

# 图像处理

## 1.1 引言

**图像**是由照相机拍摄的真实世界的快照。现实世界本质上是模拟的(即连续的),传统相机拍摄的照片也是模拟的。将其转换为数字(即离散)形式的过程称为**数字化**。数字化过程通常是根据电信号来定义的,电信号在性质上也是模拟的。任何电信号的数字化包括三个步骤:采样、量化和码字生成。**采样**是指以相等的时间间隔或空间间隔检查信号值,并仅存储这些值,同时丢弃其余值,从而基本上离散了信号的时间轴或空间轴。**量化**是根据离散信号幅度的级别(值)以进一步进行处理,同时丢弃剩余值,这本质上离散了信号的幅度轴。**码字生成**是指将称为码字的二进制值分配给在所考虑的特定采样点处保留的每个幅度级别(值)。每单位时间或长度的采样点数量称为**采样率**,而保留的幅度级别数量称为**量化级别**。量化级别的总数决定了表示它们的二进制码字的位数,称为数字信号的**位深度**。一个  $n$  位码字可以代表总共  $2^n$  个级别。例如,一个 3 位码字可以表示总共 8 个值:000、001、010、011、100、101、110、111。构成数字信号的每个离散值称为**样本**。

**数字图像**是一种 2-D 信号,样本分布在图像的宽度和高度上,这些样本称为**像素**,是图像元素的缩写。像素是数字图像的结构单位,类似于构成现实世界物体的分子。像素被可视化为在图像区域上并排排列的矩形单元。每个像素由两个参数标识:位置和值。相对于坐标系测量像素的位置。以法国数学家勒内·笛卡儿的名字命名的 2-D **笛卡儿坐标系**用于测量像素点相对于称为原点的参考点的位置。该位置表示为沿两个正交方向( $x$  轴或水平方向和  $y$  轴或垂直方向)测量的一对偏移。与两个轴(称为主轴)的距离在括号内以坐标的一对顺序数字( $x, y$ )表示,数字表示当前像素分别沿水平和垂直方向偏离原点  $O$  的像素数,它们是整数,因为像素数不能是分数。对于图像处理应用程序,原点  $O$  通常位于左上角, $x$  值从左到右增加, $y$  值从上到下增加(见图 1.1)。显然,表示为  $O$  的原点本身具有坐标  $(0,0)$ 。

像素值表示该点处图像的强度或彩色。根据像素值的不同,图像可以分为三类:二进制(二值)图像、灰度图像和彩色图像,如图 1.2 所示。第一类**二进制图像**(二值图像)是数字

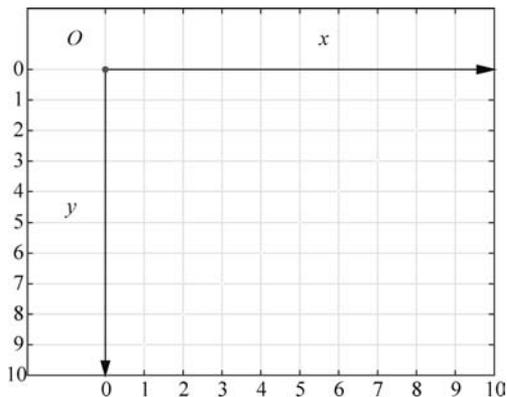


图 1.1 作为像素集合的图像



图 1.2 二进制、灰度和彩色图像

图像,像素值表示为二进制数字,即 0 或 1。此类图像通常包含两种类型的区域,包含值 0 的区域显示为黑色,包含值 1 的区域显示为白色。由于用单个位表示像素值,因此此类图像也称为 1 位图像。第二类称为**灰度图像**,信息使用各种灰度来表示。尽管可以使用任意数量的灰色阴影或灰度级,但使用的标准数字是 256,因为典型的人眼可以分辨出如此多的阴影,且它们可以用 8 位二进制数表示。从 00000000(表示一个极端的黑色)开始到 11111111(表示另一个极端的白色)结束,在这两个极端之间可能有 254 个灰度级别,因此灰度值总数达到 256。第三类称为**彩色图像**,信息是使用各种彩色来表示的。虽然理论上彩色的数量是无限的,但根据人眼分辨彩色的能力,使用的标准数字约为 1670 万,它们可以用 24 位二进制数表示。所有彩色都可表示为三种基色的组合,即红色(R)、绿色(G)和蓝色(B),它们在图像中被称为彩色通道。因此,彩色图像有三个通道,而灰度图像只有一个通道。

数字图像是使用**图像采集设备**创建的,该设备将模拟图像转换为电信号,然后再转换为数字信号。通常使用两种类型的图像采集设备,也称为**输入设备**:扫描仪和数码相机。**扫描仪**通常用于打印文档和打印照片的数字化。它由一个玻璃面板和一个扫描头组成,玻璃面板上的文件面朝下放置,扫描头下方包含一个白光源。当扫描头从左向右、从上向下移动时,光线从文件的所有部分反射出来,落在称为**电荷耦合器件(CCD)**的电子传感器阵列上。CCD 将这些光转换为振幅与光强度成比例的电信号,然后将电信号馈送至**模数转换器(ADC)**,ADC 使用前面描述的数字化过程将电信号转换为数字信号。软件接口可用于指定与数字化相关的参数。像素之间的距离称为图像分辨率,以每英寸(1 英寸=2.54 厘米)点数(dps)为单位,表示像素值的位数称为**位深度**。灰度图像的位深度通常为 8,而彩色图像的位深度为 24。其他参数(如压缩方案和文件格式)将在本节后面进行解释。**数码相机**的工作原理也类似,即来自周围物体的光通过透镜聚焦,落在 CCD 阵列上,CCD 阵列将光转

换为电信号,电信号通过 ADC 转换为数字信号,然后作为图像文件存储在相机内的存储设备中。

一旦图像以数字形式表示,即作为像素集合,则可使用编辑和编程软件工具更改像素值,从而可以通过多种方式修改图像。用于表示修改图像的各种操作的集合术语称为**图像处理**。图像处理可以分为许多类型。第一类为基本操作,只涉及图像的**几何变换**,如剪切、平移、旋转和缩放,而不需要实际修改像素值。第二类涉及使用称为**伽马曲线**的输入-输出关系更改像素值。第三类涉及使用**概率密度函数**(如**直方图**)更改像素值。第四类涉及使用基于局部邻域修改像素值的**核操作**。第五类涉及使用**算术、逻辑和形态学运算**进行修改。第六类涉及**彩色修改**和转换。**彩色模型**能够表示和传达彩色信息。研究表明,我们能看到的**所有彩色**都可以表示为三种基色的不同比例的组合,即红色( $R$ )、绿色( $G$ )和蓝色( $B$ )。这导致了**RGB 彩色模型**的发展,该模型定义了如何使用 RGB 基色生成复合彩色,并适用于彩色光。等量的基色混合在一起时产生**二次色**,即蓝绿色( $C$ )来自  $G$  和  $B$ ,品红色( $M$ )来自  $B$  和  $R$ ,黄色( $Y$ )来自  $R$  和  $G$ 。另一种彩色模型称为**CMY 彩色模型**,用于预测彩色墨水在纸张上的表现,并以蓝绿色、品红色和黄色作为其基色。RGB 和 CMY 彩色模型都可以可视化为彩色立方体,三种基色中的每一种都沿一个正交轴,且立方体内的任何一点都可表示使用三个坐标指定的特定彩色。RGB 和 CMY 彩色模型均是依赖设备的,即产生的实际彩色取决于用于显示彩色的设备的属性,基于人类视觉的固有性质开发与设备无关的彩色模型也被尝试过。研究表明,人类视网膜中的视杆细胞和视锥细胞的存在是人类对亮度(强度)和色度(彩色)感知的原因。因此,已经开发了**色调饱和度值(HSV)彩色模型**,该模型根据三个分量定义彩色,即色调、饱和度和值。其中,色调是人眼可以看到的一组彩色,饱和度是与纯色混合的灰度量,值是每种彩色的亮度。纯色被称为 100% 饱和度的**饱和色**,而混合了不同数量灰度的彩色被称为饱和度较低的去饱和色。色调沿圆周以圆形比例直观表示,并以  $0^\circ$  至  $359^\circ$  的角度测量,而饱和度(以百分比值测量)沿径向表示,从圆周的 100% 降至圆心的 0%。这种表示色调和饱和度的方式通常称为**色轮**。沿垂直于色轮平面的方向测量亮度,可生成 HSV 模型的倒锥图形。其他关于人类视觉系统的当代研究还导致了  $L^* a^* b^*$  **彩色模型**的发展,其中,  $L^*$  表示亮度轴,类似于 HSV 彩色模型的  $V$  轴;彩色则使用称为  $a^*$  的红-绿轴和称为  $b^*$  的蓝-黄轴表示。彩色模型生成的实际彩色的总范围称为模型的**彩色空间**。从一个彩色空间转换到另一个彩色空间的转换公式也已提出,这些公式本质上大多是非线性的。一个有趣的观察结果是 RGB 彩色空间比 CMY 彩色空间大,这意味着并不是屏幕上显示的所有彩色都可打印。 $L^* a^* b^*$  彩色空间是 RGB 彩色空间和 CMY 彩色空间的超集,并使用**色度图**表示。

当图像显示在**输出设备**(如监视器和打印机)上时,图像像素将映射到设备的物理像素上,以便能够看到它们。监视器可以是阴极射线管(CRT)型或液晶显示器(LCD)型。CRT 型显示器由带有荧光涂层屏幕的真空管组成,来自阴极的电子束落在荧光点上,产生被视为发光像素的光。光束通过**光栅扫描**模式在屏幕上从左向右和从上向下移动,逐个激活荧光点。为了在屏幕上获得稳定的图像,扫描模式需要每秒完成 60 次,以便让人眼的**视觉持久性**(PoV)产生所有荧光点同时发光的错觉。LCD 具有独立的像素元件,可使用晶体管电路独立控制。打印机可以是喷墨式或激光式。**喷墨**打印机为每个像素产生一滴液体墨水,使用加热元件或压电晶体从喷嘴激活。**激光**打印机使用感光鼓,在该鼓上使用激光束生成电

荷图像,粉状碳粉颗粒粘附在感光鼓的充电点上,然后这些充电点熔化并融合到纸张上,从而生成图像的打印版本。

当图像作为文件存储在存储设备(如硬盘)上时,通常会占用大量磁盘空间。例如,800×600 像素的 24 位彩色图像大约需要 1.4MB 的磁盘空间。**压缩方案**是指采用软件应用程序减小图像文件大小,以便在存储设备中存储更多图像文件。这对于数码相机这样的便携式设备特别有用。扫描仪也可以使用压缩软件来减小存储在计算机硬盘中的图像文件大小。压缩算法大致分为两类:无损压缩和有损压缩。**无损算法**是指调整文件中的数据,使其在存储设备上存储时占用更少的空间,而不会对文件进行任何永久性更改。**有损算法**则是指通过从文件中删除一些信息,从而减小文件大小。有损压缩的优点是它产生的压缩量比无损压缩大得多,但缺点是它会永久性地降低图像质量。无论压缩过程是什么,都只用于将文件存储在存储设备上,当再次查看图像时,需要使用反向**解压缩**过程恢复文件的原始状态。因此,该应用程序通常被称为编解码器(编码器/解码器)。保存图像的**文件格式**取决于所使用的压缩方案。Windows 本机图像文件格式为 BMP,通常为未压缩文件。当图像质量是一个重要因素而空间要求是次要因素时,可以使用无损压缩算法以 TIFF 或 PNG 格式保存文件;当空间要求比图像质量更重要时,可以使用有损压缩算法以 JPG 格式保存文件。如果涉及非常少量的彩色,则还可以使用 GIF 等 8 位格式保存文件。

## 1.2 工具箱和函数

MATLAB 图像处理函数可分为两类:基本函数和工具箱函数。**基本 MATLAB(BM)**函数为核心媒体处理任务(如 I/O 操作、矩阵操作和绘图例程)提供了一组支持功能。工具箱函数为特殊处理任务(如几何变换、滤波、增强和分割)提供了一组更高级的功能。对于图像处理任务,主要的工具箱是**图像处理工具箱(IPT)**,它为图像处理、分析、可视化和算法开发提供了一整套参考标准算法和 workflow 应用程序。本章讨论的大多数工具箱函数是 IPT 函数的一部分,有少数功能函数取自其他两个工具箱:**信号处理工具箱(SPT)**,它提供分析、预处理和提取均匀和非均匀采样信号特征的功能和应用程序;**小波工具箱(WT)**,它提供了使用基于小波的建模对信号和图像进行分析和合成的功能和应用程序。需要注意的是,一些函数(如基本 I/O 函数)对于 BM 集和工具箱集都是通用的。用于媒体处理任务的 MATLAB 功能在本书中以特定示例的解决方案进行了说明。对于一些特定任务,可能需要来自 BM 集和多个工具箱的函数,这些函数的来源在用于解决方案时会给出。MATLAB 支持的图像文件格式包括 BMP(Windows 位图)、GIF(图形交换格式)、ICO(图标文件)、JPEG(联合摄影专家组)、JPEG 2000、PBM(便携式位图)、PCX(Windows 画笔)、PGM(便携式灰度图)、PNG(便携式网络图形)、PPM(便携式 Pixmap)、RAS(太阳光栅)和 TIFF(标签图像文件格式)。示例中使用的大多数图像都包含在 MATLAB 软件包中,不需要从外部提供。图像示例位于以下文件夹中:(MATLAB root)/toolbox/images/imdata/。

### 1.2.1 基本 MATLAB(BM)函数

BM 函数分为五类:语言基础、数学、图形、数据导入和分析以及编程脚本和函数。下面提供了本章中使用的 BM 函数列表及其层次结构和每种函数的单行说明。BM 集实际上

由数千个函数组成,本章根据本书的范围和内容使用了其中的一个子集。

### 1. 语言基础

- (1) **ceil**: round toward positive infinity,向正无穷方向四舍五入
- (2) **clc**: clear command window,清除命令窗口
- (3) **double**: convert to double precision data type,转换为双精度数据类型
- (4) **eps**: epsilon, a very small value equal to  $2^{-52}$  or  $2.22 \times 10^{-16}$ ,  $\epsilon$ , 一个非常小的值, 等于  $2^{-52}$  或  $2.22 \times 10^{-16}$
- (5) **find**: find indices and values of nonzero elements,查找非零元素的索引和值
- (6) **format**: set display format in command window,在命令窗口中设置显示格式
- (7) **hex2dec**: convert hexadecimal to decimal,将十六进制转换为十进制
- (8) **length**: length of largest array dimension,最大数组维数的长度
- (9) **linspace**: generate vector of evenly spaced values,生成等距值的向量
- (10) **meshgrid**: generates 2-D grid coordinates,生成 2-D 栅格坐标
- (11) **NaN**: not a number, output from operations which have undefined numerical results,不是数字,是具有未定义数值结果的操作的输出
- (12) **num2str**: convert numbers to strings,将数字转换为字符串
- (13) **numel**: number of array elements,数组元素数
- (14) **ones**: create array of all ones,创建所有元素为 1 的数组
- (15) **prod**: product of array elements,数组元素的乘积
- (16) **strcat**: concatenate strings horizontally,水平连接字符串
- (17) **uint8**: unsigned integer 8-bit(0~255),无符号整数 8 位(0~255)
- (18) **unique**: unique values in array,数组中的唯一值
- (19) **whos**: list variables in workspace, with sizes and types,列出工作区中的变量,包括大小和类型
- (20) **zeros**: create array of all zeros,创建全零数组

### 2. 数学

- (1) **abs**: absolute value,绝对值
- (2) **area**: filled area 2-D plot,填充区域 2-D 图
- (3) **atan**: inverse tangent in radians,弧度反正切
- (4) **boundary**: boundary of a set of points,点集的边界
- (5) **cos**: cosine of argument in radians,以弧度表示参数的余弦
- (6) **exp**: exponential,指数
- (7) **fft**: fast Fourier transform,快速傅里叶变换
- (8) **fft2**: 2-D fast Fourier transform,2-D 快速傅里叶变换
- (9) **fftshift**: shift zero-frequency component to center of spectrum,将零频率分量移到频谱中心
- (10) **filter2**: 2-D digital filter,2-D 数字滤波器
- (11) **imag**: imaginary part of complex number,复数的虚部
- (12) **log**: natural logarithm,自然对数

- (13) **magic**: magic square with equal row and column sums, 行与列之和相等的幻方
- (14) **polyshape**: creates a polygon defined by 2-D vertices, 创建由 2-D 顶点定义的多边形
- (15) **real**: real part of complex number, 复数的实部
- (16) **rng**: control random number generation, 控制随机数生成
- (17) **sin**: sine of argument in radians, 以弧度表示参数的正弦
- (18) **sqrt**: square root, 平方根
- (19) **rand**: uniformly distributed random numbers, 均匀分布随机数
- (20) **randi**: uniformly distributed random integers, 均匀分布随机整数
- (21) **sum**: sum of elements, 元素之和

### 3. 图形

- (1) **axes**: specify axes appearance and behavior, 指定轴的外观和行为
- (2) **axis**: set axis limits and aspect ratios, 设置轴限和纵横比
- (3) **bar, bar3**: bar graph, 3-D bar graph, 条形图, 3-D 条形图
- (4) **colorbar**: displays color scale in colormap, 在彩色查找表中显示彩色比例
- (5) **colormap**: color look-up table, 彩色查找表
- (6) **comet**: 2-D animated plot, 2-D 动画图
- (7) **compass**: plot arrows emanating from origin, 绘制从原点发出的箭头
- (8) **contour, contourf**: contour plot of a matrix, 矩阵的等高线图
- (9) **cylinder**: 3-D cylinder, 3-D 圆柱体
- (10) **datetick**: date formatted tick labels, 日期格式的刻度标签
- (11) **ellipsoid**: generates 3-D ellipsoid, 生成 3-D 椭球体
- (12) **errorbar**: line plot with error bars, 带误差条的线图
- (13) **ezplot**: plots symbolic expressions, 绘制符号表达式
- (14) **feather**: plot velocity vectors, 绘制速度矢量图
- (15) **figure**: create a figure window, 创建一个图窗口
- (16) **fill**: fill 2-D polygons, 填充 2-D 多边形
- (17) **fmesh**: 3-D mesh, 3-D 网格
- (18) **fsurf**: 3-D surface, 3-D 曲面
- (19) **gca**: get current axis for modifying axes properties, 获取当前轴以修改轴特性
- (20) **gcf**: current figure handle for modifying figure properties, 用于修改图属性的当前图句柄
- (21) **grid**: display grid lines, 显示网格线
- (22) **histogram**: histogram plot, 直方图
- (23) **hold**: hold on the current plot, 保持当前绘图
- (24) **hsv2rgb**: convert colors from HSV space to RGB space, 将彩色从 HSV 空间转换到 RGB 空间
- (25) **im2double**: convert image to double precision, 将图像转换为双精度
- (26) **image, imagesc**: display image/scaled image from array, 显示数组中的图像/缩放图像

(27) **imapprox**: approximate indexed image by reducing number of colors,通过减少彩色数量来近似索引图像

(28) **imread**: read image from file,从文件读取图像

(29) **imshow**: display image,显示图像

(30) **imwrite**: write image to file,将图像写入文件

(31) **imfinfo**: display information about file,显示文件信息

(32) **ind2rgb**: convert indexed image to RGB image,将索引图像转换为 RGB 图像

(33) **legend**: add legend to axes,给轴添加图例

(34) **line**: create line,创建线

(35) **mesh, meshc**: mesh/mesh with contour plot,网格/带等高线图的网格

(36) **peaks**: sample function of two variables,双变量样本函数

(37) **pie, pie3**: pie chart,3-D pie chart,饼图,3-D 饼图

(38) **plot, plot3**: 2-D line plot,3-D line plot,2-D 线图,3-D 线图

(39) **plotyy**: plot using two y-axis labelings,使用两个 y 轴标签进行绘图

(40) **polar**: polar plot,极坐标图

(41) **polarhistogram**: polar histogram,极坐标直方图

(42) **quiver**: arrow plot,箭头图

(43) **rgb2gray**: convert RGB image to grayscale,将 RGB 图像转换为灰度图像

(44) **rgb2hsv**: convert colors from RGB space to HSV space,将彩色从 RGB 空间转换到 HSV 空间

(45) **rgb2ind**: convert RGB image to indexed image,将 RGB 图像转换为索引图像

(46) **rgbplot**: plot colormap,绘制彩色查找表

(47) **set**: set graphics object properties,设置图形对象属性

(48) **sphere**: generate sphere,生成球体

(49) **stairs**: stairs plot,阶梯图

(50) **stem**: stem plot,茎图

(51) **subplot**: multiple plots in a single figure,单个图形中的多个绘图

(52) **surf**: surface plot,曲面图

(53) **text**: text descriptions in graphical plots,图形绘制中的文本描述

(54) **title**: plot title,绘图标题

(55) **view**: viewpoint specification,视点规范

(56) **xlabel, ylabel**: label x-axis, y-axis, x 轴标签、y 轴标签

(57) **scatter, scatter3**: 2-D scatter plot,3-D scatter plot,2-D 散点图,3-D 散点图

#### 4. 数据导入和分析

(1) **clear**: remove items from workspace memory,从工作区内存中删除项

(2) **disp**: display value of variable,显示变量的值

(3) **imageDatastore**: create datastore for image data,为图像数据创建数据存储

(4) **imread**: read image from graphics file,从图形文件中读取图像

(5) **imwrite**: write image to graphics file,将图像写入图形文件

- (6) **imfinfo**: information about graphics file,有关图形文件的信息
- (7) **load**: load variables from file into workspace,将变量从文件加载到工作区
- (8) **max**: maximum value,最大值
- (9) **mean**: average value,均值
- (10) **min**: minimum value,最小值

#### 5. 程序脚本和函数

- (1) **dir**: list folder contents,列出文件夹内容
- (2) **fullfile**: build full file name from parts,从部件生成完整文件名
- (3) **if,elseif,else**: execute statements if condition is true,如果条件为真,则执行语句
- (4) **continue**: pass control to next iteration of loop,将控制传递给循环的下一个迭代
- (5) **end**: terminate block of code,终止代码块
- (6) **function**: create user-defined function,创建用户定义的函数
- (7) **pause**: stop MATLAB execution temporarily,暂时停止 MATLAB 执行

### 1.2.2 图像处理工具箱(IPT)函数

图像处理工具箱(IPT)函数分为五个不同的类别:导入导出和转换、显示和探索、变换和配准、滤波和增强以及分割和分析。每一个类别又被划分为几个子类别,每个子类别可以包含多个函数。以下提供了本章使用的 IPT 函数列表及其层次结构和各函数的单行说明。IPT 集实际上由 600 多个函数组成,本章根据本书的范围和内容使用了其中的一个子集。

#### 1. 导入、导出和转换

##### 图像类转换:

(1) **gray2ind**: convert grayscale or binary image to indexed image,将灰度或二值图像转换为索引图像

(2) **ind2gray**: convert indexed image to grayscale image,将索引图像转换为灰度图像

(3) **mat2gray**: convert matrix to grayscale image,将矩阵转换为灰度图像

(4) **imbinarize**: convert grayscale image to binary image,将灰度图像转换为二值图像

(5) **imquantize**: quantize image using specified quantization levels,使用指定的量化级别对图像进行量化

(6) **graythresh**: global image threshold using Otsu's method,使用大津方法的全局图像阈值

(7) **otsuthresh**: global histogram threshold using Otsu's method,使用大津方法的全局直方图阈值

(8) **multithresh**: multilevel image thresholds using Otsu's method,使用大津方法的多级图像阈值

##### 彩色:

(9) **rgb2xyz**: convert RGB to CIE 1931 XYZ,将 RGB 转换为 CIE 1931 XYZ

(10) **xyz2rgb**: convert CIE 1931 XYZ to RGB,将 CIE 1931 XYZ 转换为 RGB

(11) **rgb2lab**: convert RGB to CIE 1976 L\* a\* b\*,将 RGB 转换为 CIE 1976 L\* a\* b\*

(12) **lab2rgb**: convert CIE 1976 L\* a\* b\* to RGB,将 CIE 1976 L\* a\* b\* 转换为 RGB

(13) **makeform**: create color transformation structure, 创建彩色转换结构

(14) **applyform**: apply device-independent color space transformation, 应用与设备无关的彩色空间变换

合成图像:

(15) **checkerboard**: create checkerboard image, 创建棋盘图像

(16) **imnoise**: add noise to image, 给图像添加噪声

(17) **phantom**: create head phantom image, 创建头部模型图像

2. 显示和探索

基本显示:

(1) **montage**: display multiple image frames as rectangular montage, 将多个图像帧显示为矩形蒙太奇

(2) **implay**: play movies, videos, or image sequences, 播放电影、视频或图像序列

(3) **warp**: display image as texture-mapped surface, 将图像显示为纹理映射曲面

(4) **imshowpair**: compare differences between images, 比较图像之间的差异

(5) **imfuse**: composite of two images, 两幅图像的组合

交互探索:

(6) **imtool**: open image viewer app, 打开图像观察器应用程序

(7) **imageinfo**: image information tool, 图像信息工具

(8) **impixelinfo**: pixel information tool, 像素信息工具

(9) **impixelregion**: pixel region tool, 像素区域工具

(10) **imdistanline**: distance tool, 聚类工具

(11) **imcontrast**: adjust contrast tool, 对比度调整工具

(12) **imcolormaptool**: choose colormap tool, 选择彩色查找表工具

(13) **immagbox**: magnification box to the figure window, 放大盒到图窗

构建交互式工具:

(14) **imcontrast**: adjust contrast tool, 对比度调整工具

(15) **imcolormaptool**: choose colormap tool, 选择彩色查找表工具

(16) **imnoise**: add noise to image, 向图像添加噪声

(17) **phantom**: create head phantom image, 创建头部模型图像

3. 几何变换和图像配准

公共几何变换:

(1) **imcrop**: crop image, 剪切图像

(2) **imresize**: resize image, 调整图像大小

(3) **imrotate**: rotate image, 旋转图像

(4) **imtranslate**: translate image, 平移图像

通用几何变换:

(5) **imwarp**: apply geometric transformation to image, 对图像进行几何变换

(6) **affine2d**: 2-D affine geometric transformation, 2-D 仿射几何变换

(7) **projective2d**: 2-D projective geometric transformation, 2-D 投影几何变换

**图像配准：**

- (8) **imregister**: intensity-based image registration, 基于强度的图像配准
- (9) **normxcorr2**: normalized 2-D cross-correlation, 归一化 2-D 互相关

## 4. 图像滤波和增强

**图像滤波：**

- (1) **imfilter**: multidimensional filtering of images, 图像多维滤波
- (2) **fspecial**: create predefined 2-D filter, 创建预定义 2-D 滤波器
- (3) **roifilt2**: filter region of interest(ROI) in image, 滤波图像中感兴趣区域(ROI)
- (4) **wiener2**: 2-D adaptive noise-removal filtering, 2-D 自适应消噪滤波
- (5) **medfilt2**: 2-D median filtering, 2-D 中值滤波
- (6) **ordfilt2**: 2-D order-statistic filtering, 2-D 序统计滤波
- (7) **gabor**: create Gabor filter, 创建盖伯滤波器
- (8) **imgaborfilt**: apply Gabor filter to 2-D image, 对 2-D 图像应用盖伯滤波
- (9) **bwareafilt**: extract objects from binary image by size, 根据尺寸从二值图像中提取目标
- (10) **entropyfilt**: filter using local entropy of grayscale image, 基于灰度图像局部熵的滤波

**对比度调整：**

- (11) **imadjust**: adjust image intensity values or colormap, 调整图像亮度值或色图
- (12) **imsharpen**: sharpen image using unsharp masking, 使用非锐化掩模锐化图像
- (13) **histeq**: enhance contrast using histogram equalization, 使用直方图均衡化增强对比度

**形态学操作：**

- (14) **bwmorph**: morphological operations on binary images, 对二值图像形态学操作
- (15) **imclose**: morphologically close image, 形态学闭合图像
- (16) **imdilate**: morphologically dilate image, 形态学膨胀图像
- (17) **imerode**: morphologically erode image, 形态学腐蚀图像
- (18) **strel**: morphological structuring element, 形态学结构元素
- (19) **imtophat**: top-hat filtering, 高帽滤波
- (20) **imbothat**: bottom-hat filtering, 低帽滤波

**去模糊：**

- (21) **deconvblind**: deblur image using blind deconvolution, 使用盲反卷积去图像模糊
- (22) **deconvlucy**: deblur image using the Lucy-Richardson method, 使用露西·理查森方法去图像模糊
- (23) **deconvwnr**: deblur image using the Wiener filter, 使用维纳滤波器去图像模糊

**基于感兴趣区域的处理：**

- (24) **roipoly**: specify polygonal ROI, 指定多边形 ROI
- (25) **imrect**: create draggable rectangle, 创建可拖动的矩形

### 邻域和块处理：

- (26) **blockproc**: distinct block processing for image,用于图像的分块处理
- (27) **col2im**: rearrange matrix columns into blocks,将矩阵列重新排列为块
- (28) **im2col**: rearrange image blocks into columns,将图像块重新排列为列

### 图像算术：

- (29) **imadd**: add two images or add constant to image,两幅图像相加或向图像加常量
- (30) **imabsdiff**: absolute difference of two images,两幅图像的绝对差
- (31) **imcomplement**: complement image,补图像
- (32) **imdivide**: divide one image into another,图像相除
- (33) **imlincomb**: linear combination of images,图像的线性组合
- (34) **immultiply**: multiply two images,两幅图像相乘
- (35) **imsubtract**: subtract one image from another,从一幅图像中减去另一幅图像

### 5. 图像分割和分析

#### 图像分割：

- (1) **imoverlay**: burn binary mask into 2-D image,对 2-D 图像加二值掩模
- (2) **superpixels**: 2-D superpixel oversegmentation of images,图像的 2-D 超像素过分割
- (3) **boundarymask**: find region boundaries of segmentation,查找分割区域的边界
- (4) **labeloverlay**: overlay label matrix regions on 2-D image,将标签矩阵区域叠在 2-D 图像上
- (5) **grayconnected**: select contiguous image region with similar gray values,选择具有相似灰度值的连续图像区域

#### 目标分析：

- (6) **bwboundaries**: trace region boundaries in binary image,在二值图像中确定区域边界
- (7) **bwtraceboundary**: trace object in binary image,在二值图像中确定目标
- (8) **edge**: find edges in intensity image,在亮度图像中查找边缘
- (9) **imgradient**: gradient magnitude and direction of an image,图像的梯度大小和方向
- (10) **hough**: Hough transform,哈夫变换
- (11) **houghlines**: extract line segments based on Hough transform,基于哈夫变换的直线段提取
- (12) **houghpeaks**: identify peaks in Hough transform,识别哈夫变换中的峰值
- (13) **qtdecomp**: quadtree decomposition,四叉树分解
- (14) **imfindcircles**: find circles using circular Hough transform,使用圆形哈夫变换查找圆
- (15) **viscircles**: create circle,创建圆

#### 区域和图像特性：

- (16) **regionprops**: measure properties of image regions,测量图像区域的特性
- (17) **bwarea**: area of objects in binary image,二值图像的目标面积
- (18) **bwconncomp**: find connected components in binary image,在二值图像中查找连通组元
- (19) **bwdist**: distance transform of binary image,二值图像的距离变换

(20) **bwconvhull**: generate convex hull image from binary image, 从二值图像生成凸包图像

(21) **bweuler**: Euler number of binary image, 二值图像的欧拉数

(22) **bwperim**: find perimeter of objects in binary image, 在二值图像中查找目标周长

(23) **imhist**: histogram of image data, 图像数据的直方图

(24) **corr2**: 2-D correlation coefficient, 2-D 相关系数

(25) **mean2**: 2-D average or mean of matrix elements, 矩阵元素的 2-D 平均或均值

(26) **std2**: 2-D standard deviation of matrix elements, 矩阵元素的 2-D 均方差

(27) **bwlabel**: Label connected components in 2-D binary image, 在二值图像中标记连通组元

纹理分析:

(28) **graycomatrix**: create gray-level co-occurrence matrix(GLCM) from image, 从图像创建灰度共生矩阵(GLCM)

(29) **graycoprops**: compute properties of GLCM, 计算 GLCM 的性质

(30) **entropy**: entropy of grayscale image, 灰度图像的熵

图像质量:

(31) **psnr**: peak signal-to-noise ratio(PSNR), 峰值信噪比(PSNR)

(32) **immse**: mean-squared error, 均方误差

(33) **ssim**: structural similarity index(SSIM) for measuring image quality, 用于测量图像质量的结构相似性指数(SSIM)

图像变换:

(34) **dct2**: 2-D discrete cosine transform, 2-D 离散余弦变换

(35) **dctmtx**: discrete cosine transform matrix, 离散余弦变换矩阵

(36) **idct2**: 2-D inverse discrete cosine transform, 2-D 离散余弦反变换

### 1.2.3 信号处理工具箱(SPT)函数

(1) **dct**: discrete cosine transform, 离散余弦变换

(2) **idct**: inverse discrete cosine transform, 离散余弦反变换

### 1.2.4 小波工具箱(WT)函数

(1) **appcoef2**: 2-D approximation coefficients, 2-D 近似系数

(2) **detcoef2**: 2-D detail coefficients, 2-D 细节系数

(3) **dwt2**: single level discrete 2-D wavelet transform, 单级离散 2-D 小波变换

(4) **idwt2**: single level inverse discrete 2-D wavelet transform, 单级离散 2-D 小波反变换

(5) **wavedec2**: 2-D wavelet decomposition, 2-D 小波分解

(6) **waverec2**: 2-D wavelet reconstruction, 2-D 小波重建

## 1.3 导入导出和转换

### 1.3.1 读和写图像数据

图像处理的第一步是读取图像的像素值并将它们存储在内存中,这称为图像采集。可以在图窗口中查看图像,并且可以显示有关图像的信息,例如其尺寸、文件大小、文件格式和位深度。也可以将图像写回磁盘。MATLAB 支持的文件格式已在 1.2 节中提到。在例 1.1 中,BM 函数 `imread` 用于从图像中读取数据,并使用变量名存储数据。通过使用 BM 函数 `imshow` 引用变量名称,图像将显示在图窗口中。BM 函数 `whos` 用于以像素为单位显示图像的高度和宽度,还显示通道数、文件的字节大小和数据类型。字符串 `uint8` 表示 8 位无符号整数,这意味着每个像素使用 1 字节表示,因此值可以为 0~255。BM 函数 `imfinfo` 用于显示有关图像文件的信息,如文件路径、文件修改日期、文件格式、位深度等。BM 函数 `clear` 用于清除先前变量赋值的内存,BM 函数 `clc` 用于清除现有文本和命令行的工作区。BM 函数 `imwrite` 允许将剪贴板中的图像保存到文件中。使用的文件格式是 JPEG,这是一种有损格式,可以在保存过程中指定质量因子。质量因子越小,图像质量越差,文件尺寸越小。BM 函数 `figure` 用于创建一个新的图窗口来显示图像,而 BM 函数 `subplot` 通过指定用于显示多幅图像的行数和列数,将单个图窗口划分为多个分区或单元格。BM 函数 `title` 用于在单个单元格的顶部显示一个标题(见图 1.3)。

**例 1.1** 编写一个程序来读取图像,显示有关它的信息并以不同的文件格式将其写回。

```
clear; clc;
x = uint8(1000)           % returns 255
a = imread('peppers.png');
whos
imfinfo('peppers.png')
imwrite(a, 'p10.jpg', 'quality', 10);
imwrite(a, 'p5.jpg', 'quality', 5);
imwrite(a, 'p0.jpg', 'quality', 0);
b = imread('p10.jpg');
c = imread('p5.jpg');
d = imread('p0.jpg');
figure,
subplot(221), imshow(a); title('original PNG');
subplot(222), imshow(b); title('JPG quality 10');
subplot(223), imshow(c); title('JPG quality 5');
subplot(224), imshow(d); title('JPG quality 0');
```

一般的 2-D 矩阵也可以显示为图像。矩阵的元素可用作当前彩色查找表的索引,这是一个包含用于显示图像的彩色列表的表格。BM 函数 `colormap` 用于指定名为 `jet` 的包含 64 种彩色的彩色查找表的名称。BM 函数 `colorbar` 用于显示当前彩色查找表中的所有彩色。彩色查找表将在 1.3 节中详细讨论。在例 1.2 中,BM 函数 `image` 用于将矩阵显示为图像,通过使用矩阵的值来指定它们要显示的彩色。因此,彩色查找表中的彩色 0~6.4 用于显示矩阵。BM 函数 `imagesc` 用于缩放值以覆盖彩色查找表中整个值的范围。在这种情况下,这

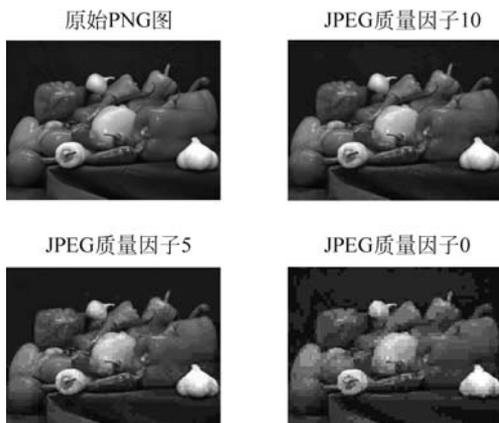


图 1.3 具有不同质量因子的 PNG 和 JPEG 格式的图像

些值大约按比例缩放 10,以便使用彩色值 10 显示值 1,使用彩色值 20 显示值 2,以此类推,直到使用彩色值 64 显示值 6.4。BM 函数 `axis` 用于指定当前轴框的形状为方形(见图 1.4)。

**例 1.2** 编写一个程序,通过指定彩色查找表将 2-D 矩阵显示为图像。

```
clear; clc;
a = [1, 3, 4 ; 2, 6.4, 0 ; 0.5, 5, 5.5];
colormap jet
subplot(121), image(a); colorbar; axis square;
subplot(122), imagesc(a); colorbar; axis square;
```

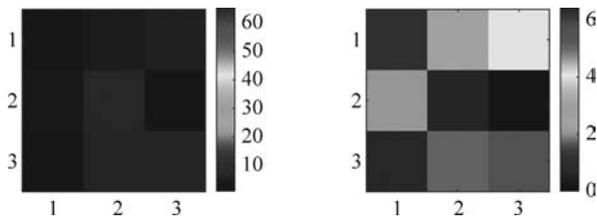


图 1.4 例 1.2 的输出

### 1.3.2 图像类型转换

彩色图像具有三个彩色通道,分别包含红色(R)、绿色(G)和蓝色(B)信息。这三种彩色称为基色,因为可以通过组合不同百分比的这些彩色来生成所有其他彩色。可以将图像的三个彩色通道分开并分别显示。由于对于彩色图像,每个像素占用 24 位空间,因此三个彩色通道各分配 8 位空间。这意味着每个彩色通道的行为就像灰度图像,强度范围从 0(没有彩色)到 255(彩色以全强度存在)。表示全强度 255 的条件称为饱和度。因此,饱和红色由三元组(255,0,0)表示,饱和绿色由(0,255,0)表示,饱和蓝色由(0,0,255)表示。三个不同强度的彩色通道的叠加创造了对彩色的感知。图像中存在特定基色的部分由相应彩色通道中较亮的区域反射,而没有基色的区域由相应通道中较暗的区域反射。例如,图像红色部分在红色通道中显示为白色区域(看起来像灰度图像)。等量的两种基色产生称为二次色

的彩色。红色和绿色产生黄色  $Y(255,255,0)$ ，绿色和蓝色产生蓝绿色  $C(0,255,255)$ ，红色和蓝色产生品红色  $M(255,0,255)$ 。在 MATLAB 中，彩色图像由三个维度表示：高度(图像中的行数)、宽度(图像中的列数)和通道(基色  $R,G,B$ )。在例 1.3 中，显示了一幅彩色图像及其三个彩色通道。请注意，每个彩色通道都显示为灰度图像，因为它们包含 8 位空间信息。冒号(:)运算符表示每个通道从开始到结束的值范围，冒号本身仅表示所有当前值。将 3-D 彩色图像读取并存入变量后，第三维的值为 1 表示  $R$  通道，值为 2 表示  $G$  通道，值为 3 表示  $B$  通道。百分比(%)符号表示注释(见图 1.5)。

**例 1.3** 编写一个程序，读取彩色图像并分别显示各彩色通道。

```
clear; clc;
a = imread('peppers.png');
ar = a(:,:,1);           % red channel
ag = a(:,:,2);           % green channel
ab = a(:,:,3);           % blue channel
subplot(221), imshow(a); title('RGB');
subplot(222), imshow(ar); title('R');
subplot(223), imshow(ag); title('G');
subplot(224), imshow(ab); title('B');
```

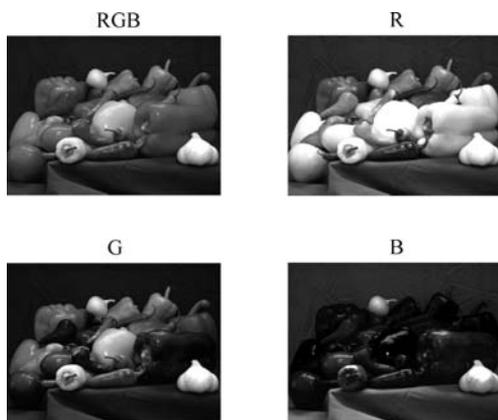


图 1.5 例 1.3 的输出

如前所述，图像通常可以分为三种类型：彩色、灰度和二进制。已有一些成熟的方法可以将一种图像类型转换为另一种图像类型。彩色图像具有三个基色  $R,G,B$  矩阵，每个矩阵中每个像素占用 8 位。另一方面，灰度图像仅由每个像素为 8 位的单个矩阵组成。因此，要将彩色图像转换为灰度图像，需要将三个原矩阵合并为单个矩阵。最简单的方法是取每个像素的  $R,G,B$  值的平均值。但是，这种简单化的方法存在两个问题：(1) 多种彩色可能映射到相同的灰度强度。事实上，所有的三基色都将转换为相同的强度  $(1/3) \times 255$ ，所有的二次色也都将转换为相同的强度  $(2/3) \times 255$ ，这将导致图像的版本降级，其中细节可能会丢失。(2) 人眼对三基色的重视程度不一。研究表明，人类视觉对可见光谱的黄绿色范围最敏感，对蓝色范围最不敏感。考虑到上述因素，国际照明委员会(其缩写 CIE 源自其法语名称 Commission Internationale de l'Éclairage)于 1931 年提出了可接受的彩色图像到灰度图像转换的方案(CIE, 1931)，将强度  $Y$  建模为比例 3 : 6 : 1 的  $RGB$  分量的加权总和。



$$Y = 0.2989R + 0.5870G + 0.1140B$$

要将灰度图像转换为二值图像,需要将特定的灰度强度指定为二值化阈值( $T$ ),低于该阈值的所有像素值将转换为黑色,高于该阈值的所有像素值将转换为白色。阈值越低,图像越多的像素将转换为白色,阈值越高,越多的像素将变为黑色。阈值通常指定为 $0\sim 1$ 的分数或指定为 $0\sim 100$ 的百分比值。阈值为 $k$ 表示 $0\sim k\times 255$ 的所有值都转换为黑色,从 $k\times 255+1$ 到 $255$ 的其余值转换为白色。在例 1.4 中,IPT 函数 `rgb2gray` 用于使用上述等式将 RGB 彩色图像转换为灰度图像。IPT 函数 `imbinarize` 用于通过指定阈值将灰度图像转换为二值图像。使用了两个不同的阈值  $T=0.3$  和  $T=0.5$ ,在第一种情况下, $0\sim 0.3\times 255=76$  的所有像素值都转换为黑色,而  $77\sim 255$  的所有值都转换为白色;对于第二种情况, $0\sim 127$  的所有值都转换为黑色,而  $128\sim 255$  的值显示为白色(见图 1.6)。

**例 1.4** 编写一个程序,转换一幅彩色图像到灰度图像和二值图像。

```
clear; clc;
a = imread('peppers.png');
b = rgb2gray(a);
c = imbinarize(b, 0.3);
d = imbinarize(b, 0.5);
figure,
subplot(221), imshow(a); title('color');
subplot(222), imshow(b); title('grayscale');
subplot(223), imshow(c); title('binary T = 30%');
subplot(224), imshow(d); title('binary T = 50%');
```

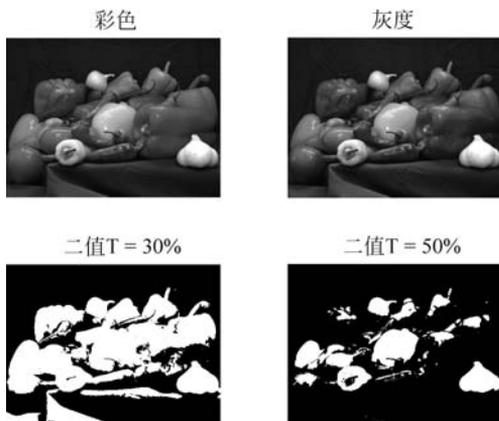


图 1.6 例 1.4 的输出

由于不同的阈值生成不同的图像,自然会出现最佳或最优阈值的问题。Nobuyuki Otsu 提出了一种方法(Otsu,1979),称为**大津方法**(Otsu's method),即从灰度图像自动计算阈值,从而最小化图像直方图中黑白像素的类内方差。直方图将在 1.6 节中讨论。如果  $t$  是阈值强度水平, $\sigma_1$  和  $\sigma_2$  是阈值两侧直方图值的标准方差,则大津方法将阈值固定在一个点,此时方差的加权总和  $q_1\sigma_1^2 + q_2\sigma_2^2$  是最小值。 $q_1$  和  $q_2$  是权重,如果  $p(i)$  是第  $i$  个强度级别的像素数, $L$  是强度级别的总数,则:



$$q_1 = \sum_{i=1}^L [p(i)]$$

$$q_2 = \sum_{i=i+1}^L [p(i)]$$

类均值定义如下：

$$\mu_1 = \sum_{i=1}^L [i \cdot p(i)/q_1]$$

$$\mu_2 = \sum_{i=i+1}^L [i \cdot p(i)/q_2]$$

类内方差定义如下：

$$\sigma_1^2 = \sum_{i=1}^L [(i - \mu_1)^2 \cdot p(i)/q_1]$$

$$\sigma_2^2 = \sum_{i=i+1}^L [(i - \mu_2)^2 \cdot p(i)/q_2]$$

除了生成二值图像的单个阈值外,还可以通过计算多个阈值,从而生成具有两个或三个灰度级的图像。在例 1.5 中,IPT 函数 `graythresh` 用于从灰度图像计算最佳阈值,以便使用大津方法转换为二进制图像。另外,IPT 函数 `otsuthresh` 可用于通过大津方法使用直方图计数计算阈值。IPT 函数 `multithresh` 可用于使用大津方法创建指定数量的多个阈值。这些阈值可以提供给 IPT 函数 `imquantize` 以生成具有多个灰度级的图像。在例 1.5 中,前两种情况使用大津方法计算单个阈值,该方法使用黑色代表 0~100 级,白色代表 101~255 级。在第三种情况中,使用三个灰度阴影表示图像的两个阈值:黑色用于 0~85 级,50%灰度或 128 用于 86~171 级,白色用于 172~255 级。在第四种情况下,计算了三个阈值,用于将图像分成四个灰度:0~49 级为黑色,50~104 级为 33.33%灰度( $0.3333 \times 255 = 85$ ),105~182 级为 66.66%灰度( $0.6666 \times 255 = 170$ ),183~255 级为白色(见图 1.7)。1.4 节将讨论如何从显示的图像中读取实际像素值。

**例 1.5** 编写一个程序,使用自动计算的单个和多个阈值将灰度图像转换为二值图像。

```
clear; clc;
a = imread('peppers.png');
b = rgb2gray(a);
t1 = graythresh(b);
h = imhist(b);
t2 = otsuthresh(h);
c = imbinarize(b, t1);
d = imbinarize(b, t2);
t1, t2
t3 = multithresh(a, 2);
t4 = multithresh(a, 3);
e = imquantize(b, t3);
f = imquantize(b, t4);
figure,
subplot(221), imshow(c); title('graythresh T = [0 101]');
subplot(222), imshow(d); title('otsuthresh T = [0 101]');
subplot(223), imshow(e, []); title('T = [0 86 172]');
subplot(224), imshow(f, []); title('T = [0 50 105 183]');
```

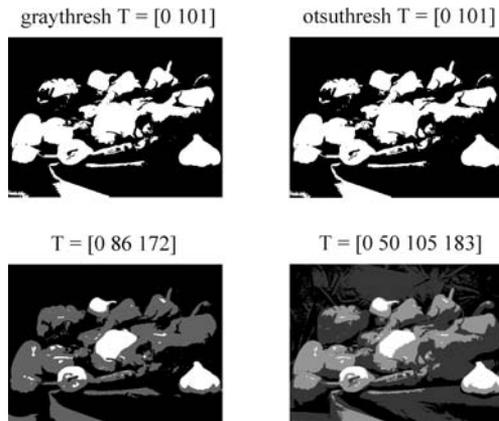


图 1.7 例 1.5 的输出

量化是用一组离散值(也称为级别)替换连续变化函数的过程。对于图像,量化级别也称为灰度级别,通常通过指定生成给定灰度级别所需的位数来表示。典型值包括 8 位(256 级)、6 位(64 级)、4 位(16 级)等直到 1 位,这时图像使用两个级别表示——黑色和白色,因此称为二值图像。在例 1.6 中,BM 函数 `round` 用于将像素值四舍五入到最接近的整数。当图像除以 255 时,0~127 的所有像素值都被舍入为 0,128~255 的值被舍入为 1。再将这些舍入值乘以 255 时,整个像素值集被分为两个离散的值: 0 和 255,从而生成具有两个灰度级(二值)的图像。当图像除以 127 时,0~63 的所有像素值都被舍入为 0,64~190 的值被舍入为 1,并且 191~255 的值被舍入为 2。再将这些舍入值乘以 127 后,整个像素值集被分成三个离散值: 0、127 和 255,从而产生具有三个灰度级的图像。当图像除以 85 时,0~42 的所有像素值四舍五入为 0,43~127 的值四舍五入为 1,128~212 的值四舍五入为 2,213~255 的值四舍五入为 3。再将这些四舍五入的值乘以 85 后,整个像素值集被分为四个离散值: 0、85、170 和 255,从而生成具有四个灰度级的图像。以类似的方式,将图像除以 63 时,将像素值集分为五个离散值: 0、63、126、189 和 252。此外,当像素除以 51 时,将像素值集分为六个离散值: 0、51、102、153、204 和 255。BM 函数 `unique` 用于从修改后的图像中返回这些唯一的离散值(见图 1.8)。

**例 1.6** 编写一个程序,降低灰度图像的量化级别。

```
clear; clc;
a = imread('peppers.png');
b = rgb2gray(a);
c = round(b/51) * 51;           % 6 levels
d = round(b/63) * 63;         % 5 levels
e = round(b/85) * 85;         % 4 levels
f = round(b/127) * 127;       % 3 levels
g = round(b/255) * 255;       % 2 levels
subplot(231), imshow(b); title('original');
subplot(232), imshow(c); title('6 levels');
subplot(233), imshow(d); title('5 levels');
subplot(234), imshow(e); title('4 levels');
subplot(235), imshow(f); title('3 levels');
```

```
subplot(236), imshow(g); title('2 levels');
unique(c), unique(d), unique(e), unique(f), unique(g)
```

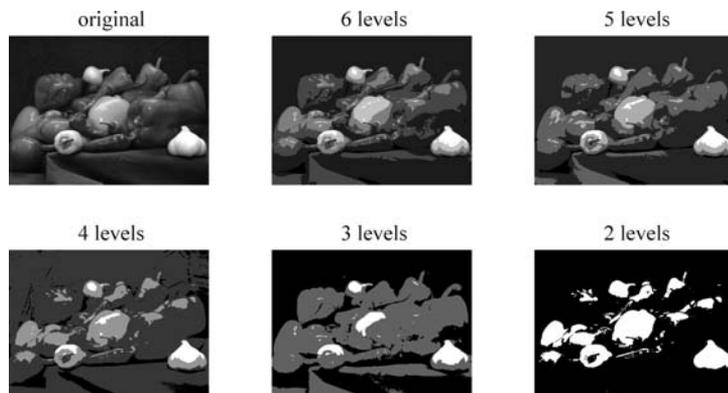


图 1.8 例 1.6 的输出

索引图像是彩色图像,其像素值不包含实际彩色信息而仅包含索引号。通过在内存中查找具有一系列行和四列的表(称为彩色查找表(CLUT)),将索引号转换为彩色值。第一列按顺序包含行的索引号,而其他三列包含每个索引号的 RGB 值。当要显示图像时,使用该表将图像中的索引号转换为彩色值,并生成相应的像素。要显示索引图像,需要指定相应的 CLUT。CLUT 通常包含 64、128 或 256 种彩色,但也可以使用较小的彩色子集来显示图像的近似版本。当使用包含比原始彩色更少彩色的 CLUT 显示图像时,将使用称为抖动的过程来近似 CLUT 中不存在的彩色。抖动是一种混合可用彩色以产生新彩色的过程。它通过改变现有彩色的分布以模拟彩色表中不存在的彩色,该方式类似于通过改变黑点的分布以模拟通常在印刷介质中看到的灰色阴影,如图 1.9 所示。

可以使用包含与原始彩色图像不同彩色数量的彩色查找表来显示索引图像。如果新图像中的彩色数量更多,则原始彩色集将显示在图像中。如果新图像中的彩色数量较少,则使用原始彩色集的子集显示图像。抖动可用于通过模拟新图中不存在的彩色来提高图像质量。BM 函数 `imapprox` 用于显示彩色数量减少的图像的近似版本。

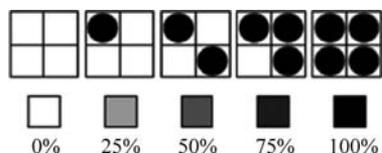


图 1.9 抖动

有两个选项可用: `dither` 选项用于产生抖动以更改变可用彩色的分布以模拟彩色查找表中不存在的彩色,而 `nodither` 选项仅用于使用彩色查找表中可用的彩色显示图像。在例 1.7 中,首先使用具有完整 128 种彩色集的彩色查找表显示索引图像。接下来,将彩色查找表中的彩色数量减少到 8 种,并使用新图显示相同的图像,首先启用抖动,然后禁用抖动。用侧面的彩色条显示彩色查找表中可用的彩色。在第一种情况下,彩色条显示所有 128 种彩色,在第二种和第三种情况下,显示相同的一组 8 种彩色(见图 1.10)。

**例 1.7** 编写一个程序,使用彩色减少的 CLUT 显示索引图像。

```
clear; clc;
load trees;
[Y, newmap] = imapprox(X, map, 8, 'dither');
[Z, newmap] = imapprox(X, map, 8, 'nodither');
```

```

a = subplot(131), image(X);
colormap(a, map); colorbar; axis square; title('original');
b = subplot(132), image(Y);
colormap(b, newmap); colorbar; axis square; title('with dither');
c = subplot(133), image(Z);
colormap(c, newmap); colorbar; axis square; title('without dither');

```

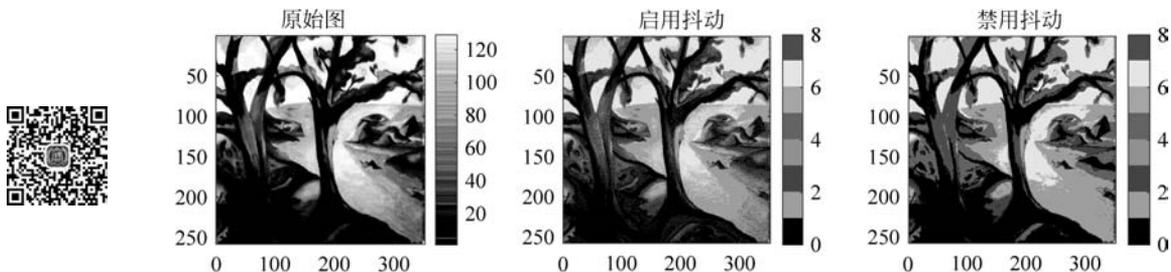


图 1.10 例 1.7 的输出

彩色 RGB 图像可以转换为索引图像。像素值由 CLUT 的索引值替换,并在内存中创建 CLUT 以引用这些值。CLUT 包含与图像中的彩色一样多的条目,每个条目包含相应彩色的  $R$ 、 $G$ 、 $B$  分量。IPT 函数 `rgb2ind` 用于将 RGB 图像转换为索引格式并创建相应的 CLUT。在例 1.8 中,图像 `image_1` 仅包含两种最具代表性的彩色,因此是一个包含两个值 0 和 1 的矩阵(见图 1.11)。对应的彩色查找表 `map_1` 包含两行,其中包含表示在  $0 \sim 1$  区间内的两种彩色的 RGB 值:

```

0.3137 0.1882 0.2314
0.8118 0.4039 0.2078

```

**例 1.8** 编写一个程序,将 RGB 图像转换为索引格式并使用不同数量的彩色显示。

```

clear; clc;
a = imread('peppers.png');
[image_1, map_1] = rgb2ind(a, 2, 'nodither');
[image_2, map_2] = rgb2ind(a, 5, 'nodither');
[image_3, map_3] = rgb2ind(a, 10, 'nodither');
subplot(221), imshow(a); title('original');
subplot(222), imshow(image_1, map_1); title('2 colors');
subplot(223), imshow(image_2, map_2); title('5 colors');
subplot(224), imshow(image_3, map_3); title('10 colors');

```

如果转换范围为  $0 \sim 255$ ,这些彩色是  $c_1(80, 48, 59)$  和  $c_2(207, 103, 53)$ 。图像 `image_2` 包含数字  $0 \sim 4$ ,代表 `map_2` 中指定的 5 种最具代表性的彩色,它们是:

```

0.2784 0.1373 0.2353
0.7608 0.1686 0.1373
0.8902 0.7255 0.6353
0.4275 0.3765 0.2235
0.8471 0.5569 0.1020

```

使用 `nodither` 选项可确保将图像的原始彩色插入图中,而不是通过混合进行修改。类似地,图像 `image_3` 包含数字  $0 \sim 9$ ,代表 `map_3` 中指定的 10 种彩色: