代数多重网格方法原理及图像工程应用

黄　颖　李伟生　周丽芳　尚凤军　高艺铭　著

電子工業出版社

Publishing House of Electronics Industry

北京・BEIJING

内 容 简 介

本书以代数多重网格方法为主题，对代数多重网格方法的理论进行了初步探讨，并将其应用到图像工程的多个领域，如清晰度检测、图像融合、特征提取和特征表达、视频检测以及物体识别之中。本书以理论分析为基础，并结合应用实践，验证了代数多重网格方法在多个应用领域的效果，并且还能在其他更多的应用领域得到延伸，具有一定的理论和应用价值。本书所有应用实例均为作者所在研究团队科研工作的总结，具有一定的前沿性和实用性。

本书可作为高等院校计算机、电子、通信、自动化等专业研究生和高年级本科生的教材，也可以作为计算机信息处理、自动控制等相关领域的工程专业技术人员的参考用书。

图书在版编目（CIP）数据

代数多重网格方法原理及图像工程应用 / 黄颖等著. —北京：电子工业出版社，2015.7

ISBN 978-7-121-26233-3

Ⅰ. ①代…　Ⅱ. ①黄…　Ⅲ. ①代数－网格－研究　Ⅳ. ①O243

中国版本图书馆 CIP 数据核字（2015）第 120473 号

策划编辑：董亚峰
责任编辑：夏平飞
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编　100036
开　　本：787×1 092　1/16　印张：12.25　字数：304.2 千字
版　　次：2015 年 7 月第 1 版
印　　次：2015 年 7 月第 1 次印刷
定　　价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

从目前来看，将图论技术应用到图像处理之中是一种比较合适的选择。传统的方法过于重视图像的局部特征，图割方法将图论以及特征值的方法应用到图像分割等应用当中，能够结合图像的全局信息得到较好的处理结果。因此将图论引入到图像处理当中，是本书的主要出发点。多重网格方法是基于嵌套迭代、松弛光滑误差，通过适当的松弛过程获得光滑误差近似，光滑误差的校正能在更粗的网格上进行，这个计算过程能递归地应用到越来越粗的网格上，这就使计算量与未知数的个数保持一定程度的正比关系。多重网格方法是一种迭代方法，作为一种加速迭代收敛技术，可以很好地求解偏微分问题，在很多领域都有应用，如波动方程、流体力学和图像处理等领域。本书以代数多重网格方法为基础，利用代数多重网格方法对图像的清晰度、图像融合、视频处理、图像的物体识别和特征检索等进行分析，得到了一些较为有用的结果。

全书共分为五章。第 1 章 代数多重网格方法基础。第 2 章 代数多重网格图像清晰度评价及在运动目标检测中的应用。第 3 章 使用代数多重网格方法进行特征表达。第 4 章 代数多重网格方法在图像融合中的应用。第 5 章 代数多重网格方法在物体识别中的应用。

本书第 1 章、第 4 章、第 5 章由黄颖编写，第 2 章由黄颖和尚凤军共同编写，第 3 章由李伟生和周丽芳共同编写，高艺铭编写了附录，全书由黄颖统一定稿。王文斌、王矿生、顾西存、任雪梅和杨明等研究生为本书进行了程序编写及图表制作。本书参考了国内外许多同行的论文、著作，引用了其中的观点、数据与结论，在此向相关作者表示感谢。

本书的编写得到了重庆市科委项目（cstc2014jcyjA40043）、重庆市教委项目（kJ1400408）、重庆市研究生教育教学改革研究项目（yjg143056）、重庆邮电大学文峰骨干教师培养项目的支持。

书中如有不妥之处，欢迎读者批评指正。

编　者

目　　录

第 1 章　代数多重网格方法基础

1.1　代数多重网格方法的基本思想

多重网格方法[1]是求解大型线性代数方程组的一种迭代算法，是对传统迭代解法，如 Gauss-Seidel 方法、Jacobi 方法等的一种卓有成效的加速解法。它以快捷高效的特点而受到人们的广泛关注，目前已经广泛应用到各种实际计算，如流体力学计算、石油储藏模拟、电磁模拟、图像处理、[2-4]经济问题仿真等问题中。它被认为是求解微分方程最快的数值方法。

根据误差理论，数值解与差分方程精确解之间的误差与傅里叶分析中不同频率分量有关。数值求解的误差可以展开成级数的形式，从级数的形式我们可以看出误差的振动分量有很多频段，只要消除了这些频段的误差，整个解也就收敛了。而网格则可以看成一种滤波器，不同尺度的网格可以过滤掉不同频段上的误差。如果网格一定，对于某些误差来说，可以很快过滤掉，对于与其频段不匹配的误差来说，这种网格可能完全失败而根本无法过滤掉。倘若采用多重网格方法，先在低分辨率（网格间隔大）上进行求解，因为此时网格间隔大、数据量小、进行松弛的时空耗费小、收敛快，而且一个很重要的优点是在低分辨率上对初值的敏感度显然要低于在高分辨率上的要求。这样，可以在不同网格层内消去不同波长的误差分量，保证所有的误差分量的收敛速度都不会降低，更快地得到精确的解。它巧妙地结合了细网格光滑和粗网格校正的思想，使得求解方程组的效率空前提高。

多重网格方法[5]最根本的哲学指导思想是一种多分辨率的方法。当我们用显微镜看物体的时候，首先用较小的放大倍数看，寻找微小物体所在的大致场所；然后增大放大倍数，寻找物体所在的精确场所；最后再放大，对物体进行仔细地观察。如果一开始，就用最大的放大倍数看，寻找物体的位置都会花费大量时间。这与代数多重网格方法的思路其实是一致的。在问题求解的过程中，使用方法跟求解域之间应该有一个比较切合的频段。如果频段一致，就能取得事半功倍的效果。

多重网格方法分为几何多重网格方法（GMG）和代数多重网格方法（AMG）。[6,7]几何多重网格方法所针对的问题是有几何背景的问题，即椭圆型偏微分方程在几何区域上的边值问题离散后所得到的大型病态线性代数方程组。几何多重网格方法要用到所求解问题的几何特性，这给实际应用带来一定的不便。代数多重网格方法从某种意义上讲是几何多重网格方法

的一种数学抽象，是多重网格法思想的进一步发展。它仍采用多重网格方法的基本循环方法来求解方程组，但是它的组元的构造仅利用方程组本身的信息，即系数矩阵的元素的信息，它仍具有与几何多重网格方法差不多的效率。代数多重网格方法是仅利用方程组本身的信息来求解方程的多水平方法，这就克服了几何多重网格方法的致命缺点。几何多重网格方法的粗化光滑步骤影响比较小的误差分量，即光滑误差，可以通过求解较粗网格上的残量方程，用较小的计算量而得到比较好的近似，然后插值到细网格，以校正光滑步骤所得到的近似解，从而得到更好的近似解。一般取 $h \to 2h$ 的一致粗化，如图 1.1 所示。插值一般用线性插值，这对光滑算子提出了较高的要求。为了设计有效的多重网格方法，必须选择复杂的光滑算子，使得它能很快地减少不在插值算子值域内的误差分量。而代数多重网格方法是出于另外的一种考虑。它是固定光滑算子，一般是 Gauss 松弛或带参数的 Jacobi 松弛。粗网格和插值算子的建立采取一种自动的方式，以使插值算子的值域能很好地近似不能被光滑步骤所消除掉的误差分量，所以它的插值算子更具有针对性。

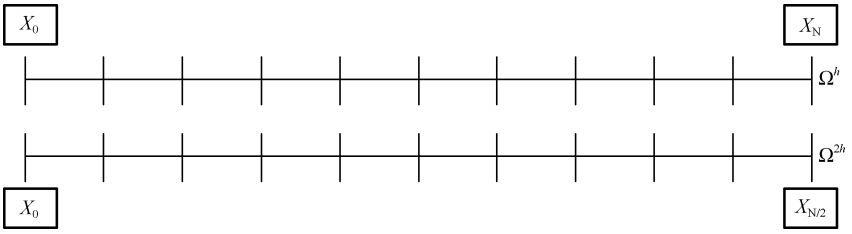


图 1.1　$h \to 2h$ 的一致粗化图示

目前代数多重网格在图像中的应用主要是图像重构、图像二值化、图像复原、图像去噪[8]、图像融合和轮廓分析[9]等。图像领域里很多处理方法如图像分割、图像去噪、图像融合的问题都可以转化成椭圆差分方程（PDE），如各向异性扩散方程可以用来进行图像去噪，Mumford 和 Shah 等提出的用偏微分方程处理图像分割，level set[10]算法也是其中的典型例子。

代数多重网格方法直接针对方程组，它的组元构造仅利用方程组本身的信息，如系数矩阵元素之间的相对大小关系、强连接点的性质等，因此更多地应用于图像处理等各种领域。首先，定义最细的网格为 Ω_0，构造一个网格序列使得式（1.1）成立。相互关系表示如图 1.2 所示。

$$\Omega_n \subset \Omega_{n-1} \subset \cdots \subset \Omega_0 \tag{1.1}$$

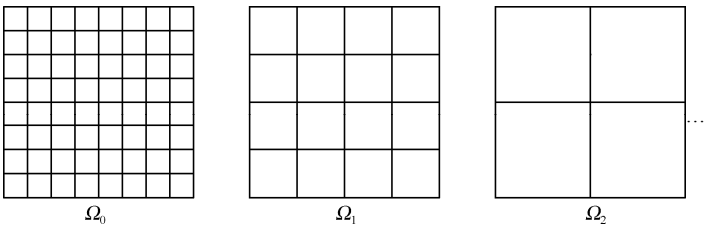


图 1.2　多重网格方法求解空间示意图

在代数多重网格方法求解过程中，细网格光滑步骤、粗网格校正步骤贯穿于网格迭代技术中，三者被巧妙地糅合在一起，求解效率得到提高，而且算法稳健性较高[11]。代数多重网

格方法最根本的指导思想是采用迭代的步骤，在细网格中把影响比较小的误差分量投影到粗网格中，而且粗网格相对于下一级别粗网格亦可认为是细网格，依次迭代到能够精确求解的粗网格级别，再按照原来的序列，由粗网格到细网格，将误差依次插值到最精细的网格中，那么原问题的精确解就可以得到。

代数多重网格方法将由具体方程离散出来的系数矩阵，重投到一系列的由细到粗的网格上，在每一层网格上只做若干次 Gauss-Seidel 迭代，网格信息只来源于系数矩阵。在粗网格中可以求解上一层网格近似解和精确解误差的投影，套迭代技术是整体求解的技术，通过限制算子和延拓算子，将相邻的相对较细网格和相对较粗网格联系起来，将误差和粗网格的解也联系起来，为同一问题进行迭代。首先这种迭代技术必须有两个基础：①方程可以转化到不同网格上求解；②细网格负责消除高频误差，而粗网格负责消除低频误差[12]。

1.2　代数多重网格方法的理论研究

代数多重网格方法分为两步：平滑过程和粗网格修正过程。平滑过程可以迅速将那些高频分量去掉，而粗网格修正过程则可以帮助修正那些光滑了的低频分量，通过不断地迭代，从而快速精确地处理问题。代数多重网格过程的具体思路是：先在细网格上做松弛迭代，然后将误差投影到粗一层网格上去，在粗网格上又做松弛迭代，继续平滑相应的高频部分。依此类推，直到最粗的一层。在最粗一层用直接法求解，然后用插值算子将所求得的误差返回到细网格上去，用以修正原有结果，直到最细的一层。

1.2.1　代数多重网格方法的理论基础

代数多重网格方法（AMG）的目标是求解离散域上的问题

$$AU = f \tag{1.2}$$

其中，A 是一个 $n\times n$ 的矩阵，U 是未知量，f 是给定的 n 维的已知向量。A 矩阵符合 M 矩阵的定义。

代数多重网格方法的较粗网格 $\Omega_{m+1}=C^m$ 是它的较细网格 Ω_m 的一个真子集，记作 $F^m=\Omega_m-C^m$，其中 F^m 为 C^m 层次对应的细网格。

一般情况下，C^m 和 F^m 的选取规则[13]为：

（1）对任意 $i\in F^m, j\in S_i^m$（S_i^m 表示网格 Ω_m 中点 i 强连接的点)，那么 $j\in C^m$ 或者 j 与 C^m 的一个点强连接。

（2）C^m 是强连接点构成的最大点集合。

实际上，对于许多方程组来说，同时严格满足这两个规则是不可能的，在构造粗网格过程中要求严格遵守规则（1），而规则（2）作为一种构造粗网格的指导，利用这两个规则就可以构造网格序列。

强连接点的定义[14]：

定义 1-1：一定条件下的矩阵，如果一个点 i 强连接到点 j，满足

$$-a_{ij}\geqslant\theta_0\max_{k\neq i}(-a_{ik}),0<\theta_0\leqslant 1 \tag{1.3}$$

这个定义实际上是针对 M 矩阵的，即矩阵对称正定，且非对角元素非正，通常取 $\theta_0 = 0.25$ 。

M 矩阵的定义如下。

定义 1-2：对于一个矩阵 A 被称作 M 矩阵，需要满足下列条件：

（1） $a_{i,i} > 0 \ \forall i$ ；

（2） $a_{i,j} < 0 \ \forall i \neq j$ ；

（3） $A^{-1} \geqslant 0$ 。

如：

$$A=\begin{bmatrix} 5 & -1 & -0.1 \\ -1 & 3 & -0.1 \\ -0.1 & -0.1 & 3 \end{bmatrix}$$ 为一个 M 矩阵。

M 矩阵也可以表示为 $A = SI - B, S > 0, B \geqslant 0$ 。若 $S \geqslant \rho(B)$ ，则称 A 为 M 矩阵，若 $S > \rho(B)$ 则称 A 为非奇异 M 矩阵。其中 $\rho(B)$ 为矩阵 B 的谱半径。

通过理论分析，粗网格的提取可以等效于做一个矩阵变换，从一个高维矩阵变换到一个低维矩阵。令细网格层图为 G_f，粗网格层图为 G_c，则

$$G_c=PG_f,\quad G_f=RG_c$$

将细网格变到粗网格类似于投影变换，而反变换对应插值变换。

1.2.2 代数多重网格方法的主要步骤

一个代数多重网格方法主要包含两个步骤：预备部分和迭代循环部分。预备部分是从描述问题的方程组的系数矩阵构造出代数多重网格方法的五个分量：粗网格序列 Ω_h 、插值算子 I_{2h}^h 、限制算子 I_h^{2h} 、粗网格算子 A^h 、光滑算子 G^h 。

（1）粗网格序列 Ω_h ，它能看作是未知数 $u_j^h (1 \leqslant j \leqslant n_h)$ 的集合，是为已知矩阵有关的有向图节点的虚构网格；定义最细的网格为 Ω_0 ，构造一个网格序列使得 $\Omega_n \subset \Omega_{n-1} \cdots \subset \Omega_k \cdots \subset \Omega_0$ ；

（2）光滑算子 G^h ， $G(\Omega_h)$ 在 Ω_h 上的光滑算子；

（3）限制算子 I_h^{2h} ， $G(\Omega_h) \to G(\Omega_{2h})$ ；

（4）插值算子 I_{2h}^h ， $G(\Omega_{2h}) \to G(\Omega_h)$ ；

（5）粗网格算子 A^h ， Ω_h 层上的系数矩阵序列。

其中插值算子和限制算子的关系为：

$$I_h^{2h} = (I_{2h}^h)^T, A^{2h} = I_h^{2h} A^h I_{2h}^h \tag{1.4}$$

这里，T 表示矩阵或者向量的转置。

上面的算子构成方法可以保证粗网格算子具有许多好的性质，比如对称正定等，而在代数多重网格方法中松弛过程是固定的，通常取 Gauss-Seidel 松弛法或者 Jacobi 松弛法，如果需要形成一个代数多重网格循环过程，就必须给出插值算子和粗网格的递推公式，这在代数多重网格中十分重要。

下面描述代数多重网格方法的主要部分[15]。

首先考虑线性方程组

$$AU = F,\quad \sum_{j=1}^{n} a_{ij}u_j = f_i (i = 1,\cdots,n) \tag{1.5}$$

其中，$A = (a_{ij})_{n*n}$；$U = (u_1,\cdots,u_n)^T$；$F = (f_1,\cdots,f_n)^T$。

根据多重网格方法可以得到越来越小的代数方程组

$$A^h U^h = F^h, \sum_{j=1}^{n_h} a_{ij}^h u_j^h = f_i^h (i = 1,\cdots,n_h) \tag{1.6}$$

其中，$A^h = (a_{ij}^h)_{n_h * n_h}$；$U^h = (u_1^h,\cdots,u_n^h)^T$；$F^h = (f_1^h,\cdots,f_n^h)^T$；$h = 1,\cdots,H$；$n = n_1 > \cdots > n_H$；$A^1 = A$；$U^1 = U$；$F^1 = F$。

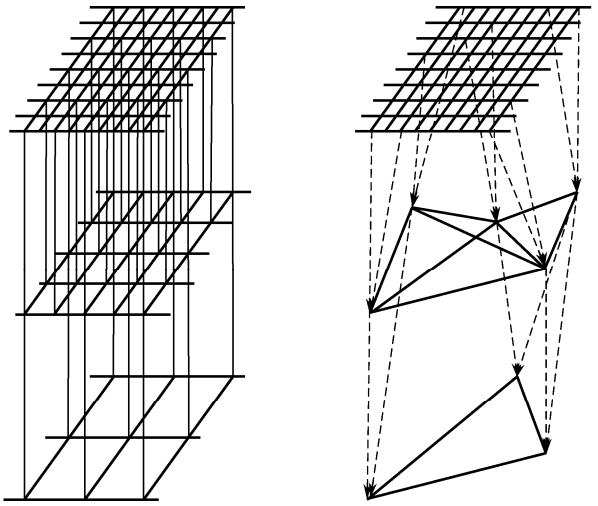
代数多重网格方法的整个流程离不开五个分量，这五个算子是构成代数多重网格方法的主体，而且在代数多重网格方法中组元分量的构造也有很多值得借鉴的地方，如粗网格选取的方法。本节中将对代数多重网格方法五个分量做一下简单介绍。

1．粗网格选取

代数多重网格方法的求解过程能够在一系列的粗网格中做若干次迭代，并将其残差投影到下一级粗网格中，然后依次迭代到在最粗的网格中，再按照原来的序列将残差插值到上一层细网格中，直至到达最细的网格，这样就可以获得问题的精确解，所以代数多重网格方法需要定义一系列的网格序列[16]。

传统多重网格方法的粗网格序列是很规则的，如图 1.3（a）所示，而代数多重网格方法却是根据所求解问题的系数矩阵构造出粗网格序列，网格大小是不规则的，如图 1.3（b）。粗网格构造仅利用方程组系数矩阵的信息，如元素之间的大小以及强连接联系等。

首先需要定义最细的网格为 Ω_0，通常由原始系数矩阵得到。



（a）多重网格方法网格序列示意　　（b）代数多重网格法网格序列示意

图 1.3　网格序列示意

2．插值算子的构造

当粗网格给定后就可以定义插值算子，设误差 $e^h = u_*^h - u^h$，其中 u_*^h 为离散方程的精确解，每个 C^h 中的变量也同时在粗网格中，这样这些点的值直接由 Ω^{2h} 中相应的量以权重 1 给出，所以插值公式的关键实际上是如何用 $\Omega^{2h} = C^h$ 中的变量来计算 F^h 中的变量，即每个点 $i \in F^h$ 的值是由一个较小的插值变量之集 $P_i^h \subseteq C^h$ 以 $e^h = I_{2h}^h e^{2h}$ 的形式给出的。可以用式（1.7）表示。

$$e_i^h = (I_{2h}^h e^{2h})_i = \begin{cases} \sum\limits_{k \in p_i^h} w_{ik}^h e_k^{2h} & i \in F^h \\ e_i^{2h} & i \in C^h \end{cases} \tag{1.7}$$

一维空间的插值算子如式（1.8）所示。

$$\frac{1}{2}\begin{bmatrix} 1 & & \\ 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \\ & & 2 \\ & & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_1/2 \\ v_1 \\ v_1/2 + v_2/2 \\ v_2 \\ v_2/2 + v_3/2 \\ v_3 \\ v_3/2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} \tag{1.8}$$

插值的基本原则：

（1）插值时，在粗网格中的点在细网格中不发生变化。

（2）如果点不在粗网格中，那么细网格中该点的值是其周围点的平均值。插值过程如图 1.4 所示。

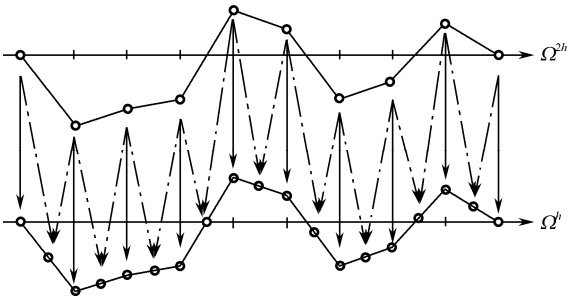


图 1.4　插值计算示意图

一维插值计算代码如下：

```
function [vcoarse] = Restrict1D(vfine,nx)
vcoarse = zeros((length(vfine)+1)/2-1,1);%vfine=7,vcoarse=3
% Do the restriction:
for i = 1:(nx+1)/2-1
        vcoarse(i) =    1/4*(vfine(2*i-1) + 2*vfine(2*i)+ vfine(2*i+1));
end
```

二维空间的插值矩阵如式（1.9）所示。

$$\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \tag{1.9}$$

根据式（1.9）可得

$$v_{2i,2j}^{h} = v_{ij}^{2h}$$

$$v_{2i+1,2j}^{h} = \frac{1}{2}\left(v_{ij}^{2h} + v_{i+1,j}^{h}\right)$$

$$v_{2i,2j+1}^{h} = \frac{1}{2}\left(v_{ij}^{2h} + v_{i,j+1}^{h}\right)$$

$$v_{2i+1,2j+1}^{h} = \frac{1}{4}\left(v_{ij}^{2h} + v_{i+1,j}^{h} + v_{i,j+1}^{h} + v_{i+1,j+1}^{h}\right)$$

3．限制算子的构造

限制算子 I_h^{2h} 与插值算子 I_{2h}^{h} 联系紧密，限制算子和插值算子一般互为转置关系。但是有时候会根据粗网格算子的不同，算子会选取其他的类型。一般插值算子产生后，限制算子可以根据插值算子获得。

一维空间的限制算子如式（1.10）所示。

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & \\ & & 1 & 2 & 1 & & \\ & & & & 1 & 2 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{1.10}$$

一维限制算子的构造过程如图 1.5 所示。

一维限制计算代码如下：

```
function [vfine] = Prolongate1D(vcoarse,nx)
%% Takes a vector vcoarse and interpolates it to a fine grid vector vfine
% the n in the multigrid tutorial book is n = nx+1;
% the nx you send in here is the x dimension of vcoarse, do not confuse it
% with the larger x dimension of vfine.
% I use ghost zero points, so grow the vector
vCoarse = zeros(1,nx+2);
% 1st unpack the vector:
vCoarse(2:nx+1) = vcoarse;
% Now allocate the fine grid
vfine = zeros((nx+1)*2-1,1);
% Do the interpolation:
```

```
vfine(2:2:end-1) = vCoarse(2:end-1);
vfine(1:2:end) = 0.5*(vCoarse(1:end-1) + vCoarse(2:end));
```

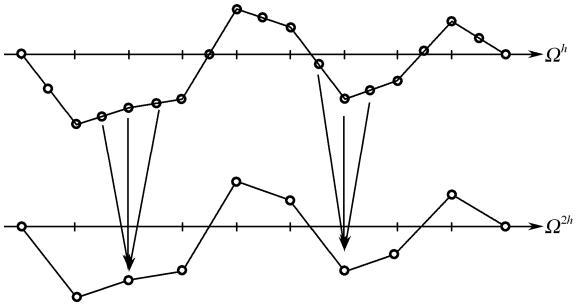


图 1.5　一维限制算子的构造过程

二维空间的限制算子如式（1.11）所示。

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{1.11}$$

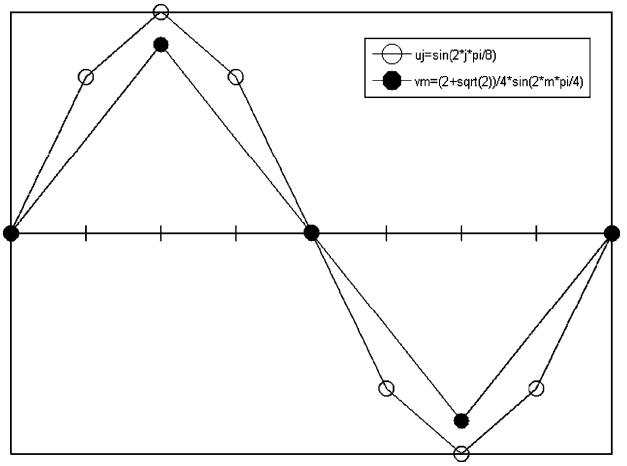


图 1.6　限制矩阵的效果（$7u-3v$）

限制矩阵的效果如图 1.6 所示。细网格上的 $u_j = \sin(2j\pi/8)$ 和粗网格上的 $v_j = \frac{2+\sqrt{2}}{4}\sin(2m\pi/4)$ 都是 sin() 函数，但是频率增加了一倍。细网格上的平滑振荡到了粗网格上振荡只有一半了，这也是限制带来的效果。

怎样使通过限制算子求出的粗网格近似值 V^{2h} 接近精确值 U，我们从图 1.7 和图 1.8 来进行分析。如图 1.7 和 1.8 表示 V^{2h} 和 U 的变化曲线图。

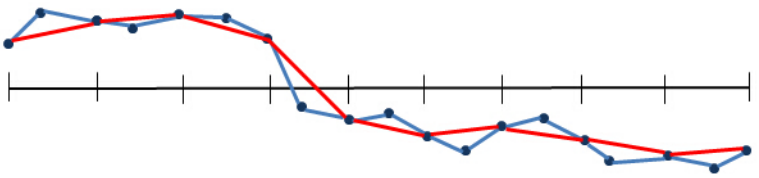


图 1.7　U 平滑变化的情况

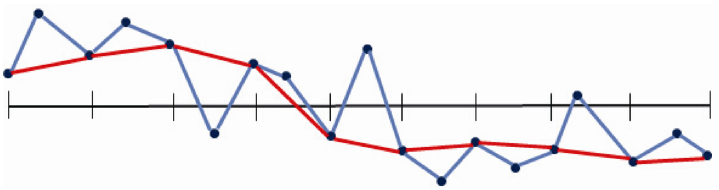


图 1.8　U 摆动变化的情况

从图 1.7 和图 1.8 中可以看出，如果 U 是平滑变化的，那么多重网格的插值 V^{2h} 会很接近精确值 U。如果 U 是摆动的，那么多重网格的插值 V^{2h} 效果不是很好。

4．粗网格算子的构造

在很多文献中，粗网格算子最常见的形成方法是取 Galerkin 算子 $A^{2h}=I_h^{2h}A^hI_{2h}^h$，这样能够保持细网格算子的许多优良特性，如非奇异性、对称正定性等。

5．光滑算子的构造

光滑算子在代数多重网格方法中是固定不变的，一般取 Gauss-Seidel 松弛或带参数的 Jacobi 松弛。

代数多重网格方法具有其他算法不具有的优良特性，如收敛速度快、稳健等性能。代数多重网格方法组元的构造仅仅是根据方程组本身的信息——所求问题的系数矩阵[17]。代数多重网格方法是固定光滑算子，粗网格和插值算子的构造根据系数矩阵自动构造。

图 1.9 为代数多重网格的大致流程，初始为 Ω^0，$AU=F$，在此网格上做若干次迭代，将误差投影到 Ω^1，得到 $A^1U^1=F^1$，再做若干次迭代，将误差投影到下一级网格中，继续迭代求解，最后在网格 Ω^m，得到 $A^mU^m=F^m$，由于阶数较低，所以就可以直接求出精确解。那么下一步迭代回去，误差一步一步地返回到原来的网格中，就可以得到问题的精确解，这是算法的大致思路。

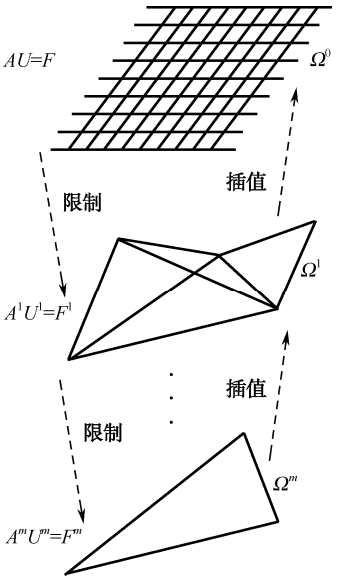
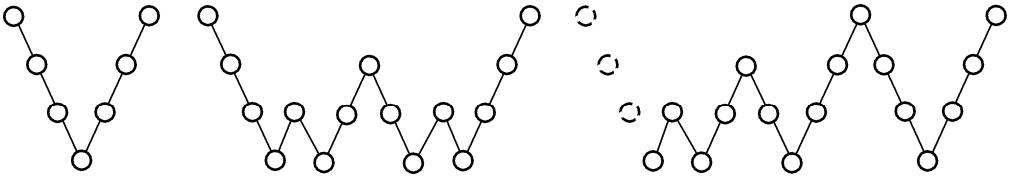


图 1.9　代数多重网格的大致流程

代数多重网格方法另一部分就是标准的多重网格循环过程。图 1.10 中的代数多重网格方法采用的是 V 循环，此外还有 W 循环和 FMG 循环，这些循环的主要思想是一致的。这些循环方法收敛性很好，而且效率也会提高。



（a）V循环　　（b）W循环　　（c）FMG循环

图 1.10　代数多重网格的循环方式

通用的二重网格方法步骤如算法 1.1 所示。

算法 1.1　通用的二重网格方法

在 f 层（精细层，粗糙层定义为 c 层）执行如下。

1．前光滑：在 f 层进行 n_1 次松弛 $A_f u_f=f_f$；

2．粗网格校正：

（1）在细网格层计算残差，得到残差为 $r_f= f_f-A_f u_f$;

（2）将残差通过限制矩阵转移到粗网格上，即 $r_c=Rr_f$;

在第 m+1 层递归执行，直到在最粗的网格层直接求解 $f_c=A_c u_c$;

（3）将错误插值到细网格层 $e_f=Pu_c$;

（4）在细网格层校正错误 $u_f= u_f+e_f$;

3．后光滑：在 f 层进行 n_2 次松弛 $A_f u_f=f_f$。

1.2.3　性能分析

代数多重网格方法能够从存储空间、迭代次数等方面优化计算速度，下面简单分析一下存储空间和计算方面的优化效果。

f^h 必须在每一阶段保存的内容有：

- 在第一次粗化过程中，粗网格的点数是细网格点数的 1/2；
- 在第二次粗化过程中，粗网格的点数是细网格点数的 1/4；
- 在第 d 次粗化过程中，粗网格的点数是细网格点数的 2^{-d}。总共的存储代价为

$$(1+2^{-d}+2^{-2d}+2^{-3d}+\cdots+2^{-Md})<\frac{2N^d}{1-2^{-d}}$$

在第一、第二和第三次粗化过程中，粗网格的存储代价分别小于细网格的 2 倍、3/4 倍和 8/7 倍。

代数多重网格方法每次迭代的代价：

- 在细网格上的一次松弛迭代为一个工作单元；
- 忽略限制与插值的花费（一般占 20%的花费）；
- 考虑一次向前松弛迭代和一次向后松弛迭代的 V 循环，那么 V 循环的计算代价为

$$2(1+2^{-d}+2^{-2d}+2^{-3d}+\cdots+2^{-Md})<\frac{2}{1-2^{-d}}$$

第一、第二、第三次向前松弛迭代计算代价分别为 4 倍、8/3 倍和 16/7 倍。

1.3　代数多重网格方法的实现

1.3.1　应用代数多重网格方法的预备过程

代数多重网格方法在计算粗网格时，会根据矩阵的实际情况构造一系列的粗网格分量，网格密度会根据矩阵中参数的各个元素实际情况变化。所以要将代数多重网格方法的粗网格提取过程应用到图像处理中，须建立多重网格所需要的系数矩阵。

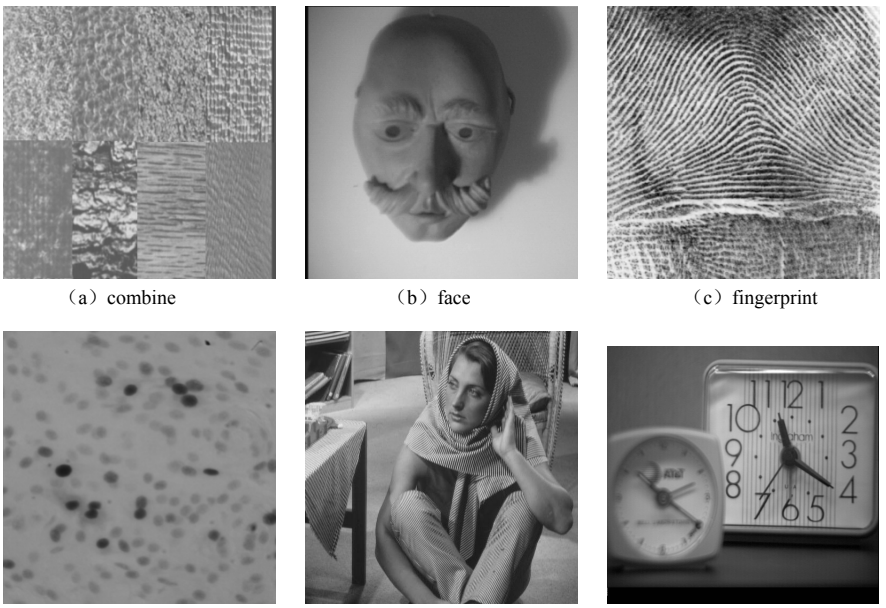
借鉴 Normalized-Cut 方法，将图像转化为图，根据元素的像素灰度值关系建立图 $G(V,E)$，V 为图像的像素点，E 为图像的像素点的相似度。那么就可以构建图像的亲和力（affinity）矩阵。若一幅图像的尺寸为 $M\times N$，那么将图像中的像素按照从左到右单行排列并作为相似度矩阵的行序列和列向量，相似度可以由相似度函数计算得来，那么亲和力矩阵行列均为 $M\times N$，元素个数为 $(M\times N)^2$。

像素间的相似度可利用权函数来计算，常见的权函数有

$$Wij=\exp\left(-\frac{\|F_i-F_j\|_2^2}{\sigma_l^2}\right)*\begin{cases}\exp\left(-\dfrac{\|x_i-x_j\|_2^2}{\sigma_x^2}\right) & \text{当}\|x_i-x_j\|_2<r\\ 0 & \text{其他}\end{cases}\tag{1.12}$$

其中，F_i，F_j 为灰度图像中像素点 i，j 的灰度值；x_i 为像素点的空间坐标；σ_l 为图像灰度值高斯函数的标准差；σ_x 为图像空间位置函数的标准差；r 是两节点的有效距离。那么由公式（1.12）可以看出，两节点距离越近，像素灰度值越接近，则相似度就越大。所以这个权函数既考虑了图像的像素关系，也兼顾图像的空间位置关系。

实验中使用的源图像如图 1.11 所示。



（a）combine　（b）face　（c）fingerprint

（d）meningg　（e）barb　（f）clockA

图 1.11　实验中使用的源图像

1.3.2 图像粗网格序列的提取

为了将代数多重网格方法运用到图像处理中，且要提取图像粗化序列，可以将图像的灰度矩阵作为基本数据，在这个基础上进行一系列的粗化。考虑到代数多重网格方法提取的粗网格序列的大小不是均匀变化的而且点数也没有一定的规律性，所以需要做以下几项工作：

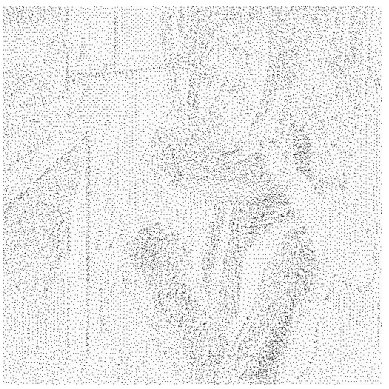
（1）相邻的两级粗化图像像素点的对应关系；

（2）粗化图像序列与原始图像的像素点位置对应关系；

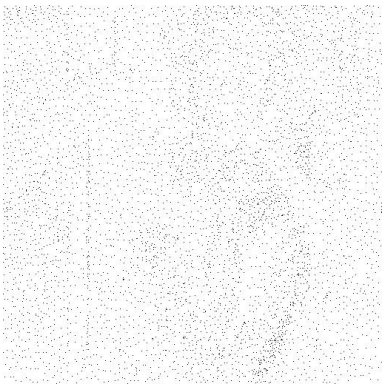
（3）粗化图像序列的显示。

考虑到图像的初始网格是等密度的，但在后续粗化网格中是不规律变化的。一幅图像可以用一个 $m\times n$ 的矩阵表示，可以构建亲和力矩阵 M，因为代数多重网格方法可以直接作用于矩阵，那么运用代数多重网格方法也可以得到图像的粗网格序列。但是图像的显示应该是一个 $m\times n$ 的矩阵，而代数多重网格方法提取的粗网格序列是一系列的不规则的稀疏矩阵，故需要将这些不规则的粗网格序列还原成图像需要的矩阵格式。

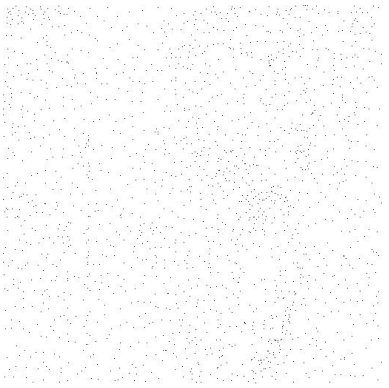
（a）第一层粗网格提取



（b）第二层粗网格提取



（c）第三层粗网格提取



（d）第四层粗网格提取

图 1.12　利用代数多重网格得到的多层网格（源图像：barb）

在 AMG 图像重构中，采用 python 语言的 pyamg 包来实现 AMG 算法，可以得到粗化后的第 1、第 2、第 3 层数据，对应于原始图像中部分像素点，可采用多种插值函数插值，如最

近邻法、线性插值法等，可以得到重构图像，详细步骤如下。

（1）得到网格序列。计算亲和力矩阵 $\boldsymbol{M}$，进行 AMG 粗化得到 N 层粗化网格 Ω_1，Ω_2，…，Ω_N。

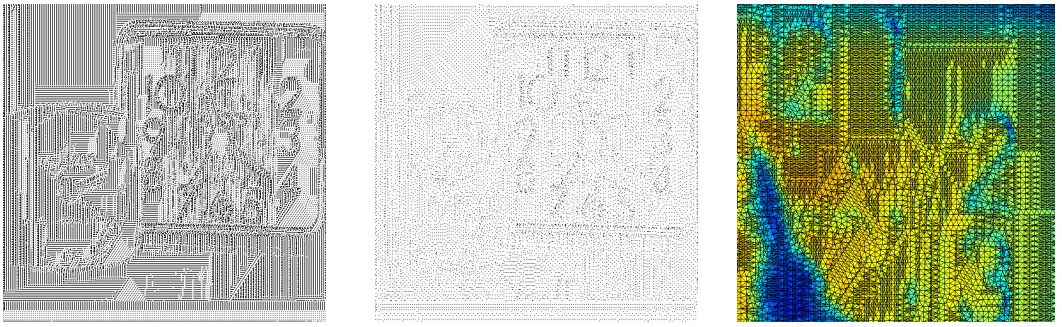
（2）扩充网格。将 N 层网格 Ω_1，Ω_2，…，Ω_N 还原成原始图像大小的 Ω'_1，Ω'_2，…，Ω'_N，网格中的元素被选中为下一层粗网格，对应的元素值为 1，否则为 0。

（3）网格与图像对应。粗网格 Ω'_1，Ω'_2，…，Ω'_N 与原始图像 I 很容易找到对应位置关系，采取如下策略：

① 如果网格 Ω'_i 中的数据是 1，则令对应位置的像素为图像相应位置的灰度值；

② 如果网格 Ω'_i 中的数据是 0，则令对应位置的像素等于 0。

这样就产生了一系列图像 I_1，I_2，…，I_N。本算法中使用的一些原始图片如图 1.11 所示。利用代数多重网格得到的多层网格分别如图 1.12 和图 1.13 所示。图 1.13（c）是对图 1.13（b）中的图进行局部放大，可以看到，局部放大的效果能把钟盘的数字较为清晰地反映出来。



（a）第一层粗化之后的网格　（b）第二层粗化之后的网格　（c）第一层图像的局部放大

图 1.13　利用代数多重网格得到的多层网格（源图像:clock A）

1.3.3　粗网格序列的插值

在 1.3.2 节中，可以得到图像的一系列粗网格序列，这些图像系列随着像素点数的减少，对源图像的信息量保留也是逐渐降低的，但保留的是源图像的信息量较大的像素点。本节将对这些粗网格序列图像进行插值，即得到各层的重构图像。代数多重网格能够提取较为丰富的轮廓信息，因此利用轮廓信息进行图像的重建能够较好地逼近原始图像，保持原始图像的结构。对代数多重网格方法得到的粗网格数据进行分析，粗网格中会保留强连接部分而去掉弱连接部分，将这种机制应用到图像处理，可以发现代数多重粗化的网格可以提供较为丰富的轮廓信息。在代数多重网格方法粗化的各层数据中，粗网格数据具有图像的轮廓，变化剧烈的细节部分点分布聚集，平滑模糊部分点分布较稀疏均匀。在计算粗网格时，网格密度会根据图像中的变化情况而改变。初始化时原始图层网格是等密度的，但在粗网格中，图像灰度变化平滑区区域，网格密度较为稀疏且较规则，而当灰度变化急剧时，网格密度则很稠密且不规则，这种现象在一定程度上反映了代数多重网格方法中自适应网格的作用。根据这种特性，在粗网格上插值进行图像的重构，因粗网格较好地保持了图像的结构，理论上可以得

到较好的重构结果。但是考虑到代数多重网格的网格序列的点集合是按照网格的代数信息（这里可以认为是图像的灰度信息）选取粗网格点的，粗网格的点是不规则的，所以应该将每层网格中的点对应到原始图像中的相应位置，这样才能得到较好的插值效果。插值效果的衡量采用的是均方误差（Mean Square Error），定义为

$$\mathrm{MSE}=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[G(i,j)-G'(i,j)]^2}{mn} \tag{1.13}$$

图 1.14 是利用多层网格进行插值得到的重建效果。它是对 barb 图利用第一到第四层粗网格数据进行插值的结果，说明代数多重网格方法对于纹理图片也能较为准确地反映出来。图 1.15 是使用不同插值方法得到的结果，第一行是使用线性插值方法得到的结果，第二行是使用 cubic 方法得到的插值结果。

（a）第一层图像重建之后的结果

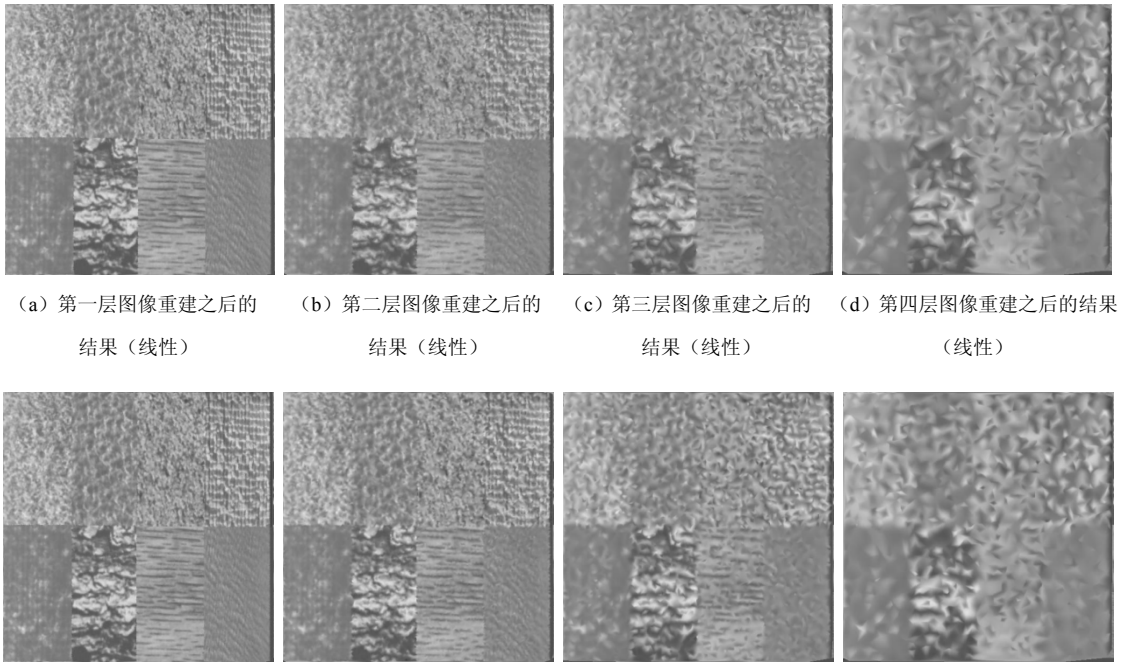
（b）第二层图像重建后的结果

（c）第三层图像重建之后的结果

（d）第四层图像重建之后的结果

图 1.14　利用第一到第四层粗网格数据进行插值的结果(源图像:barb)

比较粗化序列图像可以发现：粗化序列图像点密度分布均匀规则的地方，对应位置的源图像的灰度变化平缓；而粗化序列图像点分布不均匀的地方，源图像对应位置的灰度变化较剧烈。这个现象说明粗化序列一定程度上保持了图像轮廓特征，这种现象在一定程度上反映了代数多重网格方法中自适应网格的特性。通过对图像粗化序列信息量的比较，发现与网格提取的点数所占比例都小于图像的信息量，说明图像粗化序列能够较好地选择信息较丰富的像素点。



（a）第一层图像重建之后的结果（线性）　（b）第二层图像重建之后的结果（线性）　（c）第三层图像重建之后的结果（线性）　（d）第四层图像重建之后的结果（线性）

（e）第一层图像重建之后的结果（cubic）　（f）第二层图像重建之后的结果（cubic）　（g）第三层图像重建之后的结果（cubic）　（h）第四层图像重建之后的结果（cubic）

图 1.15　利用第一到第四层粗网格数据进行插值的结果(源图像:combine)

进一步考虑代数多重网格重建方法的效果，这里将其与小波重建算法做比较。对比图 1.11 中的 face、fingerprint、meningg 图片，分别对三幅图片做小波图像重建和代数多重网格方法重建，结果如表 1.1 所示，比较结果用 MSE 表示。

表 1.1　图像的重建效果的 MSE 比较

	face		fingerprint		meningg	
	AMG 方法	小波方法	AMG 方法	小波方法	AMG 方法	小波方法
第一层	10.21	15.50	112.89	184.99	2.08	2.67
第二层	29.71	52.88	473.71	651.80	5.92	9.49
第三层	70.01	72.69	1143.6	1213.71	17.35	26.37

从表 1.1 中可以看出，针对一些简单的结构，如 meningg，通过第三层粗网格都能得到较为精确的重建。针对比较复杂的结构，如图 1.15 所示，在第一层中能够得到较好的重建结果，说明代数多重网格方法对于纹理结构的提取也是比较有效的，由重构结果可以看出代数多重网格的重构结果比小波重构效果好。

1.4 代数多重网格方法的理论分析

1.4.1 从一个差分方程到一个线性代数问题

一个差分方程，如果需要利用计算机进行求解，首先需要将其离散化。例如，求解一个简单的差分方程

$$L\tilde{u} = f, \tilde{u}, f \in \Omega \tag{1.14}$$

满足边界条件为

$$\tilde{u}|_{\partial\Omega} = 0 \tag{1.15}$$

其中，L 是一个差分算子；$\tilde{u}$ 是一个单一变量或者多个变量的函数（$\tilde{u}(x, y, \cdots, t)$）。使用一个简单的一维空间的例子，$L\tilde{u} = \dfrac{d^2\tilde{u}}{dx^2}$。差分后可得

$$\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ & & 1 & 2 \end{pmatrix}\begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{pmatrix} \tag{1.16}$$

等效于求解

$$Au = f \tag{1.17}$$

1.4.2 单一网格上求解线性方程的问题

定义线性系统 $Ax = b$，残差 $r_k = b - Ax_k$，预调整表达式 $P \approx A$。

预调整表达式 P 要尽量接近 A，同时允许快速递归。Jacobi 选择 A 的三角矩阵作为 P 是一个极端，速度快但是与 A 之间相差较大。另一个极端是 $P = A$。将矩阵 A 分开得到 $Ax = b$ 的新的表达式

$$Px = (P - A)x + b \tag{1.18}$$

相应地得到递归的另一种表达式

$$Px_{k+1} = (P - A)x_k + b \tag{1.19}$$

从任意给定的 x_0 开始，然后从 $Px_1 = (P - A)x_0 + b$ 得到 x_1。继续使用同样的矩阵得到 x_2。所以通常通过 P=LU 分解来得到 P 的三角因子。有时 P 本身就是三角的，或者 L、U 有时近似于 A 的三角因子。P 如果满足以下两个条件，则可以得到有效解。

（1）新的 x_{k+1} 必须能迅速计算出来；

（2）$e_k = x - x_k$ 必须尽快地收敛到 0。

用式（1.19）中减去式（1.18），可得

$$Pe_{k+1} = (P - A)e_k \tag{1.20}$$

等效于

$$e_{k+1} = (I - P^{-1}A)e_k = Me_k \tag{1.21}$$

可以发现 e_k 的计算与 b 无关。每一步只需要将 e_k 乘以一个矩阵 M 即可。x_k 收敛到 x 的速

度完全依赖于矩阵 M。矩阵 M 收敛必须满足

$$M \text{ 的每个特征值必须满足} |\lambda(M)| < 1 \tag{1.22}$$

矩阵中绝对值最大的特征值也被称为谱半径 $\rho(M) = \max|\lambda(M)|$。矩阵 M 收敛要求 $\rho(M) < 1$。收敛率跟最大的特征值有关。当一个矩阵比较大时，$\rho(M)$=0.9 甚至于 $\rho(M)$=0.99 时是最好的。

当初始错误 e_0 刚好等于矩阵的特征向量时，下一个错误值 $e_1 = Me_0 = \lambda e_0$。随后的每一步的错误都要乘以一个 λ。所以我们需要 $|\lambda| < 1$。正常情况 e_0 是所有特征向量的线性组合结果。当乘以 M 时，每个特征向量都乘以自己的特征值。k 步之后，每个向量都乘以 λ^k，其中最大的是 $(\rho(M))^k$。

如果不使用预调整表达式，则 $M = I - A$。为了收敛，A 的所有的特征值都必须在以 1 为中心的单位圆的内部。Jacobi 矩阵 $M_J = I - D^{-1}A = D^{-1}(L+U)$ 的特征值为

$$\lambda_j(M) = 1 - \frac{2 - 2\cos j\theta}{2} = \cos j\theta, \theta = \frac{\pi}{N+1}, 0 \leqslant j \leqslant N \tag{1.23}$$

$|\cos j\theta| < 1$，所以肯定收敛，但是收敛速度比较慢。

j=1 时特征值最大，其频谱半径为

$$\lambda_{\max}(M) = \cos\theta \approx 1 - \frac{1}{2}\left(\frac{\pi}{N+1}\right)^2 \tag{1.24}$$

$$w_{k,j} = \sin\left(\frac{jk\pi}{N}\right), 0 \leqslant j \leqslant N$$

从式（1.24）中可以发现，低频时收敛较慢。而加权 Jacobi 方法相对较好，$M_w = I - wD^{-1}A = [(1-w)I + wM_J]$，比较结果如图 1.16 所示。

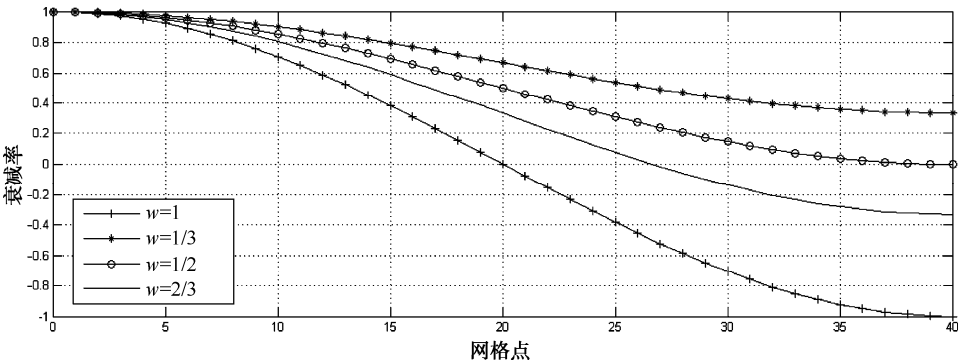


图 1.16　Jacobi 方法和加权 Jacobi 方法的特征值分析比较

从图 1.16 中的曲线可以看出，对于低频分量 $v^k\left(1 \leqslant k < \frac{n}{2}\right)$ 衰减都不会很小，特别当 k=1 时，误差分量相对最大；而对于高频分量 $v^k\left(\frac{n}{2} \leqslant k \leqslant n\right)$，则可以通过 w 进行选择而控制衰减因子。当 w=2/3 时，一次迭代就可以使得高频误差分量衰减到初始误差分量的 1/3。这些结果表明，低频误差分量是迭代收敛慢的原因，而高频误差分量则在迭代中衰减得很快。换句

话说，随着迭代次数的增加，误差的光滑性也进一步增强。对于这种现象，其他的迭代方法也是存在的。

1.4.3 多网格上求解线性方程的解决方案

多网格方法是在离散过程中利用网格属性影响收敛速度的特性的一种迭代求解方法。从傅里叶分析中可以看出，网格宽度（与空间频率成反比）是主导收敛的一个重要因素。

重新在粗网格上就选择初始的猜测值，然后在粗网格上选取其他任意一个点。粗网格上只有细网格上一半的网格点数。细网格上的傅里叶模式可以重新写成

$$w_{k,2j}^{h}=\sin\left(\frac{2jk\pi}{N}\right)=\sin\left(\frac{jk\pi}{N/2}\right)=w_{k,j}^{2h} \tag{1.25}$$

这意味着在 Ω^h 上的误差分量变成了 Ω^{2h} 上的误差分量，如图 1.16 所示。从图 1.16 上可以看出，细网格上的低频分量在粗网格上来看比在细网格上要振荡。从中可以看出，将细网格上的光滑误差分量限制到粗网格上，然后在粗网格上进行迭代处理就可以消除部分误差分量，基于这种思想我们可以一层一层地做下去直到消除所有的误差分量为止，最后还需要将求解问题一层一层地返回，直到最初的细网格上。

以一个二层网络为例来说明。对误差 e 使用式（1.22）进行平滑，如图 1.17 所示。根据 $A^{2h}=I_h^{2h}A^hI_{2h}^h$，可以得到如图 1.18 所示的结果。

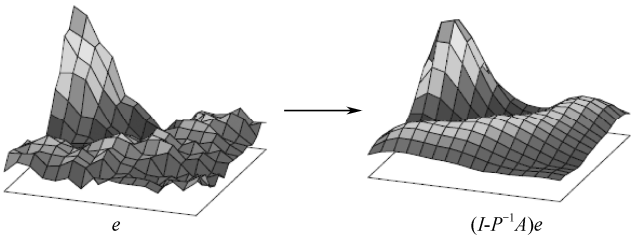


图 1.17　对误差 e 进行平滑

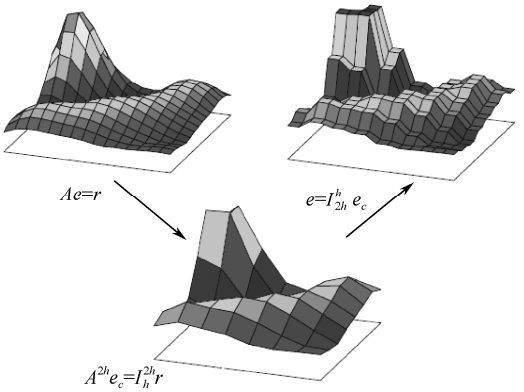
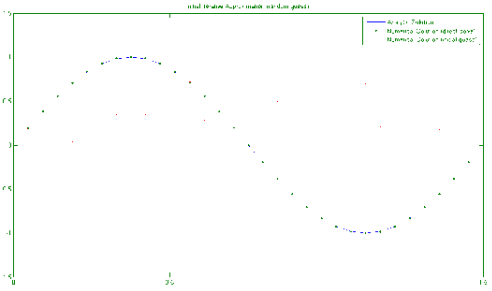


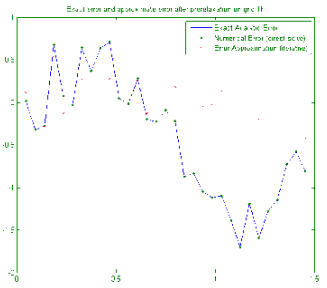
图 1.18　对误差 e 在下一个粗网格求解

从图 1.19 中也可以发现当层次越高时，求解越迅速越精确。因为此时数据量小，进行松弛时的时空耗费小，而且收敛快。而在最细层中初始值和精确值之间存在较大的误差，而且

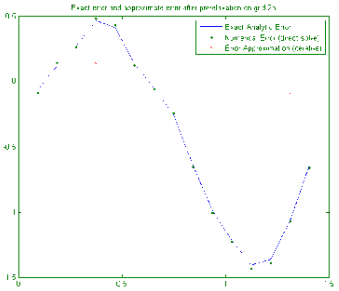
毫无规律性。但是提取的粗网格层也是有限的，当提取的粗网格层过粗，或者原始数据波动特别迅速时，此时使用多重网格方法就存在较大的问题。当层次越高，其大致形状跟原始数据之间存在较大差异，插值回去时会出现较大误差，这时就需要进行重新考虑了。



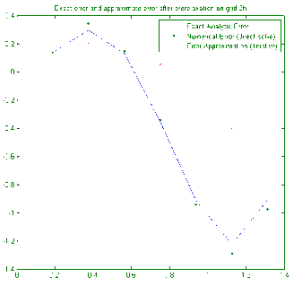
（a）随机初值与真实解



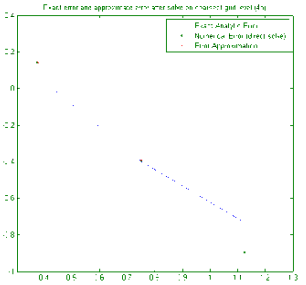
（b）第一层限制



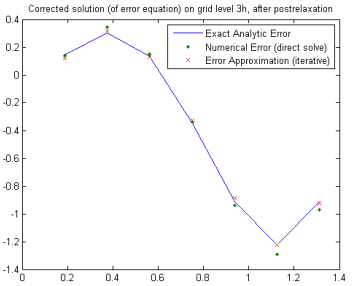
（c）第二层限制



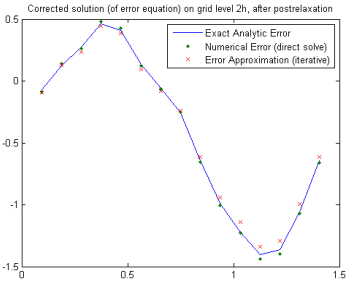
（d）第三层限制



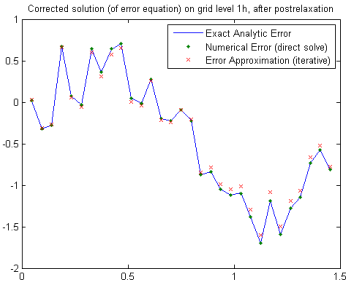
（e）第四层精确求解



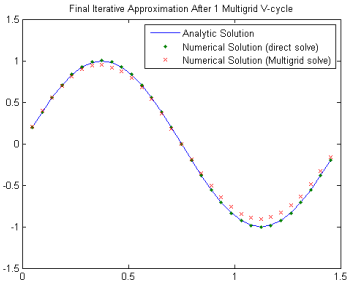
（f）第三层插值



（g）第二层插值



（h）第一层插值



（i）最终解

图 1.19　代数多重网格迭代的整个过程

1.4.4 粗网格在图分类方法中的应用

基于特征值方法的图分类方法的本质是在求解一个最优化问题。将图分成两类，表达形式如式（1.26）。

$$\begin{aligned} &\min E(x)=\sum_{i,j} w_{i,j}(x_i-x_j)^2 i\in A, j\in B \\ &\text{s.t}\ \ \min E(x)=\sum_{i,j} w_{i,j}(x_i-x_j)^2 i\in A, j\in B \end{aligned} \tag{1.26}$$

式（1.26）等效于

$$\begin{aligned} &\min E(x)=x^T Ax \\ &\text{s.t}\ \ x^T Bx=1 \end{aligned} \tag{1.27}$$

其中，$a_{ij}=-w_{ij}, a_{ij}=\sum_j w_{ij}, b_{ij}=\delta_{ij}$。得到的 A 就是图的拉普拉斯矩阵。

根据拉格朗日定理可知，x 的求解等效于求解式（1.28）的特征值。

$$Ax=\lambda BX \tag{1.28}$$

当矩阵比较大的时候，这时需要考虑对该图在粗网格层上来进行处理。通过粗网格划分，组成一个新的由粗网格上的节点构成的矩阵 A_c。

$$A_c=p^T AP \tag{1.29}$$

现在，优化式（1.29）等效于求解

$$\min E(x_c)\approx x_c{}^T A_c x_c \tag{1.30}$$

1.4.5 利用拉普拉斯矩阵提取粗网格

拉普拉斯矩阵在目前的流形学习中也有很重要的地位，使用拉普拉斯作为流形学习的基本思想是在高维空间中距离很近的点在低维空间中投影的距离也应该离得很近，通过对图像进行拉普拉斯投影到高维空间，然后通过求解特征向量的方式将其投影回低维空间，可以得到图像中像素聚类的一些关系。因此，这里使用拉普拉斯矩阵来进行相应的粗网格的提取。

拉普拉斯矩阵的优点主要有：（1）拉普拉斯矩阵是个“0 和”矩阵，最小的特征值为 0，其他的特征值都大于 0；（2）拉普拉斯矩阵的谱比邻接矩阵的谱更能反映图的图论性质，因此拉普拉斯矩阵的谱分析理论目前研究较多，较为成熟。

一个实对称矩阵可以表示成

$$W=\sum_{i+1}^{n}\lambda_i v_i v_i^T \tag{1.31}$$

其中 $|v_1|=\cdots=|v_n|=1$。

从式（1.31）中可以看出，一个矩阵可由它的特征向量完全表示，而每个向量所对应的特征值，就代表了矩阵在这个向量上的贡献率。

矩阵的能量可以表示为

$$E(W)=\sum_i \lambda_i^2 \tag{1.32}$$

根据式（1.32）的关系，对特征值进行排序，保留其中较大的特征值，也就可以保留对应的一些较大非对角线元素。

针对原始图像构建邻接矩阵，对邻接矩阵求解拉普拉斯矩阵。求解拉普拉斯矩阵的特征

值和特征向量，根据一定的准则选择几个特征值，将这几个特征值对应的节点选取出来作为粗网格的节点。这种方法与主成分分析（PCA）方法的过程和结果是比较类似的。在 PCA 中，通过构建几个分量的协方差矩阵，求解特征方程得到特征值，然后按照寻找几个较大的特征值就可以得到分量在一个低维空间的近似的分布，可以达到近似和降维的目的。

在邻接矩阵中需要选择从大到小的特征值进行排序，提取粗网格节点。而在拉普拉斯矩阵中需要提取从小到大的特征值进行排序，这里给出证明。需要证明的问题表述为：度相等的拉普拉斯矩阵和邻接矩阵的特征值之间存在 $\lambda' = d - \lambda$，其中 d 为度矩阵的特征值，λ 为拉普拉斯矩阵的特征值，λ' 为邻接矩阵的特征值。

证明

$$Lx = \lambda x \tag{1.33}$$

L 矩阵为拉普拉斯矩阵，可以分离成度矩阵 D 和邻接矩阵 W 的差，其中 $w_{ii} = 0, w_{ij}$ 为邻接矩阵。

$$(D - W)x = \lambda x \tag{1.34}$$

可以得到

$$Dx - \lambda x = Wx \tag{1.35}$$

进一步可得

$$D'x = Wx \tag{1.36}$$

其中 D' 中 $D'_{nn} = D_{nn} - \lambda$。$D_{nn} = \cdots = D_{00} = d$ 时，可以将其转换成

$$(d - \lambda)x = Wx \tag{1.37}$$

这表示度相等的矩阵的拉普拉斯矩阵和邻接矩阵的特征值之间存在

$$\lambda' = d - \lambda \tag{1.38}$$

对于其他度不相等的情况，通过理论很难去证实。因为目前大部分谱分析的研究都集中在研究每个特征值的上界和下界，而没有对两个具体的特征值之间的关系进行系统的研究。因为考虑到上面的研究结果，在此对两个特征值之间的关系进行理论和实践的研究。通过选择随机矩阵的方式来进行描述。通过随机产生邻接矩阵，构建度矩阵和拉普拉斯矩阵，分别计算特征值，根据大量随机矩阵的方式去进行分析。图 1.20 和图 1.21 分别为度矩阵相同（矩阵规模为 2500）和度矩阵不同（矩阵规模为 50）时各个特征值之间的关系图。图中，$\lambda(L)$ 为从大到小，而 $\lambda(W)$ 从小到大以及 $\lambda(D)$ 从小到大。图 1.20 中存在 $\lambda(L) + \lambda_{n-k}(W) = \lambda_k(D)$ 的关系，只有几个节点除外，主要是因为边界效应（$d_1 = d_n = d_i - 1, i = 2 \sim n$）的原因，这也验证了式（1.38）的推断。

通过图 1.20 与图 1.21 的对比分析，图 1.20 描述的是当度相同的时候，存在 $\lambda_n - k(L) + \lambda_k(W) = \lambda_k(D)$ 的关系。也就是 $\lambda_{n-k}(L) + \lambda_k(W)$ 和 $\lambda_k(D)$ 的差值能够从一定程度上反映图中不同度的分布情况，两者差值为 0 的临界值对度的分布进行进一步分析具有一定的指导意义。我们定义这个临界值为 k_1，在 k_1 之下 $\lambda_{n-k}(L) + \lambda_k(W) \leqslant \lambda_k(D)$（其中 $\lambda(L)$ 为从小到大排列，而 $\lambda(W)$ 从大到小排列）。从大量的实验分析可看出，k_1 之上就是 $\lambda_k(L) + \lambda_{n-k}(W) > \lambda_k(D)$，但也有一些例外，也就是中间存在一些较小的波动，因此定义另一个临界值 $k_2 \geqslant k_1 + 1$，在 k_2 之上存在 $\lambda_k(L) + \lambda_{n-k}(W) > \lambda_k(D)$。实际上这个波动的产生跟矩阵的内在特征也有很大的关系。

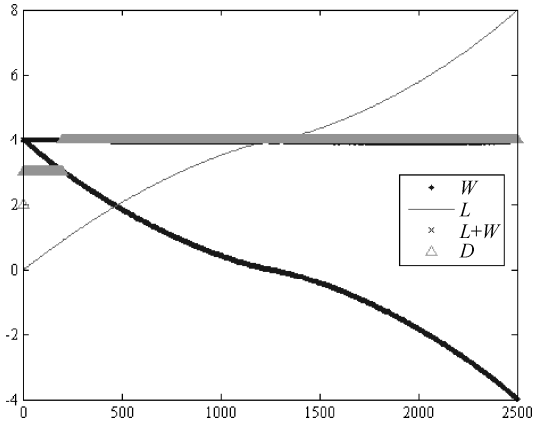


图 1.20　矩阵规模为 2500 时各个特征值的变化情况（度相同的情况）

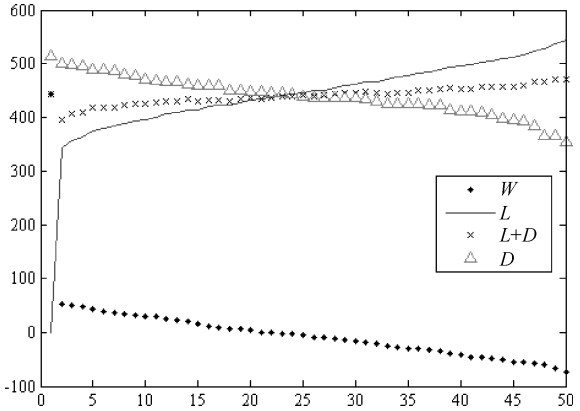


图 1.21　矩阵规模为 50 时各个特征值的变化情况（度随机产生）

定理 1-3：如果 A 和 $A+E$ 是 $n\times n$ 的实对称矩阵，其中矩阵的特征值的排序如 $\lambda_n \leqslant \lambda_{n-1} \cdots \leqslant \lambda_1$，有

$$\lambda_k(A)+\lambda_n(E) \leqslant \lambda_{k(A+K)} \leqslant \lambda_k(A)+\lambda_1(E) k=1 : n \tag{1.39}$$

证明：必定存在一个点 k_1，使得 $\lambda_{n-k+1}(L)+\lambda_k(W) \leqslant \lambda_k(D)$。

因为拉普拉斯矩阵和邻接矩阵的关系为 $L+W=D$，根据定理 1-3，有

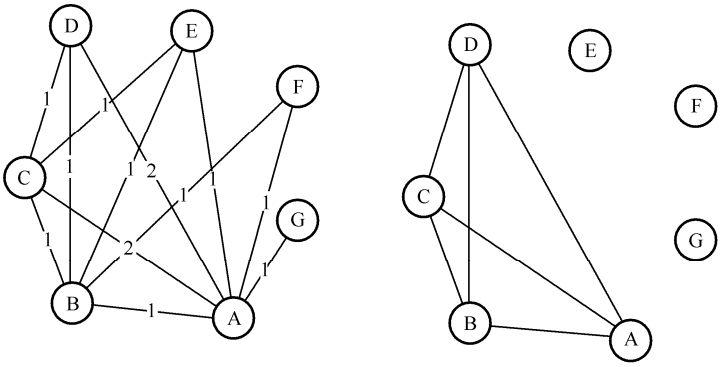
$$\lambda_n(L)+\lambda_1(W) \leqslant \lambda_1(D) \tag{1.40}$$

$$\lambda_1(L)+\lambda_n(W) \geqslant \lambda_n(D) \tag{1.41}$$

因为 $\lambda_n \leqslant \lambda_{n-1} \cdots \leqslant \lambda_1$，因此必定存在一个点 k_1，使得

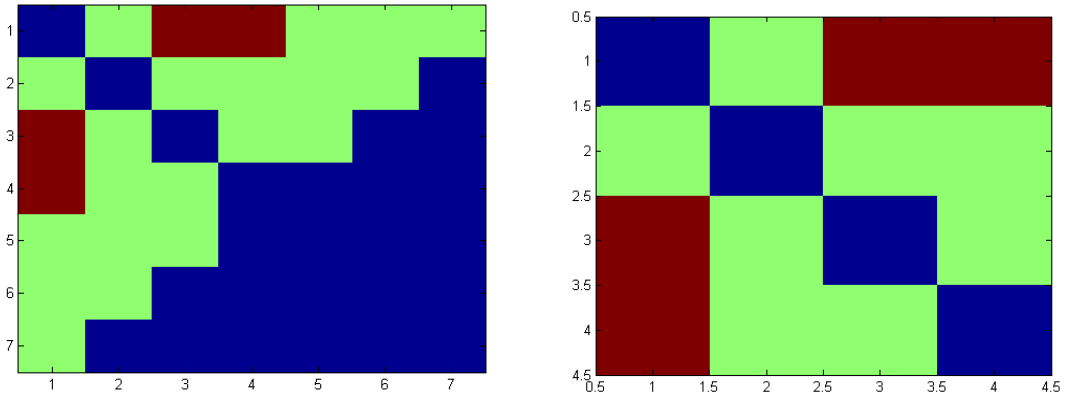
$$\lambda_{n-k+1}(L)+\lambda_k(W) \leqslant \lambda_k(D) \tag{1.42}$$

根据这个问题定义一个新的临界值 k_1 和 k_2。使用 k_1 作为选择拉普拉斯特征值个数的依据。使用下面的例子来验证选择的正确性。如图 1.22 所示，计算得到 $k_1=4$，因此根据特征值排序，得到特征值较小的四个节点，如图 1.22（b）所示。使用粗网格提取的邻接图的结果如图 1.23 所示。可见其中粗网格点都被提取出来了。尤其是图 1.23（b）中 B 节点，从图 1.22（b）中发现其度为 3，跟 E 节点的度一样，但是因为它在原始图中的度比 E 节点的高，因此还是保留在粗网格之中了。



（a）G图　　（b）由特征值排序提取的强连接子图

图 1.22　根据矩阵特征值来进行粗网格的提取



（a）原始图的图像表示　　（b）提取粗网格的图像表示

图 1.23　使用 k_1 个特征值得到的粗网格节点

1.5　代数多重网格方法与小波的对比研究

从目前查到的文献中，大部分都是关于利用代数多重网格加速有限元求解、加速椭圆差分方程求解方面的文章。关于代数多重网格中强连接分析、代数多重网格重建图像方面、将代数多重网格与小波等多尺度分析的对比分析的文献较少。从代数多重网格和小波的对比分析中可以看出，代数多重网格相当于利用一个全局的矩阵对精细层的图进行运算，作用是提取图中强连接部分的子图结构。因此从这一角度上可以看出，代数多重网格方法在提取原始图的具有强连接的子图方面与小波方法提取边缘特性有一定的联系。

下面对代数多重网格和小波方法做一些对比分析，如图 1.24 所示。

（1）代数多重网格是一种处理图的方法，小波方法是一种处理图像的方法。

（2）在代数多重网格中也用到了类似于小波算子的一个算子，小波算子可以提取图像的突变信息，而代数多重网格可以提取图中的强连接子图。

（3）代数多重网格根据强连接性构建一个新的子图，在新的子图中原始图像中的某些平滑的部分在子图中不会出现，也就是子图中每个点对应原始图像的位置信息必须能够保存下

来，否则无法重建原始图像。而小波算子是在固定位置中做的一些操作，提取的特征点的位置信息是固定的，因而在重建的时候不需要另外的位置信息。

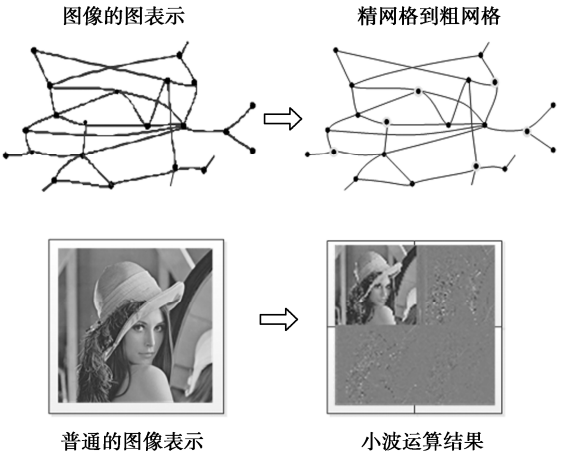


图 1.24　代数多重网格与小波变换的对比关系示意图

（4）小波算子是在像素的基础上采用的局部算子，使用的是卷积运算，而代数多重网格是在图的邻接矩阵的基础上使用的是全局算子。

代数多重网格对于多尺度分析也有一些联系，下面对代数多重网格方法与多尺度分析方面做一些对比分析。

一方面，从多尺度分析来看，它的每级分解等效于从精网格到粗网格的转换过程。每级重建等效于粗网格到精网格的转换过程。在这个处理过程中，类似于前面提及的 V 循环。另一方面，利用代数多重网格进行分析，它是对一个精网格递归粗化，然后对最粗的网格递归细化，重新得到精细网格的过程。处理中每级网格可以划分为两个部分，并且互补。每级精网格点由在下一级粗网格上的点和不在下一级粗网格上的点构成，然后粗网格作为下一级精网格点继续往下分。

$$F_m = \Omega_m = \Omega_{m+1} + \Omega_{m+1} = C_{m+1} + C_{m+1} \tag{1.43}$$

这样可以逐层构成一个类似的多尺度分析空间。与多尺度空间不同的是，粗网格方法中空间的几何位置不是有规则的，而是无规则的。

1.6　本章小结

本章针对代数多重网格方法的理论基础和算法实现进行了阐述，并与小波方法进行了对比分析。利用代数多重网格方法能够较好地提取图像的特征点集合，利用粗网格进行图像重建能较好地逼近原始图像。本书的后续章节将利用代数多重网格方法提取得到的粗网格进行图像清晰度检测、图像融合以及物体识别，从应用的角度进一步验证该方法的效果。

第 2 章　代数多重网格图像清晰度评价及在运动目标检测中的应用

2.1　图像清晰度评价

图像的清晰度直接关系图像质量的好坏，影响图像处理时的准确率[18]。清晰的图像含有更丰富的细节信息，便于图像的处理。图像的清晰度评价函数是检测一幅图像清晰程度的重要手段，也是提取图像中清晰区域的重要环节。

本章提出基于代数多重网格的无参考图像清晰度评价算法，算法改进了均方误差在评价图像清晰度时对参考图像的依赖性。算法先对待评价的图像做代数多重网格的图像重构，然后将重构后的图像和待评价的图像的均方误差值作为图像的清晰度度量值。

2.1.1　主观评价

主观图像质量评价是以终端用户对图像的感知来评价图像的质量，观测者对给定的不同清晰度的参考图像和失真图像（不同模糊算法形成的不同模糊程度的图像）打分，相对评价范围分为很好、较好、一般、较差、很差，对应的绝对分值分别是 5、4、3、2、1[19]。

主观图像质量评价以观测者的主观感知为依据评价图像的质量，在很多实际应用场景中，就是以人的感知作为评价清晰度的标尺，所以主观质量评价被认为是最可靠的评价方法。但是主观评价非常耗时，代价大，同时在实际的系统集成中是很难实现的。

2.1.2　客观评价

客观图像质量评价[20]的目的是建立一种自动评价图像质量的方法，该方法用数学公式描述的方法提取图像的信息并计算得到图像质量的数值表示。客观图像质量评价计算简单，和主观图像质量评价相比更具实时性。目前存在的图像质量评价方法有三类，分别是全参考、弱参考和无参考图像质量评价。其中，无参考图像质量评价因其不需要提供任何参考图像的

信息、计算方便而成为应用最广泛的图像质量评价方法。

图像清晰度作为衡量图像质量的重要指标，其评价方法也被广泛地研究。图像的不清晰由多种情况造成，主要有运动模糊、失焦、图像压缩、去噪和重采样[21]等。传统的图像清晰度评价方法主要分为三类，分别是基于边缘梯度的方法、基于频域的频谱函数方法和基于熵函数的方法[22]。

1．基于边缘梯度的方法

在图像清晰度评价中，梯度函数一般用于对图像进行边缘提取。通常聚焦清晰图像的边缘信息表现为更加丰富和锐利，梯度值也越大。所以，对边缘的梯度计算可以用于图像的清晰度评价。图像清晰度评价中常用的梯度函数[23]如下。

（1）Benner 梯度函数

Benner 梯度函数用于计算图像中相邻像素灰度差，为了更突出边缘的重要性，对差值结果取平方。用 $f(I)$ 定量表示图像的清晰度，则 $f(I)$ 表达式为

$$f(I)=\sum_x\sum_y\{[I(x+1,y)-I(x,y)]^2+[I(x,y+1)-I(x,y)]^2\} \tag{2.1}$$

其中，I 表示待检测图像；$I(x,y)$ 是 I 在点 (x,y) 处的像素值；$f(I)$ 是图像 I 的清晰度值。

（2）Tenengrad 梯度函数

与 Benner 梯度函数的不同在于 Tenengrad 梯度函数用于同时计算水平方向和垂直方向的梯度值，计算过程中使用 Sobel 算子作为滤波器的掩模。Tenengrad 梯度函数定量表示图像清晰度的表达式为

$$f(I)=\sum_x\sum_y s(x,y)^2, s(x,y)>T \tag{2.2}$$

其中，T 表示阈值；$s(x,y)$ 是水平和垂直方向的梯度值。$s(x,y)$ 定义式为

$$s(x,y)=\sqrt{I_x(x,y)^2+I_y(x,y)^2} \tag{2.3}$$

Tenengrad 梯度函数中边缘检测的掩模 Sobel 算子定义为

$$I_x=\frac{1}{4}\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}, I_y=\frac{1}{4}\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \tag{2.4}$$

（3）Laplacian 梯度函数

Laplacian 梯度函数和 Tenengrad 梯度函数大体上是一致的,但是在边缘检测算子上不同,Laplacian 梯度函数使用 Laplacian 算子进行边缘检测,该算子的定义如式（2.5）。

$$f(I)=\sum_x\sum_y(I(x,y+1)+I(x,y-1)+I(x+1,y)+I(x-1,y)-4I(x,y)) \tag{2.5}$$

Laplacian 梯度函数中边缘检测的掩模 Laplacian 算子的定义如式（2.6）。

$$L=\frac{1}{6}\begin{bmatrix}1 & 4 & 1\\ 4 & -20 & 4\\ 1 & 4 & 1\end{bmatrix} \tag{2.6}$$

（4）能量梯度函数

能量梯度函数又称为相邻像素灰度方差法（SMD），其清晰度计算的公式为

$$f(I)=\frac{1}{n}\left[\sum\left|I(x,y)-I(x,y-1)\right|+\sum\left|I(x,y)-I(x+1,y)\right|\right] \tag{2.7}$$

其中 n 表示图像中像素的总和。

2．基于频域的频谱函数方法

频谱函数用于图像清晰度评价的理论基础是图像细节信息集中在高频部分，具体做法是通过统计待检测图像的高频分量来达到评价图像清晰度的目的。我们可以采用傅里叶变换、拉普拉斯变换或者小波变换来获取图像中的高频分量。假如用傅里叶变换提取图像高频分量就是对图像进行 FFT 变换，基于傅里叶变换的图像清晰度评价函数定义为

$$f(I)=\sum_x\sum_y W(x,y)^2 \tag{2.8}$$

其中 $W(x,y)$ 是具体的高通滤波器。$W(x,y)$ 可由下面的公式计算得到

$$W(x,y)=I(x,y)**G_0 \tag{2.9}$$

其中，I 表示待检测图像；$I(x,y)$ 是图像 I 在点 (x,y) 处的像素值；** 表示二维卷积；G_0 是相应的空域高通滤波器。

3．基于熵函数的方法

熵函数用于图像清晰度评价的理论依据是与模糊图像相比较，清晰图像包含更多的信息量。因此，对于给定的图像，通过计算它的信息熵来定量该图像的清晰度评价。假如给定图像 I，图像熵 $H(I)$ 和图像能量 $E(I)$ 分别定义如下

$$H(I)=-\sum_x\sum_y I(x,y)\ln[I(x,y)] \tag{2.10}$$

$$E(I)=\sum_x\sum_y I(x,y) \tag{2.11}$$

根据香农信息理论的定义，我们可以知道当 $E(I)$不变时，$H(I)$越大，那么图像就越清晰。

均方误差是一种常用的全参考图像质量评价方法[24]，该方法需要提供参考图像来计算失真图像的清晰度，本章提出了基于代数多重网格的无参考图像清晰度评价方法，是对均方误差法的改进。算法首先利用代数多重网格法对图像进行粗网格提取、插值重构，然后将重构后的图像与原始图像做均方误差，计算结果用来度量图像的清晰度。改进后的方法不需要提供参考图像，是无参考的。本章算法对图像进行全区域的扫描，面向完整的图像。当图像越清晰时，图像整体像素相似度较小，使用代数多重网格选取网格点并进行插值重构后的图像与原始图像的均方误差会越大，因此可以用重构后的图像和重构前的图像的均方误差值来表征图像的清晰度。

2.1.3　代数多重网格方法和图像清晰度的关系

根据第 1 章的内容，我们对代数多重网格重建后的图像与原始图像做对比，计算 MSE。在分析的过程中发现 MSE 跟图像的清晰度有较大的关系。为此，我们采用代数多重网格方法提取图像的粗层数据，并进行图像重建。通过多组图片的实验，验证了原始图像与重构图像间的均方误差与清晰度的关系。

本节针对 lena 图和 barb 图进行不同程度的模糊，分别采用高斯卷积核来进行模糊，使用的函数为 matlab 中的 PSF=fspecial(‘gaussian’,*,*)，利用该函数与原始图像进行卷积，可以

得到不同模糊程度的图像，模糊程度依次增加；针对模糊之后的图像使用代数多重网格方法进行重建，不同模糊程度的图如图 2.1 和图 2.2 所示，重建结果的对比见图 2.3 和图 2.4，与原图计算的 MSE 见表 2.1。

（a）原始图像

（b）高斯核（5,5）的模果

（c）高斯核（10,10）的结果

（d）高斯核（15,15）的结果

（e）高斯核（20,20）的结果

（f）高斯核（25,25）的结果

图 2.1　对 lena 图进行不同程度的模糊

选用高斯卷积核的原因在于一些光学成像的机理对原始图像做了一个点扩散运算，也就是一个高斯卷积核的卷积运算，得到的清晰图像是因为这个高斯卷积核较小，在视觉可以接受的范围内表现是清晰的，而模糊在于这个高斯卷积核比较大，出现了视觉感知的不同程度的模糊。

（a）原始图像

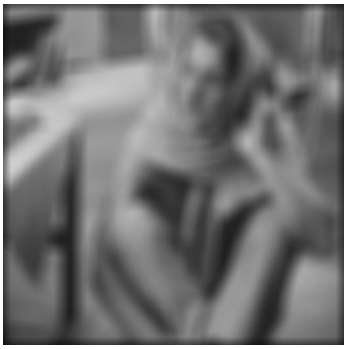
（b）高斯核（5,5）的结果

（c）高斯核（10,10）的结果

图 2.2　对 barb 图进行不同程度的模糊

（d）高斯核（15,15）的结果

（e）高斯核（20,20）的结果



（f）高斯核（25,25）的结果

图 2.2　对 barb 图进行不同程度的模糊（续）

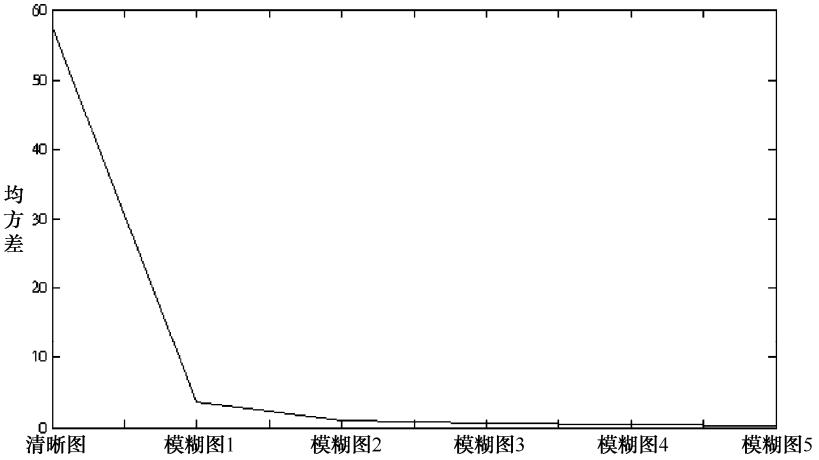


图 2.3　lena 图中不同模糊程度的对比关系图

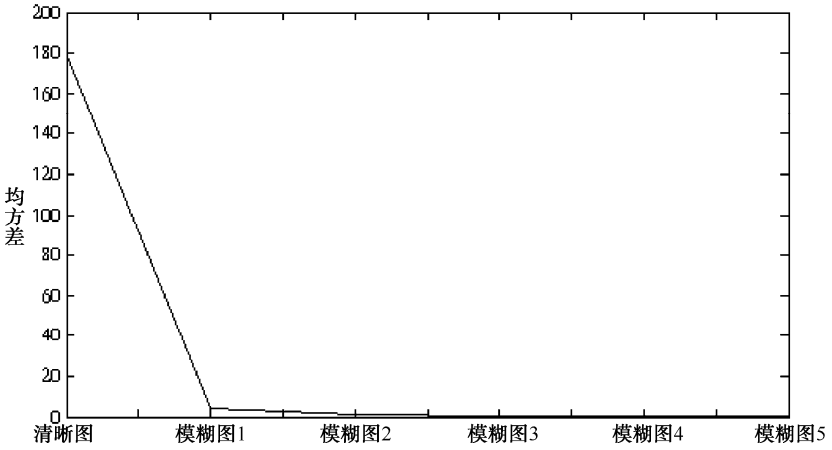


图 2.4　barb 图中不同模糊程度的对比关系图

从表 2.1 中可以看出，模糊图会得到较小的 MSE，清晰图会得到较大的 MSE。因此可以使用 MSE 的相对值来评价图像的清晰程度。

表 2.1　进行不同程度的模糊之后重建得到的 MSE

	原始图	模糊图 1	模糊图 2	模糊图 3	模糊图 4	模糊图 5
lena	57.77	3.72	0.97	0.60	0.45	0.37
barb	179.42	4.35	1.05	0.56	0.44	0.38

2.1.4　使用代数多重网格进行无参考图像的清晰度评价

在统计学中，均方误差是反映估计量与被估计量之间差异程度的一种度量，是对于无法观察的参数θ的一个估计，其在统计学中的其定义为

$$\mathrm{MSE} = E((T-\theta)^2) \tag{2.12}$$

均方误差表示“误差”的平方的期望值。其中，误差表示估计值T与被估计量θ的差。均方误差满足下面的等式

$$\mathrm{MSE}(T) = \mathrm{var}(T) + (\mathrm{bias}(T))^2 \tag{2.13}$$

其中

$$\mathrm{bias}(T) = E(T) - \theta \tag{2.14}$$

由式（2.14）可以得到，偏差bias(T)是估计函数的期望值$E(T)$与待估计参数θ的差。

我们将均方误差的思想引入到图像质量评价中，计算参考图像和目标图像的均方误差值并作为图像清晰度值，其定义的清晰度评价函数为

$$\mathrm{MSE}=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(f_{ij}'-f_{ij})^2 \tag{2.15}$$

其中，i、j是像素点坐标；f_{ij}为原始图像；f_{ij}'为f_{ij}的畸变图像；M、N是图像的尺寸大小。

均方误差法是一种传统的客观全参考图像质量评价方法，算法的优点是计算简单且物理意义明确，但是该算法在评价图像质量时忽略了像素点之间的相关性，只是对图像中像素点之间误差进行纯数学的统计。

基于代数多重网格的无参考图像的清晰度方法是对均方误差法的一种改进，改进后的评价方法是无参考的，不需要提供非失真图像，更具实时性。首先采用 AMG 包对目标图像实现粗网格化，得到粗化序列第 1 层数据，对应到原始图像中的部分像素点后，可采用插值方法得到重构后的图像，然后将重构后的图像与原始图像做均方误差，得到的值用来度量图像的清晰程度。

算法示意图如图 2.5 所示。

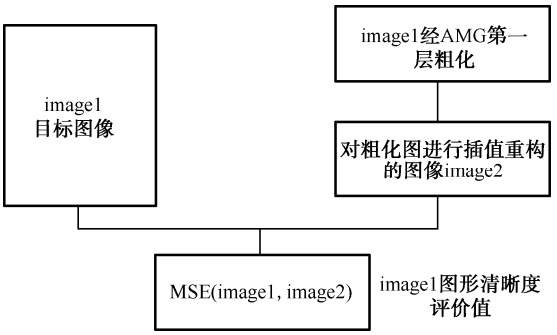


图 2.5　算法示意图

2.2 实验结果与分析

2.2.1 算法性能评价准则

根据文献 [25] 介绍，一个好的图像清晰度评价算法必须满足下列准则：

（1）预测的单调性。当图像渐变模糊或清晰时，图像清晰度评分值也应该呈现相应的单增或单减性。

（2）预测的一致性。一个好的指标应该在不考虑特定图像内容的情况下也能表现良好。一般一个新的评价方法在研究时总是针对一组固定的数据集来验证，在这组数据集中表现良好，但是应用到其他不同的数据集时就无法正确判定图像的清晰度。

（3）预测的精确性。这是指正确评价图像质量的能力，一般可通过与对应该指标主观评价的 MOS 分值进行对比确定。图像的 MOS 值是主观评价中观察者给出的图像清晰度指标。获取的方法是选取一群独立的人类个体作为观测者，他们分别给出 5 个等级的图像清晰度结果，取值范围是 0 到 100，每幅图像的平均值就是该图像的 MOS（Mean Opinion Score）值。

2.2.2 检验算法的单调性

实验数据集：6 幅 512×512 不同模糊程度的 House 图片，其中一幅原始图片，5 幅使用高斯滤波处理的图片，模板窗口大小是 7×7，图（a）～图（f）的标准差分别是 0.4、0.8、1.2、1.6、2.0。实验目的是为了验证算法评价清晰度随图像清晰或模糊而单调递增或递减。实验结果如图 2.7 所示。由图 2.7 可知，当图像渐变模糊时，图像的清晰度度量值随着图像模糊度增加单调递减，即表示基于代数多重网格的图像清晰度评价方法满足单调性的准则。

（a）House 图像

（b） fspecial（‘gaussian’,49,0.4）的模糊结果

图 2.6　House 不同程度模糊图

（c） fspecial（‘gaussian’,49,0.8）的模糊结果

（d） fspecial（‘gaussian’,49,1.2）的模糊结果

（e） fspecial（‘gaussian’,49,1.6）的模糊结果

（f） fspecial（‘gaussian’,49,2.0）的模糊结果

图 2.6　House 不同程度模糊图（续）

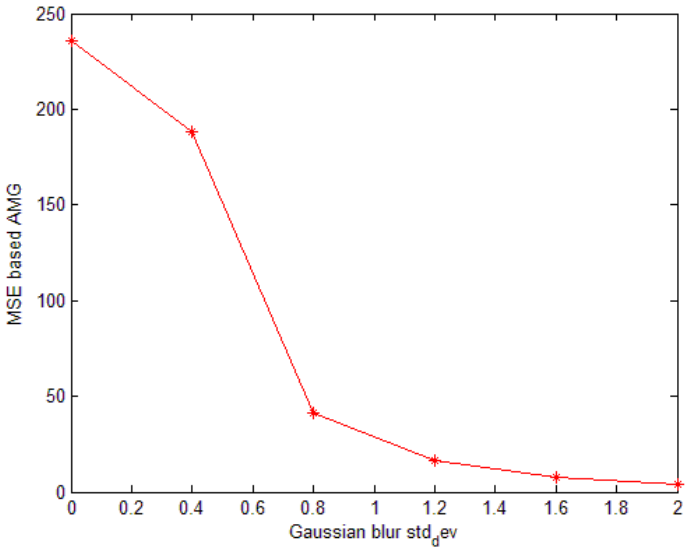


图 2.7　6 幅不同清晰程度的 House 图像对应的清晰度评价值

2.2.3　检验算法的一致性

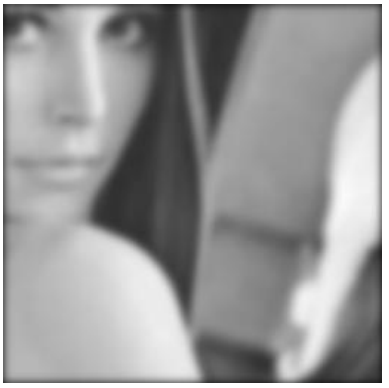
为了说明算法对图像内容的无关性，将尺寸为 512×512 的 lena 图像分成 4 个尺寸相同的区域，4 个图像块使用高斯滤波处理。实验在该数据集上做三组不同的处理。

（1）首先，对切割的 4 个图像块按从上到下、从左到右的顺序，即如图 2.8 中（a）～（d）做高斯滤波处理，模板窗口大小是 7×7，标准差分别是 1、2.5、4、5.5，其结果如表 2.2 所示，可知图像清晰度值与图像内容无关，与图像模糊程度表达一致。

（a）区域 1 fspecial（‘gaussian’,49,1）的模糊结果

（b）区域 2 fspecial（‘gaussian’,49,2.5）的模糊结果

（c）区域 3 fspecial（‘gaussian’,49,4）的模糊结果



（d）区域 4 fspecial（‘gaussian’,49,5.5）的模糊结果

图 2.8　lena 图像高斯模糊图

表 2.2　分割区域高斯模糊后清晰度评价值

image name	std_dev	清晰度评价值
（a）	1	34.99856
（b）	2.5	9.674969
（c）	4	5.505095
（d）	5.5	2.263391

（2）为了进一步表现本节算法与图像内容无关，先用本节算法对 4 个清晰的图像块计算得到表征 4 个图像块的清晰度值，如表 2.3 所示，按照清晰度值从大到小的顺序加一个逆序

的高斯滤波标准差。如最清晰的图 2.8（a）块加最大的模糊（5.5），清晰度评价值最小的图 2.8（d）加最小的模糊（1）。模板窗口大小是 7×7，处理后的图像组如图 2.9 所示，处理后的图像组清晰度值如表 2.4 所示，表 2.4 的结果说明了图像清晰度指标只与高斯模糊标准差有关，而与图像内容无关。

表 2.3　分割区域的图像清晰度值

image name	（a）	（b）	（c）	（d）
本节算法	136.4214	86.17076	208.8404	81.90912

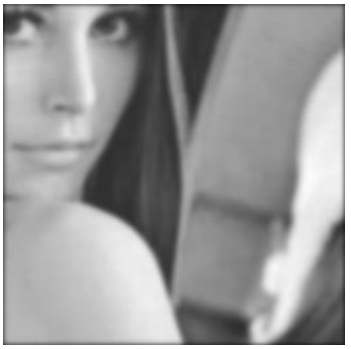
表 2.4　分割区域高斯模糊后清晰度值

image name	std_dev	清晰度评价值
（a）	5.5	1.61791
（b）	2.5	9.67496
（c）	4	5.50509
（d）	1	36.3086

（a）fspecial（‘gaussian’,49,5.5）的模糊结果

（b）fspecial（‘gaussian’,49,2.5）的模糊结果

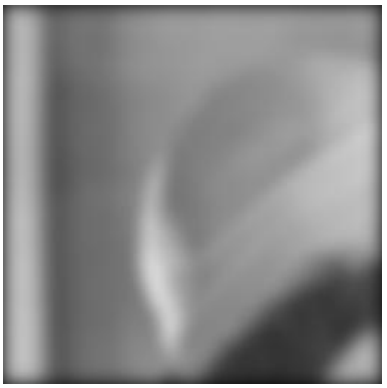
（c）fspecial（‘gaussian’,49,4）的模糊结果



（d）fspecial（‘gaussian’,49,1）的模糊结果

图 2.9　lena 图像高斯模糊图

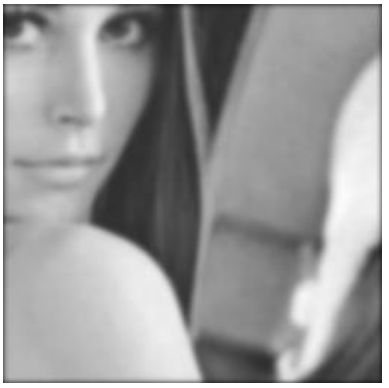
（3）表 2.4 中（a）和（c）计算结果相近，互换两幅图像的高斯模糊标准差，处理后的图像结果如图 2.10 所示，重新计算得到新的结果如表 2.5 所示，表 2.4 中（a）、（c）互换标准差后，清晰度值的大小顺序也有所改变，说明图像清晰度值与图像内容无关。



（a）fspecial（‘gaussian’,49,4）的模糊结果

（b）fspecial（‘gaussian’,49,2.5）的模糊结果

（c）fspecial（‘gaussian’,49,5.5）的模糊结果



（d）fspecial（‘gaussian’,49,1）的模糊结果

图 2.10　lena 图像高斯模糊图

表 2.5　分割区域高斯模糊后清晰度值

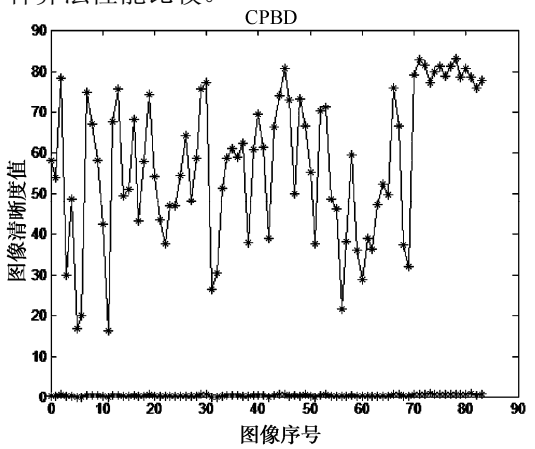
image name	std_dev	清晰度评价值
（a）	4	3.070689
（b）	2.5	9.674969
（c）	5.5	2.833922
（d）	1	36.30863

由以上三组实验可知，本节算法表征图像清晰度与图像内容无关，满足一致性的准则。

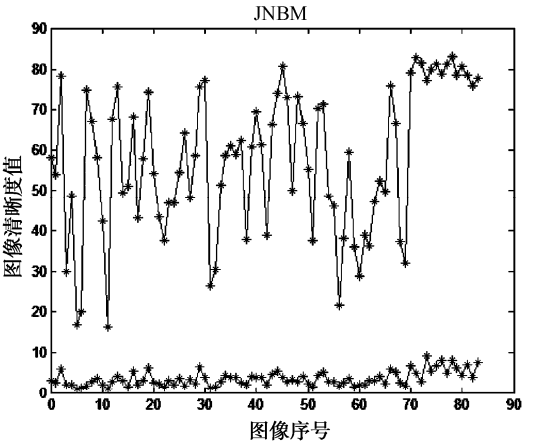
2.2.4　检验算法的精确性

为了测试算法结果是否与主观感知一致，本节采用 LIVE[26]提供的图像质量评价数据库中 14 幅原始图像，每幅图像有 6 个随机高斯模糊图像，数据库给出每幅图像的主观评价分值（MOS）。分别用 6 种算法计算这 84 幅图的清晰度值，这 6 种算法是 CPBD、JNBM、SMD、信息熵、方差法和本节提出的算法，根据 VQEG 提出的客观模型评价准则[27]，选用五个参数来比较算法的性能：Person 相关系数（精确性）、Spearman 秩相关系数（单调性）、均方误差（RMSE）、平均绝对误差（mean absolute prediction error，MAE）和离出率（outlier ratio，OR）。

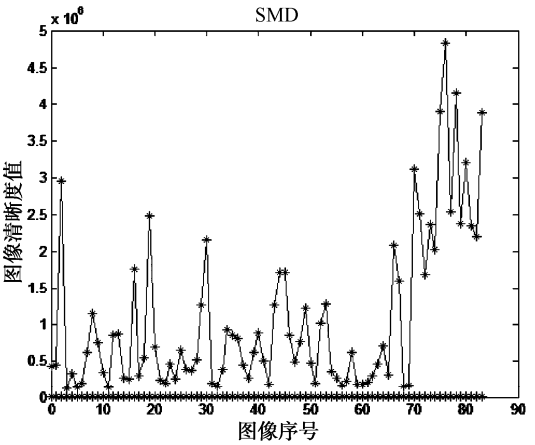
其中，前两个参数值越大越好，后三个参数值越小越好。图 2.11 给出了 6 种算法计算的清晰度值和 MOS 值的比较，由图 2.11 可以看出，本节提出的算法与 MOS 值最为相近。表 2.6 是 6 种算法性能比较。



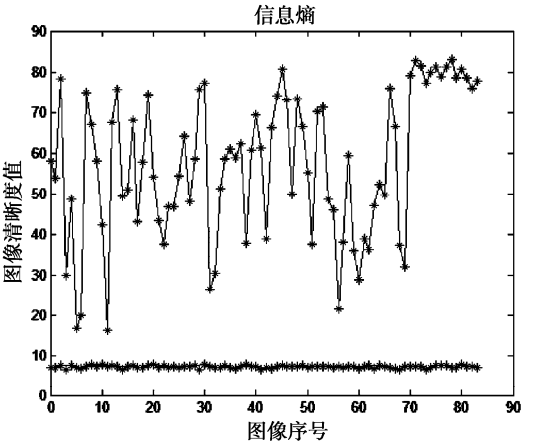
（a）CPBD计算值与MOS值



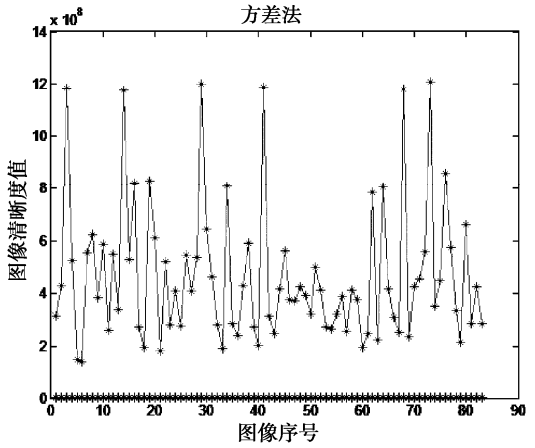
（b）JNBM计算值与MOS值



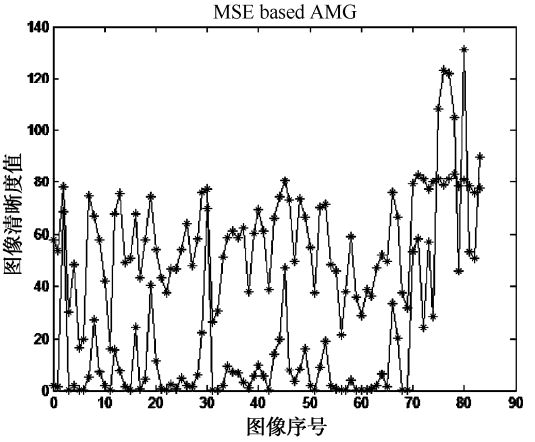
（c）SMD计算值与MOS值



（d）信息熵计算值与MOS值



（e）方差计算值与MOS值



（f）本节算法计算值与MOS值

图 2.11　各算法计算值与 MOS 值的比较

图 2.11 中红色表示 MOS 值，（a）～（f）分别表示 6 种算法的清晰度计算结果和 MOS 值的比较，从图 2.11 中可以看出本节提出的算法的结果与 MOS 值关联最大。

表 2.6　各区域的图像清晰度值

算法	Pearson	Spearman	RMSE	MAE	OR
CPBD	0.908	0.930	0.095	7.518	5.766
JNBM	0.829	0.808	0.191	10.052	7.691
SMD	0.721	0.792	0.298	12.456	9.391
信息熵	0.218	0.214	0.512	17.539	14.867
方差法	0.106	0.239	0.5	17.869	15.204
本节提出的算法	0.918	0.949	0.107	7.129	5.705

从表 2.6 可以看出，在上述 5 个指标中，本节提出的算法优于 JNBM 和 SMD 等算法，但本节算法在 RMSE 指标上不如 CPBD 算法，主要因为本节提出的算法表征清晰度的值变化范围较大。但是本节提出的算法较 CPBD 具有更高的精确性和更好的单调性，总体上优于 CPBD。

2.3　代数多重网格在运动目标检测中的应用

运动目标检测[28]是计算机视觉的一个重要研究分支，是智能视频技术的基本问题，处于视频系统的最底层，是目标识别、目标跟踪和目标行为检测的基础，在整个智能化发展中有着至关重要的作用。运动目标检测是从视频序列中有效地提取运动目标，以便后续对其识别和跟踪等做进一步研究。如何将运动目标检测出来，需要对前景和背景进行有效的分离。

2.3.1　使用代数多重网格进行图像的重建

由于在正常视频序列中帧与帧之间的差别很小，很难看出区别，为了后续的处理，本节选取的是相差几帧的图像进行实验。图 2.12 和图 2.13 是提取的视频帧中相隔几帧的图像，图像大小为 512×512，选取的是两个人正常走动的过程。图 2.12 为选取的是运动对象比较简单的一个场景，场景中有运动的行人。图 2.13 相对来说比较复杂，除了运动的行人外还有汽车。

图 2.12　简单场景图像

图 2.13　复杂场景图像

根据前面代数多重网格方法步骤，得到粗网格，然后根据粗网格对图像进行粗化，得到粗化后的三层图像，由于粗网格能较好地保留原始图像的特征信息，所以图像中的显著区域

被保留下来。将得到的三层粗化图像进行插值运算，可以得到原图像的重构图像如图 2.14～图 2.17 所示。图 2.14 和图 2.15 分别为简单场景前一帧和后一帧的重建结果，而图 2.16 和图 2.17 分别为复杂场景前一帧和后一帧的重建结果。

图 2.14　简单场景前一帧三层重构图像

图 2.15　简单场景后一帧三层重构图像

图 2.16　复杂场景前一帧三层重构图像

图 2.17　复杂场景后一帧三层重构图像

由于从第一层到第三层图像插值的数据量大幅度减少，所以前两层的整体效果比较好，第三层比较模糊。

2.3.2　基于帧间差分法的目标检测

帧间差分法[29,30]是一种通过对视频图像序列中相邻两帧或者多帧做差分运算来获得运动目标轮廓的方法。在运动场景中，帧与帧有比较明显的区别，两帧相减，可以得到两帧图像亮度差的绝对值，判断是否大于设定的阈值来分析视频帧的运动特性，确定图像序列是否有运动物体。其主要的工作流程如图 2.18 所示。

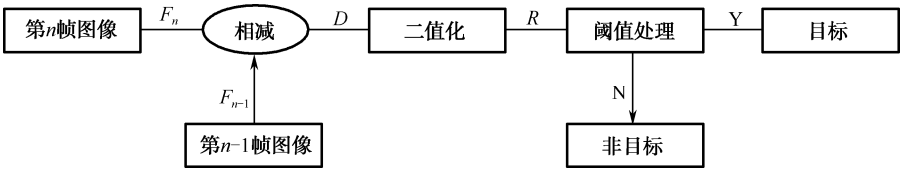


图 2.18　帧间差分法流程

其中，F_n 和 F_{n-1} 分别是第 n 帧和第 $n-1$ 帧的图像序列，将两帧进行差分处理，得到的差值进行二值化处理。将得到的结果与设定的阈值进行比较，判定是属于背景还是前景目标。其中差分公式为

$$D_n(x,y)=|f_n-f_{n-1}(x,y)| \tag{2.16}$$

$$R_n(x,y)=\begin{cases}0,\text{background}\cdots D_n(x,y)>T\\1,\text{foreground}\cdots D_n(x,y)\leqslant T\end{cases} \tag{2.17}$$

其中，T 为阈值，若两帧图像发生变化的位置像素取值为 1，两幅图像间没有发生变化的位置像素取值为 0，这样就检测出运动的目标。

在运动场景目标检测过程中，此方法主要利用时间信息，比较图像中连续帧获得对应点的像素点的灰度值，根据前面设定的阈值，如果大于阈值 T 则可以判定此位置存在运动的目标。利用帧间差分法对运动目标检测的优点是它只对运动目标感兴趣，对非运动区域的感兴趣点比较少，计算量比较小，而且相邻帧的时间间隔很短，一些外部条件如光照等的变化也很小，检测的稳健性得到很大提高。

首先根据帧间差分法对运动场景进行检测，为了加强实验结果的对比，同样选取前面的图像进行运动物体的检测，实验结果如图 2.19 和图 2.23 所示；然后根据代数多重网格插值得到的三层图像，进行帧间差分法仿真实验，分别用两帧中的相同层图像做差分处理，得到实验结果如图 2.20～图 2.22 以及图 2.24～图 2.26 所示。

由上述实验结果可以看出，只利用帧间差分法进行物体检测时受噪声干扰比较大，而使用代数多重网格的帧间差分法能有效去除场景中的噪声，能有效检测出运动目标。

图 2.19　使用原始图像检测结果（简单场景）

图 2.20　使用第一层重建结果检测结果（简单场景）

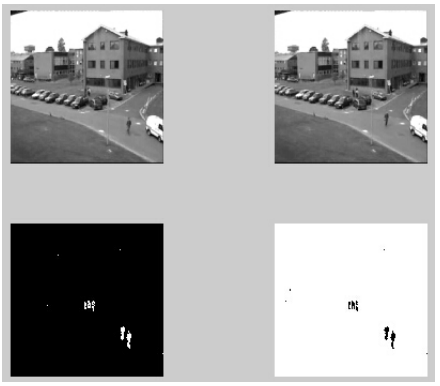


图 2.21　使用第二层重建结果检测结果（简单场景）

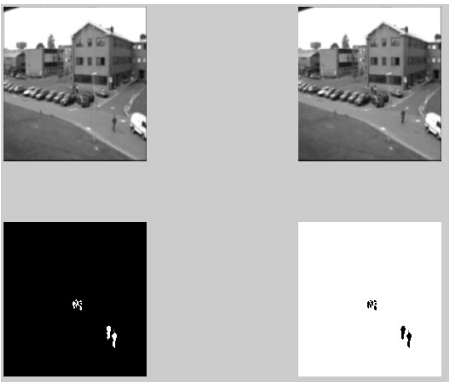


图 2.22　使用第三层重建结果检测结果（简单场景）

图 2.23　使用原始图像检测结果（复杂场景）

图 2.24　使用第一层重建结果检测结果（复杂场景）

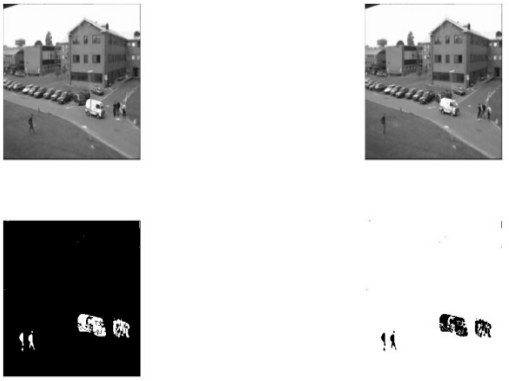


图 2.25　使用原始图像检测结果（复杂场景）

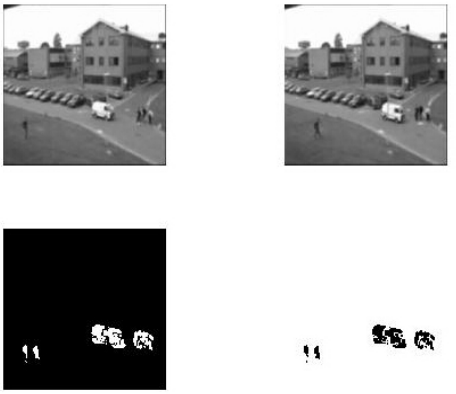


图 2.26　使用第一层重建结果检测结果（复杂场景）

2.4　本章小结

本章根据图像全参考质量评价方法 MSE，引入代数多重网格中粗网格提取算法对图像进行粗网格提取，然后插值重构，再用重构前后的差值来度量图像的清晰度。该算法是无参考的清晰度图像评价方法。算法在计算图像清晰度时相对于图像清晰程度是单调的，清晰度值与图像内容无关，实验结果与主观评价结果是一致的。

第 3 章　使用代数多重网格方法进行特征表达

3.1　图像的特征和特征提取

特征表达是任何物体识别系统的关键。利用颜色、灰度等原始信息形成的直方图[31]特征，用各种轮廓检测算法检测得到的轮廓特征，用各种兴趣点检测方法检测到的像素点特征，通过各种局部描述子对检测到的特征点进行描述，在此基础上进行图像检索和识别。各种特征对于不同环境、不同物体有不同的表达作用。例如对于草地这样的类别，由于图像纹理分布均匀，因此利用 Harris 检测器就不会检测到很多有用的特征点，而如果利用颜色信息来表示这类物体，则会使识别更为容易。

常用的图像特征有颜色特征、纹理特征、形状特征和空间关系特征[32]等。针对每种特征都有对应的特征提取方法。理论上，针对各种特征进行融合才能更有效地进行图像检索和物体识别。

3.1.1　图像的颜色特征提取

在各种基于图像外观信息的特征表达中，除形状特征直观地描述物体外，颜色特征也是不容忽视的一种可以表达物体类别的特征，许多情况下人们对一幅图像的印象往往是从图像的颜色空间分布开始的。颜色特征通常是基于图像像素点的特征，描述了图像或图像区域所对应景物的表面性质，是图像的全局特征，适合于描述不需要考虑空间位置关系的图像。

如何充分、准确地提取一幅图像的颜色信息并以合适的方式描述，将直接影响整个系统的效率和精度。最常用的描述颜色的方法有基于颜色直方图的方法、基于颜色矩的方法，近期也有许多研究人员将颜色特征直接和基于灰度信息的描述符结合在一起形成各种颜色描述符来生成图像特征向量，如 RGB-SIFT、C-SIFT、HSV-SIFT 描述符[33]等。

在不同的颜色空间模式中，颜色特征的计算也不相同。常用的颜色空间有 RGB、HSV、

LUV、LAB、CMYK 等。RGB 颜色空间是最常用的颜色空间，它采用红、绿、蓝三种基色来表示各种颜色，每幅图像都由这三个分量组成。大部分数字图像都是基于 RGB 颜色空间表达的。然而，由于 RGB 颜色空间结构与人们所感知的颜色空间不相符合，于是便提出了更接近于人们对颜色主观认识的 HSV、LUV 和 LAB 等色彩空间模型，各种颜色空间也可以通过计算进行相互转换。下面将结合颜色空间来介绍常用的各种颜色特征描述的方法。

1．颜色直方图

颜色直方图[34-37]是最常用的表达颜色特征的方法，它描述了图像中不同色彩在整幅图像中所占的比例，而不关心每种色彩所处的空间位置，因此可以反映图像颜色的统计分布和基本色调，而无法描述图像中的对象或物体。Swain 和 Ballard 最先提出了使用颜色直方图作为图像颜色特征的表示方法。他们指出，颜色直方图对以观察轴为轴心的图像旋转以及幅度不大的平移和缩放等几何变换是不敏感的，且颜色直方图对于图像质量的变化（如模糊）也有一定的鲁棒性。一维颜色直方图的表示如式（3.1）所述。

$$\begin{aligned} H(P) &= \left[h(x_1), h(x_2), \cdots, h(x_n)\right] \\ h(x_i) &= S(x_i) \Big/ \sum_j S(x_j) \end{aligned} \tag{3.1}$$

其中，$S(x_i)$ 表示某一特征对应像素数；$\sum_j S(x_j)$ 表示图像总像素数，反映了特征的统计分布情况；$h(x_i) \geqslant 0$，$\sum_i h(x_i) = 1$。

在不同的颜色空间下，常用的颜色直方图如下。

（1）RGB 直方图[38]。把三个基于图像 R、G、B 值的一维直方图组合在一起组成图像的颜色直方图。

（2）Opponent 直方图[39]。对色理论认为人类视网膜上存在三种称为视素的光化学物质，每种视素都会发生同化和异化两种变化，同化和异化的发生都是由不同光谱组成的色光刺激的结果，同化过程合成视素，异化过程则分解视素。正是由于同化和异化使人产生相应的对立颜色感，即红-绿、黄-蓝、黑-白等六种不同色觉。基于对色理论，从 RGB 空间到对色空间的变换可表示为

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \dfrac{R-G}{\sqrt{2}} \\ \dfrac{R+G-2B}{\sqrt{6}} \\ \dfrac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{3.2}$$

其中，O_1、O_2 通道表示色彩信息；O_3 通道包含了大部分图像亮度信息。Opponent 直方图是由对色空间（Opponent Color Space）三个通道组成的一维直方图。

（3）Hue 直方图[40]。HSV 颜色空间采用色调（Hue）、饱和度（Saturation）和明度（Value）这三个分量来表示图像颜色信息，成为直方图中最常用的颜色空间。色调值是颜色矢量与参

考矢量的夹角，饱和度值是相对于全饱和色的比例，明度值是图像亮度。设 $M=\max(R,G,B)$，$m=\min(R,G,B)$，色度(Chroma)$C=M-m$，则有

$$H'=\begin{cases}\text{undefined}, C=0\\ ((G-B)/C)\bmod 6, M=R\\ ((B-R)/C)+2, M=G\\ ((R-G)/C)+4, M=B\end{cases}$$

$$H=60^\circ\times H';V=M \tag{3.3}$$

$$S=(M-m)/V$$

（4）rg 直方图。规范化 RGB 颜色空间（Nrgb）是对三基色的加权混合，r 和 g 反映了图像的色度信息（b 保证了 $r+g+b=1$ 的加权条件），rg 直方图只考虑 R、G、B 之间的比例关系

$$\begin{pmatrix}r\\g\\b\end{pmatrix}=\begin{pmatrix}\dfrac{R}{R+G+B}\\ \dfrac{G}{R+G+B}\\ \dfrac{B}{R+G+B}\end{pmatrix} \tag{3.4}$$

r、g、b 称为色度坐标，只有两个坐标是独立的，形成二维色度空间。

（5）颜色变换分布（Transformed color）。由于在 RGB 空间模型中，光强度的变化会引起颜色不规则分布，导致直方图发生偏移，为了增加直方图的稳定性，对 RGB 每个通道的颜色的分布按下式进行规格化，并形成图像直方图。

$$\begin{pmatrix}R'\\G'\\B'\end{pmatrix}=\begin{pmatrix}\dfrac{R-\mu_R}{\sigma_R}\\ \dfrac{G-\mu_G}{\sigma_G}\\ \dfrac{B-\mu_B}{\sigma_B}\end{pmatrix} \tag{3.5}$$

其中 μ_c 和 σ_c 分别表示通道 C（R,G,B）上图像像素的均值和方差。颜色变换分布具有尺度和平移不变性。

2．颜色矩

颜色矩[41]是由 Stricker 和 Orengo 基于图像中任何的颜色分布都可以用相应的矩来表示这一数学基础而提出的一种颜色特征表示方法。由于颜色分布信息主要集中在低阶矩，因此只需采用颜色的一阶矩（Mean）、二阶矩（Variance）和三阶矩（Skewness）就足以表达图像的颜色分布。与其他颜色特征相比，颜色矩表达简单且不需要量化，但在实际应用中，常和其他特征结合使用以避免低次矩较弱的分辨能力。广义的颜色矩定义为

$$M_{pq}^{abc}=\iint x^p y^q\left[I_R(x,y)\right]^a\left[I_G(x,y)\right]^b\left[I_B(x,y)\right]^c \mathrm{d}x\mathrm{d}y \tag{3.6}$$

其中，$p+q$ 表示阶数；$a+b+c$ 表示度数。0 阶矩没有空间信息，0 度矩没有颜色信息。图像 I

在 RGB 空间位置(x,y)处的三个通道的信息分别用 $I_R(x,y)$、$I_G(x,y)$、$I_B(x,y)$ 来表示。广义颜色矩也可以用在任意其他的颜色空间。在图像处理中，将颜色矩离散化表示为

$$M_{pq}^{abc}=\sum_{x}^{I_{\text{width}}}\sum_{y}^{I_{\text{height}}}x^p y^q\left[I_R(x,y)\right]^a\left[I_G(x,y)\right]^b\left[I_B(x,y)\right]^c \tag{3.7}$$

通过组合不同的矩，可实现矩不变量（Moment Invariants），以增加计算为代价换取对各种光照和平移的变化的鲁棒性。

3．颜色 SIFT

由于颜色直方图描述子丢失了颜色的空间分布，人们便将各种色彩空间模型下的信息与 SIFT 描述子相结合提出了不同的颜色 SIFT[42]，如 HSV-SIFT、Hue-SIFT、Opponent-SIFT、rg-SIFT 和 Transformed Color SIFT 等，使基于 SIFT 的彩色描述子得到发展。

HSV-SIFT　通过计算 HSV 颜色空间三个通道信息为每个特征点生成颜色描述子，每个通道的特征都生成 128 维向量，因此总共生成 128×3 维的颜色特征向量。

Hue-SIFT　在 HSV 颜色空间模型中，色调捕捉了颜色的主要波长，表征了图像的彩色信息。VAN de Weijer 将级联色调直方图的方法应用于 SIFT 描述子，为每一特征点生成颜色特征向量。

Opponent-SIFT　在对色空间模型中，为每一分量计算 SIFT 特征描述子，生成 128×3 维的颜色特征向量。

rg-SIFT　在规格化 RGB 模型中，由于 r 与 g 分量具有尺度不变特征，因此不受其光照强度的变化和阴影的影响。rg-SIFT 特征描述符结合了规格化 RGB 彩色模型中 r 分量与 g 分量。

Transformed Color SIFT　计算规格化模型下每个通道分量的 SIFT 描述子，生成 128×3 维的颜色特征向量。

颜色 SIFT 描述子在尺度空间中对彩色图像特征点进行检测，确定特征点的位置，在具体颜色空间模型中统计特征点相关种子点的颜色梯度信息，对每个特征点生成 128×3 维的颜色特征向量。颜色 SIFT 描述子融合了特征点的颜色信息与几何信息，但是特征维度较高，对于数据量较大的图像库，其计算较为复杂。

总之，不同的颜色特征描述法各有特点。理想的颜色特征描述应该具有颜色不变性，即对物体颜色的判断不受光照条件的影响，也可以理解为从图像颜色中恢复物体表面的颜色，人类的视觉具有这种颜色不变性。下面从对角−平移模型（式（3.8）所示）入手来评价各种颜色描述符对光照变化引起的颜色改变的鲁棒性。此处将综合特征描述符的不变性和计算复杂性给出各种描述符的性能对比。

$$\begin{pmatrix}R^c\\G^c\\B^c\end{pmatrix}=\begin{pmatrix}a&0&0\\0&b&0\\0&0&c\end{pmatrix}\begin{pmatrix}R^u\\G^u\\B^u\end{pmatrix}+\begin{pmatrix}o_1\\o_2\\o_3\end{pmatrix} \tag{3.8}$$

式中，等式右边第一项表示尺度变换，第二项表示平移变换。各种颜色特征的性能比较如表 3.1 所示。

表 3.1　各种颜色描述符性能对比

方　　法	维　　数	尺 度 变 化	平 移 变 化	光 色 改 变
RGB 直方图	45	−	−	−
Opponent 直方图	45	−	+	−
Hue 直方图	37	+	+	−
rg 直方图	30	+	−-	−
颜色变换分布	45	+	+	+
颜色矩	30	−	+	−
矩不变量	24	+	+	+
SIFT	128	+	+	−
HSV-SIFT	384	−	−	−
Hue-SIFT	165	+	+	−
Opponent-SIFT	384	+	+	−
rg-SIFT	384	+	−	−
颜色变换 SIFT	384	+	+	+

注："+"表示特征具有某种不变性，"−"表示特征不具有不变性。尺度变化对应对角-平移模型中当参数 $a=b=c$ 发生变化时的情况，平移变化则指 $O_1=O_2=O_3$ 发生变化时的情况，光色改变指 a、b、c 各自发生变化时的变化。对于 a、b、c 和 O_1，O_2，O_3 都发生变化的情况，各种描述符的不变性和 a、b、c 发生变化时的情况相同。

3.1.2　纹理特征

纹理特征[43,44]也是一种全局特征，它也描述了图像或图像区域所对应景物的表面性质。但由于纹理只是一种物体表面的特性，通常要与其他特征结合起来使用。与颜色特征不同，纹理特征不是基于像素点的特征，它需要在包含多个像素点的区域中进行统计计算。在模式匹配中，这种区域性的特征具有较大的优越性，不会由于局部的偏差而无法匹配成功。作为一种统计特征，纹理特征常具有旋转不变性，并且对于噪声有较强的抵抗能力。但是，纹理特征也有其缺点，一个很明显的缺点是当图像的分辨率变化时，所计算出来的纹理可能会有较大偏差。另外，由于有可能受到光照、反射情况的影响，从 2D 图像中反映出来的纹理不一定是 3D 物体表面真实的纹理。例如，水中的倒影、光滑的金属面互相反射造成的影响等都会导致纹理的变化。由于这些非真实的表现，因而将纹理信息应用于检索时，有时这些虚假的纹理会对检索造成"误导"。

在检索粗细、疏密等方面具有较大差别的纹理图像时，利用纹理特征是一种有效的方法。但当纹理之间的粗细、疏密等易于分辨的信息之间相差不大的时候，通常的纹理特征很难准确地反映出人的视觉对不同纹理间的感觉差别。

纹理特征的提取与匹配方法主要有统计方法、几何法、模型法和信号处理法等。而常用纹理的表示方法有灰度共生矩阵、Tamura 纹理特征、自回归纹理模型和小波变换等。

1．统计方法

统计方法的典型代表是一种称为灰度共生矩阵的纹理特征分析方法。Gotlieb 和 Kreyszig 等人在研究共生矩阵中各种统计特征基础上，通过实验得出灰度共生矩阵的四个关键特征：能量、惯量、熵和相关性。统计方法中另一种典型方法，则是通过图像的自相关函数（即图像的能量谱函数）提取纹理特征，即通过对图像的能量谱函数的计算，提取纹理的粗细度及方向性等特征参数。

2．几何法

所谓几何法，是建立在纹理基元（基本的纹理元素）理论基础上的一种纹理特征分析方法。纹理基元理论认为，复杂的纹理可以由若干简单的纹理基元以一定的有规律的形式重复排列构成。在几何法中，比较有影响的算法有两种：Voronio 棋盘格特征法和结构法。

3．模型法

模型法以图像的构造模型为基础，采用模型的参数作为纹理特征。典型的方法是随机场模型法，如马尔可夫（Markov）随机场（MRF）模型法和 Gibbs 随机场模型法。

4．信号处理法

基于信号处理的纹理分析方法受到了视觉心理学解释纹理图像过程的启发，先对图像进行滤波处理，然后再对纹理图像进行分析和解释。现在常用的滤波有傅里叶变换、Laws 纹理、LBP 纹理、小波变换等。

常用的纹理特征表示方法可以用来对纹理特征进行加强，以便更好地进行纹理方面的处理。

灰度共生矩阵特征提取与匹配主要依赖于能量、惯量、熵和相关性四个参数。Tamura 纹理特征基于人类对纹理的视觉感知心理学研究，提出 6 种属性，即粗糙度、对比度、方向度、线像度、规整度和粗略度。自回归纹理模型（Simultaneous Auto-regressive，SAR）是马尔可夫随机场（MRF）模型的一种应用实例。小波反映图像像素点在不同方向上的频率变化，也就反映了图像的纹理特征。

3.1.3　形状特征

各种基于形状特征的检索方法[45,46]都可以比较有效地利用图像中感兴趣的目标来进行检索，但它们也有一些共同的问题，包括：①目前基于形状的检索方法还缺乏比较完善的数学模型；②如果目标有变形时检索结果往往不太可靠；③许多形状特征仅描述了目标局部的性质，要全面描述目标对计算时间和存储量有较高的要求；④许多形状特征所反映的目标形状信息与人的直观感觉不完全一致，或者说，特征空间的相似性与人视觉系统感受到的相似性有差别。另外，从 2D 图像中表现的 3D 物体实际上只是物体在空间某一平面的投影，从 2D 图像中反映出来的形状常不是 3D 物体真实的形状，由于视点的变化，可能会产生各种失真。

通常情况下，形状特征有两类表示方法，一类是轮廓特征，另一类是区域特征。图像的轮廓特征主要针对物体的外边界，而图像的区域特征则关系到整个形状区域。

几种典型的形状特征描述方法如下。

1．边界特征法

该方法通过对边界特征的描述来获取图像的形状参数。其中 Hough 变换检测平行直线法和边界方向直方图法是经典方法。Hough 变换是利用图像全局特性而将边缘像素连接起来组成区域封闭边界的一种方法，其基本思想是点-线的对偶性；边界方向直方图法首先对图像求微分得到图像边缘，然后做出关于边缘大小和方向的直方图，通常的方法是构造图像灰度梯度方向矩阵。

2．傅里叶形状描述符法

傅里叶形状描述符（Fourier Shape Descriptors）基本思想是用物体边界的傅里叶变换作为形状描述，利用区域边界的封闭性和周期性，将二维问题转化为一维问题。

由边界点导出三种形状表达，分别是曲率函数、质心距离、复坐标函数。

3．几何参数法

形状的表达和匹配采用更为简单的区域特征描述方法，如采用有关形状定量测度（如矩、面积、周长等）的形状参数法（Shape Factor）。在 QBIC 系统中，便是利用圆度、偏心率、主轴方向和代数不变矩等几何参数，进行基于形状特征的图像检索。

需要说明的是，形状参数的提取必须以图像处理及图像分割为前提，参数的准确性必然受到分割效果的影响，对分割效果很差的图像，形状参数甚至无法提取。

4．形状不变矩法

利用目标所占区域的矩作为形状描述参数。

5．基于小波和相对矩的形状特征提取与匹配

该方法先用小波变换模极大值得到多尺度边缘图像，然后计算每一尺度的 7 个不变矩，再转化为 10 个相对矩，将所有尺度上的相对矩作为图像特征向量，从而统一了区域和封闭、不封闭结构。

6．其他方法

近年来，在形状的表示和匹配方面的工作还包括有限元法（Finite Element Method，FEM）、旋转函数（Turning Function）和小波描述符（Wavelet Descriptor）等方法。

3.1.4 空间关系特征

所谓空间关系，是指图像中分割出来的多个目标之间的相互的空间位置或相对方向关系，这些关系也可分为连接/邻接关系、交叠/重叠关系和包含/包容关系等。通常空间位置信息可以分为两类：相对空间位置信息和绝对空间位置信息。前一种关系强调的是目标之间的相对情况，如上下左右关系等，后一种关系强调的是目标之间的距离大小以及方位。显而易见，由绝对空间位置可推出相对空间位置，但表达相对空间位置信息通常比较简单。

空间关系特征[47]的使用可加强对图像内容的描述区分能力，但空间关系特征常对图像或目标的旋转、反转、尺度变化等比较敏感。另外，实际应用中仅仅利用空间信息往往是不够的，不能有效准确地表达场景信息。为了检索，除使用空间关系特征外，还需要其他特征来配合。

提取图像空间关系特征可以有两种方法：一种是首先对图像进行自动分割，划分出图像中所包含的对象或颜色区域，然后根据这些区域提取图像特征，并建立索引；另一种方法则简单地将图像均匀地划分为若干规则子块，然后对每个图像子块提取特征，并建立索引。

从第 1 章可以看出，代数多重网格方法对于纹理特征和灰度特征具有较好的提取能力。事实上，代数多重网格方法本身不包含特征信息，而是对特征信息表达的亲和力矩阵进行分析，进而提取能够反映物体关键特征的粗网格，因此代数多重网格方法可以进一步推广到其他特征领域，将各种特征进行亲和力表达，通过代数多重网格方法进行粗网格提取，对物体进行更好地表达。

3.2　图像的兴趣点检测

兴趣点检测主要用于检测图像中显著性的点或区域来表示图像，理想情况下检测到的特征应具有很强的辨识能力，可以应对图像的各种变化，如噪声、光照、尺度、视角等变化，同时对于图像应尽可能具有重复性。常用的特征检测器有 Laplacian of Gaussian（LOG）、Difference of Gaussian（DOG）、Harris-Laplace、Hessian-Laplace、Harris-Affine 和 Hessian-Affine 等，SUSAN 角点检测，Linderberg 算法等。

1．LOG 检测器

LOG 检测[48]采用不同尺度的高斯核对图像做平滑处理用以建立图像尺度空间表示，若某点的三维极值存在，则将该点视为特征点。Lindeberg 用于检测图像中斑点结构的特征。首先将图像与参数为σ的高斯核做卷积，进行平滑处理以去除图像噪声。

$$G_\sigma(x,y)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.9}$$

然后用拉普拉斯算子进行计算。

$$\nabla^2(f(x,y)\otimes G(x,y))=\nabla^2 G(x,y)\otimes f(x,y) \tag{3.10}$$

$$\begin{aligned}\mathrm{LOG}\cong\nabla^2 G_\sigma(x,y)&=\frac{\partial^2}{\partial x^2}G_\sigma(x,y)+\frac{\partial^2}{\partial y^2}G_\sigma(x,y)\\&=\frac{x^2+y^2-2\sigma^2}{\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\end{aligned} \tag{3.11}$$

其中，$f(x,y)$表示原始图像；(x, y)表示图像坐标。

2．DOG 检测器

DOG 检测器[49]是由 David Lowe 提出的特征检测方法，是对 LOG 检测器的一种近似。DOG 算子定义为两个不同尺度高斯核的差分，在图像金字塔中较 LOG 计算速度快。

$$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))\otimes f(x,y) \tag{3.12}$$

3．Harris-Laplace 检测器

Harris 角点检测[50,51]是一种有效的兴趣点检测算法，其原理为：如果图像中的某点在任意方向上产生很小的偏移都会引起灰度值有很大变化，则该点为角点。Harris 角点检测具有

计算简单，对旋转、噪声及光照变化都有较强的鲁棒性。Harris-Laplace 检测是在 Harris 角点检测的基础上，利用 LOG 算子作为尺度选择准则验证每个尺度水平上的候选角点是否为极大值，从而使特征具有尺度不变性。

$$R = \det(C) - k\mathrm{tr}^2(C) > T \tag{3.13}$$

$$C(p,\sigma,s\sigma) = (s\sigma)^2 G(\sigma) \otimes \begin{bmatrix} I_x^2(p,s\sigma) & I_{xy}(p,s\sigma) \\ I_{xy}(p,s\sigma) & I_y^2(p,s\sigma) \end{bmatrix} \tag{3.14}$$

$$I_x(p,s\sigma) = I(p) \otimes G_x(p,s\sigma) \tag{3.15}$$

$$I_y(p,s\sigma) = I(p) \otimes G_y(p,s\sigma) \tag{3.16}$$

其中，I_x 和 I_y 表示图像 P 点在 x 和 y 方向的一阶导数；σ 表示计算特征点时选择的尺度参数；$s\sigma$ 表示微分尺度；k 为经验值，常取为 0.04～0.06；T 为设定的阈值，当某点的 R 值大于阈值时，该点即为特征点；det()表示矩阵的行列式；tr 代表矩阵的迹。

4．Hessian-Laplace 检测器

基于 Hessian 矩阵的 Hessian 检测与 Harris 角点检测的原理相似。由于 Hessian 矩阵计算图像的二阶导数，因此对图像的斑点和屋脊处有较强的响应。同样 Hessian-Laplace 在 Hessian 检测的基础上增加了尺度不变性。

$$M_h = \begin{bmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{bmatrix} \tag{3.17}$$

其中，M_h 为 Hessian 矩阵；I_{xx} 和 I_{yy} 表示图像 P 点在 x 和 y 方向的二阶导数。Hessian-Laplace 检测检测到的特征仅具有旋转和尺度不变性。

5．Harris-Affine 检测器和 Hessian-Affine 检测器

Harris-Affine 和 Hessian-Affine 特征检测是由 Mikolajcxyk 和 Schmid 提出的一种具有仿射不变性的检测器，是对 Harris-Laplace 和 Hessian-Laplace 特征检测器的扩展。基于二阶动差矩阵形状估计特性来实现仿射不变。把图像用 Harris-Laplace 或 Hessian-Laplace 检测后的点作为近似兴趣点，位置和尺度分别为 $x^{(0)}$,$s^{(0)}$，对兴趣点 $x^{(k-1)}$和 $s^{(k-1)}$的邻域采用变化矩阵 $U^{(k-1)}$进行规格化，在规格化块中得出特征尺度 $s^{*(k)}$，更新兴趣点的空间坐标 $x^{*(k)}$并在规格化窗口中估计二阶动差矩阵 $\mu^{(k)}$，根据式（3.18）计算变换矩阵 U。由于采用迭代的方法不断地修正初始未知的变化矩阵，因此仿射不变性检测算法的运行时间很长。

$$U^{(k)} = \prod_k \mu^{(-\frac{1}{2})(k)} U^{(0)} \tag{3.18}$$

$$x^* = \mu^{-\frac{1}{2}} x \tag{3.19}$$

6．SUSAN 算子

SUSAN（Small Univalue Segment Assimilating Nucleus）算子是一种基于灰度的特征点获取方法, 适用于图像中边缘和角点的检测，可以去除图像中的噪声， 它具有简单、有效、抗噪声能力强、计算速度快的特点。

SUSAN 算子的模板与常规卷积算法的正方形模板不同，它采用一种近似圆形的模板，

用圆形模板在图像上移动，模板内部每个图像像素点的灰度值都和模板中心像素的灰度值作比较，若模板内某个像素的灰度与模板中心像素（核）灰度的差值小于一定值，则认为该点与核具有相同（或相近）的灰度，如图 3.1 所示。由满足这一条件的像素组成的区域称为吸收核同值区（Univalue Segment Assimilating Nucleus，USAN）。

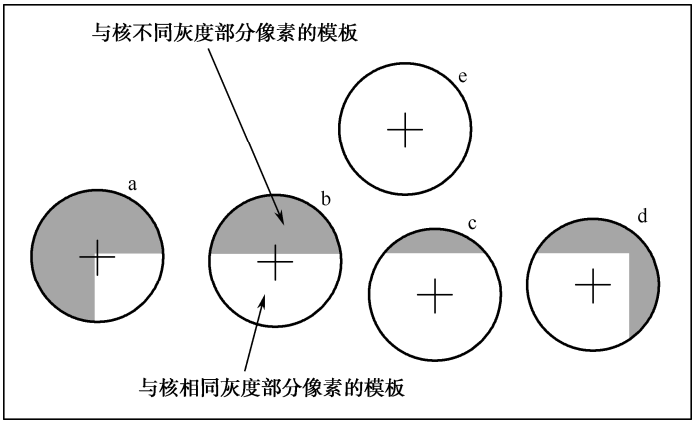


图 3.1　SUSAN 算子的模板

当圆形模板完全处在背景或目标中时，USAN 区域面积最大；当模板移向目标边缘时，USAN 区域逐渐变小；当模板中心处于边缘时，USAN 区域很小；当模板中心处于角点时，USAN 区域最小。

因此，可以通过计算每 1 个像素的 USAN 值，并与设定的门限值进行比较，如果该像素的 USAN 值小于门限值，则该点可以认为是 1 个边缘点，这就是 SUSAN 算法思想。

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：

（1）在用 SUSAN 算子对边缘和角点进行检测时不需要计算微分，这使得 SUSAN 算子对噪声更加鲁棒。

（2）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，与模版尺寸无关，所以 SUSAN 边缘算子的性能不受模版尺寸的影响。

（3）控制参数的选择很简单，且任意性小，容易实现自动化选取。

综上所述，不同检测器检测特征点的属性各有侧重，在实际应用中很难有一种检测器满足所有不变性需求，因此应根据具体应用选择组合不同的检测器，以更完整地表达图像信息，同时又要考虑算法的执行时间，图像信息采集越多，越能完整地表达图像，但计算量越大，而且一些辨别力不强的信息对最后识别产生的影响不大。

3.3　图像的特征描述子

3.3.1　图像的SIFT特征描述子

David G.Lowe 在 2004 年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

保持不变性的图像局部特征描述算子——SIFT 算子，其全称是 Scale Invariant Feature Transform，即尺度不变特征变换[52, 53]。该算子的计算过程包括四个步骤[54]：

（1）检测尺度空间极值点，以初步确定关键点位置和所在尺度。

（2）精确定位极值点，得到关键点，并确定关键点的位置和所处的尺度。通过拟合函数以精确确定关键点的位置和尺度，同时去除低对比度的关键点和不稳定的边缘响应点，以增强匹配稳定性，提高抗噪声能力。

（3）利用关键点邻域像素的梯度方向分布特性为每个关键点指定方向参数，使算子具备旋转不变性。

（4）生成关键点的 SIFT 描述符。首先将坐标轴旋转为关键点的方向，以确保旋转不变性。接下来以关键点为中心取 8×8 的窗口，计算该像素的梯度方向和梯度模值。然后在每 4×4 的小块上计算 8 个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点。实际计算过程中，对于一个关键点可以产生 128 个数据，即形成 128 维的 SIFT 特征向量。此时 SIFT 特征向量已经去除了尺度变化、旋转等几何变形因素的影响，再继续将特征向量的长度归一化，则可以进一步去除光照变化的影响，如图 3.2 所示。

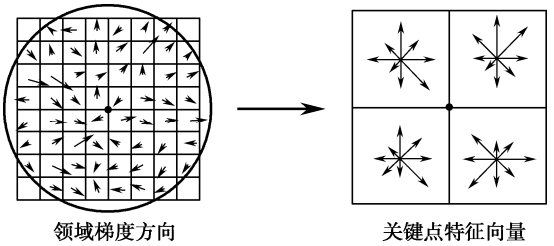


图 3.2　由关键点邻域梯度信息生成特征向量

3.3.2　方向可调滤波器

方向可调是指将一组方向不同的基滤波器进行线性组合，以实现一个滤波器在任意方向上的响应，该响应可以看作方向角的函数，可以通过控制方向角的值来决定滤波器的输出。其中每一个基滤波器都可表示为方向可调滤波器旋转到某一角度时的形式。Freeman 和 Adelso 利用这种思想组合成了不同类型的方向可调滤波器。将 $f(x,y)$表示为滤波器的脉冲响应，方向可调滤波器可表示为

$$f^{\theta}(x,y)=\sum_{j=1}^{M}k_j(\theta)f^{\theta_j}(x,y) \tag{3.20}$$

其中，$f^{\theta}(x,y)$ 为 $f(x,y)$ 旋转 θ 的脉冲响应；$f^{\theta_j}(x,y)$ 为基滤波器；M 为基滤波器个数；$k_j(\theta)$ 为第 j 个插值函数；θ_j 为第 j 个基角。由式（3.20）可知，任意方向的滤波器都可由有限个基滤波器线性组合得到。方向可调滤波器作为特征描述符可以保持特征区域的旋转不变性。

3.3.3　微分不变量

设 x 为图像 I 中的一个点，δx 表示点 x 的邻域，图像灰度值 $f(x)$的泰勒展开式为

$$f(x+\delta x)=\sum_{k=0}^{n}\frac{1}{k!}f_{i1\cdots ik}(x)\delta x_{i1}\cdots x_{ik}+O(\|\delta x\|^{n+1}) \tag{3.21}$$

由式（3.21）可以看出，图像在某邻域内的灰度信息可以用它的一系列微分算子描述。由于图像微分（特别是高阶微分）容易受噪声的干扰，常采用高斯微分算子与图像卷积来计算图像的微分。Koenderink 将卷积后得到的一系列算子称为“local jet”，这些微分算子合称微分不变量描述符。

N 阶“local jet”的定义如下

$$J^N[I](x,\sigma)=\{L_{i_1\cdots i_n}(x,\sigma)\,|\,(x,\sigma)\in I\times R^+;n=0,\cdots N\} \tag{3.22}$$

其中，$L_{i_1\cdots i_n}$（x,σ）表示图像 I 与高斯微分算子的卷积，$i_n\in\{x,y\}$，高斯微分算子 $G_{i_1\cdots i_n}$ 的形式如下

$$G_{i_1\cdots i_n}(x,\sigma)=\frac{\partial^n}{\partial_{i_1}\cdots\partial_{i_n}}G(x,\sigma) \tag{3.23}$$

由二阶微分不变量描述的图像特征向量如式（3.24）所示。

$$S=\begin{bmatrix} L \\ L_xL_x+L_yL_y \\ L_{xx}L_xL_x+2L_{xy}L_xL_y+L_{yy}L_yL_y \\ L_{xx}+L_{yy} \\ L_{xx}L_{xx}+2L_{xy}L_{xy}+L_{yy}L_{yy} \end{bmatrix} \tag{3.24}$$

S 描述了像素邻域内微分结构的信息，只与点的邻域内容有关，而与点在图像中的位置无关，因此 S 是平移、旋转不变的。由式（3.24）看出，S 中第一个因子表示了图像的平均亮度，第二个因子表示梯度幅值的平方，第四个因子是 Laplace 算子。由于 S 的各因子是由原图像与高斯微分相卷积得到的，而与高斯微分的卷积相当于图像平滑的过程，因此描述子对噪声也有很好的抑制性。微分不变量描述子在目标识别与检测中得到了广泛的应用。

3.3.4　LBP纹理特征描述符

LBP（Local Binary Pattern）方法既考虑了局部像素值之间的关系，同时也考虑了全局模式特征的统计分布情况，是用在纹理分析和人脸识别领域比较成功的一种方法，关于它的改进工作不断出现在各种文献中。但是它只考虑了局部像素值之间的大小关系，没有考虑像素值之间的对比度，虽然这在一定程度上可以消除光照变化带来的影响，但是如果能够通过图像预处理，将光照变化控制在一定的范围之内，将对比度考虑在传统 LBP 方法之内，就可以得到更好的识别效果。

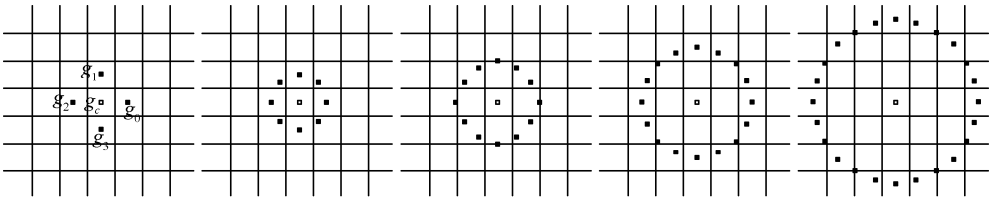
首先定义一个纹理算子 T，用来描述单色灰度纹理图像中的局部像素值分布。

$$T=t(g_c,g_0,\cdots,g_{P-1}) \tag{3.25}$$

其中，g_c 代表局部中心像素的灰度值；$g_p\,(p=0,\cdots,P-1)$代表以 g_c 为中心、$R(R>0)$为半径的圆周上的像素灰度值。

如果以 g_c 为坐标原点，g_p 的坐标为（$-R\sin(2\pi p/P),R\cos(2\pi p/P)$），当然，$g_p$ 有可能并不在像素中心点，此时可以通过插值的方法来获取 g_p 的灰度值。图 3.3 显示了在不同的（P,R）

取值情况下的分布情况。



（P=4, R=1.0）　（P=8, R=1.0）　（P=12, R=1.5）　（P=16, R=2.0）　（P=24, R=3.0）

图 3.3　纹理算子 T 所描述的局部区域像素分布情况

3.3.5　基于场景的视觉描述符

Wu 等人于 2009 年提出了一种基于场景分类的视觉描述符——CENTRIST 描述符，其全称是 Census Transform Histogram，即普查变换直方图。计算 CENTRIST 描述符包括以下两个过程。

1．普查变换（Census Transform）

最初，普查变换（CT）是为确立局部图像片之间的对应问题而设计的一种非参数局部变换，它是将一个中心像素的亮度值与该像素的 8 个邻域像素的亮度值作比较，如果该中心像素的亮度值是大于或等于它的某个邻域的亮度值，则该邻域像素的相应位置设置为 1，否则设置为 0，如图 3.4 所示。从亮度值比较产生的 8 位二进制数可以通过任何顺序放在一起，从而被变换成一个基于十进制的在 0 到 255 之间的数值。这个值就是这个中心像素的普查变换值（CT 值）。普查变换是基于亮度值比较，因此它对光照变化和伽玛变化等具有鲁棒性。

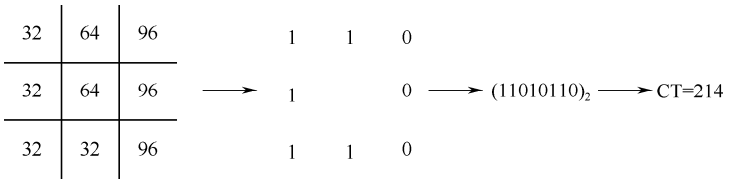


图 3.4　普查变换

2．获得 CENTRIST 描述符

普查变换直方图是 CT 值的函数，它表示图像中具有每种 CT 值的像素的个数，反映图像中每种 CT 值出现的频率。从图形上说，普查变换直方图是一个二维图，用坐标表示，横坐标表示图像中各个像素点的 CT 值（0 到 255 之间的数值），纵坐标为各个 CT 值上图像各个像素点出现的次数或概率。

代数多重网格方法本身也具备特征检测能力，它对特征进行分拣，将能更好地反映物体本质的特征提取出来，而将对物体本质反映不强的特征去掉。这对于物体识别和图像检索也是很重要的。同时，也可以对上述特征描述子做进一步特征分拣，本章 3.7.2 节会对 SIFT 特征描述子使用代数多重网格方法进行进一步提取，在一定程度上提高了 SIFT 特征区分物体的能力。

3.4 图像的特征不变性

本节以 LBP 特征描述符为例，讲述各种特征不变性的构建过程。

3.4.1 灰度尺度不变性

将每个圆周对称的邻居点像素值减去中心点像素值 g_c，可得到如下纹理算子。

$$T = t(g_c, g_0 - g_c, \cdots, g_{P-1} - g_c) \tag{3.26}$$

假设 g_p 的值与 g_c 的值没有关系。可以将式（3.26）变为

$$T = t(g_c)t(g_0 - g_c, \cdots, g_{P-1} - g_c) \tag{3.27}$$

实际上，这种假设并不总是成立的，因此式（3.27）表示的是一个估计值，但是我们可以接受这个差异。$t(g_c)$ 只表示了 g_c 点的像素值，它反映了实际图像的光照情况，其值会随着光照的变化而变化；同时，g_c 不能反映任何局部纹理特征，也就不能为纹理分析提供任何有用的信息。所以，可以将式（3.27）中的 $t(g_c)$ 项移除，得到能反映局部纹理特征的算子，即

$$T \approx t(g_0 - g_c, \cdots, g_{P-1} - g_c) \tag{3.28}$$

此时，T 是一个特征描述能力很强的纹理算子，它能够记录局部像素值之间的各种模式。在常量区域，各个方向邻居点与中心点的灰度差值均为 0；在缓慢变化的斜面边界，T 算子在梯度方向具有较大的差值而沿着边界方向差值却为 0；对于突出点而言，T 算子在各个方向的差值都比较大。光照变化可以引起图像灰度尺度变化，光照强烈时，整幅图像总体变得明亮，灰度值整体偏大（0 表示黑色，255 表示白色）；光照微弱时，整幅图像总体变得黑暗，灰度值整体偏小。由于光照变化给像素灰度值带来的变化并不是线性的，所以（$g_p - g_c$）的值在不同光照下是不同的，它不具有灰度尺度不变性。但是（g_p–g_c）的符号却是随光照变化而不变的，它反映的是像素值之间的大小关系，光照带来的非线性变化也不会改变这个符号值，它具有灰度尺度不变性。

因此，将式（3.28）转换为

$$T \approx t(s(g_0 - g_c), \cdots, s(g_{P-1} - g_c)) \tag{3.29}$$

其中

$$s(x) = \begin{cases} 1, x \geqslant 0 \\ 0, x < 0 \end{cases}$$

通过对每一个 $s(g_p - g_c)$ 指定一个二项式因子 10^3 ，可以将式（3.29）转换为一个能够描述图像局部纹理特征的 $\mathrm{LBP}_{P,R}$ 值，即

$$\mathrm{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{3.30}$$

$\mathrm{LBP}_{P,R}$ 算子也具有灰度尺度不变性，只要光照变化不改变局部像素值的大小顺序，那么 $\mathrm{LBP}_{P,R}$ 算子得到的值也是相同的。

3.4.2 旋转不变性

$\mathrm{LBP}_{P,R}$ 算子具有 2^P 个不同的取值，对应到由 P 个邻近像素构成的 2^P 个不同的二进制模式。当图像旋转时，g_p 也会沿着 g_c 为圆心，R 为半径的圆周旋转。由于 g_0 始终被指定为 g_c 右侧坐标为（0, R）的邻近点，所以 $\mathrm{LBP}_{P,R}$ 会得到不同的值。为了消除旋转对于 $\mathrm{LBP}_{P,R}$ 算子的影响，定义了一个新的具有旋转不变性特征的纹理算子 $\mathrm{LBP}_{P,R}^{ri}$

$$\mathrm{LBP}_{P,R}^{ri} = \min\{\mathrm{ROR}(\mathrm{LBP}_{P,R}, i) \mid i = 0,1,\cdots,P-1\} \tag{3.31}$$

其中 ROR(x,i)表示对 P 位数字 x 进行 i 次的右向循环移位。那么，$\mathrm{LBP}_{P,R}^{ri}$ 就代表了局部邻近像素值在不同旋转角度下的最小值，无论旋转角度有多大，这个最小值总是不变的，所以 $\mathrm{LBP}_{P,R}^{ri}$ 具有旋转不变性特征。图 3.5 给出了在 P=8 的情况下，$\mathrm{LBP}_{P,R}^{ri}$ 可能的 36 种取值。

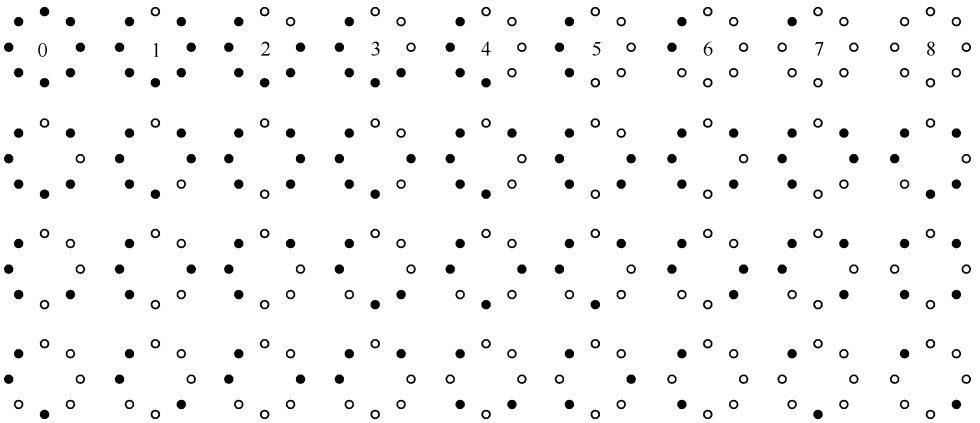


图 3.5　$\mathrm{LBP}_{P,R}^{ri}$ 所得到的各种取值示意（黑色表示 1，白色表示 0）

图中第 0 号取值代表了亮点，第 8 号取值代表了黑点和平坦区域，第 4 号取值代表了边缘。

3.4.3 统一模式特征

纹理图像中 $\mathrm{LBP}_{P,R}^{ri}$ 的值出现概率并不是一样的。经过统计，Ojala 发现一部分取值会大量出现，因而这些 $\mathrm{LBP}_{P,R}^{ri}$ 值代表了纹理的最基本特征。当 $P = 8$ 时，图 3.5 中第一行显示的 9 种模式的出现概率甚至达到 90%以上。这些大量出现的模式有一个共同特征，那就是它们内部出现的 0-1 跳转或者 1-0 跳转数非常少，甚至为 0，

为了准确描述这些最基本特征，定义了一种统一度量方法 U，它对应了模式内部循环位 0-1 跳转的次数。例如，模式 00000000 和模式 11111111 的 U 值为 0。定义 U 值不大于 2 的模式为统一模式（Uniform Pattern），则可使用下述算子来替代，即

$$\mathrm{LBP}_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{当 } U(\mathrm{LBP}_{P,R}) \leqslant 2 \\ P+1 & \text{其他} \end{cases} \tag{3.32}$$

其中 $U(\mathrm{LBP}_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{P-1} - g_c)|$。

式（3.32）中，上标 *riu*2 表示了 U 值为 2 的具有旋转不变性特征的 LBP 算子。通过这种改进，二进制模式的种类大大减小，而不会丢失任何信息。模式数量由原来的 2^P 种减少为 $P(P-1)+2$ 种，使得特征向量的维数更少，并且可以减少高频噪声带来的影响。

在使用 $\mathrm{LBP}_{P,R}^{riu2}$ 值进行纹理分析时，先求得图像中 $\mathrm{LBP}_{P,R}^{riu2}$ 值的直方图，将此直方图作为整幅图像的纹理特征，再与其他图像进行比较。如果不使用统一模式特征，而只是使用 $\mathrm{LBP}_{P,R}^{ri}$ 构造的直方图，由于取值过于分散，使得直方图比较稀疏，就不能准确代表整幅纹理图像的特征。

3.5　特征选择

所谓特征选择，就是从一组数量为 N 的特征中选择出一组数量为 M 的最优特征（$N>M$），这里有两个问题要解决：（1）选择一种可分性判据作为最优特征选择的标准；（2）找到一个好的算法，来选择出这组最优特征。下面介绍两种特征选择的思路。

一个最简单的思路是：假设 N 个特征之间相互独立，并且使用的可分性判据满足可加性 $J(X)=\sum_{i=1}^{N}J(x_i)$，这时候只要把 N 个特征每个单独使用时的可分性判据 $J(x_i)$ 计算出来，然后从大到小排序：$J(x_1)>J(x_2)>\cdots>J(x_N)$，选择出前 M 个特征就是一组最优的特征。然而问题往往没有这么简单，这种特征独立性假设多数情况下并不成立，并且可分性判据也不一定满足可加性。

另外一个简单的思路是（穷举法）：对从 N 中选择出 M 个特征的所有组合情况都计算其可分性判据，然后选择出其中的最大者作为解决方案。当 N 的数值比较小时，这种方法一定是可行的，然而当 N 比较大时，这个组合数会非常大，比如 $N=100$，$M=10$ 时，组合数的数量级是 10^3，当 $N=20$，$M=10$ 时，组合数为 184756。将所有的组合都计算一遍显然是不现实的，因此我们需要有一个搜索算法来进行特征选择。

3.5.1　特征的评价准则

在讨论特征选择和特征压缩之前，我们先要确定一个选择和提取的原则。对一个原始特征来说，特征选择的方案很多，从 N 维特征中选择出 M 个特征共有 $C_N^M=N!/M!(N-M)!$ 种选法，其中哪一种方案最佳，则需要有一个原则来进行指导。同样，特征的压缩实际上是要找到 M 个 N 元函数，N 元函数的数量是不可数的，这也要有一个原则来指导找出 M 个最佳的 N 元函数。

进行特征选择和特征提取的最终目的还是要进行识别，因此应该把对识别最有利作为原则，这样的原则被称为是类别的可分性判据。用这样的可分性判据可以度量当前特征维数下类别样本的可分性。可分性越大，对识别越有利，可分性越小，对识别越不利。

人们对特征的可分性判据研究很多，然而到目前为止还没有取得一个完全满意的结果，没有哪一个判据能够完全度量出类别的可分性。下面介绍几种常用的判据，我们需要根据实际问题，从中选择出一种。

一般来说，可分性判据希望满足以下几个条件：

（1）与识别的错误率有直接的联系，当判据取最大值时，识别的错误率最小。

（2）特征独立时有可加性，即

$$J_{ij}(x_1,x_2,\cdots,x_N)=\sum_{k=1}^{N}J_{ij}\left(x_k\right)$$

其中，J_{ij} 是第 i 类和第 j 类的可分性判据，J_{ij} 越大，两类的可分程度越大；$(x_1,x_2,\cdots,x_N)$ 为 N 维特征。

（3）应具有某种距离的特点，即

$J_{ij}>0$，当 $i\neq j$ 时

$J_{ij}=0$，当 $i=j$ 时

$J_{ij}=J_{ji}$

（4）单调性，加入新的特征后，判据不减小，即

$$J_{ij}(x_1,x_2,\cdots,x_N)\leqslant J_{ij}(x_1,x_2,\cdots,x_N,x_{N+1})$$

但是遗憾的是现在经常使用的各种判据很难满足上述全部条件，只能满足一个或几个条件。

3.5.2 类别可分离性判据

特征选择与提取的任务是求出一组对分类最有效的特征，因此我们需要一个定量的准则（或称判据）来衡量特征对分类的有效性。具体说来，把一个高维空间变换为低维空间的映射是很多的，哪种映射对分类最有利，需要一个比较标准。从 D 个原始特征中选择出 d 个特征的各种可能组合也是很多的，哪种组合的分类效果最好，也要有一个比较标准。

或者我们会很自然地想到，既然我们的目的是设计分类器，那么以分类器的错误概率作为标准就行了，也就是说，使分类器错误概率最小的那组特征，就应当是一组最好的特征。从理论上说，这是完全正确的，但在实用中却有很大困难。

散度矩阵常用来作为可分性判据，常用的散度矩阵如下。

1．类内散度矩阵

设有 M 个类别，$\Omega_1,\cdots,\Omega_M$，Ω_i 类样本集 $\{x_1^{(i)},x_2^{(i)},\cdots,x_N^{(i)}\}$，$\Omega_i$ 类的散度矩阵定义为

$$m^{(i)}=\frac{1}{N_i}\sum_{k=1}^{N_i}X_k^{(i)} \tag{3.33}$$

$$S_w^{(i)}=\frac{1}{N_i}\sum_{k=1}^{N_i}(X_k^{(i)}-m^{(i)})(X_k^{(i)}-m^{(i)})^T \tag{3.34}$$

其中，$m^{(i)}$ 表示 Ω_i 类的平均值；$S_w^{(i)}$ 表示 Ω_i 的类内散度矩阵。总的类内散度矩阵为

$$S_w=\sum_{i=1}^{M}P(\Omega_i)S_w^{(i)}=\sum_{i=1}^{M}P(\Omega_i)\frac{1}{N_i}\sum_{k=1}^{N_i}(X_k^{(i)}-m^{(i)})(X_k^{(i)}-m^{(i)})^T \tag{3.35}$$

2．类间散度矩阵

第 i 个类别和第 j 个类别之间的散度矩阵定义为

$$S_B^{(ij)} = (m^{(i)} - m^{(j)})(m^{(i)} - m^{(j)})^T \tag{3.36}$$

总的类间散度矩阵可以定义为

$$S_B = \frac{1}{2}\sum_{i=1}^{M} P(\Omega_i)\sum_{j=1}^{M} P(\Omega_j) S_B^{(ij)} = \frac{1}{2}\sum_{i=1}^{M} P(\Omega_i)\sum_{j=1}^{M} P(\Omega_j)(m^{(i)} - m^{(j)})(m^{(i)} - m^{(j)})^T \tag{3.37}$$

令 m 为总体均值

$$m = \sum_{i=1}^{M} P(\Omega_i) m^{(i)} \tag{3.38}$$

则有

$$S_B = \sum_{i=1}^{M} P(\Omega_i)(m^{(i)} - m)(m^{(i)} - m)^T \tag{3.39}$$

3．总体散度矩阵

总体散度矩阵可以定义为

$$S_T = \frac{1}{N}\sum_{l=1}^{N}(X_l - m)(X_l - m)^T \tag{3.40}$$

其中 N 为总的样本数，$N = \sum_{i=1}^{M} N_i$ 。可以证明：$S_T = S_W + S_B$ 。 （3.41）

可以看出三个散度矩阵均为实对称矩阵。

根据散度矩阵，可以定义可分性判据为 $J_1(X) = \mathrm{tr}(S_T) = \mathrm{tr}(S_W + S_B)$ 。 tr 表示取一个矩阵的迹，也就是主对角线元素之和，N 维方阵 A 的迹为 $\mathrm{tr}(A) = \sum_{i=1}^{N} a_{ii}$ 。

同样也可以用三个散度矩阵构建出一系列的可分性判据。

$$J_2 = \frac{\mathrm{tr}(S_B)}{\mathrm{tr}(S_W)}$$

$$J_3 = \ln\frac{|S_B|}{|S_W|}$$

$$J_4 = \mathrm{tr}(S_W^{-1} S_B)$$

$$J_5 = \frac{|S_B + S_W|}{|S_W|}$$

其中 $|A|$ 表示方阵 A 的行列式的值。比较常用的判据是 J_1 。

基于几何距离的可分性判据计算起来比较简单，只要知道各个类别的训练样本集，就可以计算出三个散度矩阵，同时也就可以计算出各种可分性判据。

3.5.3　特征子集的选择

特征子集选择 FSS（Feature Subset Selection）是归纳学习的一个重要研究内容，也是机器学习和模式识别中非常困难而有意义的一个课题, FSS 问题是指从一个大的候选特征集合中选择一个较好的有代表性的子集来一致地描述已知子集。过多的特征会严重地影响归纳学习的质量，一些不必要的特征会使噪声增大，损害所生成的规则的精度。因此，一个特征子集越小，所选特征的代表性可能越强，从而产生的规则质量越高。由此可见，特征子集的良

好选取能够给归纳学习的实际应用带来很大益处。

国内外学者在特征子集选择的问题上做了大量的工作。我国学者陈彬证明了最小特征子集选择问题（OFSS 问题）是 NP 难题。因此求其次，寻找一个较好特征子集选择的启发式算法是必然的。

陈彬从扩张矩阵理论出发，给出了最优特征子集的概念和一个特征子集选择的贪心算法 GFS（Greedy Feature Subset selection）。设 EM 是 PE（正例集）在 NE（反例集）背景下的连接扩张矩阵，则最优特征子集问题等价于在 EM 中找到一条非死元素组成的路径，该路径涉及最少数目的列，即特征。最优特征子集就是从学习实例的特征集合中寻找一个具有最小数目的特征子集来一致地覆盖学习实例。GFS 的基本思想是首先建立连接扩张矩阵 EM（PE/NE），然后在 EM（PE/NE）找到一列，该列的非死元素最多，把该列加入到最小特征子集中，并把 EM（PE/NE）中该列为非死元素的行删掉，重复以上步骤，直到 EM（PE/NE）为空。GFS 并不具体地生成连接扩张矩阵，只计算死元素的个数及所删除的行数，因此大大节省了存储空间。

然而，GFS 是基于正反例一致的条件下的精确学习方法，所以在处理含有噪声数据的学习问题时，效果不太理想，这也是 GFS 的一个不足之处。

研究发现，在进行学习之后得到的规则集中，去掉一些覆盖例子很少的复杂规则之后，预测识别的精度并没有减少。并且，在那些被去掉的复杂规则中，包含有较多的特征，其中某些特征只在这些规则中出现过，这说明，噪声数据的存在可能对特征子集的选择带来较大的困难，无法找到较优的特征子集。

3.6 特征优化

原始特征的数量可能很大，或者说样本是处于一个高维空间中，通过映射（或变换）的方法获取最有效的特征，实现特征空间的维数从高维到低维的变换，以便用低维空间来表示样本，这个过程叫特征优化。

经过映射后的特征称为二次特征，它们是原始特征的某种组合，最常用的是线性组合。所谓特征优化在广义上就是指一种变换。

3.6.1 基于类别可分性判据的特征优化

若使用类别可分性判据作为新特征的准则，则特征优化的问题就是求最优的 W^*，使

$$W^* = \arg\max J\left(W^T x\right) \tag{3.42}$$

其中，$J()$ 可以是基于类内类间距离的类别可分性判据，也可以是基于概率距离或熵的可分性判据。

如果采用基于类内类间距离的类别可分性判据 J_1～J_5，经过 W 的特征变换后，类内离散度矩阵和类间离散度矩阵分别变为 $W^T S_w W$ 和 $W^T S_b W$，则特征优化的问题就是求 W^*。

使下列准则最优

$$J_1(W) = \mathrm{tr}(W^T (S_w + S_b) W)$$

$$J_2(W)=\mathrm{tr}[(W^TS_wW)^{-1}(W^TS_bW)]$$

$$J_3(W)=\ln\frac{\left|W^TS_bW\right|}{\left|W^TS_wW\right|}$$

$$J_4(W)=\frac{\mathrm{tr}(W^TS_bW)}{\mathrm{tr}(W^TS_wW)}$$

$$J_5(W)=\frac{\left|W^T(S_w+S_b)W\right|}{\left|W^TS_wW\right|}$$

这些准则虽然形式不同，但得到的最优变换矩阵是相同的，如下所述：

设矩阵 $S_w^{-1}S_b$ 的本征值为 $\lambda_1,\lambda_2,\cdots,\lambda_D$，按大小顺序排列为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_D$，则选前 d 个本征值对应的本征向量作为 W，即

$$W=\left[u_1,u_2,\cdots,u_d\right] \tag{3.43}$$

所构成的变换矩阵就是在这些准则下的最优变换矩阵。

下面，以 J_1 准则为例给出这一结论的推导过程，其余准则下的推导读者可以作为练习。

J_1 准则为

$$J_1(W)=\mathrm{tr}(W^T(S_w+S_b)W) \tag{3.44}$$

要最大化 J_1，首先遇到一个问题就是，无论选取什么变换矩阵 W，只要把它再乘以一个系数，则准则函数值会再变大，但变换的方向并没有改变，这就是尺度问题。为了解决这一问题，引入一个约束条件 $\mathrm{tr}(W^TS_wW)=c$，不妨设 $c=1$。优化问题变为

$$\begin{aligned}&\max J_1(W)\\ &\text{s.t.}\qquad \mathrm{tr}(W^TS_wW)=1\end{aligned} \tag{3.45}$$

采用拉格朗日方法可以把这个有约束的优化问题变成无约束问题，拉格朗日函数是

$$g(W)=J_1(W)-\mathrm{tr}\left[\Lambda(W^TS_wW-I)\right] \tag{3.46}$$

其中，I 是单位矩阵；Λ 是对角阵，对角线元素是拉格朗日乘子。在拉格朗日函数的极值点上，应该满足 $\dfrac{\partial g(W)}{\partial W}=0$，由此可得

$$S_w^{-1}(S_w+S_b)W=W\Lambda \tag{3.47}$$

整理得

$$S_w^{-1}S_bW=W(\Lambda-I) \tag{3.48}$$

可见，W 由 $S_w^{-1}S_b$ 的本征向量组成，$\Lambda-I$ 等于 $S_w^{-1}S_b$ 对应的本征值 λ_i 组成的对角阵，即

$$\Lambda=I+\begin{bmatrix}\lambda_1 & & \\ & \cdots & \\ & & \lambda_D\end{bmatrix} \tag{3.49}$$

考虑到（3.45）中的条件和式（3.47）有

$$J_1(W)=tr(W^T(S_w+S_b)W)=\mathrm{tr}(W^TS_wW\Lambda)=\mathrm{tr}\Lambda \tag{3.50}$$

对于 $D\times d$ 维变换矩阵

$$J_1(W)=\sum_{i=1}^{d}(1+\lambda_i) \tag{3.51}$$

因此，最优的变换矩阵W就是由$S_w^{-1}S_b$的前d个本征值所对应的本征向量组成，而所得的J_1准则值由式（3.51）定义。

也可以采用基于概率距离的判据或基于熵的判据作为准则来进行特征提取。但一般情况下只能靠数值求解，在数据服从正态分布并满足某些特殊条件时可以得到形式化的解。

3.6.2 主成分分析法

基于线性的特征提取算法适合处理线性结构的数据。这类方法因其算法思想简单、直观，计算方便，而且有计算速度快等特点，因此广泛应用于数据降维。这类方法的原理是在高维空间中找出一个低维的线性子空间，并将高维数据映射到该子空间中。下面介绍经典的线性特征提取算法——主成分分析法。

主成分分析方法[55,56]简单来说就是一种正交变换，它是通过 K-L 变换将高维空间中的数据投影到低维空间中。首先介绍 K-L 变换原理：假设X是n维随机向量，则X可由一组正交向量基$\Phi=(\phi_1,\phi_2,\cdots,\phi_n)$来表示，即

$$X=\sum_{i=1}^{n}a_i\phi_i \tag{3.52}$$

其中a_i为加权系数。用矩阵的形式可以表示为

$$X=(\phi_1,\phi_2,\cdots,\phi_n)(a_1,a_2,\cdots,a_n)^T=\Phi a \tag{3.53}$$

其中，$\Phi=(\phi_1,\phi_2,\cdots,\phi_n)$；$a=(a_1,a_2,\cdots,a_n)^T$。

因为我们取的向量基为正交向量基，即

$$\phi_i\phi_j=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases} \tag{3.54}$$

即Φ中的每个向量都是正交向量，所以Φ是正交矩阵，即

$$\Phi^T\Phi=I \tag{3.55}$$

在式（3.53）两边同时左乘Φ^T，并考虑Φ是正交矩阵，得到

$$\Phi^T X=\Phi^T\Phi a \tag{3.56}$$

即

$$a=\Phi X$$

$$a_i=\phi_i X$$

接下来我们介绍怎样找出正交向量集$\Phi=\left(\phi_1,\phi_2,...,\phi_n\right)$，使向量$a$的各个向量互不相关，假设随机向量$X$的自相关矩阵为

$$R=E(XX^T) \tag{3.57}$$

将式（3.52）代入式（3.57）得

$$R=E(XX^T)=E(\Phi aa^T\Phi^T)=\Phi E(aa^T)\Phi^T \tag{3.58}$$

我们要使得向量a的各个分量互不相关，即要满足下列关系

$$E(a_j a_k^T)=\begin{cases}\lambda_i & j=k\\ 0 & j\neq k\end{cases}=\begin{bmatrix}\lambda_1 & & & & 0\\ & \ddots & & & \\ & & \lambda_i & & \\ & & & \ddots & \\ 0 & & & & \lambda_n\end{bmatrix}=\Lambda \tag{3.59}$$

则

$$R=\Phi\Lambda\Phi^T \tag{3.60}$$

在式（3.60）两边同时右乘Φ，得

$$R\Phi=\Phi\Lambda\Phi^T\Phi \tag{3.61}$$

由于Φ是正交矩阵，即得

$$R\Phi=\Phi\Lambda \quad 即 \quad R\phi_i=\lambda_i\phi_i \quad (i=1,2,\cdots,n) \tag{3.62}$$

由此可知，λ_i是X自相关矩阵R的特征值，ϕ_i就是对应特征值λ_i的特征向量。因为R是实对称的，其不同特征值对应的特征向量是正交的。

从上面的主成分分析法的原理中我们可以看出，主成分分析法可有效降低变量维数，还可以简化向量的统计特征；可以减少代数运算中噪声造成的误差；可以直观地可视化地显示高维数据。但主成分分析法基于很多假设条件，这些条件直接影响到主成分分析方法的应用。其主要假设如下。

（1）线性假设。主成分分析原理是基于线性 K-L 变换，这使得它所处理的数据之间的关系也是线性的，如果是非线性数据，会使主成分分析法效果不明显。

（2）统计的工具是中值和方差。它所能描述的模型仅限于像高斯分布这样的指数型概率分布，对于其他的模型不能很好地描述。

（3）方差大的向量，重要性越大。由于数据自身的高信噪比，把方差大的一维向量作为主元，其他的作为噪音，这也受低通滤波器的作用。

（4）主元正交。K-L 变换的前提条件是向量之间是正交的，这也是方便数学运算，提高应用效率。

3.6.3　非线性维数降低

维数约简方法可分为线性方法和非线性方法，其线性方法主要指线性降维技术，如主成分分析（PCA）、多维尺度分析（MDS）、独立分量分析（ICA）、典型相关分析（CCA）以及线性判别分析（LDA）等算法，在高维观测数据集具有线性结构时，这些算法具有良好的效果；　非线性的典型算法如核主成分分析（KPCA）、等度规映射（ISOMAP）、最大方差展开（MVU）、拉普拉斯特征映射（LE）、局部线性嵌入（LLE）、局部切空间排列算法（LTSA）、局部保持投影（LPP）等，主要目的是发现隐含在高维数据中的低维非线性结构。本节重点介绍局部保持投影和局部线性嵌入这两种方法。

1．局部线性嵌入算法

局部线性嵌入算法（Locally Linear Embedding，LLE）是流形学习方法中最常用的算法之一，该算法是 S.T.Roweis 和 L.K.Saul 等人针对非线性数据提出的一种新的无监督的降维方

法，它是流形学习算法中的一种局部方法，能够使降维的数据保持原有的拓扑结构。

从子空间思想的角度分析，LLE 是一种依赖于局部线性的非线性维数约简方法。它的提出极大地拓展了关于维数约简的认识，人们开始关注数据集所包含的内蕴特征，通过内蕴特征的探讨来进一步研究维数约简问题。LLE 算法具有平移、旋转以及缩放不变性，可以广泛地应用于非线性数据的降维、聚类、可视化以及图像分割等领域。

2000 年，Roweis 和 Saul 在 *Science* 上发表他们的研究成果 *Nonlinear Dimensionality Reduction by Locally Linear Embedding*。随后，局部线性嵌入算法以其新颖的思想、简便的求解和从局部入手处理全局问题的深层次探讨，引起了国内外诸多领域科研人员的广泛兴趣。研究人员一方面对算法本身进行更深入的纵向挖掘，考察了算法的输入参数和最终嵌入结果之间的关系；进而构造方法以得到最优的输入参数，寻求算法对新增数据点的处理方法，以使得算法不仅适用于处理静态数据也能用于处理动态数据；根据各个领域自身的特点和实际需求对算法进行改进，以得到更能满足特定要求和适合专业特性的优良算法。另一方面学者们利用算法从局部考虑问题的思想，将谱图论、局部切空间理论中的思想引进来，发展出许多其他具有各自特点的非线性降维算法，如拉普拉斯特征映射算法、Hessian 特征映射算法、切空间排列算法等。局部线性嵌入算法作为一种优秀的数据处理工具，自然地被作为数据预处理手段应用于各种数据处理工作中，如数据压缩、模式识别、图像处理、可视化分析和机器视觉等。

但是我们也注意到对算法某些具体细节的证明在现有文献中还存在不足，并且对算法自身存在的一些问题的研究还分散在不同的文献中。

LLE 算法基于简单的几何直觉，即在高维空间中相邻的点在低维空间中依然保持相邻关系，通过降维的方法映射来获取高维空间的低维非线性流形结构。LLE 算法的主要思想是根据从高维空间映射到低维空间时邻近关系不变的假设，为高维空间中的每一个数据点寻找一个邻域，并给邻域中每个数据点赋一个权值，通过权值系数刻画出局部线性子空间的几何特征，用最优的权值描述在邻域范围内的相邻关系，然后再根据这些权值得到低维空间中最优的映射表示。

如图 3.6 所示，将 Swiss roll 模型进行采样，通过 LLE 算法将数据映射到二维空间中，与 PCA 比较可以看出，降维后的数据能很好地保持原有数据的邻域特性。

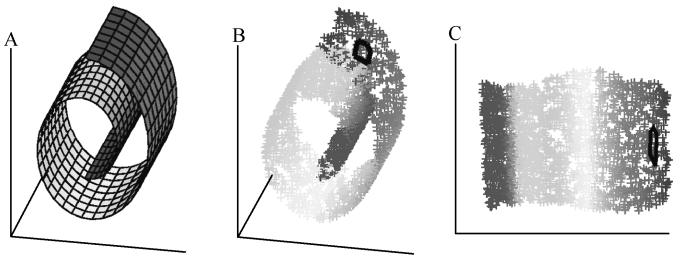


图 3.6　LLE 算法降维图示

通常对物体识别来讲，高维数据之间总是具有一定的相关性，有的是线性的，有的是非线性的，如何发现这些关系是降维的主要目的。

传统的用于物体识别的特征提取方法大都是线性的，这些算法可以很好地处理线性问

题。但是由于欧氏空间的线性特点，它们对高维空间里的复杂非线性数据很难进行有效的降维。而基于流形学习方法的 LLE 可以保持数据局部邻域之间的关系（即局部几何结构），并将高维数据映射到低维空间，其中可以很好地发现嵌入在高维空间中的低维子流形，最终获得较好的降维效果。

人脸识别是物体识别中的一个典型应用，LLE 用于人脸特征提取的效果如图 3.7 所示。图 3.7 中表示的是对单个人脸放在白噪声背景下通过左右上下平移所产生的数据进行二维降维的结果。由于原始数据本质上表征了图像中元素块的左右上下二维平移，因此它的特征结构也应该是二维的。从图 3.7 中的降维结果可以清晰地看出，由于原始数据流形的非线性特性，用 LLE 得出的降维结果与原始数据元素块的实际平移基本一致。而 PCA 则不能很好地保持数据点之间的结构，出现了不正确的位置投影，原数据的几何信息严重失真，即说明了 LLE 能很好地挖掘出数据的非线性结构。

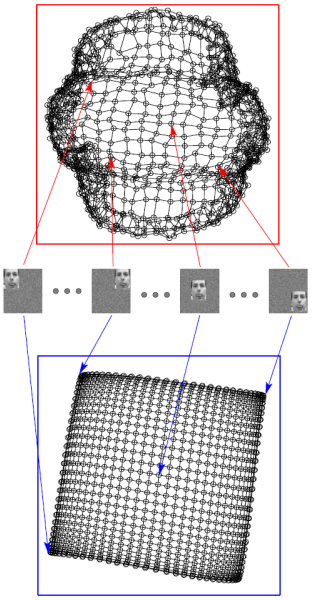


图 3.7　PCA（上）和 LLE（下）对人脸平移数据集的降维图示

LLE 算法如图 3.8 所示，该算法可以归结为三步：

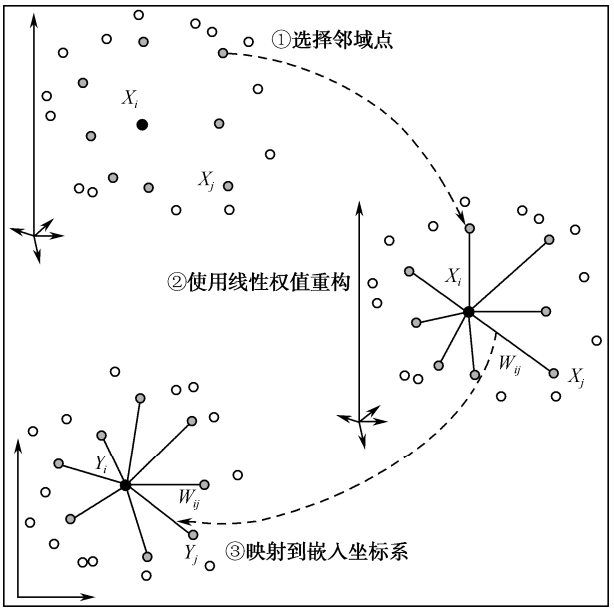


图 3.8　LLE 算法示意图

- 寻找每个样本点的 k 个近邻点；
- 由每个样本点的近邻点计算出该样本点的局部重建权值矩阵；
- 由该样本点的局部重建权值矩阵和其邻近点计算出该样本点的输出值。

给定含 N 个输入向量的矩阵 $X=(x_1,x_2,\cdots,x_N)$，其中 $x_i\in R^D$，通过 LLE 算法将得到输出值 $Y=(y_1,y_2,\cdots,y_N)$，$y_i\in R^d$，且 $d<<D$。算法步骤如下：

在高维空间中寻找每个样本的邻近点。使用度量空间的距离表示为

$$d_{ij}=\left[\sum_{k=1}^{D}\left|x_{ik}-x_{jk}\right|^p\right]^{\frac{1}{p}} \tag{3.63}$$

如果 $p=2$，表示欧氏距离；$p=1$，则表示 City-Block 距离。传统 LLE 算法使用欧氏距离。

计算出样本点的局部重建权值矩阵。通过定义的误差函数，并使误差函数值达到最小（每个采样点可以用它的近邻点线性表示），可得

$$\min\varepsilon(W)=\sum_{i=1}^{N}\left|x_i-\sum_{j=1}^{k}w_{ij}x_{ij}\right|^2 \tag{3.64}$$

其中，x_{ij} $(j=1,2,\cdots,k)$ 为 x_i 的 k 个邻近点；w_{ij} 是 x_i 与 x_{ij} 之间的权值，且满足条件

$$\sum_{j=1}^{k}w_{ij}=1 \tag{3.65}$$

把式（3.65）代入式（3.64），有

$$\begin{aligned}\min\varepsilon(W)&=\sum_{i=1}^{N}\left|x_i-\sum_{j=1}^{k}w_{ij}x_{ij}\right|^2=\sum_{i=1}^{N}\left|\sum_{j=1}^{k}w_{ij}(x_i-x_{ij})\right|^2=\sum_{i=1}^{N}\left|X_iw_i\right|^2\\&=\sum_{i=1}^{N}(X_iw_i)^T(X_iw_i)=\sum_{i=1}^{N}w_i^TX_i^TX_iw_i\\&=\sum_{i=1}^{N}w_i^TZ_iw_i\end{aligned} \tag{3.66}$$

其中，$X_i=(x_i-x_{i1},\cdots,x_i-x_{ik})$；$Z_i=X_i^TX_i$ 为第 i 个样本点的局部协方差矩阵；$w_i=[w_{i1},\cdots,w_{ik}]^T$ 为第 i 个样本点局部重建权值向量。求解式（3.66）是一个约束最小乘方问题，可利用 Lagrange 乘子，则有

$$\begin{aligned}&L(W)=\sum_{i=1}^{N}w_i^TZ_iw_i+\lambda(\sum_{j=1}^{k}w_{ij}-1)\Rightarrow\\&\frac{\partial L}{\partial w_i}=2Z_iw_i+\lambda\cdot I\Rightarrow Z_iw_i=c\cdot I\end{aligned} \tag{3.67}$$

c 通常取 1，I 为单位矩阵，可求得 w_i，如图 3.9 所示。

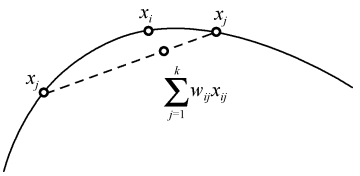


图 3.9　局部线性表示

将所有样本点映射嵌入到低维空间，且满足以下条件

$$\min \Phi(Y)=\sum_{i=1}^{N}\left|y_i-\sum_{j=1}^{k}w_{ij}y_{ij}\right|^2 \tag{3.68}$$

其中 $\Phi(Y)$ 是损失函数。固定样本点局部重建权值 w_{ij}、优化输出向量 Y 且满足两个条件：$\sum_{i=1}^{N}y_i=0$ 和 $\frac{1}{N}\sum_{i=1}^{N}y_iy_i^T=I$，这里的 w_{ij} $(i=1,2,\cdots,N)$ 可以存储在 $N\times N$ 的稀疏矩阵 W 中。当 x_j 是 x_i 的邻近点时，$W_{i,j}=w_{ij}$；否则，$W_{i,j}=0$。用 W_i 表示 W 矩阵的第 i 列，I_i 表示 $N\times N$ 单位矩阵的第 i 列，Y 表示输出向量，即 $Y=(y_1,\cdots,y_N)$；则式（3.68）可写为

$$\begin{aligned}\min \Phi(Y)&=\sum_{i=1}^{N}\left|y_i-\sum_{j=1}^{k}w_{ij}y_{ij}\right|^2=\sum_{i=1}^{N}\left|YI_i-YW_i\right|^2\\&=\sum_{i=1}^{N}\left|Y(I_i-W_i)\right|^2=\left|Y(I-W)\right|^2=\left|Y^T(I-W)\right|^2\\&=Y(I-W)(I-W)^TY^T=YMY^T\end{aligned} \tag{3.69}$$

其中 $M=(I-W)(I-W)^T$。

再利用 Lagrange 乘子，则有

$$\begin{aligned}L(Y)=YMY^T+(YY^T-NI)&\Rightarrow\frac{\partial L}{\partial Y}=2MY^T+2\lambda Y^T\\&\Rightarrow MY^T=\lambda Y^T\end{aligned} \tag{3.70}$$

要使损失函数值最小，则取 Y 为 M 的最小 d 个非零特征值所对应的特征向量。通常最小特征值几乎为零，因此取第 2～$(d+1)$ 间的特征值所对应的特征向量作为输出结果 Y。

2．局部保持投影算法

局部保持投影（Locality Preserving Projections，LPP）算法是 Xiaofei He 等人于 2002 年提出来的，它的理论基础是对拉普拉斯特征映射的线性近似，它既具有线性算法所不具备的保持高维数据流形的局部不变性，还具有非线性方法所不具备的获得新样本点的低维投影、计算简单、处理速度快等线性降维方法所具有的特点。

假设有 m 个样本，每个样本的维数是 n，存在一个数据集 $X=\{x_1,x_2,\cdots,x_m\}$，其中 $x_i\in R^n(i=1,\cdots,m)$，找到一个矩阵 A，使集合 X 中的每个数据投影到子空间 R^l 中，其中 $Y=\{y_1,y_2,\cdots,y_m\}$，$y_i\in R^l(i=1,\cdots,m)$，每一个 x_i 对应到子空间的 y_i，即 $y_i=A^Tx_i$，局部保留投影算法步骤如下。

（1）构造邻近图 G，把数据集 X 中的每个数据 x_i 当作一个无向图的顶点，如果顶点 x_i 与顶点 x_j 邻近，则把它们用一条边连接上，确定顶点 x_i 与顶点 x_j 邻近的方法有如下两种。

第一种方法是 ε 近邻法，其中 $\varepsilon\in R$。

如果顶点 x_i 和 x_j 满足 $\left\|x_i-x_j\right\|^2<\varepsilon$，说明两顶点是相邻近的，则为它们创建一条边。

第二种方法是 k 近邻法，其中 $k\in N$。

如果顶点 x_i 是顶点 x_j 的 k 近邻，同时顶点 x_j 也是顶点 x_i 的 k 近邻，则说明这两顶点是近邻点，可以为它们创建一条边。

（2）确定每条边的权值。

在上一步中，为每对相邻近的顶点建立了边，没有边相连的两个顶点说明它们不是邻近点，便没有相关性，有边相连的两个顶点可以根据以下两种方法来确定它们的相关性。

第一种方法是热核加权方法，即这条边的权值的大小与它们的距离大小呈指数衰减。

$$W_{ij} = \mathrm{e}^{-\frac{\|x_i - x_j\|^2}{t}} \tag{3.71}$$

第二种方法是最简单的方法，也就是有边相连的权值为 1，无边相连的权值为 0。

因此，m 个样本的数据集 X 所对应的权值矩阵 W 是一个 $m \times m$ 的对称矩阵，而且还是稀疏矩阵。

（3）进行特征映射。

这一步的目的是找出最优的线性嵌入并进行投影，这个问题的简单描述为：假设有一个数据集 $\{x_1, x_2, \cdots, x_m\}$，每个样本看作是一个顶点，创建一个无向图 $G = (V, E)$，每对相邻的顶点用边相连，现将无向图 G 投影到它的子空间中，使得有边相连的两个顶点间的距离越小越好。设它投影后的数据为 $y = (y_1, y_2, \cdots, y_m)^T$，要找出符合条件的 y，使下面的准则函数的值最小，即

$$\sum_{ij} (y_i - y_j)^2 W_{ij} \tag{3.72}$$

从准则函数可以知道求它的最小值的目的是使 y_i 和 y_j 尽可能地相近，即它们能够很好地保持原始数据的局部特征。也就是当 x_i 和 x_j 接近，它们的权值 W_{ij} 也就变大，也就是希望 y_i 与 y_j 也是相邻近的。

从局部保持投影算法的描述可知，存在一个向量 a，使 $y_i = a^T x_i$，其中 y_i 表示样本 x_i 在 a 维度上的映射，接下来结合权值矩阵 W_{ij} 求出投影矩阵。

$$\begin{aligned}
&\frac{1}{2}\sum_{ij} (y_i - y_j)^2 W_{ij} \\
&= \frac{1}{2}\sum_{ij} (a^T x_i - a^T x_j)^2 W_{ij} \\
&= \sum_i a^T x_i D_{ii} x_i^T a - \sum_{ij} a^T x_i W_{ii} x_i^T a \\
&= a^T X (D - W) X^T a = a^T XLXa
\end{aligned} \tag{3.73}$$

其中，X 是 $n \times m$ 的矩阵，且 $X = [x_1, x_2, \cdots, x_m]$；$D$ 是对角矩阵，且满足 $D_{ii} = \sum_j W_{ji}$，即 D 的对角线上的元素是权值矩阵 W 的每行（或每列）的和；$L = D - W$，且 L 是 Laplacian 对称半正定矩阵。

以上分析仅对每个数据的相互关系进行了讨论，还要进一步对变换后的坐标进行设置，从 $D_{ii} = \sum_j W_{ji}$ 中可以看出，D_{ii} 表示与第 i 个顶点有边相连的顶点数的个数，一定程度上代表着该点的重要性，可以对 y 进行如下限制

$$y^T Dy = 1 \Rightarrow a^T XDX^T a = 1 \tag{3.74}$$

这样限制 y 的目的是为了使重要的点，即连接其他顶点数越多的点在投影完成后该顶点

的位置越接近原点，也就是在数据最密集的区域设置原点。所以求准则函数极小值问题也就转化成了式（3.75）

$$a^* = \underset{a^T XDX^T a=1}{\arg\min}\, a^T XLX^T a \tag{3.75}$$

用拉格朗日乘子法求解上式，可令

$$L = a^T XLX^T a - \lambda a^T XDX^T a \tag{3.76}$$

为了使它对 a 的梯度为 0，可使它对 a 求偏导后为 0，即

$$\frac{\partial}{\partial a}(a^T XLX^T a - \lambda a^T XDX^T a) = 0 \tag{3.77}$$

通过简单的运算可得

$$XLX^T a = \lambda XDX^T a \tag{3.78}$$

于是，求准则函数的最小值问题又转化成了求 $XLX^T a = \lambda XDX^T a$ 的特征值与特征向量问题，要使 $a^T XLX^T a$ 最小，可以让最小的 l 个非零特征值所对应的特征向量构成 a。

假设求解出来的非零特征值是 $\lambda_1 < \lambda_2 < \cdots < \lambda_l$，对应的特征向量为 $a_1, a_2, \cdots, a_l$，投映过程为

$$x_i \rightarrow y_i = A^T x_i \tag{3.79}$$

其中 A 是 $n \times l$ 的变换矩阵，且 $A = (a_1, a_2, \cdots, a_l)$； y_i 是 l 维向量。

3.7　基于代数多重网格方法的特征研究

3.7.1　使用代数多重网格方法进行特征优化

根据前面章节的描述，粗网格可以提取更多的细节特征，这里通过特征脸的试验来验证这一方法的有效性。使用脸部数据库中的 10 幅图片，如图 3.10 所示。使用传统的 PCA 方法来进行脸部识别，距离使用欧式距离。一组数据使用原始脸部图片，另一组数据使用第一层粗网格数据重建后的脸部图片作为输入，通过对比分析来证明该特征的有效性，结果如图 3.11 和表 3.2 所示。图 3.11 得到的是输入图像与图片序列中其他图像的距离，而表 3.2 分别是不同输入图像（图 1～图 10）与样本库中图片的距离的标准差。

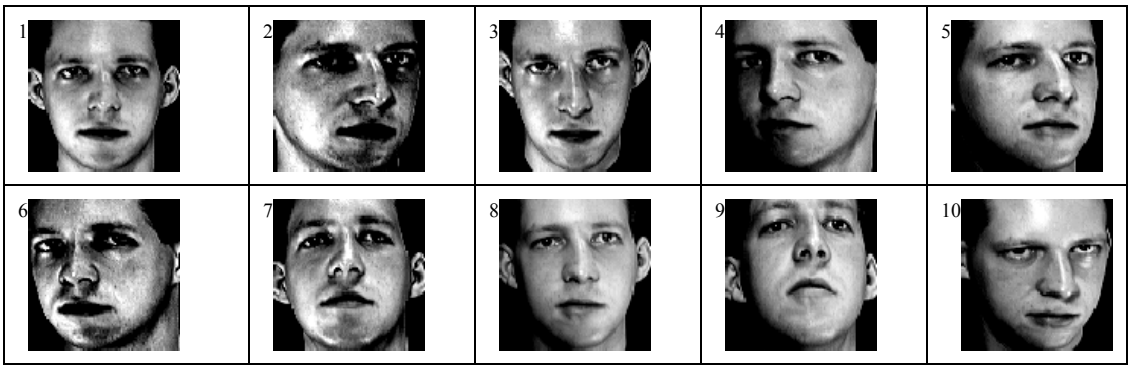


图 3.10　10 幅脸部图像

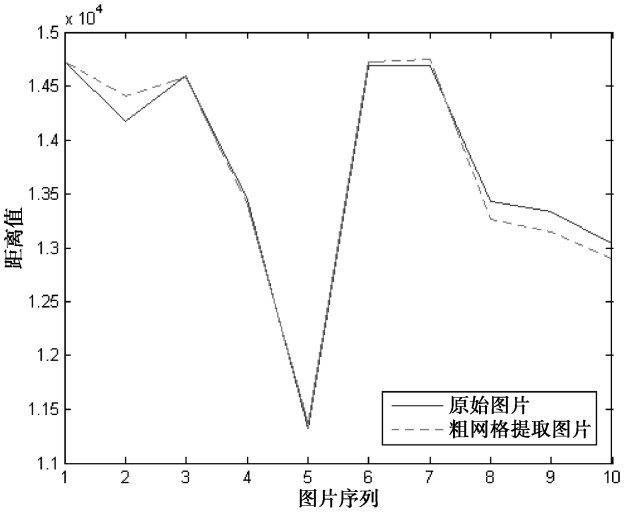


图 3.11　原始图片和粗网格提取图片序列的距离的对比图

通过对结果分析，总共分析了 10 幅图片，其中 9 幅图片（图 6 除外）通过粗网格提取之后重建的图片都能有效地提高样本与库中样本距离的标准差，增加对比度，可以更有效地进行物体识别。

表 3.2　原始图片和粗网格提取图片的距离的标准差的对比数据

	原始图片	粗网格图片
图 1	1.1246×10^3	1.2224×10^3
图 2	1.7142×10^3	1.7239×10^3
图 3	1.2218×10^3	1.2654×10^3
图 4	1.2797×10^3	1.3007×10^3
图 5	1.0726×10^3	1.1117×10^3
图 6	1.2490×10^3	1.1969×10^3
图 7	1.0380×10^3	1.0650×10^3
图 8	902.27	963.97
图 9	1.1900×10^3	1.2152×10^3
图 10	1.0858×10^3	1.1042×10^3

3.7.2　使用代数多重网格方法进行特征选择

特征选择是图像识别和图像检索的一个关键因素。从大的特征库中选择合适的特征来进行图像识别和图像检索，可以提高相关应用的准确性。使用的图像数据库来自于 Caltech-101 库(http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html#Download)。通过实验发现，当数据集包含 80 张图像时，运行时间超过 3 天（运行环境：intel core i7，内存 8G），而库中总共有 8000 多幅图像，如果实验完全进行将需要接近一年的时间。因此我们对

实验进行了设计，随机选择多个分类，然后从中随机选择两个类别进行对比实验。实验中使用原始图像和使用代数多重网格进行粗化后的图像分别进行物体识别。

1．实验原理

代数多重网格在每次进行粗化的过程中，会将关联关系大的点保留，将关联不大的点剔除。原来是直接对原始图像提取 SIFT 特征向量，而这里使用的方法是对原始图像中提取的 SIFT 特征点进行筛选，如果该点在粗网格上则保留，否则就去掉该 SIFT 特征向量。得到图像的 SIFT 特征向量后，将所有训练图像的特征向量聚类，这里将图像看成是由大量视觉单词组成的，通过视觉单词表示图像，可以将图像文本化，进而提高图像检索的效率。形成视觉单词后，针对每一幅图像统计属于每个视觉单词的特征个数，形成这幅图像的视觉单词直方图。为了减少数据量，形成视觉单词直方图后，我们只选择分布频率较高的视觉单词来表示该图像，分布频率小的视觉单词对图像识别的影响不大，所以我们在识别之前，剔除了这部分视觉单词，最终形成表示图像的特征识别向量。在特征识别向量的基础上进行图像识别，统计测试图像和每个训练图像相同的特征向量个数，相同个数越大，说明两幅图像相似度越高。找到相同个数最大的训练图像的类别，即为该测试图像的类别。

2．实验过程

（1）将图像进行规格化，将所有的图像处理成 512×512 的大小。

（2）将图像灰度化，提取每幅图像的 SIFT 特征向量。

（3）将图像分成两类，一部分作为训练图像，另一部分作为测试图像。

（4）训练图像视觉单词聚类。将两个类别的所有训练图像的 SIFT 特征向量使用 K-Means 算法进行聚类，将所有图像的 SIFT 特征向量聚成 100 个类，也就是形成 100 个视觉单词。

（5）得到聚类中心后，对于每一幅图像，统计属于每个视觉单词的特征个数，形成视觉单词直方图分布。

（6）对于每一幅图像挑选直方图分布频率最高的 10 个单词来代表这个图像，构成这个图像的特征识别向量。所有训练图像的特征识别向量构成特征识别库。

（7）将每幅测试图像的特征识别向量与特征识别库中的特征识别向量进行比较，统计出特征识别向量相同的向量的个数，找出相同个数最大值所对应的图像所属的类别，即为该测试图像所属的类别，完成物体识别。

将原始图像和代数多重网格方法提取的粗网格图像进行同样的处理，最后进行识别结果对比。

实验中随机抽取 6 组数据进行图像识别实验，第一组 accordion（55 幅）和 anchor 图像（42 幅）。第二组 barrel（47 幅）和 bass（56 幅）图像。第三组 brontosaurus（43 幅）和 camera（54 幅）图像。第四组 connon（43 幅）和 ceiling_fan（47 幅）图像。第五组 beaver（48 幅）和 binocular（35 幅）图像，第六组 cellphone（59 幅）和 cougar_body（47 幅）图像。

实验过程如图 3.12 所示。

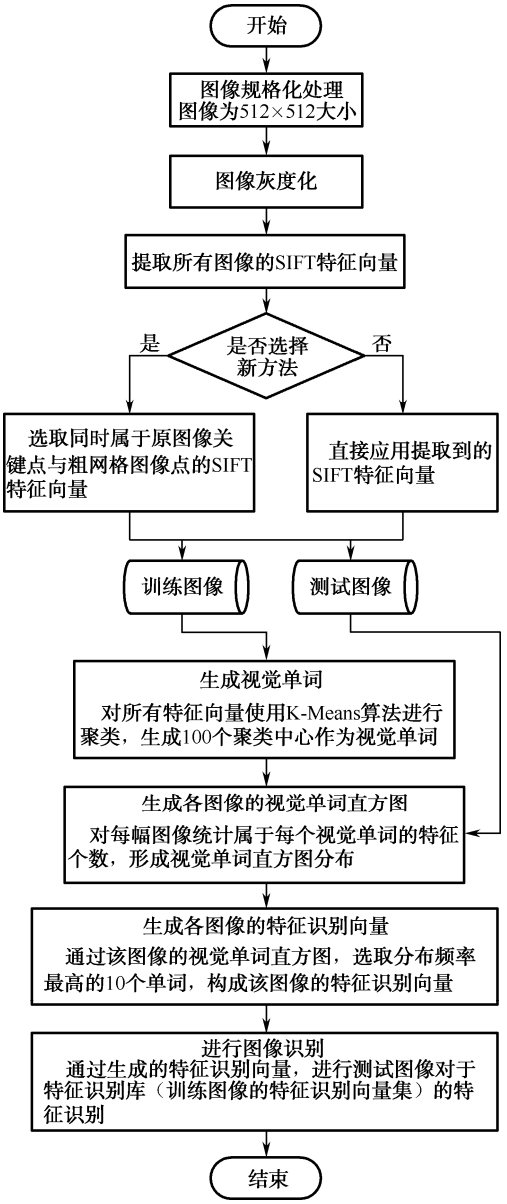


图3.12　实验流程图

3．实验结果

实验结果如表 3.3 和图 3.13 所示。图 3.13 对比了每组使用原图像和粗网格图像的物体识别率，从图 3.13 中可以看出，每组数据中，使用粗网格图像识别的准确率要高于原图像进行图像识别的准确率。由此可见，使用粗网格提取的数据能较好地将图像中能够反映物体特征的部分保留下来，更好地进行物体识别。

表 3.3　每组图像识别率对比表

序　　号	第一类图像数据集	第二类图像数据集	原图像识别率	粗网格图像识别率
1	accordion	anchor	70%	80%
2	barrel	bass	58.33%	72.22%
3	brontosaurus	camera	70.83%	80%
4	connon	ceiling_fan	76.47%	76.67%
5	beaver	binocular	61.54%	76.09%
6	cellphone	cougar_body	73.91%	78.26%

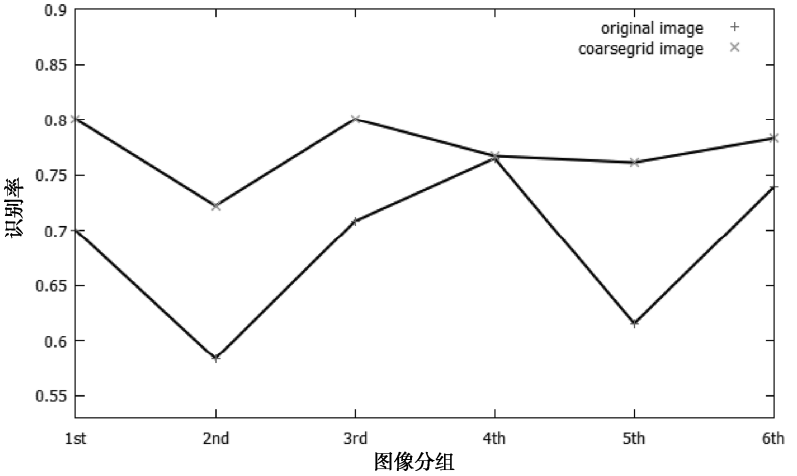


图3.13　每组图像的识别率对比图

3.8　本章小结

本章主要将代数多重网格运用到特征表达中，首先介绍了一些图像的基本特征，如颜色、纹理、形状和空间关系等特征，然后介绍了在选定使用图像的哪种特征后，怎样对特征进行描述，主要介绍了 SIFT 特征描述子、LBP 纹理特征描述符和基于场景的视觉描述符。这里着重通过 LBP 特征描述符介绍了怎样构造灰度、旋转不变性的特征描述符。在这些内容的基础上，介绍了特征的选择和特征的优化，其中详细介绍了特征的选择依据。最后，介绍了使用代数多重网格方法进行特征表达的方法，因为在该方法的每一层粗化过程中，都会去掉图像与其他图像区分不大的特征而保留图像最本质的特征，所以使用粗网格来对图像进行特征表达。同时我们做了相关实验，实验结果表明，使用粗网格提取重建后的图像确实保留了图像最能区别于其他图像的特征。

第 4 章　代数多重网格方法在图像融合中的应用

代数多重网格的多层粗网格能将重要的特征点保留，因此能较好地对图像进行特征相关的处理，如清晰图评价、特征提取、双目重建等相关处理，本章主要针对代数多重网格方法在图像融合中的应用进行分析。

4.1 多聚焦图像融合概述

图像融合通常的定义为：将两幅或两幅以上图像，采取一种特定的融合算法，合成一幅新图像。其主要思想是把不同图像，如不同波长范围的图像，具有不同成像机理的多模态图像，或者对同一场景不同时间的、不同聚焦的图像等，采用一种特定的方法，将多幅图像的信息融合到一幅新的结果图像中[57]。

图像融合的研究对象为两幅或两幅以上的不同图像，这些图像间含有彼此不同的信息，这些图像可以为同一物体在不同时间，或者是同一物体的不同波段图像，或来自不同模态的图像。图像融合研究如何将这些图像中的信息融合成一幅新的合成图像。其目的是为了获取更多的关于目标信息的图像，便于进一步地处理[58]。

作为信息融合的一个重要分支，图像融合利用多幅图像间的冗余数据处理，采取一定的方法来提高图像的可靠性，且能够将多幅图像之间的互补信息融合，使结果图像的信息量得到提高，可以融合各类数据优点[58]，提高图像解译能力，克服单一图像信息的局限性[59]。不管是在图像理解和计算机视觉等领域的学术研究上，还是在现实生活应用中，图像融合技术已经成为一项重要而实用的图像处理技术。图像融合应用十分广泛，如遥感信息系统、医学诊断和地质结构分析等。

在图 4.1 中，待融合的图像间含有互补信息以及冗余信息，那么将图像融合可获得对场景更加清晰的描述。图像融合研究的对象主要有两种：一是不同种类传感器得到关于对同一物体描述的图像，二是同一类型传感器不同时间或不同参数时得到的多个图像。每种图像都有各自独特的信息优势，图像融合可进行优势互补，使融合图像对目标描述更加全面，有利

于图像的深层处理。

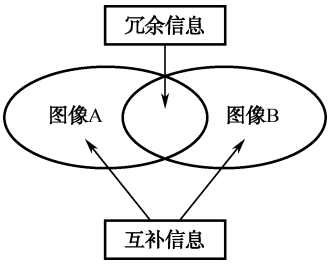
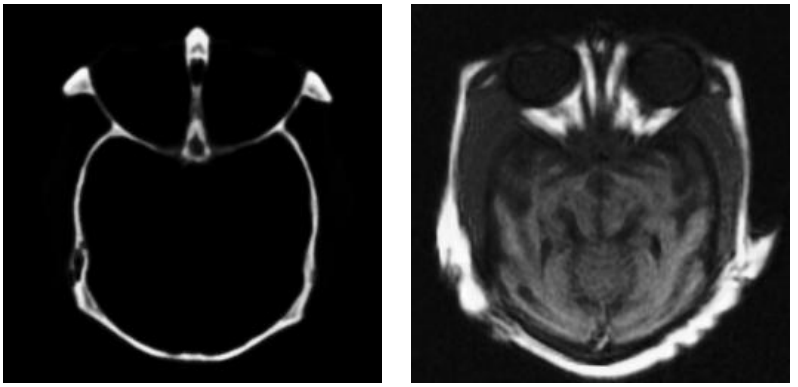


图 4.1　融合图像之间的信息划分

图 4.2 为不同模态的医学图像，图 4.2（a）为头部 CT 图像，图 4.2（b）为头部的 MRI 图像，将两种模态图像融合在一起，可以获取头部组织的全面描述，诊断更加可靠。图 4.3 为不同时间的同一场景图像，图 4.3（a）为光照充足时获取的图像，图 4.3（b）为烟雾情况获取的图像，这种图像数据融合对指导救援和人员疏导有重要作用。图 4.4 为飞机导航图像，图 4.4（a）和 4.4（b）融合结果对飞机导航有很大的意义。从这三个简单的示例可以看出图像融合技术的重要实际应用。

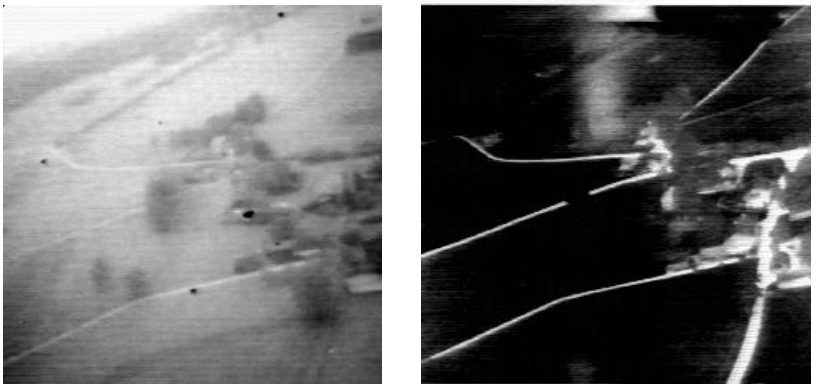


（a）CT 图像　　（b）MRI 图像

图 4.2　不同模态的医学图像

（a）光照充足的图像　　（b）有烟雾笼罩的图像

图 4.3　不同时间的同一场景图像



（a）低光传感器图像　　（b）红外热成像图像

图 4.4　导航图像

图像融合算法的预处理包含图像去噪和图像配准，如图 4.5 所示。图像去噪是一个很普遍的问题，很多图像处理过程都会将图像去噪作为第一个步骤。图像配准是针对同一对象在不同条件下获取的多幅图像，寻找图像之间的对应关系。图像配准是一类典型问题，是一种极具应用价值的技术，在很多图像处理领域中都会用到，如双目视觉、全景拼接、医学处理等。图像融合会在一定程度上依赖图像配准。本章主要是考虑图像融合，而且使用的图像是经过配准处理过的，故不对图像去噪和图像配准算法详细展开。

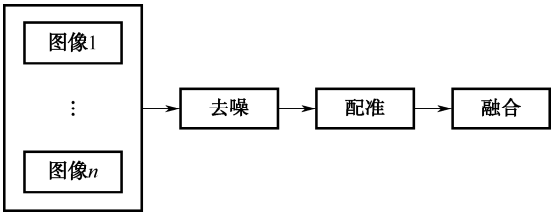


图 4.5　图像融合的一般流程

4.2　多聚焦图像融合基础和质量评价

4.2.1　多聚焦图像产生的原因

图 4.6 为光学镜头成像原理。在图 4.6 中，U 代表物距，V 代表聚焦平面的相距，S 代表非聚焦平面的相距。根据物理知识可知：如果物体在聚焦平面成像，那么图像就清晰，否则模糊[60]。在图 4.6 中，因光学镜头与非聚焦平面是平行的，因此有式（4.1）

$$x\frac{D}{2R}=\frac{V}{|S-V|} \tag{4.1}$$

由式（4.1）可得

$$R=|S-V|\frac{D}{2V} \tag{4.2}$$

将凸透镜成像方程式（4.3）

$$\frac{1}{f}=\frac{1}{U}+\frac{1}{V} \tag{4.3}$$

代入式（4.2）有

$$R=\frac{1}{2}DS\left(\frac{1}{f}-\frac{1}{U}-\frac{1}{S}\right) \tag{4.4}$$

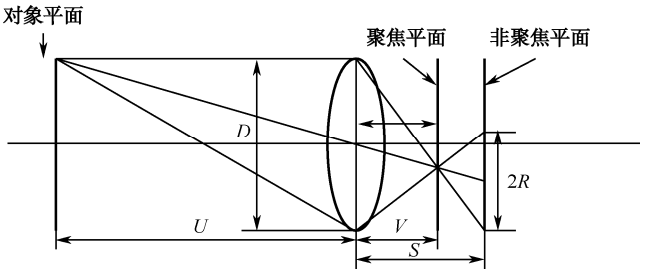


图 4.6　光学镜头成像原理

那么此时得到的 R 就表示产生的图像模糊圈大小。设可接受的最大模糊圈为 R，那么必须改变物体的物距，那么根据式（4.3）可以有

$$\frac{1}{f}=\frac{1}{U'}+\frac{1}{V'} \tag{4.5}$$

可得式（4.6）

$$U'=\frac{V'f}{V'-f} \tag{4.6}$$

其中，V' 等于前述的 S。由式（4.2）可知 $|S-V|=\frac{2RV}{D}$，即为

$$|V'-V|=\frac{2RV}{D} \tag{4.7}$$

当 $V'>V$，则有

$$V'=V+\frac{2RV}{D} \tag{4.8}$$

否则

$$V'=V-\frac{2RV}{D} \tag{4.9}$$

将式（4.5）和式（4.9）分别代入式（4.6），可得

$$U_{\max}=\frac{Uf\left(1-2\frac{R}{D}\right)}{f-2\frac{R}{D}} \tag{4.10}$$

$$U_{\min}=\frac{Uf\left(1+2\frac{R}{D}\right)}{f+2\frac{R}{D}} \tag{4.11}$$

$$\mathrm{DOF} = U_{\max} - U_{\min} \tag{4.12}$$

$U_{\max}$、$U_{\min}$ 为 R 在特定情况下物距的范围，DOF 表示光学镜头的景深，所以光学镜头产生的图像清晰与否与其景深有密切的关系。同一场景的物体由于在不同景深的情况下就会使一部分图像模糊、一部分图像清晰，出现不同的清晰度。

4.2.2 多聚焦图像融合的层次

图像融合技术一般划分为三个不同层次[62]：像素级融合、特征级融合和决策级融合。图像融合系统的层次分类如图 4.7 所示。

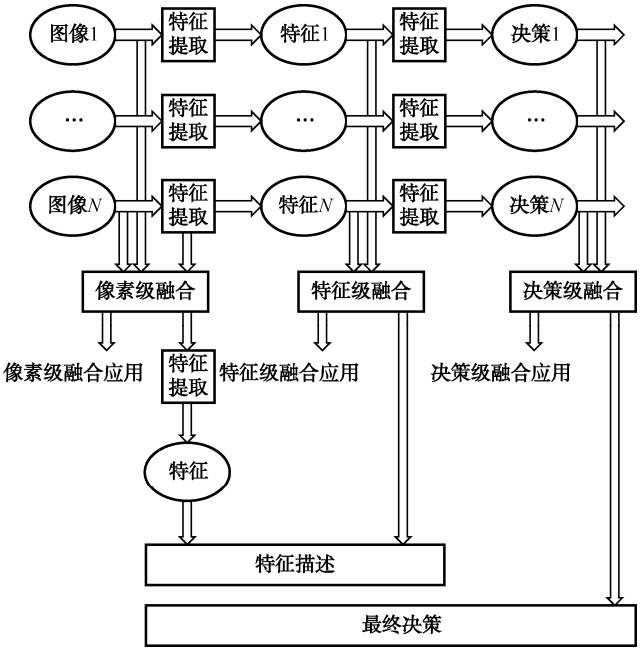


图 4.7　图像融合系统的层次分类

最低级融合层次是像素级融合，需要将图像融合的多幅图像预处理之后，比如去噪、配准等，在图像的像素级别上进行的图像信息融合。像素级融合处理的粒度是图像的像素点，没有达到较高图像理解层次。这个层次的融合算法中最具有代表性的是均值融合：将多幅源图像的像素灰度值累加求平均，就可以得到融合图像。像素级融合是对各个原始图像源进行处理，因此像素级融合准确性较高，细节信息完整、丰富、可靠，而且引入的人为信息较少，信息融合更加精确，对目标对象进一步处理更加有利。因此，像素级融合为高层次级别融合提供了最基础的融合结果。但是由于像素级别的运算量很大，所以像素级算法对设备和算法的要求很高，这样才能达到应用要求。

相比像素级融合而言，特征级融合属于较高层次，是中间层次的信息融合。特征级融合是在像素级融合的基础上，提取融合图像的特征，如线形、边缘、纹理、形状、轮廓和区域等，采取的方法有很多，如参数模板、矩方法和统计分析等，然后将不同的特征信息进行融合。特征级融合可从目标状态数据融合和目标特性融合两个方面进行研究，主要有聚类法、

贝叶斯估计法和神经网络算法等。

决策级融合层次较高，在进行特征提取之后，可以根据一系列目标对象的联合特征，按照一定的准则给出所需要的决策，决策级融合能满足一定的主观需求。决策级图像融合涉及的技术领域有图形分类和图像识别等。经过决策级融合后，可以直接将结果提供给决策者，并为决策者提供判断依据。决策级融合实时性好，并且具有一定的容错能力。其缺点是预处理代价较高，损失源图像的信息较多。决策级融合方法主要以认知模型为基础，而且需要一定的判断准则，所以决策级融合需要专家决策系统，这样才能进行分析、推理、识别和判决。

4.2.3　多聚焦图像融合的算法

在 4.2.2 节中，图像融合可以划分为像素级、特征级和决策级三个层次。其中，像素级融合精度较高，提供的目标信息较丰富。像素级融合方法主要分为空域和频域两类[62]。

（1）在空域中，主要是基于图像清晰部分的提取，有梯度差分法、分块法等，其优点是速度快、方法简单，不过融合精确度相对较低，边缘粗糙。

（2）在频域中，具有代表性的是多分辨率方法，其中有拉普拉斯金字塔算法、小波变换法等，多分辨率融合精度比较高，对位置信息的把握较好，不过算法比较复杂，处理速度比较慢[63]。

4.2.4　基于空域的多聚焦图像融合算法

在空域中，可以将图像看成一个二维函数 $f(x,y)$，对其进行处理不需要对图像进行任何变换，而且算法思想较简单，时间复杂度低。其算法可以分成三类：基于像素点的融合算法；基于区域的融合算法；基于分块的融合算法。

1．基于像素点的融合算法

加权平均法是最简单的一种基于像素点的图像融合方法，它是将源图像中每个像素值加权取平均，即

$$\mathrm{Fusion}(x,y)=0.5f_1(x,y)+0.5f_2(x,y) \tag{4.13}$$

其中，$\mathrm{Fusion}(x,y)$ 为融合的结果；$f_1(x,y)$，$f_2(x,y)$ 为不同的源图像；$x=1,2,\cdots,X$，$y=1,2,\cdots,Y$，$X\times Y$ 是源图像的尺寸大小。

灰度值取最小和灰度值取最大也是属于基于像素点的融合方法，其思想比较简单，分别选取源图像中像素值的最小值或最大值作为融合图像的像素值，即

$$\begin{aligned}\mathrm{Fusion}(x,y)&=\max[f_1(x,y,f_2(x,y)]\\ \mathrm{Fusion}(x,y)&=\min[f_1(x,y,f_2(x,y)]\end{aligned} \tag{4.14}$$

基于像素点的融合算法的缺点：由于单个像素不能体现该像素点所在区域特征，所以融合图像对比度降低，图像边缘信息会有一定程度的丢失，而且会产生重影。

2．基于区域的融合算法

基于区域的融合算法的加权系数是根据区域的特征计算的。根据区域的特征因子去设定该点的加权系数。因为此算法是建立在图像区域内像素的相关性之上的，减少了偶然性，这样融合结果较基于像素点的融合算法清晰。

主成分分析融合法是较常见的基于区域的图像融合算法。PCA 融合算法能保留变量的主要信息，考虑区域的主要特征。通过求相关矩阵的特征值和特征向量，再由这些特征值和特征向量计算各个源图像区域的权重系数，这种方法适用于多光谱图像。

3．基于分块的融合算法

在一定程度上分块融合可以认为是多聚焦图像融合的特有算法。基于分块的融合算法是将每个源图像分为若干个子图像，再在每个子图像中计算图像的特征测量尺度，根据这个测量尺度选取合适的子块放在相应位置上，大致过程如图 4.8 所示。

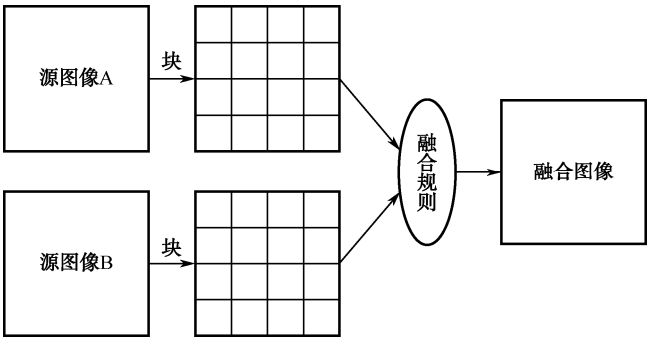


图 4.8　分块融合算法框架图

目前常用的特征因子是空间频率（Spatial Frequency），反映了块的活性程度，空间频率定义见式（4.15）。

$$\mathrm{SF}=\sqrt{\mathrm{RF}^2+\mathrm{CF}^2} \tag{4.15}$$

其中，RF、CF 分别是行和列的空间频率，定义见式（4.16）和式（4.17）。

$$\mathrm{RF}=\sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=2}^{N}[F(i,j)-F(i,j-1)]^2} \tag{4.16}$$

$$\mathrm{CF}=\sqrt{\frac{1}{MN}\sum_{i=2}^{M}\sum_{j=1}^{N}[F(i,j)-F(i-1,j)]^2} \tag{4.17}$$

其中，$M\times N$ 为图像分块的尺寸大小。

4.2.5　基于频域的多聚焦图像融合算法

基于频域的图像融合算法一直是多聚焦融合算法的研究热点[64]，其一般流程如图 4.9 所示。常见的算法主要有金字塔图像融合算法和小波变换图像融合算法。

$$\left.\begin{matrix} f_1(x,y) \\ f_2(x,y) \end{matrix}\right\} \rightarrow \left\{\begin{matrix} F_1(x,y) \\ F_2(x,y) \end{matrix}\right\} \rightarrow F(x,y) \rightarrow f(x,y)$$

源图像 → 变换到频域 → 频域融合 → 逆变换得到最终结果

图 4.9　频域融合一般流程

1．金字塔图像融合算法

金字塔图像融合算法的主要思想是将每个用于图像融合的源图像都用金字塔模型表示，

可得到源图像的每一层数据，在金字塔模型的每一层数据上按照一定的融合规则进行融合，可以得到融合图像的金字塔模型的每层数据，将这些数据经过金字塔模型逆变换，可得到融合结果。相比基于空间域的融合算法而言，融合效果得到一定程度的改善。金字塔模型融合算法优点是金字塔模型分解的每一层数据都具有冗余性，这样可以很好地在层与层之间找到联系，方便层与层之间的处理。但是金字塔模型融合算法高频损失严重，甚至会在融合结果中出现模糊现象。

高斯金字塔分解定义为

$$\begin{cases} G^0 = I \\ G^{k+1} = [w * G^k]_{\downarrow 2}, k = 0,1,\cdots,N-1 \end{cases} \tag{4.18}$$

其中 w 为一个低通滤波器窗口函数。

对比度金字塔模型定义如下：

$$\begin{cases} R^N = G^N \\ R^K = \dfrac{G^K}{4w*[G^{K+1}]_{\uparrow 2}} - I, K = 0,\cdots,N-1 \end{cases} \tag{4.19}$$

融合图像中每层图像的融合依据公式（4.20）进行。

$$R_F^K(i,j) = \begin{cases} R_A^K(i,j) & \text{当} |R_A^K(i,j)| > |R_B^K(i,j)| \\ R_B^K(i,j) & \text{其他} \end{cases} \tag{4.20}$$

从式（4.20）可以得到融合结果的每层数据，然后可以经过高斯金字塔逆变换得到最终融合结果。

2．小波变换图像融合算法

小波的理论研究已经相当完善，而且小波有相对较好的特性，因此在图像融合算法中经常被运用到[65]。基于小波变换的图像融合算法主要流程是首先运用离散小波变换，把用于图像融合的各个源图像分解成子带图像序列，这些图像序列具有不同频率、不同分辨率等性质。这样每个源图像做小波变换后，得到融合结果的不同频率子带图像；然后，在每一个子带图像中，根据融合规则，得到融合图像的各个子带图像；最后，再经过逆小波变换，就可以得到融合结果。基于小波变换的图像融合算法流程框架如图 4.10 所示。

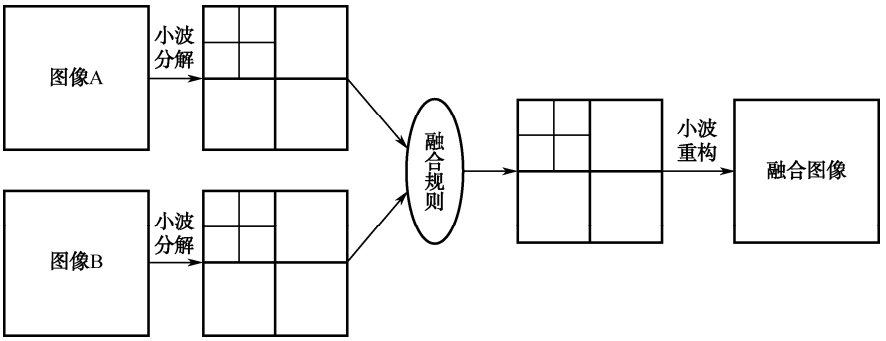


图 4.10　基于小波变换的图像融合算法流程框架

4.2.6 多聚焦图像融合质量评价

在采取特定图像融合算法获取图像融合结果后，就须采取一定的标准判别图像质量的好坏。而在实际的研究中，对图像融合评价准则的研究也是一项十分有意义的工作。目前，对融合结果的评价没有一个很好的主观或者客观的评价系统，因此在实际运用中，为了给出一定的客观定量评价，对融合结果的评价所采用的标准往往并不全面[66]。

在实际应用中，衡量融合图像的效果应遵循以下原则：

（1）尽可能包含源图像中的有用信息；

（2）在处理过程中不应引入源图像以外的人为信息。

通常情况下主要从两方面评判：一是从视觉效果上作出主观判断；二是根据客观评价指标给出客观定量评价。

4.2.7 主观评价

主观评价是通过人眼来完成的，不仅会受到客观因素的影响（如光照、图像的成像环境），还会受到评价人员的主观因素的影响（如评判标准的侧重点、个人喜好不同）。但是通常情况下，通过人眼判断是最有效、最便捷的，因为人眼对图像的敏感性和整体的把握性较强，如图像是否存在重影、块效应，是否包含清晰的边缘信息，纹理和色彩与源图像相比是否有缺失等。因此在实际运用中，主观评价往往可以作为图像的定性说明措施，弥补客观评价的不足。对图像结果进行主观评价的人员通常由非专业人员和专业人员组成，因为每个人对图像的评价不尽相同，一般是取均值作为主观评价的结果。

4.2.8 客观评价

图像融合结果需要考虑到图像的理想融合结果和源图像的有效信息，实际中很难寻找到一个完美的评价标准体系。通常，客观评价方法主要有以下几种[67]：

（1）越大越优型：主要参数有标准差、熵值、平均梯度、相关系数、联合熵、互信息、信噪比、峰值信噪比等。

（2）越小越好型：主要参数有光谱扭曲度、均方误差、交叉熵等。

（3）适当型：主要参数有均值等。

在以上参数中，不需要标准参考图像的评价参数有标准差、熵值、平均梯度、均值，除此之外还有QAB/F [67]评价参数等。

熵值衡量的是图像信息量，根据图像中具有不同灰度值像素的概率分布计算，即

$$E = -\sum_{i=0}^{L-1} p_i \log_2 p_i \tag{4.21}$$

其中，p_i 为灰度值为 i 的像素个数在图像中的概率；L 为图像灰度级数。

相关系数是评价融合图像和标准参考图像的相似程度的重要参数，可表示为

$$\rho=\frac{\left|\sum_{i=1}^{m}\sum_{j=1}^{n}[R(i,j)-\mu_R][F(i,j)-\mu_F]\right|}{\sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}[R(i,j)-\mu_R]^2\right)\left(\sum_{i=1}^{m}\sum_{j=1}^{n}[F(i,j)-\mu_F]^2\right)}} \tag{4.22}$$

峰值信噪比是评价标准融合图像和实际融合图像的接近程度的另外一种评价参数，峰值信噪比值越大，融合图像就越接近标准参考图像，表明算法融合效果越好。

$$\mathrm{PSNR}=10\log_{10}\frac{F_{\max^2}}{\left(\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[R(i,j)-F(i,j)]^2}{m\times n}\right)} \tag{4.23}$$

其中 $F_{\max}$ 为融合图像像素灰度最大值。

平均梯度：图像的梯度在一定程度上反映了图像亮度变换的显著程度，平均梯度从总体上反映融合图像的清晰度，图像越清晰，平均梯度越大。

$$\begin{aligned}\mathrm{AG}&=\frac{1}{(M-1)(N-1)}\\&*\sum_{i=1}^{M-1}\sum_{j=1}^{N-1}\sqrt{\frac{(F(i,j)-F(i+1,j))^2+(F(i,j)-F(i,j+1))^2}{2}}\end{aligned} \tag{4.24}$$

其中，$F(i,j)$ 为图像像素在位置 (i,j) 的灰度值；M、N 分别为图像的尺寸。

均方误差的定义为

$$\mathrm{MSE}=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[F(i,j)-R(i,j)]^2}{m\times n} \tag{4.25}$$

标准差：衡量融合图像与标准参考图像的差异性，可表示为

$$\delta=\frac{\sigma_R-\sigma_F}{\sigma_R} \tag{4.26}$$

$$\sigma=\sqrt{\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}[G(i,j)-\mu_G]^2} \tag{4.27}$$

信噪比可以比较被评价图像与原图像质量的参数，信噪比的数值越大，图像质量越好。

$$\mathrm{SNR}=10\log_{10}\left[\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}g(i,j)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}[g(i,j)-f(i,j)]^2}\right] \tag{4.28}$$

其中，$g(i,j)$、$f(i,j)$ 分别为图像在点 (i,j) 的灰度值；M、N 为图像的尺寸。

光谱扭曲度可以评价融合图像与源图像之间的光谱失真，所以这个评价参数通常用于多光谱图像的融合结果。

$$D=\frac{1}{m\times n}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|\frac{F(i,j)-I(i,j)}{F(i,j)}\right| \tag{4.29}$$

其中，$m \times n$ 为图像的大小；$F(i, j)$、$I(i, j)$ 为融合图像和源图像 (i, j) 的亮度值。

通常，评价融合结果的好坏并不能单靠某一种评价参数，而往往需要结合多个评价参数去考虑。如果有标准参考图像，融合结果的评价结果还需要结合标准参考图像和源图像进行对比评价。

4.3 基于代数多重网格方法的多聚焦图像重建过程

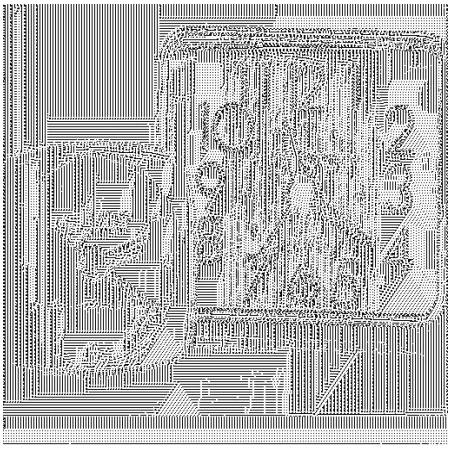
4.3.1 图像粗网格序列的提取

图 4.11 为本次仿真用到的实验图像，它是一组多聚焦的图像，大小为 512×512，其中图 4.11（a）为 ClockA 图像，是一幅包含一部分聚焦、一部分失焦的图像，图像中左前方的钟对焦较差，显得模糊不清晰，右后方对焦较好，比较清晰；而 ClockB 恰恰与图像 ClockA 相反，与之形成了信息互补关系。

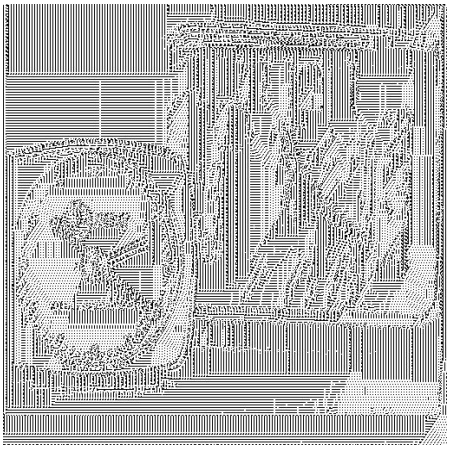
（a）ClockA 图像　　（b）ClockB 图像

图 4.11　实验图像

图 4.12～图 4.14 依次是第一层、第二层、第三层粗化图像。对比第一层粗化的图像序列，两幅粗化图像保留了图像的原始特性：ClockA 的粗化图像保留图像 ClockA 的左边时钟大致轮廓，里面的数字已经不可能很清晰地识别，数字轮廓已经没有了，只能看到时钟指针的大致轮廓；而右边的时钟保留得比较完整，时钟上的数字和指针都能被看到，而 ClockB 的图像却恰恰相反，因为 ClockA 和 ClockB 的图像的信息是互补的。第二层图像将这种特性更加清晰地显现出来，ClockA 的粗化图像的左边时钟已经退化的和周围环境无差别，根本分辨不出来，但是右边时钟的轮廓和表盘上的数字依然能够清晰地看到。第三层粗化图像由于选择的图像点数较少，图像已经模糊不清，但是图 ClockA 的图像右边时钟还是隐约可见的，ClockB 左边时钟的轮廓也可分辨出来。

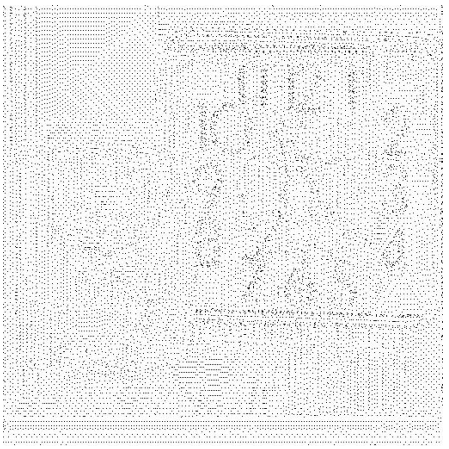


（a）ClockA 第一层粗化图像

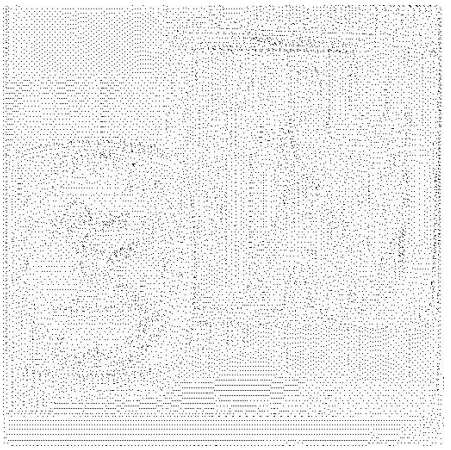


（b）ClockB 第一层粗化图像

图 4.12　第一层粗化图像

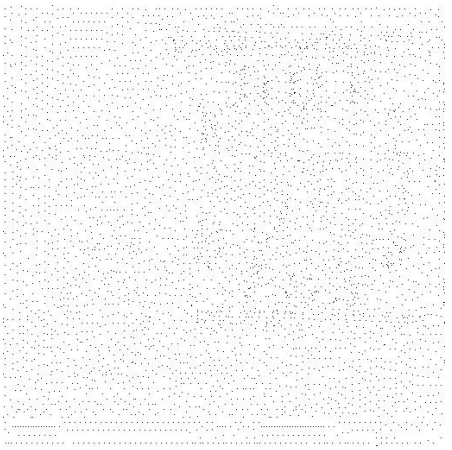


（a）ClockA 第二层粗化图像

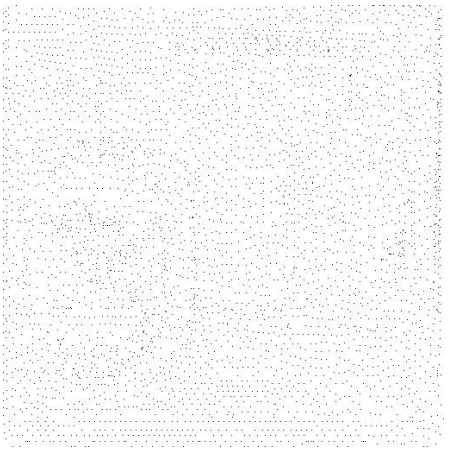


（b）ClockB 第二层粗化图像

图 4.13　第二层粗化图像



（a）ClockA 第三层粗化图像



（b）ClockB 第三层粗化图像

图 4.14　第三层粗化图像

比较粗化序列图像中可以发现：粗化序列图像点分布均匀的地方，对应位置的源图像的灰度变化平缓；而粗化序列图像点分布不均匀的地方，源图像对应位置的灰度变化较剧烈。这个现象说明粗化序列一定程度上保持了图像轮廓特征，这种现象在一定程度上反映了代数多重网格方法中自适应网格的特性。

4.3.2 粗网格序列的插值

在上一小节中，可以得到图像的一系列粗网格序列，这些图像系列随着像素点数的减少，对源图像的信息量保留也是逐渐降低的，但保留的是源图像的信息量较大的像素点。本小节将对这些粗网格序列图像进行插值，即得到各层的重构图像。

根据粗网格序列进行图像插值，因为粗网格序列点集是图像的信息量较大的像素点，理论上可以得到较好的重构结果。为了得到重构图像，必须考虑图像的像素点与粗网格序列点位置的对应问题，只有把多层粗网格序列与源图像的像素点位置对应起来，才能很好地进行图像插值，这是首先要解决的问题。每一层图像重构利用的数据量依次对应粗网格的点的数目，没有规律性。其获取的图片插值结果如图 4.15～图 4.17 所示。相对于第二、第三层插值结果而言，第一层插值结果是最好的，这是因为第一层的点数是最多的，而第三层由于点数较少，所以插值结果出现了块状模糊。

（a）ClockA 第一层插值图像

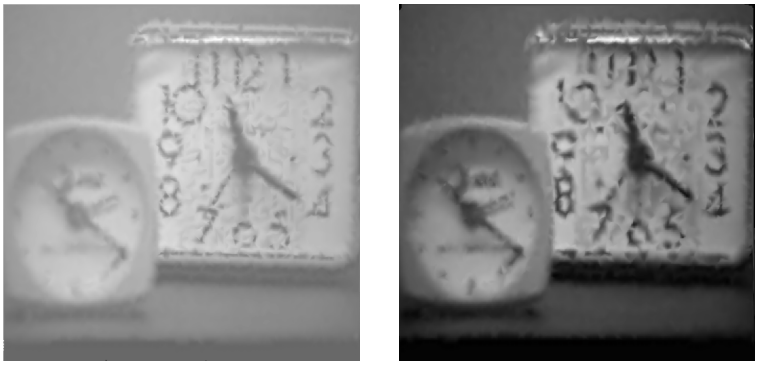
（b）ClockB 第一层插值图像

图 4.15　第一层插值图像

（a）ClockA 第二层插值图像

（b）ClockB 第二层插值图像

图 4.16　第二层插值图像



（a）ClockA 第三层插值图像　　（b）ClockB 第三层插值图像

图 4.17　第三层插值图像

通过对不同聚焦情况的图像进行代数多重网格重建结果分析，综合不同聚焦区域的处理结果，可以进行多聚焦图像的融合处理，本节后面将重点对融合处理方法进行阐述。

4.4　基于区域的多聚焦图像融合算法

本节介绍归一化割（Normalized-cut，NCut）算法，它是一种基于区域的融合算法，流程如图 4.18 所示。

在图 4.18 的图像融合流程中，首先将多聚焦图像做均值融合，利用标准割对均值融合结果做区域划分，紧接着在各个区域中分别选择清晰度较高的图像源进行融合。

基于图论的图像分割技术有很多，如 Min-cut、Normalized-cut、Rate-cut 等方法，其中 Min-cut 如图 4.19 所示，这些都是基于聚类的思想应用到图像分割中。将图像转化成一个加权图，图像中的各个像素点可以认为是图中的节点，权重的大小表示两个像素点之间的相关性，这也是一种自底向上的分割方法。将一幅图像抽象为一个加权有向图 $G=\{V, E, W\}$，其中 V 代表图像的像素，E 代表像素间的关系，W 代表两个关系的权重。基于图论的图像分割方法是近年来的一个研究热点。

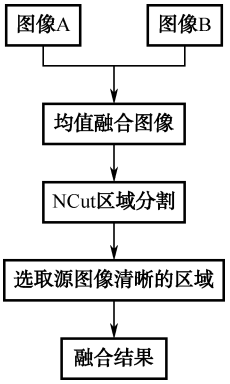


图 4.18　基于边缘提取的多聚焦图像融合流程

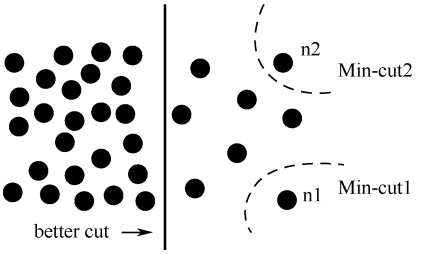


图 4.19　最小割方法示意图

基于图论的图像分割技术主要优点为：抑制噪声能力强，能够考虑区域性质分割。由于

图论方法成熟，而且应用范围广泛，效果比较理想，因此基于图论的图像分割技术也是一个重要的研究方向。基于图论的图像分割方法的特点如下：

（1）在图像分割中不仅根据全局信息分割，而且也会考虑局部特征；

（2）与聚类和谱分析的结合运用，可以扩大应用范围。

Normalized-cut 的基本思想是使用标准化的削减，而不是最小割的分区。Normalized-cut 算法逐渐被应用到各种图像分割中。它先将图像看成是一个加权图，然后对图像进行区域分割。如将图像分为两部分 A 和 B：$A\bigcup B=V$，$A\bigcap B=\Phi$，假设图像的加权图为 W，其为一个对称矩阵，其中元素为 $W(i,j)$ [65]。

Cut(A, B)为子集 A 和 B 的相似性。

$$\mathrm{Cut}(A,B)=\sum_{i\in A,\,j\in B}W(i,j) \tag{4.30}$$

$$\mathrm{NCut}(A,B)=\frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(A,V)}+\frac{\mathrm{cut}(A,B)}{\mathrm{assoc}(B,V)} \tag{4.31}$$

其中，$\mathrm{assoc}(A,V)=\sum_{u\in A,t\in V}W(u,t)$，$\mathrm{assoc}(B,V)=\sum_{u\in B,t\in V}W(u,t)$，代表子集 A 和 B 与全部节点的关系权重。

在上述情况下，最小化 NCut 可以转化为如下的标准特征系统。

$$D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}X=\lambda X \tag{4.32}$$

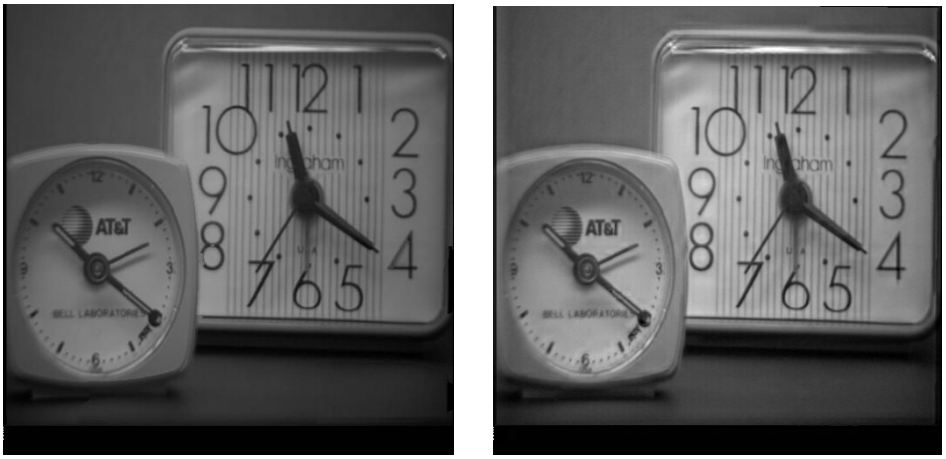
其中，D 是一个对角矩阵，对角线上的元素为 $d_i=\sum_j w(i,j)$，矩阵的大小为 $N\times N$；λ 和 X 分别为相应的特征值和特征向量。

为了测试该算法的效果，本节采用美国 Lehigh 大学提供的不同聚焦图像和标准图像进行了仿真实验，图 4.20 和图 4.11 为实验图像：其中一组为 Disk 的实验图像，另一组为 Clock 的实验图像，均为多聚焦图像。每一幅图像中都有清晰的部分和模糊的部分，分别对应着聚焦和失焦的部分。第一组是 Disk 多聚焦图像的一个时钟和一个书架，两个图像分别在前面的时钟和后面的背景聚焦，所以两个图像的清晰区是不同的；第二组是 Clock 的多聚焦图像的时钟，前面已经叙述。

（a）DiskA 图像　　（b）DiskB 图像

图 4.20　实验图像（Disk 图像）

图 4.21 是 Clock 融合图像，图 4.22 为 Disk 融合结果。由于 Clock 存在标准融合图像，可以用标准差（STD）、互信息（MI）、相关系数（ASS）、熵（E）、平均梯度（AG）这 5 个客观评价参数比较小波融合的结果；Disk 图像融合结果使用评价参数 QAB/F 和互信息（MI）进行比较。Clock 图片融合的结果如表 4.1 所示，融合的结果与图 4.11 源图像相比较，可以发现，该算法能最大程度地保留原始信息的图像，标准差、熵和平均梯度的图像，都最接近标准图像，而小波融合的结果与标准融合图像差别较大，这可以被认为是失去边缘信息或引入人工信息的原因。Disk 图片融合的结果如表 4.2 所示，从表 4.2 中可以看出，该算法更好地保留了源图像的有效信息，没有任何灰度失真，可见该算法优于小波融合算法。



（a）区域融合结果　　（b）小波融合结果

图 4.21　Clock 融合结果比较

（a）区域融合结果　　（b）小波融合结果

图 4.22　Disk 融合结果比较

表 4.1　Clock 融合结果比较

	基于区域的方法	小波融合	标准图像
标准差	40.9058	52.3415	41.0005
互信息	2.7955	2.4834	

续表

	基于区域的方法	小波融合	标准图像
相关系数	0.9957	0.9947	
熵	7.0036	7.4007	7.0610
平均梯度	2.7959	3.8126	2.8286

表 4.2　Disk 融合结果比较

	基于区域的方法	小波融合
QAB/F	0.6969	0.6455
MI	6.9592	5.4051

但是从实验过程来看，归一化割的参数选择比较难控制，进行区域划分时会存在过分割和欠分割的情况，部分结果如图 4.23 所示。

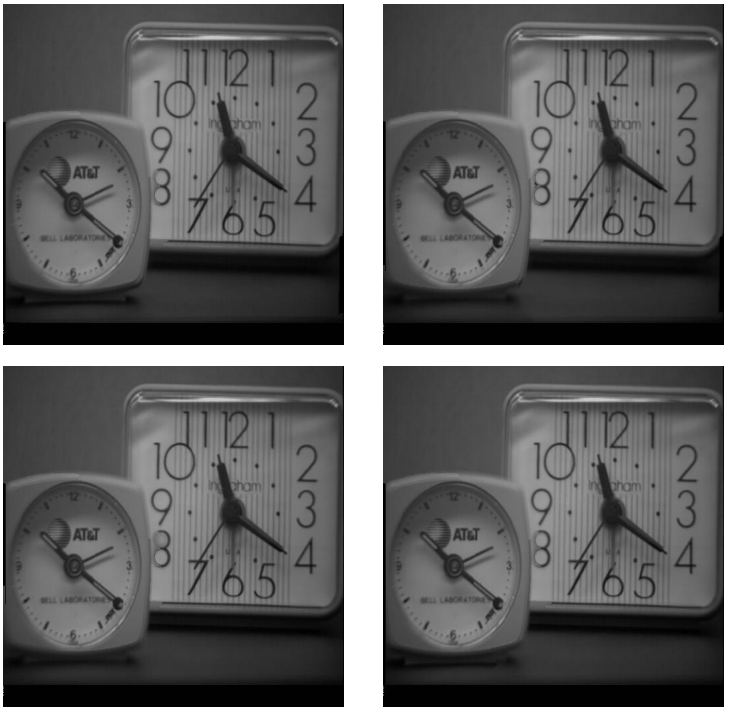


图 4.23　过分割和欠分割导致融合结果模糊

4.5 基于隶属度的多聚焦图像融合算法

由于图像按照一定大小的块来划分清晰度，这样就难免会出现一些异常块，所以本节采用了一种结合多级分块、判断块的归属的图像融合算法。

根据统计学概念，如果样本总量较小，概率不稳定；如果样本总量较大，那么概率总体就趋于稳定。用到分块图像融合中，如果划分的块的尺寸较大，块的像素个数多，等价于样

本总量较多，那么块的隶属情况就很稳定；否则就不稳定。但是子块划分得较小也是一个必然的要求，因为块越小图像融合就越精细。文献[69]中的方法为采取块大小从大到小多级分块来抑制划分区域的不稳定。

4.5.1　区域决策图

在处理多重网格算法时，可以对整幅图像进行处理，也可以对某个区域进行处理。将不同多聚焦图像同一位置的图像子块进行代数多重网格重建，然后计算重建图像与源图像的均方误差。假设多聚焦图像为 $\mathrm{Im}\,a$， $\mathrm{Im}\,b$，划分的尺寸大小为 $K\times K$，分别重建后，得到两者的均方误差 MSE1 与 MSE2，在每一块中做比较。

本节用区域决策图 C 记录各个块的清晰度，这是一个最粗糙的判定结果。判定方法如下：

（1）若 $\mathrm{MSE1}>\mathrm{MSE2}$，则表示 $\mathrm{Im}\,a$ 清晰，将该块的所有像素置为 1，表示选择为 $\mathrm{Im}\,a$；

（2）若 $\mathrm{MSE1}<\mathrm{MSE2}$，则表示 $\mathrm{Im}\,b$ 清晰，将该块的所有像素置为 0，表示选择为 $\mathrm{Im}\,b$；

（3）若 $\mathrm{MSE1}=\mathrm{MSE2}$，则表示 $\mathrm{Im}\,a$ 与 $\mathrm{Im}\,b$ 清晰度相同，将该块置为 0 或者 1 均可。

4.5.2　满足一致性原则的区域隶属图

如果仅仅通过单一尺寸的块大小来判断图像是否清晰是不可靠的，而且图像的块大小选取是很难的，单独靠一种尺寸的大小的融合结果也是不可靠的。当块大小固定时，如果块的位置是在多聚焦图像模糊与非模糊的分界区域，那么就会产生错误划分。因为这个区域的块包含了图像的清晰与模糊像素，无论把它归属于模糊还是清晰的区域都是错误的，所以要采取一致性原则将位于模糊与清晰边界的块给提取出来。

图 4.24 为 3×3 的区域决策图，其中 C5 为当前处理块，周围是它的 8 邻域块，那么满足一致性原则的区域隶属度的判别规则如下：

（1）若 C5 与周围区域 C1…C9 等块的清晰度划分一致，同时为 1 或为 0，那么 C5 的清晰度判断不改变；

（2）若 C5 与周围区域 C1…C9 等块的清晰度划分不一致，那么 C5 的清晰度就改为 0.5，表示块的清晰度尚待确定。

C1	C2	C3
C4	C5	C6
C7	C8	C9

图 4.24　3×3 的区域决策图

采用了一致性原则的区域隶属度就可将位于清晰与模糊边缘的块提取出来，以免误划分。

4.5.3　多级分块的区域划分图

在每一块上都可以得到区域决策图和区域隶属图，那么在多级分块中就可以采取多级分块的区域划分图来进一步提高区域的划分精度。随着子块的大小降低，图像就会被误划分。因此本节采取多级分块进行再次的区域划分，将尺寸大的块的稳定性尽可能地保留到图像较小的块中。

在本节中首先对图 4.13 中图（a）图像进行分块，按照从大到小的级别进行判断。将图像划分为块数为 4×4、8×8、16×16、32×32、64×64、128×128，共 6 级，那么每一块的大小依次降低，为 128×128、64×64、32×32、16×16、8×8、4×4。在每一级别上判断块隶属度、

区域决策图 C 和区域隶属图 S，多级分块的区域划分图 A 产生如下。

（1）在第一级时，区域划分图和区域隶属图是一样的，$A = S$。

（2）对于第二级以及以后的分块中，区域划分图采用如下的方法生成：

① 如果第 $(i-1)$ 级区域划分图和第 i 级区域隶属图所对应的块同为 1，或者其中之一为 1，则第 i 级区域划分图 A 对应的块为 1；

② 如果第 $(i-1)$ 级区域划分图和第 i 级区域隶属图所对应的块同为 0，或者其中之一为 0，则第 i 级区域划分图 A 对应的块为 0；

③ 如果第 $(i-1)$ 级区域划分图和第 i 级区域隶属图所对应的块同为 0.5，那么第 i 级区域划分图 A 对应的块为 0.5。

融合规则：按照区域划分图 A 为权重基准，可以得到融合图像 $R(x,y)$，即

$$R(x,y) = A(x,y) * \mathrm{Im}\,a(x,y) + (1 - A(x,y)) * \mathrm{Im}\,b(x,y) \tag{4.33}$$

图 4.25 为不同块大小的区域决策图、区域隶属图和区域划分图，图 4.26 为最终的融合结果。与原始数据做比较，可以发现本算法的融合结果较好地保留了原始图像的清晰区域，同时在一定程度上克服了分块融合的常见现象：边界模糊和块效应。但是具有以下不足：块大小难以把握，而且图像的清晰模糊边界还是定位在块边界，这与实际不符合，这两个缺点给实际运用带来很大困难，也给实际的融合结果带来很大的影响。

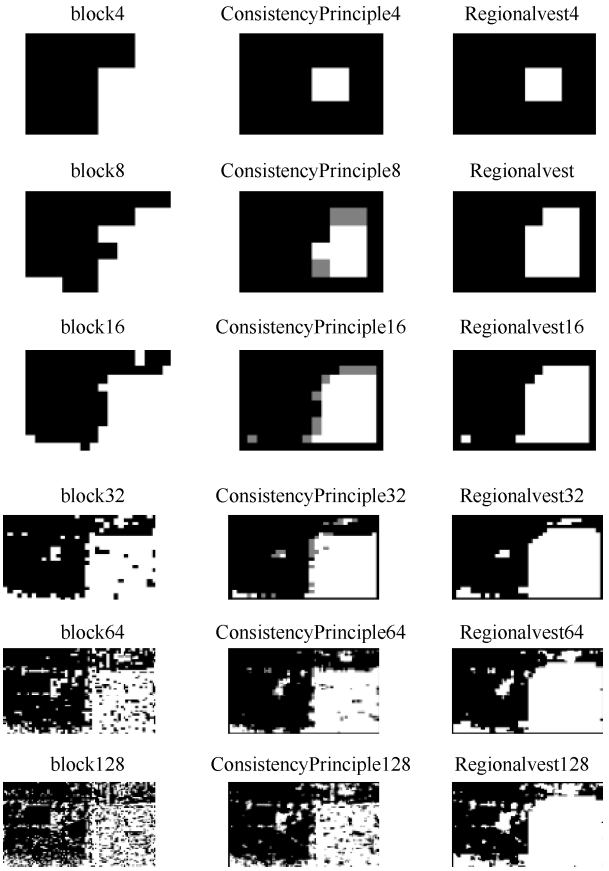


图 4.25　不同块大小的区域决策图、区域隶属图和区域划分图

图 4.26　基于隶属度的融合结果

将基于多级分块隶属度融合结果与小波融合结果做比较，采用的具体评价参数为互信息（MI）、相关系数（ASS）、信噪比（SNR）、峰值信噪比（PSNR）、均方误差（MSE）和光谱扭曲度（Spectral Distortion）。表 4.3 为融合结果评价参数比较。

表 4.3　融合结果评价参数比较

参　　数	基于隶属度方法的融合结果	小波融合结果
互信息	2.891902	2.27593
相关系数	0.99718	0.991083
信噪比	11.51373	10.527
峰值信噪比	34.49974	32.03432
均方误差	23.07311	40.70506
光谱扭曲度	2.219567	3.640687

从表 4.3 中可以看出，基于隶属度方法的融合结果优于小波融合结果。基于多级分块的隶属度融合算法结果与小波比较，互信息量为 2.89，较小波融合结果高，而均方误差和相关系数是评价融合结果与标准参考图像的近似程度，均方误差为 23.07311，远优越于小波融合结果，相关系数也较小波融合结果要好，信噪比和峰值信噪比说明了多聚焦图像融合算法的结果的质量，从表 4.3 中反映了基于隶属度方法在一定程度上优越于小波融合结果。

4.6　基于边缘提取的多聚焦图像融合算法

针对 4.4 节和 4.5 节仿真实验结果的不足，需要进行改进。首先是如何使得区域划分更为合适，其次是清晰和模糊区域的划分边界尽量应贴近实际边界。本节提出的算法，在一定程度上解决了上述两个问题，且对算法做了仿真，并对实验结果进行了定性和定量分析。

4.6.1　基于边缘提取的多聚焦图像融合算法流程

本节提出了一种基于边缘提取的多聚焦图像融合算法，首先根据均值融合算法获取融合结果，然后根据融合结果由成熟的边缘提取算法提取多聚焦图像边缘，这样可以避免单一多

聚焦图像的边缘不精确性，但是获取的图像边缘不一定是连续的，会对区域划分产生不必要的影响，为了避免这一因素，需要将断裂的边缘进行连接，寻找图像闭合区域，最后需要根据一部分的先验知识将一部分小区域剔去，这样就可以得到多聚焦图像的区域划分。如此划分的粒度与多聚焦图像中的实际物体大小一致，而且划分的边缘会包含清晰与模糊的边界，这样就避免了基于隶属度方法的弊端。根据第 2 章提出的清晰度表示，选取清晰度较大的区域进行融合，这样得到的图像区域是按照图像边缘进行划分的，不会产生边缘异常情况。总的流程如图 4.27 所示。

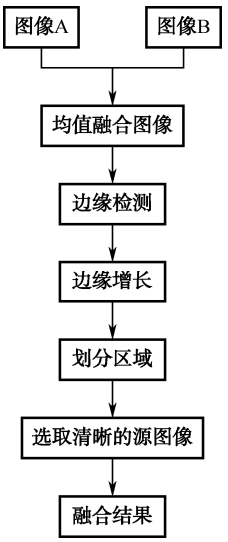


图 4.27　基于边缘提取的多聚焦图像融合流程

4.6.2　获取多聚焦图像边缘

获取多聚焦图像边缘，首先需要对图像进行均值融合，抑制从单幅多聚焦图像提取边缘的影响，然后进行边缘提取，最终将区域闭合。在此基础上进行多聚焦图像融合，因此分成以下几个步骤来讨论。

1．均值融合

考虑到多聚焦图像有多幅图像，虽然每一幅图像单独检测边缘是可行的，但是多聚焦图像中聚焦的地方清晰，失焦的区域模糊，如果采用某种边缘检测方法对不同聚焦图像进行处理，会得到同一区域不同的边缘检测结果，造成边缘检测结果的不一致。所以本节采取均值融合检测边缘，这样在一定程度上避免了以上情况的发生。

若对于两幅不同的多聚焦图像而言，记为 $I_a(x,y)$， $I_b(x,y)$，其均值融合的公式为

$$I(x,y)=0.5\times I_a(x,y)+0.5\times I_b(x,y) \tag{4.34}$$

本次实验采用的图像如图 4.20 所示，首先对图像进行均值融合，然后再进行边缘检测。均值融合结果如图 4.28 所示。

从均值融合结果可以看出，均值融合结果与图 4.20（a）相比，图像的背景区域清晰度有

所提高，但与图 4.22（b）中的背景区域比较，书架与书籍边缘蒙上了一层雾状的阴影，右前方的时钟也存在这种情况。但是均值融合结果仅仅是一个中间步骤，用来获取图像的边缘，这样做是为了抑制从单幅多聚焦图像提取边缘的影响。

图 4.28　均值融合结果

2．提取边缘

在获取均值融合图像后，接下来要根据均值融合图像结果进行边缘检测。边缘检测有很多种方法，如 Sobel 边缘检测器、Prewitt 边缘检测器、Canny 边缘检测器、基于 Hough 变换的边缘检测等。本节采用 Sobel 算子对图像进行检测，因为 Sobel 算子有以下优异特性：检测的边缘较好，能够在一定程度上平滑噪声，可以提供较为准确的边缘信息，计算不复杂[70]。

Sobel 边缘检测器使用一阶导数值 G_x 和 G_y 的近似值，故 Sobel 检测器按照以下公式计算。

$$\begin{aligned} g &= [G_x^2 + G_y^2]^{1/2} \\ &= \{[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)]^2 \\ &\quad + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2\}^{1/2} \end{aligned} \tag{4.35}$$

图像邻域如图 4.29 所示，其中 z5 是中心元素，其余为其八邻域的像素。

Sobel 算子在垂直方向和水平方向上的模板如图 4.30 所示。将此模板运用到均值融合结果图 4.28 中，检测到的边缘如图 4.31 所示。

z1	z2	z3
z4	z5	z6
z7	z8	z9

图 4.29　图像邻域

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

图 4.30　Sobel 算子

从图像中可以发现，图像检测到的边缘较清晰，而且也体现了图像的大致轮廓，但是因为图像检测的边缘不是连续的，或者说图像的边缘是断开的，有的是与图像边界有一个像素或两个像素的缝隙，一条直线的中间也可能出现断裂，所以需要对边缘进行生长。

图 4.31　均值融合检测到的边缘

3．边缘生长

由于 Sobel 算子检测的边缘存在边缘断裂的情况，因此需要对边缘进行生长和连接，如将直线上的断裂边缘连接，这样在一定程度上有助于图像轮廓的形成，使之便于后续的处理。

本算法采用一种基于主动生长的边缘连接算法[71]，利用端点沿着端点方向的生长策略，将端点进行生长和退化，生长可以把断裂的边缘连接起来，退化可以让它消除。检测端点的结构元素如图 4.32 所示。

x	0	0
1	1A	0
x	0	0

（a）水平方向向右的端点检测

1	0	0
0	1A	0
0	0	0

（b）对角线端点检测

图 4.32　检测端点的结构元素

边缘端点检测采用图 4.32 中的结构元素进行操作。图 4.32（a）为水平方向向右的端点检测结构元素，A 表示检测的端点，x 表示任意值，1 为检测的图像边缘。那么图 4.32（a）按照顺时针 90°、180°、270°旋转变成另外三个方向：向上、向左、向下的端点检测结构元素。图 4.32（b）为对角线上的边缘检测结构元素，图 4.32（b）按照顺时针 90°、180°、270°旋转变成另外三个方向的端点检测结构元素。一共有 8 个方向的端点检测结构元素。

图 4.33 是用于生长和退化的结构元素。将图 4.33（a）沿着顺时针方向旋转 45°、90°、135°、180°、225°、270°、315°可以得到另外 7 个方向端点生长结构元素，那么一共有 8 个方向上的生长结构元素，可以与图像端点检测结构元素一一对应起来。

0	0	0
0	1A	1
0	0	0

（a）生长结构元素

0	0	0
0	1A	0
0	0	0

（b）退化结构元素

图 4.33　生长和退化结构元素

生长的步骤为：

（1）使用端点检测算子检测当前像素点是否为端点；

（2）若是，将端点按照生长结构元素对端点进行生长；

（3）若否，继续（1），直至点扫描结束。

退化的步骤为：

当生长完成后，当前点如果为端点，执行退化操作，并将退化像素记录下来，否则不执行操作。

最后在生长的步骤完成的边缘图像中除去退化步骤的检测的退化像素，得到第一次生长和退化操作的结果。

对于一次生长和退化操作，可以连接两个像素断裂的边缘，而对于两个像素以上的边缘断裂，可以通过多次生长退化操作进行连接。实验表明，迭代两次就可以满足大部分断裂的边缘连接需求。

根据此边缘生长算法，对根据均值融合算法得到的融合结果检测到的边缘图像进行边缘生长和退化操作，就可以得到图 4.34 所示的闭合边缘。

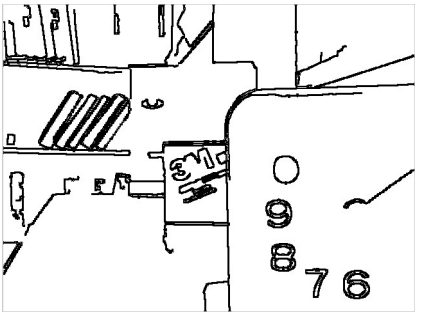
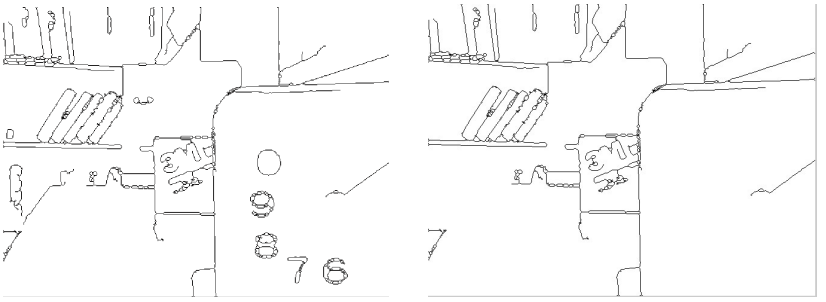


图 4.34　边缘连接后的图像

4．边缘细化和去除面积较小的区域

在得到的边缘连接后的图像中，由于算法的作用，得到的边缘较粗，边缘的宽度占用了大于 1 个像素的宽度，这种边界会影响图像融合结果的精度，因此必须要对图像进行细化操作。边缘细化的操作利用 Matlab 中的 bwmorph()函数，结果如图 4.35（a）所示。

从获取到的边缘图像中去除一些小的区域不会影响图像的融合结果。去除小面积区域后的结果如图 4.35（b）所示。



（a）细化边缘的图像　　（b）去除小面积区域后的图像

图 4.35　得到闭合区域

4.6.3 分区域融合

综上，可以得到图像的闭合区域，记为 $R_1, R_2, \cdots, R_{n-1}, R_n$，共 n 个区域。根据基于代数多重网格方法理论来评价区域清晰度，选取较清晰的区域进行融合，进行区域选取的公式为

$$\mathrm{IM} = \begin{cases} \mathrm{Im}a, & 当 C_{aRi} > C_{bRi} \\ \mathrm{Im}b, & 其他 \end{cases} \tag{4.36}$$

其中，C_{aRi} 为图像 $\mathrm{Im}a$ 在区域 R_i 的清晰度；C_{bRi} 为图像 $\mathrm{Im}b$ 在区域 R_i 的清晰度；i 取值范围为 $1, 2, \cdots, n$ 。

最后合并每个区域的融合结果，边界像素采用均值处理，就得到最终的融合结果，如图 4.36 所示。

图 4.36　基于边缘提取的融合结果

通过图 4.36 可以看出融合效果较好，每个区域内都选择了正确的清晰块作为融合结果，而在边缘部分也没有跳变点。

4.6.4 实验结果评价

将本融合结果与其他常见算法的融合结果做比较。其他算法的融合结果如图 4.37 所示。均值融合结果虽然能够提高图像模糊区域的质量，但是较其他融合结果而言，图像清晰区域的质量下降了；最大值、最小值融合的结果图像的亮度会变化，而且在一定程度上存在模糊或者阴影的地方，如图像中物体的边界部分；从整体而言，小波融合和拉普拉斯金字塔融合两种方法得到的结果是较好的，但是从人眼上难以判别两者的优劣，故必须采取定量的评价。

参照前面给出的一些评价参数，本节也给出了客观融合性能评价参数。图 4.38 为越大越好型评价参数曲线。图 4.39 为越小越好型评价参数曲线。图 4.40 为图像均值评价参数曲线。从大多数曲线来看，本节算法在总体性能评价标准上较优异，而且比基于区域的多聚焦图像融合算法和基于多级分块隶属度的多聚焦图像融合算法优异。

为体现本节算法的优越性，参考最近几年的论文中出现的数据，本节做了客观评价数据比较，并与标准图像的相关参数做了比较，如表 4.4 所示。具体参数为平均梯度（AG）、互信息量（MI）、熵值（E）、均方根误差（RMSE）、标准差（SD）、空间频率（SF）等。从表 4.4 数据可以发现，较其他文献而言，本节算法得到的结果与标准图像的相关参数是最接近的，如熵值、平均梯度、标准差、空间频率、互信息量较其他文献算法结果大，而均方根误差较小，说明本节算法较好。

（a）均值融合结果

（b）最小值融合结果

（c）最大值融合结果

（d）PCA 融合结果

（e）拉普拉斯金字塔融合结果

（f）小波融合结果

图 4.37　其他算法融合结果

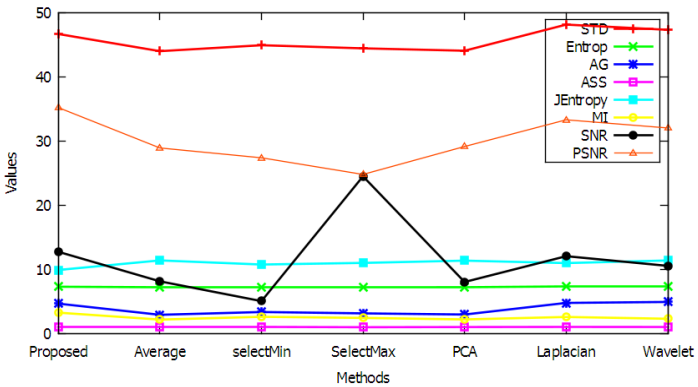


图 4.38　越大越好型评价参数曲线

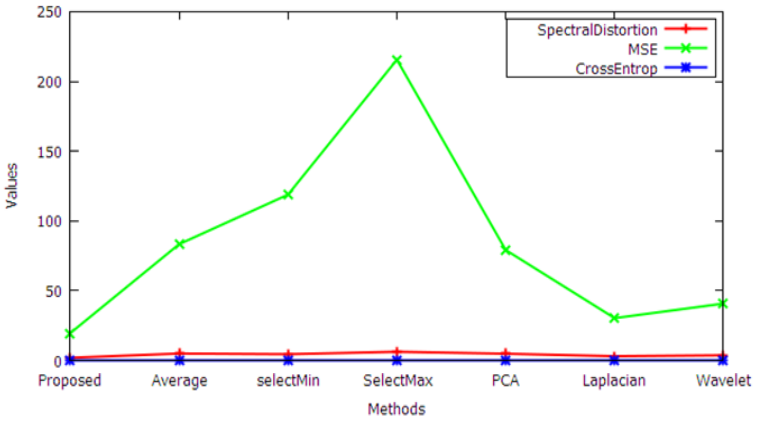


图 4.39　越小越好型评价参数曲线

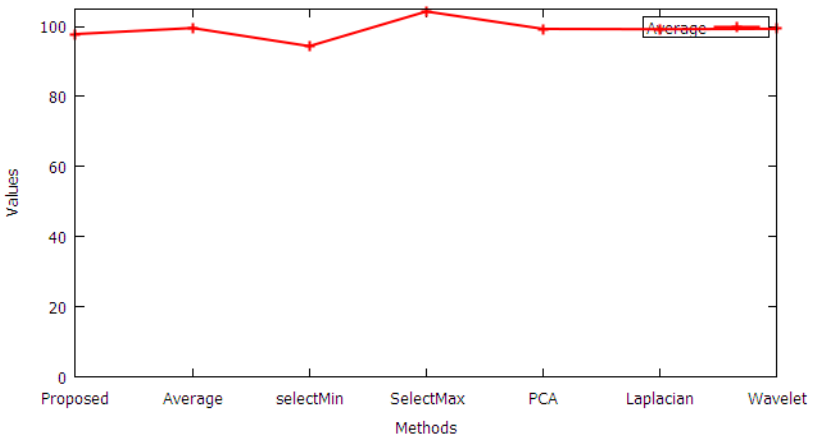
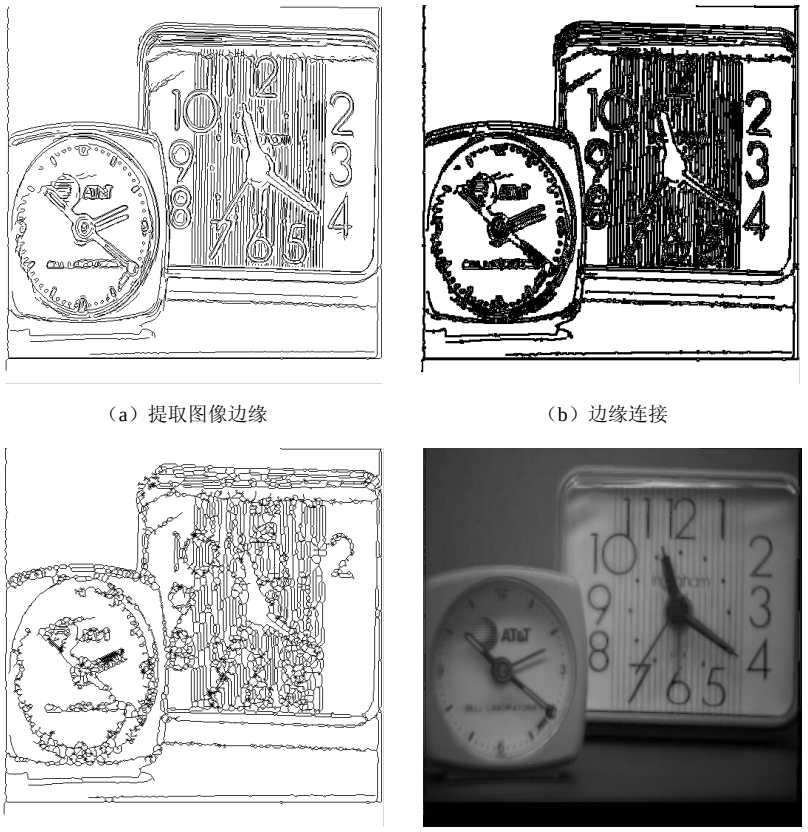


图 4.40　均值评价参数曲线

表 4.4　结合标准图像与当前文献对比结果

	AG	MI	E	RMSE	SD	SF
基于边缘结果	4.6316	3.2315	7.2705	4.4187	46.7024	15.3082
标准图像	4.63		7.2774		46.7034	15.2794
文献[72]		2.39				8.51
文献[73]			7.4274			
文献[74]	0.1735	2.6804				
文献[75]			7.318	6.2079		12.91
文献[76]			7.2903			
文献[77]	5.4227		7.4741		47.1033	18.25

使用该方法针对 Clock 图进行融合，得到的结果如图 4.41 所示。



（a）提取图像边缘　（b）边缘连接

（c）闭合区域　（d）融合结果

图 4.41　Clock 图像融合结果

4.7　基于清晰和模糊边界判定的多聚焦图像融合算法

因为代数多重网格重建的 MSE 可以在一定程度上评价图像的清晰和模糊的边缘，因此使用 MSE 值来界定清晰和模糊区域的边缘就是一种比较合适的选择。但是区域太小，MSE 值不够稳定，而区域太大，则会将清晰和模糊的区域混在一起，因此需要进行合适的算法设计，这样才能较好地判定清晰和模糊的边界。

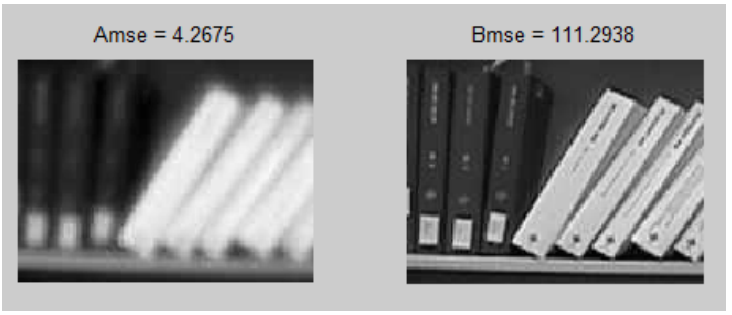
4.7.1　基于清晰和模糊边界判定的方法的思路

针对多幅原始图像进行分块，利用代数多重网格方法重构后，分块计算 MSE，这样就能区分出分块图像的清晰度，这种特性可以用来指导图像融合。当重建结果与两个源图像之间的 MSE 的差大于某个阈值时，可以确定其中只包含了某个清晰块，如图 4.42（a）所示。直接选取清晰块进入融合结果，表达式为

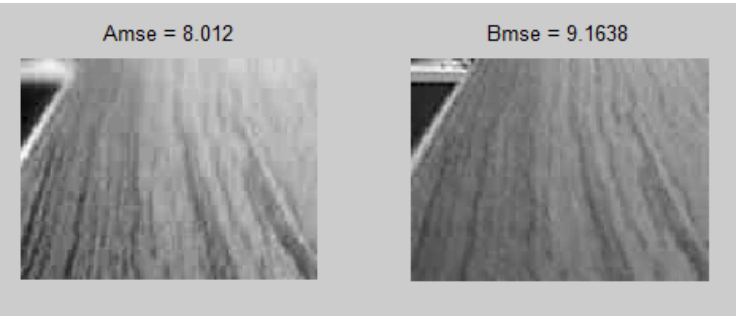
$$\mathrm{MSE} = \sum_i \sum_j (R_B(i,j) - I_B(i,j))^2 \tag{4.37}$$

$$I_{x,y}=\begin{cases} I_A(x,y) & \mathrm{MSE}_A > \mathrm{MSE}_B \\ I_B(x,y) & \mathrm{MSE}_A \leqslant \mathrm{MSE}_B \end{cases} \tag{4.38}$$

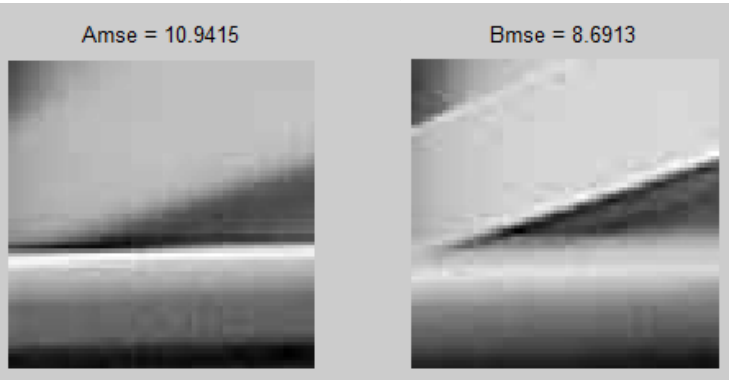
而当与两个源图像的差小于某个阈值的时，可能出现两种情况。一种情况是包含了某个清晰物体，但是该清晰物体的特征表现较弱，因此导致重建后与两个源图像的差值较小，如图 4.42（b）所示。另一种情况是该区域包含了清晰和模糊两个部分，如图 4.42（c）所示。针对这些区域分成四个子区域进行重建，如果重建后四个子区域的分区跟上一级分区一致，则认为是第一种情况，直接根据结果选择合适的源图像进入融合结果即可，否则认为是第二种情况。这时需要确定清晰块和模糊块的边界。我们最终的目的是将清晰块的图像作为最终的融合图像。这可以避免空间信息的丢失，但可能会出现块模糊的现象，如图 4.43 所示。针对分块中有清晰块和模糊块的情况，可以采用自适应分块的方法来处理。



（a）分块中只包含清晰块



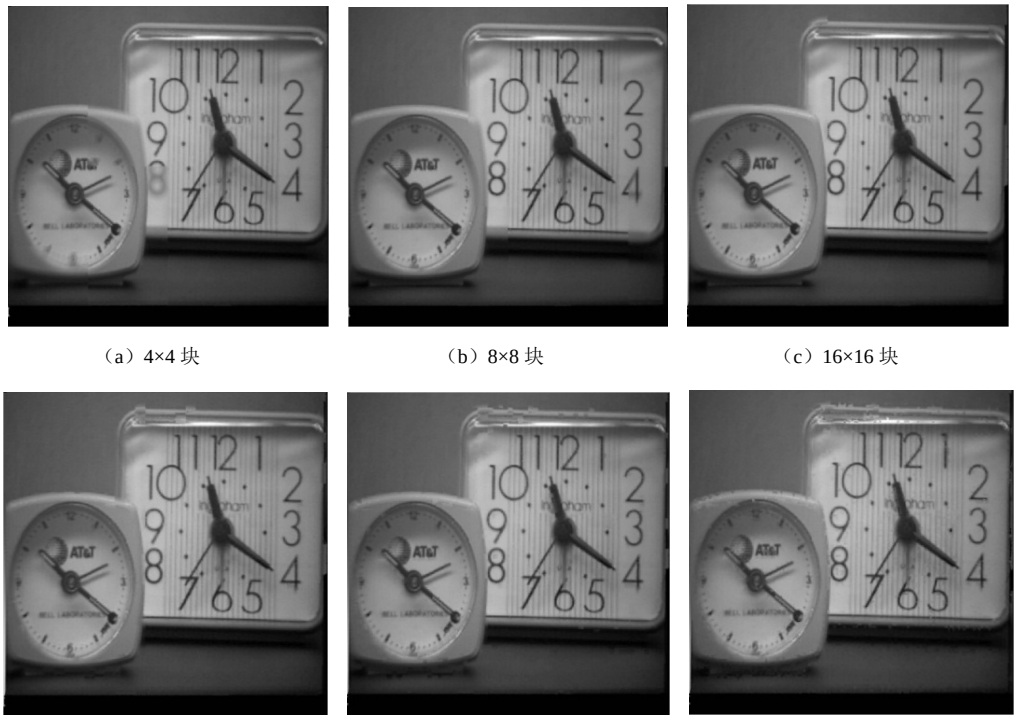
（b）分块中包含区别较小的清晰块



（c）分块中包含清晰块和模糊块

图 4.42　部分分块的检测结果

图 4.43 为采用固定分块时的结果图。可以发现，分块越少（如 4×4），每个块越大（128×128），这时会出现较大的模糊块，但是块内相对来说比较完整，如果刚好一个清晰物体在这个块中，就会得到比较准确的融合结果。随着分块变多，这时模糊块会变得越来越小，但是会出现一些锯齿状的小模糊块，随着分块越来越小，如果到像素级，锯齿现象会更明显，如图 4.43（f）所示。同时也发现，随着块变得越来越小，产生的块与块之间的边界会趋向于两个钟表之间的轮廓边界。因此，当分块变成像素级时，这时的结果会越来越倾向于两个清晰块之间的边界，这也是本算法的最大特点。



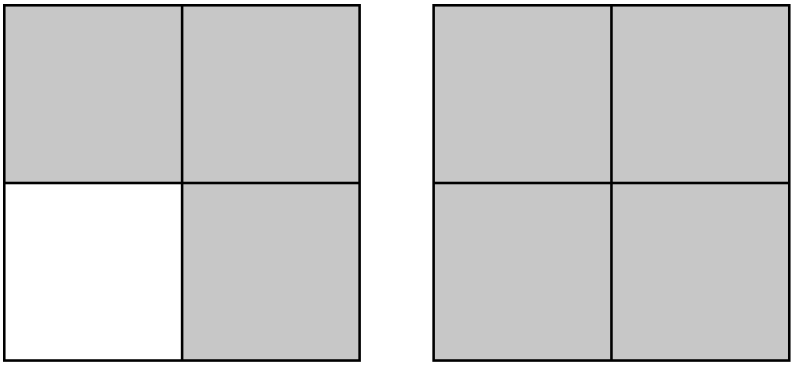
（a）4×4 块　（b）8×8 块　（c）16×16 块

（d）32×32 块　（e）64×64 块　（f）128×128 块

图 4.43　采用固定分块的结果图

根据这一发现，我们需要结合不同分块方式的优点来进行综合。首先采用较大的分块，一般将整个图像分成四份。对其中每个分块进行处理，选择代数多重网格重建结果与原始图片之间的 MSE 较大的原始图片作为结果图片。

如果在融合结果中两个分块之间存在一些较大的分歧，则说明这两个分块需要进一步检查。这时需要检查两个分块某个方向上相邻的边缘像素的梯度和。如果该方向上梯度和大于给定的阈值 T 时，则进行标记，需要考虑进一步处理。如果有两个相邻边的边缘和大于阈值，如图 4.44（a）所示，其中大于阈值的边用粗黑线表示，则需要将其进一步分块，对其中三个灰色区域做进一步的重建和融合；如果有相邻三边都具有上述情况，如图 4.44（b）所示，则直接用另外一个图像源块来替代，然后重新进行边缘的检测。



（a）两个相邻边的边缘和大于阈值的情况　　（b）三个相邻边的边缘和大于阈值的情况

图 4.44　分块边缘检测准则

在进一步处理时，需要把这个区域进一步分成四份，对其中每个块进行重建并计算 MSE，采用前面的步骤进一步判断。判断结束的一个条件是当每个块与周边的块之间的分歧较小时，这需要设置一个阈值。目前这个阈值是我们手工设定的，考虑到视觉的承受能力，可以设置稍微大一点，但是如果考虑到图片放大的需要，可以设置得稍微小点。结束的另一个条件是分块进入到像素级，目前一般还不需要做到这一步，除非两个清晰边缘之间的物体边界非常曲折。因为做到像素级意味着可能会出现一些锯齿状的边缘，从视觉上来看不是太好。基于代数多重网格的自适应算法的流程图如图 4.45 所示。

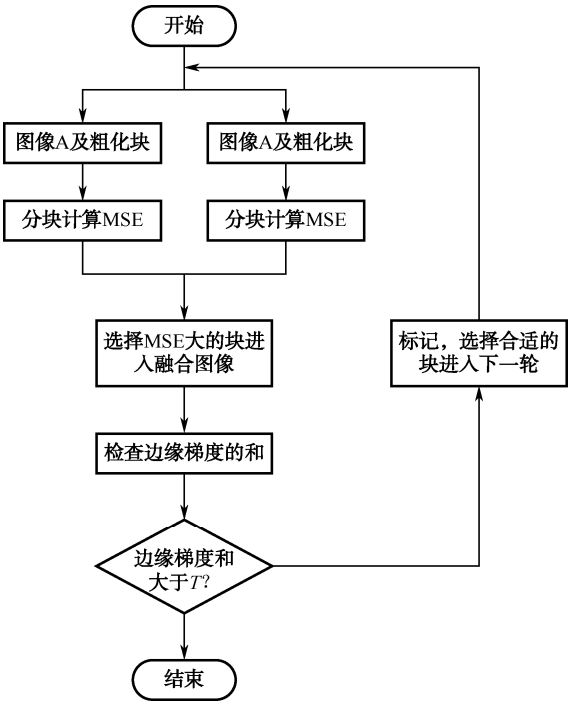


图 4.45　基于代数多重网格的自适应算法的流程图

根据上述分析，使用代数多重网格进行图像融合算法的主要步骤为算法 4.1，使用代数

多重网格进行融合的结果如图 4.46 所示。

图 4.46　使用代数多重网格进行融合的结果

当然，还有一些情况我们需要考虑，如当某个清晰物体比较小的时候，如果第一次分块时包含在一个分块之中，这时就可能无法将其检测出来。针对这一问题，我们也做了一些适当的考虑。一是在开始的时候将块分得稍微小点，一次分 8×8 块或者分更多块，如何对初始分块的个数做设定，也需要进一步地研究；二是按照前面的过程处理完后，再对每个块进行后续的检查。当然如果物体足够小以至于 AMG 方法提取的特征无法描述该分块的清晰和模糊程度时，也会将这个块进行误判。

算法 4.1　使用代数多重网格进行图像融合算法

1．利用 AMG 提取输入图像 I_A 与 I_B 的粗层数据。

2．将上轮的两幅图像块分成四块，利用粗层数据对每块进行重建，采用 Matlab 中的 griddata 函数进行。计算每块重建后的结果与对应原始块之间的 MSE，如果与两个源图像之间的差大于某个阈值，则直接选择对应的图片进入融合结果。

3．如果两个源图像之间的差小于某个阈值，则根据下一级重建情况判定其中是否包含清晰块和模糊块。

4．如果分块中包含清晰块和模糊块，则继续分块重建，按照图 4.45 给出的原则进行处理，转至步骤 2。

5．直到所有的块之间的梯度和小于阈值，终止该过程，得到融合结果。

前面所述的算法在遇到清晰与模糊的边界与分块的边界不一致的时候，会出现较大的块状模糊，如图 4.47（a）所示。

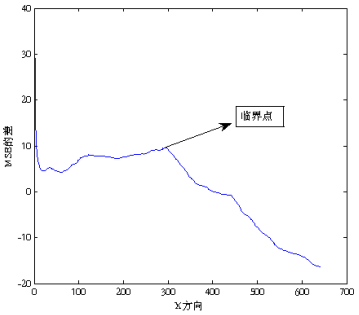
利用 AMG 方法进行提取，在较大的区域中可以得到较为精确的判定，但当区域缩小时，难以通过单独的区域划分来得到精确的判定。

所以可以通过前面大区域的处理得到清晰和模糊的大区域的边缘，也就是粗略的边缘，然后在这个区域中进行搜索，找到清晰和模糊区域的精确边缘，这可以通过对整个区域进行 MSE 差值的趋势进行分析，找到一个临界值导致 MSE 出现单边的下降，这个临界值就是我们要找的临界点，如图 4.48 所示。但有时由于清晰和模糊的边缘较为复杂，因此可能无法精确确定，这时只能通过一些近似的方法来进行处理。

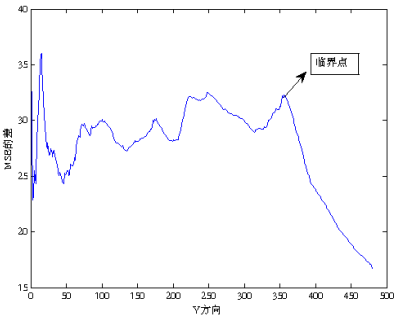
（a）清晰块和模糊块的粗略边界

（b）清晰块和模糊块的临界边界

图 4.47　获取清晰块和模糊块的临界边界的过程



（a）水平方向的临界点

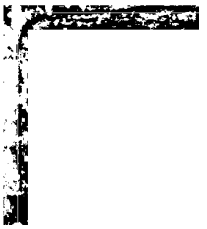


（b）垂直方向的临界点

图 4.48　通过 MSE 的差寻找水平和垂直方向的临界点

因为较小的区域关联的像素较少，得到的 MSE 值不够稳定，所以无法作为最终判决的唯一标准。因此我们对多个不同邻域大小的区域计算均值，得到一个区域决策图。如图 4.49（a）为通过对不同邻域大小的区域进行均值后得到的结果，其中不同邻域的大小分别是 1、3、5。图 4.49（b）是将粗略边界外的点去掉得到的结果。可以在这个决策图上进行相应的融合处理。

（a）



（b）

图 4.49　通过对不同邻域大小的区域进行均值后得到的决策图

进行融合处理的原则有以下几点：

（1）以临界边缘为中心，将临界边缘与粗略边缘之间的区域分别划归不同的区域，如左边粗略图边界与临界边界之间用 diskB 作为融合结果，而另外一边用 diskA 作为融合结果，在这种情况下，可能会出现较少的块状模糊。

（2）将粗略边缘包括的区域进行均值融合，以减少由于不确定性带来的误差，这种方法会在块与块的边缘处出现不连续性。

（3）对该区域结合小区域决策图来进行。将临界边缘与粗略边界之间的区域进行分区，设定该区域的邻接区域（临界边缘另一侧的不属于邻接区域）。当该区域与邻接区域的判定一致时，则认为该区域跟相邻的区域属于同一个源图像，否则暂时不予判定。针对未进行判定的区域，需要进行最终的裁定。这可以通过对该小区域与相邻区域的边缘的梯度和进行，如果该小区域与其中的某个区域的梯度和较小，则将该区域判定为相应的源图像，也就是将该区域判定为某个区域之后的损失减到最小。

因为 AMG 方法跟其他清晰度检测指标一样，对于很小的区域很难得出一个完全准确的结果，因此我们考虑的重点是怎么缩小这个粗略区的范围，将模糊减到最小，尽可能地将清晰区域提取出来，接近原始图像的熵，将模糊的程度限制在视觉感知的范围之内。

根据上述分析，本节融合算法的主要步骤为算法 4.2。

算法 4.2　使用代数多重网格进行图像融合算法

1．利用 AMG 提取输入图像 I_A 与 I_B 的粗层数据。

2．将上述两幅图像块分成 $m\times n$ 块，如针对 640×480 的图片可以分成 4×4 块，其中每块的大小为 160×160 个像素。对该区域进行 MSE 比较，将 MSE 大的块直接选入融合图像。

3．判断相邻分区的邻接性，如果相邻分区边缘的梯度和较大，就进行标记，作为精细区和模糊区的粗略边界区。

4．针对相关区域进行全局 MSE 差的比较，找到 MSE 差的临界点，作为精细区和模糊区的精细边界的起始搜寻边界。

5．按照上述三个融合处理的原则对该区域进行相应地处理，得到最终的融合结果。

目前，使用代数多重网格方法进行图像融合在实时性方面较为欠缺，在 Intel Pentium dual E2180 CPU 2.00GHz，1GB 内存，32 位 Windows XP 系统，Matlab R2009 的平台上求解一个 512×512 的图像问题需要花 26s，在 Intel core2 duo CPU 2.26GHz×2，2GB，64 位 Linux 系统，Matlab R2009 的平台上求解一个 512×512 的图像问题需要花 10s。无法做到实时的处理，这需要在后续的研究中进一步优化。我们在编程时使用了 Matlab 调用 python 的代码，如果将其通过 C++代码统一实现，就可以大大降低运算时间。

4.7.2　与其他算法的主观研究

1．模糊图片之间的融合结果

图 4.50 是对 clockA 和 clockB 进行融合之后的结果，并将结果与其他算法进行比较，如 PCA 方法、Select Minimal 方法、Laplacian 方法和 Wavelet 方法，这些方法使用 imagefuse 工具实现。因为最新的一些方法无法得到其运行结果图，故在此无法进行比较。

（a）均值融合方法

（b）PCA 方法

（c）Select Minimal 方法

（d）Lapalacian 方法

（e）Wavelet 方法

（f）AMG 方法

图 4.50　AMG 方法与其他方法的比较

从图 4.50 中可以看出，均值融合方法将能够提高模糊部分的清晰度，但其代价是丢弃了局部信息和部分边缘的信息。Select Minimal 方法选择两幅图像中像素灰度值小的那个来进行融合，减小了图像的对比度。PCA 方法和均值融合方法都是对两幅原始图像进行线性组合，只是权值的设置有所不同，其结果是丢失了部分原始信息。Lapacian 方法和小波融合方法在变换域进行融合，其关键是融合规则的设计，不同的融合规则有不同的作用，但是都存在变换域丢失信息的情况，如小波方法可能丢失图像的高频信号。AMG 方法的实质是通过代数

多重网格方法将清晰块提取出来，使用原始图像来进行替换，没有损失原始图像，而且能极为精确地将图像中清晰部分的图像检测出来；同时对图像分块之间进行梯度检测，做到梯度最小化，保证了像素或者块之间的连续性，避免了不一致的结果。

我们知道在原始图像 ClockA 中，右边的闹钟比较清晰，而在 ClockB 中左边的闹钟比较清晰，因此可以考虑将结果图与这两个图求差，然后显示两者的差异情况。通过这种方式来对算法有一个较为清楚的主观判断，结果如图 4.51 所示。

（a）均值融合结果与 ClockA 的差值图

（b）均值融合结果与 ClockB 的差值图

（c）小波融合结果与 ClockA 的差值图

（d）小波融合结果与 ClockB 的差值图

（e）AMG 融合结果与 ClockA 的差值图

（f）AMG 融合结果与 ClockB 的差值图

图 4.51　代数多重网格方法与其他方法的比较

在均值融合图中，可以发现在差值图中有明显的钟表结构，可见信息丢失较为严重。而在小波融合中，小波融合结果与 ClockA 之间有较大的差值，说明小波融合方法对于清晰的部分出现了误检，丢失了原始的正确信息。而在代数多重网格方法中，因为针对清晰部分和不清晰部分进行了明显的界定，没有丢失重要的信息。如图 4.52 是 DiskA 与 DiskB 的融合结果，并与其他算法进行了比较。

从图 4.52 的结果中可以看出，AMG 方法在大块图片特征检测时具有较好的效果，但是当清晰和模糊的边缘不在分块的边界时，会产生一些局部的小块的模糊，这也是分块融合方法的缺点。但是 AMG 方法能将图中绝大部分的清晰部分保留在融合结果中，因为两个源图像中钟表的左侧边缘之间存在一定的偏移，使用小波、Lapalacian 等方法会出现两个边缘之间的明显界限，而均值融合方法虽然没有明显的界限，但是结果会变得很模糊，而 AMG 算法能够较好地得到钟表清晰的边缘部分，显示出了 AMG 算法较好的检测效果。

（a）均值融合方法

（b）PCA 方法

（c）Select Minimal 方法

（d）Laplacian 方法

（e）Wavelet 方法

（f）AMG 方法（策略 1）

图 4.52　Disk 图片的融合结果

（g）AMG 方法（策略 2）

（h）AMG 方法（策略 3）

图 4.52　Disk 图片的融合结果（续）

2．清晰图片和构造的模糊图片融合的结果

为了更好地衡量图像的融合效果，先对清晰图片进行模糊，然后将清晰图片和模糊图片一起作为输入源，来进行融合，结果如图 4.53 所示。

（a）原始图像

（b）模糊图像

（c）均值融合方法

（d）PCA 方法

（e）Select Minimal 方法

（f）Lapalacian 方法

（g）Wavelet 方法

（h）本节方法

图 4.53　代数多重网格方法与其他方法的比较

4.7.3　图像融合效果综合评价方法（客观评价）

一种评价体系是综合多个评价参数，如果某个融合结果在这个体系中的绝大部分因素上占有优势，那么这种融合结果就是有效的。比如使用加权求和法、D-S 证据理论、粗糙集、模糊集等方法来进行综合评价。另一种评价体系是将融合结果用来进行后续处理，如图像分割、物体识别等，如果能从中发现更多有效的结果，那也可以证明该方法是有效的。

使用 AMG 融合方法对于 ClockA 和 ClockB 融合的结果与其他算法之间做了一些对比。

各种参数的结果比较如图 4.54～图 4.56 所示，计算结果如表 4.5 所示。

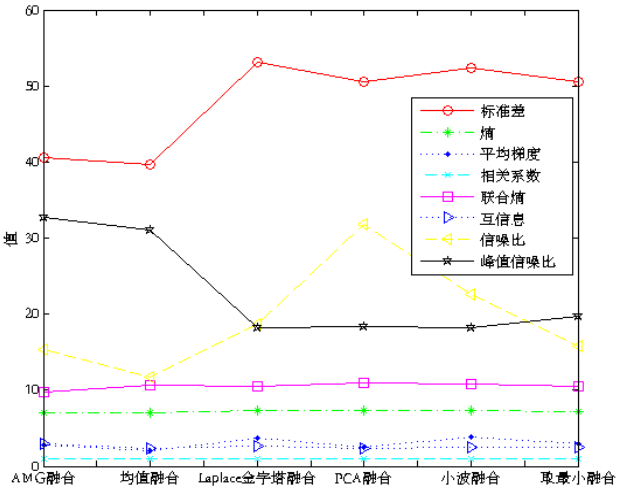


图 4.54　越大越好型的结果比较图

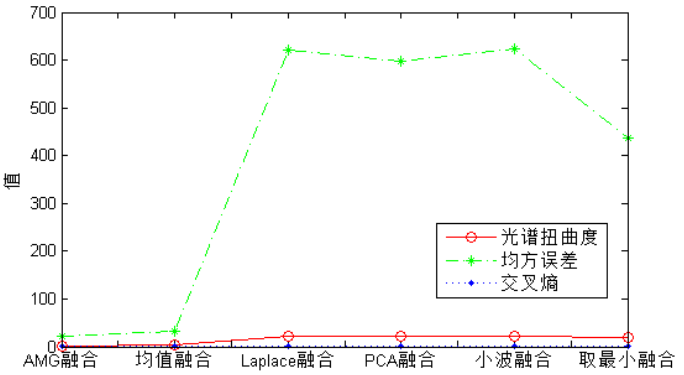


图 4.55　越小越好型的结果比较图

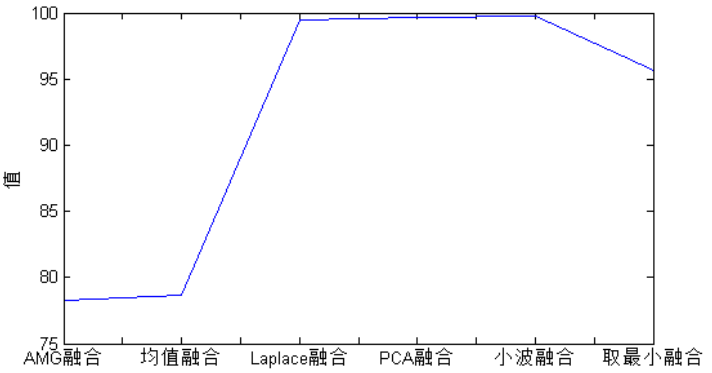


图 4.56　均值比较结果图

表 4.5　与其他算法的对比结果

评价指标		AMG	Average	Laplace	PCA	Wavelet	Min	标准
越大越好型	标准差	40.61	39.71	53.17	52.36	50.58	50.54	41.00
	熵	6.98	6.96	7.39	7.40	7.29	7.24	7.06
	平均梯度	2.83	2.00	3.73	3.82	2.52	2.98	2.83
	相关系数	0.9935	0.9907	0.9964	0.9947	0.9906	0.9906	—
	联合熵	9.75	10.58	10.48	10.84	10.93	10.57	—
	互信息	2.94	2.36	2.70	2.47	2.35	2.53	—
	信噪比	15.36	11.69	18.64	22.59	31.77	15.76	—
	峰值信噪比	32.73	31.04	18.17	18.16	18.34	19.69	—
越小越好型	光谱扭曲度	1.72	3.13	21.74	21.88	21.66	18.19	—
	均方误差	21.75	32.15	621.66	623.4	597.6	437.8	—
	交叉熵	0.0371	0.0792	0.4594	0.4263	0.4623	0.3968	—
适当型	均值	78.25	78.61	99.47	99.79	99.68	95.62	78.08

通过表 4.5 可以看出，AMG 方法在越小越好型中表现得很好，在越大越好型的峰值信噪比的结果中也表现得很好；在熵方面表现的不是很好，因为熵是信息量的表示，熵越大信息量越大。因此我们计算了标准图像的熵，其中与我们的算法结果最接近，有些算法超过了标准图像的熵，这也从另一方面说明本文的算法是对标准图像的一个接近，而没有对原始图像的空间或频率结果进行修改。

1．模糊图片之间的融合结果比较

最近还有一些新的方法，因为没有得到相应的结果图像，只能从文献中获得相关的数据来进行比较。表 4.6 为与当前出版的一些期刊和会议上的论文对同幅图片的处理结果进行比较，从比较结果来看，AMG 方法中 PSNR 和互信息最大，MSE 和交叉熵最小，熵与标准图像的熵最接近，既没有增加也没有减少图像的信息量，这也从一定程度上说明了本方法的优越性。

表 4.6　与最近文献结果的比较

	MSE	PSNR	熵	交叉熵	互信息
AMG 融合方法	21.75	32.73	6.98	0.0371	2.94
基于像素的小波 2009	106.27	27.87	—	—	—
基于能量的小波 2009	85.92	28.79	—	—	—
基于混合的小波 2009	67.97	29.81	—	—	—
加权非负矩阵分解 2011	—	—	7.3654	1.1132	—
小波变换和邻域特征 2011	43.4	31.751	7.448	—	—
冗余小波变换 2010	—	—	—	—	2.63

2．清晰图片和构造的模糊图片融合结果的比较结果

这里对清晰图片和模糊图片之间的融合做了一些对比。各种参数的结果如图 4.57～图 4.59 所示，计算结果如表 4.7 所示。

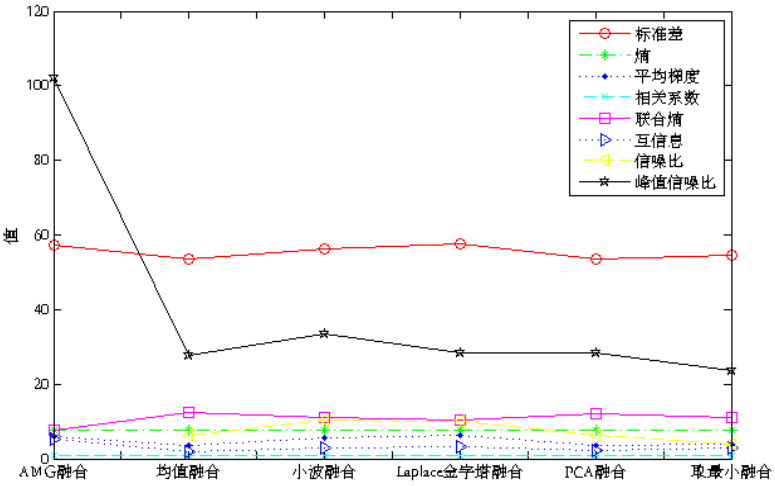


图 4.57　越大越好型的结果比较图

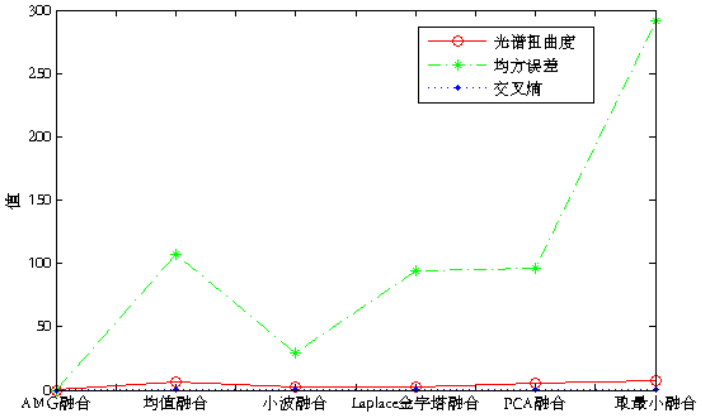


图 4.58　越小越好型的结果比较图

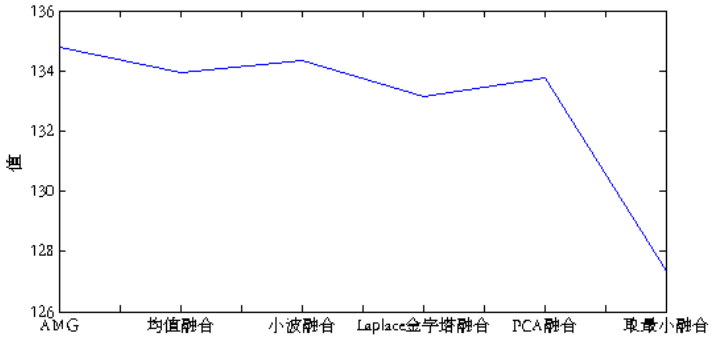


图 4.59　均值比较结果图

表 4.7　与其他算法的对比结果

评价指标		AMG	Average	Wavelet	Laplace	PCA	Min
越大越好型	标准差	57.16	53.52	56.41	57.61	53.68	54.46
	熵	7.60	7.52	7.60	7.63	7.52	7.53
	平均梯度	5.89	3.51	5.77	6.13	3.64	3.88
	相关系数	1	0.9847	0.9956	0.9862	0.9864	0.9633
	联合熵	7.60	12.26	11.05	10.53	12.14	10.93
	互信息	5.27	1.98	2.87	3.25	2.07	2.91
	信噪比	无穷	6.20	10.48	10.05	6.29	3.79
	峰值信噪比	102.3	27.82	33.46	28.40	28.28	23.48
越小越好型	光谱扭曲度	0	6.32	2.61	2.96	5.92	7.44
	均方误差	0	107.37	29.31	93.98	96.61	291.28
	交叉熵	0	0.05	0.006	0.01	0.04	0.03
适当型	均值	134.80	133.95	134.35	133.16	133.76	127.36

因为根据清晰图片和模糊图片进行图像融合，所以本算法得到的是完美的重建结果。但并不是所有的指标体系都能反映出这一点，如熵等。而在上述两种对比中，越小越好型指标的同向性比较好。因此在评价某种算法时，单一的指标体系是无法正确反映算法性能的，有些指标相关性较好，有些指标之间相互的关系很复杂，因此就需要通过某些复杂的推断来进行综合推理。而且有些指标还按照设计者的要求改变了图像的内在结构，不能反映图像的真实性；还有一些指标必须有标准图的参与才能计算，而标准图的获得往往是很难的，甚至是不可能的，因此这些指标的存在就失去了真正的意义。所以在构建指标体系时，还有很多东西需要研究。

4.8　本章小结

根据代数多重网格方法对清晰度的检测能力，本章将代数多重网格方法用于图像融合之中。区域越大，检测效果越稳定，但是如果一个模糊图像块和清晰图像块共存于一个区域时，该检测结果会出现一些误判的情况。因此如何界定区域是一个较大的问题。本章分别用区域、隶属度、边缘检测和寻找 MSE 极值点的方法进行了尝试，实验结果证明，只要能较好地确定区域的边界，代数多重网格方法都能取得较好的融合效果。

第 5 章　代数多重网格方法在物体识别中的应用

5.1　图像检索概述

近年来，由于人们越来越多的实际应用，如建筑设计、博物馆管理、教育和织物设计中对多媒体信息处理日益增加的需求，使得图像检索成为当前最活跃的研究方向之一[78-83]。此外，因为个人图像记录和互联网图像记录爆炸性地增长，开发一个能够有效地浏览和查询图像的检索工具越来越受到重视。对所有图像手动添加文本注释不仅冗长乏味，而且不切实际，一张图胜过千言万语。另外，目前的技术还不能自动添加图像注释。因此，基于内容的图像检索获得了广泛关注。基于内容的图像检索技术就是能在图像数据库中检索图像，这样检索到的图像和用户提供的待查询图像间存在一定的语义关系。基于内容的图像检索的目的是用一组低层视觉特征如颜色、纹理等信息来概念性地表示图像。图 5.1 是图像特征分布。

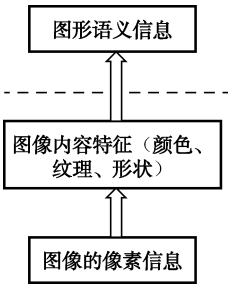


图5.1　图像特征分布

目前，基于内容的图像检索系统（Content Based Image Retrieval，CBIR）已成为图像检索系统的中流砥柱。其基本流程如图 5.2 所示。

由图 5.2 可以看出，CBIR 系统先对示例图像（图像数据库）进行特征提取，主要是图像的颜色、纹理和形状等特征，然后以此建立特征向量（特征数据库），最后就是在图像数据

库中查找与待检索图像最为相似的图像，即相似性度量值最大的图像。

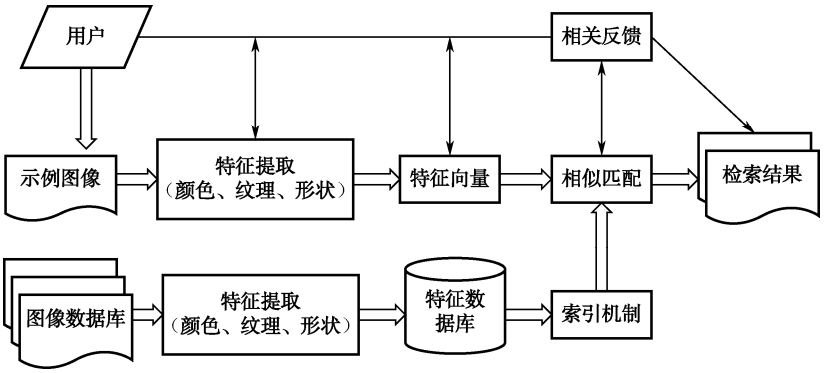


图 5.2　基于内容的图像检索的基本流程

5.1.1　清晰度评价算法在图像检索中的应用

能够对图像数据库进行快速、高效的基于内容的图像检索技术是图像检索中非常重要的组成部分。如何表述图像的内容是基于内容的图像检索中的关键技术。

先对图像的内容表述之后，然后再对内容进行相似度计算就可以得到用户请求的结果。对于图像内容的表述是通过图像特征来表达的。在基于内容的图像检索系统中，能够快速、准确地对图像内容进行描述是至关重要的。现有检索系统大多采用图像颜色、纹理、形状特征和空间等第二层次的物理特征来描述图像内容[84]。

然而，人们在判断图像相似性的时候大多考虑的是把自己感兴趣的那部分图像作为检索的条件[85]，而不是像计算机计算整体图像的特征、纹理等信息来检索图像，这就需要先寻找图像中的感兴趣区域，以感兴趣区域作为关键图对目标图像库进行检索。与此同时，随着数码相机性能的提高，人们获取图片时往往选择对自己感兴趣的部分进行聚焦显示，不感兴趣的部分就会成为散焦部分，因为图像聚焦的内容清晰些，所以感兴趣区域就取图像中的清晰部分。将这一情况引入到图像检索中，有以下两点应用。

（1）在图像检索前先对待检索图像做预处理，提取检索图像的清晰区域，再用提取的清晰区域作为关键图。

（2）对关键图做低清晰度的低维度处理，可以有效地节省索引存储空间。

5.1.2　图像中清晰度区域提取

在聚焦图像中，图像的重要信息是图像中的聚焦部分，也是图像中清晰的部分。首先提取这部分图像作为图像检索时的索引图像，这样可以更加关注在感兴趣的区域检索，减少误检率。

提取图像清晰度区域采用分块搜索的策略，寻找分块中清晰度值最大的区域作为关键图。以此为实验依据，对部分聚焦图像进行清晰度区域提取实验，图 5.3 中，分别对 4 幅聚焦图像进行清晰度区域的提取。

（a）picture1

（b）picture1 的清晰区域

（c）picture2

（d）picture2 清晰区域

（e）picture3

（f）picture3 清晰区域

（g）picture4

（h）picture4 清晰区域

图 5.3　聚焦图像清晰区域提取

由于本节中使用基于代数多重网格的图像清晰度评价方法来评价图像区域的清晰度，目

前程序主要考虑灰度图像，因此本节中用于检索的关键图都使用灰度图像（彩色图像的处理机制相同）。

图 5.3 中针对每一幅待检索图像，为了能够达到既降低检索索引存储空间又能够保留图像基本信息，我们提取原始图像约 1/4 的区域作为关键图。

对关键图做降维处理可以有效地节省存储空间，实验采用两种方法分别对关键图实现降维处理，分别是代数多重网格法和拉普拉斯投影矩阵变换法。

5.1.3　代数多重网格法降低维度

因为代数多重网格在求解过程中对细网格的粗网格提取算法能够实现对原始图像的稀疏表示，该方法有效地降低了图像的维数，实现了数据的高效存储，所以实验使用代数多重网格法实现关键图的降维处理。实验首先对图 5.3 中 picture1 提取的清晰度区域加不同的高斯模糊，产生不同程度的高斯模糊图像，然后对所有的图像进行代数多重网格的重构，得到重构后的一系列图像，重构后的图像如图 5.4 所示。

（a）关键图重构

（b）fspecial（‘gaussian’,4,2）的模糊图像重构结果 Blur1

（c）fspecial（‘gaussian’,8,4）的模糊图像重构结果 Blur2

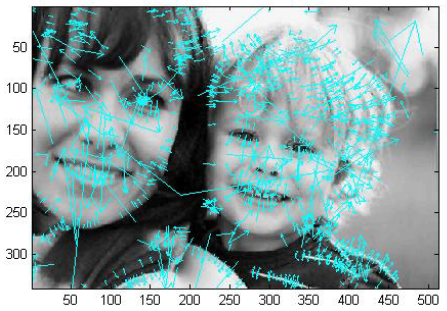
（d）fspecial（‘gaussian’,12,6）的模糊图像重构结果 Blur3

（e）fspecial（‘gaussian’,16,8）的模糊图像重构结果 Blur4

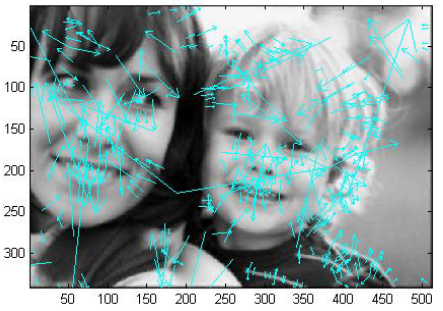
（f）fspecial（‘gaussian’,20,10）的模糊图像重构结果 Blur5

图 5.4　代数多重网格重构不同高斯模糊图像

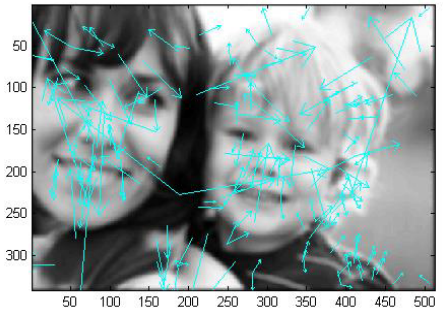
基于内容的图像检索在查询图像时，大多采用图像的特征来描述物体，也就是说图像检索时，特征匹配越多，图像间的相似性越大，则越容易被检索到。为了进一步验证降维处理后的关键图可以作为有效的检索索引，图 5.5 和图 5.6 分别给出了使用代数多重网格重构图像和小波重构图像[86]的 SIFT 特征向量检测结果。图 5.7 和图 5.8 分别给出了使用代数多重网格重构图像和小波重构图像的特征向量匹配的结果。表 5.1 和表 5.2 分别统计了代数多重网格重构图像和小波重构图像检测出的 SIFT 特征向量的个数。由表中数据可以看出，除了输入 Blur3 时小波重构图像检测出的 SIFT 特征向量的数量比代数多重网格重构的图像多以外，对于其他图像，代数多重网格重构的效果都要比小波重构的效果好。表 5.3 和表 5.4 分别统计了代数多重网格重构图像和小波重构图像 SIFT 特征向量匹配的个数。由表中数据可以看出，代数多重网格重构的图像比小波重构后的图像 SIFT 特征向量匹配的个数多。从实验结果来看，代数多重网格重构的图像效果要略优于小波重构的图像。



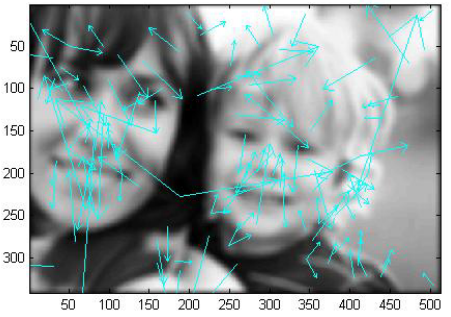
（a）关键图特征向量



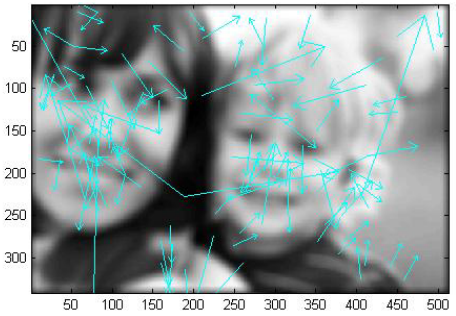
（b）Blur1 特征向量



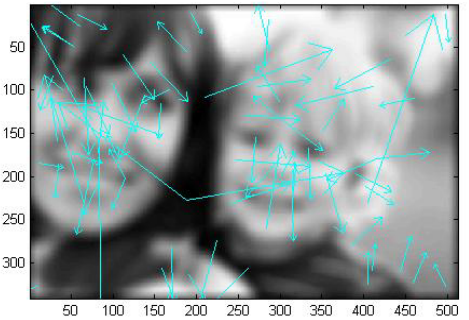
（c）Blur2 特征向量



（d）Blur3 特征向量

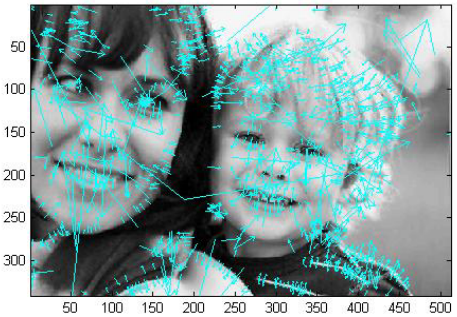


（e）Blur4 特征向量

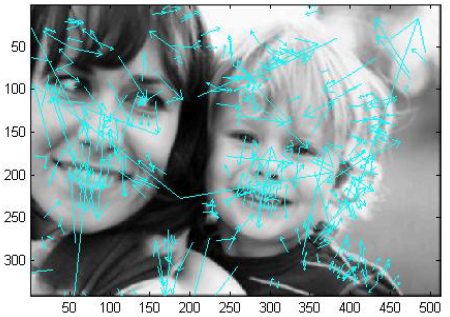


（f）Blur5 特征向量

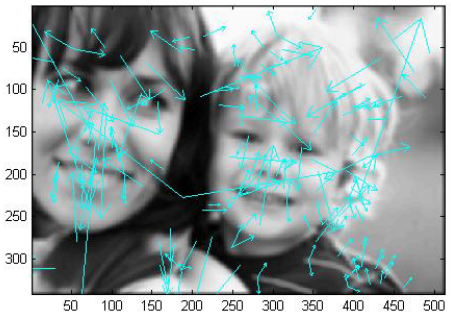
图 5.5　代数多重网格 SIFT 特征向量图



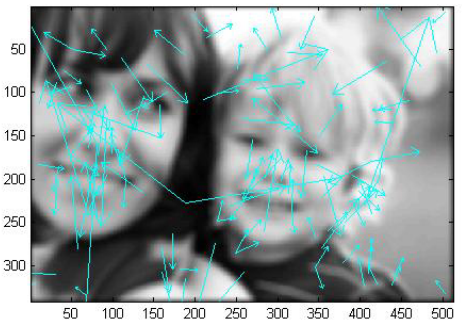
（a）关键图特征向量



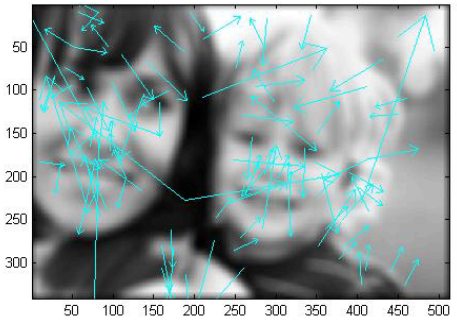
（b）Blur1 特征向量



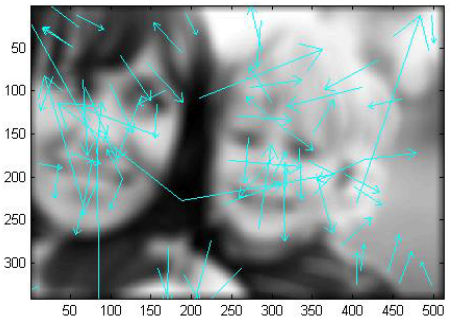
（c）Blur2 特征向量



（d）Blur3 特征向量

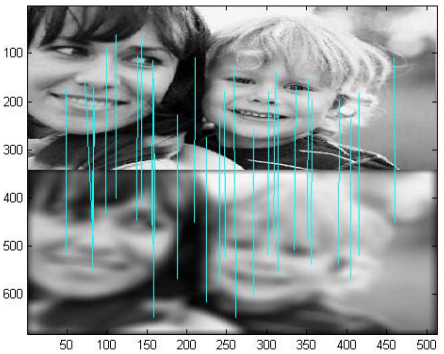


（e）Blur4 特征向量

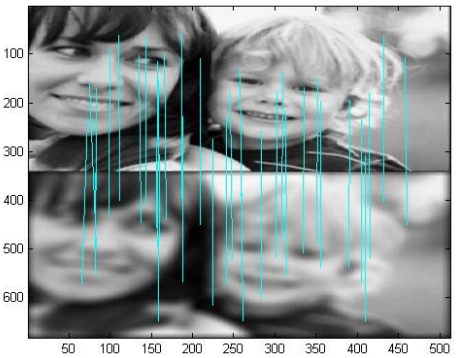


（f）Blur5 特征向量

图 5.6　小波重构 SIFT 特征向量图

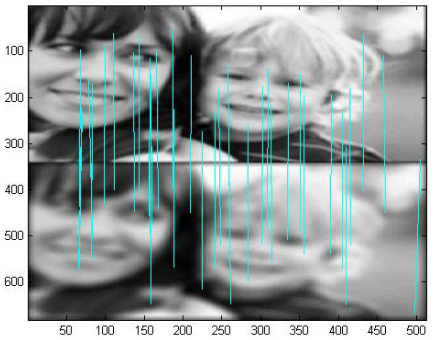


（a）关键图、Blur5 匹配结果

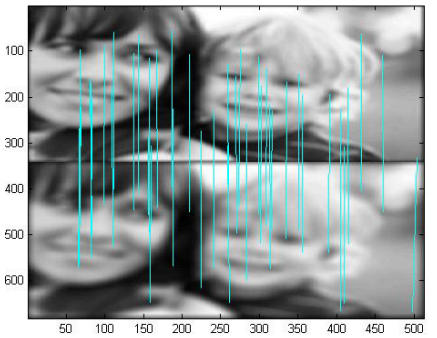


（b）Blur1、Blur5 匹配结果

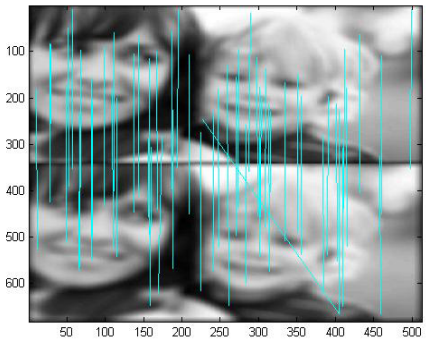
图 5.7　代数多重网格 SIFT 特征向量匹配结果



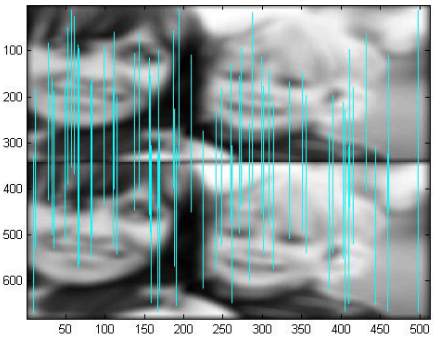
（c）Blur2、Blur5 匹配结果



（d）Blur3、Blur5 匹配结果

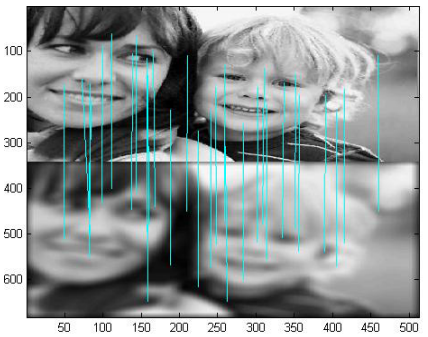


（e）Blur4、Blur5 匹配结果

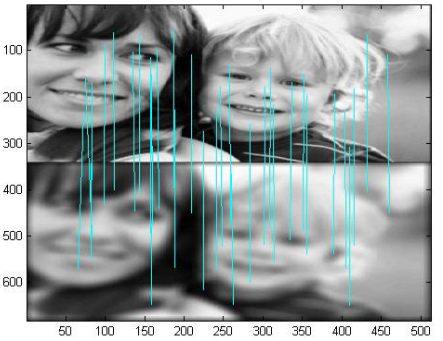


（f）Blur5、Blur5 匹配结果

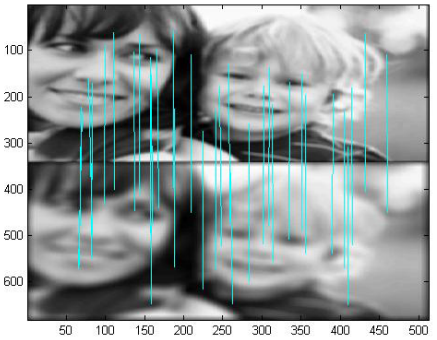
图 5.7　代数多重网格 SIFT 特征向量匹配结果（续）



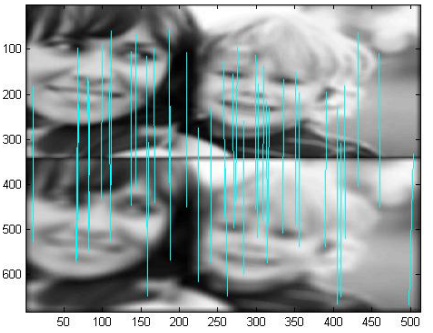
（a）关键图、Blur5 匹配结果



（b）Blur1、Blur5 匹配结果

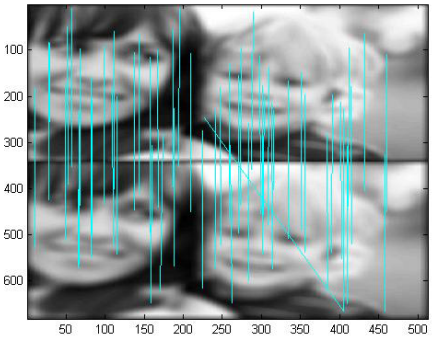


（c）Blur2、Blur5 匹配结果

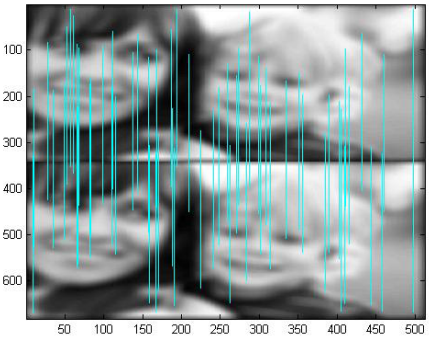


（d）Blur3、Blur5 匹配结果

图 5.8　小波重构 SIFT 特征向量匹配结果



（e）Blur4、Blur5 匹配结果



（f）Blur5、Blur5 匹配结果

图 5.8　小波重构 SIFT 特征向量匹配结果（续）

表 5.1　代数多重网格 SIFT 特征检测结果

AMG 重构图像	关键图	Blur1	Blur2	Blur3	Blur4	Blur5
SIFT 特征向量（个）	993	355	166	116	97	74

表 5.2　小波重构 SIFT 特征检测结果

AMG 重构图像	关键图	Blur1	Blur2	Blur3	Blur4	Blur5
SIFT 特征向量（个）	993	354	164	118	97	73

由图 5.5 可以看出，图像越清晰时，检测到的 SIFT 特征向量越多，而图像比较模糊时，只能检测到很少的特征向量。

特征向量匹配在图 5.7 和图 5.8 中以匹配线的形式给出，只是对特征向量匹配的点坐标连线，没有考虑向量的方向，所以对于特征向量像素坐标相同而方向不同的点之间的匹配线是重合的。但从图 5.7 和图 5.8 中可以看出，Blur1、Blur5 匹配的特征数最少，Blur5 和它自身匹配的特征数最多，由此可见，图像内容相同时，图像清晰度越接近，图像特征匹配数越多，图像相似度越大，越容易检索到。

表 5.3 中是 AMG 重构后不同清晰度图像之间的匹配结果，其中每幅图像与它本身的匹配特征数最多，另外模糊程度相近的图像之间匹配特征数也相对较多。

表 5.3　代数多重网格 SIFT 特征匹配结果

AMG 重构图像	关键图	Blur1	Blur2	Blur3	Blur4	Blur5
SIFT 特征向量（个）	35	37	41	47	65	74

表 5.4　小波重构 SIFT 特征匹配结果

AMG 重构图像	关键图	Blur1	Blur2	Blur3	Blur4	Blur5
SIFT 特征向量（个）	35	36	39	48	64	73

5.1.4 拉普拉斯投影矩阵变换法降低维度

选取 Mother 图像变换后的低频分量分别为 128 个、380 个、500 个、1120 个、2500 个系数重构的拉普拉斯图像如图 5.9 所示，6 幅图像的清晰度按从（a）～（f）的顺序依次降低。由于拉普拉斯矩阵重构时计算量很大，图像太大时内存无法满足需求，所以统一将图 5.9（b）的尺寸定为 100×100。

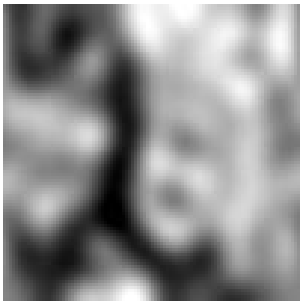
（a）原始图

（b）LMP 低频分量=2500 重构结果

（c）LMP 低频分量=1120 重构结果

（d）LMP 低频分量=500 重构结果

（e）LMP 低频分量=380 重构结果



（f）LMP 低频分量=128 重构结果

图 5.9　拉普拉斯重构结果

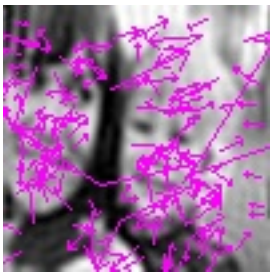
相对于 AMG 重构获取的多清晰度图像，我们在拉普拉斯矩阵重构中进一步取更模糊的图像来检测图像的特征。对重构后的图像采用 SIFT 特征检测来获取图像的 SIFT 特征向量，用于后续的特征匹配。

图 5.10 中显示的不同 LMP 低频分量重构后的图像的 SIFT 特征向量检测结果，每幅图对应的特征检测结果如表 5.5 所示。从表 5.5 中结果看，越清晰的图像，LMP 分量个数越多，

相对检测到的特征向量的个数也越多。图 5.11 中是不同 LMP 分量间的匹配结果。

（a）原始图

（b）LMP 低频分量=2500



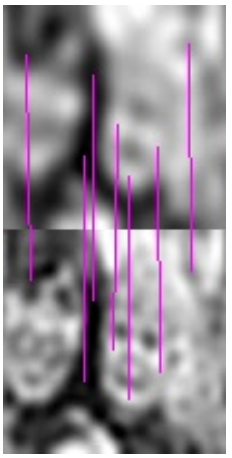
（c）LMP 低频分量=1120

（d）LMP 低频分量=500

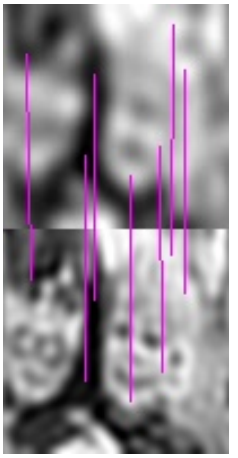
（e）LMP 低频分量=380

（f）LMP 低频分量=128

图 5.10　拉普拉斯重构后的图像 SIFT 特征向量检测结果



（a）LMP 低频分量=128 与 LMP 低频分量=380 的匹配结果



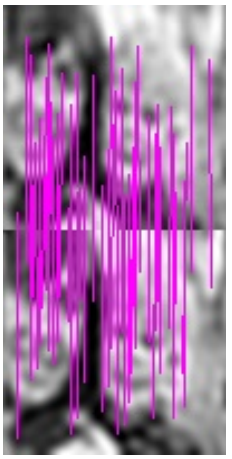
（b）LMP 低频分量=128 与 LMP 低频分量=500 的匹配结果

图 5.11　代数多重网格 SIFT 特征向量匹配结果

（c）LMP 低频分量=128 与 LMP 低频分量=1120 的匹配结果

（d）LMP 低频分量=128 与 LMP 低频分量=2500 的匹配结果

（e）LMP 低频分量=128 与原始图的匹配结果



（f）LMP 低频分量=1120 与 LMP 低频分量=2500 的匹配结果

图 5.11　代数多重网格 SIFT 特征向量匹配结果（续）

表 5.5　拉普拉斯投影变换后特征向量检测结果

LMP 分量（个）	128	380	500	1120	2500	原始图
SIFT 特征向量（个）	37	79	98	159	189	110

表 5.6 是不同 LMP 分量重构图之间的 SIFT 特征匹配结果。其中，每个 LMP 低维表示与它本身的匹配特征数最多，与 AMG 相同的结果是模糊程度相近的图像之间匹配特征数也相对较多。表中最后一行是不同数量 LMP 分量重构图像的 6 幅图像分别与包括它本身还有其他 5 幅图像的特征匹配数的求和，表 5.6 中对于 LMP 重构的图像，越清晰的图像，匹配特征总数越多。

表 5.6　不同 LMP 分量重构后匹配结果

LMP（个）	128	380	500	1120	2500	原始图
128	37	11	10	8	10	11
380	11	79	51	33	30	31

续表

LMP（个）	128	380	500	1120	2500	原始图
500	10	51	98	41	33	36
1120	8	33	41	159	62	52
2500	10	30	33	62	187	85
10000	11	31	36	52	85	108
匹配总数	87	235	269	355	407	323

5.2　图像检索实验结果及分析

5.2.1　图像检索系统介绍

将图像清晰度评价指标引入到图像检索中是为了能够在检索图像时更加关注人们感兴趣的聚焦部分的图像，同时有效降低关键图的存储空间。

检索系统使用 Fraunhofer Institute for Optronics，System Technology and Image Exploitation（IOSB）开发的基于内容的图像检索系统，简称 IOSB 系统。检索系统界面图如图 5.12 所示。实验检索数据库包含 101 幅网络上的随机图片，以图 5.13（b）图像中提取的聚焦区域图 5.13（a）作为关键图。

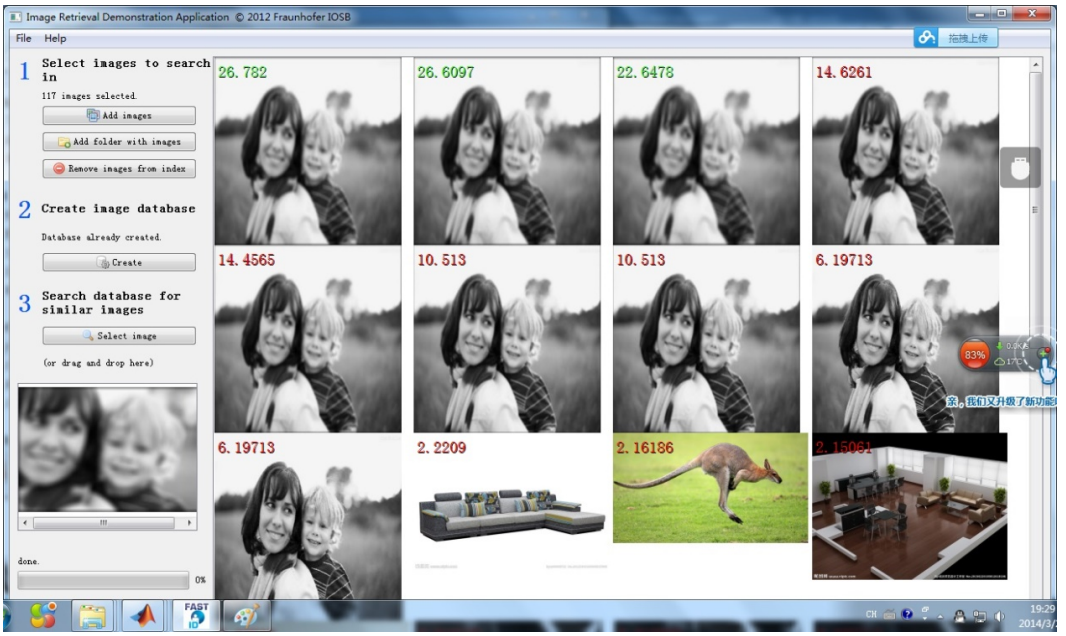


图 5.12　IOSB 检索系统

使用该系统的用户可根据自己的需要建立检索数据库和待检索的数据。因为前文中 AMG 重构、拉普拉斯矩阵重构和小波重构数据都是使用灰度图像，所以检索系统数据库中的图像也统一做灰度处理。

本节设计了三组实验来检验降低维数后的待检测关键图检索图像的准确性，即检索输出的相关图像数量与数据库中全部的相关图像数量。第一组实验检验降低维度后的图像是否能检索到数据库中的原始图像；第二组实验是检验检索的查全率，对于数据库中一组原始图像不同清晰度变化的图像是否都能检索到；第三组实验是检验对于降低维度后的关键图检索的稳健性，对数据库中的目标图像轻微变形后测试检索的查全率。

5.2.2 检索一幅图像

为了验证降低维度的关键图也能够有效检索到目标图像，实验分别使用 AMG 和 LMP 对关键图做多清晰度表征，检索数据如图 5.13 所示，将待检索图像图 5.13（b）添加到检索数据库中，检索结果如表 5.7、表 5.8 所示，表 5.7 中图像（a）～（f）对应图 5.4（a）～（f），表 8 中图像（a）～（f）对应图 5.9（a）～（f）。

（a）关键图像

（b）目标图像

图 5.13　检索数据

表 5.7　AMG 重构多清晰度图像检索结果

AMG	（a）	（b）	（c）	（d）	（e）	（f）
相关度	2675.8	1251.4	164.8	48.9	24.1	5.4

表 5.8　LMP 清晰度图像检索结果

LMP	（a）	（b）	（c）	（d）	（e）	（f）
相关度	66.7	112.7	72.3	6.1	4.2	2.5

表 5.7 和表 5.8 给出了不同清晰度的低维表示图像的检索结果，并列出了关键图像与目标图像的相关度，可以看出，关键图像越清晰，与原始图像的相关度越大，对于不同算法的低维表示图像均可以检索到目标图像。表 5.7 中的相关度与表 5.8 相比偏大，主要是因为在 AMG 降低维度时参数选择的问题，重构后的图像比较清晰，所以与原始图像有更好的相关度。但是两种方法都能检索出目标图像。

5.2.3 检索序列图像

图像检索中，希望所有目标图像都能够被检索出来，即保证图像检索的查全率较高。在将降低维度后的图像作为关键图像进行检索时，实验给出目标图像的 6 个模拟帧序列的连续清晰度值的图像，图 5.14 给出 6 幅重构的图像，实验使用 IOSB 软件进行图像检索，分别使用了通

过 AMG 方法重建、通过小波变换方法重建和 LMP 重建的不同高斯模糊的 Mother 图像以及 http://wang.ist.psu.edu/docs/related.shtml上下载的 1000 张图像组成了图像搜索库。因为 Mother 图片是灰度图片，所以首先将 1000幅图片处理成灰度图片，然后进行检索。接着使用不同模糊的图像（原始图像，Blur1 到 Blur5）为待检索图像，该系统会从刚才的数据库中寻找出 25 幅最接近待搜索图片的图像，实验结果如图 5.15～图 5.17 所示，表 5.9～表 5.11 分别列出了三种算法检索 6 幅图像的连续清晰度序列图像的结果。由实验结果我们可以看出，使用代数多重网格重建的图像，6 幅图中用不同高斯模糊的图像都可以搜索出所有的正确图片，查全率和准确度均为 100%；而使用 LMP 重建的图像，当待检索的图像是 Blur3 时，只检索出了 5 幅图像，


（a）原始图 AMG 重构


（b）fspecial（‘gaussian’,4,4）的模糊 AMG 重构结果


（c）fspecial（‘gaussian’,8,8）的模糊 AMG 重构结果


（d）fspecial（‘gaussian’,12,12）的模糊 AMG 重构结果

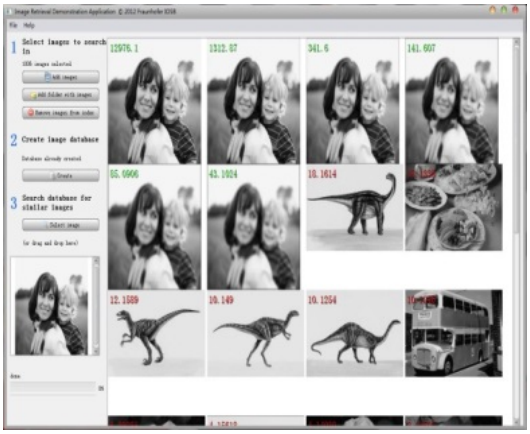

（e）fspecial（‘gaussian’,16,16）的模糊 AMG 重构结果


（f）fspecial（‘gaussian’,20,20）的模糊 AMG 重构结果

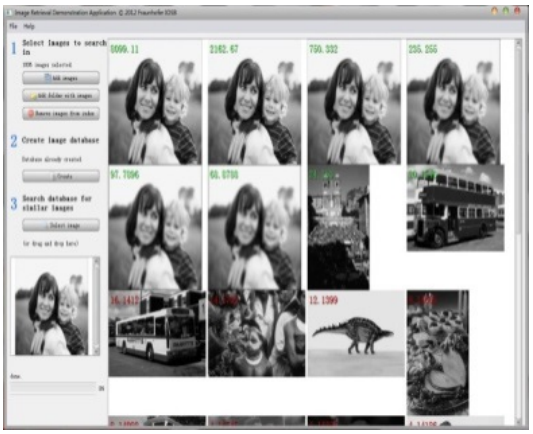
图 5.14　AMG 重构清晰度值连续序列图像

没有完全检索出所有的图像；使用小波重建的图像，当待检索的图像是 Mother 和 Blur1 时，只检索出了 5 幅图像。图 5.18、图 5.19 分别统计了待检索图像检索出的结果按检索距离的大小进行排序的相关值，本节主要考虑其稳定性，因此图 5.18 和图 5.19 中选择 Mother 图片及其模糊化的图片和第一个非该序列的图片。由图 5.18 统计结果可知，当待检索图片的模糊度越高时，检索出的正确图片的相关值越来越接近，越来越稳定；其中 Blur3 在第六到第七的跳变最大，即区分同一场景物体和不同场景物体的能力最强。从图 5.19 中可以看到，使用相同的待检索图像，小波重建后的图像检索出的图像的相关值明显比代数多重网格重建的图像的相关值小，即小波在图像的重建过程中不能很好地提取出图像的关键特征（区别于其他图像的本质特征）。从以上实验结果可以得出，使用代数多重网格重建的图像保留了图像最本质的特征，而去掉区分度不大的特征，我们只需要保存和传输某一个模糊程度图像的影像，这样可以减少网络传输的信息和存储信息的空间，保证同类场景物体的识别精度，提高不同场景物体的区分度。

图 5.15　LMP（LMP 低频分量=128）检索数据

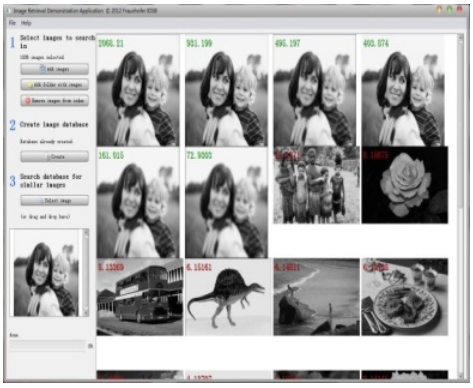


（a）Mothe 为待检索图像的搜索结果

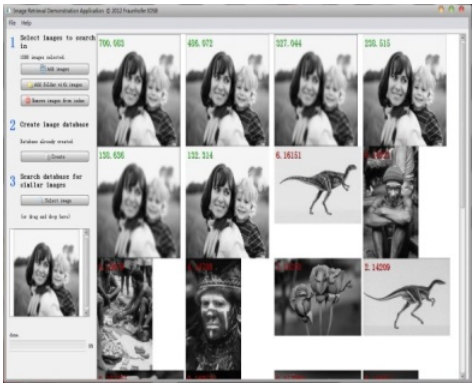


（b）Blur1 为待检索图像的搜索结果

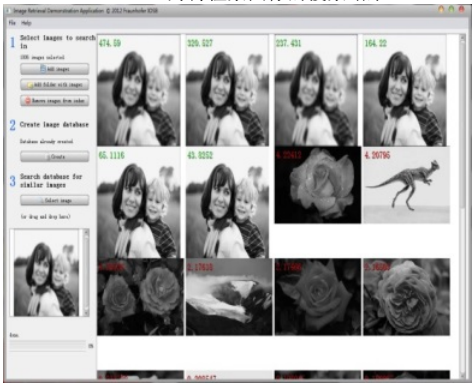
图 5.16　使用对模糊图像进行 AMG 重建的结果进行图像检索



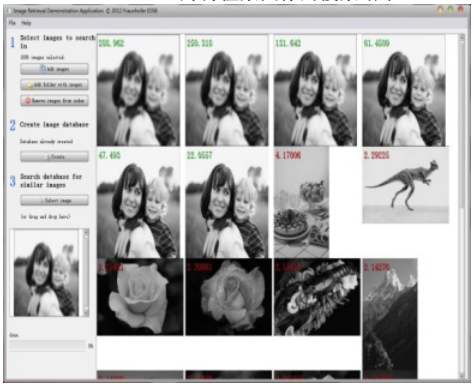
（c）Blur2 为待检索图像的搜索结果



（d）Blur3 为待检索图像的搜索结果



（e）Blur4 为待检索图像的搜索结果

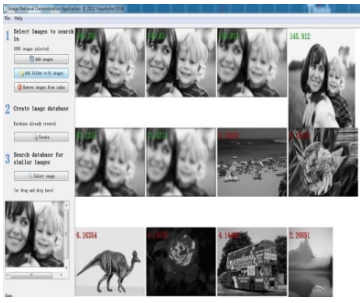


（f）Blur5 为待检索图像的搜索结果

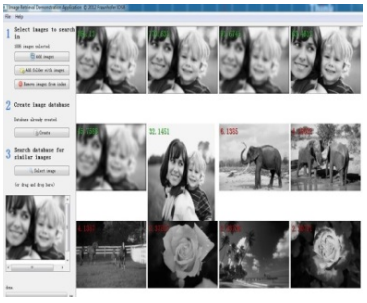
图 5.16　使用对模糊图像进行 AMG 重建的结果进行图像检索（续）

（a）Mother 为待检索图像的搜索结果

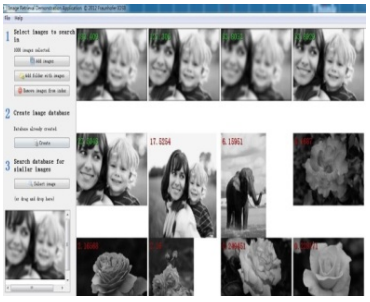
（b）Blur1 为待检索图像的搜索结果



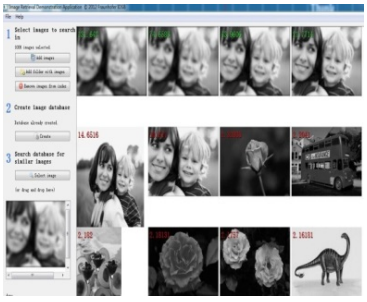
（c）Blur2 为待检索图像的搜索结果



（d）Blur3 为待检索图像的搜索结果



（e）Blur4 为待检索图像的搜索结果



（f）Blur5 为待检索图像的搜索结果

图 5.17　使用对模糊图像进行小波重建的结果进行图像检索

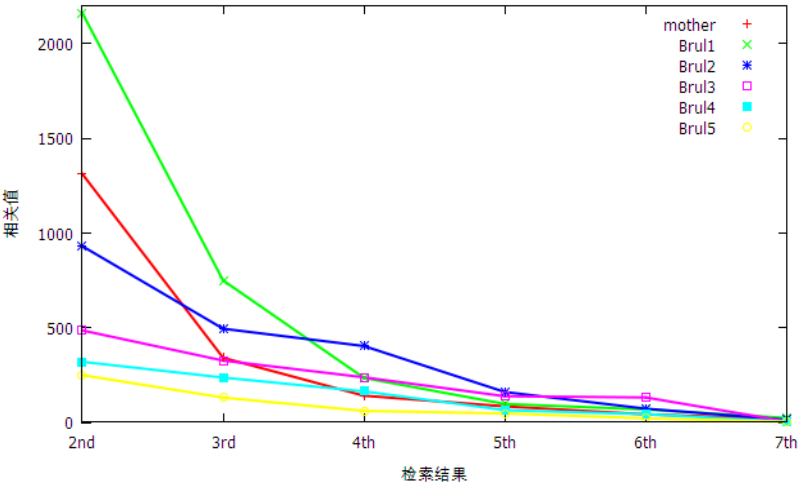


图 5.18　使用 AMG 重建不同高斯模糊的待检索图像的检索结果

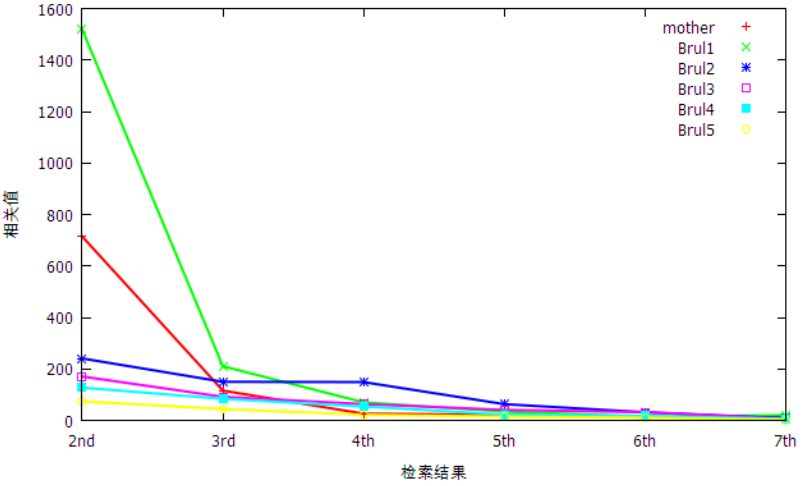


图 5.19　使用小波重建不同高斯模糊的待检索图像检索结果

表 5.9　AMG 算法检索结果

AMG 重构图像	(a)	(b)	(c)	(d)	(e)	(f)
检索结果（个）	6	6	6	6	6	6

表 5.10　LMP 算法检索结果

LMP 重构图像	(a)	(b)	(c)	(d)	(e)	(f)
检索结果（个）	6	6	6	5	6	6

表 5.11　小波算法检索结果

小波重构图像	(a)	(b)	(c)	(d)	(e)	(f)
检索结果（个）	5	5	6	6	6	6

5.3　代数多重网格方法在物体识别中的应用

通俗地讲，物体识别就是使计算机能够像人一样，对任意环境下观察到的任意物体都具有分类、识别的能力。然而，人类的这种看似简单的能力事实上蕴藏着非常复杂的细节，而且到目前为止，人们都还没有完全了解这些内部的细节，也没有研究出可以与人类视觉相匹敌的物体识别系统。

物体识别技术是人类视觉的延伸，它涉及神经学、认知科学、人工智能等多个学科知识。在计算机技术与人工智能技术日益发展的今天，物体识别技术越来越成为人工智能的基础技术，更多地体现为机器智能，其基本分析工具也随着数学工具的不断进步而不断发展。物体识别的应用范围也越来越广泛，无论是用于军事方面的目标识别、目标跟踪等领域，还是在民用方面的智能监控、多媒体检索以及各类身份认证系统，这种由机器自动识别物体的技术是所有以图像或视频作为输入的实际应用系统的核心技术，这些系统的性能均依赖于物体识别所能达到的水平。

由于自然图像中可能出现的物体种类繁多，千差万别，在不同视角与不同尺度下，同一类物体的表现形式千姿百态，不同类别的物体在某些时候又表现出很大的相似性，那么对于人类可以轻松识别的自然图像，如何让机器在给定有限训练样本的前提下，自动、有效地识别出新物体仍然是物体识别研究的主题。

目前，物体识别仍是计算机视觉和模式识别领域最重要和最具挑战性的研究课题之一，也是认知科学研究的核心内容之一，是计算机视觉的“经典”问题，是机器视觉系统的基本功能模块。物体识别水平的提高无论在理论上还是在实践中都具有重要意义。在理论上，物体识别的研究有助于促进认知科学和人工智能等学科的发展；在实践中，物体识别水平的提高对于工业、医学、军事、交通等领域会带来极大影响，甚至会改变人们的生活方式。

5.4　物体识别的研究现状

5.4.1　问题描述

在计算机视觉和模式识别领域，一个典型的物体识别系统通常应包括以下三个主要过程[87]（见图 5.20）。

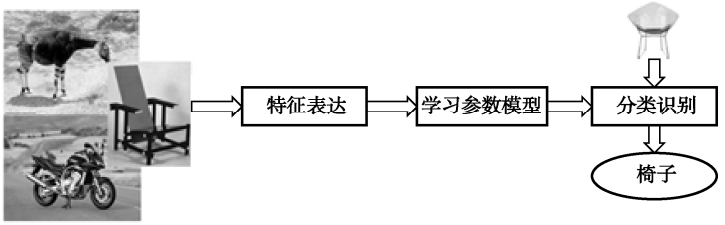


图 5.20　物体识别流程

（1）特征表达：在保证计算可行的前提下，如何提取最能表征物体类别的本质信息，忽略对分类作用不强的信息。

（2）学习参数模型：如何根据前述特征表达中形成的各类图像的特征来建立分类器模型。

（3）分类识别：将待识别物体输入到已经形成的学习参数模型中，进行分类识别。

5.4.2 物体识别难点

对于前面所述的物体识别过程，每一过程的执行都会影响到物体识别的效果。已有的研究大都是针对于特定的物体类别和表现形式，如汽车、车牌、人脸等，这样的系统无论在特征表达、模型学习还是在分类上都具有针对性，缺少通用性和可扩充性。以下具体介绍物体识别系统目前存在的困难。

（1）特征表达。特征表达是物体识别的第一步，特征表达的结果将直接影响到后续过程的执行，因此特征表达也是物体识别中最为关键的一步。理想情况下，提取的物体特征应该能反映各类物体的本质信息，对于同类图像的各种变化具有不变性，对于相似类别的物体又具有区分性。由于物体在图像中会以任意姿态和视角出现，系统无法预知具体的姿态和视角，且在不同尺度下，对识别起作用的特征也不同，光照变化和遮挡也是特征表达的一个难点。对于大规模的物体识别系统还应考虑特征共享和组合的问题。

（2）模型学习。在学习模型中，不得不考虑样本数的问题。对于大规模物体识别系统来讲，某些类别的物体类内变化大，但穷尽所有的物体进行训练又是不可能的，这就需要学习的模型可以基于小样本，同时又具有举一反三的能力。

（3）物体的客观标注。如何保证系统所采用的图像和视频是客观标准的一直是视觉中存在的问题。数据是物体识别系统进行特征表达、学习模型以及评价系统性能的基础，只有对数据进行客观的标注，才使得物体识别系统具有实际意义。

5.5 基于“词袋”模型的物体识别方法

5.5.1 文本“词袋”模型

Bag-of-words 模型[88,89]是信息检索领域常用的文档表示方法。在信息检索中，BOW 模型假定对于一个文档，忽略它的单词顺序和语法、句法等要素，将其仅仅看作是若干个词汇的集合，文档中每个单词的出现都是独立的，不依赖于其他单词是否出现。也就是说，文档中任意一个位置出现的任何单词，都是不受该文档语义影响而独立选择的。例如有如下两个文档：

1：Bob likes to play basketball，Jim likes too.

2：Bob also likes to play football games.

基于这两个文本文档，构造一个词典：

Dictionary = {1:”Bob”, 2. “like”, 3. “to”, 4. “play”, 5. “basketball”, 6. “also”, 7. “football”, 8. “games”, 9. “Jim”, 10. “too”}。

这个词典一共包含 10 个不同的单词，利用词典的索引号，上面两个文档每一个都可以

用一个 10 维向量表示（用整数数字 0～n（n 为正整数）表示某个单词在文档中出现的次数）：

1：[1, 2, 1, 1, 1, 0, 0, 0, 1, 1]

2：[1, 1, 1, 1 ,0, 1, 1, 1, 0, 0]

可以将其用直方图表示。在对文档识别的时候可以通过寻找两个直方图间的最短距离的方式来决定该文档归属于哪个类别。

5.5.2　图像“词袋”模型基本思想

常用的物体识别方法[90,91]包括：决策理论方法、句法方法、人工神经网络方法和“词袋”模型方法等。基于“词袋”模型的物体识别方法因为计算简单，对噪声、光照变化和局部遮挡更加鲁棒等特点引起了广泛关注。“词袋”模型物体识别方法如图 5.21 所示。

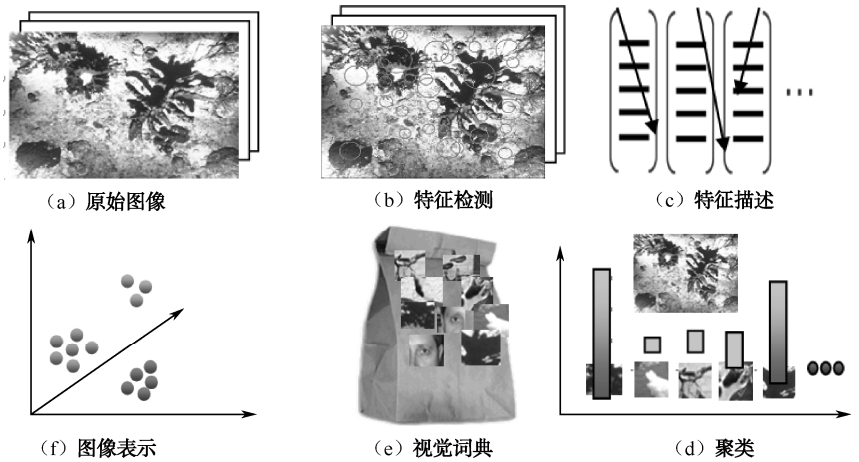


图 5.21　“词袋”模型物体识别方法

在信息检索中，BOW 模型将一个文档看作是若干个词汇的集合。由于图像中的词汇不像文本文档中的那样是现成的，我们首先需要从图像中提取出相互独立的视觉词汇，这通常需要经过三个步骤[92]：

第一步：利用 SIFT 算法从不同类别的图像中提取视觉词汇向量，这些向量代表的是图像中局部不变的特征点；

第二步：将所有特征点向量集合到一块，利用 k-Means 算法合并词义相近的视觉词汇，构造一个包含 k 个词汇的单词表；

第三步：统计单词表中每个单词在图像中出现的次数，从而将图像表示成为一个 k 维数值向量。

对于所有的输入图像（见图 5.21（a）），首先检测图像中具有某种不变性的点作为兴趣点（见图 5.21(b)），利用数学的方法对兴趣点周围的图像进行描述形成特征向量（见图 5.21(c)），通过聚类的方法对所有训练样本的特征向量进行量化（见图 5.21（d）），生成的聚类中心即为所谓的视觉词典（见图 5.21（e）），将图像的所有特征向量映射到不同的视觉单词，最后图像将表示为视觉单词的直方图（见图 5.21（f））。然后采用某种机器学习的方法对所有训练样本的直方图进行参数模型学习，最后进行分类识别。由于图像是用许多图像块组成的视觉词典

来表示的，故将其形象地称为“词袋”模型。

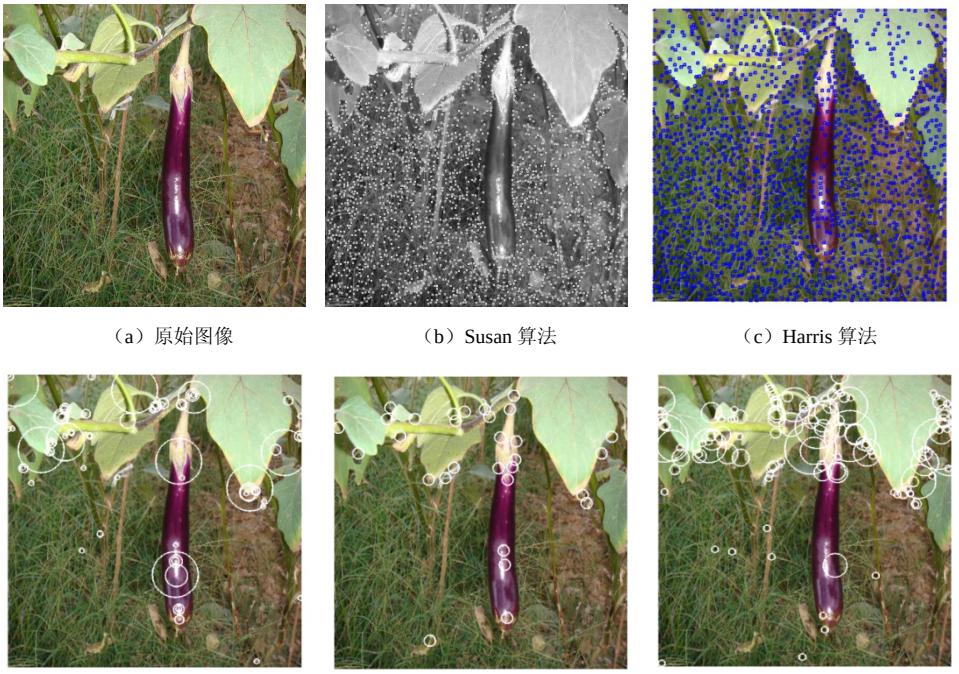
基于“词袋”模型的物体识别方法，采用具有各种不变性的点作为图像特征表示可以应对图像的诸如平移、旋转、尺度等变化，同时对于遮挡具有较好的鲁棒性。用检测到的特征生成视觉词典，不考虑物体本身的空间结构信息，不像组件-结构方法要花很多时间在统计推理求最优解上，因此计算简单，容易实现。

从整个流程来看，在基于“词袋”模型的物体识别方法中，特征表示、生成视觉词典、学习参数模型等每一过程所采用的方法都会直接影响最后的识别结果。下面分别对各个物体识别过程中常用的方法进行介绍。

5.5.3 特征表示和特征描述

特征表示是图像信息数字化的过程，因此在特征表示阶段应尽可能多地表达某类别物体区别于其他类别的本质信息。在基于“词袋”模型的物体识别方法中，图像特征的表示通常包括特征检测和描述两个阶段。多数特征检测方法都是基于图像形状特征的，常用各种兴趣点检测的方法来检测图像中局部稳定的点作为兴趣点，并用各种更为复杂的描述符对兴趣点周围的区域灰度进行计算，以此来表示物体的特征，所有的计算都是基于图像的灰度信息的。

首先需要通过各种兴趣点检测方法来检测兴趣点，如图 5.22 所示。



（a）原始图像　（b）Susan 算法　（c）Harris 算法

（d）Harris-laplace 算法　（e）Grills 算法　（f）Linderberg 算法

图 5.22　角点检测算法的检测结果

在各种兴趣点检测的方法中，不同检测器检测到特征点的属性各有侧重，在实际应用中很难有一种检测器能满足所有不变性需求。角点检测算法主要有 Harris 检测算法[94]、Harris-laplace 检测算法[94]、Susan 角点检测算法[95]、Linderberg 算法[96]等。

从图 5.22 中可以看出，Susan 算法和 Harris 算法由于更多地强调灰度梯度的变化，因而检测到角点较多，而且更多地出现在外部背景之中，因而对物体的描述不够准确。

然后需要对特征进行表述，常用 SIFT 特征、Centrist 特征等。

图 5.23 和图 5.24 分别为使用 SIFT 算法检测出来的结果，说明 SIFT 特征针对变化明显的物体具有较好的检测效果，而对于变化不明显的物体检测效果不好。针对图 5.23，物体中含有丰富的特征点，因此对于如何区分物体提供了充分的信息，而对于图 5.24，针对背景部分提供了充分的信息，而针对物体部分则显得信息不够。因此选用特征描述方法对于识别结果有很大的影响。

图 5.23　SIFT 检测物体结果较好的情况

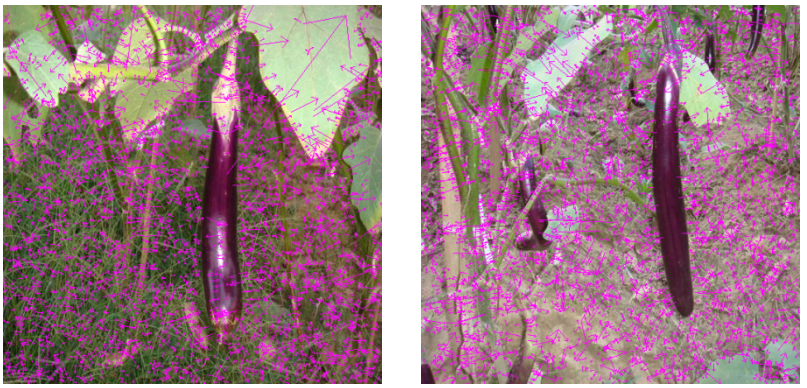


图 5.24　SIFT 检测物体结果不好的情况

5.5.4　视觉词典的形成

不同的图像因内容不同会检测到不同数目的特征点，每幅图像可能有几十个到几百个特征点，如果将图像所有特征向量直接用于模型学习，可能引起维度灾难。在“词袋”模型框架下，为了简化图像信息量，同时将每幅图像表示为具有相同长度的向量，常采用聚类的方法将所有特征点在特征空间分类，以每一聚类中心作为视觉单词，组成视觉词典，然后将所有的特征点编码到相应的视觉单词，形成图像的视觉单词直方图表示。

物体识别中常用的聚类算法如下。

1．*k*-means 算法

k-means[97]聚类算法是聚类方法中最基本的且应用最广泛的划分方法之一，是一种在无类标号数据中发现簇和簇中心的方法。该算法的目标是根据输入的参数 k，将数据集划分成 k 个聚类，使同一聚类中的对象具有较高的相似度，而不同聚类中的对象相似度较小。聚类相似度是利用从各聚类中对象的均值所获得的一个“中心对象”来进行计算的。

k-means 聚类的基本思想是：给定一个包含 n 个数据对象的数据集以及期望生成的聚类数目 k，随机地选取 k 个对象作为初始的 k 个聚类中心，计算剩余的各个样本到每一个聚类中心的距离，根据距离“中心”最近的原则，把各个样本分配到离它最近的聚类中心所属的类中，对分配后的每个新类计算其所有对象的平均值，作为该类的新的聚类中心，重复进行分配和计算新的“中心”，直到聚类中心不再发生任何变化或聚类平均误差准则函数收敛，聚类过程结束。通常，采用平方误差作为误差准则函数，其定义如下

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-m_i|^2 \tag{5.1}$$

其中，E 是数据集中所有对象的平方误差的总和；p 是空间中的点，表示给定的数据对象；m_i 是聚类 C_i 的平均值。下面给出了 *k*-means 聚类的算法概述（见算法 5.1）。

算法 5.1　*k*-means 算法

输入：聚类数目 k 和包含 n 个对象的数据集。

输出：k 个聚类，使平方误差准则最小。

步骤 1：从数据集 $\{\boldsymbol{x}_i\}_{i=1}^{n}$ 中任意选取 k 个数据对象赋给初始的聚类中心 $c_1, c_2, \cdots, c_k$；

步骤 2：对数据集中的每个样本点 $\boldsymbol{x}_i$，按下面公式计算其与各个聚类中心 c_j 的欧氏距离并根据距离最近原则获取其类别标号：

$$\text{label}(i)=\arg\min_{j}\|\boldsymbol{x}_i-\boldsymbol{c}_j\|^2,\ i=1,\cdots,n,\ j=1,\cdots,k$$

步骤 3：按下面公式重新计算 k 个聚类中心：

$$c_j=\frac{\sum_{s:\text{label}(s)=j}\boldsymbol{x}_s}{N_j},\quad j=1,2,\cdots,k$$

步骤 4：重复步骤 2 和步骤 3，直到达到最大迭代次数或聚类中心不再发生变化为止。

该算法尝试找出使平方误差函数值最小的 k 个划分，如果结果聚类是密集的，每个聚类之间区别是明显的，那么该算法获得的效果较好；对处理大数据集，该算法是相对可伸缩的且高效率的，因为它的复杂度是 $O(nkt)$，其中，n 是所有对象的数目，k 是聚类数目，t 是迭代次数。但该算法仍存在一些不足，如需要事先给出聚类个数，然而聚类个数通常是聚类之后才知道的；对孤立点敏感；不能发现非球形的类或大小差别很大的类；聚类结果不稳定，即使对于同样的输入参数，聚类的结果可能完全不同。*k*-means 算法实现过程如图 5.25 所示。

2．*k*-median 算法

由于 *k*-means 算法对于“噪声”和少数孤立点是敏感的。因此，在 *k*-median 算法中，不采用簇中对象的平均值作为参照点，而是选用簇的中心点作为参照点。该算法仍然是基于距离“中心”最近原则来执行的。

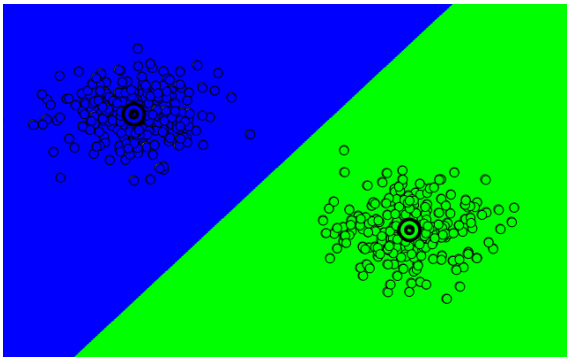


图 5.25　*k*-means 算法示意图

k-median 算法[98]的基本过程是：首先为每个簇随意选择一个代表对象，剩余的对象则根据它们与每个代表对象的距离分配给最近的代表对象所代表的簇；然后反复使用非代表对象代替代表对象来优化聚类质量，当一个中心点被某个非中心点替代时，除了未被替换的中心点外，其余各点被重新分配。通常，利用一个基于各对象与其聚类代表对象间距离的成本函数来对聚类质量进行评估，其定义如下

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-m_i| \tag{5.2}$$

其中，E 是数据集中所有对象的平方误差的总和；p 是空间中的点，表示给定的数据对象；m_i 是聚类 C_i 的中心点。下面给出了 *k*-median 聚类的算法概述（见算法 5.2）。

算法 5.2　*k*-median 算法

输入：聚类数目 k 和包含 n 个对象的数据集。

输出：满足基于各聚类中心对象的方差最小标准的 k 个聚类。

步骤 1：从 n 个数据对象任意选择 k 个对象作为初始聚类代表；

步骤 2：循环步骤 3 到步骤 5 直到每个聚类不再发生变化或达到最大迭代次数为止；

步骤 3：计算各对象与每个聚类的中心代表对象间的距离，并根据最小距离原则将各个对象分配到相应代表对象所代表的聚类中；

步骤 4：任意选择一个非代表对象 O_{random}，计算它与代表对象 O_i 进行交换的整个成本 E；

步骤 5：若 E 为负值则交换 O_{random} 与 O_i 以构成新的 k 个代表对象。

3．*k*-means++算法

k -means 算法和 *k*-median 算法都是随机选取 k 个对象作为初始聚类中心后进行迭代操作的，但是不同的初始聚类中心可能使得算法的迭代次数不同，以及获得不同的聚类结果。为了减少由于随机选择初始聚类中心带来的算法复杂度过高，并提高聚类结果的稳定性，D.Arthur 等人在文档聚类领域提出一种基于特定的概率来选择初始聚类中心的方法——D2 weighting 方法，将该初始化算法应用到 *k*-means 算法的初始中心选择上就称为 *k*-means++[99]算法。下面给出了 *k*-means++聚类的算法概述（见算法 5.3）。

算法 5.3　*k*-means++算法

输入：聚类数目 k 和包含 n 个对象的数据集。

输出：满足准则函数最小标准的 k 个聚类。

步骤 1：从数据集中随机选取一个点作为一个聚类中心 c_1；

步骤 2：对数据集中的每个点 x_i，计算它到已选中的聚类中心的最短距离 $D(x_i)$，根据以下概率来选择下一个聚类中心 c_k，满足 $c_k=x_i$；

$$p=\frac{D(x_i)^2}{\sum_{x_j\in X}D(x_j)^2}$$

步骤 3：重复步骤 2，直到选择了 k 个初始聚类中心为止；

步骤 4：初始聚类中心选定以后，就使用标准 k-means 进行聚类。

同理，对 k-median 方法也可以使用 D2 weighting 算法选择初始聚类中心。

4．核 k-means++算法

k-means 等聚类算法都没有对样本的特征进行优化，而是直接利用样本的特征进行聚类，这类方法的有效性很大程度上取决于样本的分布情况，如果一类样本散布较大，而另一类样本散布较小的话，这些方法的效果就比较差，如果样本的分布更加混乱的话，聚类结果就会惨不忍睹。对此，张莉等人提出了核聚类方法，增加对样本特征的优化，即使用 Mercer 核将输入空间的样本映射到高维特征空间，使得映射后的样本具有更好的聚类形式，并在映射后的特征空间执行聚类算法。

假设输入空间的样本 $x_k\in R^N$，$k=1,\cdots,l$，被某种非线性映射 Φ 映射到某一特征空间 H 得到 $\Phi(x^1),\Phi(x^2),\cdots,\Phi(x^l)$。那么输入空间的点积形式就可以在特征空间用 Mercer 核来表示，即

$$K(x_i,x_j)=(\Phi(x_i)g\Phi(x_j)) \tag{5.3}$$

事实上任一个函数只要满足 Mercer 条件，就可用作 Mercer 核，同时可以表示为特征空间的点积形式，见式（5.3），Mercer 条件可描述为：对任意的平方可积函数 $g(x)$，都满足

$$\iint_{L_2\otimes L_2}K(x,y)g(x)g(y)\mathrm{d}x\mathrm{d}y\geqslant 0 \tag{5.4}$$

则就可以找到核函数 K 的特征函数和特征值 $(\phi_l(x),\lambda_i)$，相应地核函数可以写成

$$K(x,y)=\sum_{i=1}^{N_H}\lambda_i\phi_l(x)\phi_l(y) \tag{5.5}$$

其中 N_H 是特征空间的维数，那么非线性映射函数可以写成

$$\Phi(x)=(\bar{\lambda}_1\phi_1(x),\bar{\lambda}_2\phi_2(x),\cdots,\bar{\lambda}_{N_H}\phi_{N_H}(x))^T \tag{5.6}$$

这样由式（5.5）和式（5.6），就可以得到式（5.3），下面我们给出一些常用的 Mercer 核函数：

多项式核 $K(x,y)=(xgy+1)^d$，其中 d 是整数，为自定义的参数；

高斯核 $K(x,y)=\exp(-\beta px-yp^2)$，$\beta>0$ 是自定义的参数；

两层神经网络 sigmoidal 核函数 $K(x,y)=\tanh(-b(xgy)-c)$，其中 b、c 是自定义的参数。

假设输入空间的样本已被映射到特征空间 $\Phi(x_1),\Phi(x_2),\cdots,\Phi(x_l)$，特征空间中 Euclidean 距离可表示为

$$d_H(x,y)=\sqrt{p\Phi(x)-\Phi(y)p^2}=\sqrt{\Phi(x)g\Phi(x)-2\Phi(x)g\Phi(y)+\Phi(y)g\Phi(y)} \tag{5.7}$$

一般情况下，非线性函数的表达式是未知的，所以由式（5.3），式（5.7）可以写为

$$d_H(x,y)=\sqrt{K(x,x)-2K(x,y)+K(y,y)} \tag{5.8}$$

因此我们将式（5.8）作为聚类的相似度的度量函数，相应的聚类准则是使得下面的目标函数最小。

$$J=\sum_{i=1}^{C}\sum_{j=1,x\in C_i}^{N_i}\left[K(x_j,x_j)-\frac{2}{N_i}\sum_{k=1}^{N_i}K(x_j,x_k)+\frac{1}{N_i^2}\sum_{k,p=1}^{N}K(x_k,x_p)\right] \tag{5.9}$$

其中，C 是聚类类别个数；N_i 是第 C_i 类样本的个数。该类中心的模为

$$pW_i p^2=\frac{1}{N_i^2}\sum_{k,p=1}^{N_i}K(x_k,x_p) \tag{5.10}$$

有些文献也采用了核聚类方法，使用的是直方图相交核，算法概述如下（见算法 5.4）。

算法 5.4　核 k-means++算法

输入：给定 n 个直方图 $h_1,h_2,\cdots,h_n\in R_+^d$，视觉词典的大小 m 以及公差 ε。

输出：从直方图到视觉单词索引的一个映射 $\omega_1(h_*):R_+^d\to\{1,\cdots,m\}$，即对任意直方图 $h_*\in R_+^d$，$\omega_1(h_*)=\arg\min_{1\leqslant i\leqslant m}\|\phi(h_*)-m_i\|^2$。

步骤 1：迭代次数 $t\leftarrow 0$，聚类质量 $e^0\leftarrow\infty$；

步骤 2：使用 k-means++方法选择 m 个直方图 $\bar{h}_1,\bar{h}_2,\cdots,\bar{h}_m$，使用 $m_i=\phi(\bar{h}_i)$ 作为初始中心；

步骤 3：重复步骤 2，直到找到 m 个初始中心；

步骤 4：对所有的 $1\leqslant i\leqslant n$，$l_i\leftarrow\arg\min_{1\leqslant j\leqslant m}\|\phi(h_i)-m_j\|^2$，即 l_i 为距离数据点 h_i 最近的聚类中心的索引；

步骤 5：对所有的 $1\leqslant i\leqslant m$，$\pi_i=\{j\mid l_j=i,i\leqslant j\leqslant n\}$，即 π_i 是属于聚类中心 m_i 的数据点索引的集合；

步骤 6：更新聚类中心：对所有的 $1\leqslant i\leqslant m$，根据下式计算 m_i；

$$m_i\leftarrow\frac{\sum_{j\in\pi_i}\phi(h_j)}{|\pi_i|}$$

步骤 7：$t\leftarrow t+1$，e^t 如下式；

$$e^t=\sum_{1\leqslant i\leqslant n}\|\phi(h_i)-m_{l_i}\|^2$$

步骤 8：直到 $e^{t-1}-e^t<\varepsilon$。

算法 5.4 的主要成分是使用直方图相交核的核 k-means++算法，由于中心 m_i 是未知向量，在高维空间 Φ，关键计算使用核技巧 $\kappa_{HI}(h_1,h_2)=\phi(h_1)\cdot\phi(h_2)$ 方式实现，具体描述如下

$$\begin{aligned}\|\phi(h_*)-m_i\|^2&=\left\|\phi(h_*)-\frac{\sum_{j\in\pi_i}\phi(h_j)}{|\pi_i|}\right\|^2\\&=\|\phi(h_*)\|^2+\frac{\sum_{j,k\in\pi_i}\kappa_{HI}(h_j,h_k)}{|\pi_i|^2}-\frac{2\sum_{j\in\pi_i}\kappa_{HI}(h_*,h_k)}{|\pi_i|}\end{aligned} \tag{5.11}$$

方程中的第一项对算法 5.4 中步骤 4 的结果不会产生影响，而且第二项能够被预先计算，因此大部分的计算被花费在计算最后一项，可以注意到，这项与使用直方图相交核的两类支持向量机分类器相似，它的形式如下

$$\operatorname{sgn}\left(\sum_{i\in\pi} y_i\alpha_i\kappa_{HI}(h_*\cdot h_i)+\rho\right) \tag{5.12}$$

其中，h_i 是支持向量；α_i 是对应的权重；y_i 是对应的标签。公式（5.11）和公式（5.12）的最后一项都是下述表达的特殊形式

$$f(h_*)=\sum_{i\in\pi} c_i\kappa_{HI}(h_*,h_i) \tag{5.13}$$

其中，π 是支持向量的索引集合；c_i 是常数。

$$\begin{aligned} f(h_*)&=\sum_{i\in\pi} c_i\kappa_{HI}(h_*,h_i)=\sum_{i\in\pi}\sum_{1\leqslant j\leqslant d} c_i\min(h_{*j},h_{ij})\\ &=\sum_{1\leqslant j\leqslant d}\sum_{i\in\pi} c_i\min(h_{*j},h_{ij})=\sum_{1\leqslant j\leqslant d}\left(\sum_{h_{*j}\geqslant h_{ij}} c_ih_{ij}+h_{*j}\sum_{h_{*j}\leqslant h_{ij}} c_i\right)\end{aligned} \tag{5.14}$$

公式（5.14）中的两个被加数能够被预先计算。如果假设 h_{*j} 是 0 到 $h_{\max}$ 中的一个整数（通常，一个在 R_+^d 上的向量能够通过如下方式转换成 N^d：首先对直方图乘一个整数，然后凑整它的分量到最近邻整数），那么让 T 是一个大小为 $d*(1+h_{\max})$的表格，并且将式（5.15）分配到表格的第 j 行的第 k 列，即 $T(j,k)$。

$$\sum_{k\geqslant h_{ij}} c_ih_{ij}+k\sum_{k\leqslant h_{ij}} c_i \tag{5.15}$$

所以

$$f(h_*)=\sum_{j=1}^{d} T(j,h_{*j}) \tag{5.16}$$

具体算法见算法 5.5。

算法 5.5　快速计算算法

输入：给定 n 个直方图 $h_1,h_2,\cdots,h_n\in N^d$，对于 $1\leqslant i\leqslant n$，$1\leqslant j\leqslant d$ 满足 $0\leqslant h_{ij}\leqslant h_{\max}$。

输出：快速方法计算式（5.13），即计算式（5.16）。

步骤 1：创建大小为 $d*(1+h_{\max})$ 的表格 T，$T\leftarrow \mathrm{Table}(d*(1+h_{\max}))$；

步骤 2：对 $1\leqslant j\leqslant d, 0\leqslant k\leqslant h_{\max}$ 计算下式。

$$T(j,k)=\sum_{k\geqslant h_{ij}} c_ih_{ij}+k\sum_{k\leqslant h_{ij}} c_i$$

5．鲁棒划分聚类算法及其扩展

尽管 k-means++聚类在时间和精确度方面都得到了许多改善，但仍存在不足，如对可变密度敏感等。因此 Wu 等人提出对 k-means++产生的聚类结果进一步使用 One-Class SVM 分类器，使每个聚类获得一个保留了大部分直方图的紧凑子集，并证实了这种紧凑的超球面在可变密度的情况下能更好地概述一个视觉单词。但是，对于不具有明显内核的距离函数不能使用 One-Class SVM，这就限制了其使用范围。为此，本书引入并扩展了鲁棒划分聚类算法，该算法是在 k-means 聚类算法的基础上提出的，与 k-means 的区别在于鲁棒划分聚类算法使

用 ROBIN（Robust Initialization）方法作为初始种子的选取方法。

鲁棒划分聚类算法是由 Mohammad Al Hasan 等人于 2009 年提出的，使用了一种新的初始中心的选取方法，即 ROBIN 算法，该算法是一种确定的和鲁棒的初始化算法，它对异常值几乎不敏感，能够处理可变密度和可变大小的聚类，传统的随机种子选择方法需要运行多次来获得好的初始中心，而它只需要运行一次就足够获得好的初始中心。除了聚类中心数 k 之外，它只需要一个参数 m_{p}，即计算一个中心点是否是一个异常点时考虑的邻域数目，并且 ROBIN 方法对 m_{p} 的选择也不是非常敏感。该算法本质上是避免将异常点作为种子点。所谓异常点就是那些密度与其邻近点密度非常不同的点。对于 ROBIN 算法，首先要计算一个点是一个异常点的程度，为了使局部异常点的估算自动考虑到可变密度区域和可变聚类大小的情况，在计算过程中必须依次考虑近邻点的局部密度。并且受异常值估量的限制，ROBIN 算法必须确保种子点尽可能是分离的。

在介绍 ROBIN 算法前，先介绍局部异常因子的概念。

为了计算一个点是一个异常点的程度，Mohammad Al Hasan 等人使用了局部异常因子的概念。给定最小的点阈值 m_{p}，点 $x \in D$ 的 m_{p} 距离定义如下

$$\mathrm{dist}(x, x_{m_{\mathrm{p}}}) = \mathrm{dist}(x, o) \tag{5.17}$$

其中，dist(x,o)表示点 x 与点 $o \in D$ 之间的欧氏距离，该距离必须同时满足两个条件：（1）至少包含 m_{p} 个点 $o' \in D \setminus \{x\}$ 使得 $d(x, o') \leqslant d(x, o)$；（2）最多包含 $m_{\mathrm{p}}-1$ 个点 $o' \in D \setminus \{x\}$ 使得 $d(x, o') < d(x, o)$。点 x 的局部近邻定义如下

$$N(x, m_{\mathrm{p}}) = \{y \in D \setminus \{x\} \mid \mathrm{dist}(x, y) \leqslant \mathrm{dist}(x, x_{m_{\mathrm{p}}})\} \tag{5.18}$$

其中，dist(x, y)表示点 x 与点 y 之间的欧氏距离，点 y 是点 x 的 m_{p} 近邻点，点 $x_{m_{\mathrm{p}}}$ 是点 x 的第 m_{p} 个近邻点，因此 $N(x, m_{\mathrm{p}})$ 至少包含 m_{p} 个点。目前，大多数视觉词典产生方法使用的是欧氏距离测量，但是由于直接在样本空间计算欧氏距离的聚类算法的有效性很大程度上取决于样本的分布情况，当分布不均甚至混乱时获得的效果不理想，同时在有监督分类学习中，直方图相交核被证明能够获得更高的识别率，这里对 ROBIN 算法在距离函数方面进行了扩展，即除了欧氏距离式（5.19）外，还使用了绝对值距离和直方图相交核，如下所示

$$\mathrm{dist}(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_h - y_h)^2} \tag{5.19}$$

$$\mathrm{dist}(x, y) = \| x_1 - y_1 \| + \cdots + \| x_h - y_h \| \tag{5.20}$$

$$\begin{aligned} \mathrm{dist}(x, y) &= \sqrt{x^2 - 2xy + y^2} \\ &= \sqrt{\kappa_{HI}(x,x) - 2\kappa_{HI}(x,y) + \kappa_{HI}(y,y)} \end{aligned} \tag{5.21}$$

按下式计算点 x 的局部密度

$$\mathrm{density}(x, m_{\mathrm{p}}) = \left(\frac{\sum_{y \in N(x, m_{\mathrm{p}})} \mathrm{reachdist}(x, y)}{| N(x, m_{\mathrm{p}}) |} \right)^{-1} \tag{5.22}$$

其中 reachdist(x, y)是点 x 到点 y 的可达距离，即

$$\mathrm{reachdist}(x, y) = \max\{\mathrm{dist}(x, x_{m_{\mathrm{p}}}), \mathrm{dist}(x, y)\} \tag{5.23}$$

图 5.26 可以说明 m_{p}=4 时的可达距离的基本思想。如果点 y 远离点 x，如图中的点 y_2，

这时可达距离就是它们之间的实际距离。然而，如果它们是“足够的”接近，如图中的点 y_1，这时，实际距离被点 x 的 m_p 距离替代。这样做是为了使所有靠近点 x 的点 y 的实际距离的统计波动可以显著降低。这种平滑作用的优势可以通过参数 m_p 来控制，m_p 的值越大，相同邻域里的点的可达距离越相似。

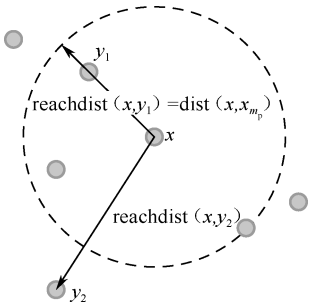


图 5.26　m_p=4 时的 reachdist(x, y_1)和 reachdist(x,y_2)

一般来说，点 x 与其近邻点的距离越小，点 x 的密度越高。这时，点 x 的平均相对密度为点 x 的密度与它的最近邻点的平均密度的比率，如式（5.24）所示。

$$\mathrm{ard}(x,m_p)=\frac{\mathrm{density}(x,m_p)}{\left(\dfrac{\sum\limits_{y\in N(x,m_p)}\mathrm{density}(y,m_p)}{|N(x,m_p)|}\right)} \tag{5.24}$$

最后点 x 的局部异常因子（Local Outlier Factor）就是点 x 的平均相对密度的逆，如式（5.25）所示。

$$\mathrm{LOF}(x,m_p)=\mathrm{ard}(x,m_p)^{-1} \tag{5.25}$$

与一个点的所有近邻点比较，如果该点是低密度的，那么它的平均相对密度较低，它的 LOF 值是高的，因此 LOF 值代表了一个点是一个异常点的程度。由于一个点的密度和其近邻点的密度大致相同，所以属于一个聚类的点都有一个约等于 1 的 LOF 值。

LOF 有两个优良属性：一是它对存在不同大小和密度的聚类簇的数据集是鲁棒的；二是尽管对于不同的 m_p 值，LOF 的值可能有些不同，但是决定一个点是否是一个异常点通常是鲁棒的，也就是说，在一个大范围的 m_p 值内，异常点的 LOF 值远高于 1，而属于一个聚类的点的 LOF 值将认为接近 1。

根据局部异常因子的定义，从已排序的数据集中依次取出数据点并计算异常程度，即 LOF 值，如果 LOF 近似等于 1（在实验中，设置固定阈值为 1.05），说明数据点不是异常点，即该数据点可以选为初始中心点，重复以上过程，直到选取到期望个数的种子点为止。ROBIN 算法描述如算法 5.6。

算法 5.6　ROBIN 算法

输入：数据集 D，聚类数目或期望的种子数 k，计算 LOF 时考虑的邻域数目 m_p。 输出：初始中心点集合 C。 处理流程：

步骤 1：取任意参考点 r（如原点）插入空的初始中心点集合 C；

步骤 2：根据到 C 中已获得的种子点的最小距离降序排序 D 中的点；

步骤 3：从 D 中依次取出每个点 x；

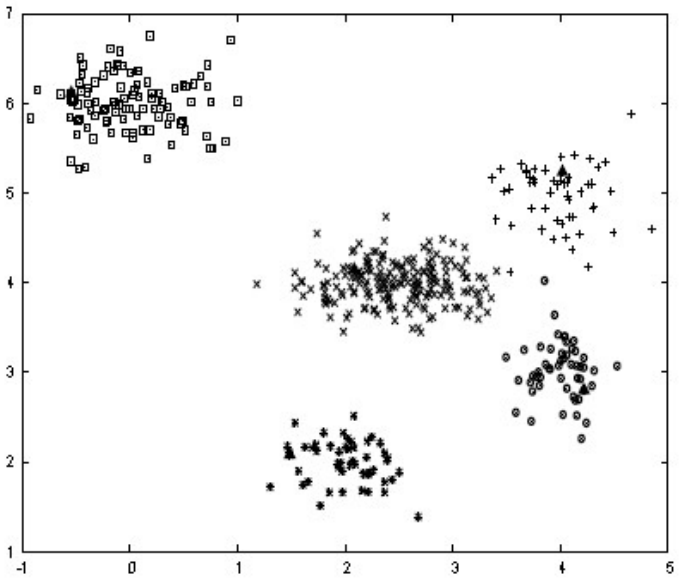
步骤 4：计算 $\mathrm{LOF}(x,m_p)$，如果 $\mathrm{LOF}(x,m_p)$约等于 1（在实验中，设置固定阈值为 1.05），添加 x 进入 C，执行步骤 5，否则执行步骤 3；

步骤 5：重复步骤 2～步骤 4，直到除了 r 以外，选择了 k 个初始中心点；

步骤 6：从 C 中移出 r，得到初始中心点集合 C。

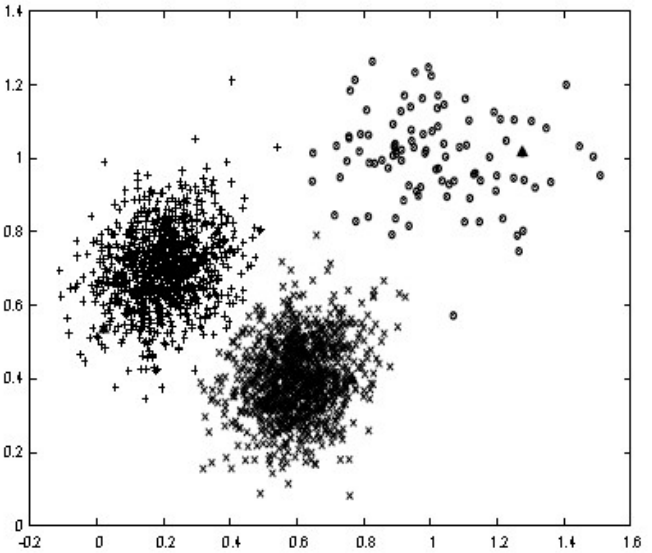
根据算法 5.6 可知，ROBIN 算法首先向初始为空的种子点集合 C 中插入参考点 r（步骤 1），如原点。稍后将会从 C 中移除参考点 r（步骤 6），因为点 r 仅仅用于寻找第一个种子点，该种子点是数据集的“边缘”点，也就是说，它到原点的距离最大而且它的 LOF 值接近于 1。后面的种子点以相似的方式获得，即先对 D 中的点按其到 C 中已有的种子点的最小距离降序排序（步骤 2）。在每次重复过程中（步骤 5），ROBIN 算法找到一个新的种子点。根据距离的排序（降序）顺序依次考虑每个点（步骤 3），只要一个点的 LOF 值近似等于 1（在实验中，设定固定阈值为 1.05）就不再考虑该点后面的点，否则依次考虑该点的下一个点的 LOF 值（步骤 4）。一旦找到 k 个种子点后算法就停止。ROBIN 算法的最后，集合 C 包含了 k+1 个点，移除参考点 r 获得最终的种子点集合（步骤 6）。值得注意的是，如果初始参考点选择原点，那么 ROBIN 算法的输出是确定性的，而且对数据点的次序不敏感。执行 ROBIN 初始化算法后，初始中心点已选定，就可以执行标准的 k-means 聚类算法对数据集进行鲁棒的聚类。

图 5.27 显示了 Mohammad Al Hasan 等人使用 ROBIN 算法在实验数据集上选择初始种子点的结果。图 5.27（a）中显示了 5 个聚类的样本集，通过 m_p=10 的 ROBIN 算法选择的初始种子点（三角形），可以看出每个聚类中都选择了一个种子点。

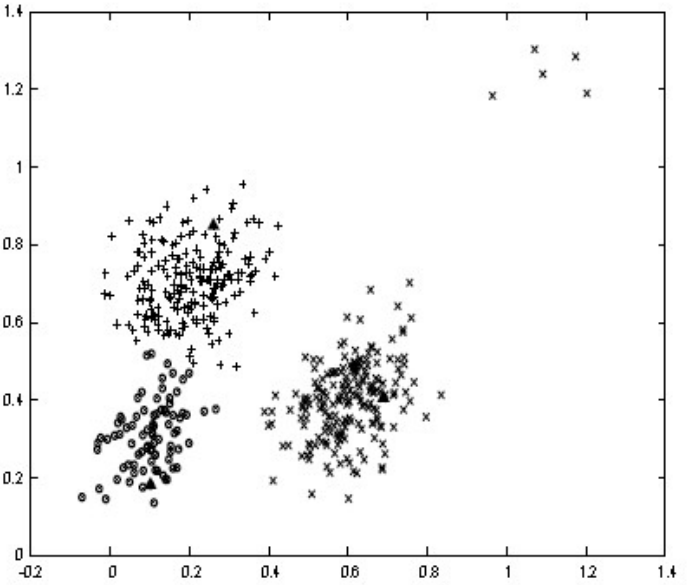


（a）没有合并或分裂

图 5.27　ROBIN 种子点（三角形）



（b）密度不敏感



（c）异常点不敏感

图 5.27　ROBIN 种子点（三角形）（续）

ROBIN 算法也可以处理不同密度的聚类，因为 LOF 基于局部密度来考虑了一个点作为异常点的程度，它不排斥任何的低密度聚类并从该聚类中选择一个种子点，如图 5.27（b）显示的 3 个聚类，分别包含 1000、1000、100 个点，ROBIN 算法选择的 3 个聚类中分别选择了不同的种子点（三角形）。另外，ROBIN 算法总是避开噪声点去选择种子点。图 5.27（c）中包含 3 个聚类，左上方还有少数噪声点，ROBIN 初始化算法不考虑任何噪声点作为种子点。

5.5.5　视觉单词的直方图

形成视觉词典后，图像将表示为视觉单词的直方图。在现有基于“词袋”模型的物体识别方法中，大多数采用文本识别中的词频（Term Frequency，TF）或逆向文件频率（Inverse Document Frequency，IDF）的方法形成图像的直方图表示。

TF 是统计某一给定词语在该文件中出现的次数，对于某一特定文件里的词语 t_j 来说，其重要性表示如下

$$\mathrm{tf}_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{5.26}$$

其中，$n_{i,j}$ 表示 t_j 在文件 d_j 中出现的次数；分母则表示文件 d_j 中所有单词出现的次数总和。

IDF 是一个词语普遍重要性的度量。某一特定文件的 IDF 可以由总文件数除以包含该词语文件的数目，所得结果取对数得到。

$$\mathrm{idf}_i=\log\frac{|D|}{|\{d\,|\,t_j\in d\}|} \tag{5.27}$$

其中，$|D|$表示文件总数；d 为包含词语 t_j 的文件数目。

在这些方法中，都是将图像的特征向量分配到与其最近的视觉单词，以视觉单词来代替所有的特征向量，然后计算每一单词对图像的贡献形成图像的单词直方图来表示。

有了直方图的表示，如何对直方图进行距离的计算？直方图交叉核是一种较为有效的方法。直方图交叉核（Histogram Intersection）对于物体识别有着较好的效果，这里专门针对直方图交叉核进行一些简单的阐述。

直方图交叉核最初来自 *The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features* 这篇论文，用来对特征构成的直方图进行相似度匹配，下面介绍其原理。

假设图像或其他数据的特征可以构成直方图，根据直方图间距的不同可以得到多种类型的直方图 $\psi(x)=[H_{-1}(x),H_0(x),\cdots,H_L(x)]$。假设 $H_0(x)$ 里每个直方图宽度为 a，那么 $H_1(x)$ 为 $2a$，以此类推。

两个数据集相似度可以表示为

$$K_\Delta(\psi(y),\psi(z))=\sum_{i=0}^{L}\omega_i N_i \tag{5.28}$$

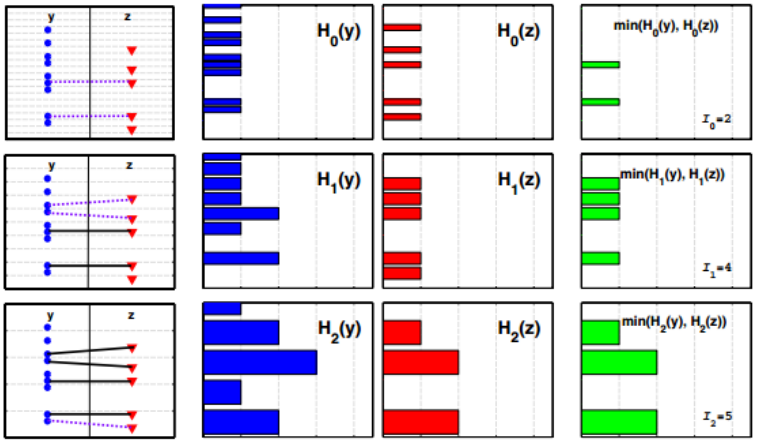
y 和 z 分别代表不同的数据集，其中 ω 代表权重，论文里将 ω_i 设置为 $1/2^i$，N 代表每两层之间的新匹配的数目，可以通过下式计算

$$N_i=\tau(H_i(y),H_i(z))-\tau(H_{i-1}(y),H_{i-1}(z)) \tag{5.29}$$

式（5.29）中的 τ 可以通过下式计算

$$\tau(A,B)=\sum_{j=1}^{r}\min(A^{(j)},B^{(j)}) \tag{5.30}$$

详细计算过程如图 5.28 所示。



（a）点集　　（b）直方图金字塔　　（c）交叉核

图 5.28　直方图交叉核示意图

图 5.28（a）里的 y 和 z 代表两种数据分布，三幅图分别代表三层金字塔，不同层中虚线之间的间距不同。图 5.28（b）中对每个间距进行直方图的统计。图 5.28（c）就是 τ 的计算结果，是通过图 5.28（b）中 y 和 z 的直方图取交集得到的，计算交集后的点的个数，得到了 τ，然后计算 N_i，最后得到两个数据集的相似度，如式（5.31）所示。

$$K_{\Delta}(\psi(y),\psi(z))=\sum_{i=0}^{L}\frac{1}{2^i}(\tau(H_i(y),H_i(\mathrm{z}))-\tau(H_{i-1}(y),H_{i-1}(\mathrm{z})))\tag{5.31}$$

5.5.6　参数模型学习

越来越多的机器学习方法引入到物体识别中，用于参数模型的学习，如 EM、神经网络、AdaBoost、支持向量机（Support Vector Machine，SVM）等。SVM 作为目前主流的小样本学习理论，在基于“词袋”模型的物体识别中，常用来对训练样本进行参数模型学习。下面将详细介绍 SVM 学习方法。

支持向量机[100-102]是由 Vapnik 和他的合作者提出，自提出后以来，在范围广大的应用中，SVM 的性能胜过其他大多数的学习系统。本节使用支持向量机进行分类识别，下面主要介绍支持向量机的分类原理。

支持向量机基于统计学理论的结构风险最小化分类器，该方法是由线性可分的情况下的最优分类线提出的，可用图 5.29 的两维两类线性可分情况说明。图中，实心点和空心点代表两类的训练样本，H 为分类线，H_1、H_2 分别为过各类中离分类线最近的样本且平行于分类线的直线，它们之间的距离称为分类间隔（Margin）。

所谓最优分类线就是要求分类线不仅能将两类正确分开（训练错误率为 0），而且使分类间隔最大。设线性可分的样本集 $\{(x_i,y_i)\mid i=1,2,\cdots,n,\ x_i\in R^d,\ y_i\in\{+1,-1\}\}$。$d$ 维空间中线性判别函数的一般形式为 $g(x)=w\cdot x+b$，分类线方程为 $w\cdot x+b=0$，对判别函数进行归一化，使两类所有样本都满足 $|g(x)|\geqslant 1$，即使离分类线最近的样本的 $|g(x)|=1$，这样分类间隔就等于 $2/\|w\|$，因此使间隔最大等价于使 $\|w\|$（或 $\|w\|^2$）最小，而要求分类线对所有样本正确分

类，就要求它满足 $y_i[(w\cdot x_i)+b]-1\geqslant 0,\ i=1,\cdots,n$ 。

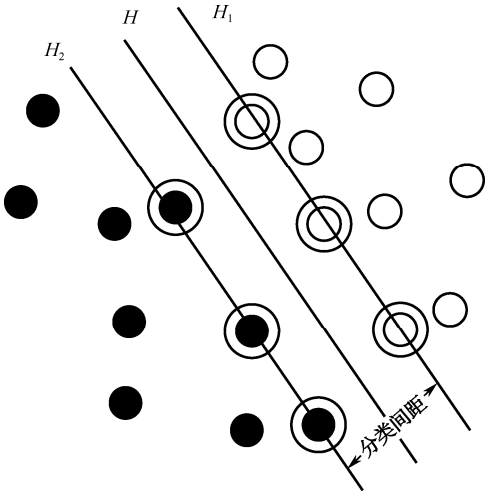


图 5.29　线性可分情况下的超平面

因此，最优分类线可以通过解方程式（5.32）的二次优化问题来获得，即

$$\mathrm{Min}(\Phi(w))=\frac{1}{2}\|w\|^2 \tag{5.32}$$

满足约束条件 $y_i(w\cdot x+b)\geqslant 1(i=1,2,\cdots,n)$ 。为此，可以定义如下 Lagrange 函数

$$L(w,b,\alpha)=\frac{1}{2}(w^{\mathrm{T}}w)-\sum_{i=1}^{n}\alpha_i\{y_i[(w\cdot x_i)+b]-1\} \tag{5.33}$$

然后对 w 和 b 求 Lagrange 函数的极小值。在求极小值过程中，可以将原二次规划问题转化为简单的对偶问题，即

$$\begin{aligned}&\mathrm{Max}W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)\\&w^*=\sum_{i=1}^{n}\alpha_i y_i x_i\\&b^*=y_i-w^*\cdot x_i\end{aligned} \tag{5.34}$$

满足条件 $\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\geqslant 0,i=1,2,\cdots,n$ ，其中，$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_n)$ 是 Lagrange 乘子，w^* 是最优超平面的法向量，b^* 是最优超平面的偏移量。这是一个不等式约束下二次函数极值问题，存在唯一解。且这个问题的解必须满足 $\alpha_i\{y_i(w\cdot x+b)-1\}=0,i=1,2,\cdots,n$ 。由此可知，那些 $\alpha_i=0$ 的样本对于分类没有任何作用，只有 $\alpha_i>0$ 的样本才对分类器起作用，这些样本即为支持向量（Support Vector）。解上述问题后得到的最优分类函数为

$$f(x)=\mathrm{sgn}\{(w^*\cdot x)+b^*\}=\mathrm{sgn}\left\{\sum_{i=1}^{n}\alpha_i^* y_i(x_i\cdot x)+b^*\right\} \tag{5.35}$$

其中 sgn 为符号函数。

对于线性不可分的情况，它的基本思想可以概括为：首先通过非线性变换（通过定义适当的内积函数实现）将低维输入空间中的线性不可分的样本变换到一个高维空间，使其成为

线性可分的，然后在这个新空间中求取最优线性分类面，如图 5.30 所示。

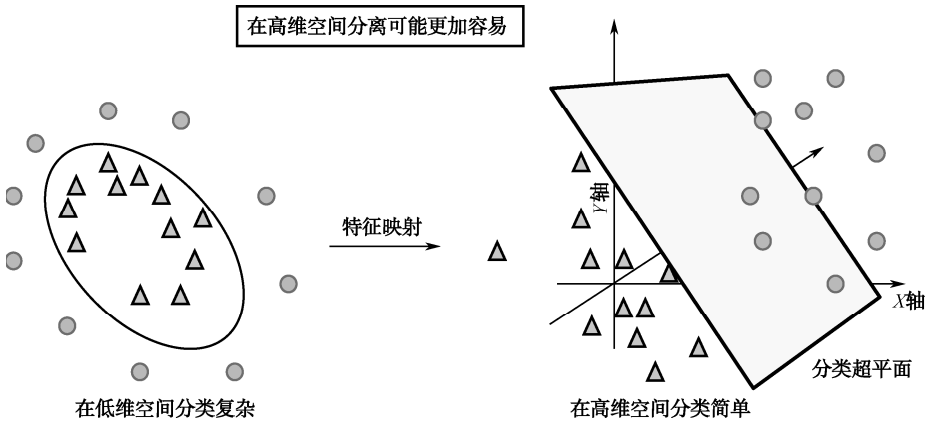


图 5.30　线性不可分的情况

设有非线性映射 Φ： $R^d \to H$ 将输入空间的样本 x 映射到一个高维（可能是无穷维）的特征空间 H，并在特征空间 H 中构造最优超平面时，训练算法仅使用空间中的点积，即 $\Phi(x_i)\cdot\Phi(x_j)$，而没有单独使用 $\Phi(x_i)$。因此能够找到一个函数 K（函数 K 称为核）使得 $K(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j)$，这样，就可以运用原空间的函数来实现高维空间上的内积运算。根据泛函的有关理论，只要一种核函数满足 Mercer 原理，它就对应某一变换空间中的内积。因此，只要在最优分类面上使用适当的内积函数就可以获得某一非线性变换后的线性分类。此时式（5.34）的目标函数变为

$$\mathrm{Max}W(\alpha)=\sum_{l=1}^{n}\alpha_l-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i\cdot x_j) \tag{5.36}$$

其相应的分类函数也变为

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{n}y_i\alpha_i K(x\cdot x_i)+b^*\right) \tag{5.37}$$

这就是支持向量机。目前有几类使用较多的内积核函数：第一类是线性核；第二类是径向基函数（RBF），也称作高斯核函数；第三类是直方图交叉核函数。各核函数如式（5.38）所示。

$$\begin{aligned}&k(x,y)=x^T y\\&k(x,y)=e^{-\rho d(x,y)}\\&k_{HI}(x,y)=\sum\nolimits_i \min\{x_i,y_i\}\end{aligned} \tag{5.38}$$

在 RBF 核中，$d(x,y)$表示两向量的距离，可以在特征空间中选择不同的距离，如χ_2距离、L_1距离、欧式距离等，当 $d(x,y)$用欧式距离时，RBF 核是常用的高斯核函数。

5.5.7　物体识别全过程

在视觉词典的产生过程中，首先通过间距为 16 的网格密集采样 16×16 图像片作为特征，并采用 SIFT 描述符描述特征，同时将所有的特征向量归一化，使得直方图的每一维为非负

整数且和为 128；然后对训练集图像的特征描述符进行鲁棒划分聚类，形成大小为 200 的视觉词典，如图 5.31 所示。

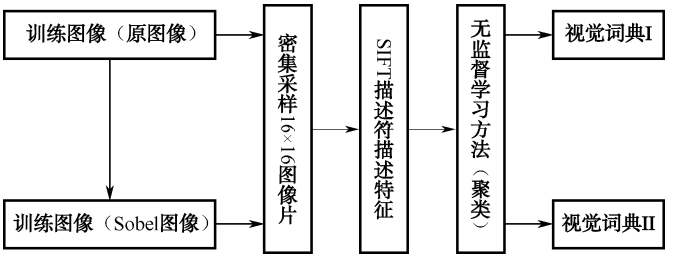


图 5.31　视觉词典产生过程

如图 5.32 所示，在图像表示过程中，首先对图像进行双线性插值，获得 5 个不同尺度大小的图像，并在 5 个不同尺度的图像上通过间距为 4 的网格密集采样特征，计算 SIFT 描述符，根据已生成的视觉词典，采用“硬分配”方法将每个特征向量映射到从 1 到视觉词典大小之间的一个整数（即视觉单词索引），表示图像或图像子窗口为视觉单词的直方图。为了纳入空间信息，采用了空间分层技术获得 31 个子窗口，每幅图像的子窗口表示为 200 维的直方图，一幅图像就表示为 6200 图像的直方图连接 0 维的直方图。实验中还使用了输入图像的 Sobel 梯度图像作为附加输入来捕捉边缘信息，并与原输入形成 12400 维的图像的直方图表示。连接原图像与其 Sobel 图像的直方图的方法有许多，本节采用如图 5.33 所示的方法，其中 $B:(1-B)=1$。

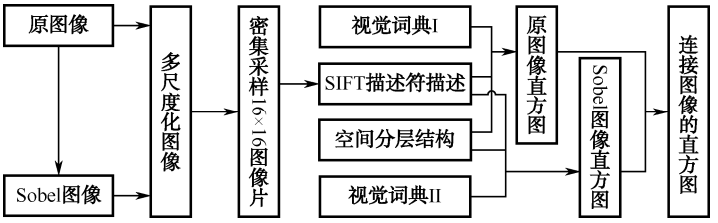


图 5.32　图像表示过程

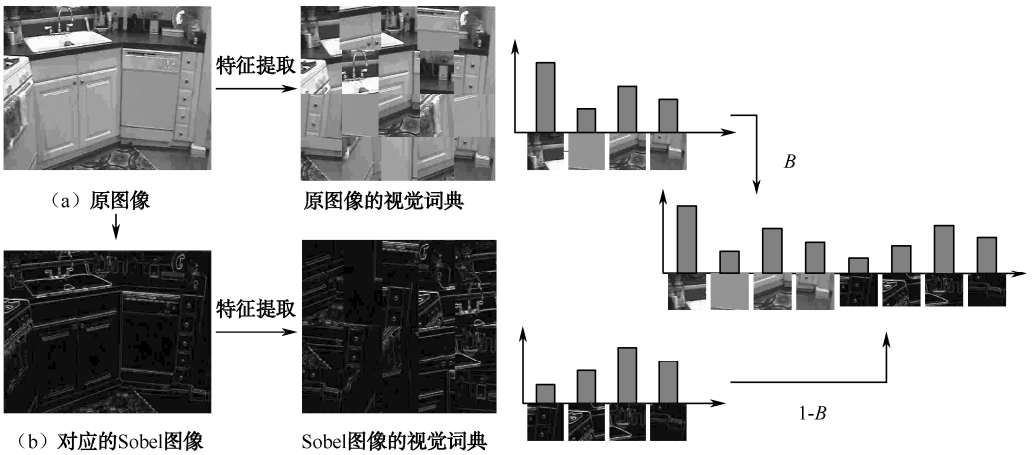


图 5.33　连接原图像与对应的 Sobel 图像的直方图

在学习和识别过程中，使用 BSVM 分类器对训练集的图像直方图训练得到模型，结合模型对测试集进行分类识别，在分类器中，使用的核函数是直方图相交核，如图 5.34 所示。

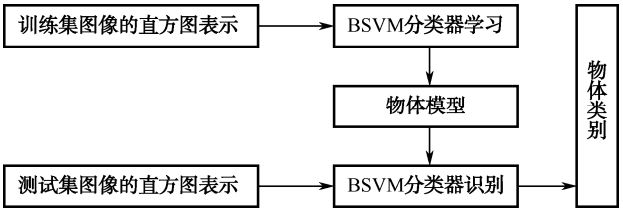


图 5.34　学习和识别过程

5.5.8　物体识别结果及其分析

1．物体识别常用数据库简介

（1）Caltech-101 数据库

Caltech-101 图像数据库（http://vision.caltech.edu）是物体识别中常用的数据库，该数据库是 Fei-Fei Li, Marco Andreetto 和 Marc’Aurelio Ranzato 等人于 2013 年 9 月收集起来的。该图像数据库包含 101 类物体，共 8677 张图片，每类物体图像包含 31～800 张图像，每幅图像的大小大概为 300×200 像素，每幅图片中仅包含一类物体，每类物体图像之间存在大量的类内变化，类与类之间也存在较大的相似性，大部分物体处于图片的中心位置，是很好的大规模物体类别的基础分类图像库，也是目前非常具有挑战性的物体识别的图像库之一，用来进行不同种物体的识别，如图 5.35 所示。

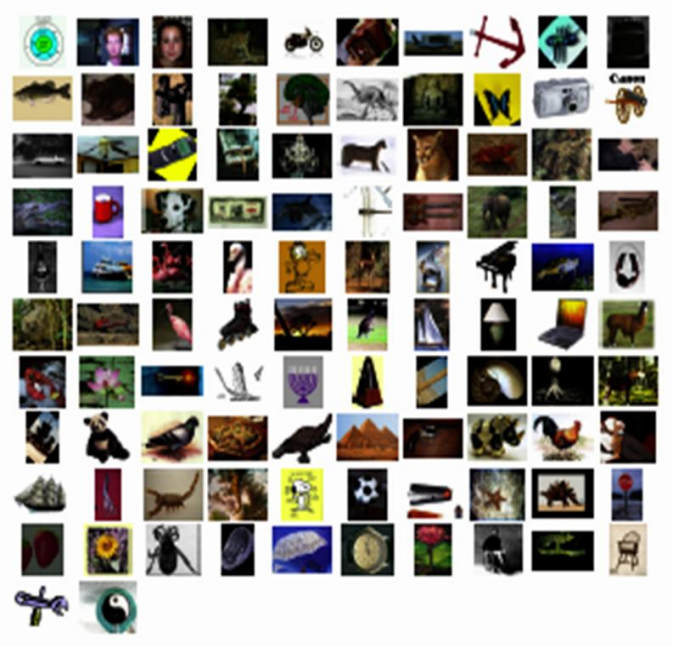


图 5.35　Caltech-101 图像集的部分图像

（2）Scene 数据集

Scene 数据集来自http://www.cs.unc.edu/-lazebnik/research/，由 Svetlana Lazebnik 等人收集。该数据集包含 Bedroom、Suburb、Industrial、Kitchen、Livingroom、Coast、Forest、Highway、Insidecity、Moutain、Opencountry、Street、Building、Office 和 Store 共 15 类场景数据，图像大小不等。

（3）Sports 数据集

Sports 数据集来自http://vision.stanford.edu/lijiali/event_dataset/，主要由 Li-Jia Li 等人收集。该数据集包含 8类体育运动：Rowing（250 幅图像）、Badminton（200 幅图像）、Polo（182 幅图像）、Bocce（137 幅图像）、Snow Boarding（190 幅图像）、Croquet（236 幅图像）、Sailing（190 幅图像）和 Rock Climbing（194 幅图像），如图 5.36 所示。这些图像按照人的主观判断分成容易和不容易两类。

图 5.36　Sports 图像集的部分图像

2．物体识别系统简介

如图 5.37 所示，根据实验过程，本书简单地设计了一个基于词袋模型的物体识别过程的演示界面，读者可以根据界面上的提示选择或输入相应的参数信息，在右边的文本框可以显示物体识别的整个执行过程。

3．粗网格基础上的识别结果

根据选取的特征，我们进行了分类实验。因为针对所有 101 个类别进行测试，测试时间过长，所以我们将茄子样本（87 幅图片）代替其中的第一类手风琴样本，加上飞机（800 幅

图片）、锚（42 幅图片）和蚂蚁（42 幅图片）共 4 个样本库、971 幅图片进行了测试，实验结果如表 5.12 和表 5.13 所示。

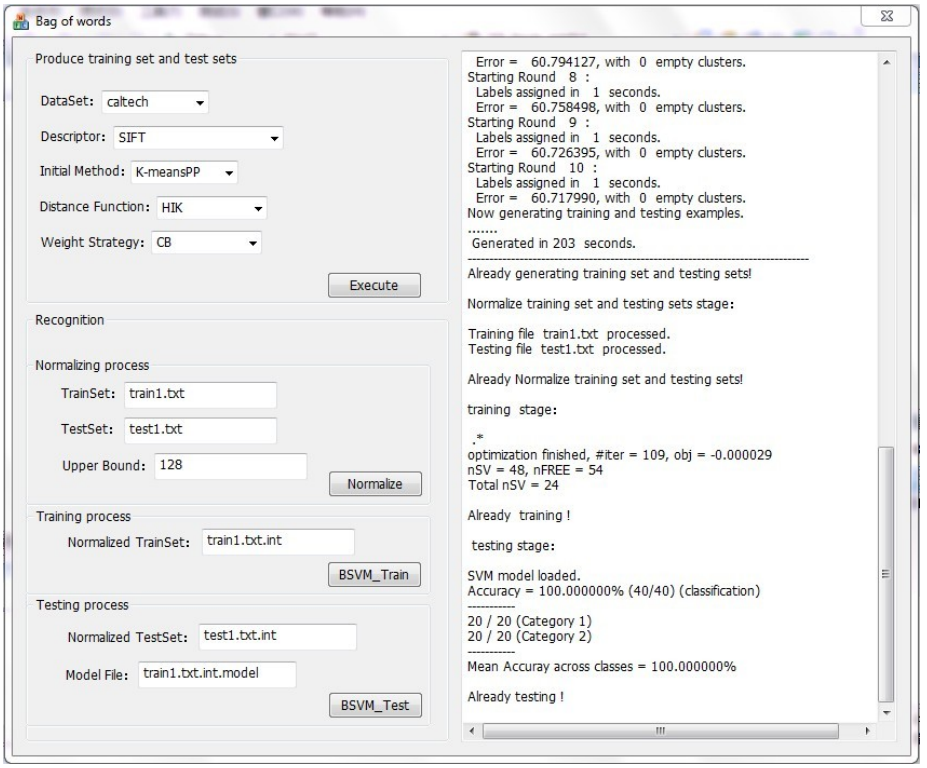


图 5.37　基于词袋模型的物体识别过程的演示界面

表 5.12　多组物体识别结果

字典数	特征	C1 样本	C2 样本	C3 样本	C4 样本	平均识别率
100	SIFT	15/15	15/15	12/15	14/15	93.33%
100	centrist	15/15	15/15	10/15	15/15	91.67%
100	colorsymcentrist	15/15	15/15	10/15	14/15	90.00%

表 5.13　多组物体识别结果

字典数	特征	C1 样本	C2 样本	C3 样本	C4 样本	平均识别率
200	SIFT	15/15	15/15	10/15	14/15	90.00%
200	centrist	15/15	14/15	13/15	14/15	93.33%
200	colorsymcentrist	15/15	15/15	10/15	13/15	88.33%

分别使用原始图片和粗网格重建后的图片进行 SIFT 特征检测和匹配，其中使用原始图片得到了 63 个特征点，而使用粗网格重建后的图片可以得到 271 个特征点。使用原始图片匹配中有 3 个匹配，而使用粗网格匹配中有 12 个匹配，从中可以看出粗网格重建之后的匹配率显著提高了。

采用代数多重网格方法来对锚图片进行粗网格提取然后进行重建，使用重建后的图片来进行物体识别，结果如表 5.14 所示。其中我们只对锚图片进行了粗网格提取，其他的图片都使用原始图片，实验的其他参数也与表 5.12 和表 5.13 的环境一样，这样就可以突出粗网格提取的图片对物体识别的贡献情况。

表 5.14　通过对 C3 样本提取粗网格重建后进行识别的结果

字典数	特征	C1 样本	C2 样本	C3 样本	C4 样本	平均识别率
100	SIFT	15/15	15/15	13/15	13/15	93.33%
100	centrist	15/15	15/15	15/15	12/15	95%
200	SIFT	15/15	15/15	13/15	14/15	95%
200	centrist	15/15	15/15	15/15	13/15	96.67%

通过对上述表 5.12～表 5.14 的分析，可以发现，随着字典数目的增加，处理时间增加了很多，但是不一定会提高测试结果。也就是较多的字典数目分散了特征的表达能力，降低了平均识别率。按照前面的分析，如对茄子进行 SIFT 匹配，匹配效率不高。而这里茄子的样本在所有的例子中都有较好的识别能力，其原因主要在于其背景虽然复杂，但是包含了较多的特征点，因此特征点越多，检测能力越高。而在锚的图片中，很多是白色背景，因此识别时特征点较少，在这些方法中锚图片的识别率最低。而针对锚图片进行粗网格提取后重建处理，锚图片的识别率显著提高。

5.6　本章小结

本章主要讲叙代数多重网格方法在物体识别中的应用。因为图像检索的核心是物体识别，所以本章先介绍了图像检索。本章分别使用代数多重网格、小波变换和拉普拉斯方法重构图像，然后将不同方法重构的图像作为待检索图像进行了检索，从检索结果可以看出，代数多重网格方法检索效果最好，不论输入哪种模糊程度的图像都能检索出所有的正确图像。接下来本章介绍了物体识别的相关知识，分别介绍了物体识别的研究现状，物体识别的过程，最后对使用代数多重网格重构的图像和没有用该方法的图像进行了物体识别实验，从实验结果来看，使用了代数多重网格方法重构的图像的识别效果要优于没有使用该方法重构的图像识别效果。

附录A　图论基础

人们常称 1736 年是图论历史元年，因为在这一年瑞士数学家欧拉发表了图论的首篇论文——《哥尼斯堡七桥问题无解》，所以人们普遍认为欧拉是图论的创世人。从 19 世纪中叶到 20 世纪中叶，图论问题大量出现，如哈密顿图问题、四色猜想等。这些问题的出现进一步促进了图论的发展。1847 年，克希霍夫用图论分析电网络，成为图论最早应用于工程科学的一个例子。在很长的一段时间内，图论被当成是数学家的智力游戏。一些著名的难题，如迷宫问题、哈密顿环球旅行问题、任务分配问题和地图着色问题等，吸引了众多的学者。图论中许多概念和定理的建立都与这些问题的解决有关。近几十年来，随着计算机科学的发展，图论以更惊人的速度向前发展。

作为描述事物之间关系的手段或工具，图论在诸如计算机科学、物理学、化学、运筹学、信息学、控制论、社会科学以及经济管理、军事、国防、工农业生产等许多领域都得到了广泛的应用。也正是在众多方面的应用中，图论自身才得到了非常迅速的发展，同时越来越受到数学家和工程技术人员的喜爱。随着信息科学的发展，图论的应用越来越广泛，同时图论理论也得到了充分的发展。

A.1　图的基本概念

图结构属于复杂的非线性数据结构，在实际应用中很多问题可以用图来描述。在图结构中，每个元素可以有零个或多个前驱元素，也可以有零个或多个后继元素，也就是说，元素之间的关系是任意的。在树型结构中，数据元素之间存在明显的层次关系，并且每层的元素可能和下一层的多个元素相邻，但只能和上一层的一个元素相邻。而在图结构中，结点之间的关系可以是任意的，图中的任意两个元素之间都可能相邻。图是计算机应用过程中对实际问题进行数学抽象和描述的强有力的工具。

A.1.1　图的定义

无论多么复杂的图都是由顶点和边构成的。采用形式化的定义，图 G(Graph) 由两个集合 V(Vertex) 和 E(Edge) 组成，记为 $G=(V,E)$，其中 V 是顶点的有限集合，记为 $V(G)$，E 是连

接V中两个不同顶点（顶点对）的边的有限集合，记为$E(G)$。

在图G中，如果代表边的顶点对是无序的，则称G为无向图，无向图中代表边的无序顶点对通常用圆括号括起来，用以表示一条无向边。如(v_i,v_j)表示顶点v_i与v_j的一条无向边，显然，(v_i,v_i)和(v_j,v_j)所代表的是同一条边。如果表示边的顶点对是有序的，则称G为有向图，在有向图中代表边的顶点对通常用尖括号括起来，用以表示一条有向边（又称为弧），如$<v_i,v_j>$表示顶点v_i到v_j的一条弧，顶点v_i称为$<v_i,v_j>$的尾，顶点v_j称为$<v_i,v_j>$的头。通常用由尾指向头的箭头形象地表示一条弧。可见$<v_i,v_j>$与$<v_j,v_i>$是两条不同的弧。

A.1.2　图的基本术语

1．端点和邻接点

在一个无向图中，若存在一条边(v_i,v_j)，则称v_i和v_j为该边的两个端点，并称它们互为邻接点。即v_i是v_j的一个邻接点，v_j也是v_i的一个邻接点。

在一个有向图中，若存在一条边$<v_i,v_j>$，则称此边是顶点v_i的一条出边，同时也是顶点v_j的一条入边；称v_i和v_j分别为此边的起始端点（简称起点）和终止端点（简称终点）；称顶点v_i邻接到v_j，并称顶点v_j是v_i的出边邻接点，顶点v_i是v_j的入边邻接点。

2．顶点的度、入度和出度

在无向图中，顶点所具有的边的数目称为该顶点的度。在有向图中，顶点v_i的度又分为入度和出度，以顶点v_i为终点的入边的数目称为该顶点的入度；以顶点v_i为起点的出边的数目称为该顶点的出度。一个顶点的入度与出度的和为该顶点的度。

若一个图中有n个顶点和e条边，每个顶点的度为$d(v_i)(1\leqslant i\leqslant n)$，则有

$$e=\frac{1}{n}\left(\sum_{i=1}^{n}d(v_i)\right)$$

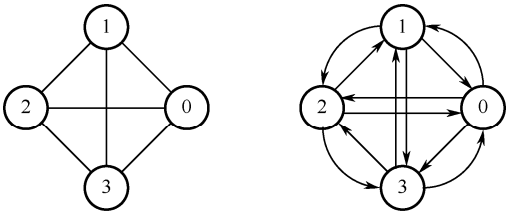
也就是说，一个图中所有顶点的度之和等于边数的两倍。

3．完全图

若无向图中的每两个顶点之间都存在着一条边，有向图中的每两个顶点之间都存在着方向相反的两条边，则称此图为完全图。显然，完全无向图包含有$\frac{n(n-1)}{2}$条边，完全有向图包含有$n(n-1)$条边。例如，图 A.1（a）所示的图是一个具有 4 个顶点的完全无向图，共有 6 条边；图 A.1（b）所示的图是一个具有 4 个顶点的完全有向图，共有 12 条边。

4．子图

设有两个图$G=(V,E)$和$G'=(V',E')$，若V'是V的子集，即$V'\subseteq V$，且E'是E的子集，即$E'\subseteq E$，则称G'是G的子图。



（a） （b）

图 A.1 两个完全图

5．路径和路径长度

在一个图 $G=(V,E)$ 中，从顶点 v_i 到顶点 v_j 的一条路径是一个顶点序列 $(v_i,v_{i1},v_{i2},\cdots,v_{im},v_j)$。若此图 G 是无向图，则边 $(v_i,v_{i1}),(v_{i1},v_{i2}),\cdots,(v_{i(m-1)},v_{im}),(v_{im},v_j)$ 属于 $E(G)$；若此图是有向图，则 $<v_i,v_{i1}>,<v_{i1},v_{i2}>,\cdots,<v_{i(m-1)},v_{im}>,<v_{im},v_j>$ 属于 $E(G)$。路径长度是指一条路径上经过的边的数目。若一条路径上除开始点和结束点相同外，其余顶点均不相同，则称此路径为简单路径。如图 A.1（b）中，(v_0,v_2,v_1) 就是一条简单路径，其长度为 2。

A.2 图和矩阵

A.2.1 图的邻接矩阵

图的邻接矩阵定义：设图 $G=(V,E)$ 的顶点集为 $V(G)=\{v_1,\ v_2,\ v_3,\cdots,\ v_p\}$，用 a_{ij} 表示 G 中顶点 v_i 与 v_j 之间的边数，则 n 阶方阵 $M(G)=(a_{ij})_{p\times p}$ 称为 G 的邻接矩阵（Adjacency Matrix）。也就是说

$$a_{ij}=\begin{cases}1 & \text{若}v_i\text{和}v_j\text{之间有弧或边存在}\\ 0 & \text{若}v_i\text{和}v_j\text{之间没有弧或边存在或}i=j\end{cases}$$

图 A.2 所示的图的邻接矩阵为

$$A(G)=\begin{bmatrix}0&2&1&0\\2&0&0&0\\1&0&0&1\\0&0&1&0\end{bmatrix}$$

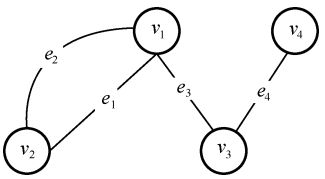


图 A.2 四个顶点的无向图

无向图的邻接矩阵有以下明显的性质：

（1）邻接矩阵是一个对称矩阵，$A[i,j]=A[j,i]$；

（2）若图 G 为无环图，则 $A(G)$ 中第 i 行（列）的元素之和等于顶点的度数；一般说来，图的邻接矩阵比它的关联矩阵小得多，通常图就以其邻接矩阵的形式存储在计算机中；

$$\forall v_i : \sum_{j=1}^{n} A[i,j] = \delta(v_i)$$

（3）图 G 是简单图 $\Leftrightarrow A[i,j] \leqslant 1$ 和 $A[i,i]=0$ 。

A.2.2　图的关联矩阵

图的关联矩阵定义：设 G 是一个简单无向图，其中 V 是图 G 的顶点集，E 是图 G 的边集，$V=\{v_1,v_2,\cdots,v_p\}$，$E=\{e_1,e_2,\cdots,e_q\}$，则定义矩阵 $M(G)=(m_{ij})_{p\times q}$ 称为 G 的关联矩阵（Incidence Matrix），其中 p 为顶点的个数，q 为边数。m_{ij} 表示在关联矩阵中点 i 和边 j 之间的关系。若点 i 和边 j 之间是连着的，则 $m_{ij}=1$；反之，则 $m_{ij}=0$ 。即

$$m_{ij}=\begin{cases}1 & 若v_i关联e_j \\ 0 & 若v_i不关联e_j\end{cases}$$

由上述定义可知，若 $G=(V,E)$ 是无向图，G 的关联矩阵 $M(G)$ 有以下的性质：

（1）每列元素之和均为 2，这说明每条边关联两个顶点 $\forall e_j : \sum_{i=1}^{n} M[i,j]=2$；

（2）每行元素之和是对应顶点的度数 $\forall v_i : \sum_{j=1}^{m} M[i,j]=\delta(v_i)$；

（3）所有元素之和是图各顶点度数的和，也是边数的 2 倍；

（4）两列相同，则对应的两个边是重边；

（5）某行元素全为零，则对应顶点为孤立顶点；

（6）同一个图当顶点或边的编号序列不同时，其对应的 $M(G)$ 仅有行序和列序的差别；

（7）G 是简单图 $\Leftrightarrow M[i,j] \leqslant 1$。

如图 A.2 所示的关联矩阵 $M(G)$为

$$M(G)=\begin{bmatrix}1&1&1&0\\1&1&0&0\\0&0&1&1\\0&0&0&1\end{bmatrix}$$

而针对有向图，对图 G 的每条边 $e=\{v_i,v_j\}$，选择其中一个顶点为正端点，另一个则为负端点。这一过程称为给图 G 的一个定向。为了与无向图的关联矩阵进行区分，对图 G 的一个定向，定义其定向关联矩阵 $C(G)=(c_{ij})_{p\times q}$ 为

$$c_{ij}=\begin{cases}1 & 如果v_i是e_j的正端点 \\ -1 & 如果v_i是e_j的负端点 \\ 0 & 如果v_i和e_j不关联\end{cases}$$

如图 A.3，其关联矩阵 $C(G)$为

$$C(G)\begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}\begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array}$$

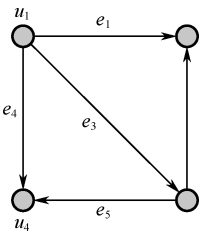


图 A.3　四个顶点的有向图

A.2.3　图的度矩阵

图的度矩阵定义：将每个节点的度放在对角线上，其他元素均设为 0，就构成了图像的度矩阵 $D(G)=[d_{ij}]p\times p$。度矩阵一般用于跟邻接矩阵一起构建图的拉普拉斯矩阵。

$$d_{i,j}=\begin{cases}\deg(v_i) & \text{如果} i=j \\ 0 & \text{其他}\end{cases}$$

如图 A.2 所示的图的度矩阵为

$$D(G)=\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A.2.4　图的拉普拉斯矩阵

图的拉普拉斯矩阵定义：设 $G=(V,E)$ 是 p 阶的简单图，顶点集为 $V=V(G)=\{v_1,v_2,\cdots,v_p\}$，边集为 $E=E(G)=\{e_1,e_2,\cdots,e_q\}$。图 G 的邻接矩阵为一个 p 阶矩阵 $A(G)=(a_{ij})_{p\times p}$。图 G 的度对角矩阵记为 $D(G)=\mathrm{diag}(d(v_1),d(v_2),\cdots,d(v_p))$。则矩阵 $L(G)=D(G)-A(G)$ 称为图 G 的拉普拉斯矩阵。

可以推出 $L(G)=C(G)C(G)^T$，其中 $C(G)^T$ 表示 $C(G)$ 的转置矩阵。由以上可以看出，图的拉普拉斯矩阵与图的定向无关，且是半正定奇异 M 矩阵。

如图 A.2 所示的图的拉普拉斯矩阵为

$$L(G)=\begin{bmatrix} 3 & -2 & -1 & 0 \\ -2 & 2 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

对于一个图，特征值具有 $\lambda_0 \leqslant \lambda_1 \leqslant \cdots \leqslant \lambda_{n-1}$ 关系的拉普拉斯矩阵 L 主要特性有：

（1）L 总是半正定，$\forall i:\lambda_i \geqslant 0;\lambda_0 = 0$，同时有 $x^t L(G)x = \sum_{i:j}(x_i - x_j)^2$；

（2）特征值中的 0 的个数代表了图中连通部分的个数；

（3）L 是一个 M 矩阵；

（4）λ_0 总是 0，因为每个 L 矩阵都有一个特征向量 $v_0 = [1,1,\cdots,1]$；

（5）L 的最小的非零特征值叫作谱隙（Spectral Gap）；

（6）L 的第二个最小的特征值叫作代数连通度，或Fiedler Value；

（7）拉普拉斯算子是函数 f 在 n 维向量空间上的算子：$V \to R$，其中，V 是 G 的顶点集合，并且 $n = |V|$；

（8）当 G 为 k 正规时，标准化拉普拉斯算子为：$l = \frac{1}{k}L = I - \frac{1}{k}A$，其中，$A$ 是邻接矩阵，I 为单位矩阵；

（9）对于一个拥有多个连通部分的图，其 L 矩阵（本身或者经过变换后）是一个块对角矩阵。

A.3　特征值与特征向量

A.3.1　特征值与特征向量的定义

设 A 是 n 阶方阵，若数 λ 和 n 维非零列向量 x，使得 $Ax = \lambda x$ 成立，则称 λ 是方阵 A 的一个特征值，x 为方阵 A 的对应于特征值 λ 的一个特征向量。

已知 $A_{n\times n} = (a_{ij})_{n\times n}$ 以及数 λ，则 $\lambda E - A$ 为 A 的特征矩阵

$$|A - \lambda E| = \begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix}$$

是关于 λ 的一个多项式，称为矩阵 A 的特征多项式。

$$f(\lambda) = |A - \lambda E| = \begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix} = 0$$

称为矩阵 A 的特征方程，其中特征方程 $|A - \lambda E| = 0$ 的根则为 A 的特征值。

A.3.2　特征值与特征向量的求解

（1）$|A - \lambda E| = 0$ 求出 λ 即为特征值；

（2）$Ax = \lambda x \Rightarrow (A - \lambda E)x = 0$ 把得到的特征值 λ_i 代入上式，求齐次线性方程组 $(A - \lambda_i E)x = 0$ 的非零解 x 即为所求特征向量。

齐次线性方程组 $(A-\lambda_i E)x=0$ 的通解 x（去掉零解）即为与 λ_i 对应的全部特征向量。

例如：求 $A=\begin{pmatrix}3 & -1\\ -1 & 3\end{pmatrix}$ 的特征值与特征向量。

$|\lambda E-A|=\begin{vmatrix}\lambda-3 & -1\\ -1 & \lambda-3\end{vmatrix}=(\lambda-2)(\lambda-4)=0$ 特征值为 $\lambda_1=2,\ \lambda_2=4$。

当 $\lambda=2$ 时，解 $(2E-A)X=0$，$\begin{pmatrix}-1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix}$，同解方程组 $x_1=x_2$，得基础解系为 $P_1=\begin{pmatrix}1\\ 1\end{pmatrix}$，$X=k_1\begin{pmatrix}1\\ 1\end{pmatrix}(k_1\neq 0)$ 是 $\lambda=2$ 时全部特征向量。

当 $\lambda=4$ 时，解 $(4E-A)X=0$，$\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix}$，同解方程组 $x_1=-x_2$，得基础解系为 $P_2=(1,-1)$，$X=k_2P_2(k_2\neq 0)$ 是 $\lambda=4$ 时全部特征向量。

A.3.3 特征值与特征向量的性质

性质 1：设 λ 是方阵 A 的特征值，对应的一个特征向量 x。

推广：设 λ 是方阵 A 的特征值，则 λ^k 是 A^k 的特征值；

$\varphi(\lambda)=a_0+a_1\lambda+a_2\lambda^2+\cdots+a_m\lambda^m$ 是 $\varphi(A)=a_0E+a_1A+a_2A^2+\cdots+a_mA^m$ 的特征值。如果 A 可逆，则 $\varphi(\lambda)=a_{-k}\lambda^{-k}+\cdots+a_{-1}\lambda^{-1}+a_0+a_1\lambda+\cdots+a_m\lambda^m$ 是 $\varphi(A)=a_{-k}A^{-k}+\cdots+a_{-1}A^{-1}+a_0+a_1A+\cdots+a_mA^m$ 的特征值。

性质 2：矩阵 A 和 A^T 的特征值相同。

性质 3：设 n 阶方阵 $A=(a_{ij})$ 特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，则

（1）$\lambda_1+\lambda_2+\cdots+\lambda_n=a_{11}+a_{22}+\cdots+a_{nn}$，$a_{11}+a_{22}+\cdots+a_{nn}$ 称为矩阵的迹，$\mathrm{tr}(A)$；

（2）$\lambda_1\lambda_2\cdots\lambda_n=|A|$；

$$f(\lambda)=|\lambda E-A|=(\lambda-\lambda_1)(\lambda-\lambda_2)\cdots(\lambda-\lambda_n)$$
$$=\lambda^n-(\lambda_1+\lambda_2+\cdots+\lambda_n)\lambda^{n-1}+\cdots+(-1)^n\lambda_1\lambda_2\cdots\lambda_n$$
$$=\lambda^n-(a_{11}+a_{22}+\cdots+a_{nn})\lambda^{n-1}+\cdots+(-1)^n|A|。$$

性质 4：设 λ_1,λ_2 是 A 的两个不同特征值，P_1,P_2 是对应于 λ_1,λ_2 的特征向量，则 P_1,P_2 线性无关。

在诸多性质中，值得注意的性质有：

（1）属于不同特征值的特征向量是线性无关的；

（2）属于同一特征值的特征向量的非零线性组合仍是属于这个特征值的特征向量；

（3）矩阵的特征向量总是相对于矩阵的特征值而言的，一个特征值具有的特征向量不唯一，一个特征向量不能属于不同的特征值。

A.3.4 特征值应用

常用特征值来描述和求解谱聚类问题。谱聚类算法建立在谱图理论基础之上，与传统的

聚类算法相比，它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。

该算法首先根据给定的样本数据集定义一个描述成对数据点相似度的亲和矩阵，并且计算矩阵的特征值和特征向量，然后选择合适的特征向量聚类不同的数据点。谱聚类算法最初用于计算机视觉、VLSI 设计等领域，最近才开始用于机器学习中，并迅速成为国际机器学习领域的研究热点。

谱聚类算法建立在图论中的谱图理论基础上，其本质是将聚类问题转化为图的最优划分问题，是一种点对聚类算法，对数据聚类具有很好的应用前景。

谱聚类算法将数据集中的每个对象看作是图的顶点V，将顶点间的相似度量化作为相应顶点连接边E的权值，这样就得到一个基于相似度的无向加权图$G=(V,E)$，于是聚类问题就可以转化为图的划分问题。基于图论的最优划分准则就是使划分成的子图内部相似度最大，子图之间的相似度最小。

虽然根据不同的准则函数及谱映射方法，谱聚类算法有着不同的具体实现方法，但是这些实现方法都可以归纳为下面三个主要步骤：

（1）构建表示对象集的相似度矩阵 W；

（2）通过计算相似度矩阵或拉普拉斯矩阵的前 k 个特征值与特征向量，构建特征向量空间；

（3）利用 k-means 或其他经典聚类算法对特征向量空间中的特征向量进行聚类。

上面的步骤只是谱聚类算法的一个总体框架，由于划分准则、相似度矩阵计算方法等因素的差别，具体的算法实现同样会有所差别，但其本质依然是图划分问题的连续松弛形式。

A.4　最短路径问题

在一个无权的图中，若从一个顶点到另一个顶点存在着一条路径，则称该路径上所经过的边的数目为该路径长度，它等于该路径上的顶点数减 1。由于从一顶点到另一顶点可能存在着多条路径，每条路径上所经过的边数可能不同，即路径长度不同，把路径长度最短（即经过的边数最少）的那条路径叫作最短路径，其路径长度叫作最短路径长度或最短距离。

对于带权的图，考虑路径上各边上的权值，通常把一条路径上所经边的权值之和定义为该路径的路径长度或称带权路径长度。从源点到终点可能不止一条路径，把带权路径长度最短的那条路径称为最短路径，其路径的长度（权值之和）称为最短路径长度或最短距离。

实际上，只要把无权图上的每条边看成是权值为 1 的边，那么无权图和带权图的最短路径和最短距离的定义是一致的。

求图的最短路径的两个问题是求图中某一顶点到其余各顶点的最短路径和图中每一对顶点之间的最短路径。

A.5　图的匹配

设$G=(V,E)$是一个图，$M\subseteq E$，如果M不含环且任意两边都不相邻，则称M为G的一个匹配。G中边数最多的匹配称为G的最大匹配。

对于图 $G=(V,E)$，在每条边 e 上赋一个实数权 $w(e)$。设 M 是 G 的一个匹配，定义 $w(M)=\sum_{e\in m} w(e)$，并称之为匹配 M 的权。G 中权最大的匹配称为 G 的最大权匹配。如果对一切，$e\in E$，$w(e)=1$，则 G 的最大权匹配就是 G 的最大匹配。

A.6 网络流和最大割问题

A.6.1 网络流

流网络 $G=(V,E)$ 是一个有向图，其中每条边 $(u,v)\in E$ 均有一个非负容量 $c(u,v)\geqslant 0$。如果 (u,v) 不属于 E，则假定 $c(u,v)=0$。流网络中有两个特别的顶点：源点 s 和汇点 t。图 A.4 展示了一个流网络的实例（其中斜线左边的数字表示实际边上的流，右边的数字表示边的最大容量）。

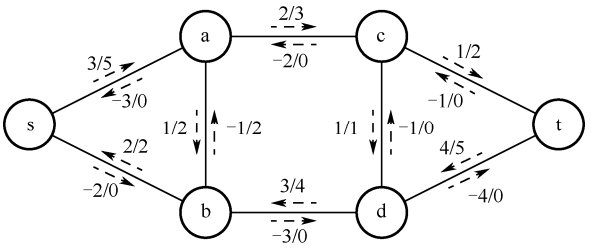


图 A.4 流网络图

对一个流网络 $G=(V,E)$，其容量函数为 c，源点和汇点分别为 s 和 t。G 的流 f 满足下列三个性质。

容量限制：对所有的 $u,v\in V$，要求 $f(u,v)\leqslant c(u,v)$。

反对称性：对所有的 $u,v\in V$，要求 $f(u,v)=-f(v,u)$。

流守恒性：对所有 $u\in V-\{s,t\}$，要求 $\sum f(u,v)=0(v\in V)$。

容量限制说明了从一个顶点到另一个顶点的网络流不能超过设定的容量；反对称性说明了从顶点 u 到顶点 v 的流是其反向流求负所得；而流守恒性说明了从非源点或非汇点的顶点出发的点网络流之和为 0。所以一般的最大流问题就是在不违背上述原则的基础上求出从源点 s 到汇点 t 的最大的流量值，显然这个流量值应该定义为从源点出发的总流量或是最后聚集到 t 的总流量，即流 f 的值定义为 $|f|=\sum f(s,t)(v\in V)$。

A.6.2 最大割问题

最大割问题是组合优化问题中最基本且重要的一类问题，许多其他领域的应用问题最终都会归结为图的最大割问题，这个问题也是 NP-完全问题。它在网络优化、统计物理以及超大规模集成电路中都有广泛的应用。

设 $G=(V,E)$ 代表无向赋权简单图，用 w_{ij} 表示边 v_i,v_j 的权，若顶点 v_i,v_j 之间无边连接，

令 $w_{ij}=0$。最大割问题是指将顶点集 V 分为两个子集 S 和 $\bar{S}$，使得分别在 S 和 $\bar{S}$ 中任意两顶点之间的边 e_{ij} 的权 w_{ij} 的总和最大即 $V(S)=\max\limits_{S\subset V}\sum\limits_{i\in S,j\in\bar{S}}w_{ij}$，引入分割向量 $X\in\{-1,1\}^n$，若顶点 $v_i\in S$，记 $x_i=1$，否则记 $x_i=-1$，可以得到大割问题的一般等价公式为

$$\sum_{i\in S,j\in V/S}w_{ij}=\sum_{i<j}w_{ij}\frac{1-x_ix_j}{2}=\frac{1}{4}\sum_{i,j}(1-x_ix_j)$$

实际上，网络上的最大流等效于最大割的求解。最大割问题在图像分割等方面应用比较广泛，如图割等方法的应用。

附录B　PyAMG的使用

PyAMG 是一个 Algebraic Multigrid（AMG）方法的实现，为使用者提供了一个 python 的接口。PyAMG 是由University of Illinois at Urbana-Champaign计算机科学系的Nathan Bell，Luke Olson和Jacob Schroder开发出来的。Python 是一种面向对象、解释型计算机程序设计语言，由Guido Van Rossum于 1989 年底发明，第一次公开发行于 1991 年。Python 语法简洁而清晰，具有丰富和强大的类库。它常被昵称为胶水语言，它能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。为了方便，本书中使用该软件实现 AMG 方法的部分功能。程序中使用的 pyamg 的版本是 1.0.0，而为了介绍 pyAMG 的各项功能，本章使用的 PyAMG 的版本是 2.0.5。

B.1　PyAMG的特点及与其他包的依赖关系

1．PyAMG 的特点

PyAMG 是一种可重复性的高效的 AMG 求解工具；基于 Python 实现，扩展性强；具有较好的可读性和使用性；依靠 C++的后端速度，执行速度快，且支持 BLAS、LAPACK 等；简单的访问接口，可转换成直观的可重用组件，可移植性较好；支持 Paraview，Matplotlib 等可视化工具。

2．依赖关系

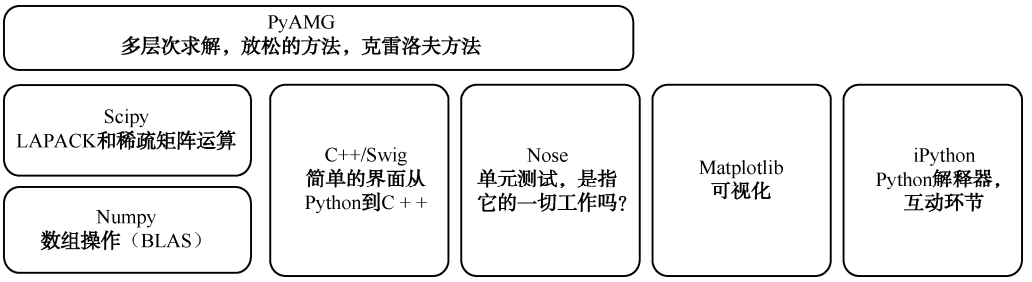


图 B.1　PyAMG 与其他包的依赖关系

B.2　PyAMG的安装

在 Linux 下 PyAMG 的安装步骤：

```
tar xvf pyamg-2.0.5.tar.gz
cd pyamg-2.0.5
sudo python setup.py install
```

在 windows 下 PyAMG 的安装步骤差不多，只是解压文件的方式不同而已。

PyAMG 依赖包 numpy、scipy 和 nose，为了可视化的需要还需要安装 pylab 和 matplotlib。Python 下安装包的方式除了上述的 python setup.py install 之外，常用的还有 pip 的方式，首先需要安装 pip 包。

```
$ wget https://pypi.python.org/packages/source/p/pip/pip-1.5.6.tar.gz
$ tar xzf pip-1.5.6.tar.gz
$ cd pip-1.5.6
$ python setup.py install
$pip install numpy
$pip install scipy
$pip install nose
$pip install pylab
$pip install matplotlib
```

安装完毕 PyAMG 软件后需要对其进行测试，测试的命令如下：

```
Import pyamg
Pyamg.test()
```

测试结果如下：

```
>>> pyamg.test()
Running unit tests for pyamg
NumPy version 1.8.1
NumPy is installed in /usr/lib/python2.7/site-packages/numpy-1.8.1-py2.7-cygwin-1.7.30-i686.egg/numpy
SciPy version 0.14.0
SciPy is installed in /usr/lib/python2.7/site-packages/scipy
Python version 2.7.3 (default, Dec 18 2012, 13:50:09) [GCC 4.5.3]
nose version 1.3.3
PyAMG version 2.0.5.dev
PyAMG is installed in /usr/lib/python2.7/site-packages/pyamg
```

B.3　PyAMG的使用

1．实例 1：开始使用 Python

在 Python 下（$ Python 模式）输入：

```
from pyamg import gallery, smoothed_aggregation_solver
A = gallery.poisson( (50,50), format='csr')
ml = smoothed_aggregation_solver(A)
```

运行结果为：

```
>>> A
<2500x2500 sparse matrix of type '<type 'numpy.float64'>'
        with 12300 stored elements in Compressed Sparse Row format>
>>> ml
multilevel_solver
Number of Levels:        2
Operator Complexity:    1.292
Grid Complexity:         1.170
Coarse Solver:           'pinv2'
  level     unknowns        nonzeros
    0          2500          12300 [77.40%]
    1           425           3591 [22.60%]
```

2．实例 2：稀疏矩阵

```
from scipy.sparse import *
# csr_matrix? # see documentation
from numpy import array
row = array([0,0,1,2,2,2])
col = array([0,2,2,0,1,2])
data = array([1,2,3,4,5,6])
B = csr_matrix( (data,(row,col)), shape=(3,3) )
# B.<tab> # see documentation
print(B.todense())
B = B.tocoo()
```

其中 B.todense()返回矩阵的稠密矩阵形式，而 B.tocoo()则返回矩阵的稀疏矩阵形式。

```
>>>print(B.todense())
[[1 0 2]
 [0 0 3]
 [4 5 6]]
>>> print(B.tocoo())
  (0, 0)        1
  (0, 2)        2
  (1, 2)        3
  (2, 0)        4
  (2, 1)        5
  (2, 2)        6
```

3．实例 3：构建层次结构矩阵

在 Python 中输入：

```
from pyamg import *
ml = smoothed_aggregation_solver(A)
print(ml)
print(ml.levels[0].A.shape)
# Use up-arrow to edit previous command
print(ml.levels[0].P.shape)
```

```
print(ml.levels[0].R.shape)
>>> ml
multilevel_solver
Number of Levels:        3
Operator Complexity:    1.126
Grid Complexity:         1.130
Coarse Solver:           'pinv2'
   level     unknowns        nonzeros
       0        10000          88804 [88.84%]
       1         1156          10000 [10.00%]
       2          144           1156 [ 1.16%]
```

4．实例 4：解决问题

在 iPython 中输入：run task1.5，其中 task1.5 中的脚本语言如下，得到的结果如图 B.2 所示。

```
# task1.5
from numpy import ones
b = ones((A.shape[0],1))
res = []
x = ml.solve(b, tol=1e-8, residuals=res)
from pylab import *
semilogy(res[1:])
xlabel('iteration')
ylabel('residual norm')
title('Residual History')
show()
```

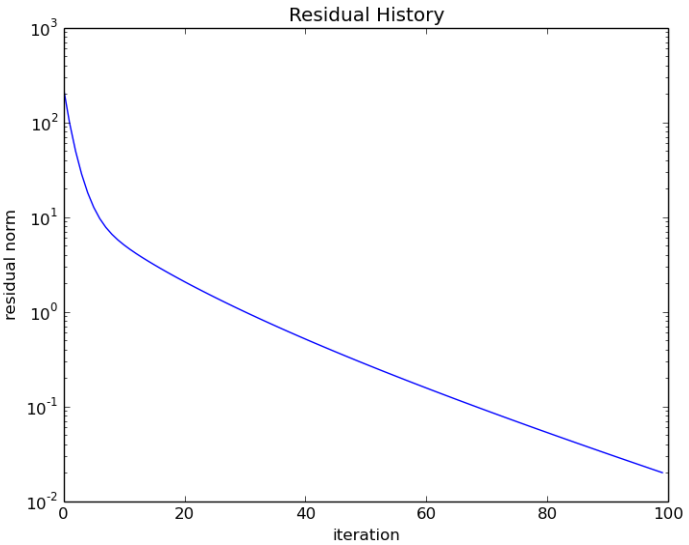


图 B.2　task1.5 的运行结果

5．实例 5：改变 MG 选项

采用先进的粗化和延长平滑选项，在 iPython 中输入如下内容，得到的结果如图 B.3 所示。

```
# task1.6
from pyamg import *
from numpy import ones
ml = smoothed_aggregation_solver(A, \
        strength='evolution',                \
        smooth=('energy', {'degree':4}) )
b = ones((A.shape[0],1))
res = []
x = ml.solve(b, tol=1e-8, residuals=res)
from pylab import *
semilogy(res[1:])
xlabel('iteration')
ylabel('residual norm')
title('Residual History')
show()
```

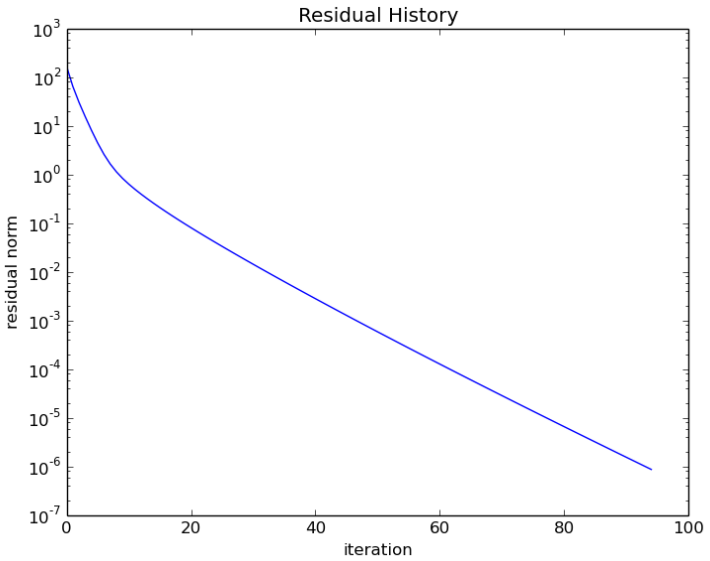


图 B.3　task1.6 的运行结果

6．实例 6：修改层次结构

修改现有的多层次求解对象，替换现有的前/后平滑器与新的用户提供的例行，结果如图 B.4 所示。

```
from pyamg import gallery, smoothed_aggregation_solver
from numpy import ones
from pylab import *
def new_relax(A,x,b):    x[:] +=    0.125*(b - A*x)                    //定义新的松弛方法
A = gallery.poisson( (100,100), format='csr')
b = ones( (A.shape[0],1))
res = []
ml = smoothed_aggregation_solver(A)
```

```
ml.levels[0].presmoother = new_relax                    //使用新的松弛方法
ml.levels[0].postsmoother = new_relax
x = ml.solve(b, tol=1e-8, residuals=res)
semilogy(res[1:])
xlabel('iteration')
ylabel('residual norm')
show()
```

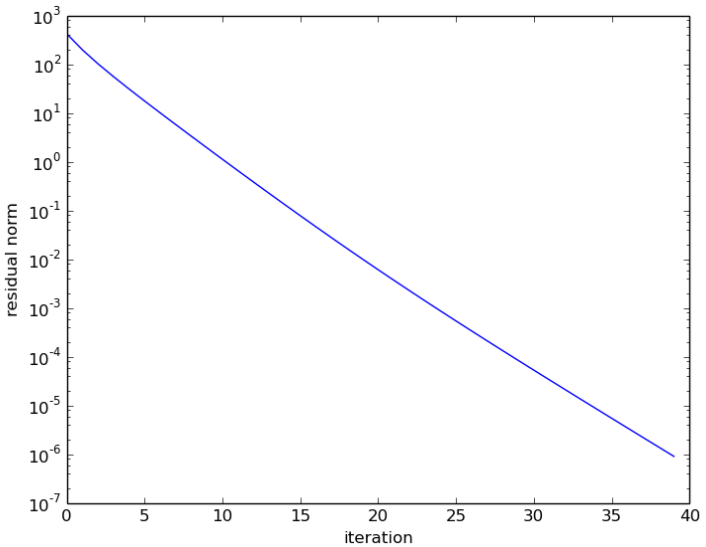


图 B.4　修改松弛方法后的运行结果

7．实例 7：添加平滑方法

首先在 shell 中输入执行命令：

```
$vi ~/pyamg/aggregation/aggregation.py
```

打开 aggregation.py 文件，添加两行，运行结果如图 B.5 所示。

```
    if fn == 'jacobi':
            P = jacobi_prolongation_smoother(A, T, C, B, **kwargs)
      elif fn == 'richardson':
            P = richardson_prolongation_smoother(A, T, **kwargs)
      中增加两行
      elif fn == 'simple':
            P = T-0.2*A*T
```

然后执行以下命令，重新编译 pyamg：

```
$ cd ~/pyamg/
$ sudo python setup.py install
```

从下列代码中可以看出，其中平滑的方法已经替换为前面新增的 simple 方法。其结果如图 B.5 所示。

```
from pyamg import gallery, smoothed_aggregation_solver
from numpy import ones
```

```
from pylab import *
A = gallery.poisson( (100,100), format='csr')
ml = smoothed_aggregation_solver(A, smooth='simple')
b = ones((A.shape[0],1));
res=[]
x = ml.solve(b, tol=1e-8, residuals=res)
semilogy(res[1:])
xlabel('iteration')
ylabel('residual norm')
show()
```

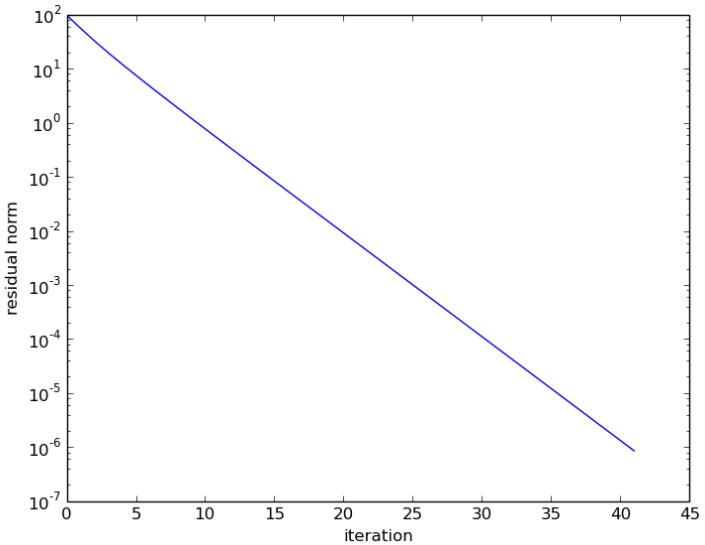


图 B.5　添加平滑方法后的运行结果

8．实例 8：绘制聚集结果

```
# 2D example of viewing aggregates from SA using VTK
from pyamg.aggregation import standard_aggregation
from pyamg.vis import vis_coarse, vtk_writer
from pyamg.gallery import load_example
from pyamg import *
from scipy import *

# retrieve the problem
data = load_example('unit_square')
A = data['A'].tocsr()
V = data['vertices']
E2V = data['elements']
# perform smoothed aggregation
ml = smoothed_aggregation_solver(A,keep=True,max_coarse=10)
b = sin(pi*V[:,0])*sin(pi*V[:,1])
x = ml.solve(b)
```

```
# create the vtk file of aggregates
vis_coarse.vis_aggregate_groups(Verts=V, E2V=E2V,
            Agg=ml.levels[0].AggOp, mesh_type='tri',
            output='vtk', fname='output_aggs.vtu')
# create the vtk file for mesh and solution
vtk_writer.write_basic_mesh(Verts=V, E2V=E2V,
                                    pdata = x,
                                    mesh_type='tri',
                                    fname='output_mesh.vtu')
```

使用 paraview 打开 output_aggs.vtu 和 output_mesh.vtu，可以看到如图 B.6 所示的结果。

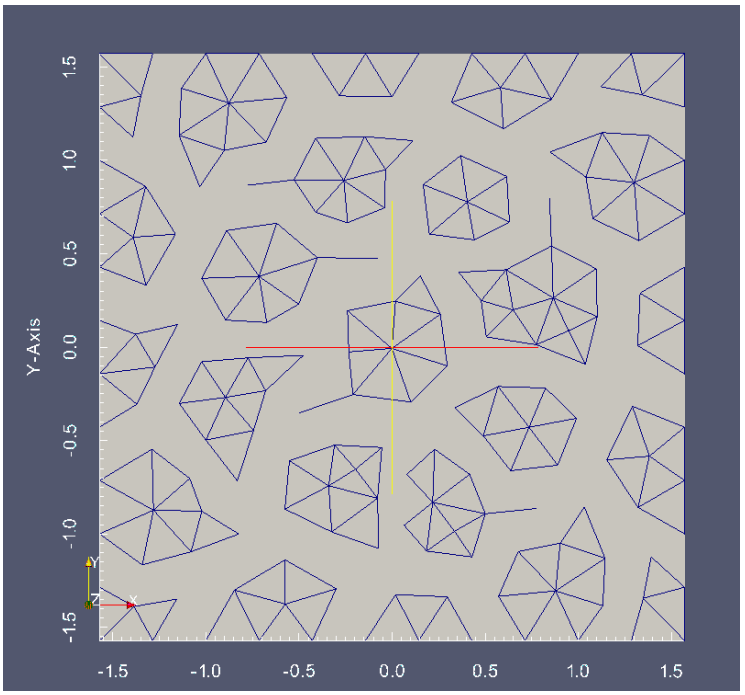


图 B.6　Paraview 中查看聚类结果

9．实例 9：载入矩阵

```
from scipy.io import loadmat
data = loadmat('../../pyamg/gallery/example_data/recirc_flow.mat')
A = data['A'].tocsr()
from pyamg import *
import numpy
b = numpy.ones((A.shape[0],))
ml = smoothed_aggregation_solver(A, symmetry='nonsymmetric',max_coarse=5)
res=[]
x = ml.solve(b,residuals=res)
from pylab import *
semilogy(res)
xlabel('iteration')
```

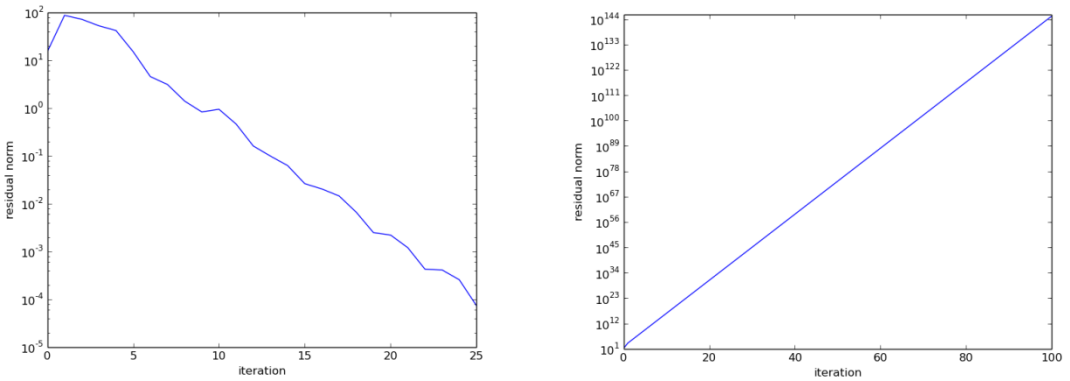
```
ylabel('residual norm')
show()
```

结果如图 B.7（a）所示。

修改其中的代码为：

```
ml = smoothed_aggregation_solver(A, symmetry='symmetric',max_coarse=5)
```

结果如图 B.7（b）所示。



（a）非对称方法　　（b）对称方法

图 B.7　实例 9 的运行结果

10．实例 10：调用 C++

splinalg.h 定义了 C + +：

```
template<class I, class T>
void forwardsolve(const I Ap[], const I Aj[], const T Ax[], T    x[], const T    b[],const I n)
{…}
```

splinalg.i 定义的 C + +接口，用于 SWIG：

```
%define T_INPLACE_ARRAY1( ctype )
%apply ctype * INPLACE_ARRAY {
   ctype       x [ ]
};
%enddef
```

SWIG 编译并创建 Python 接口：

```
$SWIG –c++ -python splinalg.i
```

重新编译和安装 SWIG 目录下的 setup.py。

```
$sudo python setup.py install
```

例如运行调用 C++例程与 SWIG：

```
$Python testbasic.py
```

C++调用：

```
time for one LU solve    = 0.09608 ms
```

修改 precondition.py 并执行的结果如图 B.8 所示。

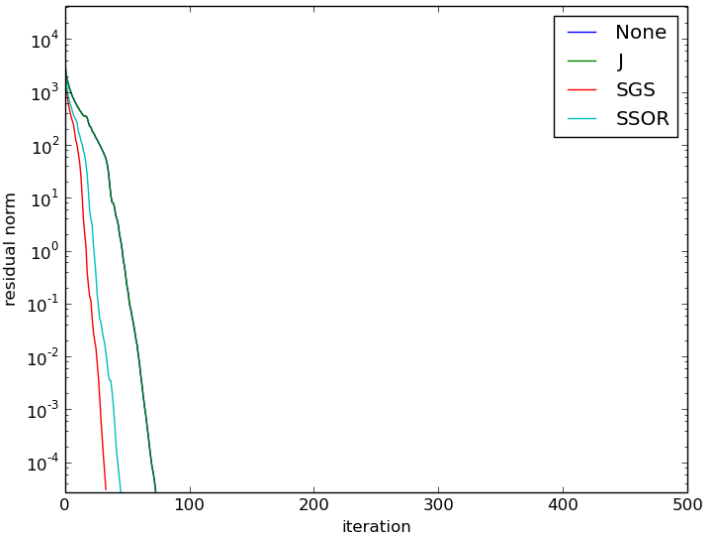


图 B.8　实例 10 运行结果

将 preconditon.py 中相应的行

```
def preconditioner_matvec(L,U):
    def matvec(x):
        return lusolve(L,U,x)
```

修改为 return lusolve_reference(L,U,x)，然后 Python testbasic.py，结果为 time for one LU solve = 32.08 ms。调用 C++代码，速度能提高几百倍。

附录C　源代码

C.1　利用PyAMG进行代数多重网格重建

```
# Illustrates the selection of Coarse-Fine (CF)
# splittings in Classical AMG.

import numpy
from scipy.io import loadmat
from pyamg import ruge_stuben_solver    //引入 pyamg 包
from pyamg.gallery import load_example
data = loadmat('A.mat')    //利用 matlab 处理结果，构建亲和力矩阵

A = data['A']                                                    # matrix
#V = data['vertices'][:A.shape[0]]                        # vertices of each variable
#E = numpy.vstack((A.tocoo().row,A.tocoo().col)).T    # edges of the matrix graph

# Use Ruge-Stuben Splitting Algorithm (use 'keep' in order to retain the splitting)
mls = ruge_stuben_solver(A, max_levels=4, max_coarse=1, CF='RS')//构建粗网格层次
print mls

# The CF splitting, 1 == C-node and 0 == F-node
splitting = mls.levels[0].splitting    //第一层粗网格
B = mls.levels[1].splitting    //第二层粗网格
C = mls.levels[2].splitting    //第三层粗网格
D = mls.levels[3].splitting //第四层粗网格
numpy.savetxt('temp.txt',splitting)     //保存结果到相应矩阵，待 matlab 进行后续处理
numpy.savetxt('B.txt',B)
numpy.savetxt('C.txt',C)
```

C.2 代数多重网格分解和插值过程

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

h=waitbar(0,'process');
cd coaseImageProcess

nsegs=30;
fprintf('\n 图像粗化序列\n');
% input image
image=imread('image/clockA.gif');//读取原始图像
[p,q,r]=size(image);
disp(['image size : ',mat2str([p,q,r]) ]);
nbPixels=p*q;
disp(['number of pixels : ',num2str(nbPixels) ]);
disp('starting affinity matrix A...');
[A,C] = ncut_multiscale(image,nsegs);//调用 ncut 方法中亲和力矩阵的构建函数，构建亲和力矩阵
save temp\A.mat   //保存结果传送到 python 中进行处理
% update waitbar
waitbar(1/4);

cd temp
disp('执行 python!')
!python demo.py

% update waitbar
waitbar(2/4);
% update waitbar
waitbar(3/4);
%  从 python 的结果中获取图像的亲和力数据，构建亲和力矩阵
A = textread('temp.txt');
B = textread('B.txt');
C = textread('C.txt');
%!del A.mat
%!del temp.txt
%!del B.txt
%!del C.txt
cd ..
%%%  归一化到原始图大小
indexA = find(A);
indexB = find(B);
```

```
indexC = find(C);
%%%%%%%%%%%%%%%%%%%%%%%%
%第一层归一化到原始图大小
AA = zeros(size(A));
AA(indexA) = 1;
%第二层归一化到原始图大小
AB = zeros(size(A));
AB(indexA(indexB)) = 1;
%第三层归一化到原始图大小
AC = zeros(size(A));
AC(indexA(indexB(indexC))) = 1;

AA=reshape(AA(1:p*q),p,q);
AB=reshape(AB(1:p*q),p,q);
AC=reshape(AC(1:p*q),p,q);
cd ..
% update waitbar
waitbar(4/4);
delete(h);
% to show the coarse seq
%第一层插值
axes(handles.axes3);
ims=reshape(AA(1:p*q),p,q);
[X,Y]=meshgrid(1:p,1:q);
[x,y,z]=find(ims.*double(image));
Z = griddata(x,y,z,X,Y,'cubic');//cubic 插值，可以换成 linear 或 v4
Aimg = uint8(CrossNaN_Rotate(Z,image));
imshow(Aimg)
%  第二层插值
axes(handles.axes5);
ims1=reshape(AB(1:p*q),p,q);
[X,Y]=meshgrid(1:p,1:q);
[x,y,z]=find(ims1.*double(image));
Z1 = griddata(x,y,z,X,Y,'cubic');
Bimg = uint8(CrossNaN_Rotate(Z1,image));
%imwrite(Bimg,'VasoRsecond.jpg','jpg');
imshow(Bimg)
axes(handles.axes7);
%  第三层插值
ims2=reshape(AC(1:p*q),p,q);
[X,Y]=meshgrid(1:p,1:q);
[x,y,z]=find(ims2.*double(image));
Z2 = griddata(x,y,z,X,Y,'cubic');
Cimg = uint8(CrossNaN_Rotate(Z2,image));
imshow(Cimg)
clearvars -EXCEPT image AA AB AC
save tmpdata
```

C.3　结合区域检测的代数多重网格融合方法

```
close all
clear
clc
ima = double(imread('disk1.gif'));
imb = double(imread('disk2.gif'));

load disk

Amse = (Z2im-ima).*(Z2im-ima);
Bmse = (Z2im2-imb).*(Z2im2-imb);
Ea = extract_curve(ima);
Eb = extract_curve(imb);

Eab = myor(Ea,Eb);
%%%%%%%%%%%%%%%%%5
figure,imshow(Eab,[]);title('myor')

% link the edges
Eab = linkEdge(Eab);
%%%%%%%%%%%%%%%%%%%%%%
Eab = bwmorph(~Eab,'spur',8);
figure,imshow(~Eab,[]);title('spur')

Eab = bwmorph(Eab,'thin',inf);
figure,imshow(~Eab,[]);title('thin')

IM2 = imclearborder(Eab,8);
figure,imshow(IM2,[]);title('imclearborder')
Eab = ~(Eab - IM2);
figure,imshow(Eab,[]);title('after imclearborder')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         区域划分
%  判断是否遍历了图像的所有的区域
% 1 为还有没有遍历的区域  0 为完全遍历
TotalFinishFlag = 0;
%  判断随机种子的有效性
% 1 有效  0  无效
RandSeedValid = 0;
%  每轮判断 TotalFinishFlag 和 RandSeedValid 的参考图像
[Row Col]=size(Eab);
imShow = Eab;
%  每轮产生出来的新区域
imtemp = [];
```

```
% 保存各个区域
imArray = zeros(Row,Col);
% 记录区域数目
SedNumber = 0;
% 产生一个图像，显示
figure,
while(TotalFinishFlag == 0)
    % 产生随机种子
    RandSeedValid = 0;
    RandX = 0;
    RandY = 0;
    while(RandSeedValid ==0 )
        RandX = randi(Row)+1;
        RandY = randi(Col)+1;
        RandSeedValid = SeedValid(RandX,RandY,imShow);
    end
    point = [RandX;RandY];
    SedNumber = SedNumber + 1;
    % 产生这个区域
    % 生长的区域为数值 1
    imtemp = myRegionGrow(Eab,point);
    % 保存这个区域
    imArray(imtemp == 1) = SedNumber;
    % 更新判断的区域
% 将区域生长的区域置为 0
imShow(imtemp==1) = 0;
    imshow(imShow,[]);
    % 判断是否全都遍历完
    TotalFinishFlag = IsOver(imShow);
    if SedNumber> 760
        break;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                       融合阶段
fusionIm = zeros(Row,Col);
% 区域融合
% 总的区域数目是 SedNumber
RegionM = [];
sumA = 0;
sumB = 0;
for i = 1:SedNumber
//判断两个区域的 Amg 统计值，如果 A 图像中的值较大，则选定 A 中的作为结果图像中的候选值，否则选定 B 中的作为结果图像中的候选值
    sumA = sum(sum(Amse(imArray == i)));
```

```
        sumB = sum(sum(Bmse(imArray == i)));
        if sumA<sumB
            fusionIm(imArray == i) = imb(imArray == i);
        else
            fusionIm(imArray == i) = ima(imArray == i);
        end
    end

    figure,imshow(fusionIm,[]),title('区域融合结果')
    % 边界融合方式
    % 边界处理方式为：两幅图像取均值
    MeanIm = (ima+imb)/2;
    fusionIm(imArray == 0) = MeanIm(imArray == 0);
    figure,imshow(fusionIm,[]),title('边界融合结果')
    imwrite(uint8(fusionIm),'fusionDisk.jpg')
```

C.4 laPlace矩阵投影方法

```
    close all
    clear
    clc
    r = 1;
    %im = imread('1.bmp');
    im1=imread('clockB.gif');
    im=resizem(im1,0.2);
    [p q] = size(im);
    pq = p*q;
    imcpy = [];
    for i = 1:size(im,1)
        imcpy = [imcpy im(i,:)];
    end
    imcpy = imcpy';
    % 4-neighbors      = 1
    W = zeros(pq);
    for i = 1:pq
        for j = 1:pq
            W(i,j) = calresult(i,j,q);
        end
    end
    % D
    D = diag(sum(W,2));
    % L
    L = D-W;
    %%%% cal the eig
    [vd,dd]=eig(D);
    [vl,dl]=eig(L);
```

```
[vw,dw]=eig(W);
ED = diag(dd);
EL = diag(dl);
EW = diag(dw);
%%%%%%%%%%%%%
% find the key K
% temp = find(EL + EW(end:-1:1<ED(end:-1:1)));
%%%%%%%%%%%%%
K = 3000;
B = vl(:,1:K);
% lmp
lmp = B'*double(imcpy);
% interplote
Lmp = zeros(pq,1);
Lmp(1:K)=lmp;
% reconstruct
imRon = vl*Lmp;
imRonnew = zeros(p,q);
for i = 1:p
    imRonnew(i,:) = imRon((1+(i-1)*q):i*q);
end
figure,imshow(imRonnew,[])
imwrite (uint8(imRonnew),'112res.png')
```

C.5 结果

图像粗网格序列与重构实验结果如图 C.1 所示。

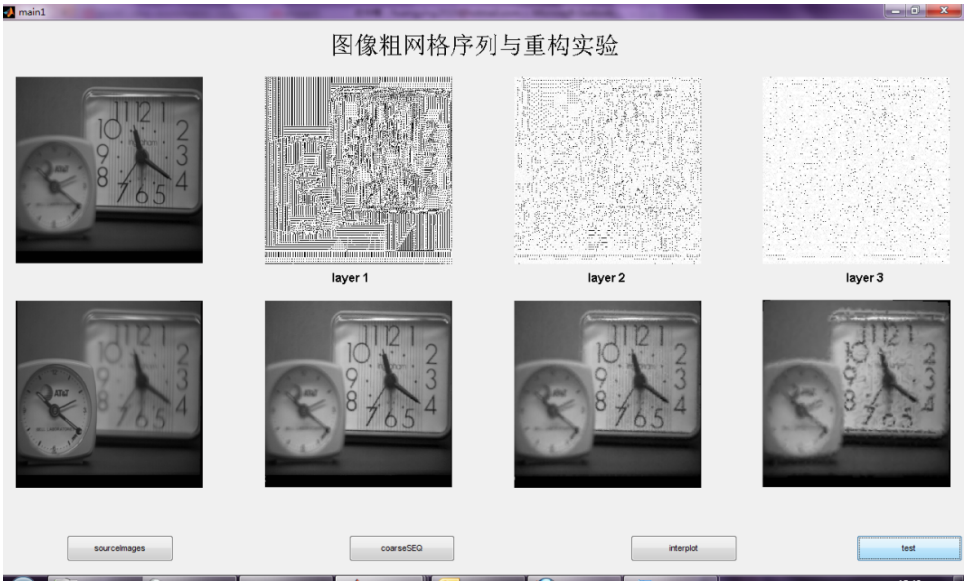


图 C.1　图像粗网格序列与重构实验结果

图像清晰度与 MSE 关系实验结果如图 C.2 所示。

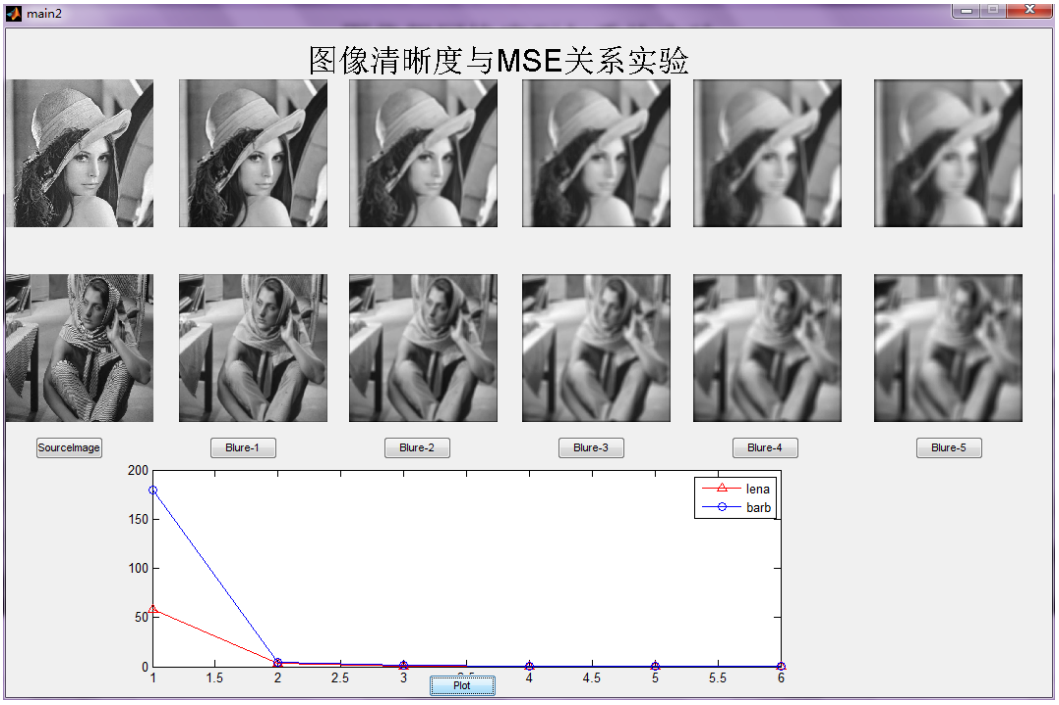


图 C.2　图像清晰度与 MSE 关系实验结果

参考文献

[1] 曾新祥，何江衡，周洁. 偏微分方程的多重网格[J]. 湘潭师范学院学报（自然科学版），2009，31(2):21-23.

[2] 王卫卫，冯象初. 图像处理中扩散方程的快速数值解法[J]. 电子与信息学报，2009，31(7):1736-1740.

[3] 刘朝霞，常谦顺. 由扩散张量导出的各向异性扩散模型的隐式数值模拟[J]. 计算物理，2005，22(4):365-370.

[4] 韩玉兵，束锋，孙锦涛，等. 基于 MG-CG 算法的图像超分辨率重建[J]. 电子学报，2007，35(7):1394-1397.

[5] 邱岳东. 求解线性椭圆型优化控制问题的多重网格方法[J]. 哈尔滨商业大学学报，2014，06(1).

[6] Brandt A, Mccormick S, Ruge J. Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations[R], Inst. Comp. Studies, Colo. State Univ., 1982.

[7] Kimmel R, Yavneh I. An Algebraic Multigrid Approach for Image Analysis[J]. SIAM, 2003, 24(4): 1218-1231.

[8] Acton S T. Multigrid anisotropic diffusion[J]. IEEE Transactions on Image Processing, 1998,7(3): 280-291.

[9] Papandreou G, Maragos P. Multigrid Geometric Active Contour Models[J]. IEEE Transactions on Image Processing, 2007, 16(1): 229-240.

[10] Wang L, Chen Y, Pan X, et al. Level set segmentation of brain magnetic resonance images based on local Gaussian distribution fitting energy[J]. J Neurosci Methods, 2010, 188(2):316-325.

[11] 孙洁. 基于多尺度分析的多重网格法[J]. 宁波大学学报（理工版），2008，21(4):538-541.

[12] Ron Kimmel, Irad Yavneh. An Algebraic Multigrid Approach for Image Analysis[J]. Society for Industrial and Applied Mathematics. 2003, 24(4):1218-1231.

[13] Congli Li, Xiushun Yang, Binbin Chu, Wei Lu, Lulu Pang. A New Image Fusion Quality Assessment Method Based on Conrourlet and ssim[C]. 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT). Chengdu, China. 2010(5):246-249.

[14] 史培林. 代数多重网格法及其在图像重构的应用[D]. 北京：中国科学院，2003:24-38.

[15] Paul M, deZeeuw. A Multigrid Approach to Image Processing[C]. Scale-Space'05 Proceedings of the 5th international conference on Scale Space and PDE Methods in Computer Vision. Springer-Verlag Berlin Heidelberg 2005:396-407.

[16] Radu Stefanescu, Xavier Pennec, Nicholas Ayache. Grid powered nonlinear image registration with locally adaptive regularization[J]. Medical Image Analysis. 2004, 8:325-342.

[17] Mohar B. The Laplacian spectrum of graphs. in Graph Theory[J], Combinatorics, and Applications, 1991: 871-898.

[18] 王勇，谭毅华，田金文. 一种新的图像清晰度评价函数[J]. 武汉理工大学学报，2007，29(3):124-126.

[19] 杨春玲，旷开智，陈冠豪，等. 基于梯度的结构相似度的图像质量评价方法术[J]. 华南理工大学学报（自然科学版），2006，34(9).

[20] 王洪姣. 图像质量评价方法的研究及实现[J]. 计算机技术，2014，03.

[21] 张涛，梁德群，王新年. 基于稀疏表示的图像模糊度评价方法[J]. 计算机工程，2013，39(4):267-271.

[22] 李祚林，李晓辉，马灵玲，等. 面向无参考图像的清晰度评价方法研究[J]. 遥感技术与应用，2011，26(2):239-246.

[23] 梅宁. 自然图像的无参考型质量评价[J]. 模式识别与智能系统，2014，03.

[24] Eskicioglu A M, Fisher P S. Image quality measures and their performance[J]. Communications, IEEE Transactions on, 1995, 43(12): 2959-2965.

[25] 褚江，陈强，杨曦晨. 全参考图像质量评价综述[J]，计算机应用研究，2014，08(1).

[26] Sheikh H R, Wang Z, Bovik A C, et al. Image and video quality assessment research at LIVE[J]. 2003.

[27] Video Quality Experts Group. Final report from the video quality experts group on the validation of objective models of video quality assessment[J]. VQEG, Mar, 2000.

[28] Zhang, S. Chan, S.C.;Qiu, R.D.On the design and implementation of a high definition multi-view intelligent video surveillance system[J].IEEE,2012,08(12):353-357.

[29] Chen Ge-Heng;Li Ya-Jing.Research of Adaptive Frame Difference Moving Target Segmentation Based on MRF[J].IEEE,2013,07(26):221-224.

[30] Huayong Liu; Wenting Meng;Zhi Liu. Key frame extraction of online video based on optimized frame difference[J].IEEE,2012,05(29):1238:1242.

[31] 杨钊. 面向图像分类和识别的视觉特征表达与学习的研究[D]，华南理工大学. 2014.

[32] 赵悦. 基于中层特征表示的图像分类研究[D]，北京交通大学. 2014.

[33] Rassem T, Khoo B. Visual Informatics: Sustaining Research and Innovations[M]. Springer Berlin Heidelberg, 2011, 151-163.

[34] 汪启伟. 图像直方图特征及其应用研究[D]，中国科学技术大学. 2014.

[35] 吕彬. 基于全局直方图及其空间关系的图像检索技术[D]. 太原理工大学，2014.

[36] 赵磊. 基于加权颜色直方图的视频目标跟踪算法研究[D]. 辽宁大学，2014.

[37] 潘永胜. 基于颜色和形状特征的图像检索技术及应用[D]. 太原理工大学，2014.

[38] 陈颖，赵勋杰. 基于 RGB 直方图的运动的目标检测算法[J]. 微计算机信息，2010，11(2).

[39] Shabani A H, Zelek J S, Clausi D A. Human Action Recognition Using Salient Opponent-Based Motion Features[C]. Canadian Conference on Computer and Robot Vision (CRV), 2010,362-369.

[40] Stricker M, Swain M. The capacity of color histogram indexing[C]. Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '94), Seattle,

WA ,1994,704-708.

[41] 王波，柯军. 融合色调和局部特征的军人图像识别方法[J]. 2014，08(08).

[42] 郭曦，周越. 融合颜色矩和码书模型运动目标模型检测算法[J]. 2011，02(16).

[43] 虞雯佳，基于颜色信息的 sift 算法和形状上下文的人脸识别研究[D]. 上海师范大学，2014.

[44] 金伟. 基于融合颜色特征与形状特征的图像检索[D]. 西安电子科技大学，2014.

[45] 郭二军. 一种基于形状特征矩的图像检索算法[D]. 信仰师范学院，2014.

[46] 程显毅，李小燕，任越美. 图像空间关系特征描述[J]. 江南大学学报，2014，12(06).

[47] 吕金建，文贡坚，李德仁，王继阳. 一种新的基于空间关系的特征匹配方法[J]. 2008，08(15).

[48] DHar,R. An analysis of CANNY and LAPLACIAN of GAUSSIAN image filters in regard to evaluating retinal image[J].IEEE,2014,03(06):1-6.

[49] Kamaladhas, M.D. Fingerprint generation of audio signal using Difference of Gaussian[J].IEEE,2012,03(15):276-279.

[50] Sharma,K. Very high resolution image registration based on two step Harris-Laplace detector and SIFT descriptor[J].IEEE,2013,07(4):1-5.

[51] Wolfgang B. Steerable filters and invariance theory[J]. Pattern Recognition Letters, 1994, 15(5):453-460.

[52] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[53] 程德志，李言俊，余瑞星. 基于改进 SIFT 算法的图像匹配方法[J]. 计算机仿真，2011，28(7):285-289.

[54] Pan Xiao. Efficient SIFT descriptor via color quantization[J],IEEE,2014,05(09):1-3.

[55] Nedevschi,S;Peter,I.R.;Mandrut,A. PCA type algorithm applied in face recognition[J]. IEEE, 2012.08(30):167-171.

[56] Rujirakul, K. PFP-PCA: Parallel Fixed Point PCA Face Recognition[J]. IEEE, 01(29):409-414.

[57] Hall D.L., Llinas J. An introduction to multisensor data fusion[J]. Prodceedings of the IEEE. 1997, 85(1):6-23.

[58] Guihong Qu, Dali Zhang, Pingfan Yan. Medical image fusion by wavelet transform modulus maxima[J]. Optics Express. 2001, 9(4):184-190.

[59] 潘泉，于昕，程咏梅，等. 信息融合理论的基本方法与进展[J]. 自动化学报，2003，29(4):599-615.

[60] Jun Kong, Kaiyuan Zheng, Jingbo Zhang, Xue Feng. Multi-focus Image Fusion Using Spatial Frequency and Genetic Algorithm[J]. IJCSNS International Journal of Computer Science and Network Security. 2008, 2(8):220-224.

[61] X.Yang, W.Yang, J.Pei. Different focus points images fusion based on wavelet decomposition[C]. Proceedings of Third International Conference on Information Fusion. Paris, France. 2000(1):3-8.

[62] Li X, He M, Roux M. Multifocus image fusion based on redundant wavelet transform[J]. Image Processing, IET. 2010, 4(4):283-293.

[63] Chin-Shiuh, Shieha, Hsiang-Cheh Huangb, Feng-Hsing Wangc, Jeng-Shyang Pan.

Genetic watermarking based on transform domain techniques[J]. Pattern Recognition. 2004(37): 555-565.

[64] 杨晓慧，李登峰. 基于 Curvelet 多尺度几何分解的快速 LHS 多传感器图像融合[J]. 传感技术学报，2009，22(4):489-494.

[65] 申晓华，艾宝，王婷. 多聚焦图像融合中小波基函数选取的研究[J]. 导弹与制导学报，2006，27(3):231-234.

[66] 胡良梅，高隽，何柯峰. 图像融合质量评价方法的研究[J]. 电子学报，2004，12(12A): 218-221.

[67] 何贵青，陈世浩，田沄. 多传感器图像融合效果综合评价研究[J]. 计算机学报. 2008，31(3):486-492.

[68] Jianbo Shi, Jitendra Malik. Normalized Cuts and Image Segmentation[J]. Pattern Analysis and Machine Intelligence. IEEE Transactions on. 2000, 22(8):888-905.

[69] 高雪妮，玉振明，张军，李陶深. 基于多级分块迭代法的不同聚焦图像融合[J]. 电子学报，2011，59(3):690-694.

[70] 章世秀，高天武，邹修国. 基于 CCS 的边缘检测算子实验[J]. 高校实验室工作研究，2012(3):56-57.

[71] 冯子亮，王翠芹，施关民. 一种基于主动生长的边缘连接算法[J]. 计算机应用研究，2009(26):3954-3956.

[72] X. Li, M. He, M. Roux. Multi-focus image fusions based on redundant wavelet transform[J]. Image Processing, IET. 2010(4):283-293.

[73] 郭敏，任娜. 基于小波变换与块分割的多聚焦图像融合[J]. 云南大学学报（自然科学版），2008(3):251-255.

[74] 时海亮，黄海洋，李刚. 一种基于像素清晰度的 NSCT 域多聚焦图像融合方法[J]. 郑州轻工业学院学报（自然科学版），2012，27(3):53-57.

[75] 刘维杰，刘斌，彭嘉雄. 基于分块的不可分小波多聚焦图像融合[J]. 计算机工程，2011，37(2):205-209.

[76] 陈焕平，何明一，刘卫华. 采用渐进像素提取和形态学检测的多聚焦图像融合[J]. 计算机辅助设计与图形学学报，2011，23(9):1468-1480.

[77] 范文涛，马莉. 一种基于像素清晰度的多聚焦图像融合方法[J]. 郑州轻工业学院学报（自然科学版），2009，24(6):100-103.

[78] Ferzli R, Karam L J. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)[J]. Image Processing, IEEE Transactions on, 2009, 18(4): 717-728.

[79] Lin W, Jay Kuo C C. Perceptual visual quality metrics: A survey[J]. Journal of Visual Communication and Image Representation, 2011, 22(4): 297-312.

[80] 周景超，戴汝为，肖柏华. 图像质量评价研究综述[J]. 计算机科学，2008，35(7):1-4.

[81] 蒋刚毅，黄大江，王旭，等. 图像质量评价方法研究进展[J]. 电子与信息学报，2010，32(1):219-226.

[82] 钱晓凡，范金萍，陈弈鹏. 数码相机模糊图像的计算机修复[J]. 昆明理工大学学报（理工版），2005，30(4).

[83] Dahane G M, Vishwakarma S. Content Based Image Retrieval System[J]. IJEIT, 2012, 1: 92-96.

[84] 马丰原. 基于视觉感兴趣区域的图像质量评价算法研究[D]. 西安电子科技大，2013.

[85] 安东. 小波分解和重构算法[J]，科技信息，2009，Vol.20，pp:64-65.

[86] Malik J. Visual grouping and object recognition[C]. 11th International conference on Image Analysis and Processing, 2001,612-621.

[87] Junfeng Wu; Hongbin Hu;Zhiyang Li.A discriminative spatial bag-of-word scheme with distinct patch[J], IEEE, 2014.07(07):266-271.

[88] Behley, J.; Steinhage, V.;Cremers, A.B.Laser-based segment classification using a mixture of bag-of-words[J], IEEE, 2013, 11(03):4195-4200.

[89] Dung Phan;Chi-Min Oh;Soo-Hyung Kim.Object Recognition by Combining Binary Local Invariant Features and Color Histogram[J],IEEE,2013,11(05):466-470.

[90] Cruz, J.P.N. Dimaala, M.L.;Francisco, L.G.L.Object recognition and detection by shape and color pattern recognition utilizing Artificial Neural Networks[J].IEEE,2013,03(20):140-144.

[91] 李伟生，赵晓霞. 一种基于优化“词袋”模型的物体识别方法[J]. 计算机应用研究，2011，28(9):3288-3290.

[92] Dong-O K, Rae-Hong P. New Image Quality Metric Using the Harris Response[J]. IEEE Signal Processing Letters, 2009, 16(7): 616-619.

[93] Zhang J, Chen Q, Sun Q, et al. A highly repeatable feature detector: improved Harris Laplace[J]. Multimedia Tools and Applications, 2011, 52(1): 175-186.

[94] Soundararajan P, Sarkar S. Analysis of MinCut, average cut and normalized cut measures[C]. Workshop on Perceptual Organization in Computer Vision,2001.

[95] Lindeberg T.Detecting Salient Blob-Like Image Structures and Their Scales with Scale-Space Primal Sketch: A Method for Focus-of-Attention[J]. International Journal of Computer Vision, 1993, 11(3): 283-318.

[96] Liu Jin. Iris image segmentation based on K-means cluster[J].IEEE,2010,12(29):194-198.

[97] Gao Cong. A k-median facility location agent for low-cost service selection in digital community network[J].IEEE,2014.11:174-186.

[98] Esteves, R.M.; Hacker, T. ;Chunming Rong. Cluster analysis for the cloud: Parallel competitive fitness and parallel K-means++ for large dataset analysis[J]. IEEE, 2012, 09(03): 177-184.

[99] Esteves, R.M. ; Hacker, T. ;Chunming Rong. Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets[J].IEEE,2012,12(02):17-24.

[100] Liu Yucheng;Liu Yubin. Incremental Learning Method of Least Squares Support Vector Machine[J].IEEE,2010,05(11):529-532.

[101] Li Suzhen. Supervisory predictive control of weighted least square support vector machine based on Cauchy distribution[J].IEEE,2014,06(31):3523-3526.

[102] Chen, Y.H.;Zheng,Y.F.;Pan, J.F.;Yang, N. A Hybrid Text Classification Method Based on K-Congener-Nearest-Neighbors and Hypersphere Support Vector Machine[J]. IEEE, 2013, 11(16): 493-497.