

## 拓展资源 5.2 补充内容



### 5.2.1 基于模糊逻辑和直方图增强算法

#### 1. 模糊逻辑原理

近年来,不少被提出的基于模糊逻辑的图像,增强算法的性能优于传统或其他先进的增强技术。模糊逻辑表示一个变量的真值可能表示为  $0 \sim 1$  之间任意的一个真实值。与布尔逻辑相反,模糊逻辑并不给出变量的明确值,模糊集的定义:

$$\mu_A(x): X \rightarrow [0,1], x \rightarrow \mu_A(x) \quad (5.2)$$

式中,表示  $\mu_A$  确定了  $x$  的其中一个模糊子集  $A$ ,  $\mu_A$  被称为  $A$  的隶属度函数,而  $\mu_A(x)$  是  $x$  对于  $A$  集合的隶属度,表示  $x$  属于  $A$  的程度。在模糊方法中,图像的灰度值通过隶属函数建模为一个模糊集,这时图像可以被看作一个存储模糊单点的数组,这些模糊单点都有各自的隶属度。模糊图像处理主要分为模糊、隶属度修正和去模糊三部分。首先利用隶属函数模糊图像,其次对得到的隶属度进行增强变换,最后通过去模糊函数得到新的灰度值。

以图像的边缘为例,从边缘到非边缘的过度不应该是干脆的,而应该是渐进的过度。对于边缘处的部分像素点,它们既可以属于边缘,也可以不属于边缘,这就存在模糊性。模糊增强的目的就在于使这些像素点要么属于边缘,要么就不属于。根据这个目的可以制定以下规则:

R1: IF 一个像素是暗的, THEN 让这个像素更暗;

R2: IF 一个像素是灰的, THEN 让他保持是灰的;

R3: IF 一个像素是亮的, THEN 让这个像素更亮;

从这个规则出发,制定模糊原则和隶属函数。

#### 2. 算法描述及实现步骤

该算法是在两个参数  $M$  和  $K$  的控制下,对 HSV 空间中的  $V$  通道进行拉伸扩展。算法首先计算  $V$  通道的直方图  $h(x)$ ,  $h(x)$  表示亮度值为  $x$  的像素个数。将  $M$  的值定义为图像亮度值的均值,计算公式如下:

$$M = \frac{\sum_M xh(x)}{\sum_M h(x)} \quad (5.3)$$

$M$  将直方图分为两部分,分别为  $[0, M-1]$  和  $[M, 255]$ , 即两个模糊子集。那么  $V$  通道的拉伸也就分为两部分,对于每一个部分,其隶属度修正的原则也不同。 $M$  在计算两个隶属度  $\mu_{D1}$  和  $\mu_{D2}$  时十分关键。参数  $K$  决定了原图像的亮度值应该被拉伸到哪个新的亮度值上,即  $K$  影响去模糊函数的表达式。根据参考文献获取的实验分析,  $K=128$  时,处理低照度和低对比度图像可获得较好的增强效果。

该算法对隶属度的确定是基于两个模糊原则:①在  $[0, M-1]$  区间内,亮度值与  $M$  相差越大的像素点,其拉伸的程度应该越小;②在  $[M, 255]$  区间内,亮度值与 255 相差越大的像素点,

其拉伸的程度应该越大。根据这两个原则，我们可以得到以下隶属度函数：

$$\mu_{D1} = 1 - \frac{M - x}{M} \quad (5.4)$$

$$\mu_{D2} = \frac{E - x}{E - M} \quad (5.5)$$

$E$  表示图像灰度值的最大值，对于 8bit 的图像来说， $E=255$ 。根据隶属函数与模糊原则，两个区间的去模糊函数表示如下：

$$X_{e1} = X + \mu_{D1}K \quad (5.6)$$

$$X_{e2} = \mu_{D2}X + E - \mu_{D2}K \quad (5.7)$$

### 3. 模糊增强代码

```
function img = FuzzyHE_original(I)
%对图像作基于模糊估计和直方图的增强变换
m=size(I,1);
n=size(I,2);
I1 = double(I);
I2 = zeros(m,n);
%求图像灰度值的期望
M = mean(I1(:));
K=128;
E=255; %8bit 图像
for i=1:m
    for j=1:n
        if I1(i,j)<M
            I2(i,j) = I1(i,j) + I1(i,j)/M*K; %区间[0,M-1]模糊增强
        else
            I2(i,j) = (E-I1(i,j))*(I1(i,j)-K)/(E-M)+E; %区间[M,E]模糊增强
        end
    end
end
end
% img = I2;%增强结果直接输出，可能出现反色的结果，部分值不在[0,255]之间
img = min(255,max(I2,0));%将像素值修正到区间[0,255]内
End
```

### 4. 实验结果

几组图像分析的实验结果如图 5.1 和图 5.2 所示。

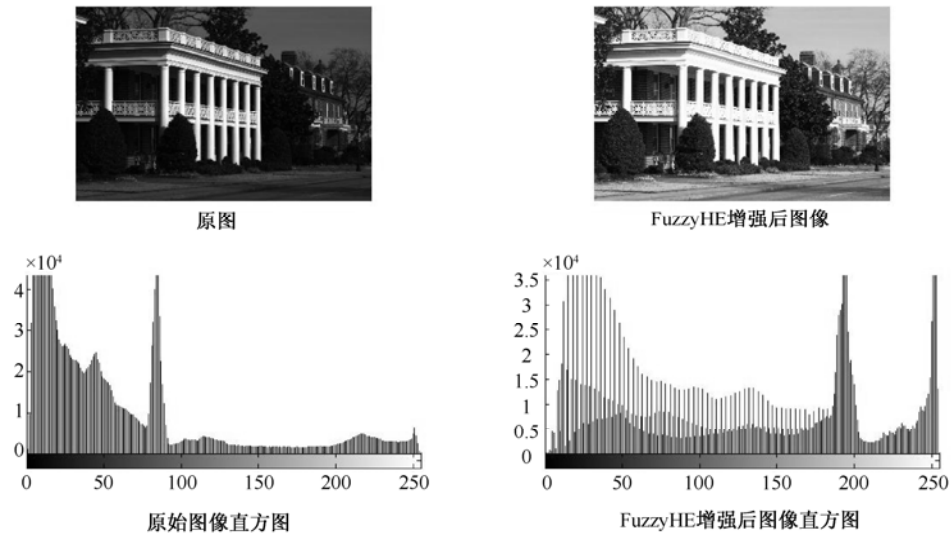


图 5.1 几组未量化到[0 255]内的实验结果

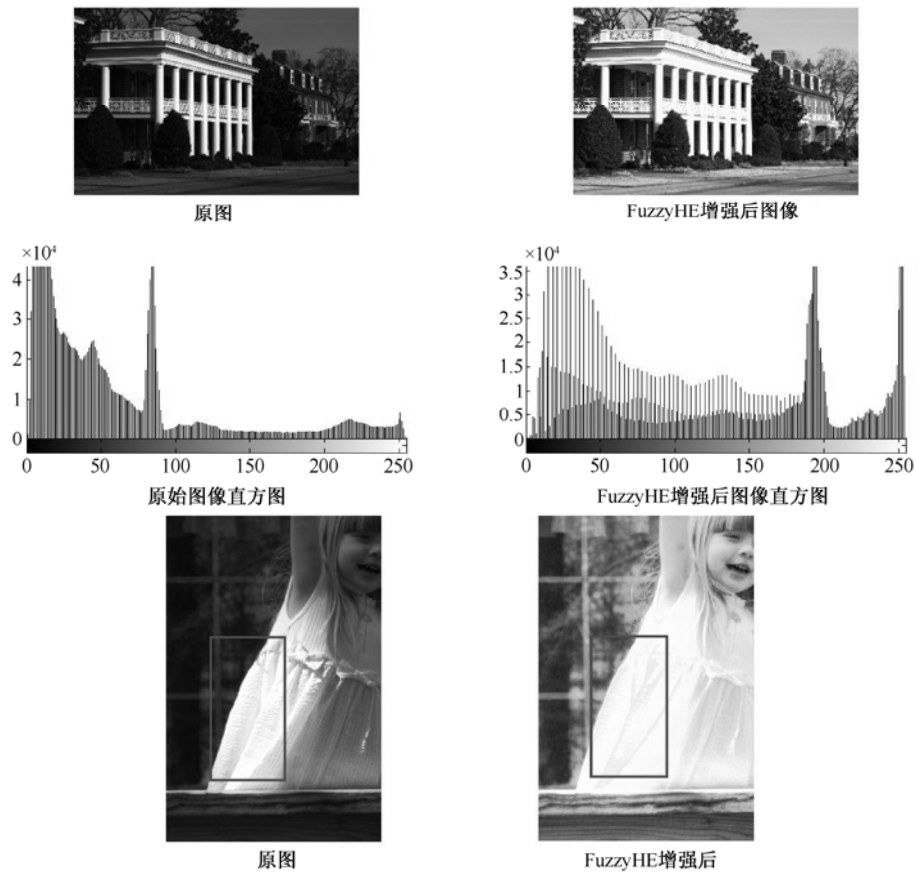


图 5.1 几组未量化到[0 255]内的实验结果（续）

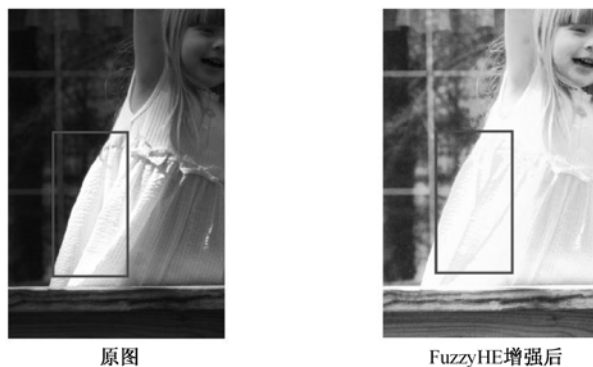


图 5.2 量化到[0 255]内的实验结果



## 5.2.2 基于模糊逻辑的自适应增强

### 1. 参数分析

5.3.1 节中的模糊增强算法是在两个关键的参数控制下进行的。图像亮度均值  $M$  是用来将图像直方图分成两个部分。改进型算法沿用此参数，将直方图分开处理。参数  $K$  的值决定了  $[0, M-1]$  区间内像素的灰度值映射范围为  $[0, M+K]$ 。然而不同的图像其亮度分布是不一样的。亮度分布不均的图像其灰度值波动较大，对于这类图像，将  $K$  设置为固定值 128 是不合理的。这样做可能会减小高亮度区域的对比度。所以考虑针对不同的图像， $K$  应该具有不同的值，且与图像的动态范围有关。因此，改进的算法引入均值和方差的影响，由这两项来决定  $K$  的取值：

$$K = M + D \times \text{Var} \quad (5.8)$$

其中， $M$  是图像的均值， $\text{Var}$  为图像的均方差， $D$  为调节系数。由式 (2.7) 和式 (3.2) 可以看出，对比强度  $K$  是隶属度的乘积系数。当  $K$  越大时，其增强后的值受隶属度的影响就越大，同时区间  $[0, M-1]$  的映射函数式 (2.7) 的斜率随着  $K$  值的增大而增大， $K$  越大， $[0, M-1]$  内像素拉伸程度越大。但是区间  $[M, 2 \times K-1]$  的映射函数式 (3.2) 的斜率随着  $K$  的增大而减小， $K$  越大，区间  $[M, 2 \times K-1]$  内像素的拉伸程度越低。因此限制了  $K$  不能过大，否则会导致  $[M, 2 \times K-1]$  区间内的对比度变小，且  $K$  也不能太小，否则达不到增强的目的。在此基础上， $D$  作为  $K$  的调节系数，在实验中，通过改变  $D$  的值，观察增强效果，选择最好增强效果时对应的  $D$  值，从而实现  $K$  的动态确定。即  $K$  值的动态确定是在增强低照度区域图像的同时又能保持高亮度区域的对比度中寻找一个平衡。通过实验，选择  $D=5$ ，可以满足大多数图像的增强效果。实验证明， $D=5$  时可以获得较好的效果。

### 2. 去模糊函数修正

由于原算法在区间  $[M, 255]$  的去模糊函数是非单调的，不符合模糊逻辑中单调的原则，导致了增强后的像素值超出了灰度级范围。从这一点出发，结合对  $K$  的修正，本文将区间改为  $[M, 2 \times K-1]$  中，此区间内的去模糊函数为：

$$X_{e2} = x - \mu_{D2} K + E \quad (5.9)$$

其中  $E=2 \times K-1$ 。修正以后，函数变为单调的，且拉伸区间变大，增强后的亮度值不局限

于 0~255，而是扩展到 0 到  $2 \times K - 1$ 。为了将增强的图像进行显示，就需要我们对增强后的亮度值进行量化，将其转换到 0~255。

### 3. 改进模糊增强的实现

```
function [img,k] = FuzzyHE(I)
%对图像作基于模糊估计和直方图的增强变换
%对原算法的改进版
m=size(I,1);
n=size(I,2);
I1 = double(I);
I2 = zeros(m,n);
%求图像灰度值的期望
M = mean(I1(:));
Var = std2(I1);
M2=M;
K=M+5*Var;    %将对比强度与均值和方差建立联系
% K=128;
E=2*K-1;    %将映射范围从[0,255]转变到[0,2*K-1]
for i=1:m
    for j=1:n
        if I1(i,j)<M2
            I2(i,j) = I1(i,j) + I1(i,j)/M2*K;
        else
            I2(i,j) =I1(i,j) -(E-I1(i,j))/(E-M2)*K+E; %修正后的去模糊函数
        end
    end
end
img = I2;
k=K;
End
```

### 4. 动态量化

量化一般是处理显示图像的前一步，量化的好坏直接决定了显示图像的品质。量化最直接的方式就是直接线性量化，表达式如下：

$$I(i, j) = \frac{I(i, j) - \min(\min(I))}{\max(\max(I)) - \min(\min(I))} 255 \quad (5.10)$$

这种直接量化对于那些高动态范围，数据分布广的图像会出现两极化的现象。本章改进的增强算法在对图像增强后，其像素值分布较广，所以不适用这种量化方式。

GIMP 在实现 Retinex 图像处理时，提出了一种基于均值和方差的量化方式，并加上一个控制图像动态的参数来实现无色偏的调节过程，简要描述如下：

(1) 计算输入图像通道的均值 mean 和均方差 var;

(2) 利用下列公式计算通道的最大值和最小值:

$$\min = \text{mean} - \text{dynamic} \times \text{var} \quad (5.11)$$

$$\max = \text{mean} + \text{dynamic} \times \text{var} \quad (5.12)$$

(3) 对每个像素值进行线性映射:

$$\text{value} = \frac{\text{value} - \min}{\max - \min} 255 \quad (5.13)$$

(4) 将大于 255 的数值赋值为 255, 对小于 0 的数值赋值为 0。

由于该量化考虑了图像的动态变化, 所以适用于数据分布广的图像。经过多次实验验证调节系数  $\text{dynamic}=2$  时, 适用于大多数图像。个别图像也可以通过调整此参数获得较好的效果。

本章希望改进的算法, 能兼顾整体低光照的和含有过亮区域的照度分布不均的图像的处理。那么处理前应该先自适应识别这些图像, 判断其光照是否分布不均。本章将采用上面计算的  $\min$  的值作为判断标准:

(1) 当  $\min$  小于 0 时, 我们判断该图像为照度分布不均的图像; 那么为了保持其拉伸效果, 不进行上述的第 (4) 步操作;

(2) 当  $\min$  不小于 0 时, 我们判断该图像为整体低照度的图像; 那么由式 (3.6) 计算得到的像素值可能会小于 0, 因此对于这类图像, 我们执行第 (4) 步操作, 以便后续的处理。

## 5. 动态量化的实现

```
function new_R = conMV(R,a)
%R: 输入图像
%a: 方差前的系数, 实验结果验证 a 值多取 2, 但也可适当增大其值, 不能小于 2
m=size(R,1);
n=size(R,2);
mm = mean(R(:));
Var = std2(R(:));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%GMIP 量化%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Min = mm-a*Var;
Max = mm+a*Var;
new_R = round((R-Min)/(Max-Min).*255);
if Min>0 %该判断可用于判断图像是否动态范围大, Min<0 则说明动态范围大, 为了
        保持其动态特性, 不作量化到[0,255]的处理
new_R = min(max(new_R,0),255); %Min>0 判断该图像动态范围小, 作量化处理
End
```

## 6. 实验结果

二组图像分析的实验结果如图 5.3 所示。

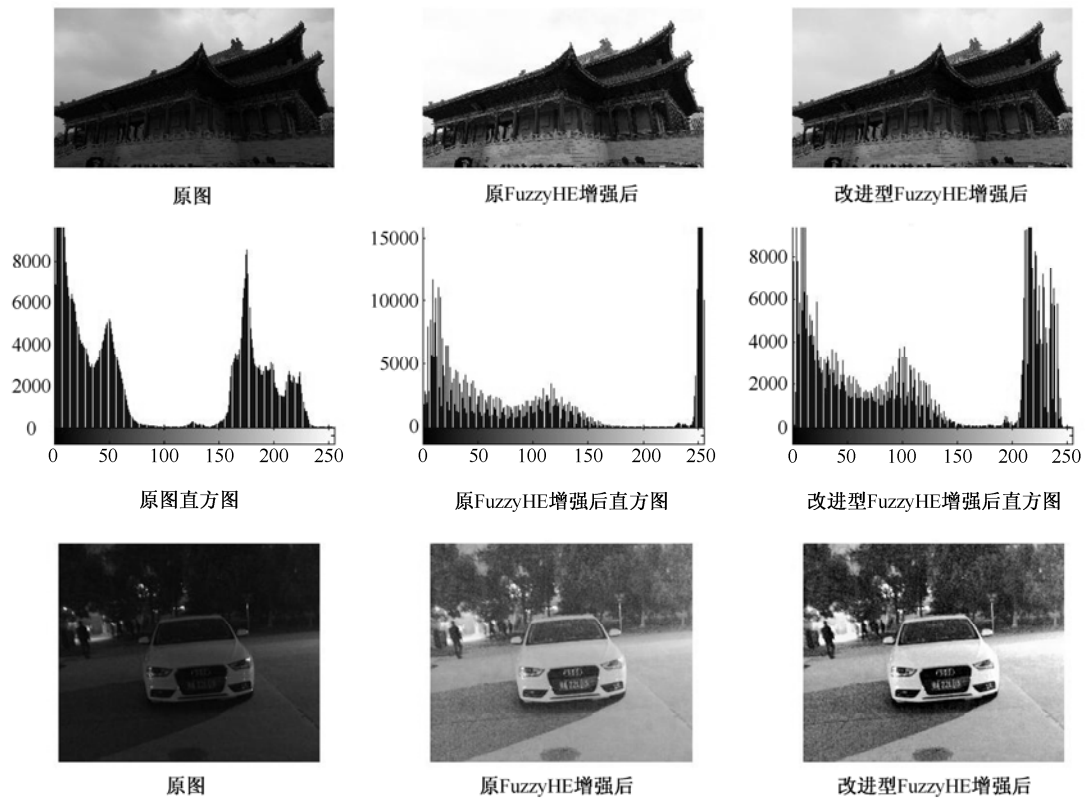


图 5.3 基于模糊逻辑的自适应增强

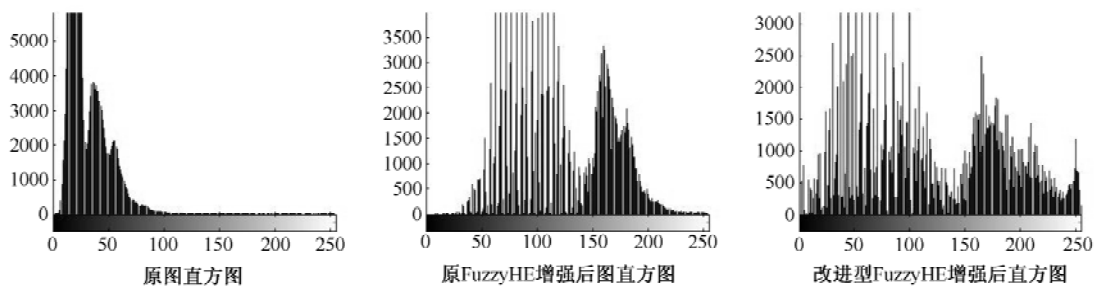


图 5.3 基于模糊逻辑的自适应增强 (续)