



第二部分 图像载体基本知识



本章提要

- 图像、数字图像
- 图像的基本类型
- 图像类型的相互转化
- 数字图像的基本文件格式
- 图像存储方式和图像文件格式的相互转换
- 其他的颜色模型



图像

- 信息隐藏算法中重要的一种载体。
- 只有了解了载体本身的特点，才能进一步学习和运用具体的隐藏算法。
- 图像：用各种观测系统以不同形式和手段观测客观世界而获得的，可以直接或间接作用于人眼而产生视知觉的实体。
- 模拟图像和数字图像

图像表示

二维平面图像：二维函数 $I=f(x,y)$ ，其中， x,y 是空间坐标， $f(x,y)$ 是点 (x,y) 的幅值， f 表示相应实际物体在该点某个性质的度量值，所有点的度量值的有序集合构成图像 I 。

1、灰度图像是一个二维灰度

（或亮度）函数 $f(x,y)$ ，灰度值，
即相应物体在每个坐标点的
明暗程度。

2、连续的：照片、绘画（模拟图像）



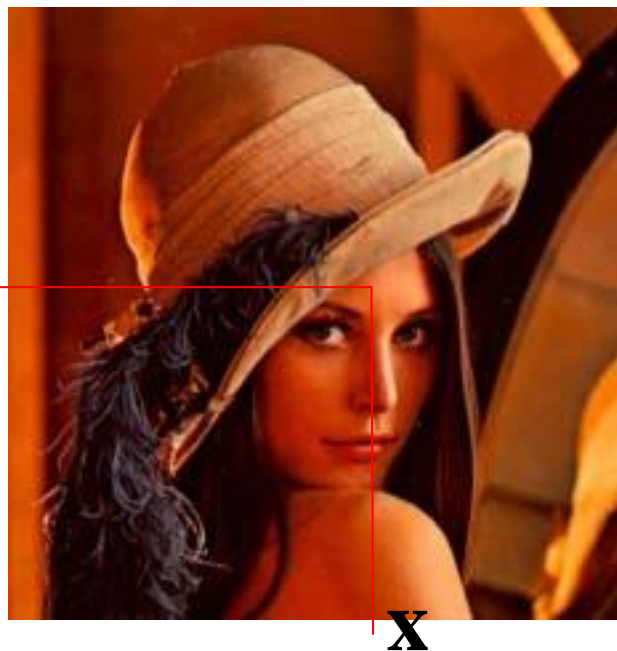
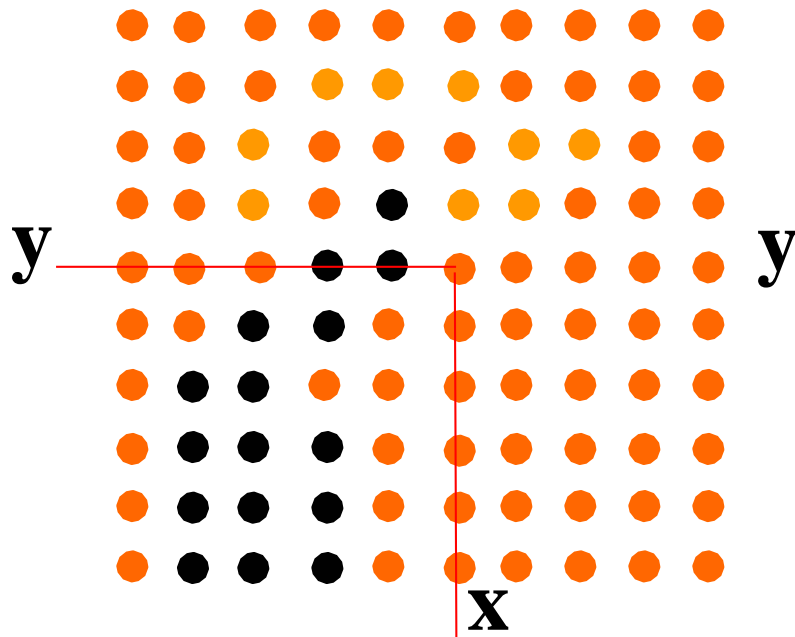


数字图像

- 模拟图像离散化： f, x, y 从实数域映射到整数域。
- 离散化的方法：从水平和竖直两个方向上同时进行采样；
- 采样点即像素（**pixel**）。
- 二维矩阵来表示数字图像，矩阵的各个元素代表一个像素的色彩信息。
- 隐藏载体涉及的图像是数字图像。

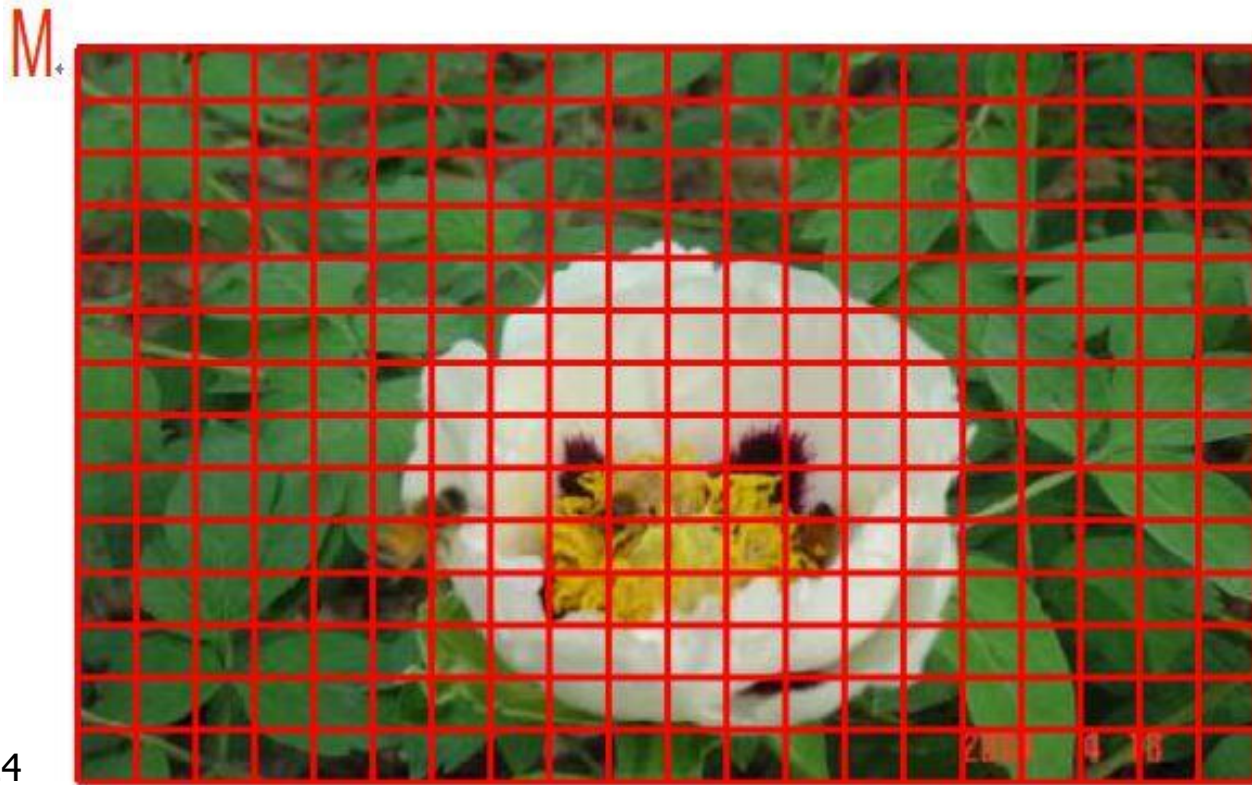
像素

数字图像由二维的元素组成，每一个元素具有一个特定的位置 (x, y) 和幅值 $f(x, y)$ ，这些元素就称为像素



图像采样

- 图像空间坐标(x, y)的数字化被称为图像采样
确定水平和垂直方向上的像素个数**N**、**M**。

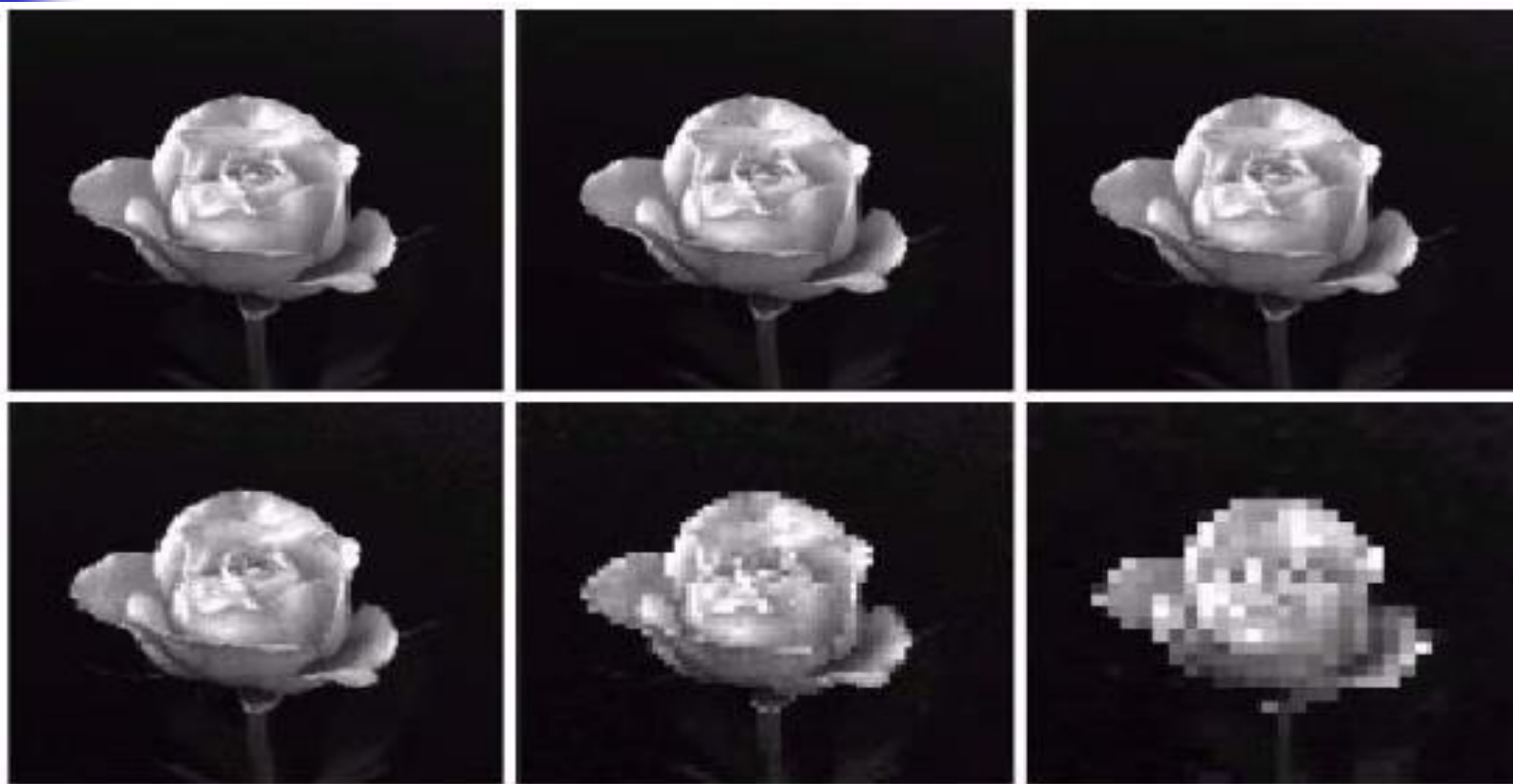


图像的采样与数字图像的质量



FIGURE 2.19 A 1024×1024 , 8-bit image subsampled down to size 32×32 pixels. The number of allowable gray levels was kept at 256.

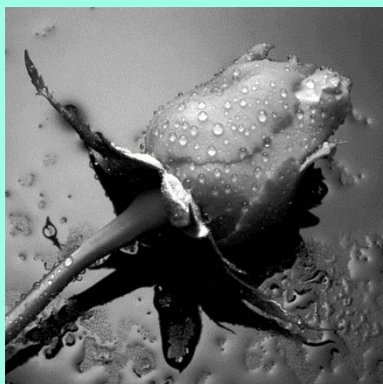
图像的采样与数字图像的质量



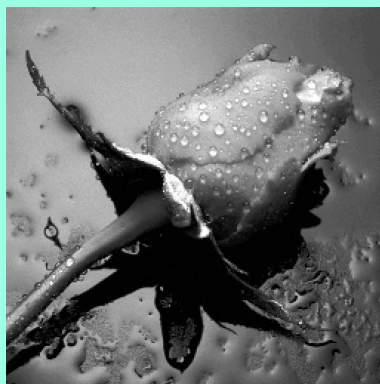
a b c
d e f

FIGURE 2.20 (a) 1024×1024 , 8-bit image. (b) 512×512 image resampled into 1024×1024 pixels by row and column duplication. (c) through (f) 256×256 , 128×128 , 64×64 , and 32×32 images resampled into 1024×1024 pixels.

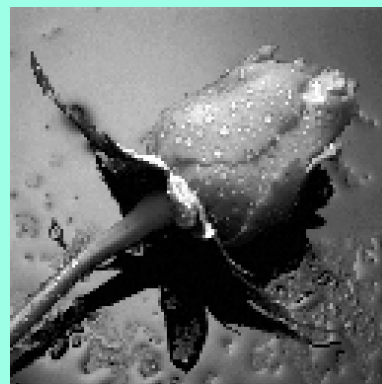
图像的采样与数字图像的质量



(a)



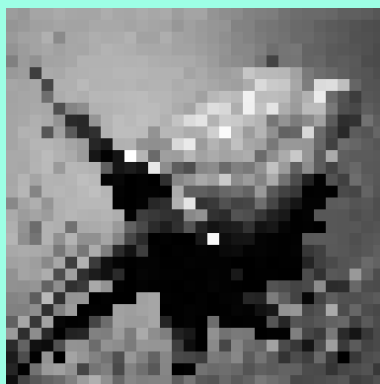
(b)



(c)



(d)



(e)



(f)

图像的采样与数字图像的质量



265x180



133x90

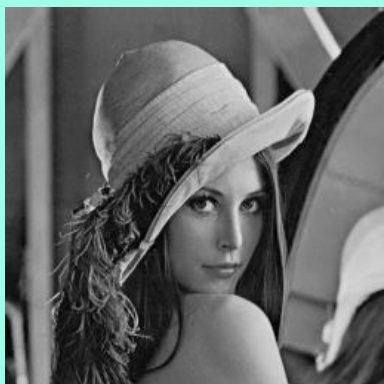


66x45

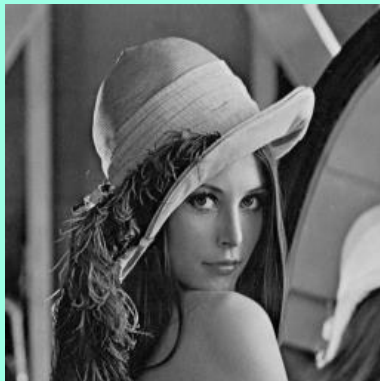


33x22

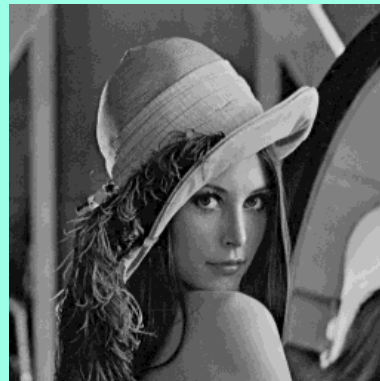
图像的量化与数字图像的质量



(a)



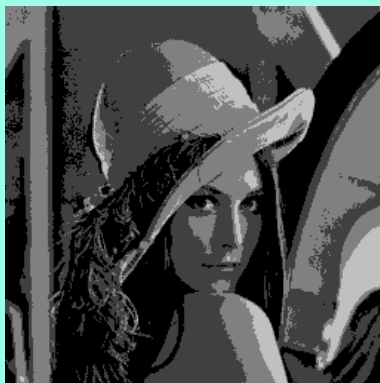
(b)



(c)



(d)



(e)



(f)

图像的量化与数字图像的质量



256灰度级



16灰度级



8灰度级



4灰度级



图形图像文件分类

- 位图 (Bitmap)
- 矢量图 (Vector)



位图文件

是目前最常用的图像表示方法

- 点阵形式描述图形图像。
- 在有足够文件量下，能真实反映图像的层次、色彩。
- 文件体积较大，放大图像过程中，图像会变模糊而失真。
- 位图的映射存储模式是将图像的每一个像素点转化为一个数据，并存放在以字节为单位的二维矩阵中。



位图

- 单色图像一个字节可以存放8个像素点的图像数据。
- 16色图像每两个像素点用一个字节存储。
- 256色图像每一个像素点用一个字节存储。
-
- 精确描述各种不同颜色模型的图像画面，它更适合内容复杂的图像和真实的照片。
- ----符合信息隐藏技术中的载体的基本要求。



矢量（Vector）文件

- 以数学方法描述的一种由几何元素组成的图形图像。
- 文件量小，任意缩放不会改变图像质量。
- 矢量图像（图形）是用一系列的线和形描述图像，也可能使用实心的或者有等级深浅的色彩填充一些区域。
- 矢量格式适用于线型图，如计算机辅助设计（CAD）的图形和图像，它们只有简单的形状、灰度和颜色。



矢量图

- 矢量文件只存储图像内容的轮廓部分, 而不存储图像数据的每一点。
- 如, 对于一个圆形图案, 只要存储圆心的坐标位置和半径长度, 以及圆形边线和内部的颜色即可。
- 该存储方式的缺点是经常耗费大量的时间做一些复杂的分析演算工作
- 但图像的缩放不会影响显示精度, 即图像不会失真, 而且图像的存储空间较位图文件要少得多。
- 所以, 向量处理比较适合存储各种图表和工程设计图。

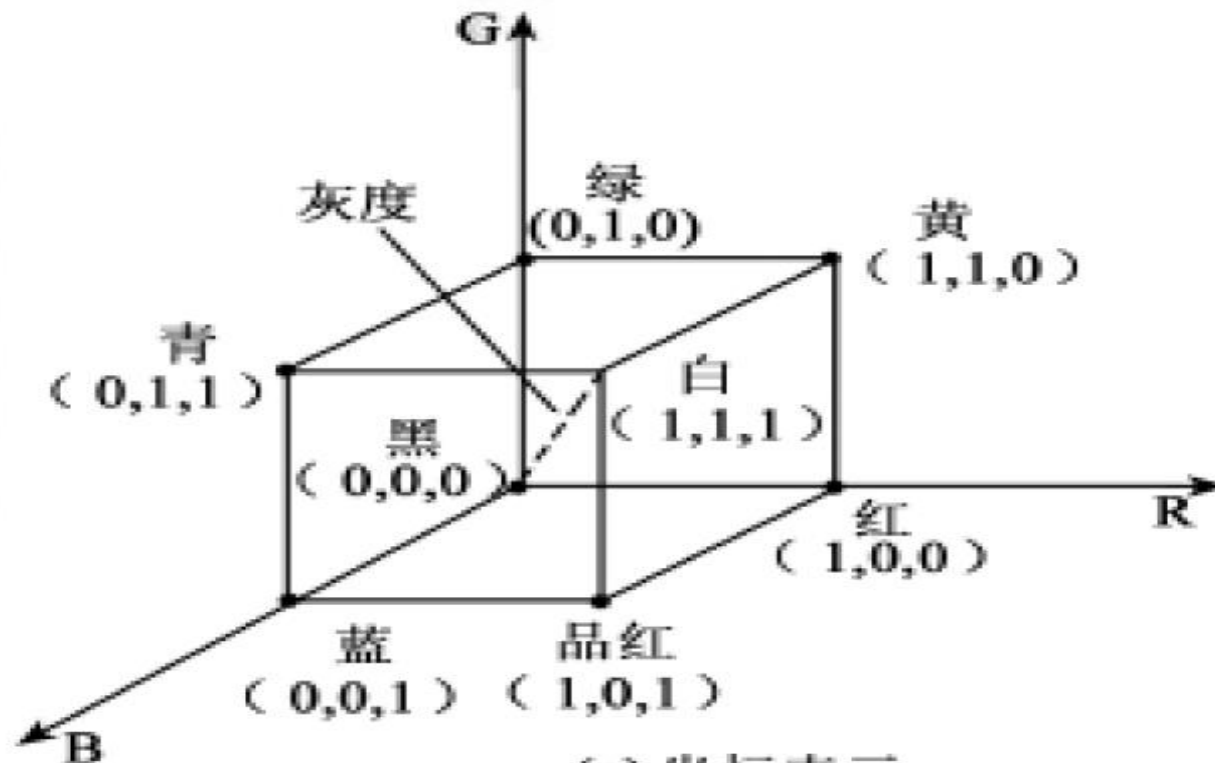


RGB颜色模型

- 色彩是图像的重要元素
- 红、绿、蓝（**RGB**）光学三颜色
- 红+绿+蓝--->白色， 红+绿--->黄色
- 用一个由**R, G, B** 为坐标轴定义的单位立方体来描述这样一个符合视觉理论的颜色模型。

RGB颜色模型

坐标原点代表黑色, $(1, 1, 1)$ 代表白色, 坐标轴上的顶点称为基色 (**Primitive Colour**) 点。立方体中的每一种颜色由一个三元组 (R, G, B) 表示, 每一个分量的数值均在 $[0, 1]$ 区间。



(c) 坐标表示



位图中调色板

- 位图中颜色如何在矩阵中体现的呢？调色板
- 调色板是包含不同颜色的颜色表, 每种颜色以红、绿、蓝三种颜色的组合来表示, 图像的每一个像素对应一个数字, 而该数字对应调色板中的一种颜色。
- 如某像素3 值为1, 则表示该颜色为调色板的编号为1 的颜色。
- 调色板的单元个数是与图像的颜色数相一致的, 256 色图像的调色板就有256 个单元。
- 所谓16 色或256色图像, 只是表示该幅图像最多只能有16 种或256 种颜色。
- 不同的图像有不同的调色板。



位图分类

位图精确描述各种不同颜色模型的图像画面，载体图像对象基本上是位图。

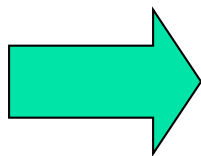
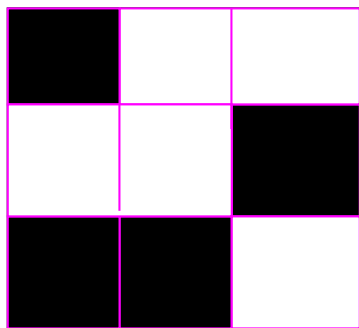
位图都可以看做一个二维数据矩阵，根据其图像调色板的存在方式及矩阵数值与像素颜色之间的对应关系。

二值图像、索引图像、

■ 灰度图像、**RGB** 图像

二值图像

- 像素：0（黑）和1（白）
- 二进制图像能够使用无符号8位整型(`uint8`) 或双精度类型(`double`)的数组来存储。
- 在MATLAB 图像处理工具箱中, 任何返回一幅二进制图像的函数都使用`uint8` 逻辑数组存储该图像。



$$X = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



图 1.3 二进制的 lenna 图像



索引图像

- 把像素值直接作为**RGB** 调色板下标的图像
- 一幅索引图包含一个数据矩阵**data** 和一个调色板矩阵**map**,
- 数据矩阵可以是**uint8**, **uint16** 或双精度类型的,
- 而调色板矩阵则总是一个 $m \times 3$ 的双精度类型矩阵(其中, m 表示颜色数目), 该矩阵的元素都是 $[0, 1]$ 范围内的浮点数。**map** 矩阵的每一行指定一个颜色的红、绿、蓝颜色分量。
- 索引图像可以把像素值直接映射为调色板数值, 每一个像素的颜色通过使用**data** 的数值作为**map** 的下标来获得: 数值1 表示**map** 的第一行, 数值2 表示**map** 的第二行, 依此类推。

MATLAB 自带的**woman** 信号构成的图像的像素索引矩阵和调色板矩阵。**woman** 图像是一幅典型的索引图像。

ray Editor: women

Edit View Web Window Help

Numeric format: shortG Size: 256 by 256

	1	2	3	4	5	6	7	8	9	10
1	125	79	111	98	111	79	125	98	91	112
2	98	112	91	112	79	111	112	79	112	79
3	98	125	98	111	112	98	79	111	112	111
4	106	98	79	112	79	125	98	94	98	79
5	79	125	98	91	98	98	79	111	112	125
6	98	111	112	98	94	111	112	79	98	79
7	112	79	111	94	98	79	111	112	91	112
8	111	112	98	111	112	91	112	79	98	91
9	106	98	79	112	79	98	111	112	111	112
10	111	112	111	91	112	111	94	111	79	98
11	79	98	125	98	112	91	98	112	111	94
12	111	94	111	98	91	112	111	79	112	111



ray Editor: map

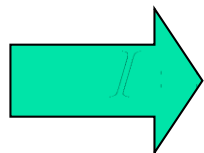
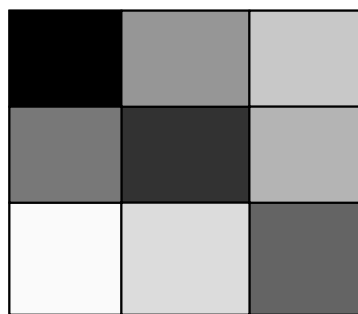
File Edit View Web Window Help

Size: 256 by 3

	1	2	3
123	0.59051	0.59051	0.59051
124	0.60194	0.60194	0.60194
125	0.60536	0.60536	0.60536
126	0.60877	0.60877	0.60877
127	0.61218	0.61218	0.61218
128	0.61557	0.61557	0.61557
129	0.61896	0.61896	0.61896
130	0.62234	0.62234	0.62234
131	0.62572	0.62572	0.62572
132	0.62908	0.62908	0.62908
133	0.63244	0.63244	0.63244
134	0.63579	0.63579	0.63579
135	0.63913	0.63913	0.63913
136	0.64247	0.64247	0.64247
137	0.64579	0.64579	0.64579
138	0.64911	0.64911	0.64911
139	0.65243	0.65243	0.65243
140	0.65573	0.65573	0.65573
141	0.65903	0.65903	0.65903
142	0.66232	0.66232	0.66232
143	0.66561	0.66561	0.66561

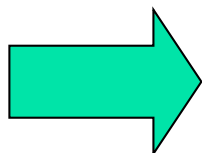
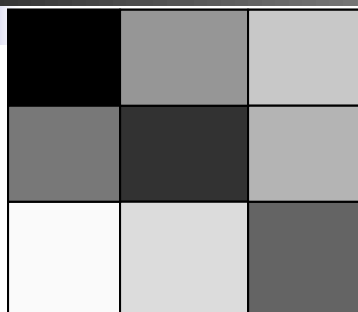
灰度图像

- 灰度图像是包含灰度级(亮度) 的图像。灰度就是我们通常说的亮度。
- 在MATLAB 中, 灰度图像由一个uint8, uint16 或双精度类型的数组来描述。灰度图像实际上是一个数据矩阵 I , 该矩阵的每一个元素对应于图像的一个像素点, 元素的数值代表一定范围内的灰度级, 通常0 代表黑色, 1, 255 或65535(不同存储类型) 代表白色。



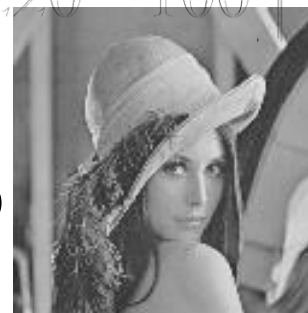
0	150	200
120	50	180
250	220	100





$$I = \begin{vmatrix} 0 & 150 & 200 \\ 120 & 50 & 180 \\ 250 & 220 & 100 \end{vmatrix}$$

灰度图象(**128x128**)及其对应的
数值矩阵（仅列出一部分(**26x31**)）

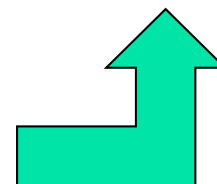


```
125, 153, 158, 157, 127,
70, 103, 120, 129, 144, 144, 150, 150, 147, 150, 160, 165, 160, 164, 165, 167, 175, 175, 16
6, 133, 60,
133, 154, 158, 100, 116, 120, 97, 74, 54,
74, 118, 146, 148, 150, 145, 157, 164, 157, 158, 162, 165, 171, 155, 115, 88, 49,
155, 163, 95, 112, 123, 101, 137, 108, 81, 71, 63,
81, 137, 142, 146, 152, 159, 161, 159, 154, 138, 81, 78, 84, 114, 95,
167, 69, 85, 59, 65, 43, 85, 34, 69,
70, 104, 101, 117, 100, 104, 140, 100, 105, 150, 140, 114, 80, 57, 45, 51, 57
```

灰度图像描述示例



	0	1	2	3	4	5	6	7
0	130	146	133	95	71	71	62	78
1	130	146	133	92	62	71	62	71
2	139	146	146	120	62	55	55	55
3	139	139	139	146	117	112	117	110
4	139	139	139	139	139	139	139	139
5	146	142	139	139	139	143	125	139
6	156	159	159	159	159	146	159	159
7	168	159	156	159	159	159	139	159





RGB 图像

- RGB 图像仅是一类图像的总称。这类图像不使用单独的调色板, 每一个像素的颜色由存储在相应位置的红、绿、蓝颜色分量共同决定。
- RGB 图像是24 位图像, 红、绿、蓝分量分别占用8 位, 理论上可以包含16M种不同颜色, 由于这种颜色精度能够再现图像的真实色彩, 所以又称RGB 图像为真彩图像。
- 在MATLAB 中, 一幅RGB 图像由一个uint8, uint16 或双精度类型的 $m \times n \times 3$ 数组(通常称为RGB 数组) 来描述, 其中, m 和 n 分别表示图像的宽度和高度。在一个双精度类型的RGB 数组中, 每一个颜色分量都是一个 $[0, 1]$ 范围内的数值, 颜色分量为 $(0, 0, 0)$ 的像素将显示为黑色, 颜色分量为 $(1, 1, 1)$ 的像素将显示为白色。每一个像素三个颜色分量都存储在数据数组的第三维中。
- 例如, 像素 $(10, 5)$ 的红、绿、蓝色分量都存储在 $\text{RGB}(10, 5, 1)$, $\text{RGB}(10, 5, 2)$, $\text{RGB}(10, 5, 3)$ 中。



原始RGB彩色图像



R分量图像



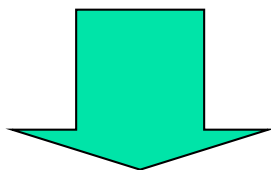
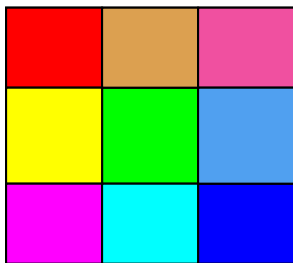
G分量图像



B分量图像

uint8 RGB图像

- 通常，三元组的每个数值也是在0到255之间，0表示相应的基色在该像素中没有，而255则代表相应的基色在该像素中取得最大值，这种情况下每个像素可用三个字节来表示。



$$R: \begin{bmatrix} 255 & 240 & 240 \\ 255 & 0 & 80 \\ 255 & 0 & 0 \end{bmatrix}$$

$$G: \begin{bmatrix} 0 & 160 & 80 \\ 255 & 255 & 160 \\ 0 & 255 & 0 \end{bmatrix}$$

$$B: \begin{bmatrix} 0 & 80 & 160 \\ 0 & 0 & 240 \\ 255 & 255 & 255 \end{bmatrix}$$

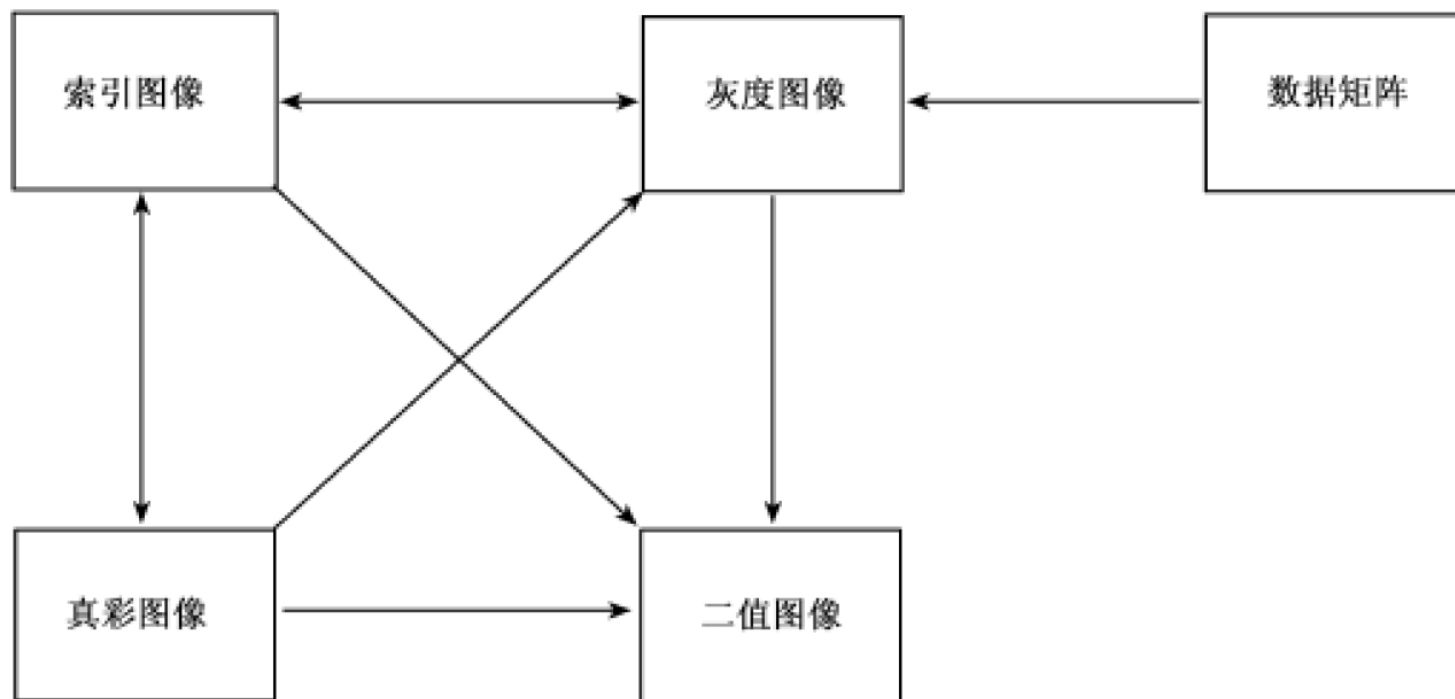
彩色图象(128x128)
及其对应的数值矩阵
(仅列出一部分(31x25))



(207, 137, 130)	(220, 179, 163)	(215, 169, 161)	(210, 179, 172)	(210, 179, 172)
(207, 154, 146)	(217, 124, 121)	(226, 144, 133)	(226, 144, 133)	(224, 137, 124)
(227, 151, 136)	(227, 151, 136)	(226, 159, 142)	(227, 151, 136)	(230, 170, 154)
(231, 178, 163)	(231, 178, 163)	(231, 178, 163)	(236, 187, 171)	(236, 187, 171)
(239, 195, 176)	(239, 195, 176)	(240, 205, 187)	(239, 195, 176)	(231, 138, 123)
(217, 124, 121)	(215, 169, 161)	(216, 179, 170)	(216, 179, 170)	(207, 137, 120)
(159, 51, 71)	(189, 89, 101)	(216, 111, 110)	(217, 124, 121)	(227, 151, 136)
(227, 151, 136)	(226, 159, 142)	(226, 159, 142)	(237, 159, 135)	(237, 159, 135)
(231, 178, 163)	(236, 187, 171)	(231, 178, 163)	(236, 187, 171)	(236, 187, 171)
(236, 187, 171)	(239, 195, 176)	(239, 195, 176)	(236, 187, 171)	(227, 133, 118)
(213, 142, 135)	(216, 179, 170)	(221, 184, 170)	(190, 89, 89)	(204, 109, 113)
(204, 115, 118)	(189, 85, 97)	(159, 60, 78)	(136, 38, 65)	(160, 56, 75)

图像类型的转换

- 在变换域的数字水印算法中, 对于索引图像的载体必须将其先转换为**RGB** 图像再加水印, 否则将破坏载体。





Matlab图像转换函数

表 1.1

MATLAB 图像类型转换函数及其功能

函 数	功 能
dither	使用抖动方法, 根据灰度图像创建二进制图像或根据 RGB 图像创建索引图像
gray2ind	根据一幅灰度图像创建索引图像
grayslice	使用阈值截取方法, 根据一幅灰度图像创建索引图像
im2bw	使用阈值截取方法, 根据一幅灰度图像、索引图像或 RGB 图像创建二进制图像
ind2gray	根据一幅索引图像创建一幅灰度图像
ind2rgb	根据一幅索引图像创建一幅 RGB 图像
mat2gray	通过数据缩放, 再根据矩阵数据创建一幅灰度图像
rgb2gray	根据一幅 RGB 图像创建一幅灰度图像
rgb2ind	根据一幅 RGB 图像创建一幅索引图像

RGB图像和索引图像互转





数字图像的基本文件格式

- 图像格式指的是存储图像采用的文件格式。
- 不同操作系统、不同图像处理系统所支持的图像格式都有可能不同。
- **BMP 文件**
- GIF 文件、TIFF 文件、PCX 文件
- **JPEG 格式**
- PSD 格式、PCD 格式
- **PNG 格式**



BMP文件

- Microsoft Windows 所定义的图像文件格式。
- BMP图像文件特点：
 - ① 每个文件只能存放一幅非压缩图像。
 - ② 只能存储四种图像数据：单色、**16 色**、**256 色**、真彩色。
 - ③ 图像数据有压缩或不压缩两种处理方式。
 - ④ **BMP** 图像文件的文件结构可分为三部分：表头、调色板和图像数据。



JPEG 格式

- 其正式名称为“连续色调静态图像的数字压缩和编码”，是一种基于离散余弦变换(DCT) 或离散小波。
- 变换(DWT) 的图像压缩编码标准。
- **JPEG** 压缩技术十分先进, 它采用最少的磁盘空间来得到较好的图像质量。



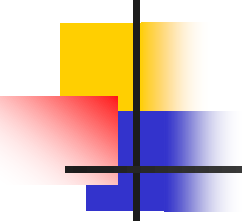
PNG 格式

- PNG 能存储32 色的位图文件格式。
- 用无损压缩方式来减少文件的大小。
- 目前, 越来越多的软件开始支持这一格式, 在不久的将来, 它可能会在整个Web 流行。
- 显示速度很快, 只需要下载1 /64 的图像信息就可以显示出低分辨率的预览图像。
- 格式不支持动画。



图像存储方式和图像文件格式的相互转换

- **MATLAB** 中最基本的数据结构是数组(矩阵) 。
- 在**MATLAB** 中, 大多数图像用二维数组(矩阵) 存储, 矩阵中的一个元素对应于要显示图像的一个像素。
- 例如, 一个由**100** 行和**200** 列的不同颜色的点组成的图像可以用一个**100 × 200** 的矩阵来存储。
- 在默认的情况下, **MATLAB** 将图像中的数据存储为双精度型 (**double**), 即**64** 位的浮点数。
- 为了减少内存需求, 提高系统运行速率, **MATLAB** 还提供了两个重要的数字类型**uint8** 和**uint16**, 用以支持**8** 位和**16** 位的无符号整数类型。即图像矩阵中的每个数据占用一个和两个字节。

- 
- 几乎所有的**MATLAB** 工具箱函数都可使用**double** 作为参数类型。 - 运算时不需要转换
 - **MATLAB**工具箱函数往往不支持**uint8** 类型。
 - **uint8** 的优势仅仅在于节省存储空间, 在涉及运算时要将其转换成**double** 型。
 - 在**MATLAB** 中, 数据矩阵中包含**uint8** 数字类型的图像称为**8 位图**; 同理, 数据矩阵中包含**uint16** 数字类型的图像称为**16 位图**。



8 位和16 位RGB 图像

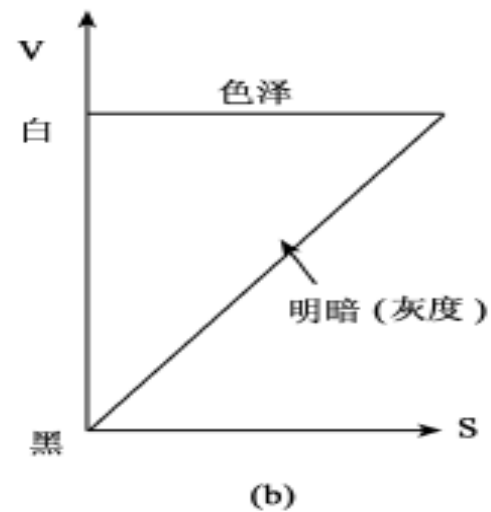
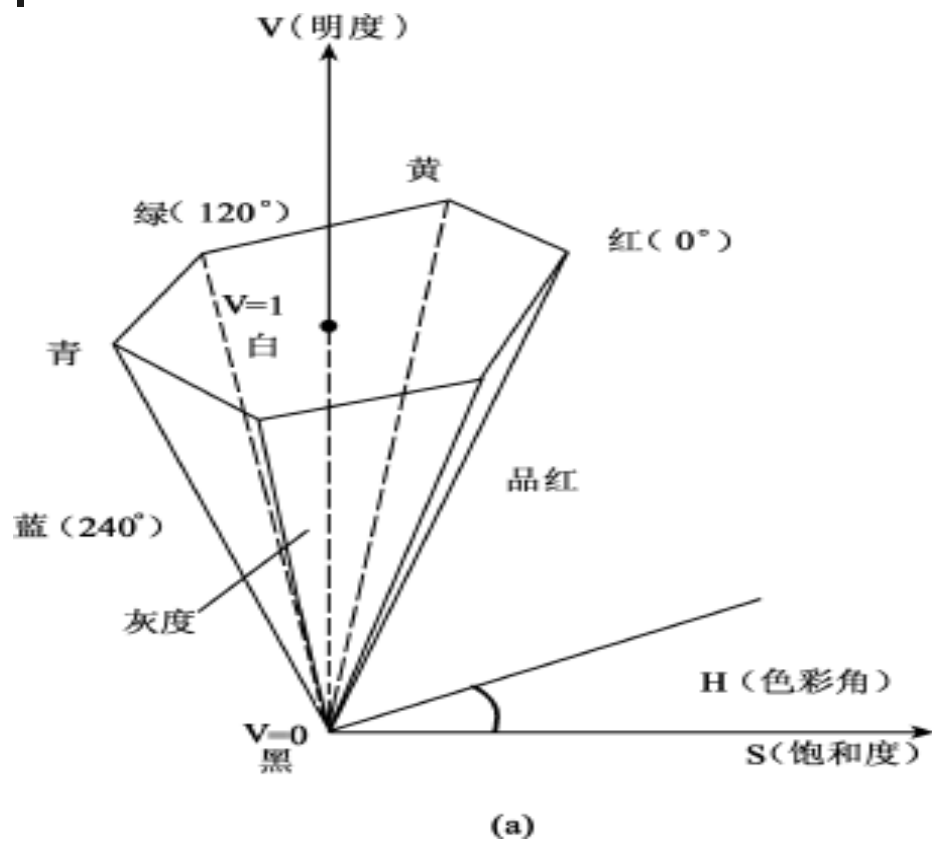
- 8 位RGB 图像的颜色数据是[0, 255] 之间的整数, 而不是[0, 1] 之间的浮点值。
- 所以, 在8 位RGB 图像中, 颜色值为(255, 255, 255) 的像素显示为白色。
- 不管RGB图像是何种类型, MATLAB 都通过以下代码来显示, 即: `image(RGB)`。
- 将RGB 图像从双精度的浮点类型转换为uint8 无符号整数类型时, 首先要乘以255, 即: `RGB8 = uint8(round(RGB64* 255))`。
- 相反, 如果将uint8 无符号整数类型的RGB 图像转换为双精度的浮点类型时, 首先要除以255, 即: `RGB64 = double(RGB8) /255`。



HSV模型

- 该模型经常为艺术家所使用。这种格式反映了人类观察色彩的方式，同时也有利于图像处理。
- H, S, V 分别指的是色调(彩) (hue)、色饱和度(saturation) 和明度(value)。
- H 分量表示颜色的种类, 取值范围为 $0^{\circ} \sim 360^{\circ}$, 相应的颜色从红、黄、绿、蓝绿、蓝、紫到黑变化, 且它的值由绕V 轴的旋转角决定, 每一种颜色和它的补色之间相差 180° 。S 分量的取值范围也是 $0 \sim 1$, 表示所选色彩的纯度与该色彩的最大纯度的比例。

HSV彩色空间



与RGB之间关系

$$\begin{cases} V = \max(R, G, B) \\ S = \frac{mm}{V} \\ H = h * 60^\circ \end{cases} \quad mm = \max(r, g, b) - \min(r, g, b)$$

$$h = \begin{cases} 5 + b' & \text{若 } r = \max(r, g, b) \text{ 和 } g = \min(r, g, b) \\ 1 - g' & \text{若 } r = \max(r, g, b) \text{ 和 } g \neq \min(r, g, b) \\ 1 + r' & \text{若 } g = \max(r, g, b) \text{ 和 } b = \min(r, g, b) \\ 3 - b' & \text{若 } g = \max(r, g, b) \text{ 和 } b \neq \min(r, g, b) \\ 3 + g' & \text{若 } b = \max(r, g, b) \text{ 和 } g = \min(r, g, b) \\ 5 - r' & \text{其它} \end{cases} \quad \begin{cases} r' = \frac{V - r}{mm} \\ g' = \frac{V - g}{mm} \\ b' = \frac{V - b}{mm} \end{cases}$$



原始图像



H分量



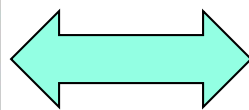
S分量



V分量



2019/5/24

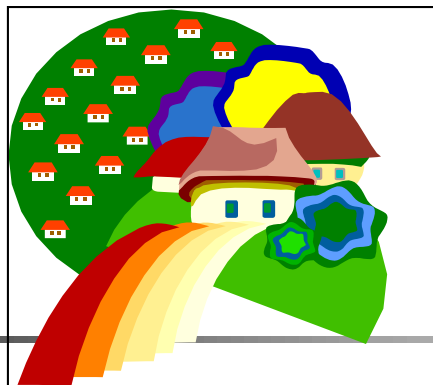




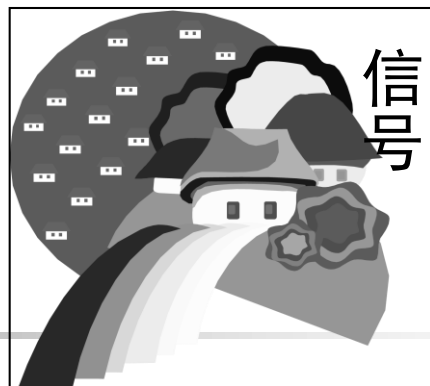
YUV彩色空间

- 一种彩色传输模型，主要用于彩色电视信号传输标准。
- **Y**—黑白亮度分量，**U,V**—彩色信息用以显示彩色图像。

信号
彩色电视

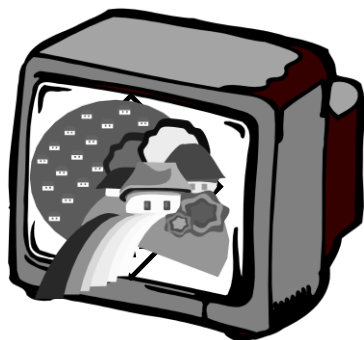


信号
黑白电视



Y,U,
V

Y



黑白电视
与RGB之间的转换关系：

Y

Y,0,
0



彩色电视
机

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.287 & 0.436 \\ 0.615 & -0.515 & -0.10 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



YCrCb模型

- YCrCb彩色空间是由YUV彩色空间派生的一种颜色空间，主要用于数字电视系统。是数字视频信号的世界标准。
- 基本上，YCrCb代表和YUV相同的彩色空间。在这两个彩色空间中Y表示明亮度，也就是灰阶值；
- “色度”则定义了颜色的两个方面——色调与饱和度，YCrCb彩色空间分别用Cr和Cb来表示。其中，Cr反映了红色部分。而Cb反映的是蓝色部分。

与RGB之间关系

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

原图像



变换后的图像





原始图像



Y分量



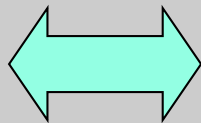
Cb分量



Cr分量



2019/5/24
YCbCr图像



RGB图像

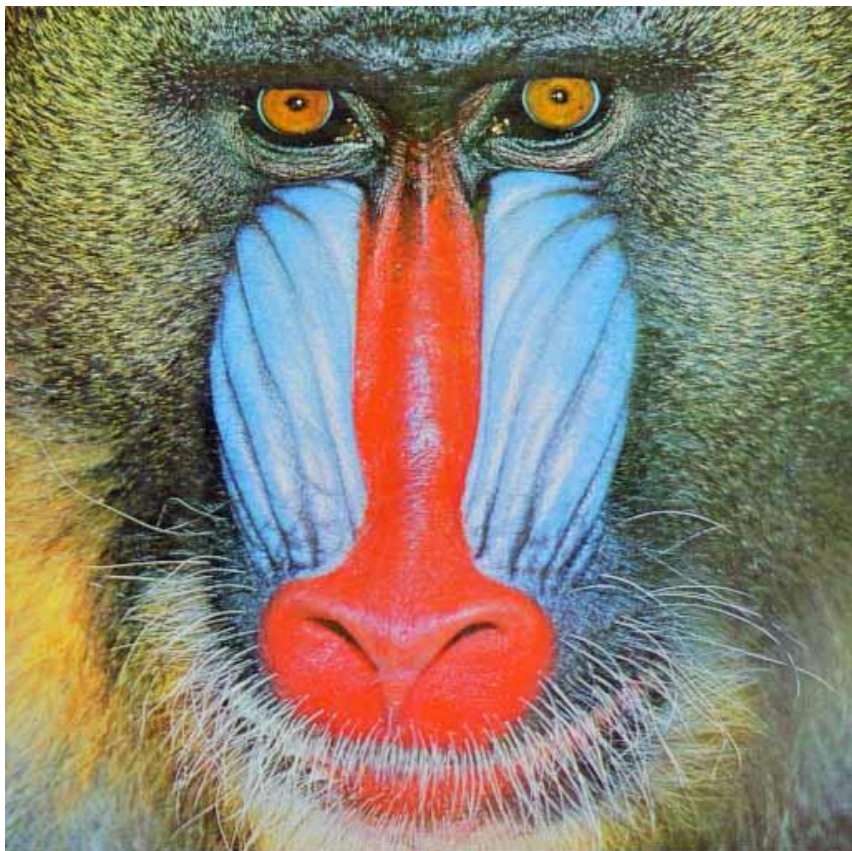
常用的测试图像

- Lenna (Miss November 1972's centerfold)



常用的测试图像

• Baboon



Barbara

