

图像增强的几个方法以及 Matlab 代码

1. 灰度线性变换

灰度线性变换， 是一种空域的方法， 直接对每一个像素的灰度值进行操作

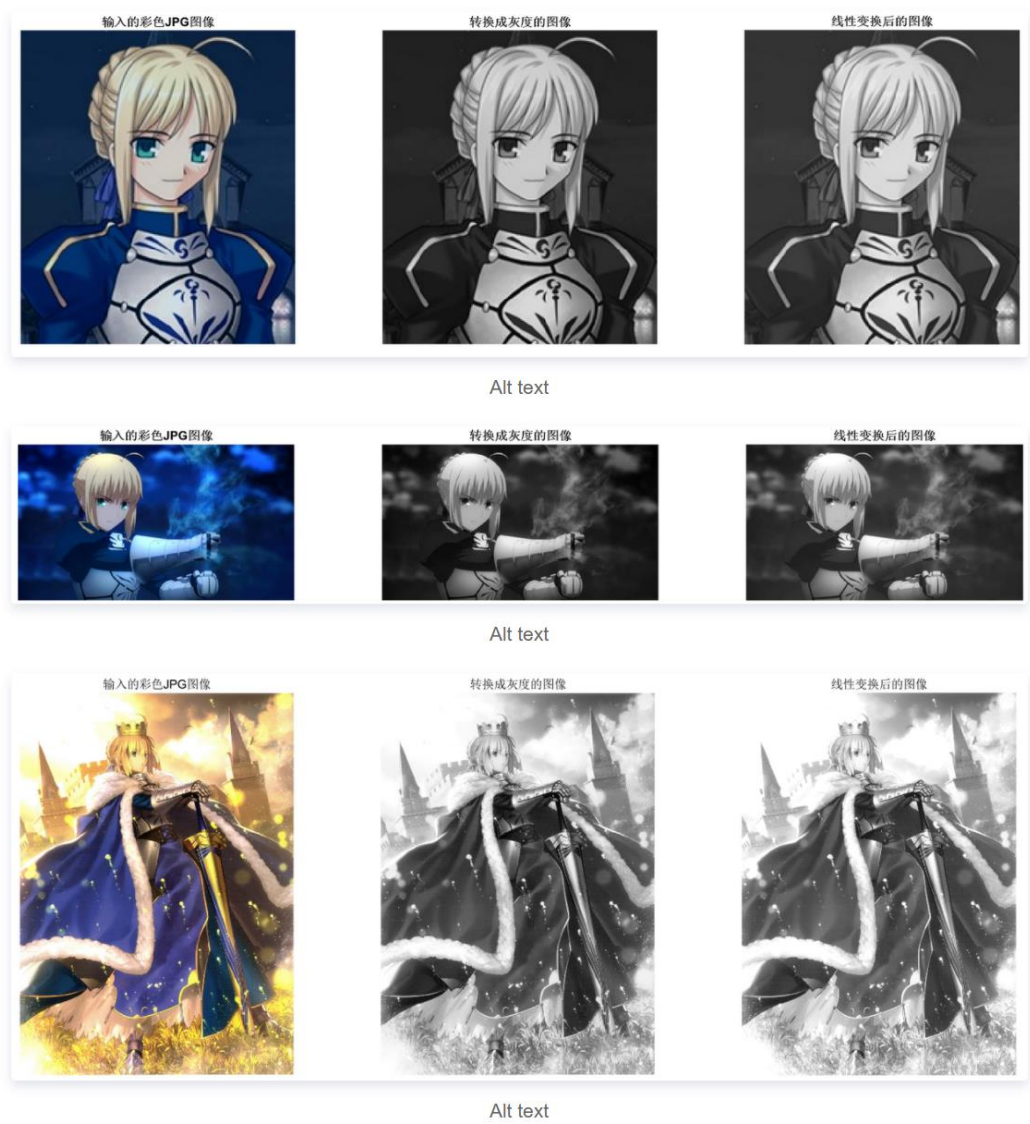
假设图像为 I

则其中每一个像素点的灰度值为 $I(x,y)$

我们利用简单的线性变换可以得到：

$$I(x,y) \ast = k \ast I(x,y) + b$$

取 $k=1, b=16$ 可以得到



这里给出关键代码

```
original = imread(strcat(strcat('resource\',name),'.bmp'));
transformed = LinearFunction(original, 1, 16);
subplot(1,3,3);
imshow(transformed)
title('线性变换后的图像')
imwrite(transformed,strcat(strcat('result\',name),'(linear).jpg'))
```

2. 直方图均衡变换

这种方法通常用来增加许多图像的全局对比度，尤其是当图像的有用数据的对比度相当接近的时候。通过这种方法，亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度，直方图均衡化通过有效地扩展常用的亮度来实现这种功能。

这种方法对于背景和前景都太亮或者太暗的图像非常有用，这种方法尤其是可以带来 X 光图像中更好的骨骼结构显示以及曝光过度或者曝光不足照片中更好的细节。这种方法的一个主要优势是它是一个相当直观的技术并且是可逆操作，如果已知均衡化函数，那么就可以恢复原始的直方图，并且计算量也不大。这种方法的一个缺点是它对处理的数据不加选择，它可能会增加背景噪声的对比度并且降低有用信号的对比度。

考虑一个离散的灰度图像{x}

让 n_i 表示灰度 i 出现的次数，这样图像中灰度为 i 的像素出现的概率为：

$$p_x(i)=p(x=i)=n_i/n, \quad 0 \leq i < L$$

L 是图像中所有的灰度数（通常是 255）

把对应于 p_x 的累积分布函数，定义为：

$$cdf_x(i)=\sum_{j=0}^i p_x(j)$$

是图像的累计归一化直方图

我们创建一个形式为 $y=T(x)$ 的变换，对于原始图像的每一个值它就产生一个 y ，这样 y 的累计概率函数就可以在所有值范围内线性化，转换公式定义为：

$$cdf_y(i)=i/K$$

对于常数 K ，CDF 的性质允许我们做这样的变换，定义为：

$$cdf_y(y')=cdf_y(T(k))=cdf_x(k)$$

其中 $k \in [0, L)$ ，注意 T 将不同的灰度级映射到 $[0, 1)$ ，为了将这些值映射到最初的域，

需要在结果上应用下面的简单变换：

$$y'=y*(\max\{x\}-\min\{x\})+\min\{x\}$$

给出部分结果：



Alt text



Alt text



Alt text

给出关键代码:

```
m = 255;
subplot(1,3,3);
H = histeq(I,m);
imshow(H,[]);
title('直方图均衡后的图像');
```

类似地， 我们可以对 rgb 图像进行直方图均衡，这里直接给出结果



Alt text



Alt text



Alt text

```
[M,N,G]=size(I1);
result=zeros(M,N,3);
%获得每一层每一个点的RGB值，并判断其值等于多少
for g=1:3
    A=zeros(1,256);
    %每处理完一层，参数要重新初始化为0
    average=0;
    for k=1:256
        count=0;
        for i=1:M
            for j=1:N
                value=I1(i,j,g);
                if value==k
                    count=count+1;
                end
            end
        end
        count=count/(M*N*1.0);
        average=average+count;
        A(k)=average;
    end
    A=uint8(255.*A+0.5);
    for i=1:M
        for j=1:N
            I1(i,j,g)=A(I1(i,j,g)+0.5);
        end
    end
end
%展示处理效果
subplot(1,3,3);
imshow(I1);
```

3. 同态滤波

同态滤波利用去除乘性噪声(multiplicative noise)，可以同时增加对比度以及标准化亮度，借此达到图像增强的目的。

一副图像可以表示为其照度(illumination)分量和反射(reflectance)分量的乘积，虽然在时域上这两者是不可分离的，但是经由傅立叶转换两者在频域中可以线性分离。由于照度可视为环境中的照明，相对变化很小，可以看作是图像的低频成分；而反射率相对变化较大，则视为高频成分。通过分别处理照度和反射率对像元灰度值的影响，通常是借由高通滤波器(high-pass filter)，让图像的照明更加均匀，达到增强阴影区细节特征的目的。

做法是：

对于一幅图像，可以表示成照射分量和反射分量的乘积，也就是：

$$m(x,y)=i(x,y)\cdot r(x,y)$$

其中， m 是图像， i 是为照度分量， r 是反射分量

为了在频域中使用高通滤波器，我们必须进行傅里叶变换，但是由于上式是一个乘积式，不能直接对照度分量和反射分量进行操作，因此对上式取对数

$$\ln(m(x,y))=\ln(i(x,y))+\ln(r(x,y))$$

然后对上式进行傅里叶转换

$$F\{\ln(m(x,y))\}=F\{\ln(i(x,y))\}+F\{\ln(r(x,y))\}$$

我们将 $F\{\ln(m(x,y))\}$ 定义为 $M(u,v)$

接下来对图像进行高通滤波，如此可以使图像的照明更均匀，高频分量增加且低频分量减少

$$N(u,v)=H(u,v)\cdot M(u,v)$$

其中 H 是高通滤波器

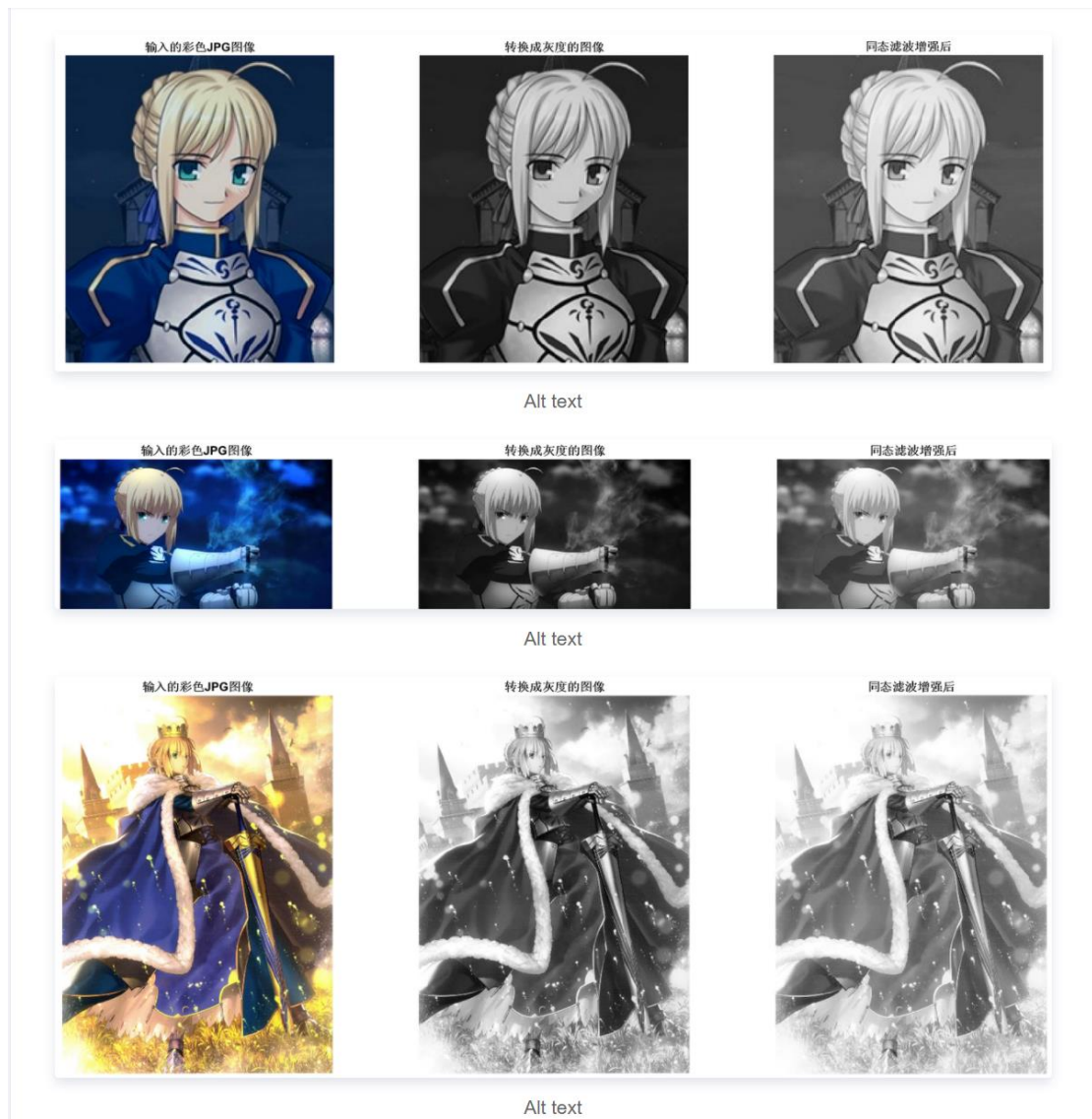
为了将图像从频域转回时域，我们对 N 做傅立叶逆转换

$$n(x,y)=F^{-1}\{N(x,y)\}$$

最后使用指数函数来复原我们一开始取的对数

$$m'(x,y)=\exp\{n(x,y)\}$$

给出处理结果



给出部分关键代码

```
I=double(I);
[M,N]=size(I);
rL=0.5;
rH=4.7;%可根据需要效果调整参数
c=2;
d0=10;
I1=log(I+1);%取对数
FI=fft2(I1);%傅里叶变换
n1=floor(M/2);
n2=floor(N/2);
H = ones(M, N);
for i=1:M
    for j=1:N
        D(i,j)=((i-n1).^2+(j-n2).^2);
        H(i,j)=(rH-rL).*(exp(c*(-D(i,j)./(d0^2))))+rL;%高斯同态滤波
    end
end
I2=ifft2(H.*FI);%傅里叶逆变换
I3=real(exp(I2));
subplot(1,3,3),imshow(I3,[]);title('同态滤波增强后');
```

学习总结：

通过这周的学习，了解了灰度线性变换、直方图均衡变换、同态滤波等方面的知识，相关理论的代码实现以及图像处理如上图所示。