



## 3.1 引言

图像配准算法的目的是寻找待配准图像间的最优空间变换参数,由特征空间、搜索空间、相似性度量及搜索策略 4 个部分组成<sup>[1]</sup>。特征空间是从待配准图像中提取的用于进行配准的特征信息,如图像的灰度、梯度信息以及其他图像特征;搜索空间是在配准时选择的 空间变换的方式(刚体变换、仿射变换、投影变换、弹性变换等),即选择什么样的变换模型;相似性度量是衡量浮动图像经过每次变换后与参考图像之间的相似性,并为下一次是否进行变换提供依据;搜索策略决定下一步空间变换的方向,即采用的优化方案,以便得到最优空间变换。图像配准算法的基本结构如图 3-1 所示。

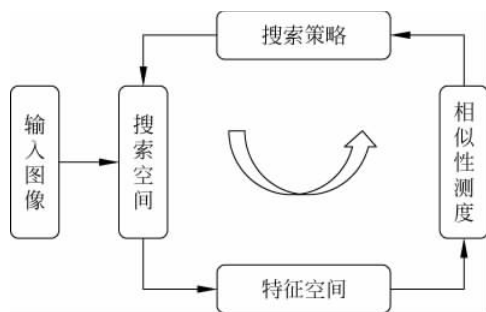


图 3-1 图像配准流程图

## 3.2 特征空间

### 3.2.1 点检测

点检测属于图像灰度级间断检测技术,寻找间断最一般的方法是:以邻域像素处理的方式对整幅图像使用一个模板进行检测<sup>[2]</sup>。对于一个 $3 \times 3$ 的模板(见图 3-2),检测过程是计算模板所包围区域内图像的灰度值与模板系数乘积之和,图像中任意点的模板响应由下式给出:

$$R = \omega_1 z_1 + \omega_2 z_2 + \cdots + \omega_9 z_9 = \sum_{i=1}^9 \omega_i z_i \quad (3.1)$$

这里 $z_i$ 是模板包围区域内图像像素点的灰度级, $R$ 是相对于中心位置的模板响应,对整幅图像的检测过程即利用模板对所有像素点进行一次遍历。

对于一幅图像,孤立点的检测在理论上是很简单的。使用如图 3-3 所示的模板,如果某个点的模板响应满足

$$g(x, y) = \begin{cases} 1 & |R(x, y)| \geq T \\ 0 & \text{其他} \end{cases} \quad (3.2)$$

$\omega_1$	$\omega_2$	$\omega_3$
$\omega_4$	$\omega_5$	$\omega_6$
$\omega_7$	$\omega_8$	$\omega_9$

图 3-2  $3 \times 3$  模板

-1	-1	-1
-1	8	-1
-1	-1	-1

图 3-3 点检测模板

则说在模板中心位置上检测到了一个点,这里 $T$ 是一个非负的阈值, $R$ 由式(3.1)给出, $g(x, y)$ 为输出图像中对应点的灰度值。实质上,这个公式是衡量中心像素点与其周围邻域点之间加权的差值。如果一个孤立的点与周围邻域点的灰度差异较大,而且它所处的位置是一个具有均匀灰度值的区域,那么认为这个点与周围点很不相同,就很容易被这类模板检测到。阈值 $T$ 的选择是能否检测出孤立点的关键。需要注意的是,模板系数之和为零表明在灰度级相同或相近的区域内,模板的响应为零。

### 3.2.2 线检测

更复杂点的检测是线检测,考虑图 3-4 中显示的模板。通常,我们要检测的是特定方向上的线。如果使用第一个模板对一幅具有恒定背景并包含各种线的图像进行滤波,那么这个模板将对水平方向的线条有很强的响应,可以检测出图像中水平方向的线条。类似地,第二个模板对  $45^\circ$  方向的线具有最佳响应;第三个模板对于垂直线具有最佳响应;第四个模板则对  $-45^\circ$  方向的线具有最佳响应。

-1	-1	-1	2	-1	-1
2	2	2	-1	2	-1
-1	-1	-1	-1	-1	2
(a)			(b)		
-1	2	-1	-1	-1	2
-1	2	-1	-1	2	-1
-1	2	-1	2	-1	-1
(c)			(d)		

图 3-4 线检测模板

令  $R_1$ 、 $R_2$ 、 $R_3$  和  $R_4$  表示图 3-4 中从左到右的各个模板的响应,其中  $R$  值由式(3.1)给出。假设使用这 4 个模板对同一幅图像滤波,在该图像中的某个给定像素点处,如果对于所有  $j \neq k$ ,有  $|R_k| > |R_j|$ ,则称该点可能与模板  $k$  方向上的线更相似。比如,如果图像中某点处,对于  $j=2,3,4$  有  $|R_1| > |R_j|$ ,那么该点可能与一条水平线更接近。

### 3.2.3 霍夫变换

霍夫(Hough)变换是一种检测、定位直线和解析曲线的有效方法<sup>[2]</sup>。霍夫变换是 1962 年由霍夫提出来的,在影像分析、模式识别等很多领域中得到了成功的应用。它将图像空间中的曲线(包括直线)变换到霍夫参

数空间,通过检测参数空间中的极值点,确定出该曲线的描述参数,从而提取图像中的规则曲线。这种变换的最大优点是抗噪性能强,能够在信噪比较低的条件下,检测出直线或解析曲线。它要求首先对图像做二值化及边缘检测等预处理,因此,原始图像会有一些的信息损失。正因如此,给它的应用带来了局限性。

考虑坐标系中任意一点 $(x_i, y_i)$ 和一条直线的方程为 $y_i = ax_i + b$ 。通过点 $(x_i, y_i)$ 的直线有无数条,且对不同的 $a$ 和 $b$ 值,它们都满足这个等式。然而,将直线方程写成 $b = -x_i a + y_i$ 的形式并参考 $ab$ 平面(参数空间)将得到对于定点 $(x_i, y_i)$ 的唯一直线方程。此外,第二个点 $(x_j, y_j)$ 在参数空间中也有一条与之相关联的直线,除非它们是平行的,否则这条直线会与和 $(x_i, y_i)$ 相关联的直线相交于点 $(a', b')$ ,其中 $a'$ 为斜率, $b'$ 是 $xy$ 平面中点 $(x_i, y_i)$ 和点 $(x_j, y_j)$ 所连直线的截距。实际上,这条直线上的所有点在参数空间中都有相交于点 $(a', b')$ 的直线,如图3-5所示。

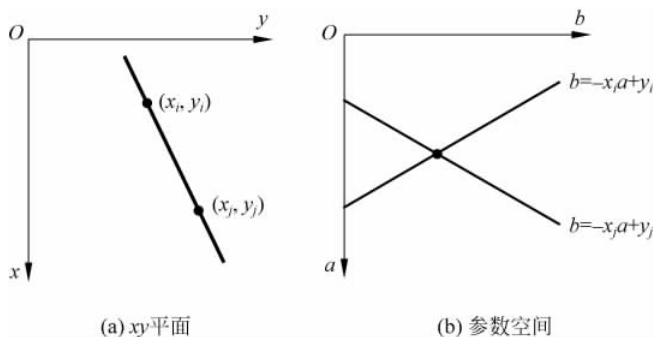


图 3-5 直线方程

理论上,可以画出对应于 $xy$ 平面中所有点 $(x_k, y_k)$ 的参数空间直线,而且空间中的主要直线可以通过确定的点来找到,大量的参数空间的线在此点处相交。然而,此方法的一个实际困难是,当该直线逼近垂直方向时, $a$ 的值(直线的斜率)会趋向于无穷大。解决该困难的方法之一是用参数 $\theta$ 和 $\rho$ 表示:

$$x \cos \theta + y \sin \theta = \rho$$

图3-6示例了参数 $\theta$ 和 $\rho$ 的几何解释。当 $\theta=0^\circ$ 时, $\rho$ 等于正的 $x$ 截距;类似地,当 $\theta=90^\circ$ 时, $\rho$ 等于正的 $y$ 截距。

霍夫变换的魅力在于可将 $\rho\theta$ 参数空间划分为所谓的累加单元,如图3-7所示,其中 $(\rho_{\min}, \rho_{\max})$ 和 $(\theta_{\min}, \theta_{\max})$ 是所期望的参数值范围: $-90^\circ \leq \theta \leq 90^\circ, -D \leq \rho \leq D$ , $D$ 是图像中对角之间的最大距离<sup>[3]</sup>。位于坐标 $(i, j)$

处的单元具有累加值  $A(i, j)$ , 它对应于参数空间坐标  $(\rho_i, \theta_j)$  相关联的正方形。最初, 将这些单元设置为零。然后, 对于  $xy$  平面中的每个非背景点  $(x_k, y_k)$ , 令  $\theta$  等于  $\theta$  轴上每个允许的细分值, 同时使用方程  $\rho = x_k \cos\theta + y_k \sin\theta$  解出对应的  $\rho$ 。对得到的  $\rho$  进行四舍五入, 得到沿  $\rho$  轴的最近的允许单元值。如果选择的一个  $\theta_p$  值得到解  $\rho_q$ , 则令  $A(p, q) = A(p, q) + 1$ 。在这一过程结束后,  $A(i, j)$  中值  $P$  将意味着  $xy$  平面中有  $P$  个点位于直线  $x \cos\theta_j + y \sin\theta_j = \rho_i$  上。  $\rho\theta$  平面中的细分数量决定了这些点的共线性的精确度。

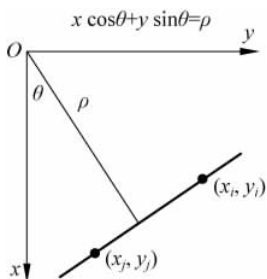


图 3-6  $xy$  平面中直线的  $\rho\theta$  参数化

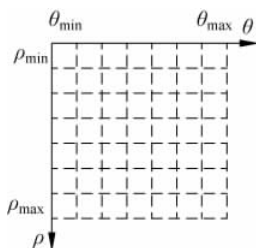


图 3-7  $\rho\theta$  平面划分为累加单元

### 3.2.4 边缘检测

图像的边缘检测对图像识别和计算机分析十分有用。边缘检测能勾画出目标物体, 使观察者一目了然。边缘包含了丰富的内在信息(如方向、阶跃性质、形状等), 是图像识别中抽取的重要特征。边缘检测首先检出图像局部特性的不连续性, 然后再将这些不连续的边缘像素连成完整的边界。边缘的特性是沿边缘走向的像素变化平缓, 而垂直边缘方向的像素变化剧烈。因此, 从这个意义上讲, 边缘检测算法是检出符合边缘特性的图像像素的数学算子。目前, 边缘检测常用的方法有算子法和模板匹配法。

#### 1. 梯度算子

为了在一幅图像  $f$  的  $(x, y)$  位置处寻找边缘的强度和方向, 梯度是最常用的方法, 图像的梯度用矢量来表示:

$$\nabla f(x, y) = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T \quad (3.3)$$

该矢量指出了  $f$  在位置  $(x, y)$  处的最大变化率的方向。梯度的幅值由下式给出:

$$|\nabla f(x, y)| = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (3.4)$$

上式可以简化为

$$|\nabla f(x, y)| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad (3.5)$$

它是梯度矢量方向上变化率的值。对于数字图像来说,

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y), \quad \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y) \quad (3.6)$$

将式(3.6)代入式(3.4)或式(3.5)中可以得到梯度的幅度。这种梯度法叫作水平垂直差分法。另外一种梯度方法采用交叉的方式计算差分,称作罗伯特(Robert)梯度法,表示为

$$\begin{aligned} & |\nabla f(x, y)| \\ &= [(f(x+1, y+1) - f(x, y))^2 + (f(x+1, y) - f(x, y+1))^2]^{1/2} \end{aligned} \quad (3.7)$$

图 3-8 所示的是几种常用的边缘检测算子,图 3-8(a)~(c)分别代表 Robert 算子、Prewitt 算子和 Sobel 算子。

-1	0	0	-1	-1	-1	-1	-1	0	1
0	1	1	0	0	0	0	-1	0	1
0	1	1	1	1	1	1	-1	0	1

(a) Robert算子

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

(c) Sobel算子

图 3-8 几种常用的梯度算子

梯度算子计算简单,但精度不高,只能检测出图像大致的轮廓,而对于比较细的边缘可能会忽略。

## 2. 拉普拉斯算子

拉普拉斯算子是常用的边缘检测算子,它是各向同性的二阶导数:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.8)$$

检测后的图像表示为:

$$g(x, y) = f(x, y) - k \nabla^2 f(x, y) \quad (3.9)$$

式中系数  $k$  与扩散效应有关。选择合适的  $k$  值是检测出理想边缘的关键。对数字图像来说,  $f$  在位置  $(x, y)$  处的二阶导数可以表示为:

$$\begin{cases} \frac{\partial^2 f(x, y)}{\partial x^2} = [f(x+1, y) - f(x, y)] - [f(x, y) - f(x-1, y)] \\ \qquad \qquad \qquad = f(x+1, y) + f(x-1, y) - 2f(x, y) \\ \frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \end{cases} \quad (3.10)$$

另外, 上式也可以表示成模板的形式, 如图 3-9 所示。从模板容易看出, 如果图像中的一个较暗的区域内出现了一个亮点, 那么使用拉普拉斯算子后会使得该点变得更亮, 因此拉普拉斯算子具有锐化图像的作用。

0	1	0
1	-4	1
0	1	0

图 3-9 拉普拉斯算子

### 3.2.5 Harris 角点

Harris 角点检测算法<sup>[4]</sup>是 Harris 和 Stephens 于 1988 年提出的, Harris 算法的原理是判断一点向任意方向的一个小小偏移是否会引起灰度值的很大变化, 如果这一点会因为小小的偏移灰度值发生了很大的变化, 则说明这一点就是角点, 它的算法思想是通过计算一个像素周围邻域的自相关函数的自相关矩阵的特征值来判断这一点是否是角点。这个局部自相关函数  $E$  定义为:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x+u, y+v) - I(x, y)]^2 \quad (3.11)$$

上式中  $u, v$  分别代表了  $x, y$  方向上的微小偏移量,  $I(x, y)$  代表了图像上坐标为  $(x, y)$  的像素点的灰度值,  $I(x+u, y+v)$  是经过  $(u, v)$  微小偏移后

的图像中该点的灰度值,  $w(x, y)$  是高斯滤波器, 其泰勒展开式为:

$$E(u, v) \approx (u, v) \mathbf{M} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.12)$$

其中  $\mathbf{M}$  为二维方阵。

$$\mathbf{M} = \sum_{x, y} w(x, y) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3.13)$$

$I_x$ 、 $I_y$  分别代表图像一点在  $x$ 、 $y$  方向上的梯度。

设  $\lambda_1$ 、 $\lambda_2$  是矩阵  $\mathbf{M}$  的特征值, 则用  $\lambda_1$ 、 $\lambda_2$  可以表示局部自相关函数的曲率。通过对矩阵  $\mathbf{M}$  的特征值分析可大致可以对角点进行检测, 但实际运用中用来响应函数  $R$  作为标准:

$$R = \det(\mathbf{M}) - k \cdot \text{tr}^2(\mathbf{M}) \quad (3.14)$$

上式中  $\det$  为矩阵的行列式;  $\text{tr}$  为矩阵的迹; 按照大量的实验, 一般情况下  $k$  取值 0.04。其判断标准为: 设定一个合理的阈值  $T$ , 当实际计算出来的  $R$  值大于  $T$  时, 则表示找到了一个角点, 否则不是角点。

### 3.2.6 SIFT 特征点

David G. Lowe 在 2004 年结合其 1999 年的经典文章并总结了已有的很多种基于不变量技术的特征提取方法, 提出了一种基于尺度空间的、对图像缩放、旋转甚至仿射变换保持不变性的特征描述算法 SIFT<sup>[5]</sup>, 它的全称是 Scale Invariant Feature Transform, 即尺度不变特征变换。通过 SIFT 算法提取出的特征点具有尺度不变特性, 对于平移、旋转、缩放、仿射、光照、视角变化等都能保持很强的稳定性, 正因为它具有的这些优秀的特性, SIFT 特征点提取算法成了近几年来图像匹配、图像拼接方面研究的一个热点。

SIFT 算法首先要在尺度空间上进行特征提取, 并且要确定特征点的位置和所在的尺度, 然后使用特征点邻域中梯度的主方向作为该点的方向特征, 此方法的具体步骤如下:

- (1) 在尺度空间上进行极值检测;
- (2) 确定特征点的位置和特征点所在的尺度;
- (3) 确定特征点的方向;
- (4) 根据确定的特征点计算特征矢量。

尺度空间的理论告诉我们: 任何图像特征都是在一定的尺度下考虑才有意义。图像的特征依赖于一定的空间尺度, 该尺度与具体的成像无关, 如果能够充分分析目标本身的尺度, 就能得到与图像本身的尺度无关

而只与空间尺度、目标尺度相关联的尺度不变特征。

在不同的解析度下,一幅图像的各种表示构成了此图像的尺度空间。一幅图像与高斯核函数的卷积可以得到该图像在某一个解析度下的表示,其中高斯核函数的标准差就是此图像的尺度。那么可以说图像的尺度空间是由  $x$ 、 $y$  和  $\sigma$  组成的一个三维空间, $x$ 、 $y$  决定位置,而  $\sigma$  代表解析度。图像尺度越低, $\sigma$  的值就越小,图像就越清晰,对应于图像的细节特征;而当尺度值越大, $\sigma$  值越大,图像就会越模糊,主要对应与概貌特征。

二维图像  $I(x, y)$  的尺度空间  $L(x, y, \sigma)$  的表达如下:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.15)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.16)$$

其中  $\sigma$  表示高斯正态函数的标准差,它表示三维空间中的尺度坐标,  $G(x, y, \sigma)$  为尺度可变高斯函数,  $*$  代表卷积操作。

不同尺度的高斯核滤波之差构成了高斯尺度差分空间 DoG (Difference of Gaussian):

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.17)$$

其中  $k$  为一个常量。

该算法的基本步骤如下:

第一,尺度空间极值检测,初步确定特征点位置和所在尺度。在检测尺度空间极值时,图 3-10 中标记为叉号的像素需要跟与它同一尺度的周围邻域 8 个像素和相邻尺度对应位置的周围邻域  $9 \times 2$  个像素(总共 26 个像素)进行比较,以确保在尺度空间和二维图像空间都检测到局部极值。

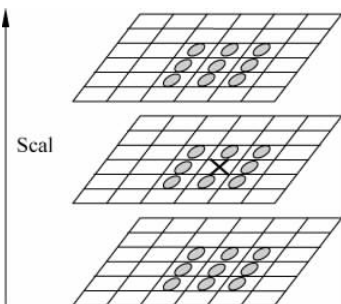


图 3-10 DoG 尺度空间的 3 个邻域尺度

第二,利用拟合函数确定特征点的位置和尺度,去除对比度较低的关键点和边缘点,增加匹配的稳定性,增强抗噪声能力。

第三,利用特征点邻域像素点的特征方向指定其方向参数,使算子具有旋转不变性。

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.18)$$

$$\theta(x, y) = \arctan((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (3.19)$$

上式中  $m(x, y)$  与  $\theta(x, y)$  分别为坐标为  $(x, y)$  的像素点处梯度的模和方向,其中  $L$  为每个关键点各自所在的尺度。在实际计算时,对于每一个特征点使用式(3.19)进行计算,经过此步骤之后,可以得到特征点的坐标位置、所在尺度及其方向。

第四,生成 SIFT 特征矢量。首先将坐标轴旋转为特征点的方向,以确保其旋转不变性。以特征点为中心取  $8 \times 8$  的窗口。图 3-11 的左半部分的每个方框代表一个像素点,箭头的大小和方向分别代表此像素点的梯度值和方向,圆圈代表高斯加权的范围。在每  $4 \times 4$  的小块上计算 8 个方向的梯度方向直方图并计算每个梯度方向的累加值,形成种子点,统计每个窗口中的像素点的梯度值和方向。实际计算中,可使用  $4 \times 4$  个共 16 个种子点来描述,这样一个特征点就可产生 128 个数据,最终形成 128 维的 SIFT 特征矢量。此时,基于 SIFT 的特征矢量已经去除了旋转及尺度变化等变形因素的影响,然后继续归一化特征矢量的长度,可以更进一步地去除光照差异的影响。

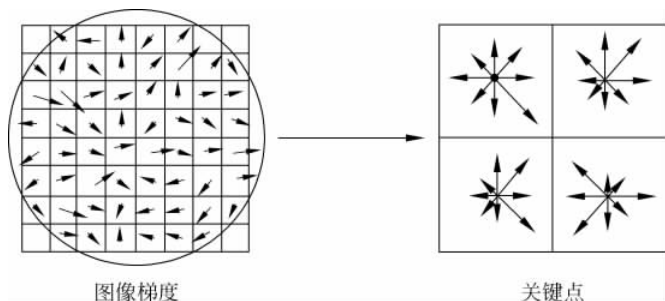


图 3-11 特征点邻域信息得到的特征矢量

### 3.2.7 SURF 特征点

#### 1. Hessian 矩阵

Hessian 矩阵具有很好的定位准确性<sup>[6]</sup>,斑点型结构处 Hessian 矩阵的值能得到最大值,假设给定图像中任意点  $X(x, y)$ ,该点在尺度  $\sigma$  下的 Hessian 矩阵定义为

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (3.20)$$

其中  $L_{xx}(X, \sigma)$  为高斯二阶偏导数  $\frac{\partial^2 g(\sigma)}{\partial x^2}$  与  $X$  点的卷积,  $L_{xy}(X, \sigma)$  和  $L_{yy}(X, \sigma)$  也用类似方法得到。

在实际的尺度空间分析时,高斯滤波都要进行离散和截断,在任何情况下,滤波器都不是理想的。为了提高运算速度,用方块滤波来代替高斯滤波。

图 3-12 中  $9 \times 9$  的方块滤波是  $\sigma=1.2$  的高斯滤波的近似估计,假设将这 3 个方块滤波设为  $D_{xx}$ 、 $D_{yy}$  和  $D_{xy}$ ,Hessian 矩阵的行列式可以近似估计为:

$$\det(\mathbf{H}) \approx D_{xx}D_{yy} - (\omega D_{xy})^2 \quad (3.21)$$

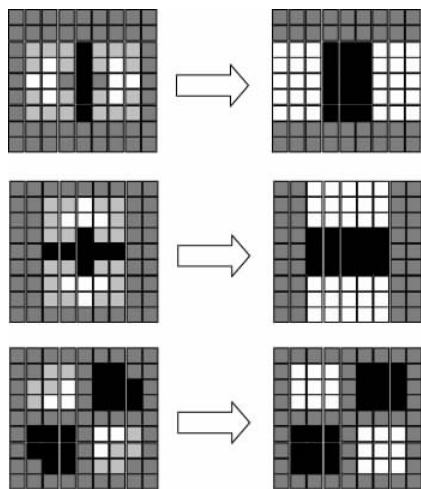


图 3-12 方块滤波近似高斯滤波

$\omega$  是一个权值,对于  $9 \times 9$  的方块滤波来说有:

$$\omega = \frac{|L_{xy}(1, 2)|_F |D_{yy}(9)|_F}{|L_{yy}(1, 2)|_F |D_{xy}(9)|_F} = 0.912... \approx 0.9 \quad (3.22)$$

式中  $|\cdot|_F$  表示 F 范数, 所选择的尺度取决于  $\omega$  的值, 但是此项对于特征检测意义不大, 这里统一把  $\omega$  的值定为 0.9, 则式(3.21)可以变为:

$$\det(\mathbf{H}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (3.23)$$

这个近似的 Hessian 矩阵行列式值代表一点的斑点响应, 计算得到不同尺度下此响应的最大值及为特征点。

## 2. 尺度空间

通常情况下, 图像金字塔用来生成尺度空间(如图 3-13 所示), 前面章节中也有提到, SIFT 算子是采用图像金字塔建立高斯差分(DoG)图像。本书对于图像金字塔的构建是利用放大滤波器模板来实现的, 并不是靠图像尺寸的缩小来实现。

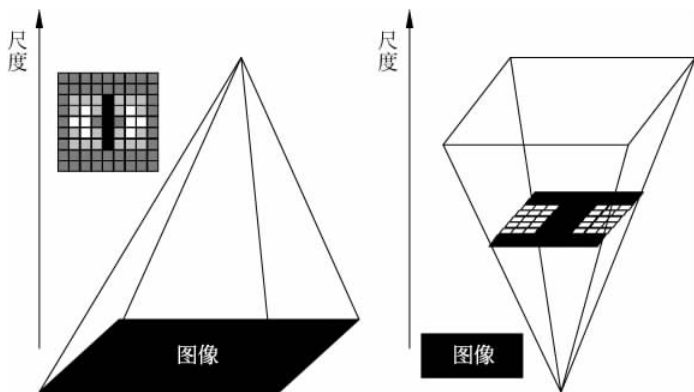


图 3-13 尺度空间建立方法对比

首先要把尺度空间分为几个层, 将每一层滤波器的尺寸成倍增加, 同时提取特征点的尺度间隔也成倍变化。与传统的方法相比, 此算法能够降低计算时间, 同时还能提高精度。第一层的滤波器尺寸为 9, 15, 21, 27, 第二层的滤波器尺寸为 15, 27, 39, 51, 则第三层为 27, 51, 75, 99, 如果原始的图像尺寸仍然比滤波器尺寸大, 则第四层用来分析尺度空间, 第四层的滤波器尺寸为 51, 99, 147, 195, 图 3-14 显示了尺度空间的构建图。

## 3. 确定特征点

此部分内容可以分为几点, 首先, 要对滤波响应进行筛选; 然后, 对通过阈值筛选后的点进行非极大值抑制操作, 这里采用如图 3-14 一样的方法获得既满足阈值条件又是极大值的特征点; 最后利用最近邻域差值法得到在尺度中亚像素精确的特征点, 我们将 Hessian 矩阵的行列式按照泰勒级数展开并省略高次项, 并对其求导便得到极值点的解:

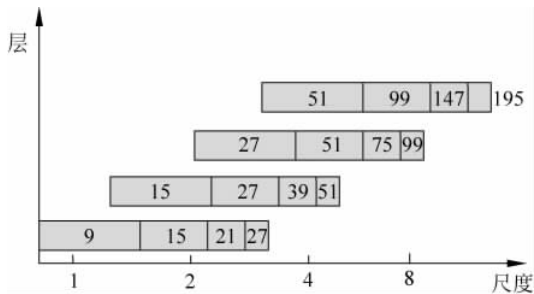


图 3-14 尺度空间构建图

$$\hat{x} = -\frac{\partial^2 \mathbf{H}^{-1}}{\partial x^2} \frac{\partial \mathbf{H}}{\partial x} \quad (3.24)$$

通过上述的方法就能够得到稳健的特征点。

#### 4. 特征矢量的描述

为了保证图像旋转不变性,要保证特征点的一个可重复方向。为达到此目的,首先要在特征点周围半径为  $6\sigma$  的圆形区域内计算哈尔小波响应( $\sigma$  为被检测出的特征点所在的尺度),如图 3-15 所示,用哈尔小波滤波器计算在  $x, y$  方向上的哈尔小波响应。然后再累积各方向上  $60^\circ$  范围内的小波响应,获得的最大的响应的方向为主方向,如图 3-16 所示。

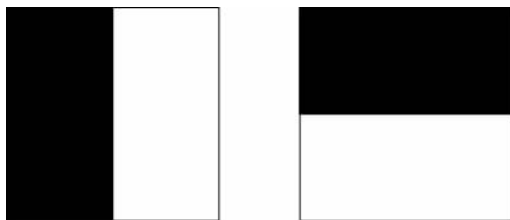


图 3-15 哈尔小波

为了建立特征矢量,首先以特征点为中心构建一个尺寸为  $20\sigma$  的方形区域,并且以刚刚确定的主方向建立坐标系,然后再将这个方形区域分割成  $4 \times 4$  个子区域,这些区域具有重要的空间信息。对于每一个子区域,计算  $5 \times 5$  个采样点的哈尔小波响应,如果将  $x, y$  轴的哈尔小波响应定义为  $d_x, d_y$ ,最后,将这 25 个采样点的响应值及其绝对值进行累加,得到 4 个矢量(见图 3-17):  $\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|$ 。这样,每个子区域内形成了一个四维的矢量,那么每个特征点形成了一个  $4 \times 4 \times 4$  的 64 维描述矢量,则对于每一个子区域的描述矢量如下:

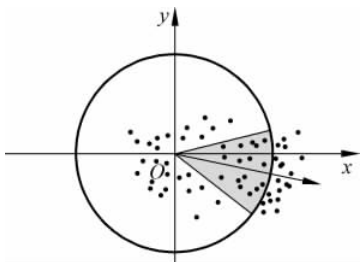


图 3-16 确定主方向

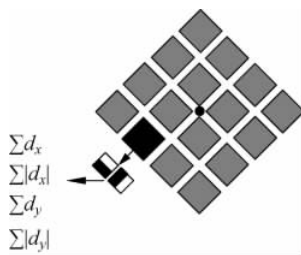


图 3-17 建立特征矢量

$$\mathbf{v}_{\text{subregion}} = \begin{bmatrix} \sum d_x \\ \sum |d_x| \\ \sum d_y \\ \sum |d_y| \end{bmatrix} \quad (3.25)$$

如果需要扩展至 128 维的描述矢量,那么只要根据  $d_x, d_y$  大于或小于 0,分别计算对应的  $\sum d_y, \sum |d_y|$  及  $\sum d_x, \sum |d_x|$  就可以将每个子区域的描述矢量扩展至 8 维。这样整个特征点就有  $4 \times 4 \times 8$  个描述矢量,也就是共有 128 个特征矢量。为了提高算法的鲁棒性, $d_x, d_y$  都经过了尺度为  $3.3\sigma$  的高斯加权。

相对于 SIFT 特征, SURF 特征具有更强的抗噪能力,因为 SURF 特征的描述矢量只取决于邻域梯度的延长水平和垂直方向的累积,而 SIFT 特征的描述矢量是利用特征点邻域内 8 个方向的梯度值表示。

### 3.3 搜索空间

搜索空间包括空间变换范围和空间变换模型<sup>[1]</sup>。依据空间变换范围可分为全局变换和局部变换。全局变换说明整幅图像的变换可以利用相同的变换参数,而局部变换是指在图像上指定大小的子区域可以应用相同的变换参数。常用的空间变换模型有刚体变换、仿射变换、投影变换及弯曲变换(或弹性形变),如图 3-18 所示。选择不同的空间变换模型,配准中优化的参数也不一致。比如三维刚体变换需要求解 6 个变换参数。

#### 3.3.1 刚体变换

刚体变换(Rigid Transformation)只允许平移和旋转两种变换,而且

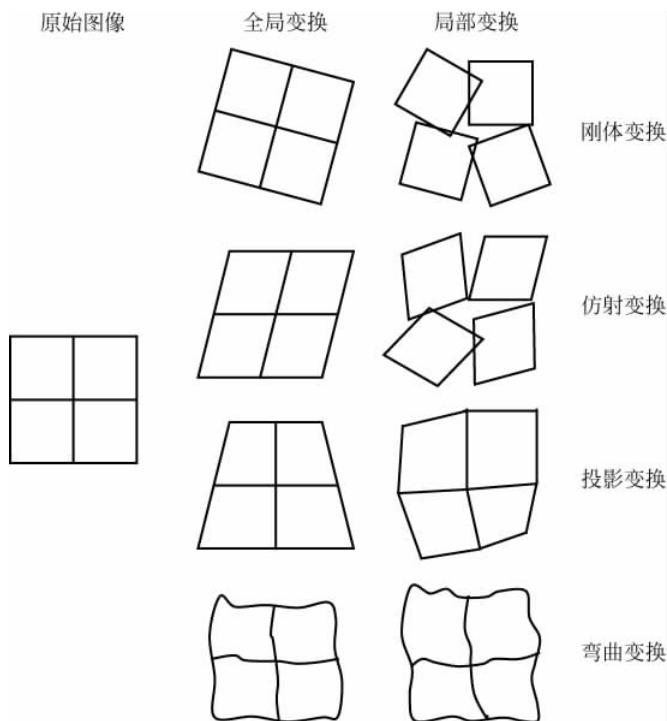


图 3-18 空间变换示意图

图像在经过刚体变换后,上面任意两点的距离保持不变。对于三维图像来说,刚体变换有 6 个自由度:  $x$ 、 $y$ 、 $z$  方向上的平移量与绕  $x$ 、 $y$ 、 $z$  轴的旋转角度。坐标的刚体变换可以用下式表示:

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{t} \quad (3.26)$$

式中  $\mathbf{X} = (x, y, z)^T$  和  $\mathbf{X}' = (x', y', z')^T$  分别代表变换前后的图像中像素点的坐标;  $\mathbf{R}$  是  $3 \times 3$  的旋转矩阵;  $\mathbf{t}$  为  $3 \times 1$  的平移矢量。旋转矩阵  $\mathbf{R}$  可以用 3 个矩阵的乘积来表示:

$$\mathbf{R} = \mathbf{r}^{(x)} \mathbf{r}^{(y)} \mathbf{r}^{(z)} \quad (3.27)$$

式中矩阵  $\mathbf{r}^{(x)}$ 、 $\mathbf{r}^{(y)}$  和  $\mathbf{r}^{(z)}$  可以用 3 个旋转角度表示:

$$\mathbf{r}^{(x)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \quad (3.28)$$

$$\mathbf{r}^{(y)} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \quad (3.29)$$

$$\mathbf{r}^{(z)} = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.30)$$

式(3.28)、式(3.29)和式(3.30)中的 $\alpha$ 、 $\beta$ 、 $\gamma$ 分别代表图像绕 $x$ 、 $y$ 、 $z$ 轴的旋转角度。刚体变换是空间变换中最简单、最基本的变换,同时在医学图像配准中也起到至关重要的作用,例如刚体变换经常用于配准头部图像。此外在某些非刚体配准实验中,先对待配准图像执行刚体配准,使得目标的空间位置大体一致,从而降低后续非刚体配准的计算量。

### 3.3.2 仿射变换

仿射变换(Affine Transformation)除了平移和旋转之外,还允许尺度和剪切变换。图像中的任意平行线经仿射变换后仍然为平行线。三维图像的仿射变换关系同样可以用式(3.27)表示,与刚体变换不同的是仿射变换的旋转矩阵 $\mathbf{R}$ 并不能简单地用3个角度来表达,而是一个普通的 $3 \times 3$ 矩阵。用下边的公式表示 $\mathbf{R}$ :

$$\mathbf{R} = \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 \\ \mu_4 & \mu_5 & \mu_6 \\ \mu_7 & \mu_8 & \mu_9 \end{pmatrix} \quad (3.31)$$

可以用齐次坐标的形式将图像上坐标的仿射变换表示为:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 & t_x \\ \mu_4 & \mu_5 & \mu_6 & t_y \\ \mu_7 & \mu_8 & \mu_9 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.32)$$

式中 $t_x$ 、 $t_y$ 、 $t_z$ 分别为 $x$ 、 $y$ 、 $z$ 方向上的位移; $\mu_i$ ( $i=1,2,\dots,9$ )为式(3.31)中旋转矩阵 $\mathbf{R}$ 的9个参数。式(3.32)表明,仿射变换一共有12个参数(自由度)。

### 3.3.3 投影变换

投影变换(Projective Transformation)将图像中直线变换成直线。多用于三维图像向二维平面的投影。投影变换的表达式为:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 & t_x \\ \mu_4 & \mu_5 & \mu_6 & t_y \\ \mu_7 & \mu_8 & \mu_9 & t_z \\ \mu_{10} & \mu_{11} & \mu_{12} & \mu_{13} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.33)$$

投影变换主要用于二维图像与三维图像的配准算法中。

### 3.3.4 弯曲变换

弯曲变换(Curved Transformation)也称弹性形变(Elastic Deformation),是几种变换模型中最复杂的一种变换,它将直线变换为曲线,反映图像间细微、局部的变形。弯曲变换把图像中的直线转化为曲线,可以用下式来表示:

$$(x', y') = F(x, y) \quad (3.34)$$

$F(x, y)$ 是个非线性函数,它可以把一幅图像中的直线转化为第二幅图像中的任意一种形式。

弹性形变是医学图像中最常见的一种变换方式,在医学图像成像过程中,由于人体体位的变化而使一些软组织或器官的大小、形状发生变化,因此不同时间拍摄的医学图像之间可能存在着弹性形变。弹性配准是目前医学图像配准领域的研究热点,而且因为弹性形变模型的参数多,计算量大,所以如何得到精确的配准结果及怎样提高算法的执行速度是目前医学图像配准需要解决的问题。

形变模型的选择对配准过程非常重要,因为选择合适的形变模型是在计算效率和描述图像局部形变多少之间取得一个折中。配准方法利用优化策略估计的参数应对选取的形变模型的自由度。自由度的数量变化很大,从全局刚体变换的6个增加到弹性形变的数百万个。选择的自由度个数越多,对图像本身局部形变的描述也越详细,得到的配准效果就越好,随之带来的计算成本越高;然而选择的自由度少,图像中有些区域的局部形变就会缺失,从而影响图像配准的结果。常用的弹性几何变换模型可以分为四大类:基于物理模型、基于插值和估计理论、基于知识和特定任务约束的形变模型<sup>[7]</sup>。

不同的几何变换模型表示图像之间不同的映射关系,图3-19展示了几种常见的空间变换。当然对于不同的图像映射关系,要选取合适的变换模型,这样既能提高配准的精度,又能提高配准效率。

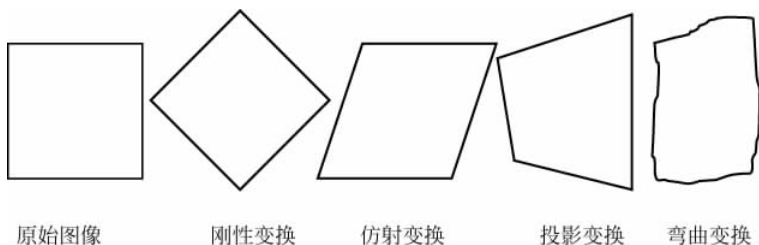


图 3-19 几何变换模型示意图

## 3.4 相似性度量

相似性度量是医学图像配准算法的重要组成部分。从数学意义上讲,相似性度量是测量某些信息、图像、数据或形状相似程度的一个函数。在医学图像配准领域,相似性度量主要用来衡量两幅医学图像间的相似度,是优化过程中目标函数的重要组成部分,它的选择取决于待配准图像中提取的特征信息。相似性度量的值随着空间变换参数的选择而发生变化,当相似度达到最大时,浮动图像与参考图像完全配准。在医学图像配准中,常用的相似性度量有差值平方和(Sum of Squared Differences, SSD)、绝对误差和<sup>[8]</sup>(Sum of Absolute Differences, SAD)、归一化互相关<sup>[9]</sup>(Normalized Cross Correlation, NCC)、相关比率<sup>[10]</sup>(Correlation ratio)及互信息<sup>[11]</sup>(Mutual Information, MI)等。

### 3.4.1 差值平方和

设参考图像和浮动图像分别为  $R$ 、 $F$ ,以三维图像为例, $r(x_i, y_i, z_i)$ 表示参考图像中坐标点  $(x_i, y_i, z_i)$  位置处的灰度值,浮动图像中对应点的坐标在空间变换  $T_\mu$  作用下的位置为  $T_\mu(x_i, y_i, z_i)$ ,  $f(T_\mu(x_i, y_i, z_i))$  为浮动图像中该点的灰度值,由此得到两幅图像的差值平方和:

$$S_{SSD} = \sum_{i=1}^N (f(T_\mu(x_i, y_i, z_i)) - r(x_i, y_i, z_i))^2 \quad (3.35)$$

式中  $N$  表示图像的尺寸,即图像中像素点的数目, $\mu$  为空间变换参数,例如选择三维刚体变换, $\mu$  是包含 6 个参数的矢量,即在 3 个平面内的平移和绕 3 个坐标轴的旋转。

### 3.4.2 绝对误差和

三维图像的绝对误差和定义如下:

$$S_{SAD} = \sum_{i=1}^N |f(T_\mu(x_i, y_i, z_i)) - r(x_i, y_i, z_i)| \quad (3.36)$$

上式的值越大,表明两幅图像的相似度越低,反之,上式的数值越小,两幅图像的相似度越高。绝对误差和与差值平方和都是建立在图像对应点灰度差异不大的基础上,因此,这两种相似性度量对图像的灰度变化比较敏感。对于存在灰度差异较大的图像,尤其对不同模态的医学图像,SSD 和 SAD 不能很好地衡量待配准图像间的相似程度。

### 3.4.3 归一化互相关

归一化互相关相似性度量通过计算参考图像和浮动图像间的互相关值,来衡量待配准图像的相似程度,三维图像的归一化互相关可以表示为:

$$S_{\text{NCC}} = \frac{\sum_{i=1}^N (f(T_{\mu}(x_i, y_i, z_i)) - \bar{f})(r(x_i, y_i, z_i) - \bar{r})}{\sqrt{\sum_{i=1}^N (f(T_{\mu}(x_i, y_i, z_i)) - \bar{f})^2} \sqrt{\sum_{i=1}^N (r(x_i, y_i, z_i) - \bar{r})^2}} \quad (3.37)$$

式中

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f(T_{\mu}(x_i, y_i, z_i)), \quad \bar{r} = \frac{1}{N} \sum_{i=1}^N r(x_i, y_i, z_i) \quad (3.38)$$

分别表示浮动图像和参考图像灰度的平均值。式(3.37)的值越大,表明两幅图像的相关性越大,相似度越高;反之相关性越小,相似度越低。当参考图像和浮动图像间的归一化互相关数值最大时,两幅图像完全配准。

### 3.4.4 相关比率

设参考图像和浮动图像为  $R, F$ , 空间变换为  $T_{\mu}$ ,  $F(T_{\mu})$  是在变换  $T_{\mu}$  作用下的浮动图像,  $F(T_{\mu})$  和  $R$  的相关比率为:

$$\eta(F(T_{\mu}) | R) = \frac{\text{Var}[E(F(T_{\mu}) | R)]}{\text{Var}(F(T_{\mu}))} \quad (3.39)$$

上式等价于

$$\eta(F(T_{\mu}) | R) = 1 - \frac{\text{Var}[F(T_{\mu}) - E(F(T_{\mu}) | R)]}{\text{Var}(F(T_{\mu}))} \quad (3.40)$$

式中  $\text{Var}$  表示方差操作。相关比率衡量变换的浮动图像  $F(T_{\mu})$  和参考图像  $R$  之间的依赖性,取值为  $0 \sim 1$ ,数值越趋向于  $0$ ,表明两幅图像依赖性越小,越接近  $1$  依赖性越大。需要注意的是,相关比率是一个不对称的测度,即  $\eta(F|R) \neq \eta(R|F)$ ,因为在图像配准中,参考图像和浮动图像作用不同,空间变换  $T_{\mu}$  只作用于浮动图像  $F$ ,后面章节会详细介绍基于相关比率的医学图像配准算法。

### 3.4.5 互信息测度

互信息衡量随机变量  $X$  和  $Y$  之间的依赖程度,用来测量联合概率分

布与二者完全独立时的分布之间的距离,使用 KL 散度(或称作相对熵, Relative entropy)<sup>[12]</sup>来定义  $X$  与  $Y$  的互信息:

$$I(X, Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x) \cdot p(y)} \quad (3.41)$$

互信息与联合熵、边缘熵和条件熵之间有着紧密的关系。根据联合熵及条件熵的定义,式(3.41)可以写成:

$$\begin{aligned} I(X, Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X | Y) \\ &= H(Y) - H(Y | X) \end{aligned} \quad (3.42)$$

由式(3.42)可以看出,互信息表示随机变量  $X$  中包含关于  $Y$  的信息量的多少,同时也代表  $Y$  中包含  $X$  的信息量,因此互信息是对称的,即  $I(X, Y) = I(Y, X)$ 。如果  $X$  与  $Y$  相互独立,则  $I(X, Y) = 0$ ; 若二者一对一相关,则  $I(X, Y) = H(X) = H(Y)$ 。除了对称性之外,互信息还具有非负性、有界性等。

## 3.5 搜索策略

### 3.5.1 Powell 优化算法

Powell 优化算法是一种有效的直接搜索方法,根据线性最小化的思路,可以想到求解一般多维最小化问题的简单方法:考虑一组单位矢量  $e_0, e_1, \dots, e_{N-1}$  作为方向集。利用线性最小化方法,求第一个搜索方向的最小值,然后再求第二个搜索方向的最小值,以此类推,循环所有的搜索方向,直到目标函数停止下降。

实际上,这种简单的方法对很多目标函数有效,即使它对一些其他目标函数效果很差。考虑一个二维目标函数,它的轮廓图如图 3-20 所示,轮廓图中存在一个长的狭窄的山谷,那么在每个阶段,都需要一系列很多小的步骤接近基矢量。更普遍地,对于  $N$  个搜索方向,如果在一些方向上,目标函数二阶导数的模比其他方向上的大,那么需要在所有  $N$  个基矢量上做多次循环。

很明显,我们所需要的是比单位矢量  $e_i$  更好的搜索方向集。所有的方向集方法都应该具有随着优化的执行,自动更新搜索方向集的能力;而且能够远离狭窄的山谷,或者包含一些“无干扰”(noninterfering)的搜索方向。

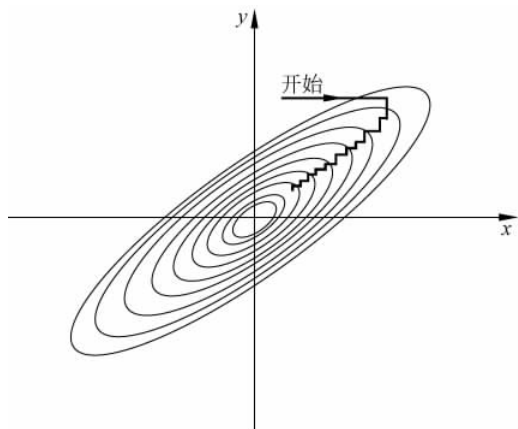


图 3-20 在一个长而狭窄的山谷内沿着坐标方向连续的最小化

### 1. 共轭方向

“无干扰”方向通常也称作共轭方向，下面从数学角度进行解释。首先，如果沿着一些方向  $\mathbf{u}$  对函数执行最小化，在线性最小化中该函数的梯度必须垂直于  $\mathbf{u}$ ；否则，沿着  $\mathbf{u}$  方向将仍然存在一个非零的方向导数。

接着考虑某个特定点  $P$ ，然后任何函数  $f$  用泰勒级数展开

$$f(x) = f(P) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \dots$$

$$\approx c - b \cdot x + \frac{1}{2} x \cdot \mathbf{A} \cdot x \tag{3.43}$$

这里

$$c \equiv f(P) \quad b \equiv -\nabla f|_P \quad [A]_{ij} \equiv \frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_P \tag{3.44}$$

矩阵  $\mathbf{A}$  的元素是函数的二阶导数，即函数在点  $P$  处的 Hessian 矩阵。

在式(3.43)的近似中， $f$  的梯度容易计算：

$$\nabla f = \mathbf{A} \cdot x - b \tag{3.45}$$

沿着某个搜索方向移动时，函数的梯度会怎样变化呢？很明显有

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta x) \tag{3.46}$$

假定我们已经沿着  $\mathbf{u}$  方向移动，现在再沿着某个新方向  $\mathbf{v}$  移动。沿  $\mathbf{v}$  方向的移动并不影响沿  $\mathbf{u}$  方向的最小化，梯度依旧垂直于  $\mathbf{u}$  方向，也就是说，梯度的变化也垂直于  $\mathbf{u}$ ，那么

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v} \tag{3.47}$$

当上式成立时,称矢量  $\mathbf{u}$  和  $\mathbf{v}$  是共轭的。当一个矢量集中所有元素都对满足上述关系时,我们就称之为一个共轭集。如果沿着一个共轭方向集对函数执行连续的最小化,那么就不需要重新执行其他的方向。一个方向集方法的成功在于  $N$  个方向之间线性无关、相互共轭。

## 2. Powell 二次收敛方法

Powell 首先提出了包含  $N$  个互相共轭方向的方向集方法。将方向集  $\mathbf{u}_i$  初始化为基矢量,

$$\mathbf{u}_i = \mathbf{e}_i \quad i = 0, 1, \dots, N-1 \quad (3.48)$$

重复下面的步骤,直至函数停止下降:

- 设定起始位置  $P_0$ 。
- 沿着方向  $\mathbf{u}_i$  移动  $P_i$  至最小值,  $i=0, 1, \dots, N-1$ , 记为  $p_{i+1}$ 。
- 对于  $i=0, 1, \dots, N-2$ , 将  $\mathbf{u}_i$  设置成  $\mathbf{u}_{i+1}$ 。
- 将  $P_N - P_0$  设置成  $\mathbf{u}_{N-1}$ 。
- 沿着方向  $\mathbf{u}_{N-1}$  移动  $P_N$  至最小值, 记为  $P_0$ 。

1964年, Powell 指出, 对于二次形式, 上述过程的  $k$  次迭代产生方向集  $\mathbf{u}_i$ , 其中最后  $k$  个元素相互共轭。因此,  $N$  次迭代一共产生  $N(N+1)$  次线性最小化, 正好是对一个二次形式进行最小化, Brent 也证明该方法是可行的。

不幸的是, Powell 二次收敛算法存在一个问题。在每一阶段中, 舍弃  $\mathbf{u}_0$  采用  $P_N - P_0$  的过程会产生互相折叠的方向, 从而变成线性相关。一旦遇到这个问题, 函数的最小化过程将给出错误的结果。有以下几种方法来处理 Powell 算法中的线性相关问题:

- (1) 可以在  $N$  或者  $N+1$  次迭代后重新将  $\mathbf{u}_i$  初始化为  $\mathbf{e}_i$ 。
- (2) Brent 指出可以将方向集等价地设置成任意正交矩阵的列矢量。他将计算的矩阵  $\mathbf{A}$  的主方向重置为方向集, 而不是舍弃共轭方向的信息。
- (3) 放弃二次收敛的特性, 选择一个启发式的方案, 搜索一些沿狭窄山谷的方向。

## 3. 舍弃最大下降方向

改进的 Powell 方法的基本思想仍然是将  $P_N - P_0$  作为新的搜索方向, 这是在尝试所有  $N$  个可能的方向后的平均方向。对于一个长方向扭转很慢的山谷来说, 这个方向可能给我们一个好的执行。这种变换就是舍弃旧方向, 沿该方向目标函数的下降幅度最大。既然这个方向是目前迭代的最佳方向, 这似乎是矛盾的。下面给出一些解释, 有时增加一个新方向并不可行。定义

$$f_0 \equiv f(P_0) \quad f_N \equiv f(P_N) \quad f_E \equiv f(2P_N - P_0) \quad (3.49)$$

这里  $f_E$  是离提出新方向有些远的一个推测点的函数值,此外定义  $\Delta f$  是沿目前迭代过程的某个特定方向的最大下降幅度( $\Delta f$  是一个正数)。然后:

(1) 如果  $f_E \geq f_0$ , 保持旧方向集, 因为平均方向  $P_N - P_0$  都失去作用。

(2) 如果  $2(f_0 - 2f_N + f_E)[(f_0 - f_N) - \Delta f]^2 \geq (f_0 - f_E)^2 \Delta f$ , 仍然保持旧方向集, 一方面, 由于任何单个方向的下降, 使得沿平均方向的下降并不主要; 另一方面, 沿着平均方向存在大量的二阶导数, 很接近目标函数的最小值。

### 3.5.2 Simplex 优化算法

单纯形方法<sup>[13]</sup>也叫内尔德-米德优化(Nelder-Mead optimization, NM)方法, 是一种不需要计算目标函数导数的优化方法, 本章将采用 NM 方法来获得目标函数的最小值。下面根据 NM 方法的几何特性来描述其优化过程。

一个单纯形可以看作几何图形, 在  $N$  维空间中, 该几何图形包括  $N+1$  个顶点、所有这些点之间的线段及多边形的面等。在二维环境中, 单纯形是三角形, 而对于三维情况, 它为四面体。通常, 我们感兴趣的是非退化的单纯形, 如那些封闭在一个有限的内部  $N$  维容积内的单纯形。若一个非退化单纯形中的任何点被当作原点, 那么其他的  $N$  个点就定义了跨度为  $N$  维矢量空间的矢量方向。

在一维最小化中, 它可能包括一个最小值, 以保证随后的孤立能够成功。而在多维空间中并不存在类似的过程。对于多维最小化, 我们最需要的是给优化算法一个初始值, 即用包括  $N$  个独立变量的矢量作为优化的初始点。然后, 优化算法通过  $N$  维拓扑难以想象的复杂度按照自己的方式下降, 直到算法找到一个最小值。

下坡单纯形法不一定必须从一个单一点开始, 也可以从  $N+1$  个点出发, 定义一个初始的单纯形。假设从这些点中选一个点作为初始点  $P_0$ , 可以通过如下公式得到其他  $N$  个点,

$$P_i = P_0 + \Delta e_i \quad (3.50)$$

式中  $e_i$  为  $N$  个单位矢量,  $\Delta$  是一个常量。NM 方法需要获取一些步长, 大部分的步长仅仅是将单纯形上函数值最大的点(最高点)穿过单纯形的反面移动至函数值较小的点(较低点)。这些步长被称作“反射”

(Reflections), 构建这些反射以保存纯形的体积, 由此才能维持纯形的非退化性。当这些完成时, 算法在一个或另外的方向上膨胀纯形来获得更大的步长。当它到达谷底时, 算法在相反的方向上收缩纯形, 使它从谷底渗出。当纯形试图穿过一个针眼时, 它将会在所有方向上收缩, 使其停在低点附近, 纯形的基本移动见图 3-21。

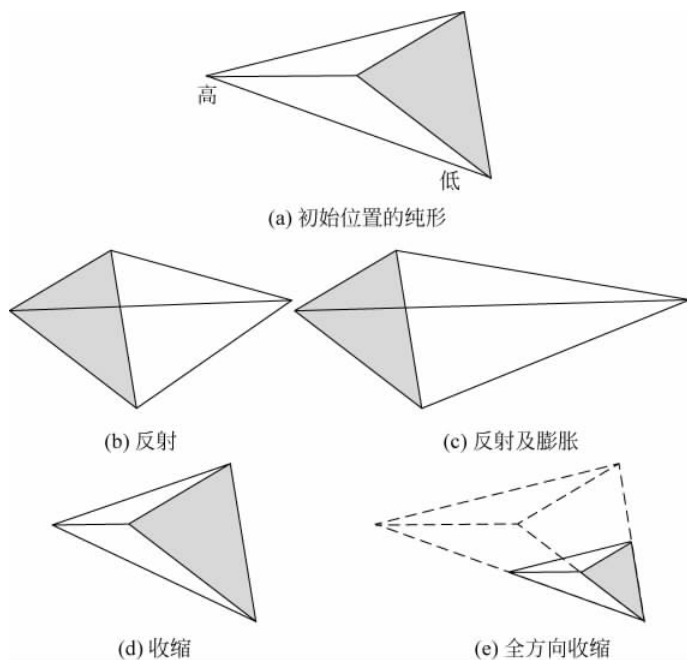


图 3-21 纯形的移动示意图

### 3.5.3 共轭梯度法

共轭梯度法是在每一迭代步采用当前点处的最速下降方向来生成关于凸二次函数  $f$  的 Hessian 矩阵  $\mathbf{G}$  的共轭方向, 并建立求  $f$  在  $\mathbf{R}^n$  上的极小点的方法。这一方法最早由 Hestenes 和 Stiefel 于 1952 年为求解对称正定线性方程组而提出来的, 后经 Fletcher 等人研究并应用于无约束优化问题并取得了丰富的成果, 共轭梯度法也因此成为当前求解无约束优化问题的重要算法类。

考虑求解正定二次目标函数的极小点, 设

$$\min f(x) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \quad (3.51)$$

其中  $\mathbf{G}$  是  $n$  阶对称正定矩阵,  $\mathbf{b}$  为  $n$  维常矢量,  $c$  为常数, 则  $f$  的梯度和 Hessian 矩阵分别为

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{G}\mathbf{x} + \mathbf{b}, \quad \mathbf{G}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \mathbf{G} \quad (3.52)$$

取初始方向  $\mathbf{d}_0$  为初始点  $\mathbf{x}_0$  处的负梯度方向, 即

$$\mathbf{d}_0 = -\nabla f(\mathbf{x}_0) = -\mathbf{g}_0 \quad (3.53)$$

从  $\mathbf{x}_0$  出发沿  $\mathbf{d}_0$  方向进行精确线搜索, 需求步长  $\alpha_0$ , 令

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 \quad (3.54)$$

其中  $\alpha_0$  满足

$$\nabla f(\mathbf{x}_1)^T \mathbf{d}_0 = \mathbf{g}_1^T \mathbf{d}_0 = 0 \quad (3.55)$$

在  $\mathbf{x}_1$  处, 用  $f$  的负梯度方向  $-\mathbf{g}_1$  与  $\mathbf{d}_0$  组合生成  $\mathbf{d}_1$ ,

$$\mathbf{d}_1 = -\mathbf{g}_1 + \beta_0 \mathbf{d}_0 \quad (3.56)$$

然后取系数  $\beta_0$  使  $\mathbf{d}_1$  与  $\mathbf{d}_0$  关于  $\mathbf{G}$  共轭, 即令

$$\mathbf{d}_1^T \mathbf{G} \mathbf{d}_0 = 0 \quad (3.57)$$

确定  $\beta_0$ , 将式(3.56)代入式(3.57), 得

$$\beta_0 = \frac{\mathbf{g}_1^T \mathbf{G} \mathbf{d}_0}{\mathbf{d}_0^T \mathbf{G} \mathbf{d}_0} \quad (3.58)$$

由式(3.52), 得

$$\mathbf{g}_1 - \mathbf{g}_0 = \mathbf{G}(\mathbf{x}_1 - \mathbf{x}_0) = \alpha_0 \mathbf{G} \mathbf{d}_0 \quad (3.59)$$

由式(3.54)~式(3.56), 可得

$$\mathbf{g}_2^T \mathbf{g}_0 = 0, \quad \mathbf{g}_2^T \mathbf{g}_1 = 0, \quad \mathbf{d}_0^T \mathbf{g}_0 = -\mathbf{g}_0^T \mathbf{g}_0, \quad \mathbf{d}_1^T \mathbf{g}_1 = -\mathbf{g}_1^T \mathbf{g}_1 \quad (3.60)$$

设已得到相互共轭的搜索方向  $\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{k-1}$ , 精确线搜索得到的步长为  $\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{k-1}$ , 且满足

$$\begin{cases} \mathbf{d}_{k-1}^T \mathbf{G} \mathbf{d}_i = 0, & i = 0, 1, \dots, k-2 \\ \mathbf{d}_i^T \mathbf{g}_i = -\mathbf{g}_i^T \mathbf{g}_i, & i = 0, 1, \dots, k-1 \\ \mathbf{g}_k^T \mathbf{g}_i = 0, \mathbf{g}_k^T \mathbf{d}_i = 0, & i = 0, 1, \dots, k-1 \end{cases} \quad (3.61)$$

令

$$\mathbf{d}_k = -\mathbf{g}_k + \beta_{k-1} \mathbf{d}_{k-1} + \sum_{i=0}^{k-2} \beta_k^{(i)} \mathbf{d}_i \quad (3.62)$$

其中  $\beta_{k-1}, \beta_k$  的选择要满足

$$\mathbf{d}_k^T \mathbf{G} \mathbf{d}_i = 0, \quad i = 0, 1, \dots, k-1 \quad (3.63)$$

由式(3.62), 可得

$$\beta_{k-1} = \frac{\mathbf{g}_k^T \mathbf{G} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{G} \mathbf{d}_{k-1}}, \quad \beta_k^{(i)} = \frac{\mathbf{g}_k^T \mathbf{G} \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{G} \mathbf{d}_i}, \quad i = 0, 1, \dots, k-2 \quad (3.64)$$

类似于式(3.59),

$$\mathbf{g}_{i+1} - \mathbf{g}_i = \mathbf{G}(\mathbf{x}_{i+1} - \mathbf{x}_i) = \alpha_i \mathbf{G} \mathbf{d}_i, \quad i = 0, 1, \dots, k-1 \quad (3.65)$$

由归纳法假设式(3.61)可得

$$\beta_k^{(i)} = \frac{\mathbf{g}_k^T \mathbf{G} \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{G} \mathbf{d}_i} = \frac{\mathbf{g}_k^T (\mathbf{g}_{i+1} - \mathbf{g}_i)}{\mathbf{d}_i^T (\mathbf{g}_{i+1} - \mathbf{g}_i)} = 0, \quad i = 0, 1, \dots, k-2 \quad (3.66)$$

于是,第  $k$  步的搜索方向为

$$\mathbf{d}_k = -\mathbf{g}_k + \beta_{k-1} \mathbf{d}_{k-1} \quad (3.67)$$

其中  $\beta_{k-1}$  由式(3.64)确定,即

$$\beta_{k-1} = \frac{\mathbf{g}_k^T \mathbf{G} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{G} \mathbf{d}_{k-1}} \quad (3.68)$$

这样式(3.54)、式(3.67)和式(3.68)就确定了一组由负梯度方向形成的共轭方向,而把沿着这组方向进行迭代的方法称作共轭梯度法。

### 3.5.4 拟牛顿法

牛顿法的优点是具有二阶收敛速度,但当 Hessian 矩阵不正定时,不能保证所产生的方向是目标函数在  $x_k$  处的下降方向。特别地,当 Hessian 矩阵奇异时,算法就无法继续进行下去。尽管修正牛顿法可以克服这一缺陷,但其中的修正参数  $\mu_k$  的选取很难把握,过大或过小都会影响到收敛速度。此外,牛顿法的每一迭代步都需要目标函数的二阶导数,即计算 Hessian 矩阵,对于大规模的问题,其计算量惊人。本节介绍的拟牛顿法克服了这些缺点,并且在一定条件下仍然具有较快的收敛速度。

拟牛顿法的基本思想是在基本牛顿法中用 Hessian 矩阵  $\mathbf{G}_k$  的某个近似矩阵  $\mathbf{B}_k$  取代  $\mathbf{G}_k$ ,通常情况下,  $\mathbf{B}_k$  应该具有 3 个特点:

(1) 在某种意义上有  $\mathbf{B}_k \approx \mathbf{G}_k$ , 使相应算法产生的方向近似于牛顿方向,以确保具有较快的收敛速度。

(2) 对于所有的  $k$ ,  $\mathbf{B}_k$  是对称正定的,使算法产生的方向是函数  $f$  在  $x_k$  处的下降方向。

(3) 矩阵  $\mathbf{B}_k$  更新规则相对比较简单,通常采用秩 1 或秩 2 矩阵进行校正。

下面介绍满足上述条件的  $\mathbf{B}_k$  矩阵的构造方法。设  $f: \mathbf{R}^n \rightarrow \mathbf{R}$  在开集  $D \subset \mathbf{R}^n$  上二次连续可微。那么,  $f$  在  $x_{k+1}$  处的二次近似模型为

$$f(\mathbf{x}) \approx f(\mathbf{x}_{k+1}) + \mathbf{g}_{k+1}^T (\mathbf{x} - \mathbf{x}_{k+1}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_{k+1})^T \mathbf{G}_{k+1} (\mathbf{x} - \mathbf{x}_{k+1}) \quad (3.69)$$

对上式求导可得

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}_{k+1} + \mathbf{G}_{k+1} (\mathbf{x} - \mathbf{x}_{k+1}) \quad (3.70)$$

令  $\mathbf{x} = \mathbf{x}_k$ , 位移  $s_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ , 梯度差  $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ , 则有

$$\mathbf{G}_{k+1} s_k \approx \mathbf{y}_k \quad (3.71)$$

注意到, 对于二次函数  $f$ , 上式是精确成立的。现在, 要求在拟牛顿法中构造出 Hessian 矩阵的近似矩阵  $\mathbf{B}_k$  满足这种关系, 即

$$\mathbf{B}_{k+1} s_k \approx \mathbf{y}_k \quad (3.72)$$

上式通常称作拟牛顿方程或拟牛顿条件。令  $\mathbf{H}_{k+1} = \mathbf{B}_{k+1}$  的逆矩阵, 得

$$\mathbf{H}_{k+1} \mathbf{y}_k \approx s_k \quad (3.73)$$

其中  $\mathbf{H}_{k+1}$  是 Hessian 矩阵的近似。搜索方向由  $dk = -\mathbf{H}_k \mathbf{g}_k$  或者确定, 根据  $\mathbf{B}_k$  的第三个特点, 可令

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{E}_k, \quad \mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{D}_k \quad (3.74)$$

其中  $\mathbf{E}_k, \mathbf{D}_k$  是秩 1 或秩 2 矩阵。

### 1. BFGS 算法

BFGS 算法是目前最流行也是最有效的拟牛顿校正算法, 它是由 Broyden、Fletcher、Goldfarb 和 Shanno 在 1970 年各自独立提出的拟牛顿法, 故称为 BFGS 算法。其基本思想是: 在式 (3.74) 中取修正矩阵  $\mathbf{E}_k$  为秩 2 矩阵

$$\mathbf{E}_k = \alpha \mathbf{u}_k \mathbf{u}_k^T + \beta \mathbf{v}_k \mathbf{v}_k^T \quad (3.75)$$

其中  $\mathbf{u}_k, \mathbf{v}_k \in \mathbf{R}^n$  是待定矢量,  $\alpha, \beta \in \mathbf{R}$  是待定实数。由式 (3.72) 可得

$$(\mathbf{B}_k + \alpha \mathbf{u}_k \mathbf{u}_k^T + \beta \mathbf{v}_k \mathbf{v}_k^T) s_k = \mathbf{y}_k \quad (3.76)$$

满足上式的矢量  $\mathbf{u}_k$  和  $\mathbf{v}_k$  不唯一, 可取  $\mathbf{u}_k$  和  $\mathbf{v}_k$  分别平行于  $\mathbf{B}_k s_k$  和  $\mathbf{y}_k$ , 即令  $\mathbf{u}_k = \boldsymbol{\gamma}, \mathbf{v}_k = \boldsymbol{\theta}$ , 其中  $\boldsymbol{\gamma}, \boldsymbol{\theta}$  是待定参数, 于是有

$$\mathbf{E}_k = \alpha \boldsymbol{\gamma}^2 \mathbf{B}_k s_k s_k^T \mathbf{B}_k + \beta \boldsymbol{\theta}^2 \mathbf{y}_k \mathbf{y}_k^T \quad (3.77)$$

代入式 (3.76) 整理得

$$(\alpha \boldsymbol{\gamma}^2 s_k^T \mathbf{B}_k s_k + 1) \mathbf{B}_k s_k + (\beta \boldsymbol{\theta}^2 \mathbf{y}_k^T s_k - 1) \mathbf{y}_k = 0 \quad (3.78)$$

令上式两项分别等于 0, 得

$$\alpha \boldsymbol{\gamma}^2 = -\frac{1}{s_k^T \mathbf{B}_k s_k}, \quad \beta \boldsymbol{\theta}^2 = \frac{1}{\mathbf{y}_k^T s_k} \quad (3.79)$$

从而得到 BFGS 算法的秩 2 修正公式

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k s_k s_k^T \mathbf{B}_k}{s_k^T \mathbf{B}_k s_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T s_k} \quad (3.80)$$

显然, 由上式可知, 若  $\mathbf{B}_k$  对称, 校正后的  $\mathbf{B}_{k+1}$  也对称。

### 2. DFP 算法

DFP 是第一个拟牛顿校正, 是 1959 年由 Davidon 提出的, 后经 Fletcher 和 Powell 解释和改进, 因此称作 DFP 算法, 它也是求解无约束

优化问题最有效的算法之一。类似于 BFGS 校正公式的推导,可得 DFP 校正公式如下:

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{\mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} \quad (3.81)$$

同样地,若  $\mathbf{H}_k$  对称,校正后的  $\mathbf{H}_{k+1}$  也对称。设  $\mathbf{H}_k$  对称正定,那么  $\mathbf{H}_{k+1}$  对称正定的充要条件是

$$\mathbf{s}_k^T \mathbf{y}_k > 0 \quad (3.82)$$

当采用精确线搜索或 wolfe 搜索准则时,上式条件可以被满足。但一般来说,Armijo 搜索不能满足这一条件,需要做如下修正:

$$\mathbf{H}_{k+1} = \begin{cases} \mathbf{H}_k, & \mathbf{s}_k^T \mathbf{y}_k \leq 0 \\ \mathbf{H}_k - \frac{\mathbf{H}_k \mathbf{y}_k \mathbf{y}_k^T \mathbf{H}_k}{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k} + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}, & \mathbf{y}_k^T \mathbf{s}_k > 0 \end{cases} \quad (3.83)$$

### 3.5.5 随机梯度下降法

梯度下降算法将目标函数的负梯度作为搜索方向

$$\mu_{k+1} = \mu_k - a_k g(\mu_k) \quad (3.84)$$

这里  $g(\mu_k)$  表示在当前位置  $\mu_k$  处估计的目标函数的导数。随机梯度下降算法采用与梯度下降相同的策略,目的仍然是要得到目标函数的导数,然而该算法采用  $g(\mu_k)$  的一个估计作为搜索方向:

$$\mu_{k+1} = \mu_k - a_k \tilde{g}_k \quad (3.85)$$

如果上述估计误差趋于 0,就能保证上式的收敛:

$$E(\tilde{g}_k) \rightarrow g(\mu_k), \quad k \rightarrow \infty \quad (3.86)$$

式中  $E$  表示期望运算。当目标函数导数的计算成本太高时,通常采用随机梯度下降算法。利用目标函数导数的估计能够减少每次迭代的计算时间,但同时也可能对收敛速度造成一些影响。下面介绍 3 种随机梯度下降算法: KW、SP 和 RM。

**Kiefer-Wolfowitz (KW):** 该方法基于导数的有限差分估计:

$$[\tilde{g}_k]_i = \frac{C(\mu_k + c_k \mathbf{e}_i) - C(\mu_k - c_k \mathbf{e}_i)}{2c_k} \quad (3.87)$$

这里  $[\tilde{g}_k]_i$  表示  $\tilde{g}_k$  的第  $i$  个元素,  $c_k$  是一个标量,  $\mathbf{e}_i$  是除了第  $i$  个元素为 1 其余都为 0 的单位矢量。KW 方法假设目标函数值的近似为:

$$\tilde{C}_{ki}^+ = C(\mu_k + c_k \mathbf{e}_i) + \epsilon_{ki}^+ \quad (3.88)$$

$$\tilde{C}_{ki}^- = C(\mu_k - c_k \mathbf{e}_i) + \epsilon_{ki}^- \quad (3.89)$$

式中  $\epsilon_{ki}^+$  和  $\epsilon_{ki}^-$  表示近似误差,将式(3.88)和式(3.89)代入式(3.87),可得:

$$[\tilde{g}_k]_i = \frac{\tilde{C}_{ki}^+ - \tilde{C}_{ki}^-}{2c_k} \quad (3.90)$$

对于导数的近似,除了有限差分方法引入的近似误差之外,还要考虑一个额外的误差,即式(3.88)和式(3.89)中的 $\epsilon$ 项。而对于 $c_k$ ,通常用下式表示:

$$c_k = c / (k + 1)^\gamma \quad (3.91)$$

这里 $c$ 和 $\gamma$ 都是用户指定的常数, $c > 0, 0 \leq \gamma \leq 1$ 。需要注意,对于一个 $N$ 维的优化问题,KW方法的每次迭代需要估计 $2N$ 次目标函数值。在实际配准应用中,可以利用三次B样条的紧支集来减小计算成本<sup>[17]</sup>。

**Simultaneous Perturbation (SP):** 该方法在文献[18]中首次提出,同样是基于目标函数近似估计的方法。对于每次迭代,KW算法要求 $2N$ 次估计,然而SP方法仅用两次估计,与维数 $N$ 无关:

$$\tilde{C}_k^+ = C(\mu_k + c_k \Delta_k) + \epsilon_k^+ \quad (3.92)$$

$$\tilde{C}_k^- = C(\mu_k - c_k \Delta_k) + \epsilon_k^- \quad (3.93)$$

这里 $\Delta_k$ 表示随机扰动矢量,每次迭代中,扰动矢量中的每个元素都随机等概率地设置为 $\pm 1$ 。 $\epsilon$ 项表示近似误差。导数矢量的每个元素由下式计算:

$$[\tilde{g}_k]_i = \frac{\tilde{C}_k^+ - \tilde{C}_k^-}{2c_k [\Delta_k]_i} \quad (3.94)$$

标量 $c_k$ 通过式(3.91)定义。

**Robbins-Monro (RM):** KW和SP都是采用基于目标函数近似估计的一种导数估计方法,而RM算法并未指明怎么计算导数,而是直接给出目标函数的一个估计:

$$\tilde{g}_k = g(\mu_k) + \epsilon_k \quad (3.95)$$

事实上,KW和SP是RM算法的特殊情况,如果在每次迭代中, $\epsilon$ 项等于0,那么该算法就相当于梯度下降方法。

## 3.6 本章小结

本章主要介绍了图像配准算法的4个基本组成部分。首先从图像的点、线、边缘等基本特征出发,描述了几种常用的特征检测方法,特征空间的选择决定了用于图像配准的特征类型。其次介绍了刚体变换、仿射变换、投影变换及弯曲变换等空间变换,变换类型决定了优化的参数数量,

变换类型越复杂,要优化的参数数量越多,反之参数数量越少。然后概述了插值平方和、互相关及互信息等几种常用的相似性度量,相似度用来衡量参考图像与待配准图像间的相似程度,为配准的下一步提供依据。最后描述了医学图像配准中常用的几种优化算法,若优化条件满足,则配准过程停止,得到最佳变换参数。

## 参考文献

- [1] Brown L G. A survey of image registration techniques [J]. *ACM Computing Surveys*,1992,24(4): 325-376.
- [2] Rafael C Gonzalez, Richard E Woods, Steven L Eddins, 等. 数字图像处理[M]. 北京: 电子工业出版社,2014.
- [3] 李朝晖,张弘. 数字图像处理及应用[M]. 北京: 机械工业出版社,2004.
- [4] C. Harris, M. Stephens. A Combined Corner and Edge Detector [C]. *Proceedings of the 4th Alvey Vision Conference, Manchester*,1988,147-151.
- [5] D. Lowe. Object recognition from local scale-invariant features [C]. in: *ICCV*,1999.
- [6] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF) [J]. *Computer vision and image understanding*,2008,110(3): 346-359.
- [7] Sotiras A, Davatzikos C, Paragios N. Deformable medical image registration: A survey [J]. *IEEE Transactions on Medical Imaging*,2013,32(7): 1153-1190.
- [8] Hoh C K, Dahlbom M, Harris G, et al. Automated iterative three-dimensional registration of positron emission tomography images [J]. *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 1993, 34 (11): 2009-2018.
- [9] Lemieux L, Jagoe R, Fish D R, et al. A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs [J]. *Medical physics*,1994,21(11): 1749-1760.
- [10] Roche A, Malandain G, Pennec X, et al. The correlation ratio as a new similarity measure for multimodal image registration [C]. In: *Medical Image Computing and Computer-Assisted Intervention—MICCAI' 98*. Springer Berlin Heidelberg,1998: 1115-1124.
- [11] Viola P, Wells III W M. Alignment by maximization of mutual information [C]. In: *International Conference on Computer Vision (ICCV)*. IEEE Computer Society Press, Los Alamitos, CA, 1995: 16-23.
- [12] Cover T M, Thomas J A. *Elements of information theory* [M]. John Wiley & Sons,2012.
- [13] Nelder J A, Mead R. A simplex method for function minimization [J]. *The*

- computer journal,1965,7(4): 308-313.
- [14] J Nocedal, S J Wright. Numerical Optimization. New York: Springer-Verlag,1999.
- [15] H. J. Kushner, G. G. Yin, Stochastic Approximation and Recursive Algorithms and Applications,2nd ed. New York: Springer-Verlag,2003.
- [16] Brent, R P. Algorithms for Minimization without Derivatives (Englewood Cliffs,NJ; PrenticeHall),1973; reprinted 2002 (New York; Dover),Chapter 7. [1].
- [17] Klein S, Staring M, Pluim J P W. Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines[J]. IEEE transactions on image processing,2007,16(12): 2879-2890.
- [18] Spall J C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation [J]. IEEE transactions on automatic control,1992,37(3): 332-341.