

数字图像处理报告七：图像增强

姓名：鲁国锐

学号：17020021031

专业：电子信息科学与技术

2020 年 4 月 22 日

目录

1	题目描述	2
2	RGB 空间和 HSI 空间图像增强对比	2
2.1	RGB 空间图像增强方法	2
2.1.1	实验思路	2
2.1.2	实验代码	2
2.1.3	实验结果	3
2.2	HSI 空间图像增强方法	3
2.2.1	实验思路	4
2.2.2	实验代码	4
2.2.3	实验结果	4
3	水下图像增强	5
3.1	概述	5
3.2	暗通道先验算法	5
3.2.1	原理：暗通道先验	5
3.2.2	原理：模型	5
3.2.3	算法复现	8
3.2.4	实验结果	11
4	总结	13

1 题目描述

1. 在 *RGB* 和 *HSI* 空间做图像增强的区别是什么？
2. 如何增强水下图像的色彩？（需要查文献）。

2 *RGB* 空间和 *HSI* 空间图像增强对比

何为“彩色空间”？教材上给出的解释为 [2] “彩色模型（也称为彩色空间或彩色系统）的目的是，在某些标准下用通常可以接受的方式方便地对彩色加以说明”。其实就是人们对于彩色的一种描述方式。比如本文要讨论的 *RGB* 模型和 *HSI* 模型：前者用红、绿、蓝三种原色作为三个分量图像表示一副彩色图像，主要面向硬件；后者则用色调、饱和度、亮度作为分量描述图像，主要面向人类。简单来说，不同的彩色空间的差别就在于所选择的描述标准不同，根据实际的需要，选择最为适合的标准。

由上述定义我们便可以得到一个对两种方法区别的大体印象：*RGB* 空间图像增强方法基于的是对红、绿、蓝三种原色强度的操作；而 *HSI* 空间基于的是对色调、饱和度、亮度三个指标的操作。下面将进一步对两类方法展开讨论。

2.1 *RGB* 空间图像增强方法

RGB 空间的图像增强方法主要参考教材 [2] 第 3 章中的方法，大体可以分为以下几类：

1. 基于灰度变换函数的方法：通过将整幅图像的像素经过一个或多个函数映射达到增强的目的，典型的方法有对比度拉伸变换和伽马变换等；
2. 基于直方图处理的方法：通过对三个通道中像素的统计特性进行操作来达到增强的目的，典型的方法有直方图均衡和直方图匹配等；
3. 基于锐化空间滤波器的方法：通过对像素的直接操作（主要是微分或差分）来提取图像的边缘特性，将其加到原图像上以达到增强的目的，典型的方法有拉普拉斯算子和 *Sobel* 算子等。

上述方法最大的缺点是：由于三个通道的像素值分布往往不一样，所以在进行处理后会导导致图像各分量的比例发生变化，使得颜色偏离了实际的样子。

注意这里并未将第 4 章中的频率域滤波方法加入进来，因为个人认为当将一副图像变换到频率域之后，对图像的描述方式也相应地变了（由红、绿、蓝三原色变为了三个通道的频率的相位和幅度），超出了 *RGB* 彩色空间的表示范围。

2.1.1 实验思路

分别对 *R*、*G*、*B* 三个通道进行直方图均衡，观察实验结果，尤其是色调是否产生变化。

2.1.2 实验代码

Listing 1: 分别对三个通道进行直方图均衡操作

```
1 %函数 histeq () 进行直方图均衡化处理
2 I=imread('3153987-fd1e1de03f59e3a8.jpeg');
3 J=histeq(I); %直方图均衡化
4
5 J(:, :, 1) = histeq( I(:, :, 1) );
6 J(:, :, 2) = histeq( I(:, :, 2) );
```

```

7 J(:, :, 3) = histeq( I(:, :, 3) );
8 % J = histeq(I);
9
10 figure ,
11 subplot(221),imshow(I);
12 title('原图')
13 subplot(222),imshow(J);
14 title('均衡化后')
15 subplot(223),imhist(I,64);
16 title('原图像直方图');
17 subplot(224),imhist(J,64);
18 title('均衡化后的直方图');
19
20 imwrite(J, 'C:\Users\Asus-\Desktop\数字图像\codes\contrast_stretching\test.png');

```

2.1.3 实验结果



(a) 原图

(b) 经直方图均衡处理后的结果

图 1: 对图像进行直方图均衡操作后可能会导致颜色偏离原始色彩¹

如图1(a)和图1(b)所示，分别对 R 、 G 、 B 三个通道进行直方图均衡后的结果，其部分区域的色调发生了明显改变，如云的颜色，有偏蓝变成了金黄。

另外需要注意的一点是，并不是对所有的图像进行上述操作都会导致色调的改变，此次实验中尝试了若干幅图像，目前只有图1(a)这一张能非常明显地看到色调被改变了。

2.2 HSI 空间图像增强方法

在 HSI 彩色空间中进行图像增强最大的好处是 [3]：“各颜色分量是相互独立的，这样就可以消除各颜色分量之间的相关性，所以保持色调分量 ($hueH$) 不变，只需调整饱和度分量 ($saturabilityS$) 和亮度分量 ($intensityI$) 就可达到增强图像的目的。”

从上面的话可以得出，在 HSI 空间对图像进行增强，只要不去改动 H 分量，理论上就不会对结果的色调产生影响。

¹原图来源于网络：<https://www.jianshu.com/p/5a8d12d6c649>

2.2.1 实验思路

将一副 *RGB* 图像变换到 *HSI* 空间，只对其强度分量进行直方图均衡，再变换回 *RGB* 空间，观察实验结果。

2.2.2 实验代码

Listing 2:] 在 *HST* 彩色空间对强度分量做直方图均衡¹

```
1 % 读取图像显示原图
2 img = imread('3153987-fd1e1de03f59e3a8.jpeg');
3 figure(1)
4 imshow(img)
5 title('原图')
6
7 img_hsi = rgb2hsi(img); % 转到HSI空间
8 img_hsi(:, :, 3) = histeq(img_hsi(:, :, 3)); % 对强度进行直方图均衡
9 img = hsi2rgb(img_hsi);
10 figure(3)
11 imshow(img)
12 title('对强度进行直方图均衡后的结果')
13 imwrite(img, 'C:\Users\Asus\Desktop\数字图像\report\07\enhanced_res_using_hsi.png')
```

2.2.3 实验结果



(a) 原图



(b) 只针对强度分量进行直方图均衡后的结果

图 2: 在 *HSI* 空间对强度分量进行直方图均衡可以有效缓解色调改变的问题

如图2(a)和图2(b)所示，在 *HSI* 空间对强度分量进行直方图均衡后，其结果对比度有明显提高，同时色调也能与原图保持一致。

¹*MATLAB* 没有现成的 *RGB* 与 *HSI* 空间相互转换的函数，代码中 *rgb2hsi* 和 *hsi2rgb* 这两个函数取自本课程实验五提供的资料，即冈萨雷斯所著的《数字图像处理（*MATLAB*）版》中提供的的代码。

3 水下图像增强

3.1 概述

海洋中拥有着丰富的资源，具有巨大的开发潜力。水下光学图像作为传输海洋信息的主要载体，对探索与开发海洋起着至关重要的作用 [5]。然而，由于海水的良导体特性，再加之水体中微粒杂志的散射作用，导致光在水中传播时会有强烈的衰减，使得水下图像往往会出现失真、模糊、对比度低等问题 [6]。因此，如何有效地提升水下光学图像的质量是一个亟待解决的问题。

在 [4] 中，作者总结了水下图像增强的五类主要方法，分别为：

1. 基于直方图的水下图像增强算法；
2. 基于 *Retinex* 的水下图像增强算法；
3. 基于滤波和信号处理的水下图像增强算法；
4. 基于图像融合的水下图像增强算法；
5. 基于 CNN 的水下图像增强算法。

另外，在多篇关于水下图像增强的文献中 [4] [5] [6]，都引用了同一个算法——由何恺明等人提出的暗通道先验算法 [1]。可见该算法对于水下图像增强有着重要意义，本次报告将在接下来的一节中着重介绍这一算法。

3.2 暗通道先验算法

3.2.1 原理：暗通道先验

论文提出的算法基于一个非常简单事实：在一张没有雾的图像中，除了类似于天空这样的区域，其余地方像素的 *R*、*G*、*B* 三个通道一般会有至少一个通道的强度非常小。为了说明这样一个事实，作者提出了“暗通道”这一概念，它的定义由公式1给出：

$$J^{dark}(\mathbf{x}) = \min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} J^c(\mathbf{y}) \right) \quad (1)$$

公式1的意思是对图像中的每一个像素，取出它 *R*、*G*、*B* 三个通道中的最小值，得到一个与原图长宽一样的二维矩阵，再对这个二维矩阵进行一次最小值滤波，得到的即为暗通道。论文里对为何还要在进行一次最小值滤波的原因没有详细说明，个人猜测应当是为了消除部分类似于天空这样的区域的干扰，因为天空区域往往没有暗通道，其三个分量的强度都比较大。

用暗通道来表示论文提出的事实，可以写成：对于一张没有雾的自然图像，有：

$$J^{dark} \rightarrow 0 \quad (2)$$

作者将这个称为“暗通道先验”。

为了证明这一事实的普遍性，作者还随机选取了 5000 张图片，裁剪掉天空区域，统计了剩余部分暗通道中的像素值（如图3所示），结果显示，大约有 85% 的像素值在 16 以下，大约 75% 的像素值为 0。

3.2.2 原理：模型

在计算机视觉和计算机图形学中，雾图像可以被建模为：

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})) \quad (3)$$

其中 \mathbf{I} 是待去雾的图像； \mathbf{J} 是无雾的图像； $t(x)$ 是媒质传播函数，用以描述没有被散射的、到达相机上的光的比例； \mathbf{A} 是全局大气光照。去雾的任务就是要求出 $t(x)$ 和 \mathbf{A} 以解出 \mathbf{J} 。

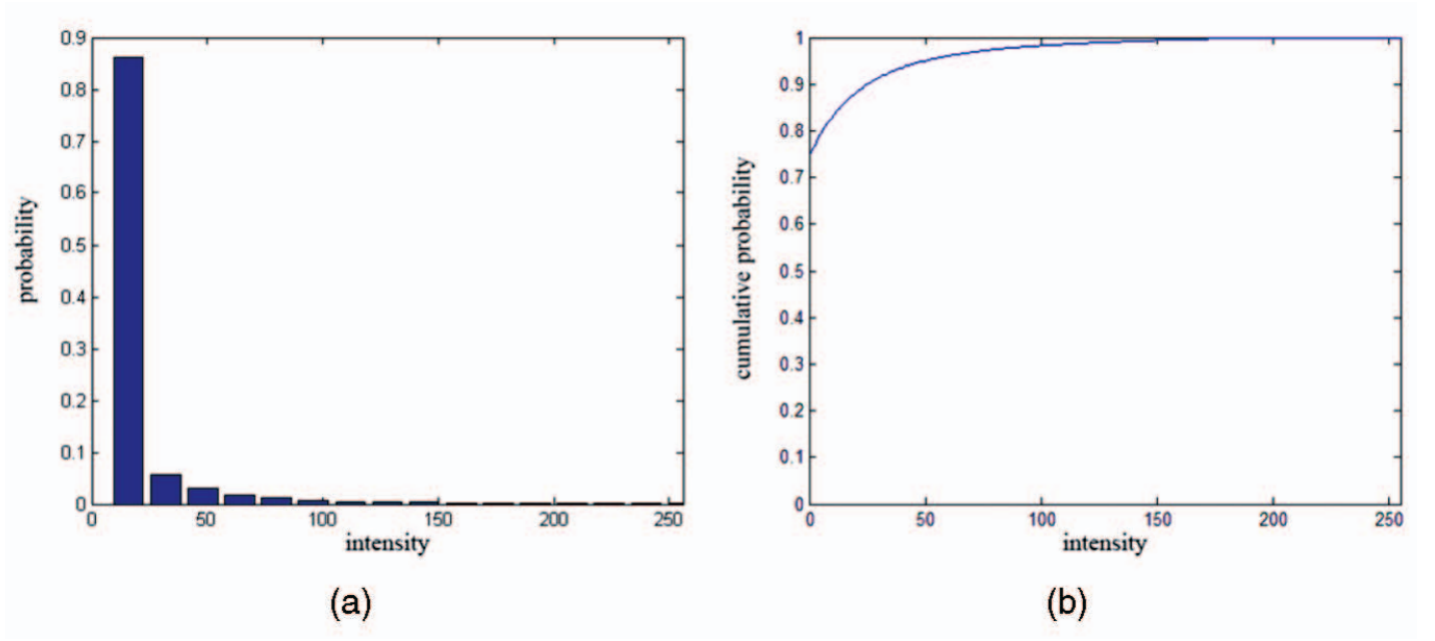


图 3: 暗通道像素值统计结果

大气光照 A 的求解

在过去的工作中，一般用雾图像中最不透明区域的颜色来表示 A 。当太阳光可以被忽略时，大气光就是图像中唯一的光源。此时有：

$$J(\mathbf{x}) = R(\mathbf{x}) A \quad (4)$$

将4式代入3式，可得：

$$I(\mathbf{x}) = R(\mathbf{x}) A t(\mathbf{x}) + (1 - t(\mathbf{x})) A \quad (5)$$

又因为

$$t(\mathbf{x}) = e^{-\beta d(\mathbf{x})} \quad (6)$$

这里 β 表示大气光的散射系数， d 表示景深。所以当景深趋近于无穷大的点存在于图像中时， $\lim_{d \rightarrow \infty} t(\mathbf{x})$ 趋于 0。代入5式，可得：

$$\lim_{d \rightarrow \infty} I(\mathbf{x}) = R(\mathbf{x}) A \lim_{d \rightarrow \infty} t(\mathbf{x}) + \left(1 - \lim_{d \rightarrow \infty} t(\mathbf{x})\right) A \quad (7)$$

$$= A \quad (8)$$

作者从这里直接得出了此时图中最亮的点就是雾图像中最不透明区域的颜色，并且它的值近似等于 A 。也就是说， $\lim_{d \rightarrow \infty} I(\mathbf{x})$ 所表示的就是图中最亮的点。这一结论对我而言似乎并不是显而易见的，暂时还没有想明白为什么。

接下来，文中又提到，太阳光往往是不能忽略的，所以4式需要被修正为：

$$J(\mathbf{x}) = R(\mathbf{x}) (A + S) \quad (9)$$

代入3式得：

$$I(\mathbf{x}) = R(\mathbf{x}) S t(\mathbf{x}) + R(\mathbf{x}) A t(\mathbf{x}) + (1 - t(\mathbf{x})) A \quad (10)$$

作者认为此时图像中最亮的点比大气光更亮，不能直接用来求 A 。同样，这里关于 $\lim_{d \rightarrow \infty} I(\mathbf{x})$ 和最亮的点之间的关系我并没有想明白，也暂时略过。

于是作者提出可以用暗通道来估计大气光。算法如下：

1. 对暗通道中的点按强度降序排序，选取前 0.1% 的点；
2. 获取这些点在暗通道中的坐标，也就是不透明区域在原图中的坐标；
3. 在原图中遍历这些坐标，找出最亮的点，即得 A 。

按照作者所说的，暗通道中强度排在前 0.1% 的点往往表示的就是雾图像中最不透明的区域，按大气光的定义，只需在这样的区域中找到最亮的点就得到 A 了。

至于为什么暗通道中强度排在前 0.1% 的点能够表示雾图像中最不透明的区域，个人猜测是因为对暗通道做了最小值滤波，将太阳光的影响给滤除了。正如 3.2.1 中所猜测的那样，最小值滤波是为了消除类似于天空这样的区域的干扰。现在看来，所谓“天空这样的区域的干扰”应该指的就是太阳光的干扰。

媒质传递函数 $t(\mathbf{x})$ 的求解

对 3 式两边同时除以 A ，得：

$$\frac{I^c(\mathbf{x})}{A^c} = t(\mathbf{x}) \frac{J^c(\mathbf{x})}{A^c} + 1 - t(\mathbf{x}) \quad (11)$$

对 11 两边同时求暗通道，得：

$$\min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \frac{I^c(\mathbf{y})}{A^c} \right) = \tilde{t}(\mathbf{x}) \min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \frac{J^c(\mathbf{y})}{A^c} \right) + 1 - \tilde{t}(\mathbf{x}) \quad (12)$$

由于 $\tilde{t}(\mathbf{x})$ 在滤波时是一个常数（为什么？），所以可以提到算子的外面。再根据暗通道先验理论，将 1 式代入 12 式，可得：

$$\tilde{t}(\mathbf{x}) = 1 - \min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \frac{I^c(\mathbf{y})}{A^c} \right) \quad (13)$$

作者提到，对于雾图像中的天空区域颜色很趋近于大气光 A ，所以在天空区域有：

$$\min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \frac{I^c(\mathbf{y})}{A^c} \right) \rightarrow 1 \quad (14)$$

所以 13 式在天空区域求出来的值会趋于 0，所以不需要单独考虑天空，13 可以同时处理天空和非天空区域。然而，我并没有看出天空区域的 t 趋于 0 和该公式可以直接处理天空区域之间的逻辑关系。实际上，从最后求解 J 的公式（下一节给出）来看， t 趋于 0 反而会加剧对该区域的作用，因为 t 出现在分母上。

作者还提到，即便是晴朗的天气下，也是有雾存在的；而且，雾还是人感知深度的一条线索。所以如果把雾彻底去除，图像看起来反而会不自然。因此还需用一个参数 ω 对 13 式进行修正：

$$\tilde{t}(\mathbf{x}) = 1 - \omega \min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r, g, b\}} \frac{I^c(\mathbf{y})}{A^c} \right) \quad (15)$$

这里 ω 介于 0 到 1 之间，论文里的所有结果都是取的 0.95。

无雾图像 J 的求解

至此，所有的位置参数都已求出，解 3 式，可得：

$$J(\mathbf{x}) = \frac{I(\mathbf{x}) - A}{t(\mathbf{x})} + A \quad (16)$$

当 $t(\mathbf{x})$ 很小的时候，可能会导致结果出现噪声，需要对 $t(\mathbf{x})$ 进行限制，对 16 式进行修正，得到最终的表达式为：

$$J(\mathbf{x}) = \frac{I(\mathbf{x}) - A}{\max(t(\mathbf{x}), t_0)} + A \quad (17)$$

式中 t_0 的标准值为 0.1。

3.2.3 算法复现

Listing 3:] 暗通道先验算法代码复现¹

```
1  classdef dark_channel_prior < handle
2
3      properties( SetAccess=private , GetAccess=public )
4          % 在构造函数中初始化的变量
5          im_path;% 图像路径
6          I;% 读取的图像，待去雾
7          M;% 图像的行数
8          N;% 图像的列数
9          path;% 读取的图像所在路径
10         name;% 所读图像文件名
11         ext;% 所读图像的扩展名
12
13         % 待求变量
14         dark_channel;% 图像的暗通道
15         dark_path;% 暗通道结果路径
16         tx;% the medium transmission, 媒介传播函数
17         A;% global atmospheric light, 大气光
18         I_by_A;%  $I^c / A^c$ 
19         I_by_A_dark;%  $I^c / A^c$ 的暗通道
20         %A_loc;% A在I中的位置
21         J;% 去雾后的图像
22         num;% 前num个最大值，用于求A
23
24         % 常数，有默认值
25         w;% 一个常系数，用于保留一定的雾，默认0.95
26         patch_size;% 最小值滤波的滤波器大小，默认15X15
27         t0;% 防止t太小导致结果有噪声，默认0.1
28
29     end
30
31     methods
32         % 构造函数
33         function obj = dark_channel_prior(im_path, varargin)
34
35             % 设置默认参数
36             obj.w = 0.95;
37             obj.patch_size = 15;
38             obj.t0 = 0.1;
39             if length(varargin) == 1
40                 obj.w = varargin{1};
41             elseif length(varargin) == 2
42                 obj.w = varargin{1};
43                 obj.patch_size = varargin{2};
44             elseif length(varargin) == 3
45                 obj.w = varargin{1};
46                 obj.patch_size = varargin{2};
47                 obj.t0 = varargin{3};
48             end
49         end
50     end
51 end
```



```

49
50
51 % 初始化成员变量
52 obj.im_path = im_path;
53 obj.I = imread(im_path);
54 obj.I = im2double(obj.I);
55 [obj.M, obj.N, ~] = size(obj.I);
56 [obj.path, obj.name, obj.ext] = fileparts(im_path);
57 obj.path = strcat(obj.path, '\');
58 obj.num = round( obj.M * obj.N / 1000 );
59
60 % 执行去雾操作
61 obj.dehaze();
62 end
63
64
65 end
66
67 methods
68 % 求暗通道
69 function img_dark = find_dark_channel(obj, img)
70 % 取每个通道上的最小值
71 img_dark = min( img, [], 3 );
72 % 最小值滤波
73 img_dark = ordfilt2( img_dark, 1, ones(obj.patch_size), 'symmetric' );
74 end
75 end
76
77 methods
78 % 求A
79 function find_A(obj)
80 t = sort( obj.dark_channel(:), 'descend' );
81 [m, n] = find( obj.dark_channel >= t(obj.num), obj.num);
82 obj.A = [0, 0, 0];
83 %obj.A_loc = [0, 0];
84 for x = 1 : obj.num
85 % intensity = max( obj.I( m(x), n(x), : ) );
86 % if intensity > max(obj.A)
87 % obj.A(1) = obj.I( m(x), n(x), 1 );
88 % obj.A(2) = obj.I( m(x), n(x), 2 );
89 % obj.A(3) = obj.I( m(x), n(x), 3 );
90 % end
91 if obj.I( m(x), n(x), 1) > obj.A(1)
92 obj.A(1) = obj.I( m(x), n(x), 1);
93 end
94 if obj.I( m(x), n(x), 2) > obj.A(2)
95 obj.A(2) = obj.I( m(x), n(x), 2);
96 end
97 if obj.I( m(x), n(x), 3) > obj.A(3)
98 obj.A(3) = obj.I( m(x), n(x), 3);
99 end

```

```

100
101         end
102
103     end
104 end
105
106
107 methods
108     % 求tx
109     function find_tx(obj)
110         obj.tx = zeros( obj.M, obj.N );
111         obj.I_by_A = obj.I;
112         obj.I_by_A(:, :, 1) = obj.I_by_A(:, :, 1) / obj.A(1);
113         obj.I_by_A(:, :, 2) = obj.I_by_A(:, :, 2) / obj.A(2);
114         obj.I_by_A(:, :, 3) = obj.I_by_A(:, :, 3) / obj.A(3);
115         obj.I_by_A_dark = obj.find_dark_channel( obj.I_by_A );
116         obj.tx = 1 - obj.w * obj.I_by_A_dark;
117     end
118 end
119
120 methods
121     % 去雾
122     function dehaze(obj)
123         % 求原图像的暗通道并显示
124         obj.dark_channel = obj.find_dark_channel( obj.I );
125         figure
126         imshow(obj.dark_channel);
127         title('暗通道');
128         obj.dark_path = strcat( obj.path, obj.name );
129         obj.dark_path = strcat( obj.dark_path, '_dark.png');
130         imwrite( obj.dark_channel, obj.dark_path );
131
132         % 求A, 结果存放在obj.A中
133         obj.find_A();
134         %求tx, 结果存放在obj.tx中
135         obj.find_tx();
136
137         obj.J = obj.I;
138         for x = 1 : obj.M
139             for y = 1 : obj.N
140                 for c = 1 : 3
141                     obj.J(x, y, c) = obj.I(x, y, c) - obj.A(c);
142                     obj.J(x, y, c) = obj.J(x, y, c) / max( obj.tx(x, y), obj.t0);
143                     obj.J(x, y, c) = obj.J(x, y, c) + obj.A(c);
144                 end
145             end
146         end
147
148         figure
149         imshow(obj.J)
150         title('去雾后的结果')

```

```

151     res_path = strcat( obj.path, obj.name);
152     res_path = strcat( res_path, 'res.jpg');
153     imwrite( obj.J, res_path );
154
155     end
156 end
157 end

```

3.2.4 实验结果



(a) 原图 1



(b) 去雾结果 1

图 4: 结果 1



(a) 原图 2



(b) 去雾结果 2

图 5: 结果 2

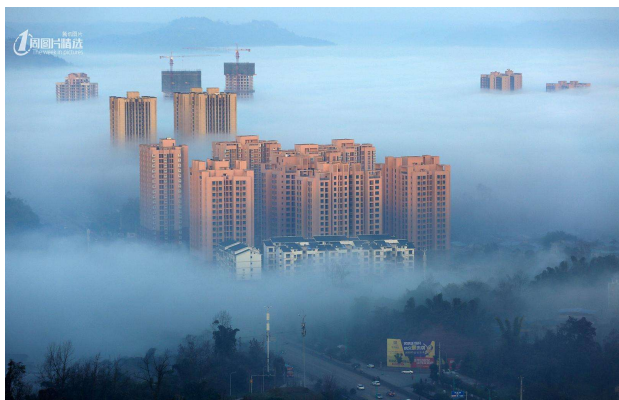


(a) 原图 3

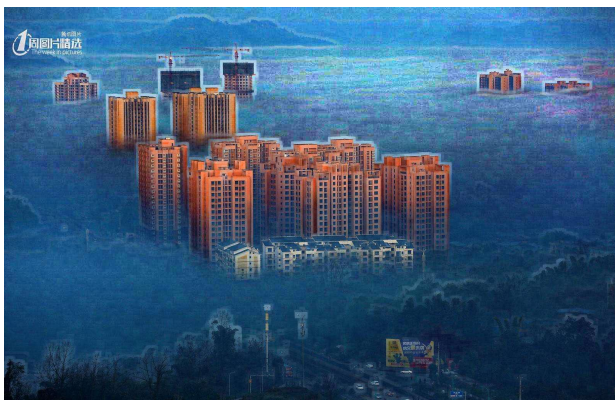


(b) 去雾结果 3

图 6: 结果 3



(a) 原图 4



(b) 去雾结果 4

图 7: 结果 4



(a) 原图 5



(b) 去雾结果 5

图 8: 结果 5

其实论文中还有一个环节 *Soft Matting*，由于这一节实在是没看懂，所以最终没有实现这一块。得到的结果质量也不如论文中的稳定。图5图8所示的结果还比较不错，在去除了雾的同时还大体能维持色调不变。但剩余的三幅图像颜色失真就比较严重，尤其是天空所在的区域。正如本次报告在3.2.2中提出的那样，6式究竟能否处理天空区域？为什么？

另外，从图5和图7中可以看到，去雾后的图像边界有明显的白带。对比论文中的 Fig. 6，可以知道这些白带产生原因是由于未加 *Soft Matting* 而造成的。

4 总结

本次报告分为两部分：第一部分总结了 *RGB* 空间和 *HSI* 空间下的图像增强方法的特点，并通过相应的实验来说明；第二部分总结了几类水下图像增强的方法，并着重介绍了在该领域起着重要作用的暗通道先验算法并对该算法进行了一定程度的复现。

这次报告的绝大多数时间都花在了暗通道先验算法上。由于这篇论文还有很多地方暂时没有想明白，因此对算法的复现也很难说是成功。从论文里的结果来看，就算没有加 *Soft Matting*，去雾结果的颜色也不应该有如图5和图7这样大的失真，也不应该有像图4这样多的彩色噪声。暂时还不清楚问题具体出在哪里，只能推测应该是 t 和 A 的求解可能有问题。

参考文献

- [1] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010. 5
- [2] Rafael C. Gonzalez, Richard E. Woods. 数字图像处理（第三版）. 电子工业出版社, 2011.6. 2
- [3] 宋瑞霞, 李达, 王小春. 基于 hsi 色彩空间的低照度图像增强算法. *图学学报*, 2017. 3
- [4] 林明星, 代成刚, 董雪, 张东, 管志光. 水下图像处理技术研究综述. *测控技术*:1-14, 2020. 5
- [5] 林森, 迟凯晨, 李文涛, 唐延东. 基于优势特征图像融合的水下光学图像增强. *光子学报*, 49(3), 2020. 5
- [6] 邹立, 卢俊彦, 胡易, 赵猛, 高正中, 程学珍. 非均匀光照条件下的水下图像增强算法. *山东科技大学学报（自然科学版）*, 39(02), 2020. 5