照片去雾算法论文

一、**研究背景:**

一方面随着大气污染的日益严重，设法改善自己主动获取的图像质量其意义不言而喻。还有一方面，随着数码设备的普及，消费类电子产品的市场也催生出很多新的需求，当中人们对所拍照片质量的修正和优化就是一个显而易见的需求。所以说研究出来一种去除雾霾的算法迫在眉睫。于是乎我和我的小组成员一起研究了照片去雾的算法，以下是研究成果.

**二、暗通道先验理论**

在绝大多数非天空的局部区域里，某些像素总会有至少一个颜色通道具有很低的值，该区域光强度的最小值是个很小的数。以下是暗通道的一个数学定义，对于任意的输入图像J,其暗通道可以用下式表达:

J[dark](x) = min[min Jc(y)](c<-(r,g,b),y<-o(x))

上式中Jc表示彩色图像的每个通道，O(x)表示以像素x为中心的一个窗口。上式的意义用代码表达也很简单，首先求出每个像素RGB中分量的最小值，滤波的半径由窗口大小决定，一般有windowsize = 2\*radius+1

暗通道先验理论指出:

Jdark --> 0

实际生活中造成暗原色低通道的原因很多，例如：汽车，建筑物和城市中玻璃窗户的阴影，或者是树叶，树与岩石等自然景观的投影；色彩鲜艳的物体或表面，在RGB的三个通道中有些通道值很低（比如绿色草地，数目等植物），颜色较暗的物质或者表面，例如灰色的树干。石头以及路面。总之，自然景物中到处都是阴影或者彩色，这些景物图像的暗原色总是表现出较为灰暗的状态。

求图像暗通道的代码实现:

for (Y = 0, DarkPt = DarkChannel; Y < Height; Y++) {

ImgPt = Scan0 + Y \* Stride;

for (X = 0; X < Width; X++) {

Min = \*ImgPt;

if (Min > \*(ImgPt + 1))

Min = \*(ImgPt + 1);

if (Min > \*(ImgPt + 2))

Min = \*(ImgPt + 2);

\*DarkPt = Min; ImgPt += 3; DarkPt++;

}

}

MinFilter(DarkChannel, Width, Height, Radius);

下面是一些图片暗通道案例:

原图:



暗通道图:



原图:



暗通道:



当我们有了这些先验，接着就可以进行一些数学方面的推导，从而实现问题的最终解决办法

1. **雾图形成模型**

首先，在计算机视觉与计算机图形中，下述方程描述的雾图模型被广泛使用:

I(x) = J(x)t(x)+A[1-t(x)]

其中，I(x)就是现在已有的图像，及待去雾的图像，J(x)是要恢复的无雾图像，参数A是大气光成分，t(x)为透射率。现在的已知条件是I(x),求目标值J(x).根据基本代数只是，可知这是一个有无数解的方程。只有在一些先验信息基础上才能求出定解。

将上式稍作处理，变形为下式:

Ic(x)/Ac = t(x){Jc(x)/Ac}+1-t(x)

如上所述，上标c表示R,G,B三个通道的意思。

首先假设在一个窗口内透射率t(x)为常数，将其定义为t(x),并且A值已经确定，然后对上式两边求最小值，得到下式:

Min [min Ic/Ac] = t(x)min[min Jc(y)/Ac]+1-t(x)

上式中，J是待求的无雾图像，根据前述的暗原色先验理论有:

Jdark(x) = min[min Jc(y)] =0

由此，可以推到出:

Min[min Jc(y)/Ac] = 0;

把上式的结论带回原式，得到:

t(x) = 1 - min[min Ic(y)/Ac]

这就是透射率t(x)的值。

在现实生活中，即使是蓝天白云，空气中也存在的一些颗粒，因此，看远处的物体还是能感觉到雾的影响。此外，雾的存在让人感觉到景深的存在，因此，有必要在去雾的时候保存一定程度的雾。这可以在上式中引入一个[0,1],之间的因子实现，所以上式可以修改为

T(x) = 1 - w\*min[min Ic/Ac]

上述推论中都是假设全球大气光的A值已知，在实际中，可以借助暗通道从有雾图中获取该值。具体操作步骤大致为:先从暗通道中按照亮度大小提取最亮的前0.1%像素。然后，在原始有雾图像I中寻找对应位置上具有最高亮度点的值，并以此作为A的值，至此，我们就可以进行无雾图像的恢复了。

考虑到当透射率t很小时，会导致J的值偏大，从而使图像整体向白场过度，因此，一般可以设置一个阀值t0，当t值小于t0时，令t = t0.后续展示均采用t0=0.1为标准计算。因此，最终图像恢复公式如下：

J(x) = {I(x)-A}/{max[t(x),t0]}+A

**四、最终效果图:**

 

有雾图 去雾图

**五、部分代码**

Main.m:

clear

clc

close all

kenlRatio = 0.0001;

minAtomsLight = 240;

% image\_name = 'test images\21.bmp';

image\_name = 'G:/C.jpg';

img=imread(image\_name);

figure,imshow(uint8(img)), title('src');

sz=size(img);

w=sz(2);

h=sz(1);

dc = zeros(h,w);

for y=1:h

for x=1:w

dc(y,x) = min(img(y,x,:));

end

end

figure,imshow(uint8(dc)), title('Min(R,G,B)');

krnlsz = floor(max([3, w\*kenlRatio, h\*kenlRatio]));

dc2 = minfilt2(dc, [krnlsz,krnlsz]);

dc2(h,w)=0;

figure,imshow(uint8(dc2)), title('After filter ');

t = 255 - dc2;

figure,imshow(uint8(t)),title('t');

t\_d=double(t)/255;

sum(sum(t\_d))/(h\*w);

A = min([minAtomsLight, max(max(dc2))]);

J = zeros(h,w,3);

img\_d = double(img);

J(:,:,1) = (img\_d(:,:,1) - (1-t\_d)\*A)./t\_d;

J(:,:,2) = (img\_d(:,:,2) - (1-t\_d)\*A)./t\_d;

J(:,:,3) = (img\_d(:,:,3) - (1-t\_d)\*A)./t\_d;

figure,imshow(uint8(J)), title('J');

% figure,imshow(rgb2gray(uint8(abs(J-img\_d)))), title('J-img\_d');

% a = sum(sum(rgb2gray(uint8(abs(J-img\_d))))) / (h\*w)

% return;

%----------------------------------

r = krnlsz\*4;

eps = 10^-6;

% filtered = guidedfilter\_color(double(img)/255, t\_d, r, eps);

filtered = guidedfilter(double(rgb2gray(img))/255, t\_d, r, eps);

t\_d = filtered;

figure,imshow(t\_d,[]),title('filtered t');

J(:,:,1) = (img\_d(:,:,1) - (1-t\_d)\*A)./t\_d;

J(:,:,2) = (img\_d(:,:,2) - (1-t\_d)\*A)./t\_d;

J(:,:,3) = (img\_d(:,:,3) - (1-t\_d)\*A)./t\_d;

%

img\_d(1,3,1)

imwrite(uint8(J),'G:/B.jpg');

figure,imshow(uint8(J)), title('J\_guild\_filter');

%----------------------------------

%imwrite(uint8(J), ['\_', image\_name])

指导教师:杜振龙

职称：教授

研究人:孙宏伟，余峰

职称:学生

单位：南京工业大学计算机科学与技术学院