符的分割结果是错误的。由于该图像的灰度最小值为 68，灰度最大值为 217，因此使用阈值从 68 到 217 对它进行序贯二值化，得到的字符个数直方图如图 5-15(d)所示。在图 5-15(d)中看到有 2 个字符个数稳定的阈值区间，当阈值在[84，146]变化时字符个数一直为 6，当阈值在[153，194]变化时字符个数一直为 10，因此分别使用阈值 $t_1=\frac{84+146}{2}=115$ 和 $t_2=\frac{153+194}{2}=173$ 进行二值化，得到的图像分割结果分别如图 5-15(e)和图 5-15(f)所示。

(a) 原始图像　(b) 直方图

(c) Otsu阈值化(阈值为149)　(d) 字符个数直方图

(e) 阈值为115　(f) 阈值为173

图 5-15　基于目标个数稳定性的图像分割

5.6.2　基于次数关系稳定性的图像分割

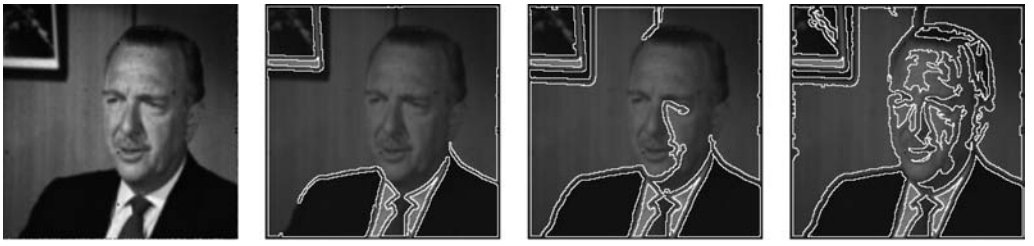
基于目标能在一个较宽的阈值范围内存在的事实,本书作者在 2003 年提出了一种基于次数关系稳定性的图像分割方法(Split-and-Merge segmentation using relation stable-state, Optical Engineering[J]. 2003, 42(8): 2362-2367),在 2008 年提出了目标多表象的概念(国家自然科学基金课题"基于目标区域演化分割的图像描述研究",2008—2010,编号为 60875010)。

次数关系稳定性是指采用不同的阈值对图像进行二值化时,目标区域会得到若干条轮廓线,这些轮廓或大或小,或重叠或相邻,但是处在区域边界上的像素被有效轮廓经过的次数一定大于区域内部像素被经过的次数,这种次数大于关系在光照均匀和非均匀的图像中能够保持不变。

目标的多表象是指同一个目标区域在不同的分割方法或者阈值下有着不同的边界,目标的尺寸及形状是变化的;在某种分割方法或者阈值下的分割结果可能会是多个目标粘连在一起,而在另一种方法或者阈值下则可能会是一个目标分成了几个小的部分,故将同一个目标在不同的分割阈值或分割方式下的不同表现形式称为目标的多表象。

基于次数关系稳定性的图像分割方法首先用一系列不同的阈值对图像进行二值化,把区域面积大于 t_s 的各条轮廓线累加到一个像素初值为零的图像中,称该图像为轮廓累积图像;其次,对轮廓累积图像进行序贯阈值分割,即用一个区间内的值逐一作为阈值 t_e,将轮廓累积图像分割为像素值为 0 的区域(小于阈值的像素)和像素值为 255 的区域(大于或等于阈值的像素),像素值为 0 的区域就是目标的一个表象。当 t_e 按由大到小顺序采用时,会得到一个个的二值图像,这些二值图像中位置基本相同的像素值为 0 的区域就构成了目标的多个表象。随着 t_e 越来越小,区域的面积也越来越小,区域对应的边缘强度也越来越小,区域的个数也越来越多。t_e 与边缘强度等效。图 5-16(a)是原始图像,图 5-16(b)、图 5-16(c)、图 5-16(d)是用轮廓线表示的分割结果,分别对应 $t_e=18$,$t_e=8$

和 $t_e=4$。



(a) 原始图像　(b) t_e=18　(c) t_e=8　(d) t_e=4

图 5-16　Man 图像的基于次数关系稳定性的图像分割

图 5-17 和图 5-18 的图像尺寸都是 512×512，都使用 $t_s=50$。图 5-17(b)和图 5-18(b)是使用 Otsu 阈值化的结果，图 5-17(c)和图 5-18(c)分别是使用 $t_e=3$ 和 $t_e=4$ 的轮廓累积图像阈值化结果，图 5-17(d)和图 5-18(d)是目标分割结果。

(a) 原始图像　(b) Otsu阈值化　(c) 二值化轮廓累积图像　(d) 分割结果

图 5-17　Desk 图像的基于次数关系稳定性的图像分割

(a) 原始图像　(b) Otsu阈值化　(c) 二值化轮廓累积图像　(d) 分割结果

图 5-18　Ship 图像的基于次数关系稳定性的图像分割

该方法的优点是通过对原始图像的序贯二值化，在分割结果中使用面积约束 t_s 筛选轮廓，消除了噪声，轮廓累积形成的图像又为不同灰度值的目标筑起了隔离带，光照不均时隔离带的宽度会变大但数目不变；通过对轮廓累积图像的序贯二值化，可以得到目标的多表象；在单次分割时，只需要设定参数 t_e 的值，就可以得到边缘强度大于 t_e 的各个目标，与目标的灰度值无关。

该方法较巧妙地结合了灰度特征和边缘特征，且仅需要目标的面积参数 t_s 和等效边缘强度参数 t_e 共计两个语义明确的直观参数，对灰度值不同的多类目标分割结果良好，具有较好的通用性和实用性。

5.7 光照不均的消除与图像分割

图像中存在光照不均是一个普遍的问题。目标像素和背景像素的光照不均来源非常多,有的是像素到光源距离的远近不同导致的,比如在灯下就亮,远离灯光就暗;有的是光源本身发光不均匀导致的;有的是不同位置的像素代表的物理尺寸大小不同导致的,比如近处暗远处亮,这是因为远处场景中很大的一个区域才能变成一个像素,这个大区域的光线反射量累积成了一个像素的灰度值;有的是因为目标本身是立体的,有着不同的表面,会产生不同方向的反射光,当目标是球体时还会产生镜面反射;有的是强逆光或者其他的强光干扰;有的是建筑物的遮挡、树木的阴影、云朵的影子、地面的水渍。即使图像是来自白纸黑字的纸面,也可能存在光照不均的情况,比如图 5-4(a)。

但是,光照不均严重影响图像的阈值化。当光照不均时,A 区域目标像素的灰度值可能比 B 区域背景像素的灰度值小,却比 C 区域背景像素的灰度值大。分块阈值或者自适应阈值也只是能够解决某些光照不均图像的分割,比如能够较好地分割图 5-3(a),但无法正确分割图 5-4(a)。

根据目标像素和背景像素的灰度值特点,估计原始图像的光照,然后在原始图像中减去光照图像,是解决图像中光照不均问题的有效方法。下面通过举例,来讲述使用高斯平滑的光照估计方法和极值滤波结合高斯平滑的光照估计方法。

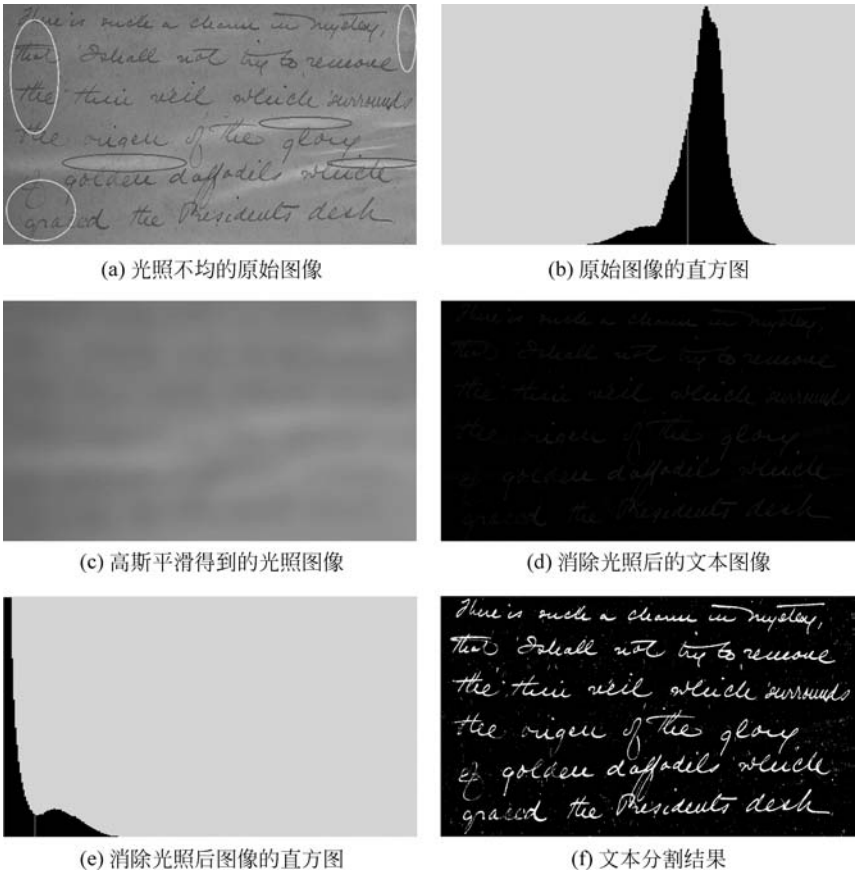
5.7.1 文本图像分割

在原始图像图 5-4(a)中,既有图 5-19(a)中横圈位置所示的亮光区,也有图 5-19(a)中竖圈位置所示的暗光区,也有正常的光照区域。考虑到字符笔画稀疏,此时若采用大尺度的高斯平滑,则平滑后的像素灰度值会基本上不受字符的影响,可以认为平滑后的像素灰度值就代表此处的光照。对图 5-4(a)进行高斯平滑后,得到了光照图像 L,如图 5-19(c)所示。考虑到文字是黑色的,背景是白色的,采用式(5-16)得到消除光照的图像,如图 5-19(d)所示。

$$G(x,y)=\begin{cases}-(g(x,y)-L(x,y)), & g(x,y)<L(x,y)\\ 0, & \text{其他}\end{cases} \tag{5-16}$$

图 5-19(d)的直方图如图 5-19(e)所示,对图 5-19(d)采用 Otsu 阈值化的分割结果如图 5-19(f)所示。在图 5-19(b)所示的直方图中,由于光照不均,存在大量的过渡像素,因此直方图中没有明显的波谷,竖线所示位置是其得到的 Otsu 阈值。但是,在图 5-19(e)所示的消除光照后的字符图像直方图中峰谷鲜明,竖线所示位置是其得到的 Otsu 阈值。

可以看出,图 5-19(f)所示的分割结果明显优于图 5-4(b)所示的全局阈值分割结果、图 5-4(c)所示的局部阈值分割结果和图 5-4(d)所示的自适应阈值分割结果。



(a) 光照不均的原始图像　(b) 原始图像的直方图

(c) 高斯平滑得到的光照图像　(d) 消除光照后的文本图像

(e) 消除光照后图像的直方图　(f) 文本分割结果

图 5-19　光照不均的文本图像分割

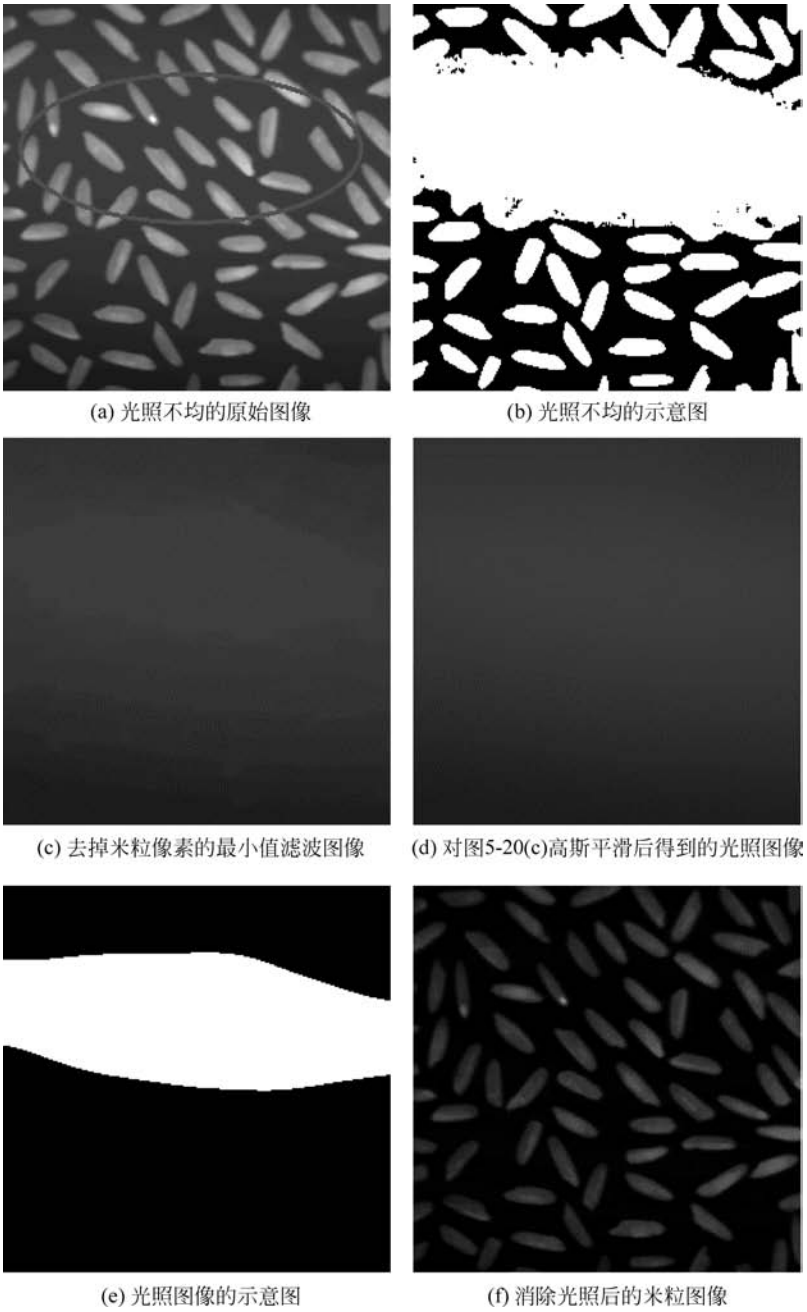
5.7.2　颗粒图像分割

第 4 章和本章都用到了米粒图像。图 4-31(a)和图 5-3(a)是原始的米粒图像，图 5-1(a)和图 5-5(a)是消除了光照不均的米粒图像，图 5-6(a)是图 5-5(a)加了噪声后的米粒图像。原始的米粒图像是存在光照不均的，图 5-20(a)的横圈位置示意了光照最强的区域，为了显示其光照不均，对它做 threshold＝91 的二值化，得到了如图 5-20(b)所示的二值图像。从图 5-20(b)中明显看到了光照不均现象和光源的位置与形状。

考虑到在米粒图像中，背景是黑色的，作为目标的米粒是白色的，而且米粒不像字符那样笔画稀疏，因此采用了最小值滤波来去掉米粒，得到了如图 5-20(c)所示的光照图像。为了消除图 5-20(c)中的层次感，采用高斯平滑得到了图 5-20(d)所示的光照图像 L。为了验证光照图像的正确性，对它做 threshold＝91 的二值化，得到了如图 5-20(e)所示的二值图像，对比图 5-20(b)，可以看到其对光照估计是正确的。采用式(5-17)得到了消除光照的米粒图像，如图 5-20(f)所示。

$$G(x,y)=\begin{cases} g(x,y)-L(x,y), & g(x,y)>L(x,y) \\ 0, & \text{其他} \end{cases} \tag{5-17}$$

图 5-20(f)的直方图如图 5-20(g)所示,竖线所示是其得到的 Otsu 阈值。对图 5-20(f)采用 Otsu 阈值化的分割结果如图 5-20(h)所示,可以看出分割结果明显优于图 5-3(b)所示的全局阈值分割结果。

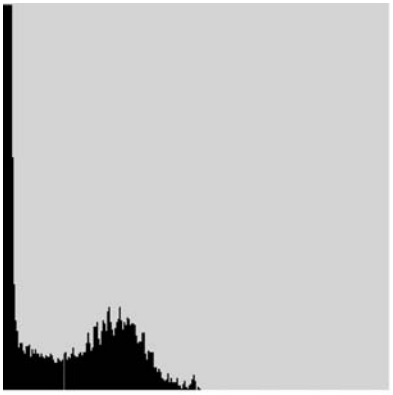


(a) 光照不均的原始图像　(b) 光照不均的示意图

(c) 去掉米粒像素的最小值滤波图像　(d) 对图5-20(c)高斯平滑后得到的光照图像

(e) 光照图像的示意图　(f) 消除光照后的米粒图像

图 5-20　光照不均的米粒图像分割



(g) 米粒图像的直方图

(h) 米粒分割结果

图 5-20(续)

5.8　本章小结

场景未知和参数适配是影响图像分割算法通用性的两个主要原因。场景未知常表现为：背景不是单一的，被检测目标分布在复杂的背景中；因目标距离摄像机的远近不同，同一目标像素的灰度值随其位置不同而不同；需要分割灰度值不同的多类目标，但不知道目标的种类数；目标仅在图像中占很小的比例而且和背景的灰度差异不大。参数适配的难度常表现为：需要预先设定的参数太多，参数的语义性差、不直观，参数对图像亮度和对比度有较强的依赖等。

图像分割是图像分析和计算机视觉中非常困难的问题，本章给出了一些方法和思路，希望它们能对解决实际问题提供一定的启发。近些年来随着机器学习的发展，对于已知场景，语义分割能够较好地解决一些问题。

作业与思考

5.1　在什么情况下，阈值可以取为直方图中两个最大波峰对应灰度值的均值？

5.2　用 C/C++语言编程实现 H0501Gry.bmp(见图 5-21)的直方图求取、Otsu 阈值的求取和二值化。

5.3　设题 5.2 中得到的直方图为 H_1，得到的阈值为 t_1。用 C/C++语言编程实现对 H_1 进行相邻 5 个数据的平滑，得到 H_2，使用 H_2 求取 Otsu 阈值，得到 t_2。t_2 约等于 t_1 吗？

5.4　设题 5.2 中得到的直方图为 H_1，得到的阈值为 t_1。用 C/C++语言编程实现将 H0501Gry.bmp 中的灰度值除以 4 后统计直方图，得到 H_2，使用 H_2 求取 Otsu 阈值，得到 t_2。t_2 约等于 $t_1/4$ 吗？

5.5　平板扫描器扫描了一张 A4 纸(210mm×297mm)的图像，该纸上有若干炭笔

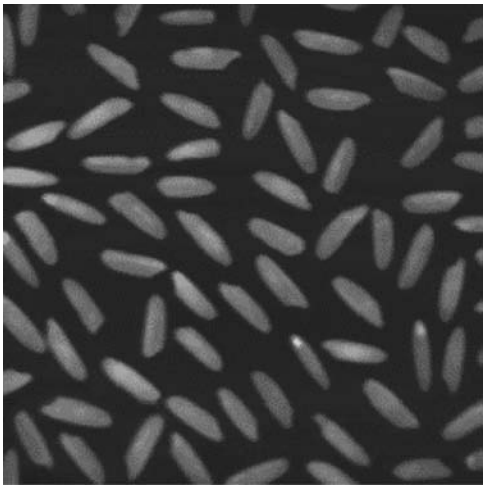


图 5-21 H0501Gry.bmp

涂鸦的线条,该图像的 32 个灰度级的直方图数据如下,涂鸦的总面积是多少平方毫米?

[0 0 278 1791 369 356 415 463 562 421 373 372 374 286 298 357 344 313 0 0 365 414 435 453 541 729 1394 15662 27054 7462 489 0]

5.6 在射击训练的视频靶中,摄像机通过拍摄靶子的图像来判定弹着点情况。由于靶纸在背板上张贴不平,在室外光线的照射下,靶子图像的背景并不均匀,很难通过全局图像分割来可靠地找出弹孔;还会存在两发子弹形成的弹孔基本重叠的情况。下面是靶子图像的同一个局部区域在连续 4 次拍摄中的 32 个灰度级直方图数据,试判定每次拍摄中是否有新子弹射中该区域。假设灰度值越低越是弹孔。

第 1 帧

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 157 439 346 52 20 2]

第 2 帧

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 173 579 233 31 0 0 0 0 0]

第 3 帧

[0 0 0 19 4 1 8 3 4 2 0 2 1 4 3 2 11 0 0 0 30 119 360 293 98 36 12 6 4 2 0 0]

第 4 帧

[0 0 0 5 15 24 19 9 16 19 14 22 52 113 309 207 123 29 10 16 9 6 1 0 0 0 3 2 0 1 0 0]

5.7 二维直方图、边缘强度反比加权直方图、边缘强度正比加权直方图、等量像素法直方图各有什么特点?各自适合在什么情况下使用?在什么情况下不宜使用?

5.8 用 C/C++语言编程实现 H0502Gry.bmp(见图 5-22)的环线和弹孔的分割,要求采用光照估计+图像减法+Otsu 阈值法、多次分割法、聚类分割法 3 类方法,并对这 3 类方法的效果进行比较和分析。

5.9 用 C/C++语言编程实现 H0501Gry.bmp(见图 5-21)的 k-均值聚类,$k=2$;实现 H0502Gry.bmp(见图 5-22)的 k-均值聚类,$k=3$ 和 $k=8$;给出聚类分割后的图像,并进行分析。

图 5-22　H0502Gry. bmp