

## 图像增强实验

实验内容

实验参考

实验核心代码

实验结果

实验参考

# 图像增强实验

## 实验内容

天气对图像的质量有很大的影响，请利用图像分析的相关知识，实现基于暗通道先验的图像去雾算法，对有雾霾的图像进行增强。

## 实验参考

### □ 算法流程总结：

#### ■ 计算暗通道先验

✓ 
$$J^{dark}(x) = \min_{y \in \Omega(x)} \left( \min_{c \in \{r, g, b\}} J^c(y) \right)$$

#### ■ 利用先验，计算透射率和大气光

✓ 
$$t(x) = 1 - \omega \min_{y \in \Omega(x)} \left( \min_{c \in \{r, g, b\}} \frac{I^c(y)}{A^c} \right)$$

✓ 选取暗通道中灰度（intensity）最大的0.1%像素，在这些像素中再选择最亮的像素点作为大气光  $A$

#### ■ 使用导向滤波对透射率图进行滤波

#### ■ 根据公式，计算去雾图像

✓ 
$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A$$

实验原理参考

## 实验核心代码

- 运行平台: Windows + Matlab(9.8.0.1323502 (R2020a))
- 实验main.m文件及简要说明

```
clear; close all; clc;
# 选择待去雾文件
[f, p] = uigetfile({'*.jpg'}, 'Open');
if f
    I = imread([p f]);
end
%% 参数设置
```

```

t0 = 0.1;w = 0.85;
%% 调用去雾函数
[I,darkChannel,t,J] = dehaze(I,t0,w);
%% 计算结果显示
figure;
subplot(2,2,1);imshow(I);title('原图像');
subplot(2,2,2);imshow(darkChannel);title('暗通道');
subplot(2,2,3);imshow(t);title('透射率图');
subplot(2,2,4);imshow(J);title('目标图');
%% 目标图保存
imwrite(J,"output/de" + f);

```

- dehaze.m文件及说明

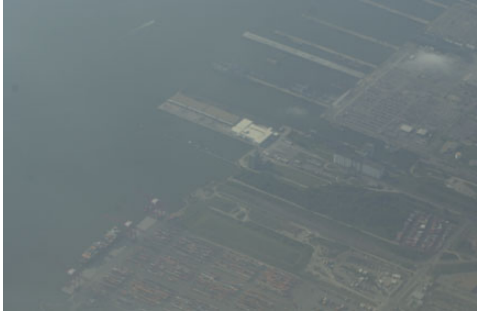

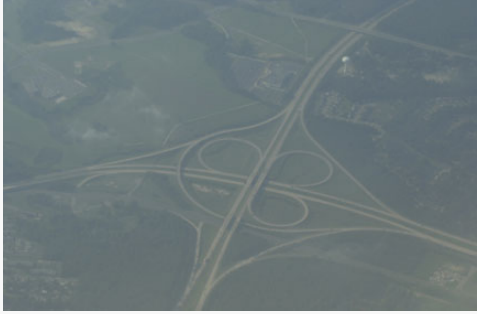





```

function [RGB,darkChannel,t,J] = dehaze(RGB,t0,w)
    %% 读取图片并分离channel
    RGBD = double(RGB);
    R = RGBD(:, :, 1);
    G = RGBD(:, :, 2);
    B = RGBD(:, :, 3);
    %% 求解暗通道
    minChannel = min(min(R,G),B);
    kernel = ones(15);
    darkChannel = imerode(minChannel, kernel);
    darkChannel = uint8(darkChannel);
    %% 估计大气光
    [t, ~] = sort(darkChannel(:), 'descend');
    p = 0.001;
    n = floor(length(t) * p);
    A = zeros(1,3);
    dark_bright = darkChannel>=t(n);
    for i = 1:3
        pic = RGBD(:, :, i);
        A(i) = max(pic(dark_bright));
    end
    %% t(x)
    tR = double(R)./double(A(1));
    tG = double(G)./double(A(2));
    tB = double(B)./double(A(3));
    t_hat = min(min(tR,tG),tB);
    t = 1 - w * t_hat;
    t = max(t,t0);
    %% guide filter
    t = guidedfilter(RGBD,t,5,0.0001);
    %% dehaze
    J(:, :, 1) = (R - A(1))./t + A(1);
    J(:, :, 2) = (G - A(2))./t + A(2);
    J(:, :, 3) = (B - A(3))./t + A(3);
    J = uint8(J);
end

```

- 导向滤波实现见guidedfilter.m文件。

## 实验结果

| 图片id  | 去雾前   | 去雾后  |
|-------|---|--|
| haze1 |    |    |
| haze2 |    |    |
| haze3 |   |   |
| haze4 |  |  |
| haze5 |  |  |

## 实验参考

- Single Image Haze Removal Using Dark Channel Prior\_cvpr\_09

