

机器视觉与图像处理

第3讲 图像分析基础和图像变换

汪凯巍
2019-03-12

部分资料取自互联网，版权归原作者所有

回顾

第一讲 绪论

机器视觉的定义、系统组成——“机器视觉是很有用的”

第二讲 图像的获取

图像传感器、镜头、光照——“好的图像成功一半”

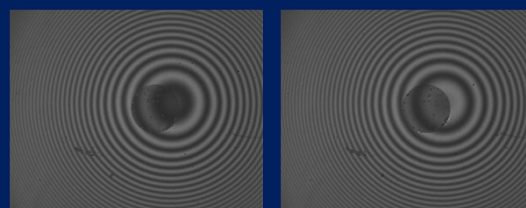
第三讲？

如何消除图像中的周期性干扰



陶瓷基片上有周期性分割线

两幅反相的干涉图中如何找到图像中间的低反射率圆斑？



$$I_0 = a + b \cos(\varphi)$$

$$I_2 = a + b \cos(\varphi + \pi)$$

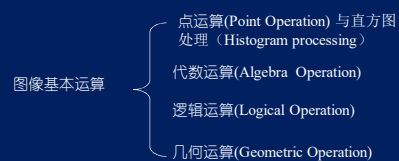
内容

- 图像的点处理-直方图、线性变换、对数变换、
- 图像几何变换（平移、镜像、缩放、旋转）
- 图像空间域滤波
- 图像频率域滤波-傅立叶变换、余项变换、KT变换、小波变换

图像基本运算

图像基本运算的分类

按图像处理运算的数学特征, 图像基本运算可分为:



图像基本运算

点运算

点运算是指对一幅图像中每个像素点的灰度值进行计算的方法。

灰度直方图 (Gray-level Histogram)

表示图像中具有某种灰度级的像素的个数，反映了图像中每种灰度出现的频率。

代数运算、逻辑运算

代数运算或逻辑运算是指将两幅或多幅图像通过对应像素之间的加、减、乘、除运算或逻辑与、或、非运算得到输出图像的方法。

7

图像基本运算

几何运算

几何运算就是改变图像中物体对象（像素）之间的空间关系。

从变换性质来分，几何变换可以分为图像的**位置变换**（平移、镜像、旋转）、**形状变换**（放大、缩小）以及图像的复合变换等。

8

3.2 点运算 (Point Operation)

1. 点运算的定义

设输入图像的灰度为 $f(x, y)$ ，输出图像的灰度为 $g(x, y)$ ，则点运算可以表示为：

$$g(x, y) = T[f(x, y)]$$

灰度变换函数

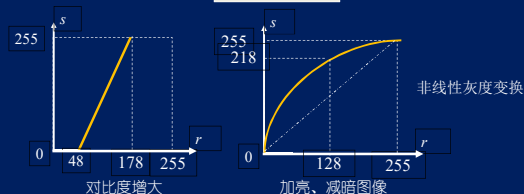
其中 $T[\]$ 是对 f 在 (x, y) 点值的一种数学运算，即点运算是一种**像素的逐点运算**，是灰度到灰度的映射过程，故称 $T[\]$ 为灰度变换函数。

9

3.2 点运算 (Point Operation)

若令 $f(x, y)$ 和 $g(x, y)$ 在任意点 (x, y) 的灰度级分别为 r 和 s ，则灰度变换函数可简化表示为：

$$s = T[r]$$



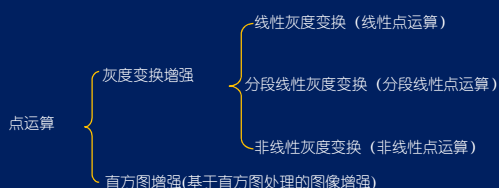
点运算可以改变图像数据所占据的灰度值范围，从而改善图像显示效果。

10

3.2 点运算 (Point Operation)

2. 点运算的分类

点运算又称为“对比度增强”、“对比度拉伸”、“灰度变换”等，按灰度变换函数 $T[\]$ 的性质，可将点运算分为：



11

3.2.1 线性点运算 (Linear Point Operation)

1. 线性点运算

线性点运算的灰度变换函数形式可以采用线性方程描述，即

$$s = ar + b$$

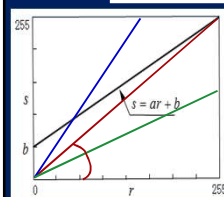


图 3.3 线性点运算

黑线： $0 < a < 1, \quad b > 0$ 输出灰度压缩

红线： $a = 1, \quad b = 0$ 输出灰度不变

蓝线： $a > 1, \quad b = 0$ 输出灰度扩展 整体变亮

绿线： $0 < a < 1, \quad b = 0$ 输出灰度压缩，整体变暗

12

3.2.1 线性点运算 (Linear Point Operation)

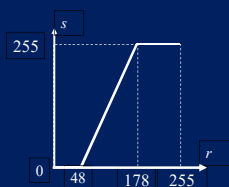
线性点运算的应用

$$s = ar + b$$

- 1) 如果 $a > 1$, 输出图像的对比度增大 (灰度扩展)



变换前



变换后

对比度增大

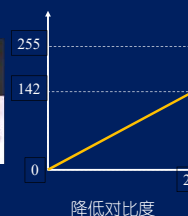
13

3.2.1 线性点运算 (Linear Point Operation)

- 2) 如果 $0 < a < 1$, 输出图像的对比度减小 (灰度压缩)



变换前



降低对比度



变换后

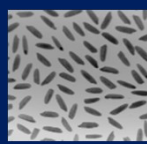
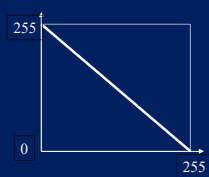
14

3.2.1 线性点运算 (Linear Point Operation)

- 3) 如果 a 为负值, 暗区域将变亮, 亮区域将变暗



变换前



变换后

15

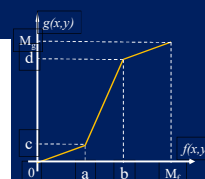
3.2.1 线性点运算 (Linear Point Operation)

2、分段线性点运算

将感兴趣的灰度范围线性扩展, 相对抑制不感兴趣的灰度区域。

设 $f(x, y)$ 灰度范围为 $[0, M_f]$, $g(x, y)$ 灰度范围为 $[0, M_g]$,

$$g(x, y) = \begin{cases} \frac{M_g - d}{M_f - b} [f(x, y) - b] + d & b \leq f(x, y) \leq M_f \\ \frac{d - c}{b - a} [f(x, y) - a] + c & a \leq f(x, y) < b \\ \frac{c}{a} f(x, y) & 0 \leq f(x, y) < a \end{cases}$$



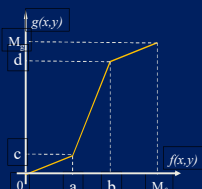
16

3.2.1 线性点运算 (Linear Point Operation)

分段线性点运算的应用



变换前



变换后

17

3.2.2 非线性点运算 (Non-Linear Point Operation)

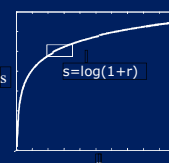
1、非线性点运算

非线性点运算的输出灰度级与输入灰度级呈非线性关系, 常见的非线性灰度变换为对数变换和幂次变换。

- 1)、对数变换

对数变换的一般表达式为: $s = c \log(1 + r)$

其中 c 是一个常数。



对数曲线图

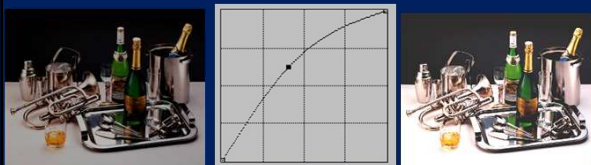
低灰度区扩展, 高灰度区压缩。
图像加亮、减暗。

非线性拉伸不是对图像的整个灰度范围进行扩展, 而是有选择地对某一灰度值范围进行扩展, 其他范围的灰度值则有可能被压缩。

18

3.2.2非线性点运算 (Non-Linear Point Operation)

非线性点运算应用实例1



对比度拉伸效果：图像加亮、减暗

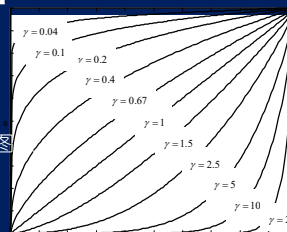
19

3.2.2非线性点运算 (Non-Linear Point Operation)

2)、幂次变换

幂次变换的一般形式 $s = cr^\gamma$ 其中C和 γ 为正常数。 $0 < \gamma < 1$

加亮、减暗图

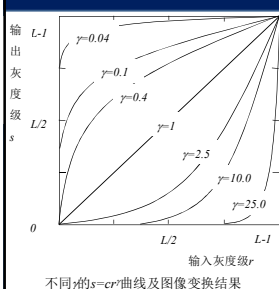
 $\gamma > 1$

加暗、减亮图像

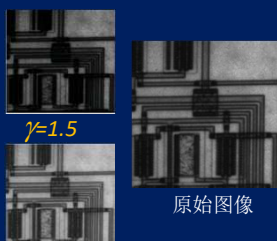
20

3.2.2 非线性点运算 (Non-Linear Point Operation)

非线性点运算应用实例3

不同 γ 的 $s=cr^\gamma$ 曲线及图像变换结果

加暗、减亮图像

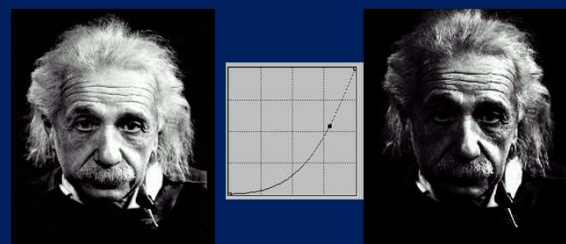


原始图像

加亮、减暗图像

21

3.2.2非线性点运算 (Non-Linear Point Operation)



加暗、减亮图像

22

3.2.2非线性点运算 (Non-Linear Point Operation)

思考问题:

1、点运算是否会改变图像内像素点之间的空间位置关系？

点运算是一种像素的逐点运算，它与相邻的像素之间没有运算关系，点运算不会改变图像内像素点之间的空间位置关系。

2、非线性运算与分段线性运算的区别？

非线性拉伸不是通过在不同灰度值区间选择不同的线性方程来实现对不同灰度值区间的扩展与压缩，而是在整个灰度值范围内采用统一的非线性变换函数，利用函数的数学性质实现对不同灰度值区间的扩展与压缩。

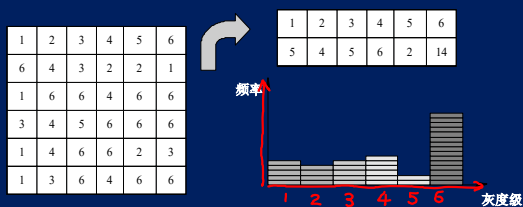
23

直方图处理

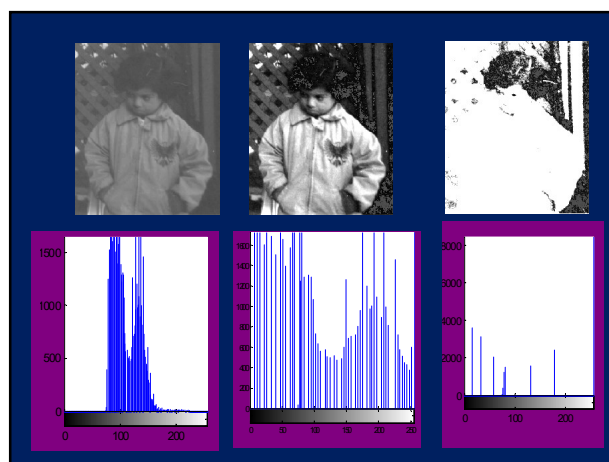
灰度直方图

灰度直方图 (Gray-level Histogram)

表示图像中具有某种灰度级的像素的个数，反映了图像中每种灰度出现的频率。



有什么作用呢？



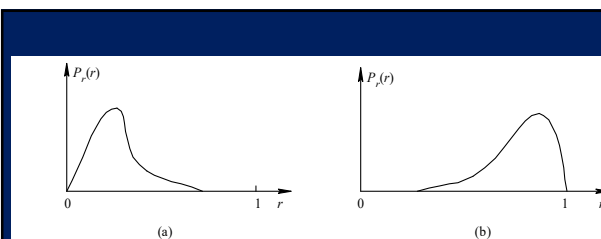
设 r 代表图像中像素灰度级，作归一化处理后， r 将被限定在 $[0, 1]$ 之内。在灰度级中， $r=0$ 代表黑， $r=1$ 代表白。

图像像素 r_i 出现的概率为

$$p(r_i) = \frac{\text{灰度值为 } r_i \text{ 的像素数}}{\text{图像上总的像素数}}$$

且有

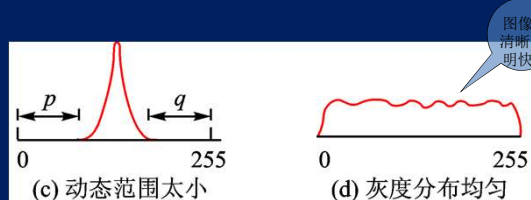
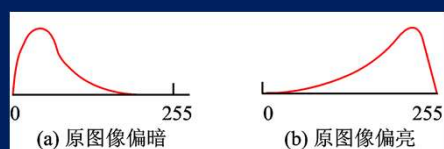
$$\sum_{i=0}^{L-1} p(r_i) = 1$$



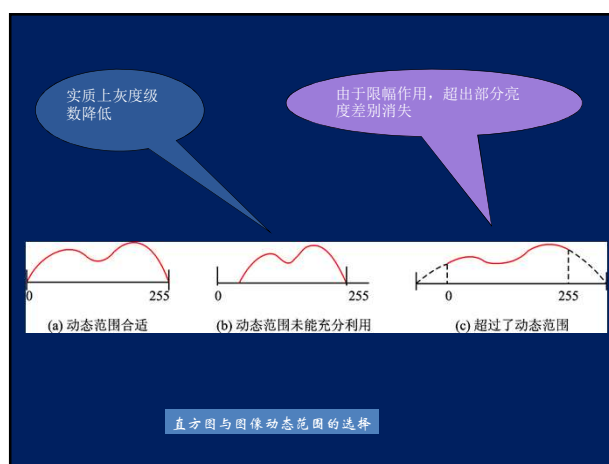
图像灰度分布概率密度函数

MATLAB的imhist函数

imhist(I, N) %在长度为N的灰度条上显示灰度图像I的直方图



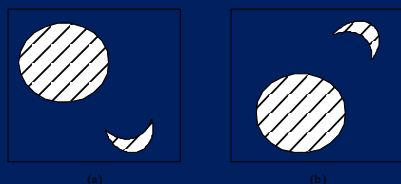
观测直方图分析原图像性质



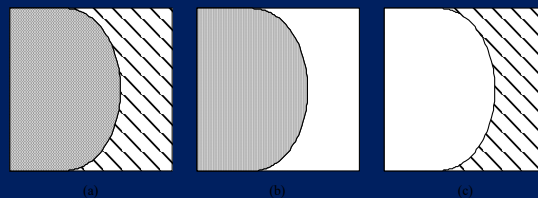
2. 直方图的性质：

(1) 与位置无关。只包含了该图像中某一灰度值的像素出现的概率，而丢失了其所在位置的信息。

(2) 任一幅图像，都能惟一地确定出一幅与它对应的直方图，但不同的图像，可能有相同的直方图。



(3) 具有可迭加性。由于直方图是对具有相同灰度值的像素统计得到的，因此，一幅图像各子区的直方图之和就等于该图像全图的直方图。



• 图像——直方图

不可逆变换，多对一的变换

□ 直方图是多对一的映射结果，即多个图像可以生成相同的直方图，因此直方图作为一阶统计特征未反映相邻点之间的关系。

□ 但反映了图像的灰度散布范围等特征，在很多场合下，往往是重要特征。

直方图修正

一幅给定图像的灰度级分布在 $0 \leq r \leq 1$ 范围内。可以对 $[0, 1]$ 区间内的任一个 r 值进行如下变换： $s = T(r)$

通过上述变换，每个原始图像的像素灰度值 r 都对应产生一个 s 值。变换函数 $T(r)$ 应满足下列条件：

- (1) 在 $0 \leq r \leq 1$ 区间内， $T(r)$ 值单调增加；
- (2) 对于 $0 \leq r \leq 1$ ，有 $0 \leq T(r) \leq 1$ 。

条件 (1) 保证了图像的灰度级从黑到白的次序不变，

条件 (2) 保证了映射变换后的像素灰度值在容许的范围内。

直方图均衡化

均衡化后直方图趋于平坦化，灰度间隔（动态范围）拉大，对比度加强，图像清晰，便于读取、分析和处理。



原始图像 均衡化结果

原始图像 均衡化结果

histeq函数实现直方图均衡化

$J = \text{histeq}(I, N)$

将灰度图像 I 进行均衡变换，

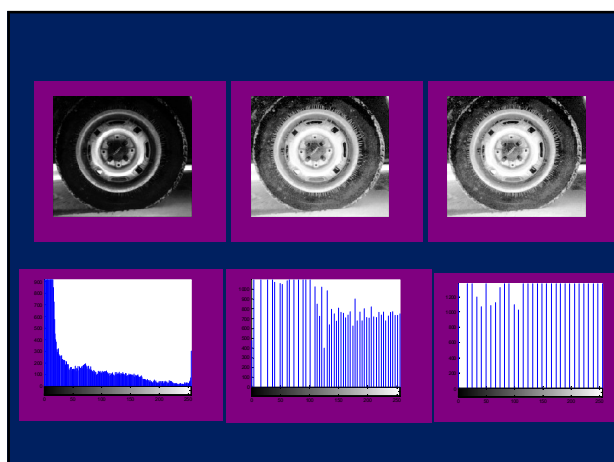
成为 N 个灰度级的图像

```

I = imread('tire.tif');
imshow(I);
figure;
imhist(I);
J = histeq(I, 64);
figure;
imshow(J);
figure;
imhist(J);

J = histeq(I, 32);
figure;
imshow(J);
figure;
imhist(J);

```



3.3代数运算与逻辑运算 (Algebra and Logical Operation)

1. 概念

代数运算是指两幅或多幅输入图像之间进行点对点的加、减、乘、除运算得到输出图像的过程。如果记输入图像为 $A(x, y)$ 和 $B(x, y)$ ，输出图像为 $C(x, y)$ ，则有如下四种形式：

代数运算的四种基本形式

$$\begin{cases} C(x, y) = A(x, y) + B(x, y) \\ C(x, y) = A(x, y) - B(x, y) \\ C(x, y) = A(x, y) \times B(x, y) \\ C(x, y) = A(x, y) \div B(x, y) \end{cases}$$

38

3.3代数运算与逻辑运算 (Algebra and Logical Operation)

逻辑运算：

逻辑运算是指将两幅或多幅图像通过对应像素之间的与、或、非逻辑运算得到输出图像的方法。

- 在进行图像理解与分析领域比较有用。
- 运用这种方法可以为图像提供模板
- 与其他运算方法结合起来可以获得某种特殊的效果。

39

3.3.1 加法运算(Addition)

1、加法运算

$$C(x, y) = A(x, y) + B(x, y)$$

主要应用举例：

- 去除“叠加性”随机噪音
- 生成图像叠加效果

40

3.3.1加法运算(Addition)

去除“叠加性”噪音

对于原图像 $f(x, y)$ ，有一个噪音图像集 $\{g_i(x, y)\}$ $i = 1, 2, \dots, M$
其中： $g_i(x, y) = f(x, y) + e_i(x, y)$

M个图像的均值为：

$$\begin{aligned} \overline{g(x, y)} &= \overbrace{f(x, y)}^{\text{混入噪声的图像}} + \overbrace{e(x, y)}^{\text{原始图像}} \\ \overline{g(x, y)} &= \frac{1}{M} \sum_{i=1}^M [f_i(x, y) + e_i(x, y)] \\ &= f(x, y) + \frac{1}{M} \sum_{i=1}^M e_i(x, y) \end{aligned}$$

当：噪音 $e_i(x, y)$ 为互不相关，且均值为0时，上述图像均值将降低噪音的影响。

41

3.3.1加法运算(Addition)

$$\overline{g(x, y)} = \frac{1}{M} \sum_{i=1}^M [f_i(x, y) + e_i(x, y)] \quad \text{则 } \overline{g(x, y)} \text{ 是 } f(x, y) \text{ 的无偏估计}$$

$$= f(x, y) + \frac{1}{M} \sum_{i=1}^M e_i(x, y)$$

$$\begin{aligned} \therefore E\{\overline{g(x, y)}\} &= E\left\{\frac{1}{M} \sum_{i=1}^M g_i(x, y)\right\} = \frac{1}{M} \sum_{i=1}^M E\{g_i(x, y)\} \\ &= \frac{1}{M} \sum_{i=1}^M [E\{f_i(x, y)\} + E\{e_i(x, y)\}] \\ &= \frac{1}{M} \sum_{i=1}^M f_i(x, y) = f(x, y) \end{aligned}$$

利用同一景物的多幅图像取平均，消除噪声。取M个图像相加求平均得到1幅新图像，一般选8幅取平均。

42

加法运算降低噪声影响

当：噪声 $e(x, y)$ 为互不相关，且均值为0时，上述图像均值将降低噪声的影响。

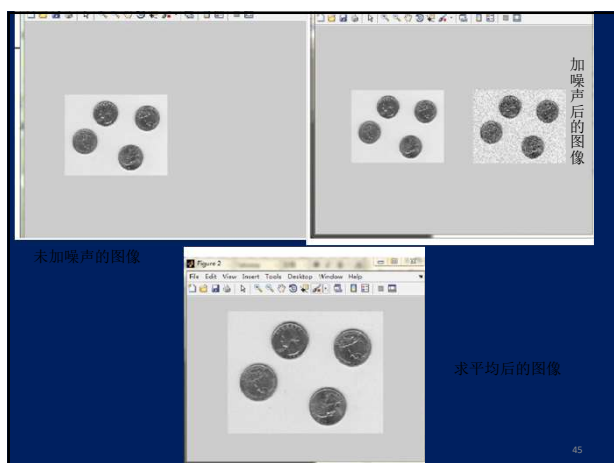
【例3.1】把一幅图像加上高斯噪声，再通过100次求平均的方法去除噪声，其MATLAB程序如下：

```
I=imread('eight.tif'); %读取一幅图片
J=imnoise(I,'gaussian',0,0.02); %向这幅图片加入高斯噪声
subplot(1,2,1),imshow(I); %显示图片
subplot(1,2,2),imshow(J); %显示图片
```

43

```
K=zeros(242,308);
%产生全零的矩阵，大小与图片的一样
for i=1:100
%循环100加入噪声
J=imnoise(I,'gaussian',0,0.02);
J1=im2double(J);
K=K+J1;
end
K=K/100;
>> figure;imshow(K);
```

44

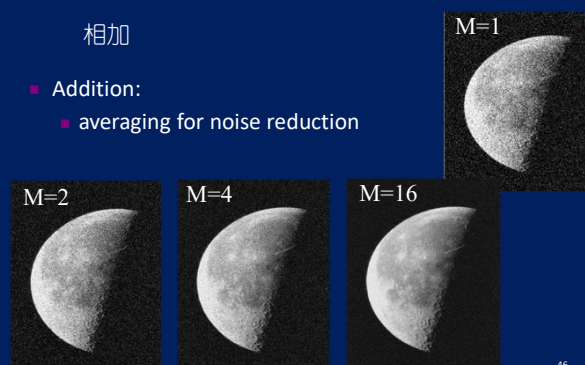


45

3.3.1加法运算(Addition)

相加

- Addition:
- averaging for noise reduction



46

3.3.1加法运算(Addition)

生成图像叠加效果：可以得到各种图像合成的效果，也可以用于两张图片的衔接。



47

3.3.2减法运算 (Subtraction)

减法运算

将同一景物在不同时间拍摄的图像或同一景物在不同波段的图像相减，这就是图像的减法运算。实际中常称为**差影法**。

$$C(x, y) = A(x, y) - B(x, y)$$

差值图像提供了图像间的差值信息，能用于指导动态监测、运动目标的检测和跟踪、图像背景的消除及目标识别等。

主要应用举例：

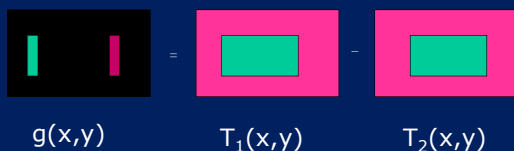
- 差影法 (检测同一场景两幅图像之间的变化)
- 混合图像的分离

48

3.3.2减法运算 (Subtraction)

检测同一场景两幅图像之间的变化

设：时刻1的图像为 $T_1(x, y)$ ，
时刻2的图像为 $T_2(x, y)$
 $g(x, y) = T_2(x, y) - T_1(x, y)$



49

3.3.2减法运算 (Subtraction)

差影法在应用

- 1、在银行金库内，摄像头每隔一固定时间拍摄一幅图像，并与上一幅图像做差影，如果图像差别超过了预先设置的阈值，则表明可能有异常情况发生，应自动或以某种方式报警；
- 2、用于遥感图像的动态监测，差值图像可以发现森林火灾、洪水泛滥，监测灾情变化等；
- 3、也可用于监测河口、海岸的泥沙淤积及监视江河、湖泊、海岸等的污染；
- 4、利用差值图像还能鉴别出耕地及不同的作物覆盖情况。

50

图像相减——运动检测



51

3.3.2减法运算 (Subtraction)

混合图像的分离（或消除背景干扰）



差影法进行混合图像的分离

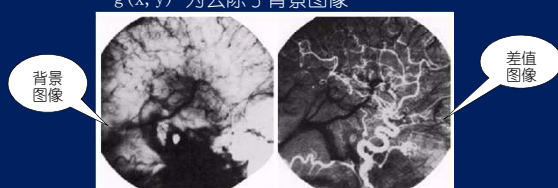
52

3.3.2减法运算 (Subtraction)

消除背景影响

去除不需要的叠加性图案

设：背景图像 $b(x, y)$ ，前景背景混合图像 $f(x, y)$
 $g(x, y) = f(x, y) - b(x, y)$
 $g(x, y)$ 为去除了背景图像



(a) 从病人头顶向下拍摄的X光照片 (b) 碘元素注入后拍摄的X光照片与背景图像的差值

53

3.3.3乘法运算(Multiplication)

乘法运算

$$C(x, y) = A(x, y) \times B(x, y)$$

主要应用举例：

图像的局部显示
改变图像的灰度级

图像的局部显示



54

3.3.3乘法运算(Multiplication)

改变图像的灰度级



(a) 原图

(b) 乘以1.2

(c) 乘以2

55

3.3.4除法运算(Division)

- 除法运算 $C(x, y) = A(x, y) \div B(x, y)$

简单的除法运算可用于改变图像的灰度级，常用于遥感图像处理中。

可产生对颜色和多光谱图像分析十分重要的比率图像。

在四种算术运算中，减法与加法在图像增强处理中最为有用。

56

3.3.5逻辑运算(Logical Operation)

“与”、“或”，“非”逻辑运算

逻辑运算主要以像素对像素为基础在**两幅或多幅图像**间进行。



(a) A图



(b) B图



(c) A、B相与结果图



(d) A、B相或结果图



(e) A取反结果图

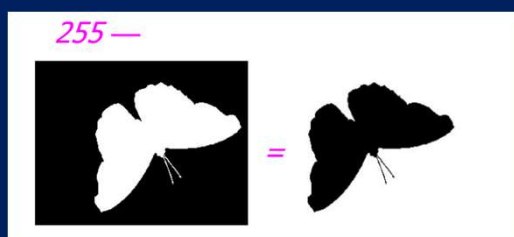
57

求反运算——获得阴图像



58

求反运算——求图像的补图像



59

逻辑运算

- 或运算

$$g(x, y) = f_1(x, y) \vee f_2(x, y)$$

- 主要应用：

合并子图像



60

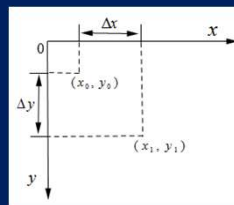
3.4 几何运算(Geometric Operation)

几何运算

- 几何运算就是**改变图像中物体对象（像素）之间的空间关系**。
- 从变换性质来分，几何变换可以分为图像的位置变换（平移、镜像、旋转）、形状变换（放大、缩小）以及图像的复合变换等。

61

3.4.1 图像的平移(Image Translation)



两点之间存在如下关系：

$$\begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + \Delta y \end{cases}$$

像素点的平移

62

齐次坐标

2D图像中的点坐标 (x, y) 表示成齐次坐标 (H_x, H_y, H) 。当 $H=1$ 时，则 $(x, y, 1)$ 就称为点 (x, y) 的**规范化齐次坐标**。

规范化齐次坐标的前两个数是相应二维点的坐标，没有变化，仅在原坐标中增加了 $H=1$ 的附加坐标。

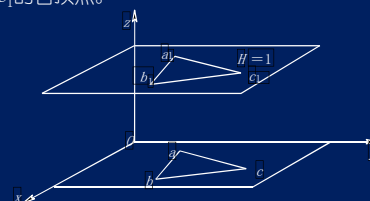
由点的齐次坐标 (H_x, H_y, H) 求点的**规范化齐次坐标** $(x, y, 1)$ ，可按如下公式进行：

$$x = \frac{H_x}{H} \quad y = \frac{H_y}{H}$$

63

齐次坐标

齐次坐标的几何意义相当于点 (x, y) 落在3D空间 $H=1$ 的平面上。如果将XOY平面内的三角形abc的各顶点表示成齐次坐标 $(x_i, y_i, 1)$ ($i=1, 2, 3$) 的形式，就变成 $H=1$ 平面内的三角形 $a_1 b_1 c_1$ 的各顶点。



64

3.4.1 图像的平移(Image Translation)

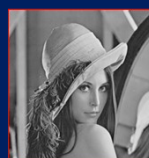
$$\begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + \Delta y \end{cases}$$

以矩阵形式表示平移前后的像素关系为：

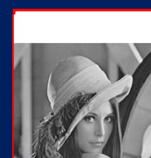
$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

65

3.4.1 图像的平移(Image Translation)



(a) 原始图像



(b) 平移后的图像

图像的平移

66

3.4.2 图像的镜像(Image Mirror)

图像的镜像 (Mirror) 是指原始图像相对于某一参照面旋转180° 的图像。

设原始图像的宽为 w 高 h 原始图像中的点为 (x_0, y_0)

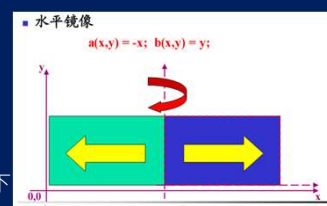
，对称变换后的点为 (x_1, y_1)

- (1) 水平镜像 (相对于 y 轴)

水平镜像的变换公式 如下:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

67



水平镜像的变换公式 如下

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

68

3.4.2 图像的镜像(Image Mirror)



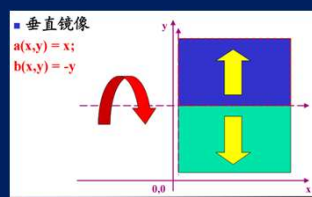
(a) 原始图像



(b) 水平镜像

图像水平镜像变换

69



70

图像的镜像(Image Mirror)

- (2) 垂直镜像 (相对于 x 轴)

垂直镜像的变换公式为如下:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

71

3.4.2 图像的镜像(Image Mirror)



(a) 原始图像



(b) 垂直镜像

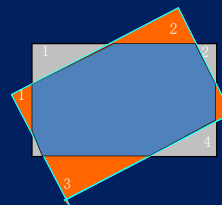
图像垂直镜像变换

72



3.4.3 图像的旋转 (Image Rotation)

一般图像的旋转是以图像的中心为原点，旋转一定的角度，即将图像上的所有像素都旋转一个相同的角度。



3.4.3 图像的旋转 (Image Rotation)

- 设原始图像的任意点 $A_0(x_0, y_0)$ 经旋转角度 β 以后到新的位置 $A(x, y)$ 为表示方便，采用极坐标形式表示，原始的角度为 α ，如下图所示：

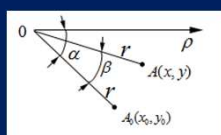


图3.12 图像的旋转

原始图像的点 $A_0(x_0, y_0)$

的坐标如下：

$$\begin{cases} x_0 = r \cos \alpha \\ y_0 = r \sin \alpha \end{cases}$$

75

3.4.3 图像的旋转 (Image Rotation)

旋转到新位置以后点 $A(x, y)$ 的坐标如下：

$$\begin{cases} x = r \cos(\alpha - \beta) = r \cos \alpha \cos \beta + r \sin \alpha \sin \beta \\ y = r \sin(\alpha - \beta) = r \sin \alpha \cos \beta - r \cos \alpha \sin \beta \end{cases}$$

$$\begin{cases} x = x_0 \cos \beta + y_0 \sin \beta \\ y = -x_0 \sin \beta + y_0 \cos \beta \end{cases}$$

- 图像旋转用矩阵表示如下：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

76

3.4.3 图像的旋转 (Image Rotation)



(a) 原图

(b) 旋转图

(c) 旋转图

如：车牌的旋转校正

77

3.4.3 图像的旋转 (Image Rotation)

图像旋转之后，由于数字图像的坐标值必须是整数，因此，可能引起图像部分像素点的局部改变，因此，这时图像的大小也会发生一定的改变。

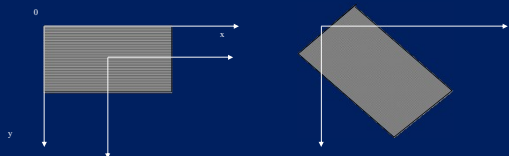
若图像旋转角 $\beta=45^\circ$ 时，则变换关系如下：

$$\begin{cases} x = 0.707x_0 + 0.707y_0 \\ y = -0.707x_0 + 0.707y_0 \end{cases}$$

78

图像绕任意点旋转

上述的旋转是绕坐标轴原点 $(0, 0)$ 进行的，如果是绕某一个指定点 (a, b) 旋转，则先将坐标系平移到该点，再进行旋转，然后将旋转后的图像平移回原坐标系。例如，我们这里以图像的中心为旋转中心：



79

3.4.3 图像的旋转(Image Rotation)

图像旋转 $\beta = 45^\circ$ 时，则变换关系如下：

$$\begin{cases} x = 0.707x_0 + 0.707y_0 \\ y = -0.707x_0 + 0.707y_0 \end{cases}$$

以原始图像的点 $(1, 1)$ 为例，旋转以后，均为小数，经舍入后为 $(1, 0)$ ，产生了位置误差。因此，图像旋转之后，可能会出现一些空白点，需要对这些空白点进行灰度级的插值处理，否则影响旋转后的图像质量。

80



旋转前的图像

81



图旋转15°并进行插值处理的图像

82

3.4.4 图像的缩放(Image Zoom)

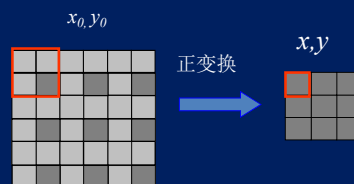
以 $a = 1/2$ 为例，即图像被缩小为原始图像的一半。图像被缩小一半以后根据目标图像和原始图像像素之间的关系，有如下两种缩小方法。

- 第一种方法是取原图像的偶数行列组成新图像
- 另一种方法是取原图像的奇数行列组成新图像

83

3.4.4 图像的缩放(Image Zoom)

缩小 $x = x_0/2$
 $y = y_0/2$

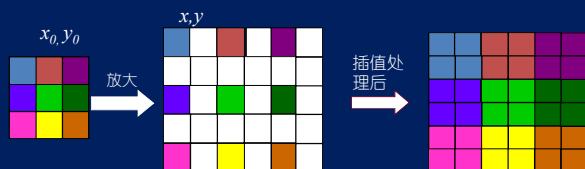


84

图像比例缩放变换

$$\text{放大} \quad \begin{aligned} x &= 2x_0 \\ y &= 2y_0 \end{aligned}$$

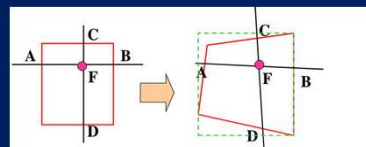
但放大后图像的像素点 $(0, 1)$ 对应于原始图中的像素点 $(0, 0.5)$ ， $(1, 0)$ 对应于原始图中的 $(0.5, 0)$ ，原始图像中不存在这些像素点，那么放大图像如何处理这些问题呢？



在图像放大的正变换中，出现了很多的空格。因此，需要对放大后所多出的一些空格填入适当的像素值。一般采用最邻近插值和线性插值法。

85

• 拉伸变换



86

■ 基本几何变换的特征

➤ 坐标空间的变化

✓ 范围发生变化

✓ 大小发生变化

➤ 像素值的变化

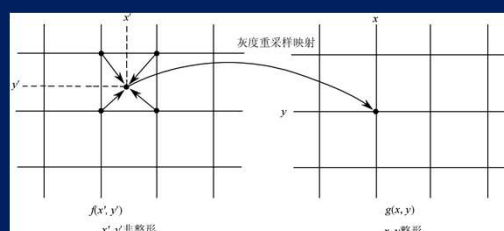
✓ 像素值不发生变化——位置改变

✓ 像素值发生变化——旋转、缩放、变形变换

87

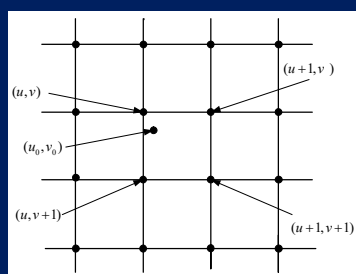
3.4.5 灰度重采样 (Gray Resampling)

几何运算还需要一个算法用于灰度级的重采样。如果一个输出像素映射到四个输入像素之间，则其灰度值由灰度插值算法决定，如图3.24所示。



88

3.4.5 灰度重采样 (Gray Resampling)



最近邻法

89

3.4.5 灰度重采样 (Gray Resampling)

最近邻法:

最近邻法是将 (u_0, v_0) 点最近的整数坐标 (u, v) 点的灰度值取为 (u_0, v_0) 点的灰度值。在 (u_0, v_0) 点各相邻像素间灰度变化较小时，这种方法是一种简单快捷的方法，但当 (u_0, v_0) 点相邻像素间灰度差很大时，这种灰度估值方法会产生较大的误差。

90

最近邻插值

特点:

- 简单快速
- 误差较大
- 视觉特性较差

马赛克效应

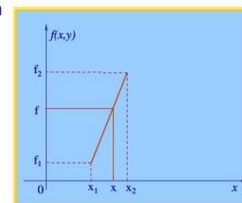


91

一维线性插值

— linear Interpolation

$$f = \frac{f_2 - f_1}{x_2 - x_1}(x - x_1) + f_1$$



92

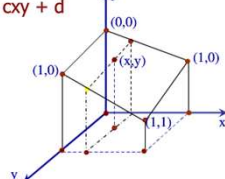
■ 双线性插值（一阶插值）

— Bilinear Interpolation

已知正方形的4个顶点，求正方形内部的点，有双线性方程： $f(x,y) = ax + by + cxy + d$

设4个顶点的坐标为：

$(0,0), (1,0), (0,1), (1,1)$



93

■ 双线性插值（一阶插值）

— Bilinear Interpolation

插值计算可用公式表示为：

$$f(x, 0) = f(0, 0) + x[f(1, 0) - f(0, 0)]$$

$$f(x, 1) = f(0, 1) + x[f(1, 1) - f(0, 1)]$$

$$f(x, y) = f(x, 0) + y[f(x, 1) - f(x, 0)]$$

$$f(x, y) = [f(1, 0) - f(0, 0)]x + [f(0, 1) - f(0, 0)]y + [f(1, 1) + f(0, 0) - f(0, 1) - f(1, 0)]xy + f(0, 0)$$

94

- 双线性插值的特点
- 1. 计算量大，但缩放后图像质量高，不会出现图像不连续的情况。
- 2. 具有低通滤波器的性质，使高频分量减弱，所以使图像的轮廓在一定程度受损

95

3.4.5 灰度重采样（Gray Resampling）

• 三次内插法

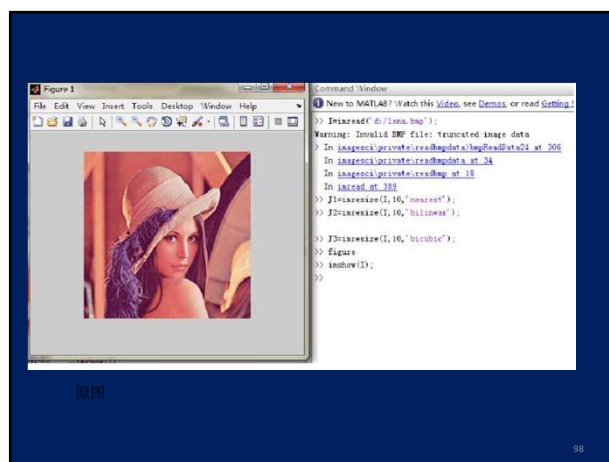
三次内插法不仅考虑 (u_0, v_0) 点的直接邻点对它的影响，还考虑到该点周围16个邻点的灰度值对它的影响。由连续信号采样定理可知，若对采样值用插值函数 $S(x) = \text{sin}(\pi x) / (\pi x)$ 插值，则可精确地恢复原函数，当然也就可精确得到采样点间任意点的值。此方法计算量很大，但精度高，能保持较好的图像边缘。

96

【例3.7】采用三种不同插值法进行图片的放大比较，其MATLAB程序如下：

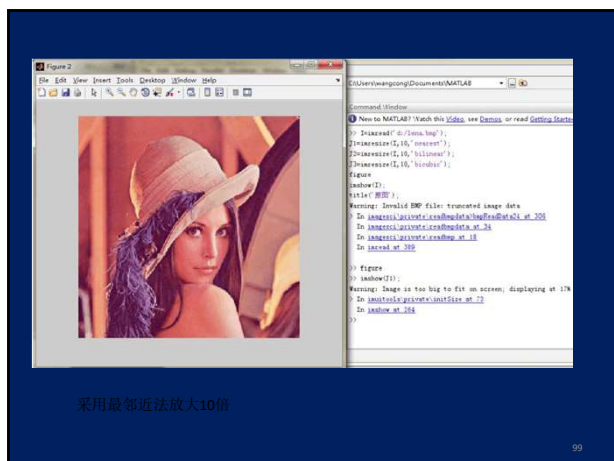
```
>> I=imread('d:\lena.bmp');
>> J1=imresize(I,10,'nearest'); %采用最近邻法进行放大10倍
>> J2=imresize(I,10,'bilinear'); %采用双线性插值法进行放大10倍
>> J3=imresize(I,10,'bicubic'); %采用三次内插法进行放大10倍
>> figure
>> imshow(I);
>> title('原图');
>> figure
>> imshow(J1);
>> title('最近邻法');
>> figure
>> imshow(J2);
>> title('双线性插值法');
>> figure
>> imshow(J3);
```

97



原图

98



采用最近邻法放大10倍

99



采用双线性插值法进行放大10倍

采用三次内插法进行放大10倍

100

小结 (Summary)

- 本讲主要介绍了图像的基本运算，包括点运算、代数运算、逻辑运算和几何运算，举了相应的Matlab实例，并对其相应的应用做了介绍。比如说代数运算可用于去除图像的噪声，进行混合图像的分离等等。其中的几何运算包括两个步骤，一个是空间变换，一个是重采样。然后简单介绍了下常用的三种灰度插值方法——最近邻法、双线性插值法和三次内插法，比较了优缺点。

101

讨论

讨论1：由于监控相机无法正对车辆安装导致拍摄的车牌有倾斜，如果要进行识别，如何对图像进行预处理？



讨论2：想要设计一个自动监控别墅有无陌生人闯入的机器视觉系统，如何设计硬件和软件？