

杭州电子科技大学 李黎

第三章 图像增强

■ 图像增强技术是一大类基本的图像处理技术,目的是对一幅图像进行加工,突出图像中的某些信息,削弱或除去某些不需要的信息,以得到对具体应用来说视觉效果更好、更有用的图像,或转换成一种更适合人或机器进行分析处理的形式。



■ 图像的灰度变换

- 直方图的定义
- 对比度展宽
- 灰度级的修正
- 动态范围调整
- 非线性动态范围调整
- 直方图均衡化
- 图像的几何变换
- 图像的位置变换
- 图像的形状变换

■ 1) 定义

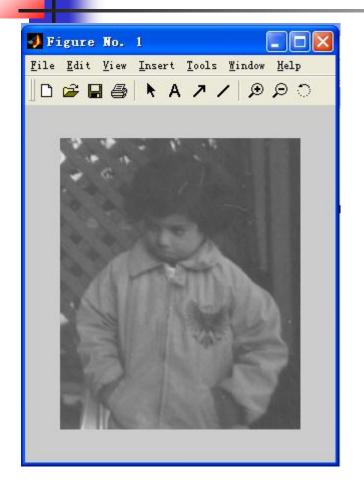
灰度直方图(histogram)是灰度级的函数,描述的是图像中每种灰度级像素的个数,反映图像中每种灰度出现的频率。横坐标是灰度级,纵坐标是灰度级出现的频率。

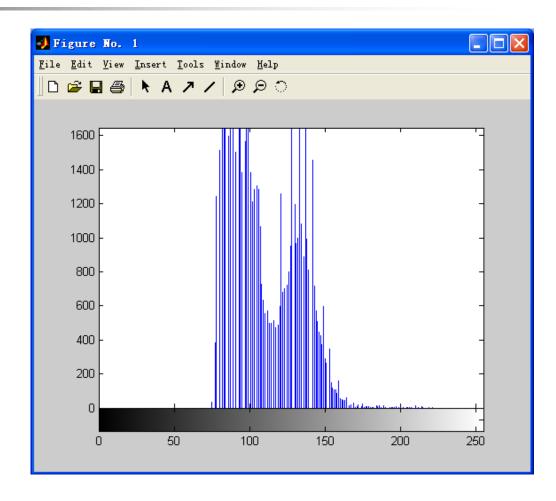
图像及其灰度直方图的例 (512像素*512像素)



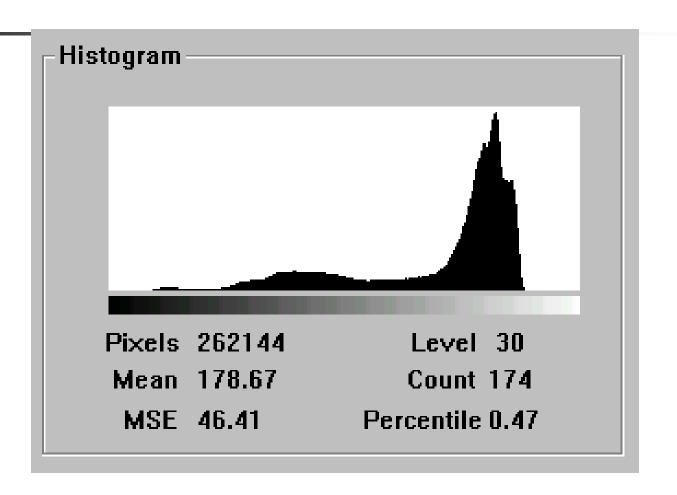
统计各个象素值出现的次数或概率

直方图的概念

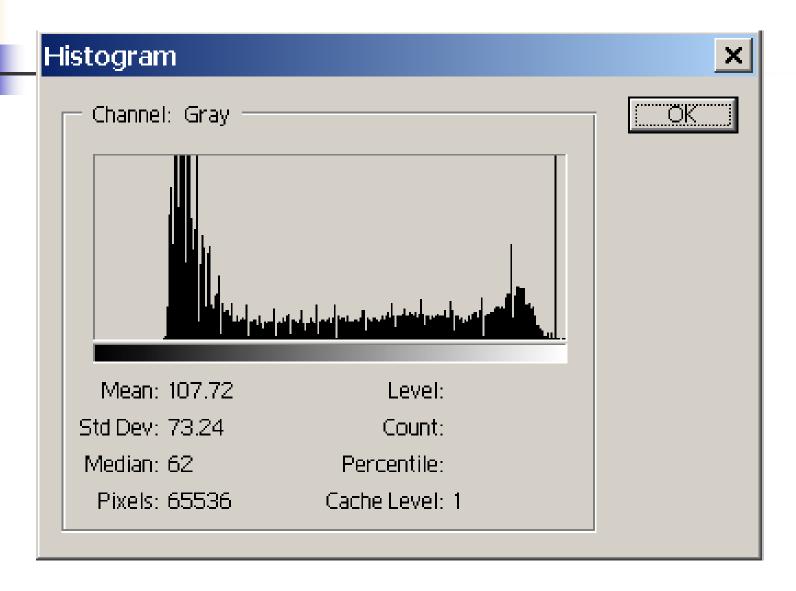




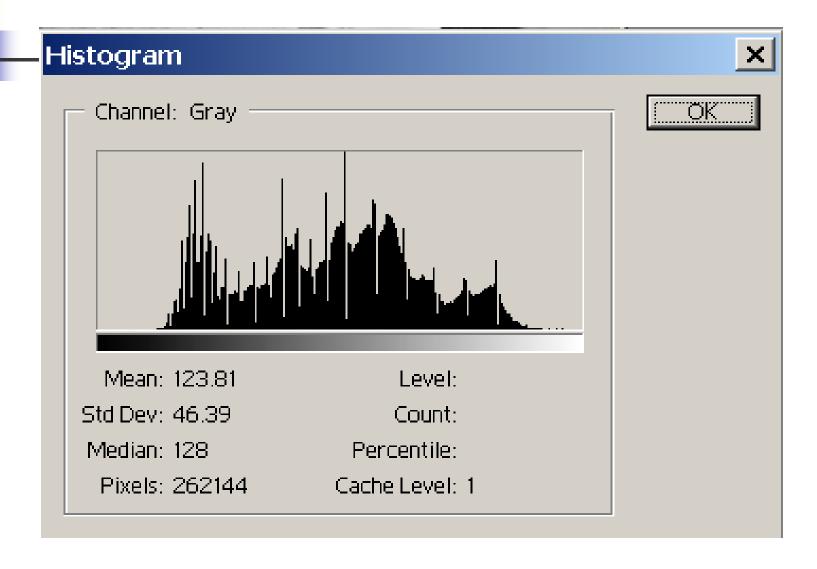


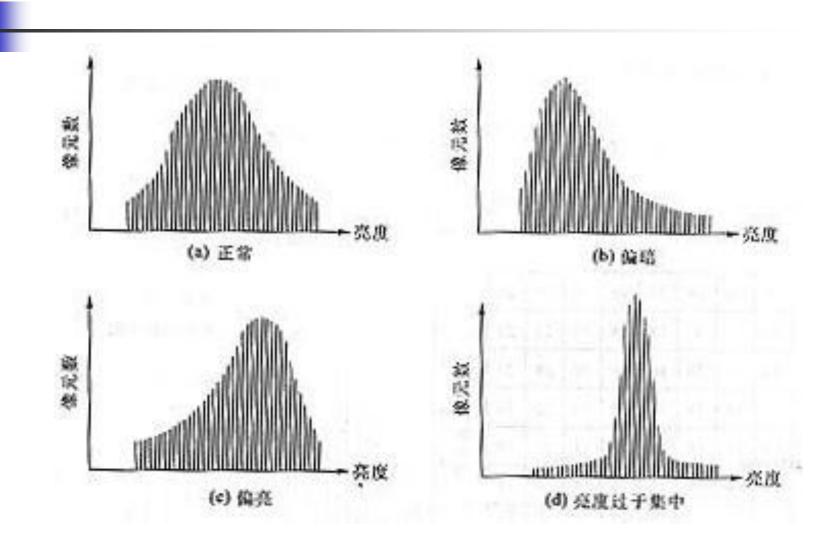












直方图的性质

- 直方图的性质
 - 不表示图像的空间信息;
 - 任一特定图像都有<mark>唯一直方图</mark>,但反之并不成立;
 - 直方图的<mark>可相加性</mark>

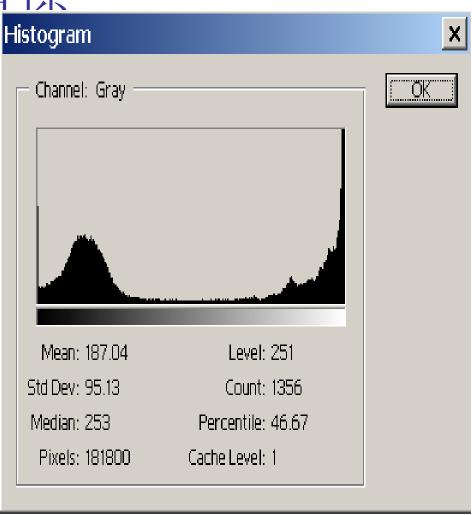
例如一副图像由若干个不相交的区域构成,则整幅图像的直方图是这若干个区域直方图之和。

直方图的用途

- 边界阈值选择
 - 使用轮廓线确定简单物体的边界的方法,称为阈值 化;
 - 对物体与背景有较强对比的景物的分割特别有用;
 - 例 双峰直方图

直方图的用涂Histogram





3. 2 对比度展宽

一、对比度展宽的目的:

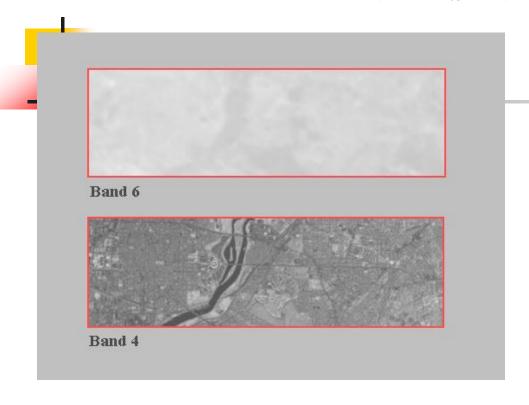
是一点对一点的灰度级的影射。设图的大声的大声,不是一点大声,

新、旧图的灰度级分别为g和f,g和f均在

[0,255]间变化。

目的:将人所关心的部分强调出来。

对比度变换

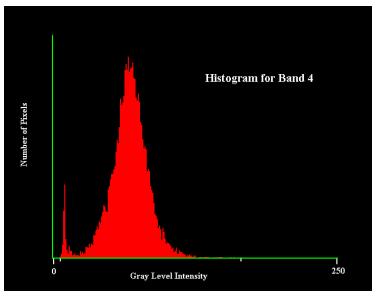


Histogram for Band 6

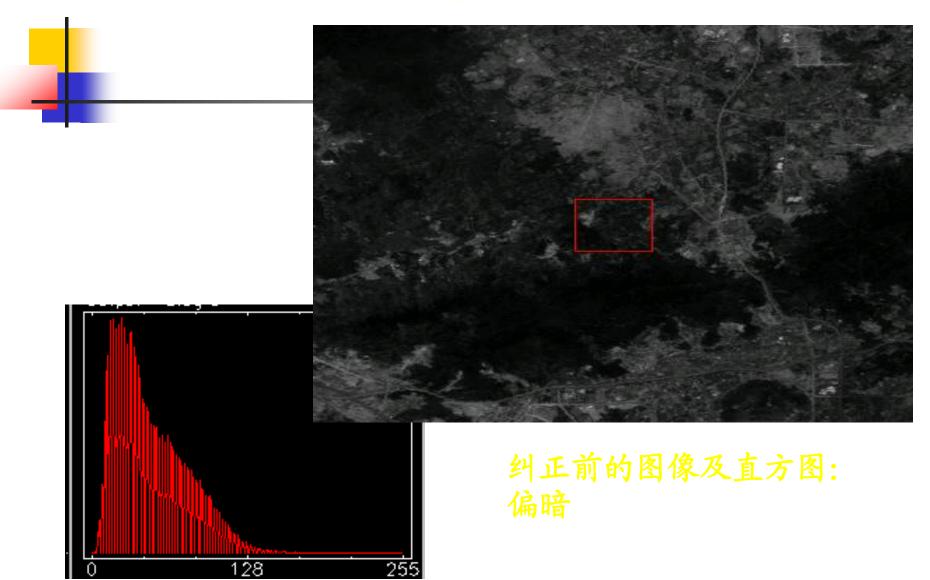
Gray Level Intensity

250

通过改变像元的亮度值来改变图像 像元的对比度,从而改善图像质量 的图像处理方法。

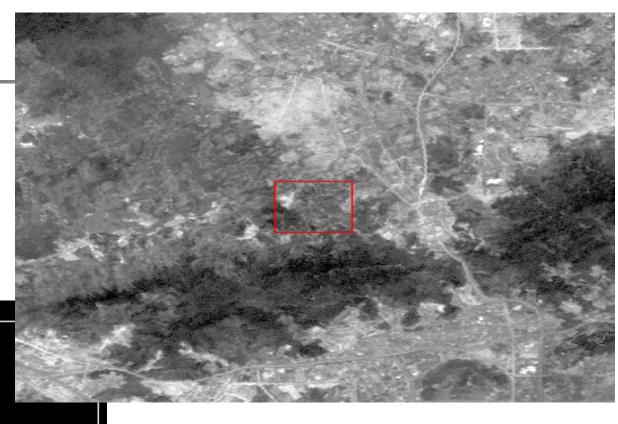


对比度变换



对比度变换

Output Histogram

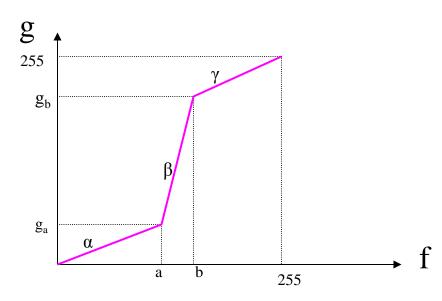


纠正后的图像及直方图

3.2 对比度展宽

方法:

$$g = \begin{cases} \alpha f & 0 \le f < a \\ \beta (f - a) + g_a & a \le f < b \\ \gamma (f - b) + g_b & b \le f < L \end{cases}$$

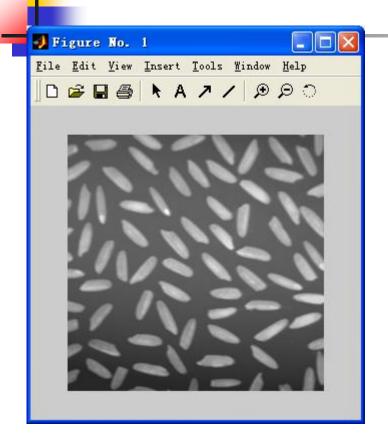


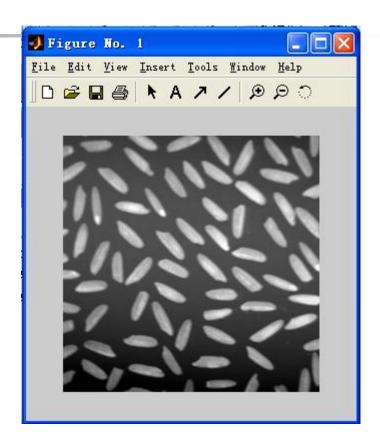
对比度展宽效果

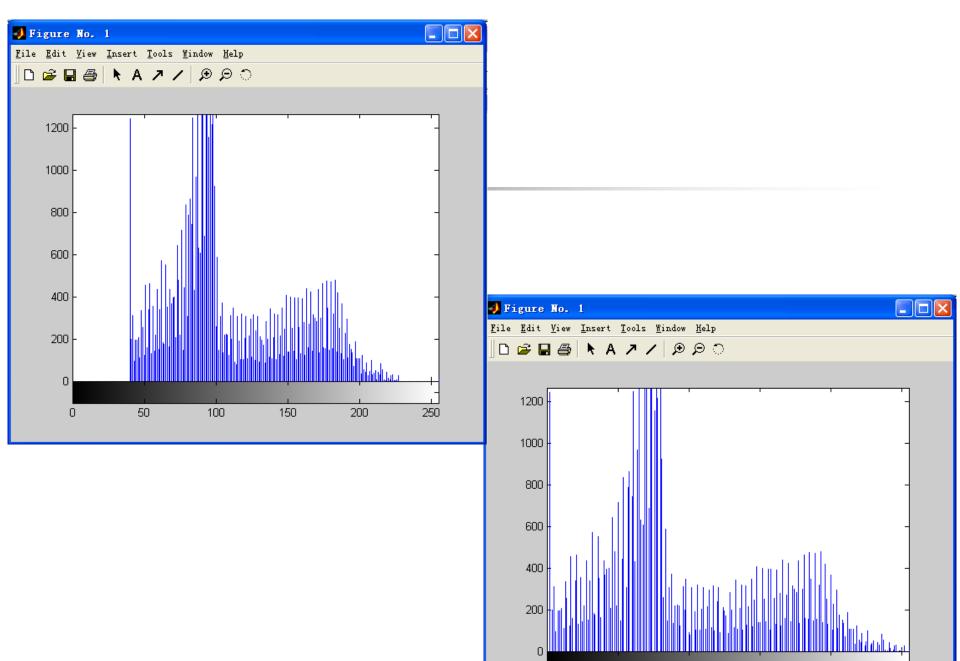




返回







- I =imread('rice.tif');
- >> J=imadjust(I,[0.15 0.9],[0 1]); %进 行直方图变换
- >> subplot(1,2,1);
- >> imshow(J);
- >> subplot(1,2,2);
- >> imhist(J)

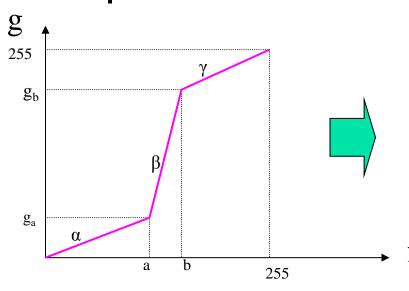


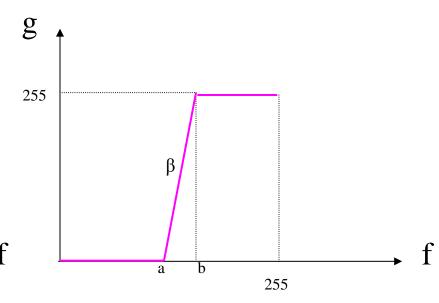
3. 2 对比度展宽

二、<u>灰级窗</u>:

只显示指定灰度级范围内的信息。如:

$$a=y=0$$





灰级窗效果示意图



灰级窗



对比度展宽

返回

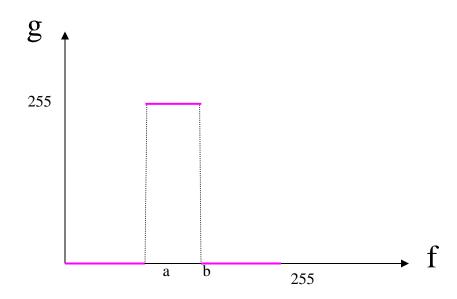


3.2 对比度展宽

三、灰级窗切片:

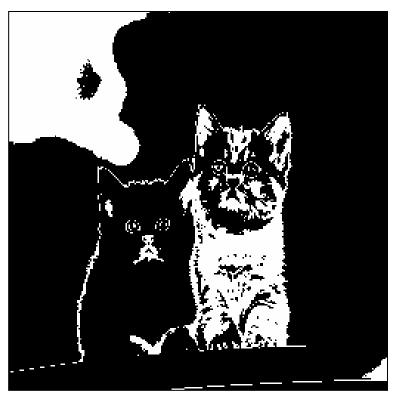
只保留感兴趣的部分,其余部分置为

0。



灰级窗切片效果示意图







3.3 动态范围调整

动态范围: 是指图像中从暗到亮的变化范围。

由于人眼所可以分辨的灰度的变化范围 是有限的,所以当动态范围太大时,很高的 亮度值把暗区的信号都掩盖了。

通过<mark>动态范围的压缩可以将所关心部分</mark> 的灰度级的变化范围扩大。

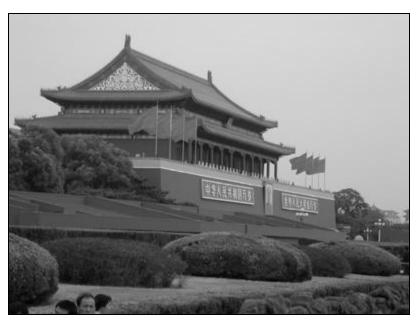
3.3 动态范围调整

一、线性动态范围调整:



$$h^*(x,y) = \begin{cases} 0 & h(x,y) <= a \\ \frac{255}{b-a}h(x,y) - \frac{255a}{(b-a)} & h(x,y) \in (a,b) \\ 255 & h(x,y) >= b \end{cases}$$

线性动态范围调整效果



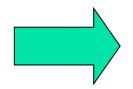






线性动态范围调整

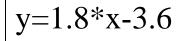
1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0



黑: 0→2

白: 9→7

2	3	7	7	7
2	2	3	7	3
3	6	2	6	4
6	7	2	2	5
0	7	2	6	2





作用:	进行亮暗限幅
11 / 17 •	

0	2	9	9	9
0	0	2	9	2
2	7	0	7	4
7	9	0	0	5
0	9	0	7	0

灰度级的修正





返回

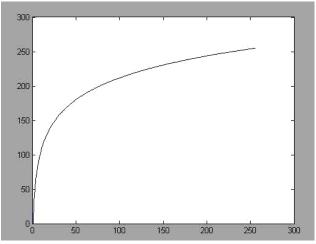


3.3 动态范围调整

二、非线性动态范围调整:

通常用取对数的方法。原因是<mark>人眼</mark>对信号的处理是有一个近似对数算子的 环节。

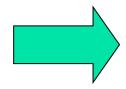
$$g = c \log_{10}(1+f)$$





二、非线性动态范围调整

1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0

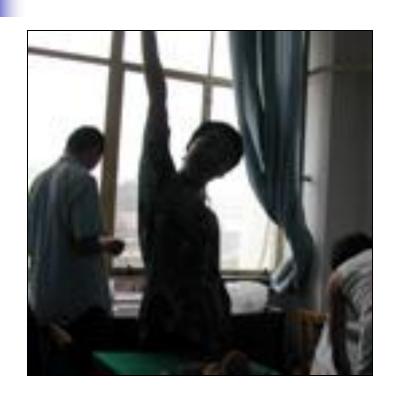


$$g=9*log(f+1)$$

3	5	9	9	9
4	3	5	8	5
5	8	0	8	6
8	9	4	0	7
4	9	4	8	0

作用:将暗的部分扩展,而将亮的部分抑制。

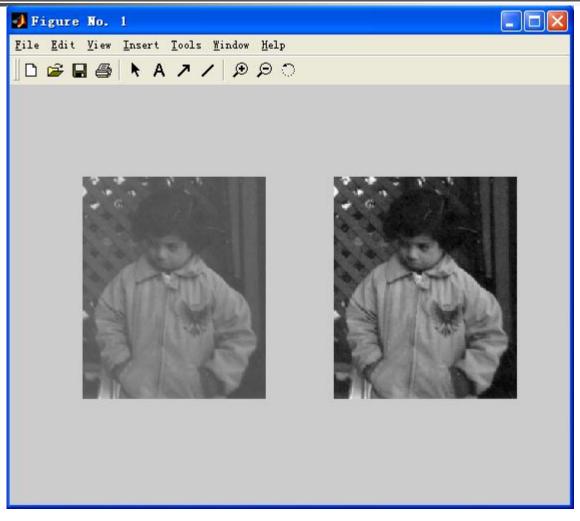
非线性动态范围调整





返回

g = log(f+1)



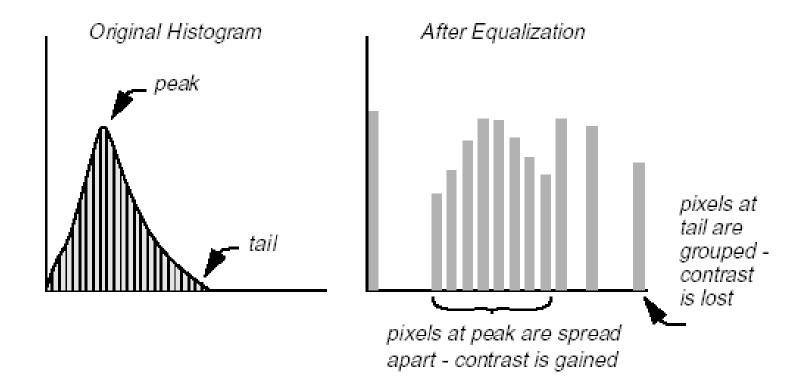


- I =imread('pout.tif');
- subplot(1,2,1);
- imshow(I);
- I=double(I);
- J=log(I+1);
- subplot(1,2,2); 1之间。
- imshow(mat2gray(J));

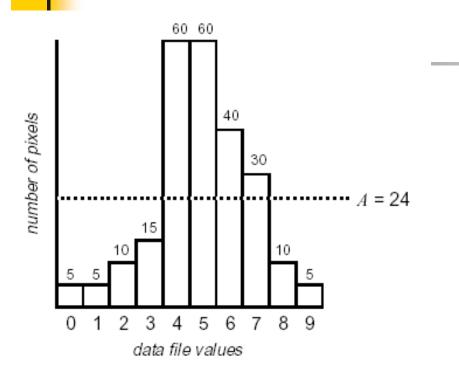
功能是实现图像矩阵的归一化操作。[1] 所谓"归一化"就是使矩阵的每个元素的值都在0和1之间。

3.5 直方图均衡化

- 直方图均衡方法的基本思想是对在图像中像素个数多的灰度级进行展宽,而对像素个数少的灰度级进行缩减。从而达到清晰图像的目的。
- 基本思想: 把原始图的直方图变换成为均匀分布的形式, 这样,就增加了像素灰度值的动态范围,从而达到增强图像整体对比度的效果。
- 非线性的增强方法;
- 将每个灰度区间等概率分布,代替了原来的随机分布,即增强后的每个灰度级内有大致相同的象元数;通过改变灰度区间来实现;
- 根据<mark>灰度值的出现频率来分配它们的亮度显示范围</mark>,频率 高的部分被增强了,频率低的部分被压缩。

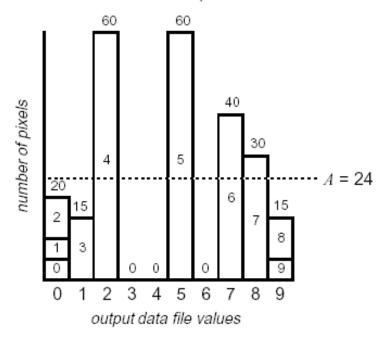


效果:增强了峰值处的对比度,两端(最亮和最暗)的对比度减弱了



原始的直方图

numbers inside bars are input data file values



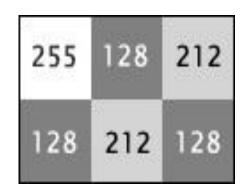
均衡化后的直方图

Photoshop的算法

Photoshop通过对比度拉伸的方法使最暗的像素值变为0,其它像素也相应变暗,最亮的像素保持255不变。对比度拉伸后的效果可能会比经典算法稍显偏暗。

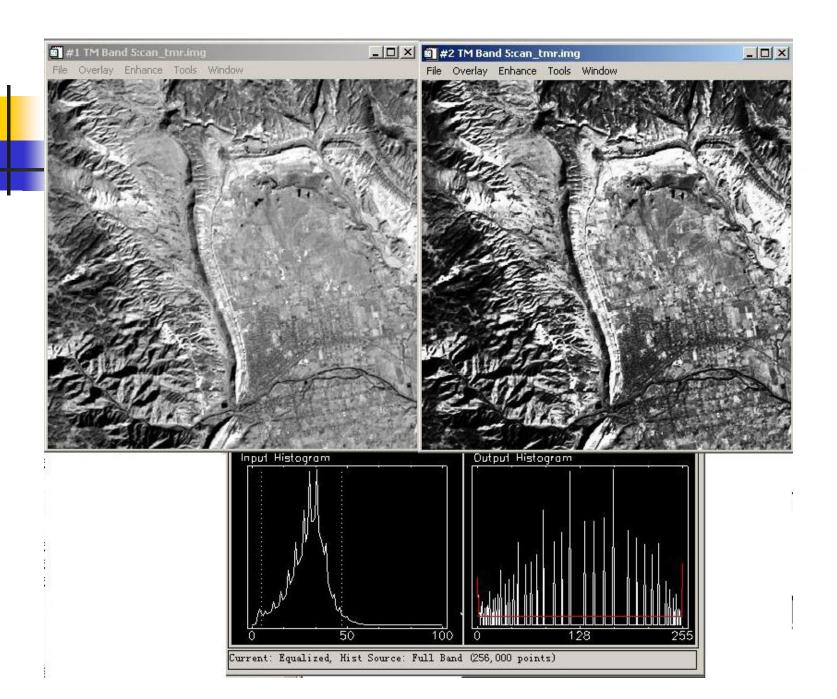
Photoshop的算法

- 对比度拉伸的公式: C = (Level Min) * Scale = (Level Min) * 255 / (255-Min)
- 原图均衡化之后的灰度值分别是128、212、255,为了精确,我们使用保留2位小数的
- 形式(127.50、212.42、255.00)来进行对比度拉伸的计算。
- Min = 127.50 '均衡化之后的最小值
- Scale = 255/(255-Min) = 2
- (127.50-Min)*Scale = 0*2 = 0
- (212.42-Min)*Scale = 84.92*2 = 170
- (255.00-Min)*Scale = 127.5*2 = 255
- '新的映射表:
- **50** -> 0
- **100 -> 170**
- **200 -> 255**



对比

200	50	100	原图
50	100	50	
255	128	212	经典算法
128	212	128	
255	0	170	Photoshop
0	170	0	

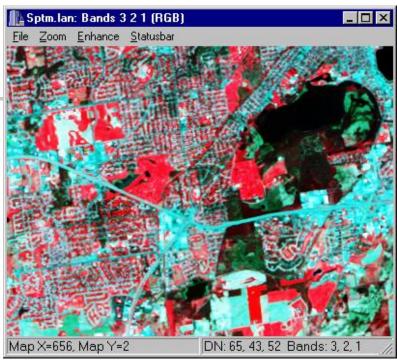


彩色算法

彩色的直方图均衡化其实就是对<mark>图像某个或多个颜色通道</mark>进行<mark>灰度直方图均衡化运算</mark>,常见的有以下几种方法:

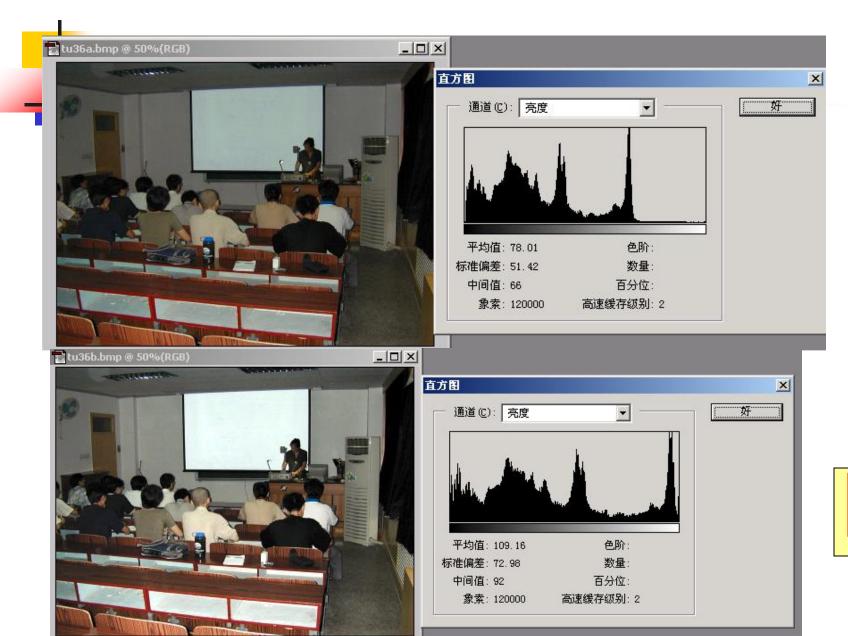
- 统计所有RGB颜色通道的直方图的数据并做均衡化运算,然后根据均衡化所得的映射表分别替换R、G、B通道颜色值。
- 分别统计R、G、B颜色通道的直方图的数据并做均衡 化运算,然后根据R、G、B的映射表分别替换R、G、B 通道颜色值。
- 用亮度公式或求RGB的平均值的方式计算亮度通道, 然后<mark>统计亮度通道的直方图</mark>的数据并做均衡化运算, 然后根据映射表分别替换R、G、B通道颜色值。





直方图均衡化实例

直方图均衡化的效果



算法:

设f、g分别为原图象和处理后的图像。

1) 求出原图f的灰度直方图,设为h。 h为一个256维的向量。



f

1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0



h

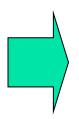
2) 求出图像f的总体像素个数

Nf=m*n (m,n分别为图像的长和宽)

计算每个灰度级的像素个数在整个图像中所占的百分比。

hs(i)=h(i)/Nf (i=0,1,...,255)





hs

0	0.12
1	0.08
2	0.16
3	0.16
4	0.04
5	0.04
6	0.16
7	0.04
8	0.08
9	0.12

3) 计算图像各灰度级的累计分布hp。

$$hp(i) = \sum_{k=0}^{i} h(k)$$

$$i = 1, 2, \dots, 255$$



hs

0	0.12
1	0.08
2	0.16
3	0.16
4	0.04
5	0.04
6	0.16
7	0.04
8	0.08
9	0.12



hp

0	0.12
1	0.20
2	0.36
თ	0.52
4	0.56
5	0.60
6	0.76
7	0.80
8	0.88
9	1.00

4) 求出新图像g的灰度值。

$$g = 255 \cdot hp(i)$$
 $i = 1, 2, ..., 255$
 $g = 0$ $i = 0$

hp

0	0.12
1	0.20
2	0.36
3	0.52
4	0.56
5	0.60
6	0.76
7	0.80
8	0.88
9	1.00



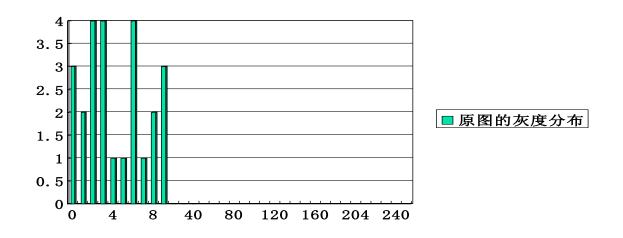
g

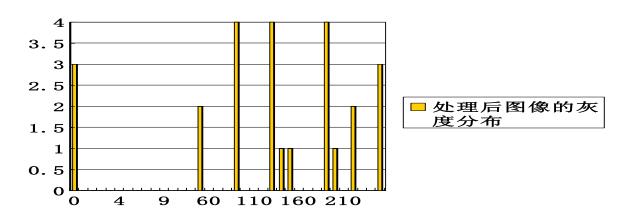
51	133	255	255	224
92	51	133	204	133
133	194	0	194	143
194	224	92	0	153
92	255	92	194	0

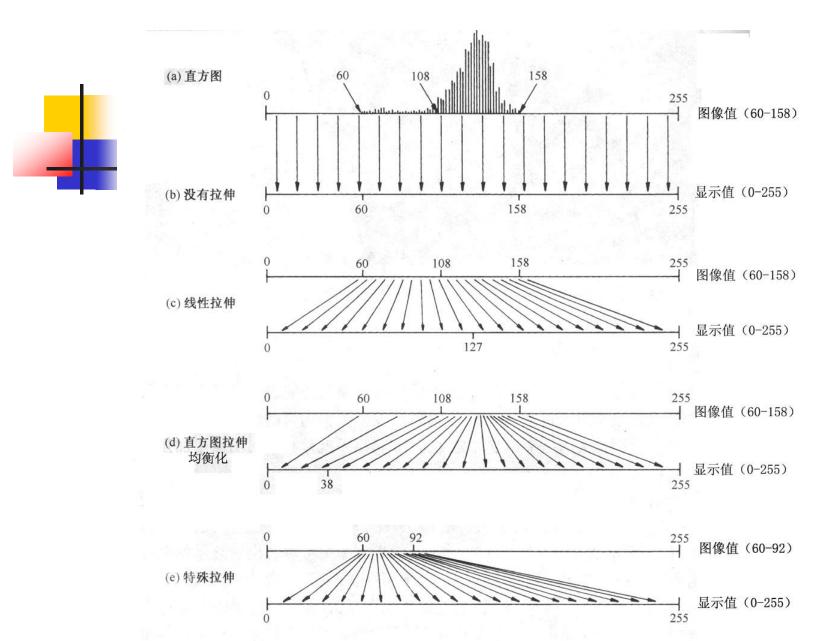
f

1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0









反差拉伸增强的原理



3.5 假彩色与伪彩色

一、假彩色

假彩色是指将<mark>一幅彩色图像映射为另一幅彩色图像,</mark>从而达到<mark>增强对比度</mark>的目的。

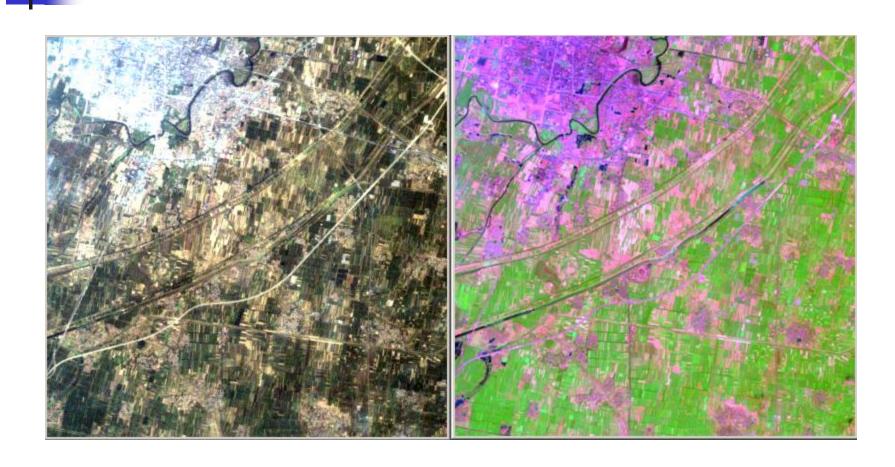
假彩色效果例图





返回

标准假彩色 (TM432)





3.6 假彩色与伪彩色

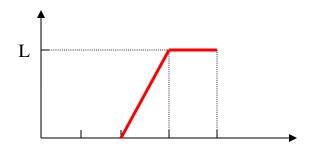
二、伪彩色

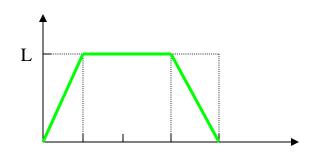
由于人眼分辨不同彩色的能力比分别不同的灰度级的能力强,因此,把人眼无法区别的灰度变化,施以不同的彩色来提高识别率,这便是伪彩色增强的基本依据。

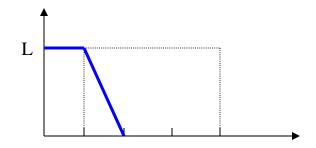


3.5 假彩色与伪彩色

■ 一种伪彩色的算法:

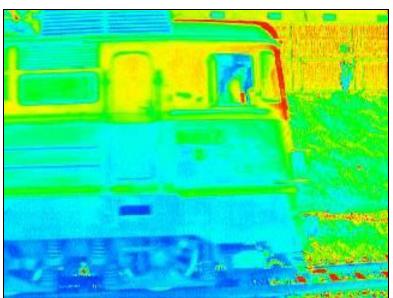


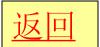


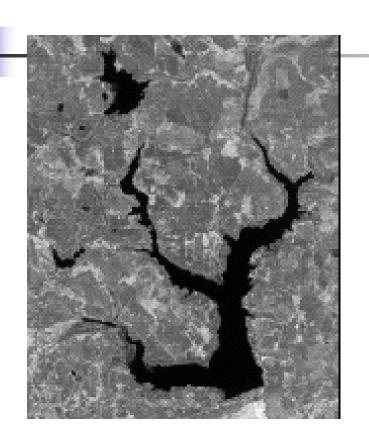


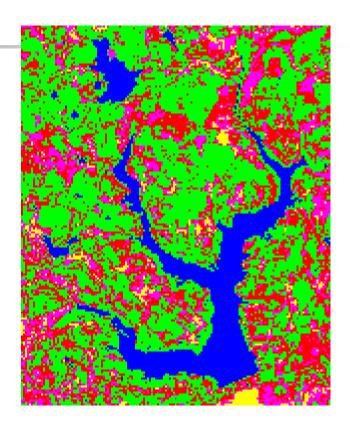
伪彩色效果图

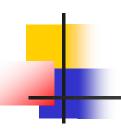












图像的几何变换

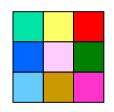
我们知道,图像是对三维实际景物的平面投影。为了观测需要,常常需要进行各种不同的几何变换。注意一点,实际上几何变换不改变像素值,而是改变像素所在的位置。





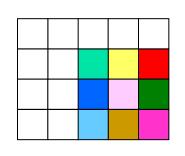
一、图像的平移

$$\begin{cases} x' = x + \Delta x \\ y' = y + \Delta y \end{cases}$$



$$\Delta x = 1, \Delta y = 2$$



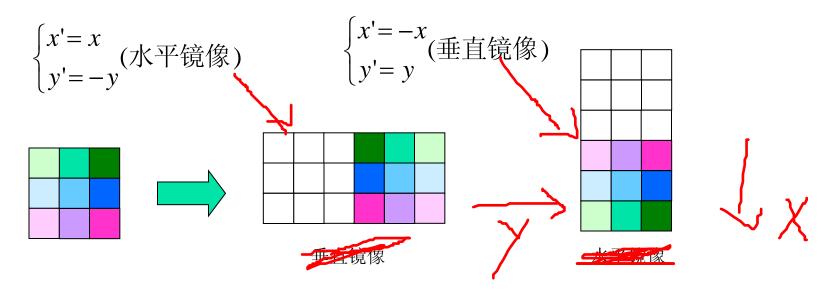




注意: 平移后的景物与原图像相同,但"画布"一定是扩大了。否则就会丢失信息。



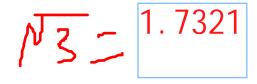
二、图像的镜像

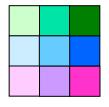


注意: 做镜像时,实际上需要对坐标先进行 平移,否则将出错。因为矩阵的下标不能为 负。

三、图像的旋转

$$\begin{cases} x' = x \cos \theta - y \sin \theta \\ y' = x \sin \theta + y \cos \theta \end{cases}$$



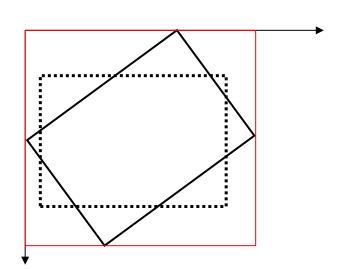


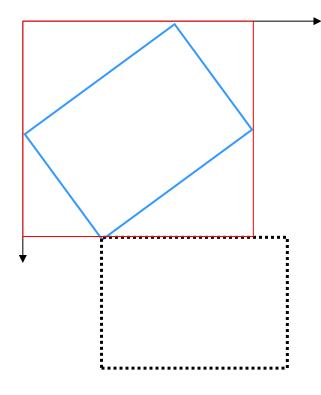
$$\begin{cases} x' = 0.866x - 0.5y \\ y' = 0.5x + 0.866y \end{cases}$$

$$x'_{min} = 0.866 - 0.5*3 = -0.634$$
 $x'_{max} = 0.866*3 - 0.5 = 2.098$
 $y'_{min} = 0.866 + 0.5 = 1.366$
 $y'_{max} = 0.866*3 + 0.5*3 = 4.098$

图像的旋转注意点:

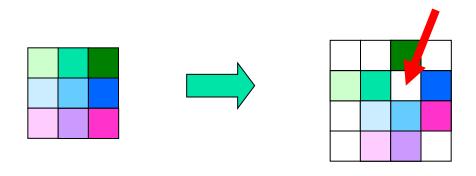
1) 图像旋转之前,为了避免 信息的丢失,一定有平移 坐标,具体的做法有如图 所示的两种方法。





图像的旋转注意点:

2) 图像旋转之后,会出现许多的空洞点,对 这些空洞点必须进行填充处理,否则画面 效果不好。称这种操作为插值处理。

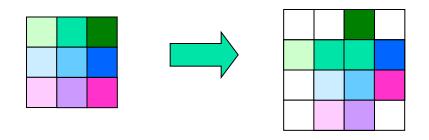




- 最简单的方法是行插值或是列插值方法:
- 1. 找出当前行的最小和最大的非白点的坐标,记作: (i,k1)、(i,k2)。
- 2. 在(k1,k2)范围内进行插值,插值的方法 是:空点的像素值等于前一点的像素值。
- 3. 同样的操作重复到所有行。



4.1 图像的位置变换



经过插值处理之后, 图像效果就变得自然。



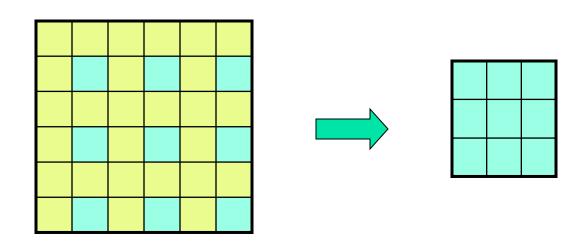
一、图像的缩小

图像的缩小一般分为按比例缩小和不按比例 缩小两种。图像缩小之后,因为承载的信息 量小了,所以画布可相应缩小。



1. 图像按比例缩小:

最简单的是<mark>减小一半</mark>,这样只需取原图的偶<mark>(奇)数行和偶(奇)数列</mark>构成新的图像。



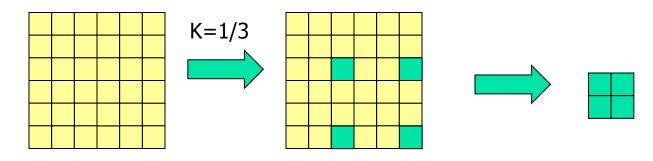
如果图像按任意比例缩小,则需要计算选择的行列。

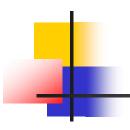
M*N大小的图像缩小为: kM*kN大小,(k<1)。

设旧图像是F(x,y),新图像是I(x,y)

则: I(x,y)=F(int(c*x),int(c*y)) c=1/k

I(x,y)=F(int(3x),int(3y))





2. 图像不按比例缩小:

这种操作因为在x方向和y方向的缩小比例不同,一定会带来图像的几何畸变。



二、图像的放大

图像的<mark>缩小操作</mark>中,是在现有的信息里<mark>如何挑选</mark> 所需要的有用信息。

图像的<mark>放大操作</mark>中,则需<mark>对尺寸放大后所多出来的空格填入适当的值,这是信息的估计问题</mark>,所以较图像的缩小要难一些。

图像不按比例缩小方法:

M*N大小的图像缩小为: k_1M*k_2N 大小, $(k_1<1, k_2<1)$ 。

设旧图像是F(x,y),新图像是I(x,y)

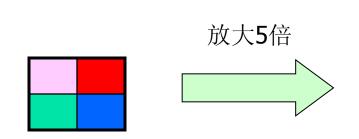
则: $I(x,y)=F(int(c_1*x),int(c_2*y))$

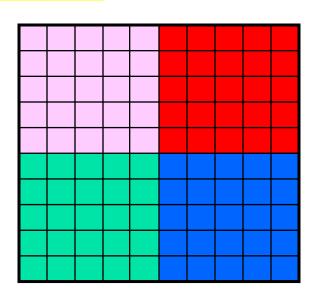
 $c_1 = 1/k_1$ $c_2 = 1/k_2$



1. 按比例放大图像

如果需要将原图像<mark>放大k倍</mark>,则将一个像 素值添在新图像的**k*k**的子块中。







思考一个问题:

如果<mark>放大倍数太大,按照前面的方</mark>法处理会出现<u>马赛克</u>效应。如果这个问题交给你,有没有办法解决?或者 至少有所改善?

2. 图像的任意不成比例放大:

这种操作由于x方向和y方向的放大倍数不同,一定带来图像的几何畸变。

放大的方法是:

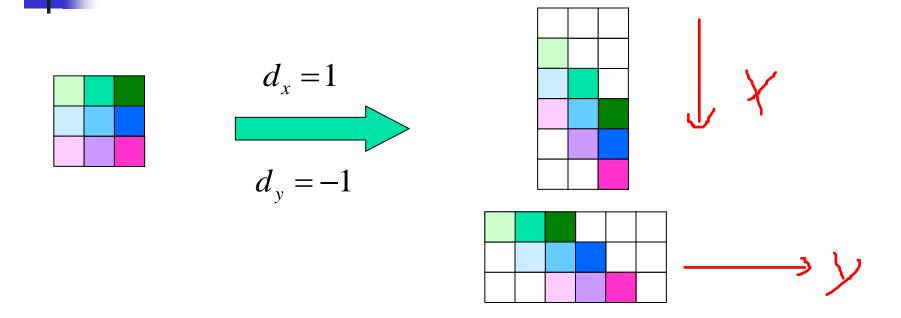
将原图像的<mark>一个像素</mark>添到新图像的一个 k₁*k₂的子块</mark>中去。

三、图像的错切变换

图像的错切变换实际上是<mark>景物在平面上的</mark> 上垂直投影效果。

$$\begin{cases} x' = x + d_x y \\ y' = y \end{cases} (x 方 向 的 错 切)$$

$$\begin{cases} x' = x \\ y' = y + d_y x \end{cases} (y 方 向 的 错 切)$$



可以看到,错切之后原图像的像素排列方向改变。与 前面<mark>旋转不同的是,x方向与y方向独立变化</mark>。

四、几何畸变的矫正

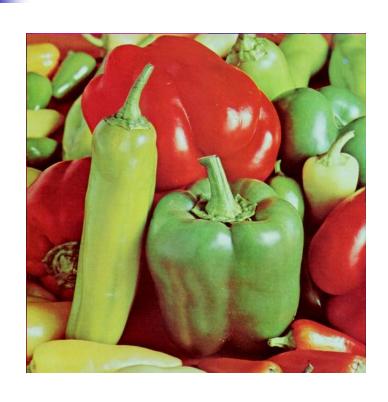
受到错切变换效果的启发,将其进行简单的延伸,如教材58页,当景物在图像上是非垂直投影时,可以通过几何变换将其进行矫正。

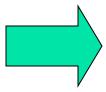
矫正方法为:

$$\begin{cases} x' = a_1 x + b_1 y + c_1 \\ y' = a_2 x + b_2 y + c_2 \end{cases}$$

变换参数可通过对应点的坐标来确定。

图像的减半缩小效果

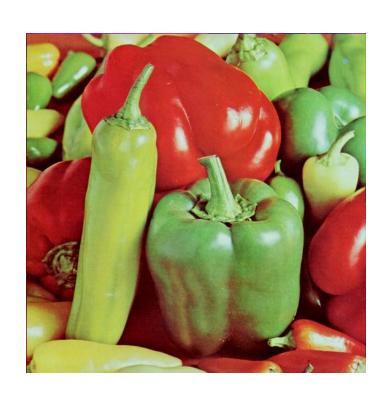






返回

图像的按比例缩小效果









图像的不按比例任意缩小









图像的成倍放大效果

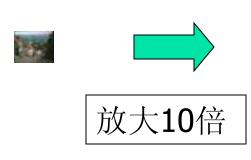






返回

图像大比例放大时的马赛克效应













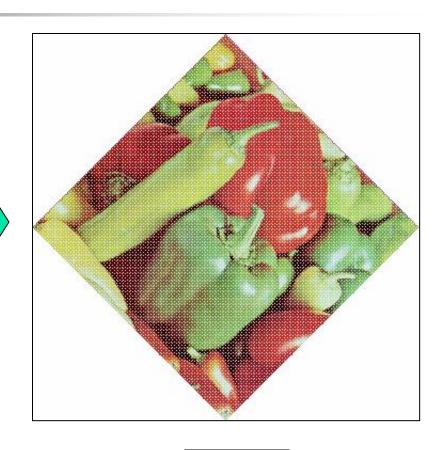


返回

图像的旋转效果

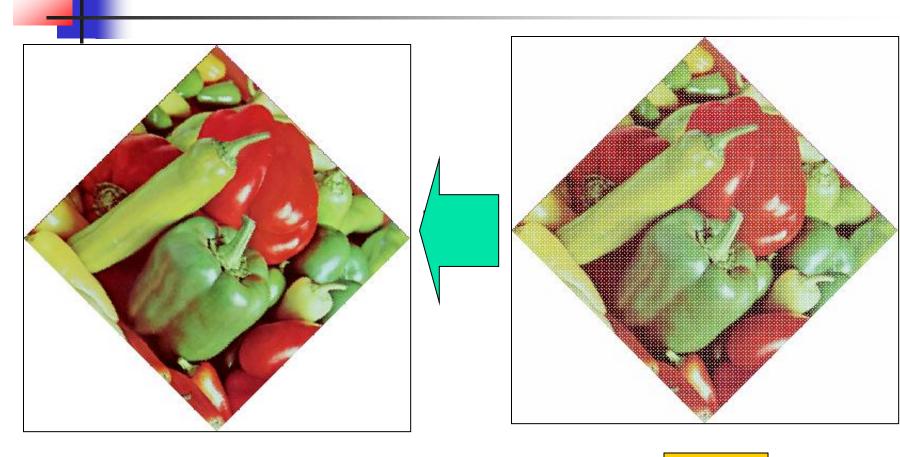






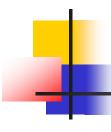


图像旋转中的插值处理效果







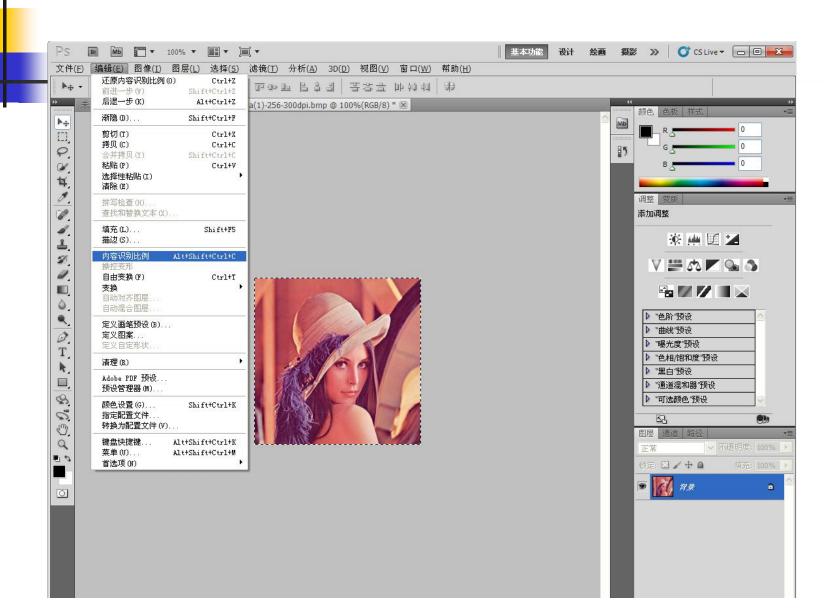






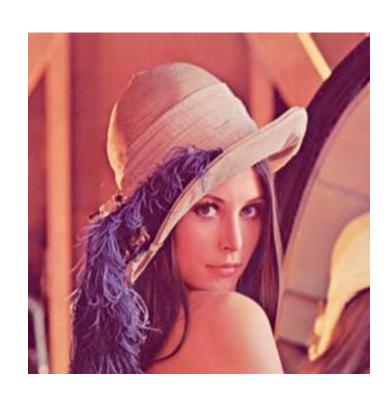








256*256 缩放到221*256

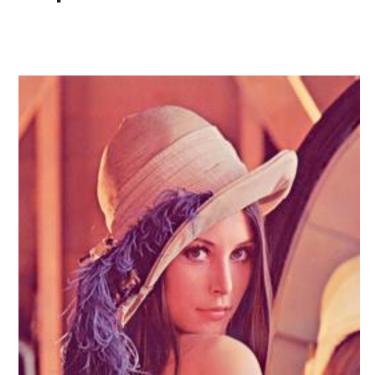








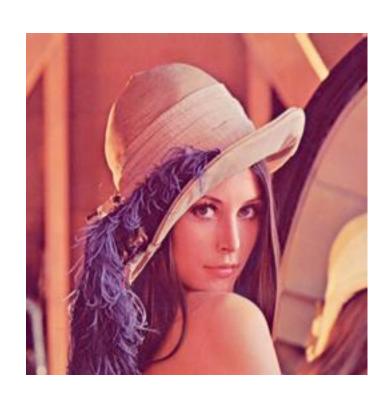
256*256 缩放到256*206







256*256 缩放到256*164









512*512 405*512

