

**数字信号处理实验报告**

基于Retinex理论的图像去雾算法研究

**姓 名**

**学 号**

**学 院 湖南理工学院**

**专 业 电子信息工程**

**班 级 电信**

**指导教师**

**编写日期 2020.05.30**

# 摘 要

图像去雾技术主要分为两类，一种是基于大气散射物理模型的去雾方法，另一种是基于图像增强的去雾方法。本文的研究基于后者，提出了一种基于Retinex理论图像增强去雾算法，对于传统的去雾算法得到的问题，将受到雾霾天气影响的图片用Retinex算法进行处理，获得清晰图像。根据Retinex算法先构建高斯环绕函数利用高斯环绕函数分别对图像的三个通道（R,G,B）进行滤波，再在对数域中的原始图像和光照分量进行相减得到反射分量作为输出结果图像。实验表明，该算法对雾霾图像处理过后有较好的增强效果，信噪比和信息熵明显提高，画面清晰。

**[关键词]:Retinex;图像去雾;**

第一章 绪 论

## 1.1开发背景

由于中国工业化发展迅速，工业生产和个人交通出行产生的废气不断增多，恶劣条件频繁增加，给人们生活产生严重影响。随着图像处理和计算机视觉技术的发展，图像去雾对视频监控，目标识别等有重要应用价值。目前的图像去雾技术，处理的结果图会出现失真较高，稳健性差，细节模糊等缺点。

雾霾天气条件下，空气能见度大幅度降低，从而导致成像传感器采集的图像严重降质，因此，一个稳定的室外监视系统应该有在任何天气下工作的能力，为了解决这一问题，图像去雾问题的研究由此产生。

## 1.2图像去雾技术的研究现状

图像去雾介于图像增强与图像修复之间，算法主要可以分成两类：一类是基于图像增强的去雾方法；另一类是基于大气散射规律建立图像物理退化模型，再根据模型进行去雾处理。这种模型利用先验知识，具有内在的优越性。实际条件下获取的往往是没有附加任何景深与大气前提信息的图像，因为已知信息量不够，所以图像去雾具有不确定性。

## 1.3 Retinex雾霾图像算法简介及原理

Retinex 理论是一种建立在科学实验和科学分析基础上、 基于人类视觉系统的图像增强理论。该算法与线性和非线性 变换、图像锐化等传统图像增强算法不同，这些算法只能对图 像的某类特征进行增强处理，而Retinex具有大动态范围压缩、 颜色恒常性、高色彩保真度等特点，适用于对受光照影响严重 的图像进行补偿，能够提高图像的清晰度，使图像的颜色更接 近原图像，图像增强后的视觉效果更佳。

Retinex的基础理论是物体颜色是由物体对长波、中波和短波光线的反射能力决定，而不是由反射光强度的绝对值决定。 物体的色彩不受光照非均匀性的影响，具有一致性， Retinex 是以色感一致性（颜色恒常性）为基础的，如图1：

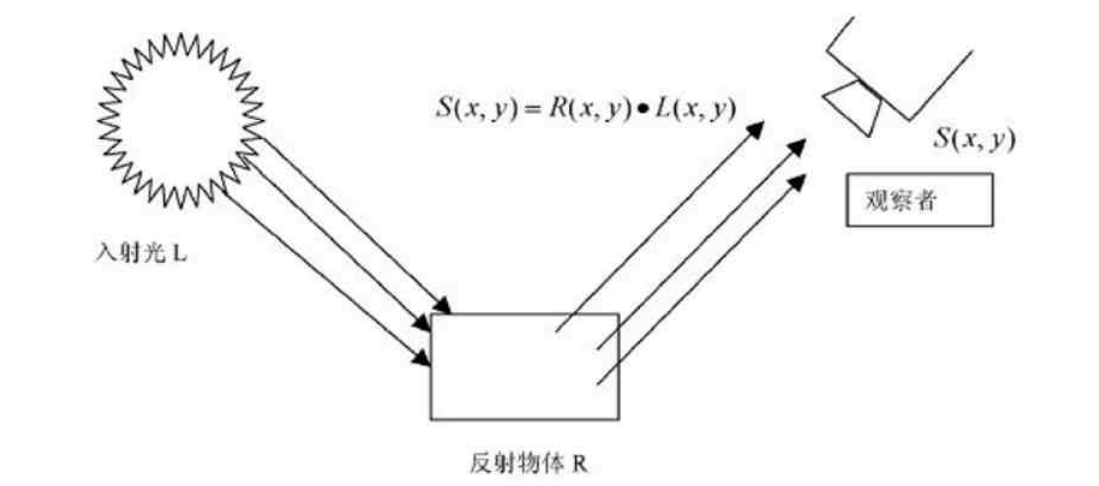


图 1 Retinex 原理示意图

实际上，Retinex理论就是通过图像来得到物体的反射性质，也就是设法去除（或者降低）入射光的影响从而得到物体原本该有的样子。但是具体该如何来估计并没有一个明确的答案，因此根据不同的估计方法，也就产生了各种各样的Retinex算法。

## 1.4图像增强处理流程

本论文设计基于Retinex，图像增强处理流程如下图2:

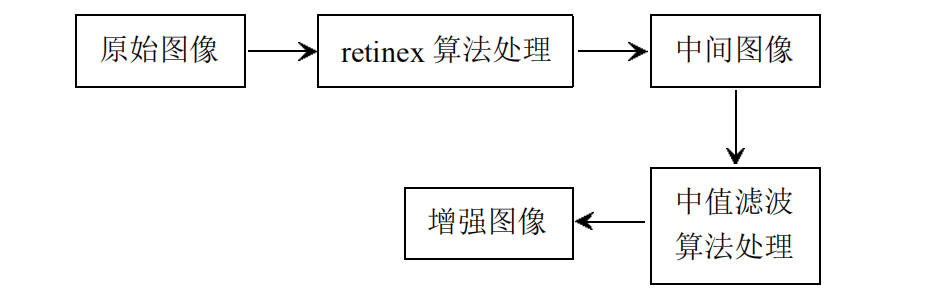


图 2图像增强处理流程

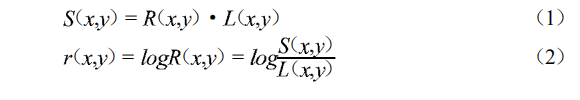
# 第二章 单尺度Retinex算法的研究

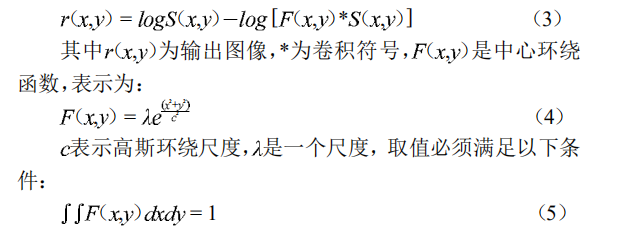
## 2.1引言

单尺度的Retinex算法（SSR, Single Scale Retinex）是最基础、最简单的一种Retinex算法，而且这个算法也给出了广义上Retinex算法的大致框架。本小节探究了单尺度的Retinex算法的实现以及使用代码实现并分析对比了处理前后的图像。

## 2.2算法研究

单尺度（Single-Scale Retinex，SSR）算法的公式为：





对上述5个公式解释如下：

式(1)：其中，R(x, y)表示了物体的反射性质，即图像内在属性，我们应该最大程度的保留；而L(x, y)表示入射光图像，决定了图像像素能达到的动态范围，我们应该尽量去除。一般，我们把照射图像假设估计为空间平滑图像，原始图像为S(x, y)，反射图像为R(x, y)，亮度图像为L(x, y)

式(3)，r(x, y)是输出图像，后面中括号里的运算是卷积运算。F(x, y)是中心环绕函数，表示为式(4)

式(4)中的C是高斯环绕尺度，λ是一个尺度，它的取值必须满足式(5)

上面的式中可以看出，SSR算法中的卷积是对入射图像的计算，其物理意义是通过计算像素点与周围区域在加权平均的作用下，估计图像中照度的变化，并将L(x,y)L(x,y)去除，只保留S(x,y)S(x,y)属性。

## 2.3分析与实现

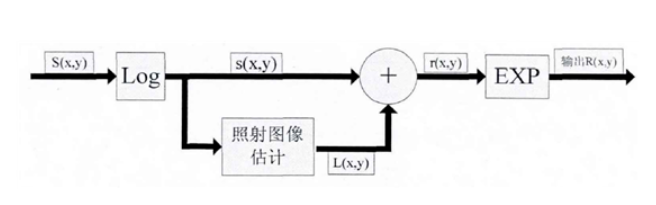
实现流程如下面4个步骤：

1.利用取对数的方法将照射光分量和反射光分量分离

2.我们把最终的反射图像假设地估计为空间平滑图像（其物理解释就是通过计算图像中像素点与周围区域中像素的加权平均来对图像中照度变化做估计，并将其去除，最后只保留图像中物体的反射属性），所以可以用高斯模板对原图像作卷积，即相当于对原图像作低通滤波，得到低通滤波后的图像，表示高斯滤波函数

3.在对数域中，用原图像减去低通滤波后的图像，得到高频增强的图像

4.对取反对数，得到增强后的图像



实现代码如下：

% 读取雾霾图片对其(R,G,B)三个通道分别处理

I = imread('a.png');

% 将图像R通道

R = I(:, :, 1);

% 使用单独的参数返回各个维度长度

[N1, M1] = size(R);

R0 = double(R);

Rlog = log(R0+1);

% 使用快速傅里叶变换算法返回矩阵的二维傅里叶变换

Rfft2 = fft2(R0);

sigma = 250;

% 返回大小为 [N1,M1] 的旋转对称高斯低通滤波器，标准差为 sigma

F = fspecial('gaussian', [N1,M1], sigma);

Efft = fft2(double(F));

% 图转换到频域 滤波器转换到频域 相乘之后换回去

% 不然直接在图上做卷积很难做

DR0 = Rfft2.\* Efft;

% 使用快速傅里叶变换算法返回矩阵的二维傅里叶逆变换

DR = ifft2(DR0);

DRlog = log(DR +1);

Rr = Rlog - DRlog;

EXPRr = exp(Rr);

MIN = min(min(EXPRr));

MAX = max(max(EXPRr));

% 归一化

EXPRr = (EXPRr - MIN)/(MAX - MIN);

% 限制对比度的自适应直方图均衡化 (CLAHE) 来变换值，从而增强图像的对比度

EXPRr = adapthisteq(EXPRr);

G = I(:, :, 2);

G0 = double(G);

Glog = log(G0+1);

Gfft2 = fft2(G0);

DG0 = Gfft2.\* Efft;

DG = ifft2(DG0);

DGlog = log(DG +1);

Gg = Glog - DGlog;

EXPGg = exp(Gg);

MIN = min(min(EXPGg));

MAX = max(max(EXPGg));

EXPGg = (EXPGg - MIN)/(MAX - MIN);

EXPGg = adapthisteq(EXPGg);

B = I(:, :, 3);

B0 = double(B);

Blog = log(B0+1);

Bfft2 = fft2(B0);

DB0 = Bfft2.\* Efft;

DB = ifft2(DB0);

DBlog = log(DB+1);

Bb = Blog - DBlog;

EXPBb = exp(Bb);

MIN = min(min(EXPBb));

MAX = max(max(EXPBb));

EXPBb = (EXPBb - MIN)/(MAX - MIN);

EXPBb = adapthisteq(EXPBb);

result = cat(3, EXPRr, EXPGg, EXPBb);

subplot(121), imshow(I);

subplot(122), imshow(result);

## 2.4结果对比

我们任意找到一张带有雾霾的图片，命名为a.png，使用上述代码载入图片，

处理前图片效果如下：



图 3

处理后图片显示效果如下：



图 4

由图 3、图 4 可看出图像在增强处理之后对比度、亮度和画面细节都有着显著的提升，有效信息显著增多。

# 第三章 多尺度Retinex算法的研究

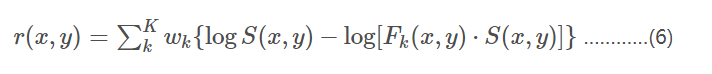
## 3.1引言

多尺度 Retinex（Multi⁃scale Retinex，MSR）算法是基于图像增强的去雾方式，即基于非模型的去雾方法，它用传统的图像增强方法作为基本的图像去雾处理技术，不考虑图像景深信息、图像退化原因，是一种常用的图像去雾方法。MSR 算法是对 SSR 算法的拓展和延伸，由入射光的反射分量经过高斯函数与原先图像做卷积而得到。此方法去除外界光的入射分量的影响，获得反射分量，含有所需原始图像的细节信息。因此，MSR 既凸显图像的细节，又突出图像的色调。本小节探究了多尺度 Retinex的实现，并通过代码验证了对雾霾天气图片的处理程度，也对比解决了上面单尺度 Retinex的不足。

## 3.2算法研究

MSR是在SSR基础上发展来的，优点是可以同时保持图像高保真度与对图像的动态范围进行压缩的同时，MSR也可实现色彩增强、颜色恒常性、局部动态范围压缩、全局动态范围压缩，也可以用于X光图像增强。

MSR计算公式如下：



式中，K是高斯中心环绕函数的个数。当K=1时，MSR退化为SSR。  
通常来讲，为了保证兼有SSR高、中、低三个尺度的优点来考虑，K取值通常为3，且有：



此外，实验表明，ci分别取15, 80, 200可以得到较好效果

一般的Retinex算法对光照图像估计时，都会假设初始光照图像是缓慢变化的，即光照图像是平滑的。但实际并非如此，亮度相差很大区域的边缘处，图像光照变化并不平滑。所以在这种情况下，Retinex增强算法在亮度差异大区域的增强图像会产生光晕。

另外MSR常见的缺点还有边缘锐化不足，阴影边界突兀，部分颜色发生扭曲，纹理不清晰，高光区域细节没有得到明显改善，对高光区域敏感度小等。

## 3.3分析与实现

在MSR算法的增强过程中，图像可能会因为增加了噪声而导致图像中局部区域的色彩失真，使得物体真正的颜色效果不能很好地显现出来，从而影响了整体的视觉观感。为了改进这方面的不足，一般情况下会使用带色彩恢复因子C的多尺度算法（MSRCR）来解决（Rehman等提出）。带色彩恢复因子C的多尺度算法是在多个固定尺度的基础上考虑色彩不失真恢复的结果，在多尺度Retinex算法过程中，通过引入一个色彩因子C来弥补由于图像局部区域对比度增强而导致的图像颜色失真的缺陷,带色彩恢复的多尺度Retinex算法通过色彩恢复因子C这个系数来调整原始图像中3个颜色通道之间的比例关系，从而把相对有点暗的区域的信息凸显出来，以达到消除图像色彩失真缺陷的目的。处理后的图像局域对比度得以提高，而且其亮度与真实的场景很相似，图像在人们的视觉感知下显得更为逼真。因此，MSRCR算法会具有比较好的颜色再现性、亮度恒常性与动态范围压缩等特性。

实现代码如下：

I = imread('a.png');

R = I(:, :, 1);

G = I(:, :, 2);

B = I(:, :, 3);

R0 = double(R);

G0 = double(G);

B0 = double(B);

[N1, M1] = size(R);

Rlog = log(R0+1);

Rfft2 = fft2(R0);

sigma1 = 128;

F1 = fspecial('gaussian', [N1,M1], sigma1);

Efft1 = fft2(double(F1));

DR0 = Rfft2.\* Efft1;

DR = ifft2(DR0);

DRlog = log(DR +1);

Rr1 = Rlog - DRlog;

sigma2 = 256;

F2 = fspecial('gaussian', [N1,M1], sigma2);

Efft2 = fft2(double(F2));

DR0 = Rfft2.\* Efft2;

DR = ifft2(DR0);

DRlog = log(DR +1);

Rr2 = Rlog - DRlog;

sigma3 = 512;

F3 = fspecial('gaussian', [N1,M1], sigma3);

Efft3 = fft2(double(F3));

DR0 = Rfft2.\* Efft3;

DR = ifft2(DR0);

DRlog = log(DR +1);

Rr3 = Rlog - DRlog;

Rr = (Rr1 + Rr2 +Rr3)/3;

a = 125;

II = imadd(R0, G0);

II = imadd(II, B0);

Ir = immultiply(R0, a);

C = imdivide(Ir, II);

C = log(C+1);

Rr = immultiply(C, Rr);

EXPRr = exp(Rr);

MIN = min(min(EXPRr));

MAX = max(max(EXPRr));

EXPRr = (EXPRr - MIN)/(MAX - MIN);

EXPRr = adapthisteq(EXPRr);

Glog = log(G0+1);

Gfft2 = fft2(G0);

DG0 = Gfft2.\* Efft1;

DG = ifft2(DG0);

DGlog = log(DG +1);

Gg1 = Glog - DGlog;

DG0 = Gfft2.\* Efft2;

DG = ifft2(DG0);

DGlog = log(DG +1);

Gg2 = Glog - DGlog;

DG0 = Gfft2.\* Efft3;

DG = ifft2(DG0);

DGlog = log(DG +1);

Gg3 = Glog - DGlog;

Gg = (Gg1 + Gg2 +Gg3)/3;

Ig = immultiply(G0, a);

C = imdivide(Ig, II);

C = log(C+1);

Gg = immultiply(C, Gg);

EXPGg = exp(Gg);

MIN = min(min(EXPGg));

MAX = max(max(EXPGg));

EXPGg = (EXPGg - MIN)/(MAX - MIN);

EXPGg = adapthisteq(EXPGg);

Blog = log(B0+1);

Bfft2 = fft2(B0);

DB0 = Bfft2.\* Efft1;

DB = ifft2(DB0);

DBlog = log(DB +1);

Bb1 = Blog - DBlog;

DB0 = Bfft2.\* Efft2;

DB = ifft2(DB0);

DBlog = log(DB +1);

Bb2 = Blog - DBlog;

DB0 = Bfft2.\* Efft3;

DB = ifft2(DB0);

DBlog = log(DB +1);

Bb3 = Blog - DBlog;

Bb = (Bb1 + Bb2 +Bb3)/3;

Ib = immultiply(B0, a);

C = imdivide(Ib, II);

C = log(C+1);

Bb = immultiply(C, Bb);

EXPBb = exp(Bb);

MIN = min(min(EXPBb));

MAX = max(max(EXPBb));

EXPBb = (EXPBb - MIN)/(MAX - MIN);

EXPBb = adapthisteq(EXPBb);

result = cat(3, EXPRr, EXPGg, EXPBb);

subplot(121), imshow(I);

subplot(122), imshow(result);

## 3.4结果对比

我们任意找到一张带有雾霾的图片，命名为a.png，使用上述代码载入图片，

处理后图片效果如图6：



图 6

从上面的结果来看，MSRCR比SSR的图像增强效果更佳，色彩也更逼真。

# 总结与展望

本文分别单尺度和多尺度 Retinex 的雾霾图像去雾增强算法。受雾霾天气影响的失真图像用 Retinex 算法进行处理，再使用中值滤波对图像进行去噪处理。Matlab 仿真实验表明该算法对受到雾霾天气影响的图像有着较好的增强效果，在降低噪声的情况下有较好的画面观感。但是总的来说还是有很多东西不达标，上面实现的MSRCR并没有为去雾做过多的特别设计，所以这也不完全算是一种很完善的去雾算法。如果实验更多的有雾图片，就会发现它的一些弱点，例如对天空部分的处理效果还不尽如人意，还有某些色彩比较深的地方会变得更暗而导致辨识度下降。不过我会在课题结束后继续研究弥补这些缺陷。

# 参考文献

1. 韩祥辉, 王好贤. 恶劣天气环境下退化图像的清晰化算法 [J]. 计算机工程, 2011.37(22):207-208．
2. 王萍,张春,罗颖昕.一种雾天图像低对比度增强的快速算法[J].计算机应用, 2006.26(1):152-154.
3. 李学明.基于 Retinex 理论的图像增强算法[J].计算机应用研究,2005.22(2): 235-237.