第3章 典型分割算法

图像分割的技术一直在不断发展，已有很多种方法提了出来。本章仅介绍几个具有比较特殊思路的典型方法。在2.1节中曾将图像分割技术分为四大类，本章以下各节介绍的内容也分别对应这四大类分割技术。

根据上述的讨论，本章各节将安排如下。

3.1节先介绍SUSAN算子的基础——USAN原理，再讨论既可用于边缘检测也可用于角点检测的SUSAN算子的具体方法和特点，它属于并行边界类技术。

3.2节介绍一种结合了图像整体信息和局部信息，并利用了图结构的特殊方法——图割方法，它属于串行边界类技术。

3.3节介绍3种不同的属于并行区域类技术的方法，分别是借助小波变换的多分辨率特性的阈值化方法、借助过渡区的阈值化方法和借助均移确定聚类的分割方法。

3.4节介绍一种结合了图像整体信息和局部信息的特殊方法——基于分水岭的方法，除了基本原理还讨论了对基本算法的改进和扩展。虽然分水岭本身对应边界，但这种方法更多是利用边界所包围的区域性质来工作的。

3.1 SUSAN检测算子

如果一个像素在其小邻域中具有两个明显不同的边缘方向则常可看作一个角点，也有人将局部曲率较大的边缘点看做**角点**。典型的角点检测器仍多基于像素灰度差，例如Harris角点检测器就通过计算像素邻域中灰度平方差的和来检测角点[Sonka 2008]。

SUSAN算子是一种很有特色的检测算子[Smith 1997]，它既可以检测边缘点，也可以检测角点。

3.1.1 USAN原理

先借助图3.1.1来解释检测的原理。这里设有一幅长方形的图像，上部为亮区域，下部为暗区域，分别代表目标和背景。现在考虑有一个圆形的模板(其大小由模板边界所限定)，其中心称为“核”，用“+”表示。图中给出该模板放在图中6个不同位置的示意情况，从左边数过去，第1个模板全部在亮区域，第2个模板大部在亮区域，第3个模板一半在亮区域，第

4 个模板大部在暗区域，第 5 个模板全部在暗区域，第 6 个模板的 1/4 在暗区域。

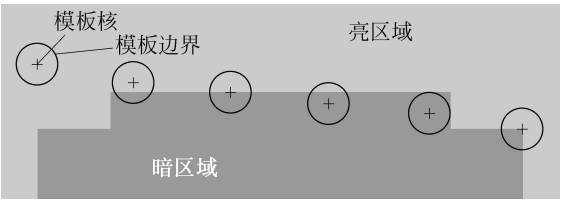

图 3.1.1　圆形模板在图像中的不同位置

如果将模板中各个像素的灰度都与模板中心核像素的灰度进行比较，那么就会发现总有一部分模板像素的灰度与核像素的灰度相同或相似。这部分区域可称为 **USAN** 区域（即与核同值的区域，简称核同值区）。USAN 区域包含了很多与图像结构有关的信息。利用这种区域的尺寸、重心、二阶矩等可以帮助检测图像中的边缘和角点。从图 3.1.1 可见，当核像素处在图像中的灰度一致区域时，USAN 的面积会达到最大（超过一半），第 1 个模板和第 5 个模板就属于这种情况。这个面积当核处在直边缘处约为最大值的一半，第 3 个模板就属于这种情况。这个面积当核处在角点位置时更小，约为最大值的 1/4，第 6 个模板就属于这种情况。

利用 USAN 面积的上述变化性质可检测边缘或角点。具体说来，USAN 面积较大（超过一半）时表明核像素处在图像中的灰度一致区域，在模板核接近边缘时减少为一半，而在接近角点时减少得更多，即在角点处取得最小值（约 1/4）。如果将 USAN 面积的倒数作为检测的输出，则可以通过计算极大值方便地确定角点的位置。

由上讨论可见，使用 USAN 面积作为特征可起到增强边缘和角点的效果。基于这种模板的检测方式与其他常用的检测方式有许多不同之处，最明显的就是不需要计算微分因而对噪声不很敏感。

3.1.2　SUSAN 算子边缘检测

在 USAN 区域的基础上可讨论 SUSAN 算子。

1. 边缘检测

SUSAN 算子采用圆形模板来得到各向同性的响应。在数字图像中，圆可用一个含有 37 个像素的模板来近似，见图 3.1.2。这 37 个像素排成 7 行，分别有 3,5,7,7,7,5,3 个像素。这相当于一个半径为 3.4 个像素的圆。如考虑到计算量，也有用普通的 3×3 模板来粗略近似的。

设模板用 $N(x,y)$ 表示，将其依次放在图像中每个点的位置，在每个位置，将模板内每个像素的灰度值与核的灰度值进行比较：

$$C(x_0,y_0;\ x,y)=\begin{cases}1 & |\ f(x_0,y_0)-f(x,y)\ |\leqslant T\\ 0 & |\ f(x_0,y_0)-f(x,y)\ |>T\end{cases} \tag{3.1.1}$$

式中，(x_0, y_0)是核在图像中的位置坐标；(x, y)是模板 $N(x, y)$中其他位置；$f(x_0, y_0)$和 $f(x, y)$分别是在(x_0, y_0)和(x, y)处像素的灰度；T 是一个灰度差的阈值；函数 $C(\cdot\,;\,\cdot)$ 代表输出的比较结果。一般当检测到角点时，该函数如图 3.1.3 所示，其中这里阈值 T 取为 27(见下)。

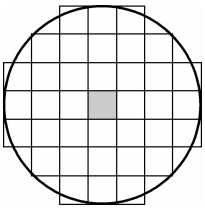

图 3.1.2　37 个像素的模板

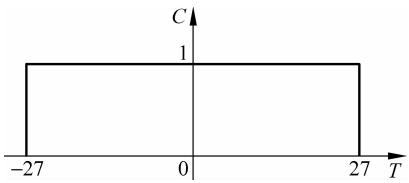

图 3.1.3　函数 $C(\cdot\,;\,\cdot)$示例

上述比较要对模板中的每个像素进行，由此可得到一个输出：

$$S(x_0, y_0) = \sum_{(x,y)\in N(x_0,y_0)} C(x_0, y_0;\, x, y) \tag{3.1.2}$$

这个总和其实就是 USAN 区域中的像素个数，或者说它给出了 USAN 区域的面积。由上讨论可见，这个面积在角点处会达到最小。结合式(3.1.1)和式(3.1.2)可知，阈值 T 既可用来帮助检测 USAN 区域面积的最小值，也可以确定可消除的噪声的最大值。

实际应用 SUSAN 算子时，需要将游程和 S 与一个固定的几何阈值 G 进行比较以做出判断。该阈值设为 $3S_{\max}/4$，其中 $S_{\max}$是 S 所能取得的最大值(对 37 个像素的模板，最大值为 36)。初始的边缘响应 $R(x_0, y_0)$根据下式得到

$$R(x_0, y_0) = \begin{cases} G - S(x_0, y_0) & S(x_0, y_0) < G \\ 0 & \text{其他} \end{cases} \tag{3.1.3}$$

上式是根据 USAN 原理获得的，即 USAN 的面积越小，边缘的响应就越大。

当图像中没有噪声时，完全不需要几何阈值。但当图像中有噪声时，将阈值 G 设为 $3S_{\max}/4$ 可给出最优的噪声消除性能。考虑一个阶跃边缘，S 的值总会在某一边小于 $S_{\max}/2$。如果边缘是弯曲的，小于 $S_{\max}/2$ 的 S 值会出现在凹的一边。如果边缘不是理想的阶跃边缘(有坡度)，S 的值会更小，这样边缘检测不到的可能性更小。

上面介绍的方法一般已可以给出相当好的结果，但一个更稳定的计算 $C(\cdot\,;\,\cdot)$的公式是

$$C(x_0, y_0;\, x, y) = \exp\left\{-\left[\frac{f(x_0, y_0) - f(x, y)}{T}\right]^2\right\} \tag{3.1.4}$$

这个公式对应的曲线是图 3.1.4 中的曲线 b(曲线 a 对应式(3.1.1))。可见，式(3.1.4)给出式(3.1.1)的一个平滑的版本。它允许像素的灰度有一定的变化而不会对 $C(\cdot\,;\,\cdot)$造成太大的影响。

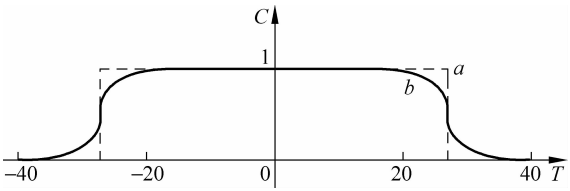

图 3.1.4　不同的函数 $C(\cdot;\cdot)$示例

例 3.1.1　角点检测示例

图 3.1.5 给出两个用 SUSAN 算子检测图像中角点得到的结果。

图 3.1.5　用 SUSAN 算子检测到的角点 ❑

2. 边缘方向的检测

在很多情况下，不仅需要考虑边缘的强度也需要考虑边缘的方向。首先，如果要除去非最大边缘值，就必须要借助边缘的方向。另外，如果需要确定边缘的位置到亚像素精度，也常需要利用边缘方向的信息。最后，许多应用中常把边缘的位置、强度和方向结合使用。对大多数现有的边缘检测算子，边缘方向是借助边缘强度来确定的。根据 USAN 的原理确定边缘方向需根据边缘点的种类采用不同的方法。

根据 USAN 区域的特点，可将边缘点分成两类。参见图 3.1.6，其中左图给出图像中一个典型的直线边缘。图中方格里的阴影用来近似地区分具有不同灰度的像素。对三个感兴趣点的局部 USAN 区域分别显示在右边的 3 个 3×3 模板中。其中“○”代表 USAN 区域的重心，而“＋”代表模板的核。区域 A 和 B 中都是同一类的边缘点，边缘都通过 USAN 区域的重心，只是模板核分别落在边缘的两边。区域 C 中是另一类边缘点，这里模板核与 USAN 区域的重心重合。

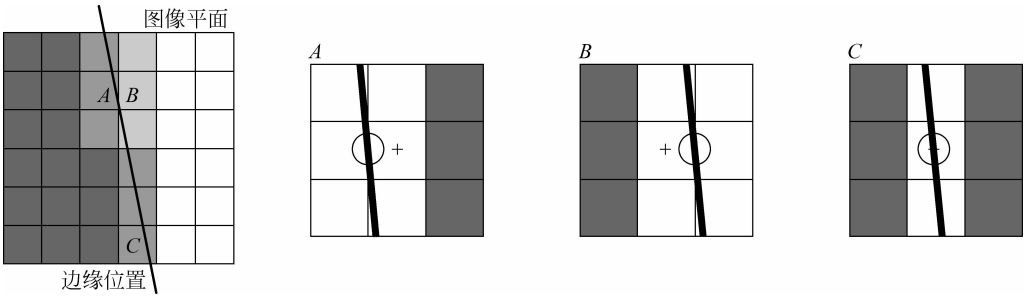

图 3.1.6　主要的边缘种类示意

在区域 A 和区域 B 中，边缘落在像素之间，从 USAN 区域的重心到模板核的矢量与边缘的局部方向（几乎）垂直。在区域 C 中，边缘通过像素的中心而不是像素之间，且边缘两边有较高的反差。这种情况对应像素内部边缘。在这种情况下，所获得的 USAN 区域是沿边缘方向的细条，如图 3.1.6 所示。通过寻找最长的对称轴就可以确定边缘方向。具体可通过如下求和所得到的结果来估计：

$$F_x(x_0,y_0)=\sum_{(x,y)\in N(x_0,y_0)}(x-x_0)^2C(x_0,y_0;x,y) \tag{3.1.5}$$

$$F_y(x_0,y_0)=\sum_{(x,y)\in N(x_0,y_0)}(y-y_0)^2C(x_0,y_0;x,y) \tag{3.1.6}$$

$$F_{xy}(x_0,y_0)=\sum_{(x,y)\in N(x_0,y_0)}(x-x_0)(y-y_0)C(x_0,y_0;x,y) \tag{3.1.7}$$

用 $F_y(x_0,y_0)$ 和 $F_x(x_0,y_0)$ 的比值就可确定边缘的朝向，而用 $F_{xy}(x_0,y_0)$ 的符号可帮助区别对角方向的边缘具有正的或负的梯度。

剩下来的问题就是如何自动地确定哪个图像点属于哪种情况。首先，如果 USAN 区域的面积比模板的直径小，那么就应该是像素内部边缘的情况。如果 USAN 的面积比模板的直径大，那么就可以确定出 USAN 区域的重心，并可用来根据像素之间边缘的情况来计算边缘的方向。当然，如果重心处在离核不到一个像素的位置，那么这更有可能属于像素内部边缘的情况。如果用较大的模板使得处于中间灰度的带比一个像素还宽时，这种情况就会发生。

3. SUSAN 算子的特点

与其他边缘和角点检测算子相比，SUSAN 算子有一些独特的地方。

在用 SUSAN 算子对边缘和角点进行检测增强时不需要计算微分，这可帮助解释为什么在有噪声时 SUSAN 算子的性能会较好。这个特点再加上 SUSAN 算子的非线性响应特点都有利于减少噪声。为理解这一点，可考虑一个混有独立分布的高斯噪声的输入信号。只要噪声相对于 USAN 面积比较小，就不会影响基于 USAN 面积所做的判断。这里在面积计算中对各个像素值的求和操作进一步减少了噪声的影响。

SUSAN 算子的另一个特点可以从图 3.1.1 中的 USAN 区域看出。当边缘变得模糊时，在边缘中心的 USAN 区域面积将减少。所以，对边缘的响应将随着边缘的平滑或模糊而增强。这个有趣的现象对一般的边缘检测算子是不常见的，但它在实际中很有用。

最后，大多数的边缘检测算子会随图像或模板尺度的变化而改变所检测出来的边缘的位置，但 SUSAN 检测算子能提供不依赖于模板尺寸的边缘精度。换句话说，最小 USAN 区域面积的计算是个相对的概念，与模板尺寸无关，所以 SUSAN 算子的性能不受模板尺寸影响。这是一个很有用的期望特性。SUSAN 算子的另一个优点是控制参数的选择很简单，且任意性较小，所以比较容易实现自动化选取。

如果对边缘位置的精度要求比用整个像素作为计算单元所能获得的要高，可采用下面的方法来改进以获得亚像素的精度（还可参见 4.2 节）。对每个边缘点，先确定在该点的边

缘方向，然后在与该点垂直的方向上细化边缘。对这样剩下来的边缘点用 3 点的二阶曲线来拟合初始的边缘响应，在这条拟合线上的转向点（应该和细化后边缘点的中心距离小于半个像素）可取作边缘的准确位置。

3.2 图 割 方 法

图割方法是一类基于图论的图像分割技术，本质上采用了基于边缘的串行分割思路。下面先给出用图割方法进行图像分割的主要步骤，再具体对每个步骤进行解释。

用图割方法进行图像分割的主要步骤为：

（1）将待分割图像 I 映射为一个对弧加权的有向图 G，它在尺寸上和维数上都与 I 对应。

（2）确定目标和背景的种子，并针对它们构建两个特殊的图节点，即源节点 s 和汇节点 t；然后将所有种子根据它们的目标或背景标号分别与源节点或汇节点相连接。

（3）计算弧代价函数，并对图 G 中的各个弧赋予一定的弧代价。

（4）使用最大流图优化算法来确定对图 G 的图割，从而区分对应目标和背景像素的节点。

下面对 4 个步骤的一些原理和方法分别进行简单介绍。

1. 构建图 *G*

一个**图**结构可表示为 $G=[N,A]$，其中 N 是一个有限非空的节点集合，A 是一个无序节点对的集合。集合 A 中的每个节点对 (n_i,n_j) 称为一段弧（$n_i\in N,n_j\in N$）。如果图中的弧是有向的，即从一个节点指向另一个节点，则称该弧为有向弧，称该图为有向图。对任一段弧 (n_i,n_j) 都可定义一个代价（或费用），记为 $C(n_i,n_j)$，它可看作是对弧的加权。

对给定的待分割图像 I，要将其转化表示为一个对弧加权的图 G。其中，将图像 I 中每个像素看成图 G 中的一个节点，即节点集合 N 由所有像素构成；而将像素间的邻接关系用图 G 中的弧来表示，即节点对集合 A 表示像素间的（加权）联系。

2. 构建源节点和汇节点

首先需要确定目标种子和背景种子，它们应分别是最终分割结果中目标和背景的一部分。这个工作目前采用的方法常借助人机交互来进行。根据所确定的种子，可以对应构建两个特殊的图节点：源节点 s 和汇节点 t。对目标种子和背景种子都赋予相应的目标或背景标号，并对所有种子根据其标号或者与源节点或者与汇节点相连接。

如上可获得一个弧加权图 $G_{st}=[N\cup\{s,t\},A]$，其中节点集 N 对应图像 I 中的像素，s 和 t 是两个特殊的终端节点。在 G_{st} 中的弧集合 A 中的元素可以分为两类：连接一对相邻像素的弧与将像素和终端节点连接起来的弧。在 G_{st} 中的一个割将图中节点分成两组，它的代价是这个割所对应的弧（割所穿过/跨越的弧）的代价之和。代价最小的割称为最小 s-t 割，它将节点分成两组不重叠的子集 S（所有与源连接的节点，$s\in S$）和 T（所有与汇连接的

节点，$t \in T$)，且从 s 到 t 没有有向的通路。图 3.2.1 给出一个解释上面利用图割进行分割的示意。图 3.2.1(a)给出待分割图像 I，且已确定了目标种子 o 和背景种子 b，$o \subset N$，$b \subset N$，$o \cap b = \varnothing$。图 3.2.1(b)是所构建的图 G_{st}(其中，连接相邻像素的弧用虚线表示，连接像素和终端节点的弧用不同的实线表示)。图 3.2.1(c)在图 G_{st} 中给出 s-t 割(将相应的弧割断了)，其中所有目标像素都与目标种子节点相连而所有背景像素都与背景种子节点相连。图 3.2.1(d)给出根据 s-t 割将节点映射回图像而得到的分割结果。

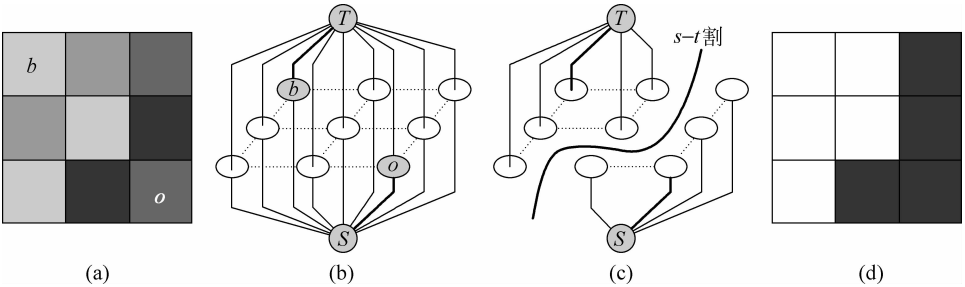

图 3.2.1　图割分割示意

3. 弧的代价

代价最小 s-t 割的代价是其所对应的所有弧的代价之和。下面考虑各种弧的**代价函数**的计算。

给各个像素 i_k 一个二值的标号 $L_k \in \{o, b\}$，其中 o 和 b 分别代表目标和背景的标号。标号矢量 $\boldsymbol{L} = \{L_1, L_2, \cdots, L_{|I|}\}$ 定义所得到的二值分割结果。为计算一段弧的代价，既要考虑该弧两个端节点所对应像素的灰度，也要考虑该弧两个端节点所对应像素间的灰度差。这里，需要最小化以获得最优标号的代价函数可以定义为一个 λ 加权的区域性质项 $R(\boldsymbol{L})$ 和边界性质项 $F(\boldsymbol{L})$ 的组合：

$$C(\boldsymbol{L}) = \lambda R(\boldsymbol{L}) + F(\boldsymbol{L}) \tag{3.2.1}$$

其中

$$R(\boldsymbol{L}) = \sum_{p \in I} R_p(L_p) \tag{3.2.2}$$

$$F(\boldsymbol{L}) = \sum_{(p,q) \in A} F_{(p,q)} \delta(L_p, L_q) \tag{3.2.3}$$

$$\delta(L_p, L_p) = \begin{cases} 1 & L_p \neq L_q \\ 0 & \text{其他} \end{cases} \tag{3.2.4}$$

先考虑连接一对相邻像素的弧。一方面，给定一个像素，根据其灰度将其标为目标或背景都会有一定的代价。这里，$R_p(o)$ 可看做是将像素 p 标为目标的代价，而 $R_p(b)$ 可看做是将像素 p 标为背景的代价。当亮目标在暗背景上时，$R_p(o)$ 的值在暗像素(低 I_p 值)处大而在亮像素处小。反之，当暗目标在亮背景上时，$R_p(b)$ 的值在亮像素(高 I_p 值)处大而在暗像

素处小。另一方面，对两个相邻的像素 p 和 q，根据其灰度将其赋予不同的标号也会有一定的代价。如果它们都属于目标或背景，则弧 (p,q) 的代价 $F_{(p,q)}$ 应比较大；如果它们一个属于目标而另一个属于背景，即跨越目标和背景的边界，则弧 (p,q) 的代价 $F_{(p,q)}$ 应比较小。例如，可取两个相邻像素 p 和 q 之间弧 (p,q) 的代价 $F_{(p,q)}$ 与它们间的梯度幅度成反比。

再考虑将像素和终端节点连接起来的弧，它应由一系列相邻像素的弧所构成，所以其总代价应是这些相邻像素的弧的代价之和。实际中常再加个 1 以使弧不饱和（B 和 O 分别代表背景和目标像素集合）：

$$K = 1 + \max_{p \in B} \sum_{q:\ (p,q) \in O} F(p,q) \tag{3.2.5}$$

将上述分析结果结合起来，赋予各种弧的代价函数如表 3.2.1 所示。

表 3.2.1　各种弧的代价函数

弧	(p,q)	(s,p)	(p,t)
代价	$C_{(p,q)}$　$(p,q) \in N$	$\lambda R_p(b)$　$p \in I, p \notin (O \cup B)$ K　$p \in O$ 0　$p \in B$	$\lambda R_p(o)$　$p \in I, p \notin (O \cup B)$ 0　$p \in O$ K　$p \in B$

4. 图割的计算

计算最小 s-t 割的问题可借助它的对偶——计算最大流问题来进行，即可通过搜索从源 s 到汇 t 的最大流来解决。式 (3.2.5) 的弧代价也可解释成从源 s 到 $p \in O$（或从 $p \in B$ 到汇 t）的最大需要的流容量。在最大流算法中，从源 s 到汇 t 的“最大量的水流”通过图中的弧来输送，而通过各个弧的水流由它的容量或弧代价所决定。从源 s 到汇 t 的最大流能使图中的一组弧达到饱和，这些饱和的弧将节点分为不重合的两部分（对应最小割）S 和 T。最大流的值等于最小割的代价。

最小 s-t 割问题和它的对偶最大流问题是传统的**组合问题**，可用多项式时间的算法解决。有许多算法可用来解决这个组合优化的问题[Sonka 2008]，如**增加通路算法**，优化最大流的压入和重标记（push-relabel）算法等。

增加通路算法考虑推动从源 s 到汇 t 的流直至达到最大流。算法开始时将流的状态初始化为 0，在将流逐渐饱和的过程中，把流分布的当前状态保存在一个**残留图** G_r 中。当 G_r 的拓扑与 G_{st} 相同时，弧的值在当前流的状态下保留了余下弧的流量。在各个迭代步骤，算法沿残留图中未饱和的弧来确定最短的 $s \rightarrow t$ 通路。流过这条通路的流量借助推动最大可能的流而增加，使得这条通路上的至少一个弧达到饱和。换句话说，沿该通路的流量增加 Δr，则通路中弧的残留流量就减少 Δr。每个流量增加的步骤都能增加从源 s 到汇 t 的总流量。一旦流量不能再增加（不能定义新的 $s \rightarrow t$ 通路组成非饱和的弧），则达到最大流，最优化过程结束。此时分开 S 和 T 的图节点确定了分割——最小 s-t 割，它由饱和的弧构成。

确定最短通路的算法可采用宽度优先搜索。一旦所有长度为 k 的通路都饱和了，算法

开始搜索长度为 $k+1$ 的 $s\rightarrow t$ 通路。算法的复杂度是 $O(mn^2)$，其中 n 是节点数而 m 是弧数。图 3.2.2 给出一个使用增加通路最大流算法来确定最小割的所需步骤示例。其中，图 3.2.2(a)所示为与图 3.2.1(a)对应的原始图像；图 3.2.2(b)所示为根据 4-连接计算得到的图像灰度的边缘幅度；图 3.2.2(c)所示为根据表 3.2.1 构建的图 G_{st}，其中 $\lambda=0$；图 3.2.2(d)所示为当唯一具有非饱和 $s\rightarrow t$ 连接的最短通路被确定且饱和后的残余图 G_r，此时不能再找到新的非饱和 $s\rightarrow t$ 通路了；图 3.2.2(e)为用粗线表示的饱和图的弧；图 3.2.2(f)加上了分开 S 和 T 节点的最小 s-t 割；图 3.2.2(g)是最终的图像分割结果。

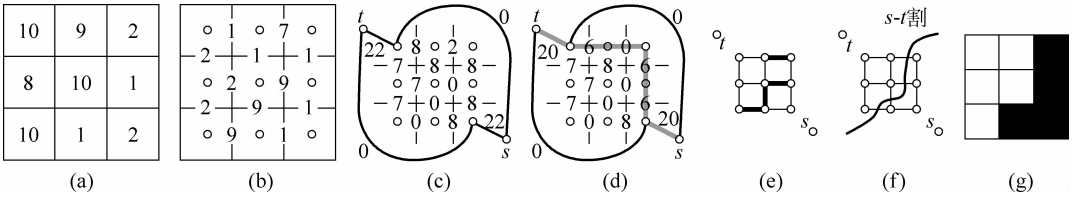

图 3.2.2　利用图割和最大流优化的图像分割

一旦获得了图 3.2.1(c)中的 s-t 割，则将节点映射回图像就可得到分割结果。图割方法的一个重要特性是提供了一种借助交互以有效方法改进先前获得的分割结果的能力。设用户确定了初始的种子，有了代价函数，但图割产生的结果还未达到要求。用户可通过增加目标或背景的种子进行改进，此时不需要从头计算，可以使用先前的图割结果来初始化下一个图割优化过程[Sonka 2008]。

3.3　特色的阈值化和聚类技术

第 2 章介绍并行区域分割方法时主要讨论了阈值化和聚类方法。本节介绍两种有一定代表性的借助特殊理论和方法确定阈值的技术以及一种近年常用的确定空间聚类以分割图像的技术。

(1) 借助小波变换的多分辨率特性来帮助进行阈值选取。

(2) 借助过渡区选取阈值进行分割。

(3) 借助均移计算特征空间的聚类中心来进行分割。

3.3.1　多分辨率阈值选取

利用图像的**直方图**帮助选取阈值是常用的方法，其中的关键是确定峰点和谷点。由于场景的复杂性，图像成像过程中各种干扰因素的存在等原因，峰点和谷点的有无检测和位置确定常比较困难。峰点和谷点的检测与直方图的尺度有密切的联系。一般在较大尺度下常能较可靠地检测到真正的峰点和谷点，但在大尺度下对峰点和谷点的定位不易准确。相反，在较小尺度下对真正峰点和谷点的定位常比较准确，但在小尺度下误检或漏检的比例会增加。

图像在小波变换后可分解为一系列尺度不同的分量。对小波变换后的图像直方图也可进行多分辨率分析，具体就是先在较大尺度下检测出真正的峰点和谷点，再在较小尺度下对这些峰点和谷点进行较精确的定位。这里主要有如下两个步骤。

1. 确定分割区域的类数

首先利用在较大尺度(粗分辨率)下的直方图细节信息确定分割区域的类数。引入相当于低通滤波器的尺度函数 $\phi(x)$，则图像直方图 $H(x)$的低通分量可表示为

$$S_{2^i}[H(x)] = H(x) \otimes \phi_{2^i}(x) \tag{3.3.1}$$

设原始图像直方图信号的分辨率为 1，最低分辨率尺度为 2^I，则尺度 2^1 和 2^I之间的各阶小波变换为$\{W_{2^i}[H(x)],1 \leqslant i \leqslant I\}$。可以证明，对信号在尺度为 2^I时被平滑掉的高频部分可以用尺度在 2^1 和 2^I之间的小波变换来恢复。这里集合$\{S_{2^i}[H(x)],W_{2^i}[H(x)],1 \leqslant i \leqslant I\}$就是直方图的多分辨率小波分解表示。

先在分辨率为 2^1 时确定初始的分割区域类数，即判断直方图中独立峰的个数。这里要求独立峰应满足三个条件：①具有一定的灰度范围；②具有一定的峰下面积；③具有一定的峰谷差。

2. 确定最优阈值

确定类数后，可利用多分辨率的层次结构在直方图的相邻峰之间确定**最优阈值**。这个过程首先在最低分辨率一层进行，然后逐渐向高层推进，直到最高分辨率层。选高斯函数作为平滑函数 $\theta(x)$，令小波函数为 $\psi(x)$，有

$$\psi(x) = \frac{\mathrm{d}^2\theta(x)}{\mathrm{d}x^2} \tag{3.3.2}$$

则 $\psi(x)$对应的二进小波变换为

$$W_{2^i}[H(x)] = (2^i)^2 \cdot \frac{\mathrm{d}^2}{\mathrm{d}x^2}[H(x) \otimes \theta_{2^i}(x)] \tag{3.3.3}$$

由上式可知小波变换的零交叉位置对应在分辨率 2^i 时的低通信号 $H(x) \otimes \theta_{2^i}(x)$的剧烈变化点。当尺度 2^i 减小时，信号的局部细节增多；而当尺度 2^i 增加时，信号中结构较大的轮廓比较明显。

对图像的直方图来说，式(3.3.1)中的 $S_{2^i}[H(x)]$是一个最低分辨率下的近似信号，式(3.3.3)中的 $W_{2^i}[H(x)]$代表不同分辨率下的细节信号，它们联合构成直方图的多分辨率小波分解表示。给定直方图，可考虑其多分辨率小波分解表示的零交叉点和极值点来确定直方图的峰值点和谷点。具体可利用以下四个准则(参见图 3.3.1 的直方图)：

(1) 用从负值变化到正值的零交叉点确定峰的起点(图中各 s 点)；

(2) 用从正值变化到负值的零交叉点确定峰的终点(图中各 e 点)；

(3) 用起点和终点间的最大值点确定峰的位置(图中各 T 点)；

(4) 用前一个峰的终点和后一个峰的起点间的最小值点确定这两个峰之间谷点的位置(图中各 B 点)。