

水下图像复原

小组成员

冯浩	凌国明
21307155	21307077

摘要

本文详细探讨了三种主要的水下图像复原技术：白平衡、局部直方图均衡化和去雾，以及它们的结合对提高水下图像质量的效果。通过对这些方法进行综合应用，我们不仅解决了色偏问题，还改善了图像的对比度和清晰度。本于探索了这些方法如何有效结合可以最大化图像复原效果，给出了一种较优的解决方案，适应不同的水下摄影条件和需求。

背景

在海洋探索、海洋生态保护、海洋资源开发以及海洋军事领域，高质量的水下图像扮演着至关重要的角色。然而，由于水的折射、吸收和水中悬浮颗粒的散射特性，水下摄影面临着诸多挑战，如显著的色偏、对比度降低和细节丢失。这些问题严重限制了水下图像在科学研究和实际应用中的有效性。

因此，水下图像复原技术成为了一个研究热点，其目的是通过恢复图像的自然色彩和提高可视性，以支持更精准的海洋环境分析和监测。本文提出的综合复原方法，通过结合白平衡、局部直方图均衡化和去雾等技术，旨在提供一种有效的水下图像处理方案，以应对这些视觉挑战，支持海洋科学和技术的发展。

处理前图像 和 处理后图像



方法

白平衡

白平衡是图像处理中用于调整图像中的颜色偏差，使其更接近人眼所看到的自然颜色的技术。其主要目的是消除由于光源色温不同而导致的色彩偏差。白平衡的基本原理是通过调整图像中的红色、绿色和蓝色通道的强度，使得白色或灰色物体在图像中显现为中性色（无色偏）。

简单来说，白平衡处理可以使得图像的色彩更平衡，以使其更接近人眼所见的自然状态。其核心思想是识别图像中的白色或灰色区域，并据此调整所有颜色，以补偿光源色温的影响。

以下是一种基于 YCbCr 色彩空间的白平衡处理的数学表达：

$$Mb = \frac{1}{N} \sum_{i=1}^N Cb_i$$
$$Mr = \frac{1}{N} \sum_{i=1}^N Cr_i$$

其中，Mb 和 Mr 分别是 Cb 和 Cr 分量的平均值，N 是像素点总数。

$$Db = \frac{\sum (Cb_i - Mb)}{N}$$
$$Dr = \frac{\sum (Cr_i - Mr)}{N}$$

Db 和 Dr 分别是 Cb 和 Cr 分量的均方差。

$$Rgain = \frac{Ymax}{Rav}$$
$$Ggain = \frac{Ymax}{Gav}$$
$$Bgain = \frac{Ymax}{Bav}$$

Rgain、Ggain 和 Bgain 是为了调整 RGB 通道而计算的增益系数，其中 Ymax 是亮度最大值，而 Rav、Gav 和 Bav 是参考白点在 RGB 通道中的平均值。

代码

```
function balancedImage = whiteBalance(img)
    im1 = rgb2ycbcr(img); % 将图片的 RGB 值转换成 YCbCr 值
    Lu = im1(:,:,1);
    Cb = im1(:,:,2);
    Cr = im1(:,:,3);
    [x, y, ~] = size(img);
    tst = zeros(x, y);

    % 计算 Cb、Cr 的均值 Mb、Mr
    Mb = mean(mean(Cb));
    Mr = mean(mean(Cr));

    % 计算 Cb、Cr 的均方差
    Db = sum(sum(Cb - Mb)) / (x * y);
    Dr = sum(sum(Cr - Mr)) / (x * y);

    % 根据阈值要求提取出 near-white 区域的像素点
    cnt = 1;
    for i = 1:x
        for j = 1:y
            b1 = Cb(i,j) - (Mb + Db * sign(Mb));
            b2 = Cr(i,j) - (1.5 * Mr + Dr * sign(Mr));
            if (b1 < abs(1.5 * Db) & b2 < abs(1.5 * Dr))
                Ciny(cnt) = Lu(i,j);
                tst(i,j) = Lu(i,j);
                cnt = cnt + 1;
            end
        end
    end
    cnt = cnt - 1;
    iy = sort(Ciny, 'descend'); % 将提取出的像素点按亮度值从大到小排列
    nn = round(cnt / 10);
    Ciny2(1:nn) = iy(1:nn); % 提取出 near-white 区域中 10% 亮度值较大的像素点做参考白点

    % 提取出参考白点的 RGB 三信道的值
    mn = min(Ciny2);
    for i = 1:x
        for j = 1:y
            if tst(i,j) < mn
                tst(i,j) = 0;
            else
```

```

        tst(i,j) = 1;
    end
end
end
R = img(:,:,1);
G = img(:,:,2);
B = img(:,:,3);
R = double(R) .* tst;
G = double(G) .* tst;
B = double(B) .* tst;

% 计算参考白点的 RGB 均值
Rav = mean(mean(R));
Gav = mean(mean(G));
Bav = mean(mean(B));
Ymax = double(max(max(Lu))) / 15; % 计算出图片的亮度最大值

% 计算 RGB 三信道的增益
Rgain = Ymax / Rav;
Ggain = Ymax / Gav;
Bgain = Ymax / Bav;

% 通过增益调整图片的 RGB 三信道
img(:,:,1) = img(:,:,1) * Rgain;
img(:,:,2) = img(:,:,2) * Ggain;
img(:,:,3) = img(:,:,3) * Bgain;
balancedImage = img;
end

```

处理前图像 和 处理后图像



去雾

去雾算法的基本思想是估计每个像素的雾霾程度，并相应地调整像素值以减少雾霾的影响。这通常涉及到大气光估计和透射图的恢复。

以下是去雾算法的数学表达：

暗通道先验

暗通道先验是基于自然图像的一个观察，即除去天空等少数区域外，大多数非天空区域的某个颜色通道至少在某个像素上有很低的值

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

其中， $J^{dark}(x)$ 是暗通道图像， $J^c(y)$ 是颜色通道 c 在位置 y 的强度， $\Omega(x)$ 是以 x 为中心的局部窗口。

大气光估计

使用暗通道图像估计大气光。通常选择暗通道图像中最亮的一些像素，然后在原始图像中对应的像素中找到具有最高强度的像素作为大气光。

$$A = \max_{x \in top(J^{dark})} (I(x))$$

其中， A 是大气光， $I(x)$ 是原始图像在位置 x 的强度。

透射图估计

透射图描述了场景中的每个像素有多少光直接到达相机而没有被大气光散射

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

其中， $t(x)$ 是透射图，在位置 x 的透射率， ω 是保持一些大气光的常数（通常取 0.95）。

恢复去雾图像

最后，使用大气光和透射图恢复无雾图像。

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

其中， $J(x)$ 是恢复的无雾图像， t_0 是透射图的下限阈值，用于避免过度放大噪声。

代码

```
function dehazedImage = imageDehazing(img)
% 将图像转换为double类型
img = im2double(img);

% 1. 计算暗通道
darkChannel = min(img, [], 3); % 计算每个像素在所有颜色通道中的最小值
patchSize = 15; % 定义局部窗口的大小
darkChannel = ordfilt2(darkChannel, 1, ones(patchSize, patchSize), 'symmetric');

% 2. 估计大气光
[~, idx] = sort(darkChannel(:), 'descend');
idx = idx(1:max(floor(numel(idx) * 0.001), 1)); % 选择暗通道图像中最亮的 0.1% 的像素
atmosphericLight = max(img(idx));

% 3. 估计透射图
omega = 0.95; % 保留一部分大气光
transmission = 1 - omega * min(min(img ./ atmosphericLight, [], 3), [], 3);
transmission = max(transmission, 0.1); % 设置透射图的下限阈值

% 4. 恢复无雾图像
dehazedImage = (img - atmosphericLight) ./ max(transmission, 0.1) + atmosphericLight;
dehazedImage = min(max(dehazedImage, 0), 1); % 确保像素值在合理范围内
end
```

这段代码首先计算了图像的暗通道，然后用它来估计大气光。接着，基于大气光估计透射图。最后，使用这些信息来恢复无雾图像。代码中使用了一个小窗口在图像上滑动，以找到每个区域的最小值，这有助于确定雾霾的程度。大气光和透射图的估计是基于暗通道先验的关键步骤，用于恢复图像。

处理前图像 和 处理后图像



直方图均衡化

原理

局部直方图均衡化是一种通过将图像分成小块并对每个小块进行直方图均衡化的方法。这种方法允许在不同区域内进行对比度调整，有助于增强图像的局部细节。在本实验中，采用 `adapthisteq` 函数实现局部直方图均衡化。

对比度拉伸

对比度拉伸是通过调整图像的像素值范围来增强图像对比度的一种方法。通过设定最小和最大像素值，将原始图像的像素值映射到新的范围，从而增强图像的对比度。在本实验中，采用 `ContrastStretch` 函数实现对比度拉伸，也属于直方图均衡化。

对比度拉伸通过调整像素值的范围，对整体图像进行对比度增强，而限制对比度自适应直方图均衡化在保持局部对比度增强的同时，限制了对比度的过度增强。

限制对比度自适应直方图均衡化

限制对比度自适应直方图均衡化是对传统直方图均衡化方法的改进，它通过在小块上应用直方图均衡化来增强图像的对比度，同时引入限制参数以避免过度增强对比度。在本实验中，采用 MATLAB 中的 `adapthisteq` 函数，同时设置了限制参数来实现限制对比度自适应直方图均衡化。

具体实现

1. 局部直方图均衡化处理：使用 `adapthisteq` 函数对图像进行局部直方图均衡化处理

```
for channel = 1:3
    histEqImage(:, :, channel) = adapthisteq(img(:, :, channel), 'NumTiles', [8, 8], 'ClipLimit', 1.5)
end
```

1. 对比度拉伸和限制对比度自适应直方图均衡化：通过 `ContrastStretch` 函数实现对比度拉伸。

`ContrastStretch` 函数实现：

- 首先检查输入参数 `x1` 和 `x2` 是否相同，如果相同则输出错误信息并退出
- 然后，对于每个像素，计算线性映射参数 `k` 和 `b`
- 然后根据输入像素值的范围进行线性映射
- 最后，输出调整后的图像 `I2`

处理前图像 (左) 和 处理后图像 (右)



综合的方法

将白平衡、直方图均衡化和去雾这三种方法进行排列组合，可以形成多种不同的处理流程。这些不同的组合方式可以根据特定的图像特性和需要解决的问题来选择。以下是所有可能的组合及其逻辑顺序：

1. 单一方法：仅白平衡，仅直方图均衡化，仅去雾
2. 两种方法组合：白平衡 -> 直方图均衡化，白平衡 -> 去雾，直方图均衡化 -> 白平衡，直方图均衡化 -> 去雾，去雾 -> 白平衡，去雾 -> 直方图均衡化
3. 三种方法组合：省略。

我们初步分析选择最佳的处理流程取决于水下图像的特定问题和目标：

1. 如果颜色失真是主要问题，那么首先进行白平衡可能是最优先的。
2. 如果图像的对比度和细节不足，那么直方图均衡化可能更加重要。
3. 如果图像因散射而变得模糊，去雾则可能是首选的步骤。

方法效果对比

以下两张是【原图】和【白平衡】



以下两张是【白平衡+直方图均衡化】和【白平衡+直方图均衡化+去雾】



以下两张是【原图】和【白平衡】



以下两张是【白平衡+直方图均衡化】和【白平衡+直方图均衡化+去雾】



认为较好的组合方法

我们通过实验和肉眼对比，得出了这样的结论：先做白平衡，再做直方图均衡化是一种较好的处理方法。这个结论和我们的预期也是较为符合的，以下是我们的两点想法：

颜色校正的重要性：在处理任何图像时，首先确保其颜色尽可能接近自然是非常重要的。白平衡恢复了图像的真实色彩，为后续的处理打下了基础。如果在色彩失真的情况下直接进行直方图均衡化，可能会加剧色彩偏差，导致最终结果不自然。

对比度和细节的增强：在颜色校正后，直方图均衡化进一步提高图像的对比度和细节。这在视觉上使图像更加吸引人，特别是在水下环境中，光线条件往往不理想，细节容易丢失。直方图均衡化使得暗区域变亮，同时保持高光区域的细节，这对于水下图像来说尤为重要。

实验结果 （左边处理前， 右边处理后）



总结

在方法实现的部分，我们集中讨论了水下图像处理的三个核心技术：白平衡、直方图均衡化和去雾。每一项技术都针对水下图像处理中的特定问题，如色彩失真、对比度不足和图像模糊等。通过深入分析每项技术的原理和应用，我们不仅理解了它们各自的优势和局限性，还发现了这些技术之间的相互补充性。例如，白平衡在校正色温偏差方面非常有效，而直方图均衡化则能显著提升图像的对比度和细节，去雾技术则主要用于提高图像的清晰度。这一部分的讨论让我们更加深刻地理解了水下图像处理的复杂性和挑战。

在方法结合的部分，我们探索了将这些技术结合应用的最佳实践。实验表明，先进行白平衡处理，再进行直方图均衡化，最后实施去雾处理，这样的顺序在多数情况下能够取得最佳效果。这种方法的逻辑在于，首先通过白平衡消除色偏，确保图像颜色的自然性；接着，通过直方图均衡化增强图像的对比度和细节；最后，通过去雾技术清除图像中的模糊和散射效果，提高清晰度。这种结合方法的探讨不仅提高了我们处理水下图像的灵活性，也为我们如何根据不同的应用场景选择适当的处理流程提供了指导。

通过这次作业，结合我在数字图像处理课程中的学习，我们不仅深化了对水下图像处理技术的理解，还体会到了在实际问题中综合运用多种技术的重要性。数字图像处理课程为我们提供了理论基础，而这次实验则是将理论知识应用于实践的一次尝试。通过将白平衡、直方图均衡化和去雾等技术综合运用，我们更加深刻地认识到，处理复杂的图像问题需要多角度的思考和综合方法的应用。这不仅增强了我们对图像处理技术的操作能力，也提升了我们解决实际图像处理问题的综合素养。通过这种实践，我们能够更有效地将课堂上学到的知识应用到具体的问题解决中。

参考文献

- [1] He K, Sun J, Tang X. Single Image Haze Removal Using Dark Channel Prior[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 1956-1963.
- [2] Zuiderveld K. Contrast limited adaptive histogram equalization[C]//Graphics gems IV. Academic Press Professional, Inc., 1994: 474-485.

<https://github.com/wangyanckxx/Single-Underwater-Image-Enhancement-and-Color-Restoration>

<https://github.com/rawi707/UNDROIP>

<https://github.com/UNDROIP/UNDROIP>

<https://zhuanlan.zhihu.com/p/96767081>

<https://zhuanlan.zhihu.com/p/98834353>