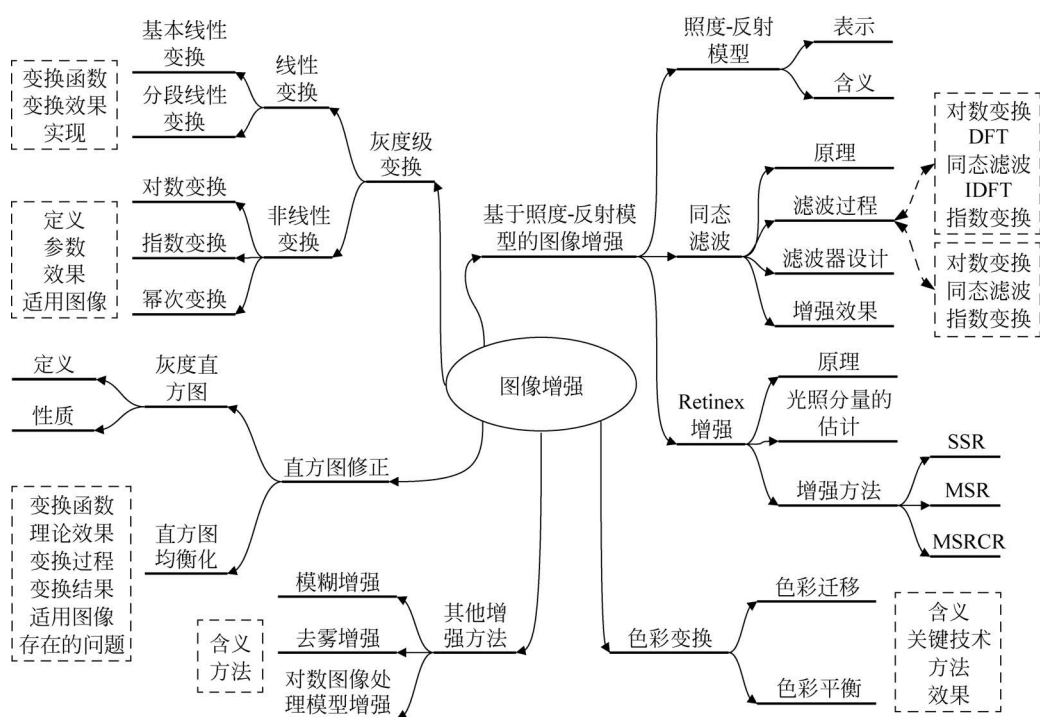


图像增强

本章思维导图



图像增强(Image Enhancement)主要是为了改善图像的质量以及增强感兴趣部分,改善图像的视觉效果或使图像变得更利于计算机处理。相关的图像增强技术有针对单个像素点的点运算,也有针对像素局部邻域的模板运算,根据模板运算的具体功能还可以分为图像平滑、图像锐化等。本章主要讲解图像增强技术中的灰度级变换、直方图修正法、基于照度-反射模型的图像增强和色彩变换等。

5.1 灰度级变换

灰度级变换就是借助于变换函数将输入的像素灰度值映射成一个新的输出值,通过改变像素的亮度值来增强图像,如式(5-1)所示。

$$g(x, y) = T[f(x, y)] \quad (5-1)$$

其中, $f(x, y)$ 是输入图像, $g(x, y)$ 是变换后的输出图像, T 是灰度变换函数。

变换函数 T 的不同将导致不同的输出,其实现的变换效果也不一样。因此,在实际应用中,可以通过灵活地设计变换函数 T 来实现各种处理。

根据变换函数的不同,灰度级变换可以分为线性灰度级变换和非线性灰度级变换。

5.1.1 线性灰度级变换

1. 基本线性灰度级变换

最基本的线性灰度级变换如图 5-1 所示。将输入图像 $f(x, y)$ 变换为 $g(x, y)$, 变换结果由倾角 α 所决定,如式(5-2)所示。

$$g(x, y) = f(x, y) \cdot \tan \alpha \quad (5-2)$$

可以看出,当 $\alpha = 45^\circ$, 图像无变化; 当 $\alpha < 45^\circ$, 变换后灰度取值范围压缩, 图像均匀变暗; 当 $\alpha > 45^\circ$, 变换后灰度取值范围拉伸变长, 图像均匀变亮。

当线性灰度级变换如图 5-2 所示, 将图像高低灰度值发生翻转, 实现对图像求反, 简单说来就是使黑变白, 使白变黑。

2. 分段线性灰度级变换

分段线性灰度级变换如图 5-3 所示, 它是将输入图像 $f(x, y)$ 的灰度级区间分成两段乃至多段, 分别作线性灰度级变换, 以获得增强图像 $g(x, y)$ 。典型的三段线性灰度级变换如式(5-3)所示。

$$g(x, y) = \begin{cases} \frac{c}{a} f(x, y), & 0 \leq f(x, y) < a \\ \frac{d-c}{b-a} [f(x, y) - a] + c, & a \leq f(x, y) < b \\ \frac{L-1-d}{L-1-b} [f(x, y) - b] + d, & b \leq f(x, y) < L-1 \end{cases} \quad (5-3)$$

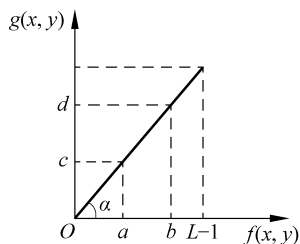


图 5-1 基本线性灰度级变换

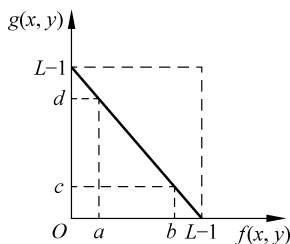


图 5-2 图像求反

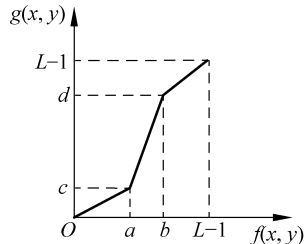


图 5-3 分段线性灰度级变换

其中,参数 a 、 b 、 c 、 d 为用于确定三段线段斜率的常数,取值可根据具体变换需求灵活设定。但也存在一些情况,用户仅对某个范围内的灰度感兴趣,只需对其进行线性拉伸,而将其余灰度级保持不变或变为某固定值。

MATLAB 也提供了函数 `imadjust` 用于实现图像的灰度级映射变换。

【例 5.1】 编写程序,对图像进行线性灰度级变换。

解:程序如下。

```
Image = im2double(rgb2gray(imread('img5_1.bmp'))); % 读取图像,灰度化并转换为 double 型
[h,w] = size(Image); % 获取图像尺寸
NewImage1 = tand(60) * Image; % 基本线性灰度级变换
NewImage2 = 1 - Image; % 图像求反
NewImage4 = Image; % 参数设置
a = 30/256; b = 100/256; c = 75/256; d = 200/256;
for x = 1:w
    for y = 1:h
        if Image(y,x) < a
            NewImage3(y,x) = Image(y,x) * c/a;
        elseif Image(y,x) < b
            NewImage3(y,x) = (Image(y,x) - a) * (d - c)/(b - a) + c; % 三段线性灰度级变换
        else
            NewImage3(y,x) = (Image(y,x) - b) * (1 - d)/(1 - b) + d;
        end
        if Image(y,x) > a && Image(y,x) < b
            NewImage4(y,x) = (Image(y,x) - a) * (d - c)/(b - a) + c; % [a,b]内线性变换,其余值不变
        end
    end
end
NewImage5 = imadjust(Image,[a b],[c d]); % [a,b]内线性变换,其余低端值变为 c,高端值变为 d
NewImage6 = imadjust(Image,[a b],[]); % 将[a,b]线性拉伸到[0,1],进行对比度拉伸
imshow(Image);title('原始图像');
figure,
subplot(2,3,1),imshow(NewImage1);title('基本线性灰度级变换');
subplot(2,3,2),imshow(NewImage2);title('图像取反');
subplot(2,3,3),imshow(NewImage3);title('分段线性灰度级变换');
subplot(2,3,4),imshow(NewImage4);title('高低端灰度保持不变');
subplot(2,3,5),imshow(NewImage5);title('高低端灰度赋固定值');
subplot(2,3,6),imshow(NewImage6);title('[a,b]线性拉伸到[0,1]');
```

线性灰度级变换的处理结果如图 5-4 所示。

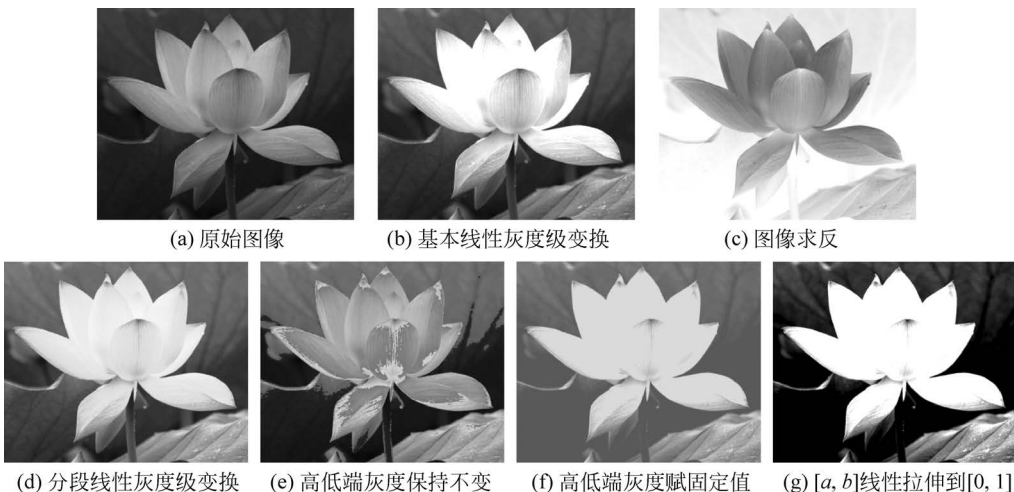


图 5-4 线性灰度级变换效果

5.1.2 非线性灰度级变换

当用某些非线性变换函数作为灰度变换的变换函数时,可实现图像灰度的非线性变换。对数变换、指数变换和幂变换是常见的非线性变换。

1. 对数变换

基于对数变换的非线性灰度级变换如式(5-4)所示。

$$g(x, y) = c \cdot \log[f(x, y) + 1] \quad (5-4)$$

式中, c 是尺度比例常数,其取值可以结合输入图像的范围来定。

对数变换函数示意图如图 5-5 所示。当希望对图像的低灰度区作较大拉伸、高灰度区压缩时,可采用这种变换,它能使图像的灰度分布与人的视觉特性相匹配。对数变换一般适用于处理过暗图像。

2. 指数变换

基于指数变换的非线性灰度变换如式(5-5)所示。

$$g(x, y) = b^{c \cdot [f(x, y) - a]} - 1 \quad (5-5)$$

式中, a 用于决定指数变换函数曲线的初始位置。当取 $f(x, y) = a$ 时, $g(x, y) = 0$, 曲线与 x 轴交叉。 b 是底数, c 用于决定指数变换曲线的陡度。

指数变换函数示意图如图 5-6 所示。当希望对图像的低灰度区压缩,高灰度区作较大拉伸时,可采用这种变换。指数变换一般适用于处理过亮图像。

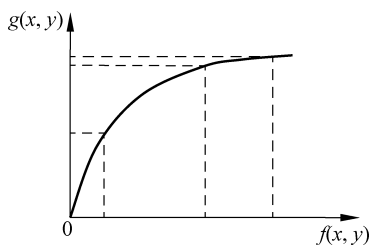


图 5-5 对数变换函数

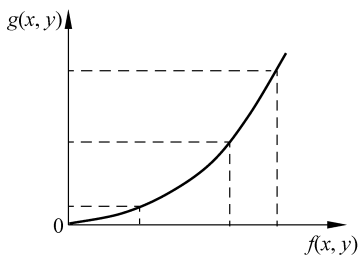


图 5-6 指数变换函数

3. 幂次变换

基于幂次变换的非线性灰度变换如式(5-6)所示。

$$g(x, y) = c \cdot [f(x, y)]^\gamma \quad (5-6)$$

式中, c 和 γ 为正常数。当 c 取 1, γ 取不同值时,可以得到一族变换曲线,如图 5-7 所示。

与对数变换的情况类似,幂次变换可以将部分灰度区域映射到更宽的区域中,从而增强图像的对比度。当 $\gamma = 1$ 时,幂次变换转变为线性正比变换;当 $0 < \gamma < 1$ 时,幂次变换可以扩展原图中低灰度级、压缩高灰度级,从而使图像变亮,增强原图中暗区的细节;当 $\gamma > 1$ 时,幂次变换可以扩展原图中高灰度级,压缩低灰度级,从而使图像变暗,增强原图中亮区的细节。

幂次变换常用于图像获取、打印和显示的各种装置设备的伽马校正中,这些装置设备的光电转换特性都是非线性的,是根据幂次规律产生响应的。幂次变换的指数值就是伽马值,因此幂次变换也称为伽马变换。

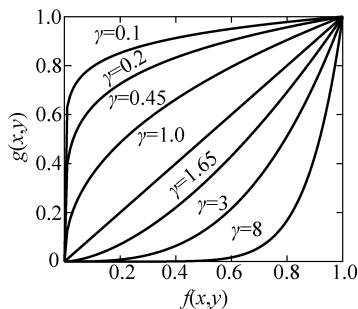


图 5-7 幂次变换函数

【例 5.2】 编写程序对图像进行非线性灰度变换。

解：程序如下。

```
Image = rgb2gray(imread('img5_2.bmp'));
Image = double(Image);
NewImage1 = log(Image + 1); % 对数函数非线性灰度级变换
NewImage2 = exp(0.325 * (Image - 225)/30) + 1; % 指数函数非线性灰度级变换
a = 0.5; c = 1.1;
NewImage3 = [(Image/255).^a] * 255 * c; % 幂次函数非线性灰度级变换
imshow(Image,[]);title('原图');
figure;imshow(NewImage1,[]);title('对数变换');
figure;imshow(NewImage2,[]);title('指数变换');
figure;imshow(NewImage3,[]);title('幂次变换');
```

程序运行结果如图 5-8 所示。可以看出,原图经过对数变换处理后(如图 5-8(b)),整体亮度增强;经过指数变换处理后(如图 5-8(c)),整体亮度变暗;图 5-8(d)为幂次变换处理结果,幂为 0.5,图像变亮,但变亮程度低于对数变换。



图 5-8 非线性灰度变换处理效果

5.2 直方图修正法

在数字图像处理中,灰度直方图是最简单和常用的工具。本节介绍灰度直方图的概念及直方图修正技术。

5.2.1 灰度直方图

1. 灰度直方图的定义

灰度直方图是灰度级的函数,表示的是数字图像中每一灰度级与其出现频数(呈现该灰度的像素数目)间的统计关系。通常,用横坐标表示灰度级,纵坐标表示频数或相对频数(呈现该灰度级的像素出现的概率)。灰度直方图的定义如式(5-7)所示。

$$p(r_k) = \frac{n_k}{N} \quad (5-7)$$

式中, N 为一幅数字图像的总像素数, n_k 是第 k 级灰度的像素数, r_k 表示第 k 个灰度级, $p(r_k)$ 为该灰度级 r_k 出现的相对频数。

给定一幅 6×6 的图像,如图 5-9(a)所示,共有 $0 \sim 7$ 八个灰度级。灰度分布统计如表 5-1 所示,则可以绘制并显示图像的灰度直方图,如图 5-9(b)所示。

表 5-1 灰度分布统计

r_k	0	1	2	3	4	5	6	7
n_k	6	9	6	5	4	3	2	1
$p(r_k)$	6/36	9/36	6/36	5/36	4/36	3/36	2/36	1/36

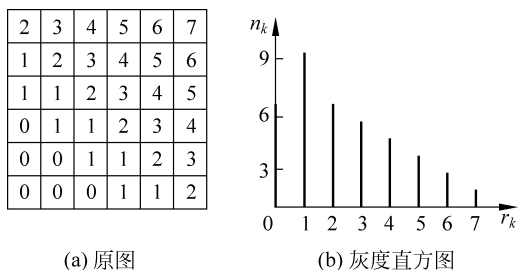


图 5-9 数字图像及其直方图

【例 5.3】 编写程序,统计图像的灰度直方图分布。

解: 统计图像的灰度直方图可以通过扫描图像并统计各个灰度出现的次数,进而计算频数并绘制直方图的方法。

程序如下。

```
Image = rgb2gray(imread('img5_3.bmp'));
histgram = zeros(256);
[h w] = size(Image);
for x = 1:w
    for y = 1:h % 循环扫描
        histgram(Image(y,x) + 1) = histgram(Image(y,x) + 1) + 1; % 统计并累加
    end
end
imshow(Image);title('couple 灰度图像');
figure;stem(histgram(),' ');
axis tight;
```

也可以直接调用 MATLAB 提供的函数 `imhist`,程序如下。

```
figure;imhist(Image);
axis tight;
```

程序运行结果如图 5-10 所示。

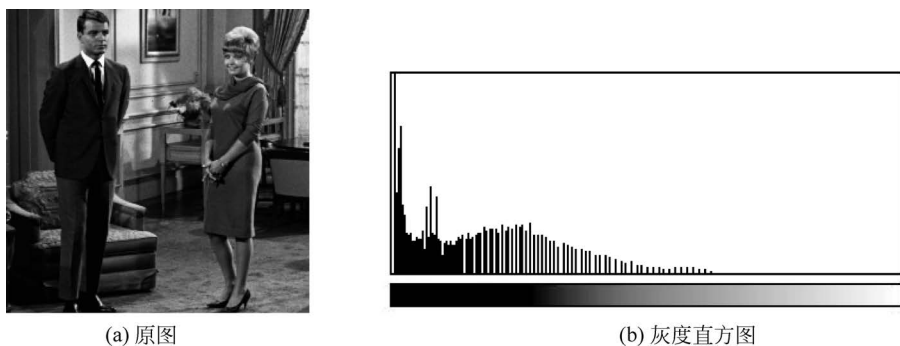


图 5-10 原图及其直方图

2. 灰度直方图的性质

一幅图像的灰度直方图通常具有如下性质。

(1) 直方图不具有空间特性。直方图描述了每个灰度级具有的像素的个数,但不能反映图像像素空间位置信息,即不能为这些像素在图像中的位置提供任何线索。

(2) 直方图反映图像的大致描述,如图像灰度范围、灰度级分布、整幅图像平均亮度等。图 5-11 为两幅图像的直方图,可以从中判断出图像的相关特性。图 5-11(a)中,大部分像素值

集中在低灰度级区域,图像偏暗;图 5-11(b)中的图像则相反,大部分像素的灰度集中在高灰度级区域,图像偏亮;两幅图像都存在动态范围不足的现象。

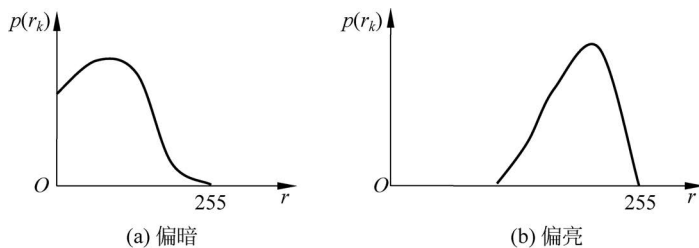


图 5-11 灰度动态范围不足的图像灰度直方图

(3) 一幅图像唯一对应相应的直方图,而不同的图像可以具有相同的直方图。因直方图只是统计图像中灰度出现的次数,与各个灰度出现的位置无关,因此,不同的图像可能具有相同的直方图。图 5-12(a)为四幅大小相同、空间灰度分布不同的二值图像,图 5-12(b)为它们具有的相同的直方图。

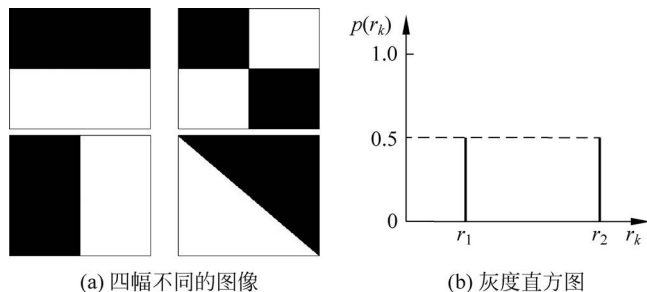


图 5-12 不同图像具有相同直方图分布特性

(4) 若一幅图像可分为多个子区,则多个子区直方图之和等于对应的全图直方图。

5.2.2 直方图修正法理论

直方图修正法的基本原理就是通过构造灰度级变换函数改造原图像的直方图,使变换后的图像的直方图达到一定的要求。

设变量 r 代表要增强的图像中像素的灰度级,变量 s 代表增强后新图像中的灰度级。为了研究方便,将 r 、 s 归一化,得

$$0 \leq r \leq 1, \quad 0 \leq s \leq 1 \quad (5-8)$$

则直方图修正法变换函数 $T(\cdot)$ 的定义如式(5-9)所示。

$$s = T(r) \quad (5-9)$$

其中,每一像素灰度值 r 对应产生一个 s 值。

对图像的直方图修正变换过程中,要满足:

- (1) $T(r)$ 在 $0 \leq r \leq 1$ 区域内单值单调增加,以保证灰度级从黑到白的次序不变;
- (2) $T(r)$ 在 $0 \leq r \leq 1$ 区域内满足 $0 \leq s \leq 1$,以保证变换后的像素灰度级仍在允许的灰度级范围内。

直方图修正法的核心就是寻找满足这两个条件的变换函数 $T(r)$ 。

5.2.3 直方图均衡化

直方图均衡化是采用灰度级 r 的累积分布函数作为变换函数的直方图修正法。

假设用 $p_r(r)$ 表示原图像灰度级 r 的灰度级概率密度函数,直方图均衡化变换函数为

$$s = T(r) = \int_0^r p_r(\omega) d\omega \quad (5-10)$$

$T(r)$ 是 r 的累积分布函数,随着 r 增大, s 值单调增加,最大为 1,满足直方图修正法的两个条件。

根据概率论知识,用 $p_r(r)$ 和 $p_s(s)$ 分别表示 r 和 s 的灰度级概率密度函数,得

$$p_s(s) = p_r(r) \cdot \frac{dr}{ds} = p_r(r) \cdot \frac{1}{p_r(r)} = 1 \quad (5-11)$$

即利用 r 的累积分布函数作为变换函数可产生一幅灰度级分布具有均匀概率密度的图像。

给出一幅数字图像,共有 L 个灰度等级,总像素个数为 N ,其中,第 j 级灰度 r_j 对应的像素数为 n_j 。根据式(5-7)进行灰度直方图统计,则图像进行直方图均衡化处理的变换函数 $T(r)$ 为

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{N} \quad (5-12)$$

对一幅数字图像进行直方图均衡化处理的算法步骤如下。

- (1) 由式(5-7)统计原始图像直方图。
- (2) 由式(5-12)计算新的灰度级。
- (3) 修正 s_k 为合理的灰度级。
- (4) 计算新的直方图。
- (5) 用处理后的新灰度代替处理前的灰度,生成新图像。

【例 5.4】 假定一幅大小为 64×64 、灰度级为 8 级的图像,其灰度级分布如表 5-2 所示,对其进行直方图均衡化处理。

表 5-2 例 5.4 中图像的灰度级分布

灰度级 r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
像素数 n_k	790	1023	850	656	329	245	122	81
$p_r(r_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

解: 由原图的灰度分布统计可以看出,该图像中绝大部分像素灰度值集中在低灰度区,图像整体偏暗。

- (1) 计算新的灰度级:

$$s_0 = T(r_0) = \sum_{j=0}^0 p_r(r_j) = p_r(r_0) = 0.19$$

$$s_1 = T(r_1) = \sum_{j=0}^1 p_r(r_j) = p_r(r_0) + p_r(r_1) = 0.19 + 0.25 = 0.44$$

以此类推,可得到

$$s_2 = 0.19 + 0.25 + 0.21 = 0.65$$

$$s_3 = 0.19 + 0.25 + 0.21 + 0.16 = 0.81, \quad s_4 = 0.89$$

$$s_5 = 0.95, \quad s_6 = 0.98, \quad s_7 = 1$$

- (2) 修正 s_k 为合理的灰度级 s'_k :

$$s_0 = 0.19 \approx \frac{1}{7}, \quad s_1 = 0.44 \approx \frac{3}{7}, \quad s_2 = 0.65 \approx \frac{5}{7}, \quad s_3 = 0.81 \approx \frac{6}{7}$$

$$s_4 = 0.89 \approx \frac{6}{7}, \quad s_5 = 0.95 \approx 1, \quad s_6 = 0.98 \approx 1, \quad s_7 = 1$$

则新图像对应只有 5 个不同灰度级别,为 1/7、3/7、5/7、6/7、1,即

$$s'_0 = \frac{1}{7}, \quad s'_1 = \frac{3}{7}, \quad s'_2 = \frac{5}{7}, \quad s'_3 = \frac{6}{7}, \quad s'_4 = 1$$

(3) 计算新的直方图:

$$p_s(s'_0) = p_r(r_0) = 0.19$$

$$p_s(s'_1) = p_r(r_1) = 0.25$$

$$p_s(s'_2) = p_r(r_2) = 0.21$$

$$p_s(s'_3) = p_r(r_3) + p_r(r_4) = 0.16 + 0.08 = 0.24$$

$$p_s(s'_4) = p_r(r_5) + p_r(r_6) + p_r(r_7) = 0.06 + 0.03 + 0.02 = 0.11$$

(4) 生成新图像:按照表 5-3 中变换前后的灰度对应关系改变像素的灰度,即可生成新的图像。

表 5-3 直方图均衡化变换前后灰度级对应关系

变换前灰度级	0	1/7	2/7	3/7	4/7	5/7	6/7	1
变换后灰度级	1/7	3/7	5/7	6/7	6/7	1	1	1

原始图像直方图和直方图均衡化处理后的图像直方图显示结果如图 5-13 所示。可以看出,图 5-13(b)中对应的变换后的新直方图比图 5-13(a)中的原图像的直方图要平坦多了。理想情况下,经过直方图均衡化处理后的图像直方图应是十分均匀平坦的,但实际情况并非如此,和理论分析有差异,这是由于图像在直方图均衡化处理过程中,灰度级作“近似简并”引起的结果。

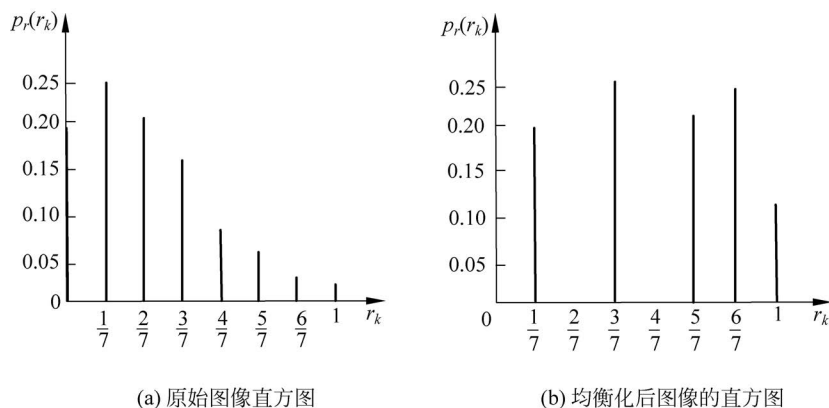


图 5-13 直方图均衡化处理前后的直方图分布对比

【例 5.5】 编写程序,对图像进行直方图均衡化。

解: 程序如下。

```
Image = rgb2gray(imread('img5_3.bmp'));
histgram = imhist(Image); % 统计图像直方图
[h w] = size(Image);
NewImage = zeros(h,w);
s = zeros(256); s(1) = histgram(1);
for t = 2:256
```

```

        s(t) = s(t-1) + histogram(t);                % 计算新的灰度值
    end
    for x = 1:w
        for y = 1:h
            NewImage(y,x) = s(Image(y,x) + 1)/(w * h);    % 生成新图像
        end
    end
    imshow(Image);title('原始灰度图像');
    figure;imhist(Image);title('原始灰度图像的直方图');
    axis tight;
    figure;imshow(NewImage);title('直方图均衡化处理后图像');
    figure;imhist(NewImage);title('直方图均衡化处理后图像的直方图');
    axis tight;

```

也可以直接调用 MATLAB 提供的函数 `histeq`, 程序如下。

```
NewImage = histeq(Image,256)
```

程序运行结果如图 5-14 所示。

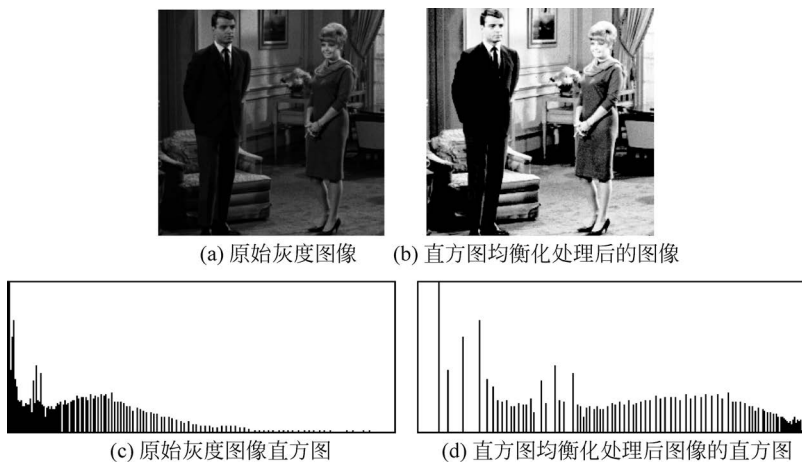


图 5-14 直方图均衡化处理前后的图像及直方图

前面介绍的灰度级变换和直方图修正都是以灰度图像为处理目标的,假如是一幅彩色图像比较暗,能不能采用这两种方法进行增强呢? 请扫二维码,查看讲解。

5.3 基于照度-反射模型的图像增强

一般情况下,自然景物图像 $f(x,y)$ 可以表示为光源照度场(照明函数) $i(x,y)$ 和场景中物体反射光的反射场(反射函数) $r(x,y)$ 的乘积,如式(5-13)所示。

$$f(x,y) = i(x,y) \cdot r(x,y) \quad (5-13)$$

其中, $0 < i(x,y) < \infty$, $0 < r(x,y) < 1$ 。一般把式(5-13)称为图像的照度-反射模型。

近似认为,照明函数 $i(x,y)$ 描述景物的照明,其性质取决于照射源,与景物无关。反射函数 $r(x,y)$ 描述景物内容,其性质取决于成像物体的特性,而与照明无关。由于照明亮度一般是缓慢变化的,所以认为照明函数的频谱集中在低频段。由于反射函数随图像细节不同在空间快速变化,所以认为反射函数的频谱集中在高频段。这样,就可根据式(5-13)将图像理解为高频分量与低频分量的乘积的结果。

基于照度-反射模型的处理算法,通常会借助于对数变换,将式(5-13)中两个相乘分量变成两个相加分量。这样不仅能够简化计算,而且对数变换接近人眼亮度感知能力,能够增强



微课视频

图像的视觉效果。

下面主要介绍两种基于照度-反射模型的处理算法:同态滤波和 Retinex 增强。

5.3.1 同态滤波

若物体受到照度明暗不匀,则图像上对应照度暗的部分的细节就较难辨别。同态滤波的主要目的就是消除不均匀照度的影响,增强图像细节。

同态滤波的基本原理是根据图像的照度-反射模型,对原始图像 $f(x, y)$ 中的反射分量 $r(x, y)$ 进行扩展,对光照分量 $i(x, y)$ 进行压缩,以获得所要求的增强图像。

同态滤波的具体算法步骤如下。

(1) 对图像函数 $f(x, y)$ 取对数,即进行对数变换处理

$$\begin{aligned} z(x, y) &= \ln[f(x, y)] = \ln[i(x, y) \cdot r(x, y)] \\ &= \ln[i(x, y)] + \ln[r(x, y)] \end{aligned} \quad (5-14)$$

(2) 进行傅里叶变换处理

$$\begin{aligned} Z(u, v) &= \text{DFT}\{z(x, y)\} = \text{DFT}\{\ln[i(x, y)]\} + \text{DFT}\{\ln[r(x, y)]\} \\ &= I(u, v) + R(u, v) \end{aligned} \quad (5-15)$$

(3) 进行同态滤波处理

$$\begin{aligned} S(u, v) &= \text{Homo}(u, v) \cdot Z(u, v) \\ &= \text{Homo}(u, v) \cdot I(u, v) + \text{Homo}(u, v) \cdot R(u, v) \end{aligned} \quad (5-16)$$

如前所述,图像的对数傅里叶变换的高频分量主要对应反射分量,低频分量主要对应照射分量,因此,需要设计合适的同态滤波函数 $\text{Homo}(u, v)$,对高频、低频成分产生不同的影响,目的是为了压低照明分量、扩大反射分量。

可以看出,同态滤波函数的功能类似于高通滤波函数,因此,可在传统高通滤波函数的基础上加以改动,逼近同态滤波函数。假设高通滤波转移函数为 $\text{High}(u, v)$,则由 $\text{High}(u, v)$ 到 $\text{Homo}(u, v)$ 的映射关系如式(5-17)所示。

$$\text{Homo}(u, v) = (\gamma_H - \gamma_L) \cdot \text{High}(u, v) + \gamma_L \quad (5-17)$$

式中, γ_H 和 γ_L 分别表示高频分量频率场和低频分量频率场滤波特性,且当取值 $\gamma_H > 1, 0 < \gamma_L < 1$ 时,照射分量受到抑制,反射分量得到增强,从而突出图像的轮廓细节。同态滤波函数 $\text{Homo}(u, v)$ 的特性曲线如图 5-15 所示。

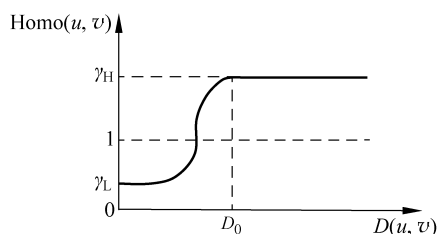


图 5-15 同态滤波函数特性曲线

(4) 求傅里叶反变换

$$\begin{aligned} s(x, y) &= \text{DFT}^{-1}\{S(u, v)\} \\ &= \text{DFT}^{-1}\{\text{Homo}(u, v) \cdot I(u, v)\} + \text{DFT}^{-1}\{\text{Homo}(u, v) \cdot R(u, v)\} \\ &= i'(x, y) + r'(x, y) \end{aligned} \quad (5-18)$$

(5) 求指数变换,得到经同态滤波处理的图像

$$\begin{aligned} g(x, y) &= e^{\{s(x, y)\}} = e^{\{i'(x, y) + r'(x, y)\}} \\ &= i_0(x, y) \cdot r_0(x, y) \end{aligned} \quad (5-19)$$

式中, $i_0(x, y)$ 是处理后的照射分量, $r_0(x, y)$ 是处理后的反射分量。

需要指出的是,在傅里叶平面上用同态滤波器来增强高频分量以突出轮廓细节的同时,也

平滑了低频分量,使得图像中灰度变化平缓区域出现模糊。因此,通常会增加一个后滤波处理补偿低频分量,以使得图像得到很大的改善。

【例 5.6】 编写程序,对图像进行同态滤波增强。

解: 程序如下。

```
Image = rgb2gray(imread('img5_6.jpg'));
[M,N] = size(Image);
Hmin = 0.5; Hmax = 2.0; % 可根据需要效果调整参数
c = 1.1; d0 = 1800;
Image1 = log(double(Image) + 1); % 取对数
FImage = fft2(Image1); % 傅里叶变换
n1 = floor(M/2); n2 = floor(N/2);
for i = 1:M
    for j = 1:N
        D(i,j) = ((i - n1)^2 + (j - n2)^2); % 高斯同态滤波
        H(i,j) = (Hmax - Hmin) * (1 - exp(c * (-D(i,j)./(d0^2)))) + Hmin;
    end
end
Image2 = ifft2(H. * FImage); % 傅里叶反变换
result = real(exp(Image2));
imshow(I),title('原图');
figure,imshow(result,[]);title('同态滤波增强后');
```

程序运行结果如图 5-16 所示。可以看出,原始的光照不均匀图像[见图 5-16(a)]经过同态滤波增强处理后,质量有了很大改善。

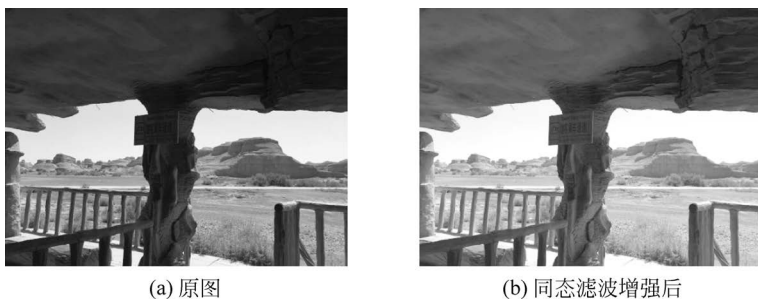


图 5-16 同态滤波增强

5.3.2 Retinex 增强

Retinex 源于 Retina(视网膜)和 Cortex(大脑皮层)合成词的缩写,故 Retinex 理论又被称为“视网膜大脑皮层理论”。Retinex 理论的基本原理模型是以人类视觉系统为出发点发展而来的一种基于颜色恒常性的色彩理论,该理论认为:①人眼对物体颜色的感知与物体表面的反射性质有着密切关系,即由物体对红、绿、蓝三色光线的反射能力来决定,反射率低的物体看上去较暗,反射率高的物体看上去较亮;②人眼对物体色彩的感知具有一致性,不受光照变化的影响。

基于 Retinex 理论的增强方法的基本原理就是根据图像的照度-反射模型,通过从原始图像中估计光照分量,然后设法去除(或降低)光照分量,获得物体的反射性质,从而获得物体的本来面貌。

根据采用的不同的估计光照分量的方法,产生了各种 Retinex 算法。这里主要介绍中心环绕 Retinex 方法。在中心环绕 Retinex 方法中,估计光照分量的计算如式(5-20)所示,其物理

意义是通过计算被处理像素与其周围区域加权平均值的比值消除照度变化的影响。

$$i'_C(x, y) = F(x, y) * f_C(x, y) \quad (5-20)$$

其中, $C \in \{R, G, B\}$, $f_C(x, y)$ 是图像 $f(x, y)$ 的第 C 颜色通道的亮度分量。 $i'_C(x, y)$ 是第 C 颜色通道的光照分量估计值。 $F(x, y)$ 是中心环绕函数, 一般采用高斯函数形式的环绕函数, 即

$$F(x, y) = K \cdot e^{-\frac{x^2+y^2}{\sigma^2}} \quad (5-21)$$

其中, σ 为标准差, 表示高斯环绕函数的尺度常数, 决定了卷积核的作用范围。 K 为归一化因子, 使得

$$\iint F(x, y) dx dy = 1 \quad (5-22)$$

中心环绕 Retinex 方法主要分为单尺度 Retinex(Single-Scale Retinex, SSR)增强方法、多尺度 Retinex(Multi-Scale Retinex, MSR)增强方法和带色彩恢复的多尺度 Retinex(Multi-Scale Retinex with Color Restoration, MSRCR)增强方法。

1. 单尺度 Retinex 增强方法

单尺度 Retinex 增强方法的具体步骤如下:

- (1) 根据式(5-20)、式(5-21)计算第 C 颜色通道的光照分量估计值 $i'_C(x, y)$;
- (2) 对 $f_C(x, y)$ 取对数, 即进行对数变换处理, 有

$$\begin{aligned} \ln[f_C(x, y)] &= \ln[i_C(x, y) \cdot r_C(x, y)] \\ &= \ln[i_C(x, y)] + \ln[r_C(x, y)] \end{aligned} \quad (5-23)$$

(3) 在对数域中, 用 $f_C(x, y)$ 减去光照分量估计 $i'_C(x, y)$, 计算反射分量, 即获得图像的高频分量, 有

$$\begin{aligned} R_C(x, y) &= \ln[r'_C(x, y)] = \ln[f_C(x, y)] - \ln[i'_C(x, y)] \\ &= \ln[f_C(x, y)] - \ln[F(x, y) * f_C(x, y)] \end{aligned} \quad (5-24)$$

式中, $R_C(x, y)$ 是第 C 颜色通道的单尺度 Retinex 增强输出图像。

2. 多尺度 Retinex 增强方法

根据式(5-21), 高斯环绕函数 $F(x, y)$ 的尺度 σ 的大小会直接影响图像增强结果。尺度 σ 越小, 越能够较好地动态范围压缩, 但全局照度损失, 图像呈现“白化”; 尺度 σ 越大, 越能够较好地保证图像的色感一致性, 但局部细节模糊, 强边缘处有明显“光晕”。单尺度 Retinex 增强方法很难在动态范围压缩和色感一致性上寻找到平衡点。因此, 对单尺度改进后, 就产生了多尺度 Retinex 增强方法。

多尺度 Retinex 增强实质上是多个不同单尺度 Retinex 增强的加权平均。

多尺度 Retinex 增强方法的具体步骤如下。

- (1) 设置不同尺度 $\sigma_n, n=1, 2, \dots, N$ 。其中, N 为设置的不同尺度个数。
- (2) 根据式(5-21), 计算不同尺度的中心环绕函数 $F_n(x, y)$ 。
- (3) 根据式(5-25), 求图像的不同尺度的 Retinex 增强输出。

$$R_{n_C}(x, y) = \{\ln[f_C(x, y)] - \ln[F_n(x, y) * f_C(x, y)]\} \quad (5-25)$$

式中, $R_{n_C}(x, y)$ 是第 C 颜色通道第 n 个尺度的 Retinex 增强输出。

- (4) 对多个不同尺度的 Retinex 增强输出结果进行加权平均。

$$R_{MSR_C}(x, y) = \sum_{n=1}^N w_n R_{n_C}(x, y) \quad (5-26)$$

式中, $R_{\text{MSRC}}(x, y)$ 是第 C 颜色通道的多尺度 Retinex 增强输出。 w_n 是给不同尺度 σ_n 分配的权重因子, 满足: $\sum w_n = 1$ 。

3. 带色彩恢复的多尺度 Retinex 增强方法

在上述算法的彩色图像增强过程中, 由于是对 RGB 每一个通道直接进行 N 次不同尺度滤波, 导致三通道增幅不一, 恢复的 RGB 比值与原图比值不太一样, 引起颜色失真的问题。因此, 产生了带色彩恢复的多尺度 Retinex 增强方法。

带色彩恢复的多尺度 Retinex 增强方法的表达式如下:

$$R_{\text{MSRCR}_C}(x, y) = R_{\text{MSRC}}(x, y) \cdot \gamma_C(x, y) \quad (5-27)$$

式中, $\gamma_C(x, y)$ 是第 C 颜色通道的色彩恢复系数, 其定义如式(5-28)所示。

$$\gamma_C(x, y) = \eta \cdot \ln \left[\beta \cdot \frac{f_C(x, y)}{\sum_{C \in \{R, G, B\}} f_C(x, y)} \right] \quad (5-28)$$

式中, η 为增益常数, β 为非线性强度的控制因子。

【例 5.7】 编写程序, 对图像进行 Retinex 增强方法。

解: 程序如下。

```
Image = double(imread('img5_6.jpg')); % 打开图像并转换为 double 数据
[height,width,c] = size(Image); % 获得图像尺寸
RI = Image(:, :, 1); GI = Image(:, :, 2); BI = Image(:, :, 3);
alpha = 0.4; beta = 125;
CR = alpha * (log(beta * (RI + 1)) - log(RI + GI + BI + 1)); % R 通道色彩恢复系数
CG = alpha * (log(beta * (GI + 1)) - log(RI + GI + BI + 1)); % G 通道色彩恢复系数
CB = alpha * (log(beta * (BI + 1)) - log(RI + GI + BI + 1)); % B 通道色彩恢复系数
sigma = [100 15 80 250]; filtersize = [height,width]; % 设置高斯滤波器参数
Rhigh1 = zeros(height,width); Rhigh2 = zeros(height,width);
Ghigh1 = zeros(height,width); Ghigh2 = zeros(height,width); % 初始化
Bhigh1 = zeros(height,width); Bhigh2 = zeros(height,width);
for i = 1:4
    gaussfilter = fspecial('gaussian', filtersize, sigma(i)); % 构造高斯低通滤波器
    Rlow = imfilter(RI, gaussfilter, 'replicate', 'conv'); % 高斯卷积后, 获得 R 通道低频分量
    Glow = imfilter(GI, gaussfilter, 'replicate', 'conv'); % 获得 G 通道低频分量
    Blow = imfilter(BI, gaussfilter, 'replicate', 'conv'); % 获得 B 通道低频分量
    if i == 1
        Rhigh = log(RI ./ Rlow + 1); % 获得 R 通道高频分量
        Ghigh = log(GI ./ Glow + 1); % 获得 G 通道高频分量
        Bhigh = log(BI ./ Blow + 1); % 获得 B 通道高频分量
        SSRI = cat(3, Rhigh, Ghigh, Bhigh); % % SSR 增强
    else
        Rhigh1 = 1/3 * (log(RI ./ Rlow + 1) + Rhigh1); % 获得 R 通道高频分量
        Ghigh1 = 1/3 * (log(GI ./ Glow + 1) + Ghigh1); % 获得 G 通道高频分量
        Bhigh1 = 1/3 * (log(BI ./ Blow + 1) + Bhigh1); % 获得 B 通道高频分量
        Rhigh2 = 1/3 * (CR * log(RI ./ Rlow + 1) + Rhigh2); % 获得 R 通道高频分量
        Ghigh2 = 1/3 * (CG * log(GI ./ Glow + 1) + Ghigh2); % 获得 G 通道高频分量
        Bhigh2 = 1/3 * (CB * log(BI ./ Blow + 1) + Bhigh2); % 获得 B 通道高频分量
    end
end
MSRI = cat(3, Rhigh1, Ghigh1, Bhigh1); % MSR 增强图像
MSRCRI = cat(3, Rhigh2, Ghigh2, Bhigh2); % MSRCR 增强图像
subplot(2,2,1); imshow(uint8(Image)); title('原图');
subplot(2,2,2); imshow(SSRI); title('SSR 增强后图像');
subplot(2,2,3); imshow(MSRI); title('MSR 增强后图像');
subplot(2,2,4); imshow(MSRCRI); title('MSRCR 增强后图像');
```

程序运行结果如图 5-17 所示。

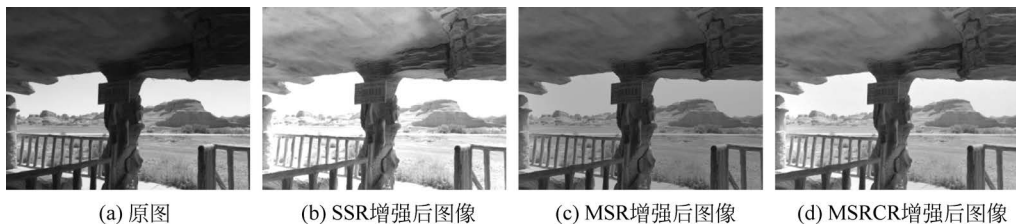


图 5-17 Retinex 增强效果

启发：

Retinex 理论诠释了同样物体在有差异的光源或光线底下颜色恒定的机理,即认为人眼能够以某种方式过滤掉光照等影响,而只保留反映物体本质特征的信息,如反射率,从而确定颜色。正如“透过现象看本质”,需要突破认知上的束缚,识别出信息背后的本质,做出更有效的决策。

5.4 色彩变换

色彩是表征图像信息的重要组成部分。色彩信息使得图像能够提供更丰富的内容、更清晰的细节,从而具有更好的视觉表达力。

5.4.1 色彩迁移

图像色彩迁移是一种改变图像颜色基调的方法,实现将一幅图像的色彩信息传递到另一幅图像中,使两幅图像具有相类似的色彩特征。

色彩空间的选取对图像的色彩迁移算法是否有效有很大的影响。Smith 等于 1975 年基于人体视网膜锥状细胞对光的波长相当敏感这一现象提出了 LMS 颜色空间,其中 L、M、S 三个通道分别表示长、中、短激发光谱。由于 LMS 颜色空间三个通道间有较大的相关性,给图像处理过程带来一定的困难。针对这种情况,Ruderman 等于 1998 年在 LMS 颜色空间基础上提出了一种不相关的、近似正交的均匀色彩空间—— $l\alpha\beta$ 颜色空间,其中, l 表示非彩色的亮度通道, α 表示彩色的黄-蓝相关通道, β 表示彩色的红-绿相关通道, $0 \leq l \leq 100$, $-128 \leq \alpha, \beta \leq 127$ 。 $l\alpha\beta$ 颜色空间不仅假设人类视觉系统理想地适应自然基色的处理,建立把光线波长转换为亮度和色相的一套描述色彩数据,而且最大限度地降低了各通道间的相关性,使得每个通道的处理和运算都可以独立进行,不会影响到其他通道的信息,避免了通道交叉的问题,它为图像色彩迁移的发展奠定了基础。

Reinhard 等于 2001 年通过对 $l\alpha\beta$ 颜色空间各分量进行统计分析,首次提出了彩色图像的色彩迁移算法,算法思想是在 $l\alpha\beta$ 颜色空间,通过对目标图像和源图像的色彩特征信息分别进行统计分析,进而得到两者间的变换关系,可使得目标图像和源图像最终具有相近的均值和标准差,这样就将源图像的色彩特征传递给了目标图像,使之看上去更加符合人类的视觉审美要求。

1. $l\alpha\beta$ 与 RGB 相互转换

RGB 转换为 $l\alpha\beta$ 的过程如下。

由 RGB 空间先转换到 LMS 空间,再转换到 $l\alpha\beta$ 空间,有

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5-29)$$

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \log_{10} L \\ \log_{10} M \\ \log_{10} S \end{bmatrix} \quad (5-30)$$

$l\alpha\beta$ 转换为 RGB 的过程如下。

由 $l\alpha\beta$ 空间先转换到 LMS 空间,再转换到 RGB 空间,有

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} \quad (5-31)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} 10^L \\ 10^M \\ 10^S \end{bmatrix} \quad (5-32)$$

式(5-30)中选择 10 为底的对数运算,因此式(5-32)中进行 10 为底的指数运算,其他合理底数也可以。

2. Reinhard 色彩迁移

Reinhard 色彩迁移算法的具体步骤如下。

(1) 将源图像和目标图像由 RGB 变换到 $l\alpha\beta$ 颜色空间,并分别计算 l 、 α 、 β 三分量的均值和方差。

(2) 利用下式匹配色彩统计量,使其从源图像传递到目标图像。

$$\begin{cases} l_{\text{res}} = (l_{\text{in}} - \mu_{\text{in}}^l) \times \frac{\sigma_{\text{ref}}^l}{\sigma_{\text{in}}^l} + \mu_{\text{ref}}^l \\ \alpha_{\text{res}} = (\alpha_{\text{in}} - \mu_{\text{in}}^\alpha) \times \frac{\sigma_{\text{ref}}^\alpha}{\sigma_{\text{in}}^\alpha} + \mu_{\text{ref}}^\alpha \\ \beta_{\text{res}} = (\beta_{\text{in}} - \mu_{\text{in}}^\beta) \times \frac{\sigma_{\text{ref}}^\beta}{\sigma_{\text{in}}^\beta} + \mu_{\text{ref}}^\beta \end{cases} \quad (5-33)$$

其中, l_{res} 、 α_{res} 、 β_{res} 为色彩迁移结果图像的 l 、 α 、 β 三分量值; l_{in} 、 α_{in} 、 β_{in} 为目标图像的 l 、 α 、 β 三分量值; μ_{ref}^l 、 μ_{ref}^α 、 μ_{ref}^β 、 σ_{ref}^l 、 $\sigma_{\text{ref}}^\alpha$ 、 $\sigma_{\text{ref}}^\beta$ 分别为源图像的 $l\alpha\beta$ 三分量的均值和标准差; μ_{in}^l 、 μ_{in}^α 、 μ_{in}^β 、 σ_{in}^l 、 $\sigma_{\text{in}}^\alpha$ 、 σ_{in}^β 分别为目标图像的 l 、 α 、 β 三分量的均值和标准差。

(3) 将调整后的目标图像分量由 $l\alpha\beta$ 变换到 RGB 颜色空间,完成目标图像的色彩处理。

【例 5.8】 编写程序,对彩色图像进行色彩迁移。

解：程序如下。

```
clear,clc,close all;
ImageS = imread('img5_8_s.jpg');
ImageT = imread('img5_8_t.jpg');
subplot(131),imshow(ImageS);title('源参考图像');
subplot(132),imshow(ImageT);title('要处理的目标图像');
[sh,sw,sc] = size(ImageS); [th,tw,tc] = size(ImageT);
[sR,sG,sB] = imsplit(double(ImageS));
[tR,tG,tB] = imsplit(double(ImageT));
Trans1 = [0.3811,0.5783,0.0402;0.1967,0.7244,0.0782;0.0241,0.1288,0.8444];
Trans2 = [1 1 1;1 1 -2;1 -1 0];
Trans3 = [1/sqrt(3) 0 0;0 1/sqrt(6) 0;0 0 1/sqrt(2)]; % 色彩空间变换矩阵
svR = sR(:)'; svG = sG(:)'; svB = sB(:)';
sLMS = Trans1 * [svR;svG;svB];
tvR = tR(:)'; tvG = tG(:)'; tvB = tB(:)';
tLMS = Trans1 * [tvR;tvG;tvB]; % 将 RGB 数据转换为 LMS 数据
slab = Trans3 * Trans2 * [log10(sLMS)];
tlab = Trans3 * Trans2 * [log10(tLMS)]; % 再转到  $l\alpha\beta$  色彩空间
s_mu = mean(slab,2); s_sigma = std(slab,0,2); % 源图像  $l\alpha\beta$  各通道均值和标准差
t_mu = mean(tlab,2); t_sigma = std(tlab,0,2); % 目标图像  $l\alpha\beta$  各通道均值和标准差
tlab = (tlab - t_mu) .* s_sigma ./ t_sigma + s_mu; % 目标图像色彩迁移
tLMS = (Trans3 * Trans2) \ tlab;
tRGB = Trans1 \ (10.^tLMS); % 从  $l\alpha\beta$  色彩空间变换回 RGB 空间
tRGB(tRGB > 255) = 255; tRGB(tRGB < 0) = 0; % 限幅
result1 = reshape(tRGB(1,:),th,tw);
result2 = reshape(tRGB(2,:),th,tw);
result3 = reshape(tRGB(3,:),th,tw);
result = uint8(cat(3,result1,result2,result3));
subplot(133),imshow(result);title('Reinhard 色彩迁移图像');
```

程序运行结果如图 5-18 所示。



图 5-18 Reinhard 色彩迁移效果

色彩迁移不仅可以在两幅彩色图像之间进行,也可以在彩色图像和灰度图像之间进行,灰度图像色彩迁移主要是利用源参考图像的彩色信息,对灰度图像进行自动彩色化。关于灰度图像色彩迁移算法的设计实现,请扫二维码查看讲解。

5.4.2 色彩平衡

当一幅彩色图像数字化后,在显示时,颜色有时会看起来不正常,这是因为颜色通道中不同的敏感度、增光因子、偏移量等,导致数字化中的三个图像分量出现不同的变换,使结果图像的三原色“不平衡”,从而使景物中所有物体的颜色都偏离了其原有的真实色彩。因此,为了获得正常颜色的图像,将有色偏的图像进行颜色校正的过程就是色彩平衡。

下面主要介绍两种色彩平衡处理算法:白平衡和最大颜色值平衡算法。



微课视频

1. 白平衡算法

白平衡算法基本原理是,对于原始场景中某些理应为白色的像素点(即 $R_k^* = G_k^* = B_k^* = 255$),由于色偏现象的影响,这些点的 RGB 三分量的值不再保持相同,通过调整这些点的三分量值,使之达到平衡,由此获得一套色彩平衡映射关系式,将这套映射关系式应用于整幅彩色图像,即可达到图像色彩平衡的目的。

白平衡方法有多种,下列是一种基本算法的步骤。

(1) 对存在色偏的图像,按照下式计算亮度分量:

$$Y = 0.299R + 0.587G + 0.114B \quad (5-34)$$

(2) 由于现实场景中的白色点,在图像中可能不是理想状态的白色,即 $Y \neq 255$,但白色点亮度为图像中的最大亮度,则要求出图像中的最大亮度 $Y_{\max}$ 和平均亮度 $\bar{Y}$ 。

(3) 考虑到环境光照的适应性,寻找图像中所有亮度 $\leq 0.95Y_{\max}$ 的像素点构成的运算点集。

(4) 计算白色点集中的所有像素的 RGB 三分量的均值 $\bar{R}$ 、 $\bar{G}$ 、 $\bar{B}$ 。

(5) 按下式计算色彩平衡调整参数 k_R 、 k_G 、 k_B 。

$$k_R = \frac{\bar{Y}}{\bar{R}}, \quad k_G = \frac{\bar{Y}}{\bar{G}}, \quad k_B = \frac{\bar{Y}}{\bar{B}} \quad (5-35)$$

(6) 按下式对整幅图像的 R 、 G 、 B 三个颜色分量进行色彩平衡调整:

$$R_k^* = k_R \cdot R, \quad G_k^* = k_G \cdot G, \quad B_k^* = k_B \cdot B \quad (5-36)$$

【例 5.9】 编写程序,对图像进行白平衡处理。

解: 程序如下。

```
clear,clc,close all;
Image = im2double(imread('img5_9.jpg'));
[R,G,B] = imsplit(Image);
Y = 0.299 * R + 0.587 * G + 0.114 * B;
maxY = max(Y(:)); muY = mean(Y(:));
R_set = R(Y <= 0.95 * maxY); G_set = G(Y <= 0.95 * maxY); B_set = B(Y <= 0.95 * maxY);
% 运算点集对应的 RGB 值
kR = muY/mean(R_set(:)); kG = muY/mean(G_set(:)); kB = muY/mean(B_set(:));
% 计算色彩平衡调整参数
result = cat(3,kR * R,kG * G,kB * B);
subplot(121),imshow(Image),title('原图');
subplot(122),imshow(result),title('白平衡后图像');
```

程序运行结果如图 5-19 所示。



(a) 原图



(b) 白平衡后图像

图 5-19 白平衡效果

2. 最大颜色值平衡算法

白平衡算法对存在白色像素点的图像有较好的色彩平衡效果。但若一幅图像中白色点不

存在,或者仅占很少比例,则白平衡方法的处理就不是很有效。最大颜色值平衡算法就是针对这种情况提出的色彩平衡算法。

最大颜色值平衡基本原理是,如果一幅图像存在色偏现象,则 RGB 三通道中存在某个颜色信息比较强的通道,通过对该颜色通道的抑制,或者对另外两个颜色信息较弱的颜色通道的增强,就可以达到色彩平衡的目的。

最大颜色值平衡算法的具体步骤如下。

(1) 对存在色偏的图像,计算其 RGB 三通道中每个颜色通道的最大强度值 $R_{\max}$ 、 $G_{\max}$ 、 $B_{\max}$ 。并计算最小值 $C_{\max} = \min\{R_{\max}, G_{\max}, B_{\max}\}$ 。

(2) 统计 RGB 三通道中每个颜色通道强度值 $\geq C_{\max}$ 的像素的个数 N_R 、 N_G 、 N_B 。并计算最大值 $N_{\max} = \max\{N_R, N_G, N_B\}$,其所对应的为颜色信息最强通道。

(3) 将 RGB 三通道中每个颜色通道像素值从大到小排序,寻找每个颜色通道中第 $N_{\max}$ 个最大的强度值 R_{th} 、 G_{th} 、 B_{th} 。

(4) 按下式计算色彩平衡的调整参数 k_R 、 k_G 、 k_B :

$$k_R = \frac{C_{\max}}{R_{th}}, \quad k_G = \frac{C_{\max}}{G_{th}}, \quad k_B = \frac{C_{\max}}{B_{th}} \quad (5-37)$$

(5) 按下式对整幅图像的 R、G、B 三个颜色分量进行色彩平衡调整。

$$R_k^* = k_R \cdot R, \quad G_k^* = k_G \cdot G, \quad B_k^* = k_B \cdot B \quad (5-38)$$

【例 5.10】 编写程序,对图像进行最大颜色值平衡处理。

解: 程序如下。

```
Image = double(imread('3.jpg'));
IR = Image(:, :, 1); IG = Image(:, :, 2); IB = Image(:, :, 3);
Cmax = min([max(IR(:)) max(IG(:)) max(IB(:))]); % 从 RGB 每个通道的最大强度值中取最小值
numR = length(find(IR >= Cmax)); numG = length(find(IG >= Cmax));
numB = length(find(IB >= Cmax)); % 统计 RGB 每个通道中像素值 >= Cmax 的像素数目
Nmax = max([numR numG numB]);
Tr_temp = sort(IR(:), 'descend'); Tg_temp = sort(IG(:), 'descend');
Tb_temp = sort(IB(:), 'descend'); % 排序
Tr = Tr_temp(Nmax); Tg = Tg_temp(Nmax);
Tb = Tb_temp(Nmax); % 寻找每个通道中第 Nmax 个最大的强度
kr = Cmax/Tr; kg = Cmax/Tg; kb = Cmax/Tb; % 计算色彩平衡调整参数
res = cat(3, kr * IR, kg * IG, kb * IB);
imshow(uint8(Image));
figure, imshow(uint8(res)); title('最大颜色值平衡后图像');
```

程序运行结果如图 5-20 所示。



(a) 原图



(b) 最大颜色值平衡后图像

图 5-20 最大颜色值平衡效果

5.5 其他图像增强方法

除了前述的常见的图像增强方法外,模糊增强、基于对数图像处理模型的图像增强、图像去雾增强等方法在图像增强处理中也得到了成功应用。

5.5.1 模糊增强

模糊性就是事物的性质或类属的一种不分明性。1965年,美国加利福尼亚大学控制论专家 L. A. Zadeh 教授提出了用模糊集来处理这种模糊性。从此,以模糊集为基础的模糊数学诞生了。目前,模糊技术已被广泛地应用于自然科学与社会科学的许多领域。

令 U 为元素(对象)集, u 表示 U 的一类元素,即 $U = \{u\}$,则该集合称为论域 U 。论域 U 到 $[0,1]$ 闭区间的任一映射 μ_A 为

$$\mu_A: U \rightarrow [0,1], \quad u \rightarrow \mu_A(u) \quad (5-39)$$

都确定 U 的一个模糊集合 A , μ_A 称为模糊集合的隶属函数。 $\mu_A(u)$ 称为 u 对于 A 的隶属度,取值范围为 $[0,1]$,其大小反映了 u 对于模糊集合 A 的从属程度。

因此,模糊集合 A 完全可由 u 值和相应的隶属度函数 $\mu_A(u)$ 来表示描述,即

$$A = \{u, \mu_A(u) \mid u \in U\} \quad (5-40)$$

相应地,模糊集合的运算可由其隶属函数的运算来定义。

1. 图像的模糊特征平面

依照模糊集的概念,一幅灰度级为 L 的 $M \times N$ 的二维图像 $\mathbf{X}$,可以看作一个模糊点阵,记为

$$\mathbf{X} = \begin{bmatrix} \mu_{11}(x_{11}) & \mu_{12}(x_{12}) & \cdots & \mu_{1N}(x_{1N}) \\ \mu_{21}(x_{21}) & \mu_{22}(x_{22}) & \cdots & \mu_{2N}(x_{2N}) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{M1}(x_{M1}) & \mu_{M2}(x_{M2}) & \cdots & \mu_{MN}(x_{MN}) \end{bmatrix}$$

或

$$\mathbf{X} = \bigcup_{m=1}^M \bigcup_{n=1}^N \mu_{mn}(x_{mn}) \quad (5-41)$$

式中, x_{mn} 是图像中像素 (m, n) 的灰度, $\mu_{mn}(x_{mn})$ 表示图像中像素 (m, n) 的灰度 x_{mn} 相对于某些特定灰度级 x 的隶属度,且 $0 \leq \mu_{mn} \leq 1$ 。换句话说,一幅图像 $\mathbf{X}$ 的模糊集合是一个从 $\mathbf{X}$ 到 $[0,1]$ 的映射 μ_{mn} 。对于任一点 $x_{mn} \in \mathbf{X}$,称 $\mu_{mn}(x_{mn})$ 为 x_{mn} 在 μ_{mn} 中的隶属度。

可以看出,隶属度函数 $\mu_{mn}(x_{mn})$ 可以用于表征图像的模糊特征,可将图像从空间灰度域变换到模糊域。

2. 图像的模糊增强

图像的模糊增强的具体算法步骤如下。

(1) 进行模糊特征提取,将图像从空间灰度域变换到模糊域,有

$$\mu_{mn} = T(x_{mn}) = \left[1 + \frac{x_{\max} - x_{mn}}{F_d} \right]^{-F_e} \quad (5-42)$$

式中, x_{mn} 是图像中像素 (m, n) 的灰度, $x_{\max}$ 是图像中最大灰度值, F_d 和 F_e 分别为指数和分数模糊因子,这些模糊因子可以在模糊域内改变 μ_{mn} 的值。一般情况下,指数因子 F_e 取值为 2,分数因子 F_d 取值为

$$F_d = \frac{x_{\max} - x_c}{2^{\frac{1}{F_e}} - 1} \quad (5-43)$$

式中, x_c 为渡越点, 其取值需要满足: $\mu_c = T(x_c) = 0.5$ 且 $x_c \in \mathbf{X}$ 。

(2) 在模糊域, 对模糊特征进行一定的增强变换处理, 有

$$\begin{cases} \mu'_{mn} = I_r(\mu_{mn}) = \begin{cases} 2\mu_{mn}^2, & 0 \leq \mu_{mn} < 0.5 \\ 1 - 2(1 - \mu_{mn})^2, & 0.5 \leq \mu_{mn} < 1 \end{cases} \\ I_r(\mu_{mn}) = I_1(I_{r-1}(\mu_{mn})) \end{cases} \quad (5-44)$$

式中, μ'_{mn} 为增强后的模糊域像素灰度值; r 为正整数, 表示迭代次数。可以根据不同需求选择迭代次数。

(3) 反变换, 得到新的模糊增强后的输出图像。反变换如下:

$$z_{mn} = I^{-1}(\mu'_{mn}) = x_{\max} - F_d \left[(\mu'_{mn})^{\frac{1}{F_e}} - 1 \right] \quad (5-45)$$

图像的模糊增强结果如图 5-21 所示。

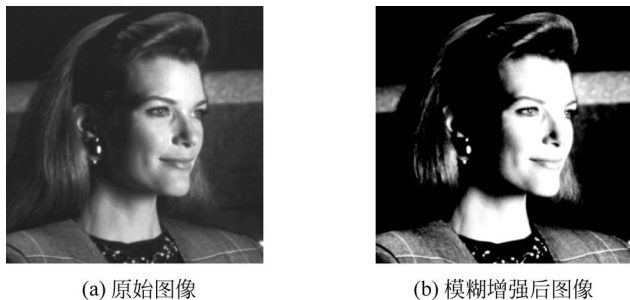


图 5-21 模糊增强效果

5.5.2 基于对数图像处理模型的图像增强

通常, 在对图像进行线性运算处理过程中, 会出现许多边际效应。首先, 可能会使输出值超出允许的灰度取值范围, 产生“超区间值”问题, 这类输出值会被修剪掉, 从而导致图像信息的严重丢失; 其次, 线性运算结果与人类的视觉感官往往有所偏差。所以, 一般的标量乘法“ $\times$ ”和加法“ $+$ ”运算并不通用于图像的形成法则。对数图像处理(Logarithmic Image Processing, LIP)模型的提出, 不仅解决了“数值溢出”的缺陷问题, 而且该模型符合人类视觉系统的非线性(对数)感知处理过程, 能更好地分析图像的细节, 突出图像中人们感兴趣的区域, 一直是人们研究的热点。后来, 人们从经典的对数图像处理模型发展出许多改进的对数图像处理模型, 例如 GLIP、PLIP、HLIP、SLIP、PSLIP 等。本节主要介绍经典的对数图像处理模型及其在对比度增强中的应用。

1. 对数图像处理模型

在对数图像处理模型中, 一幅图像通常用灰度色调函数表示。设图像支撑集为 $D \subset \mathbf{R}^2$, 灰度色调函数的定义如下所示。

$$f(x, y) = M - I(x, y) \quad (5-46)$$

式中, $I(x, y)$ 为输入的原始图像, $f(x, y)$ 为定义在 $[0, M)$ 区间内的灰度色调函数, M 是严格的正值。对于一般的 8 比特的图像, M 一般取值为 256。

灰度色调函数的物理意义是光线通过一个光强滤波器形成了进入人眼的透射光从而成

像,本质上就是把原图像转换为对应的光强滤波函数表示,该光强滤波函数就被定义为灰度色调函数。

对数图像处理模型是一个完备的数学理论,它规定了一系列特殊的代数运算和函数使运算处理前后像素的灰度值都在允许的灰度范围 $[0, M)$ 内。

1) 基础向量运算

对数图像处理模型定义了三种基本向量运算:对数加法、对数乘法和对数减法。

假设 $f_1(x, y)$ 、 $f_2(x, y)$ 是定义在 $[0, M)$ 区间内的灰度色调函数,则有如下 3 种运算。

(1) 对数加法运算: $f_1(x, y)$ 和 $f_2(x, y)$ 的对数加的定义为

$$f_1(x, y) \oplus f_2(x, y) = f_1(x, y) + f_2(x, y) - \frac{f_1(x, y) \cdot f_2(x, y)}{M} \quad (5-47)$$

(2) 对数乘法运算: 设 λ 为一个正实标量,则 $f_1(x, y)$ 和 λ 的对数乘的定义为

$$\lambda \otimes f_1(x, y) = M - M \left(1 - \frac{f_1(x, y)}{M} \right)^\lambda \quad (5-48)$$

(3) 对数减法运算: $f_1(x, y)$ 和 $f_2(x, y)$ 的对数减的定义为

$$f_1(x, y) \ominus f_2(x, y) = M \cdot \frac{f_1(x, y) - f_2(x, y)}{M - f_2(x, y)} \quad (5-49)$$

可以根据以上基本运算来构造图像处理的一些复杂运算来对图像进行相应的处理。

2) 基本同态函数

对数图像处理模型的基本同态函数的正变换定义为

$$\varphi(f) = -M \cdot \ln \left(1 - \frac{f(x, y)}{M} \right) \quad (5-50)$$

其反变换定义为

$$\varphi^{-1}(f) = M \cdot (1 - e^{\frac{-f(x, y)}{M}}) \quad (5-51)$$

通过对数图像处理模型的基本同态函数的正变换,灰度色调函数 $f(x, y)$ 的取值范围由区间 $[0, M)$ 扩大到 $(-\infty, M)$,但小于零的部分没有实际意义,且容易扭曲图像的负值信息。由于对数图像处理模型的运算空间和输入图像的实数空间通过正变换、反变换公式同构,所以可以由此推出对数图像处理模型空间的运算公式在实数空间的表示式。

2. 基于对数图像处理模型的增强

假设 $f(x, y)$ 是原始输入图像的灰度色调函数, $f'(x, y)$ 是增强处理后图像的灰度色调函数, $A(x, y)$ 是以像素点 (x, y) 为中心的 $n \times n$ 的邻域窗口的灰度色调函数的平均值,基于对数图像处理模型的增强算法表达式为

$$f'(x, y) = \alpha \otimes A(x, y) \oplus \beta [f(x, y) \ominus A(x, y)] \quad (5-52)$$

其中,参数 α 用来调整图像对比度。当 $\alpha > 1$ 时,图像变亮;当 $\alpha < 1$ 时,图像变暗。参数 β 用来调整图像的锐化效果。当 $\beta > 1$ 时,增强图像边缘;当 $\beta < 1$ 时则反之,图像边缘模糊。

若将 $f(x, y)$ 进行基本正态函数的正变换处理,且令 $\bar{f}(x, y) = 1 - f(x, y)/M$,则可进一步得到化简后的基于对数图像处理模型的增强算法表达式为

$$\ln \bar{f}'(x, y) = \alpha \cdot \ln \bar{A}(x, y) + \beta \cdot [\ln \bar{f}(x, y) - \ln \bar{A}(x, y)] \quad (5-53)$$

其中

$$\ln \bar{A}(x, y) = \frac{1}{n \times n} \sum_{k=x-\frac{n}{2}}^{x+\frac{n}{2}} \sum_{l=y-\frac{n}{2}}^{y+\frac{n}{2}} \ln \bar{f}(x, y) \quad (5-54)$$

简化后的增强算法的计算复杂度大幅度降低,同时增大了动态调节范围及边缘信息,实现了快速增强图像的目的。

基于对数图像处理模型的图像增强的具体算法步骤如下。

- (1) 对输入图像 $I(x, y)$ 进行变换,求得灰度色调函数 $f(x, y)$ 。
- (2) 对 $f(x, y)$ 进行基本正态函数的正变换 $\varphi(f)$,且有 $\varphi(f) = \ln \bar{f}$ 。
- (3) 如式(5-54)所示,求以像素点 (x, y) 为中心的 $n \times n$ 的邻域窗口的平均值 $\ln \bar{A}$ 。
- (4) 如式(5-53)所示,进行基于对数图像处理模型的增强处理。
- (5) 进行基本正态函数的反变换 $\varphi^{-1}(f)$ 。

基于对数图像处理模型的图像增强结果如图 5-22 所示。图 5-22(a)为原始图像;图 5-22(b)为基于对数图像处理模型增强后的图像,可以看出图像的对比度明显增强。



图 5-22 基于对数图像处理模型的图像增强效果

5.5.3 图像去雾增强

摄像机所拍摄的图像,由于受到雾的影响,往往出现模糊不清、对比度下降、色彩失真等质量退化的现象,严重影响到后续的处理工作。图像去雾就是采用一定的处理方法或手段,去除图像中雾的影响,提高图像的对比度,增强图像的细节信息,获得视觉效果较好的图像。

目前,图像去雾方法主要分为两类。一类方法是基于非物理模型的去雾方法,是不考虑雾导致图像退化的成因,只通过实现对比度增强方法来达到去雾的目的。该类方法虽然能一定程度上达到去雾效果,但是可能会造成部分信息的损失。例如,子块部分重叠的局部直方图均衡化方法、对比度受限自适应直方图均衡化方法、Retinex 增强方法等。另一类方法是基于物理模型的去雾方法,考虑雾导致图像退化的成因,进行数学建模,补偿退化过程造成的失真,恢复无雾图像。该类方法获得的无雾图像,视觉效果自然,一般无信息损失。

1. 图像去雾物理模型

在恶劣的大气环境(霾、雾、灰尘等)中进行拍摄时,光线从物体表面到达观测点(如摄像头)之前,被大气中的悬浮粒子散射,使得拍摄到的图像模糊不清,图像质量下降,雾天图像退化模型如图 5-23 所示。这种复杂的散射作用分为两类:光线通过大气介质时引起的入射光衰减效应和大气介质中的粒子散射的环境光所引起的大气光散射效应。

那么,入射光线到达摄像头的总光强度,为入射光线衰减后到达摄像头的光强和周围环境中的各种散射光线进入摄像头后的附加光强之和,即

$$I(x) = E_d(d, \lambda) + E_s(d, \lambda) \quad (5-55)$$

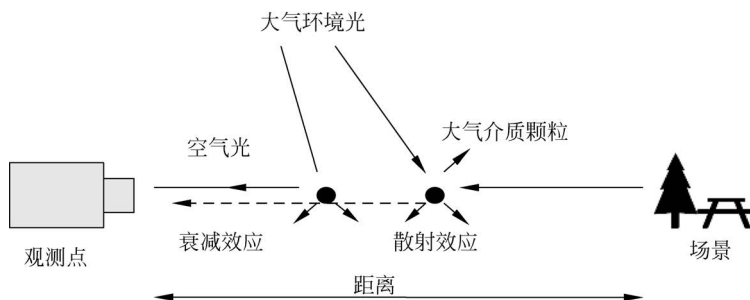


图 5-23 雾天图像退化模型

其中,

$$E_d(d, \lambda) = J(x) \cdot e^{-\beta(\lambda) \cdot d(x)} \quad (5-56)$$

$$E_s(d, \lambda) = A(1 - e^{-\beta(\lambda) \cdot d(x)}) \quad (5-57)$$

式中, x 代表二维空间位置。 $d(x)$ 为场景深度(图像上的目标到观察者的距离)。 $E_d(d, \lambda)$ 为直接衰减, 描述了场景目标反射光在介质中衰减的结果。 $E_s(d, \lambda)$ 为空气光, 反映了全局大气光的散射导致杂散光成像的情况。 $I(x)$ 为观察图像, 即输入图像。 $J(x)$ 为场景目标直接反射光成像的亮度。 A 为大气光, 是沿着观测视线场景物体在无穷远处的光强, 与 x 无关。 β 为空气的散射系数, 由于大雾天气对可见光的衰减几乎不受波长影响, 因此可以认为 β 与波长 λ 无关, 将 β 假设为一个不变的常量值, 得

$$I(x) = J(x) \cdot e^{-\beta \cdot d(x)} + A(1 - e^{-\beta \cdot d(x)}) \quad (5-58)$$

这就是图像去雾的物理模型。

2. 基于暗原色先验的去雾方法

He Kaiming 等通过对户外大量清晰无雾的自然图像观察统计得到: 在绝大多数非天空的局部区域内, 某一些像素在 RGB 三色通道中至少有一个通道的像素颜色值比较低。换言之, 该区域光强度的最小值是个很小的数。

若给出一幅清晰无雾图像 J , 暗通道 J^{dark} 用数学公式定义为

$$J^{\text{dark}}(i, j) = \min_{C \in \{r, g, b\}} \left(\min_{y \in \Omega(i, j)} (J^C(y)) \right) \quad (5-59)$$

式中, J^C 表示图像 J 的 RGB 三色通道中的一个颜色通道, $\Omega(i, j)$ 表示以点 (i, j) 为中心的一个局部区域块。对于非天空清晰无雾图像 J , J^{dark} 值非常低, 趋于 0, 那么 J^{dark} 称为 J 的暗原色, 并且把以上观察得出的经验性规律称为暗原色先验。

基于图像去雾的物理模型, 定义 $t(x)$ 为介质传播函数, 表示透射率, $0 \leq t(x) \leq 1$, 且为

$$t(x) = e^{-\beta \cdot d(x)} \quad (5-60)$$

则将式(5-58)变换为

$$I(x) = J(x) \cdot t(x) + A(1 - t(x)) \quad (5-61)$$

暗原色先验去雾的目的就是希望能够通过雾天图像 $I(x)$ 恢复无雾图像 $J(x)$ 。

附加的大气光导致图像被雾干扰之后往往要比其本身亮度更大, 透射率一般较小, 所以被浓雾覆盖的图像的暗原色具有较高的强度值。视觉上看来, 暗原色强度值是雾浓度的粗略近似。因此, 可利用这一性质来估算大气光 A 和透射率 $t(x)$ 。

1) 估计大气光

对于大气光 A 的估计, 由于其亮度值通常较高, 所以主要通过对图像最不透明区域像素的亮度值估算而得。通常通过计算图像的暗原色最亮区域来估算大气光 A 。

估计大气光 A 的具体算法步骤如下。

- (1) 划分雾天图像 I 的局部块 Ω , 求取其暗原色图 $I_{\text{dark_channel}}$;
- (2) 从暗原色图 $I_{\text{dark_channel}}$ 中按照亮度值从大到小取最亮的前 0.1% 像素, 该部分像素的位置通常对应图像中最不透明的区域;
- (3) 在图像 I 中定位步骤(2)求取的像素位置区域, 求该区域像素的亮度平均值 $I_{\text{mean_dc}}$;
- (4) 设置参数 $A_{\text{max}} = 240$, 计算 $\min(A_{\text{max}}, I_{\text{mean_dc}})$ 作为大气光 A 的估计值。这里, 设置参数 A_{max} 表示所允许的大气光 A 的估算最大值, 以避免所计算的 $I_{\text{mean_dc}}$ 值接近于 255 时可能会造成的处理后图像的色调偏差。

大气光 A 的估计如图 5-24 所示。其中, 图 5-24(a) 为原始图像, 所标记的蓝框区域, 代表原始图像中亮度最大区域, 值为 254。图 5-24(b) 为暗原色图, 所标记的红框区域, 代表暗原色图中的最亮点, 值为 208, 作为估计的大气光 A 的值。可以看出, 估计的大气光 A 并不一定为图像中亮度的最大值。

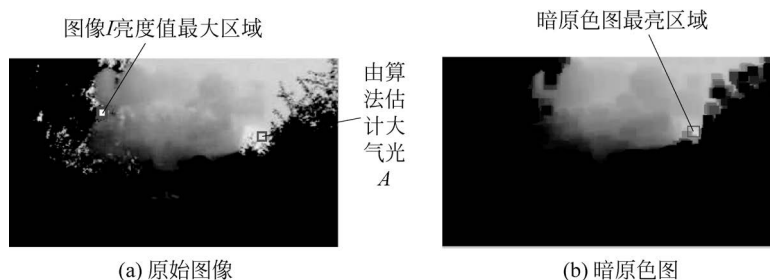


图 5-24 大气光估计

2) 估计透射率

基于暗原色先验理论和雾天图像成像模型, 可以直接估计出成像时刻的雾浓度, 从而粗略估算透射率 $t(x)$ 。首先已知大气光 A , 假定局部区域 Ω 内透射率保持一致, 对式(5-62)在 RGB 三个颜色通道进行最小运算, 可求得暗原色图。

$$\min_{C \in \{r, g, b\}} \left(\min_{x \in \Omega(i, j)} \left(\frac{I^C(x)}{A} \right) \right) = t(x) \cdot \min_{C \in \{r, g, b\}} \left(\min_{x \in \Omega(i, j)} \left(\frac{J^C(x)}{A} \right) \right) + (1 - t(x)) \quad (5-62)$$

这里, C 表示图像 RGB 三通道中的一个颜色通道。又根据暗原色先验规律, 无雾自然图像的暗原色值接近于 0, 且 $A > 0$, 则得

$$\min_{C \in \{r, g, b\}} \left(\min_{x \in \Omega(i, j)} \left(\frac{J^C(x)}{A} \right) \right) \rightarrow 0 \quad (5-63)$$

将式(5-63)代入式(5-62)进行变换, 可估算透射率为

$$t(x) = 1 - \min_{C \in \{r, g, b\}} \left(\min_{x \in \Omega(i, j)} \left(\frac{I^C(x)}{A} \right) \right) \quad (5-64)$$

又由于考虑空间透视现象、人眼的视觉特性及人观看景物时对于图像真实感和深度感的需求, 针对式(5-64)引入一个调节参数 $\omega \in (0, 1)$, ω 值的选取通过经验获得, 则得

$$t(x) = 1 - \omega \cdot \min_{C \in \{r, g, b\}} \left(\min_{x \in \Omega(i, j)} \left(\frac{I^C(x)}{A} \right) \right) \quad (5-65)$$

3) 获得无雾图像

将式(5-65)变形, 可得

$$J^C(x) = A + \frac{I^C(x) - A}{t(x)} \quad (5-66)$$

将有雾图像 $I(x)$ 、估计大气光 A 和透射率 $t(x)$ ，代入式(5-66)，计算获得无雾图像 $J(x)$ 。

基于暗原色先验去雾的图像增强结果如图 5-25 所示。

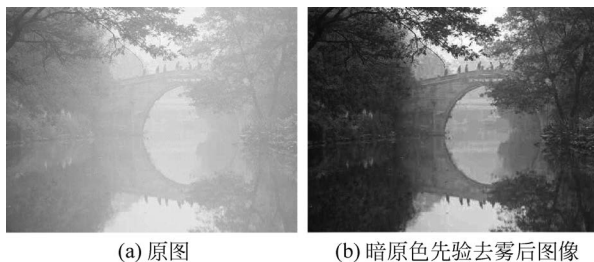


图 5-25 暗原色先验去雾增强效果

3. 基于暗原色先验的低照度图像增强

低照度图像通常是指在光照较暗或者夜间所拍摄得到的图像。如果将低照度图像反转后，其图像表征及直图表征与雾天图像具有很高的相似性，因此，可以采用图像去雾技术来实现低照度图像的增强。该算法的基本思想是：对输入的低照度图像进行反转，对反转图像进行去雾处理，然后对去雾结果反转得到低照度图像增强效果。

基于暗原色先验的低照度图像增强的具体算法步骤如下：

- (1) 对输入低照度图像 $I^C(x)$ 进行取反操作，求得反转图像 $R^C(x)$ ：

$$R^C(x) = 255 - I^C(x) \quad (5-67)$$

式中， C 表示图像的 RGB 三个颜色通道， x 表示图像的坐标点。

- (2) 对反转图像 $R^C(x)$ 进行暗原色先验去雾处理，获得无“雾”图像 $J^C(x)$ 。

- (3) 将无“雾”图像 $J^C(x)$ 反转，获得低照度增强图像 $P^C(x)$ ：

$$P^C(x) = 255 - J^C(x) \quad (5-68)$$

基于暗原色先验的低照度图像增强结果如图 5-26 所示。



图 5-26 基于暗原色先验的低照度图像增强结果

习题

5.1 试给出把灰度范围 $[0, 10]$ 伸长为 $[0, 15]$ ，把范围 $[10, 20]$ 移到 $[15, 25]$ 并把范围 $[20, 30]$ 压缩为 $[25, 30]$ 的变换方程。

5.2 在对图像进行直方图均衡化时，为什么会产生简并现象？

5.3 一幅图像的直方图如图 5-27 所示，可以对其进行什么处理？

5.4 一幅大小为 64×64 的图像，8 个灰度级对应像素数及概率 $p_r(r)$ 如表 5-4 所示，试对其进行直方图均衡化。

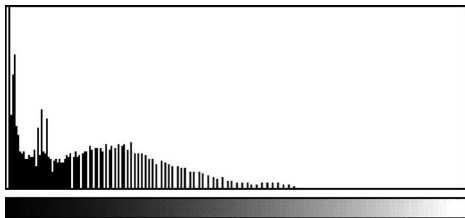


图 5-27 习题 5.3 图

表 5-4 习题 5.4 中图像的灰度级分布

灰度级 r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
像素数 n_k	560	920	1046	705	356	267	170	72
概率 $p_r(r)$	0.14	0.22	0.26	0.17	0.09	0.06	0.04	0.02

5.5 已知一幅图像的灰度级为 8,图像的左边一半为深灰色,其灰度值为 1/7,右边一半为黑色,其灰度值为 0。试对此图像进行直方图均衡化,并描述处理后的图像视觉效果。

5.6 同态滤波的特点是什么? 适用于什么情况?

5.7 Retinex 增强方法的特点是什么? 单尺度和多尺度 Retinex 增强方法的区别是什么?

5.8 Reinhard 色彩迁移方法的设计思想是什么?

5.9 白平衡的基本原理是什么? 色彩补偿的基本原理是什么?

5.10 编写程序,实现习题 5.1 中的灰度级变换。

5.11 编写程序,实现习题 5.3 中所设计的处理方法。

5.12 编写程序,实现真彩色图像的同态滤波。

5.13 编写程序,实现真彩色图像的基于对数图像处理模型的增强。

5.14 编写程序,实现有雾图像的基于暗原色先验的去雾处理,并且与其他基于非物理模型的去雾方法进行结果对比。