

数字图像处理

期末大作业技术报告

(2023学年秋季学期)

课程名称: Digital Image Processing

选题

组员

水下图像复原 刘俊杰、张礼贤、唐成文

摘要

水下图像采集中，光在水中传播会经历折射、散射和吸收等复杂过程，导致图像出现模糊、色彩失真和对比度降低等问题。水体中的颗粒和溶解物质对光传播产生显著影响，使得水下图像复原成为一个具有挑战性的研究方向。本文提出了一种基于暗通道先验去雾、白平衡和局部直方图均衡化的水下图像复原方法，旨在提高水下目标的识别性能和图像质量。

背景

在水下图像采集中，光在水中的传播经历了多种复杂的光学过程，包括折射、散射和吸收。这些现象导致水下图像普遍存在模糊、色彩失真以及对比度降低等问题。水体中的颗粒和溶解物质对光的传播产生显著影响，进一步加剧了图像的退化效应。

由于水体对不同波长的光的吸收程度不同，水下图像中蓝色和绿色波段相对较好地传播，而红色和近红外波段则受到更为严重的吸收。这种波长相关性使得水下图像呈现出典型的色偏，给目标的视觉感知带来了挑战。

在水下环境中，光强度随深度的增加呈指数衰减。深层目标由于光照不足，通常呈现出较低的对比度和细节信息，导致其在图像中难以准确识别。这一现象对于水下目标检测和识别造成了严重的限制。

因此，水下图像复原是一个具有挑战性的研究方向，涉及光学物理、图像处理 and 计算机视觉等多个领域的交叉。有效地处理水下图像的退化问题对于提高水下目标的识别性能和图像质量至关重要。

文献综述

在水下环境中，光的传播受到水质、颗粒散射和吸收等因素的影响，导致水下图像普遍存在色偏、模糊和对比度降低等问题。为了解决这些问题，先前的研究主要集中在去雾、白平衡和对比度增强等方面。其中，基于暗通道先验的去雾方法通过估计透射率和大气光照，有效地减少雾霾效应。白平衡方法通过调整图像的颜色通道，消除水下环境引起的色彩偏移。局部直方图均衡化则通过增强图像的对比度，提升细节可见性。这些方法为水下图像复原提供了有力的工具和思路。

方法描述

1. 暗通道先验去雾：

- 原理解释：**暗通道先验去雾是基于以下观察：在室外自然场景中，图像的暗通道（Dark Channel）在大多数情况下都会包含低亮度的像素值。这是由于大气散射引起的雾霾效应。因此，通过估计图像的暗通道，可以得到关于大气光和透射率的先验信息。去雾过程实际上就是通过推测透射率，对图像进行调整，以减少雾霾的影响。
- 模型建构** 定义以下模型：

$$I(x) = \hat{I}(x)t(x) + A(1 - t(x))$$

- 其中， $I(x)$ 是指的观测到的亮度，即从拍到图片中获取到的亮度，这个是已知值。
- $\hat{I}(x)$ 理解成为去雾、恢复之后的图像，是我们的目标。
- $t(x)$ 是透射率。
- A 是全球大气光成分。通过这个公式，我们的目标就是从已有的相片中计算得到原始无雾图像，透射率以及估算全球大气光成分。

。步骤解释：

1. **计算图像的暗通道：** 暗通道是对图像每个位置的 RGB 通道的最小值的估计。

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

其中， $J(x)$ 表示图像在位置 x 的暗通道， $I(y)_c$ 表示图像在位置 y 的通道 c 的像素值。

2. **暗通道先验理论：**

- 暗通道先验理论指出，非天空区域无雾图像的暗通道

$$J^{dark}(x) \rightarrow 0$$

- 在水下图像复原中，有以下因素导致了暗通道：

1. **散射和吸收：** 当光线穿过水体时，会发生散射和吸收，特别是对蓝色光的吸收更为显著。这导致水下场景中的光线强度降低，使得在深度较大的区域（远离水面）中，光线更容易被吸收，从而减弱了图像的亮度。

2. **水体颗粒：** 水体中的悬浮颗粒（如悬浮泥沙、藻类等）也会散射光线，影响图像的明暗程度。这些颗粒使得在水下场景中，即使是在无遮挡区域，也可能存在较强的散射效应，使得图像较为暗淡。

3. **水质和能见度：** 不同水体的水质和能见度差异很大，这会影响光线在水中传播的方式。在水质较差的水体中，散射和吸收现象更为显著，导致暗通道更为突出。

3. **估计大气光 A：** 通过对暗通道进行局部滤波，得到大气光的估计。

$$A(x) = \max_{y \in \text{neighborhood}(x)} (J(y))$$

即从暗通道中提取最亮的前0.1%像素点作为大气光照的估计值

4. **计算透射率 tr：** 将公式变形得：

$$\frac{I(x)}{A} = t(x) \frac{\hat{I}(x)}{A} + (1 - t(x))$$

对两边同时进行最小值运算：

$$\min_{y \in neighbor} (\min_c (\frac{I^c(y)}{A})) = t(x) \min_{y \in neighbor} (\min_c \frac{\hat{I}^c(y)}{A}) + (1 - t(x))$$

由暗通道公式得到：

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

$$J^{dark}(x) \rightarrow 0$$

因此式子中

$$\min_{y \in neighbor} (\min_c \frac{\hat{I}^c(x)}{A}) = 0$$

利用估计的大气光和预设的参数计算透射率。

另外，雾的存在让人们感受到景深，因此在去雾的同时有必要保留一定程度的雾。可以通过引入一个0到1之间的因子 w （一般取0.95）对预估透射率进行修正

$$tr(x) = 1 - w_0 \frac{J(x)}{A}$$

在暗通道先验去雾方法中，设置阈值的目的是为了透射率（transmission）的值过小，导致除法运算中的分母接近零，从而引起数值不稳定或者产生较强的伪影。

5. 进行去雾处理： 利用透射率对原始图像进行去雾处理。

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

其中， $\hat{I}(x)$ 表示去雾后的图像在位置 x 的像素值， $t(x)$ 表示透射率， t_0 是一个阈值。

。代码实现：

```

I1 = zeros(m, n);
for i = 1:m
    for j = 1:n
        I1(i, j) = min(I(i, j, :));
    end
end

% 计算暗通道
Id = ordfilt2(I1, 1, ones(wh, wh), 'symmetric');
dark_channel = Id;
A_temp = max(max(dark_channel)) * 0.999;
A = A_temp;
tr = 1 - w0 * Id / A;

% 进行去雾处理
t0 = 0.1;
t1 = max(t0, tr);
I_out = zeros(m, n, 3);
for k = 1:3
    for i = 1:m
        for j = 1:n
            I_out(i, j, k) = (I(i, j, k) - A) /
t1(i, j) + A;
        end
    end
end
end

```

2. 白平衡:

- **原理解释:** 白平衡的目标是消除图像中的颜色偏移, 使得白色在图像中呈现为真实的白色。在水下图像中, 由于水体吸收不同波长的光线, 导致颜色发生偏移。通过计算图像各通道的增益系数, 对每个通道进行调整, 达到白平衡的效果。

$$\text{新通道值} = \text{增益系数} \times \text{原通道值}$$

具体而言, 对于红色通道:

$$R_{\text{new}}(x) = \frac{\text{aveGray}}{\text{aveRed}} \times R(x)$$

对于绿色和蓝色通道也有类似的公式。

。步骤解释：

1. 计算原始图像每个通道的平均亮度值。

```
imgR = Im(:, :, 1);  
imgG = Im(:, :, 2);  
imgB = Im(:, :, 3);  
RAve = mean2(imgR);  
GAve = mean2(imgG);  
BAve = mean2(imgB);  
aveGray = (RAve+GAve + BAve) / 3;
```

2. 计算各通道的增益系数，使得平均亮度值一致。

```
RCoef = aveGray / RAve * 0.6;  
GCoef = aveGray / GAve;  
BCoef = aveGray / BAve;
```

3. 使用增益系数调整原始图像的每个通道，完成白平衡。

```
RCorrection = RCoef * imgR; %水下图片红色较少  
GCorrection = GCoef * imgG;  
BCorrection = BCoef * imgB;  
imgDst = originalImage;  
imgDst(:, :, 1) = RCorrection;  
imgDst(:, :, 2) = GCorrection;  
imgDst(:, :, 3) = BCorrection;
```

。代码实现：

```

imgR = Im(:, :, 1);
imgG = Im(:, :, 2);
imgB = Im(:, :, 3);
RAve = mean2(imgR);
GAve = mean2(imgG);
BAve = mean2(imgB);
aveGray = (RAve + GAve + BAve) / 3;

% 计算增益系数
RCoef = aveGray / RAve * 0.6;
GCoef = aveGray / GAve;
BCoef = aveGray / BAve;

% 使用增益系数来调整原始图像
RCorrection = RCoef * imgR;
GCorrection = GCoef * imgG;
BCorrection = BCoef * imgB;
imgDst = originalImage;
imgDst(:, :, 1) = RCorrection;
imgDst(:, :, 2) = GCorrection;
imgDst(:, :, 3) = BCorrection;

```

3. 局部直方图均衡化:

- **原理解释:** 局部直方图均衡化是一种增强图像对比度的方法，它将图像分成小块，对每个小块进行直方图均衡化。这有助于提高图像中细节的可见性。
- **步骤解释:**

1. 将图像的 RGB 通道分离。

```

redChannel = imgDst(:, :, 1);
greenChannel = imgDst(:, :, 2);
blueChannel = imgDst(:, :, 3);

```

2. 对每个通道应用局部直方图均衡化。

```

enhancedRedChannel = adapthisteq(redChannel,
'NumTiles', [blockSize, blockSize]);
enhancedGreenChannel =
adapthisteq(greenChannel, 'NumTiles',
[blockSize, blockSize]);
enhancedBlueChannel = adapthisteq(blueChannel,
'NumTiles', [blockSize, blockSize]);

```

- 'NumTiles', [blockSize, blockSize]: 指定图像被划分的块的大小。blockSize 是之前设定的块的大小，也就是局部直方图均衡化处理的基本单元。
- adapthisteq 函数是基于自适应直方图均衡化（AHE）算法实现的。自适应直方图均衡化与全局直方图均衡化的主要区别在于，AHE 将图像划分为小块，并对每个小块的直方图进行均衡化，以适应局部灰度分布的变化。下面是 AHE 的主要步骤：

1. 图像分块：将输入图像划分为大小相等的小块。
2. 计算每个块的直方图：对每个小块计算直方图，统计不同灰度级别的像素数量。
3. 对每个块进行直方图均衡化：对每个小块的直方图进行均衡化。具体而言，对于每个块，计算其累积分布函数（CDF），并将块内的像素值映射到均匀的分布。这个过程使得块内的灰度级别更均匀地覆盖整个灰度范围。
4. 合并块：将均衡化后的小块重新组合成一幅图像。

- 这一过程分别应用于独立的红色、绿色和蓝色通道，以确保每个通道都能够得到适当的局部直方图均衡化。

3. 合并增强后的通道，得到最终的增强彩色图像。

```

% 合并增强后的通道
enhancedColorImage = cat(3,
enhancedRedChannel, enhancedGreenChannel,
enhancedBlueChannel);

```


。 代码实现：

```
% 设定局部直方图均衡化的块大小
blockSize = 2;

% 分离RGB通道
redChannel = imgDst(:, :, 1);
greenChannel = imgDst(:, :, 2);
blueChannel = imgDst(:, :, 3);

% 应用局部直方图均衡化到每个通道
enhancedRedChannel = adapthisteq(redChannel,
    'NumTiles', [blockSize, blockSize]);
enhancedGreenChannel = adapthisteq(greenChannel,
    'NumTiles', [blockSize, blockSize]);
enhancedBlueChannel = adapthisteq(blueChannel,
    'NumTiles', [blockSize, blockSize]);

% 合并增强后的通道
enhancedColorImage = cat(3, enhancedRedChannel,
    enhancedGreenChannel, enhancedBlueChannel);
```

实验结果

参考文献

- [1] [He K, Sun J, Tang X. Single image haze removal using dark channel prior\[J\]. IEEE transactions on pattern analysis and machine intelligence, 2010, 33\(12\): 2341-2353.](#)
- [2] [Gijssenij A, Gevers T. Color constancy using natural image statistics\[C\]//2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007: 1-8.](#)
- [3] [Pizer S M, Amburn E P, Austin J D, et al. Adaptive histogram equalization and its variations\[J\]. Computer vision, graphics, and image](#)

processing, 1987, 39(3): 355-368.