

# 数字媒体技术基础 20 组进展报告

## 小组成员

姓名	学号
张子扬	15331414
钟敏欣	15331426
朱文豪	15331443
赵寒旭	15331416
邹伊宁	15331449
李启明	14331133

## 进度总结

此次进展报告的主要内容是对《Single Image Haze Removal Using Dark Channel Prior》的详细阅读报告，附于下文。

根据相关资料，小组实现了暗通道先验去雾的算法，实验结果在展示 ppt 中已有简要的描述。

接下来的工作是结合其他论文的算法，基于暗通道先验对去雾算法进行改进，希望得到适应性更强的去雾效果。

### 《Single Image Haze Removal Using Dark Channel Prior》相关算法详解

#### 1. 暗通道先验 (Dark Channel Prior)

##### 1.1 原理

###### (1) 观察得到的结论

通过对室外无雾图像的观察可以发现，我们用 patch 表示与某一像素点关联的一片区域（正方形区域），在大多数非天空 patch（大小定为 $15 \times 15$ ）内，至少有一个颜色通道有某些光强度极低（接近于 0）的像素点，即 patch 中的最小光强度趋近于 0。

###### (2) 导致暗通道中的低强度产生的三个因素

1) 阴影，例如汽车，建筑物，城市景观图像中的窗户内部的影子，或者景观图像中的叶子，树木和岩石的阴影；

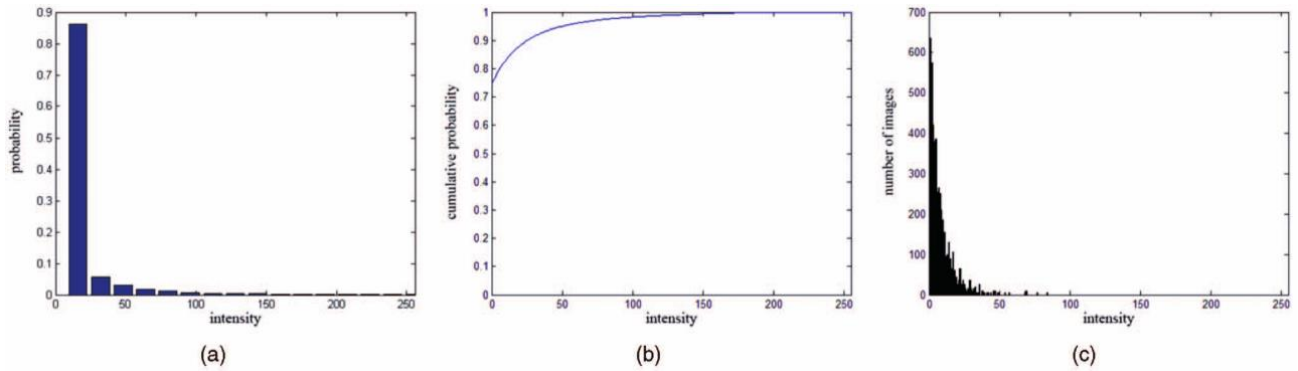
2) 彩色物体或表面，例如任何颜色通道（例如绿草/树/植物，红色或黄色花/叶和蓝色水面）的任何反射率低的物体将导致黑暗通道中的低值；

3) 黑暗的对象或表面，例如黑树干和石头。由于自然的户外图像通常是多彩的，充满阴影，这些图像的暗通道都十分黑暗。

###### (3) 利用大数据集的统计性验证

为了验证暗通道先验的正确性，选取户外图像集中无雾图片和城市图片（均为白天）。随机选择 5000 张图像，手动切割天空区域，调整图像大小，使得宽度和高度的最大值为 500 像素，并且使用 patch 大小为 15。

图(a)是 5,000 个暗通道的强度直方图，图(b)是相应的累积分布。暗通道中约 75%的像素具有零值，90%的像素的强度低于 25。这个统计量极大支持了暗通道先验算法。我们还计算每个暗通道的平均强度，并在图(c)中绘制相应的直方图，大多数暗通道具有非常低的平均强度，表明只有少部分室外无雾图像偏离我们的先验。



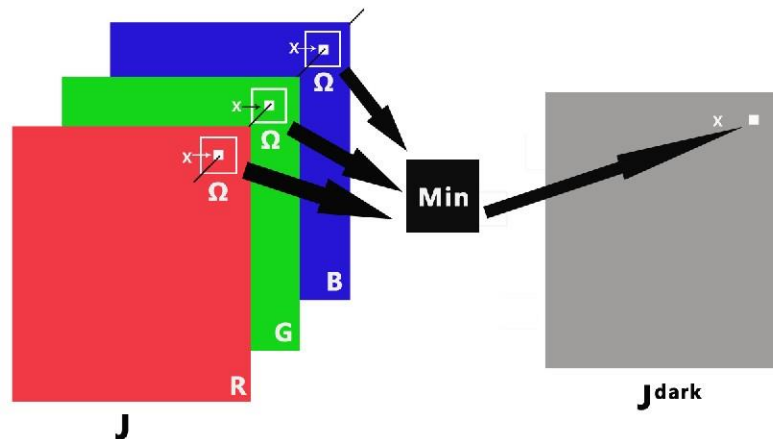
## 1.2 实现方法

定义：对任意图像  $J$ ，其暗通道  $J^{dark}$  表示为：

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

$J^c$  代表  $J$  的某一个颜色通道，而  $\Omega(x)$  是以  $x$  为中心的一块正方形区域 (local patch)。

以像素点  $x$  为中心，分别取三个通道内窗口  $\Omega$  内的最小值，然后再取三个通道的最小值作为像素点  $x$  的暗通道的值，如下图所示：



## 1.2 代码实现

```

1. J_darkchannel = min(Image, [], 3); % find the minimum value in RGB channel
2. for i = (1 : dimr)
3.     for j = (1 : dimc)
4.         winLeft = i - dx; winRight = i + dx;
5.         winUp = j - dx; winDown = j + dx;
6.
7.         % check the windows range
8.         if(i - dx < 1)
9.             winLeft = 1;
10.        end
11.        if(i + dx > dimr)
12.            winRight = dimr;
13.        end
14.        if(j - dx < 1)
15.            winUp = 1;
16.        end
17.        if(j+dx>dimc)
18.            winDown=dimc;
19.        end
20.
21.        % find the minimum value in the patch windows
22.        J_darktemp(i,j) = min(min(J_darkchannel(winLeft : winRight, winUp : wi
nDown)));
23.    end
24. end
25. J_darkchannel = J_darktemp;

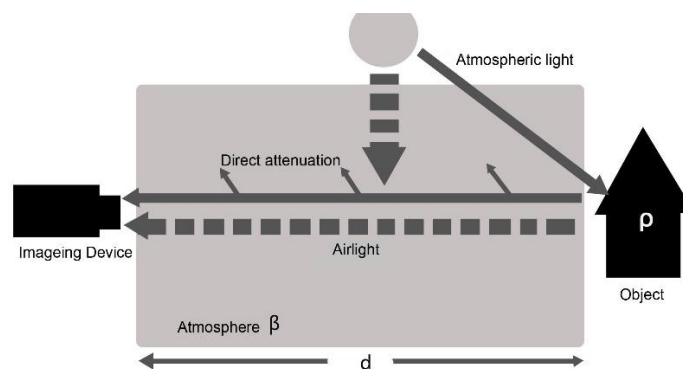
```

## 2. 估计大气光 (Estimating the Atmospheric Light)

### 2.1 原理

#### (1) 大气物理模型

要想从物理模型角度对有雾图像进行清晰化处理，就要了解雾天的大气散射模型。大气散射物理模型包含两部分，第一部分称为直接衰减项，第二部分称为大气光照 (Airlight)



$$I(x) = J(x)t(x) + A(1 - t(x))$$

$I$  是观测到的有雾图像， $J$  是景物反射光强度（也就是清晰的无雾图像）， $A$  是全局大气光照强度， $t$  用来描述光线通过介质透射到成像设备过程中没有被散射的部分，去雾的目标就是从  $I$  中复原  $J$ 。那么也就是要通过  $I$  求  $A$  和  $t$ 。

方程右边的第一项  $J(x)t(x)$  叫做直接衰减项，第二项  $A(1-t(x))$  则是大气光照。直接衰减项描述的是景物光线在透射媒介中经衰减后的部分，而大气光则是由前方散射引起的，会导致景物颜色的偏移。因为大气层可看成各向同性的，透射率  $t$  可表示为：

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

$\beta$  为大气的散射系数，该式表明景物光线是随着景物深度  $d$  按指数衰减的。

## (2) 估计大气光基本方法

通常有雾图像中最亮的像素被认为是雾最浓的地方，然而这个规律只适用于阴天阳光可以被忽略的情况，此时大气光是唯一照明源。

由于整个图像的最亮的像素可能比大气光更亮（如在白色的汽车或白色的建筑物上的像素点），可以使用暗通道来检测雾最浓的区域，改善大气光的估计，使其不易受输入图像中高亮像素的影响。

- 1) 从暗通道中选择出最亮的 0.1% 像素（认为这些像素所在位置是雾最浓处）
- 2) 从这些像素中选出在输入图像中对应最高强度的像素作为大气光（这些像素在整个输入图像中可能不是最亮的）

## 2.2 实现方法

求解全局大气光照  $A^c$

- (1) 对输入的有雾图像  $I$  求解其暗通道图像  $J^{dark}$
- (2) 选择暗通道  $J^{dark}$  内最亮的 0.1% 像素点，并记录这些像素点的  $(x, y)$  坐标。
- (3) 根据这些点的坐标分别在原图像  $I$  的三个通道  $(r, g, b)$  内找到这些像素点并加和得到  $(sum(r), sum(g), sum(b))$
- (4)  $A^c = [A^r, A^g, A^b]$ ，其中  $A^r = sum(r)/N$ ,  $A^g = sum(g)/N$ ,  $A^b = sum(b)/N$

## 2.3 代码实现

```
1.      % get the 0.1% most brightest pixels in the dark channel
2.      lightLimit = quantile(J_darkchannel(:), [.999]);
3.      [lightRow, lightCol] = find(J_darkchannel >= lightLimit);
4.      [enum, ~] = size(lightRow);
5.      for i = (1 : enum)
6.          A_r = Image(lightRow(i), lightCol(i), 1) + A_r;
7.          A_g = Image(lightRow(i), lightCol(i), 2) + A_g;
8.          A_b = Image(lightRow(i), lightCol(i), 3) + A_b;
9.      end
10.
11.      % get the Airlight
12.      Airlight = [A_r / enum, A_g / enum, A_b / enum];
```

## 3. 估计透射率 $t$ (Estimating the Transmission)

### 3.1 原理

$$I(x) = J(x)t(x) + A(1 - t(x))$$

现已知  $I(x), A$ ，需要求出透射率  $t(x)$  用于恢复无雾图像。

(1) 规范化表示

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

每个颜色通道都被独立表示。

(2) 假设每个局部 patch  $\Omega(x)$  里的透射率  $t$  都是常数，用  $\tilde{t}(x)$  表示透射率。对(1)中式两边求暗通道：

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

(3)  $J$  是无雾图像，暗通道值趋于 0， $A^c$  总为正

$$\min_{y \in \Omega(x)} (\min_c \frac{J^c(y)}{A^c}) = 0$$

(4) 得到透射率  $\tilde{t}$

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

$\min_{y \in \Omega(x)} (\min_c \frac{I^c(y)}{A^c})$  是规范化后有雾图像  $\frac{I^c(y)}{A^c}$  的暗通道。

(5) 有雾图像的天空颜色通常非常接近大气光，在天空区域，有：

$$\begin{aligned} \min_{y \in \Omega(x)} (\min_c \frac{J^c(y)}{A^c}) &\rightarrow 1 \\ \tilde{t}(x) &\rightarrow 0 \end{aligned}$$

由于天空区域无限远，其透射率的确接近于 0，因此我们不需要事先把天空区域分离开处理。

(6) 为使得恢复图像更加自然，不应彻底移除雾气，我们可以通过引入一个恒定的参数  $\omega$  ( $0 < \omega \leq 1$ )，为遥远物体选择性地保持非常小的雾度。

$$\tilde{t}(x) = 1 - \omega \min_{y \in \Omega(x)} (\min_c \frac{I^c(y)}{A^c})$$

### 3.2 实现方法

$$\text{透射率 } t \text{ 的公式 } \tilde{t}(x) = 1 - \omega \min_{y \in \Omega(x)} (\min_c \frac{I^c(y)}{A^c})$$

$I^c$  为输入的有雾图像，对其除以全局大气光照  $A^c$  后在利用暗通道定义公式进行求解暗通道。 $\omega$  ( $0 < \omega \leq 1$ ) 是雾的保留系数通常取 0.95。

这里需要值得注意的是，求得的  $t$  是粗透射率图，并不能直接带入大气模型公式求解，所以需要进行细化后再处理。

### 3.3 代码实现

```

1. % Estimating the raw transmission map(color)
2. Im_n(:, :, 1) = Image(:, :, 1) ./ Airlight(1);
3. Im_n(:, :, 2) = Image(:, :, 2) ./ Airlight(2);
4. Im_n(:, :, 3) = Image(:, :, 3) ./ Airlight(3);
5. tmap = min(Im_n, [], 3);
6.
7. for i = (1 : dimr)
8.     for j = (1 : dimc)
9.         winLeft = i - dx; winRight = i + dx;
10.        winUp    = j - dx; winDown  = j + dx;
11.
12.        % check the windows range
13.        if(i - dx < 1)
14.            winLeft = 1;
15.        end
16.        if(i + dx > dimr)
17.            winRight = dimr;
18.        end
19.        if(j - dx < 1)
20.            winUp = 1;
21.        end
22.        if(j + dx > dimc)
23.            winDown = dimc;
24.        end
25.
26.        % find the minimum value in the 惟 windows
27.        % get the image transmittance
28.        t_map(i, j) = 1 - w * min(min(tmap(winLeft : winRight, winUp : winDown)));
29.    end
30. end

```

#### 4. 精细化透射率

论文中的细化透射率的方法为软抠图（soft matting）的方法，详见论文如下：A Closed-Form Solution to Natural Image Matting[2]，作者：Anat Levin

#### 5. 恢复清晰图像（Recovering the Scene Radiance）

##### 5.1 原理

由大气光  $A$  和透射率  $t$  根据  $I(x) = J(x)t(x) + A(1 - t(x))$  恢复图像时，有

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

当  $t(x) \rightarrow 0$  时，可能有  $J(x)t(x) \rightarrow 0$ ，恢复后  $J(x) \rightarrow 0$  显示为噪点，影响效果。

为避免噪点的产生，给透射率选取一个下限  $t_0$ （典型取值为 0.1）

##### 5.2 实现方法

得到了光照强度  $A$  和透射率  $t$ ，代入大气模型公式：

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

这里， $t_0$  参数用来限定透射率  $t$  的下限值，其作用也就是在输入图像的浓雾区域保留一定的雾。

### 5.3 代码实现

```
1. % Getting the clear image(color)
2. % set the lowest t0
3. [lightRow, lightCol] = find(tmap_ref < 0.1);
4. [enum, ~] = size(lightRow);
5. for i = (1 : enum)
6.     tmap_ref(lightRow(i), lightCol(i)) = 0.1;
7. end
8.
9. J(:, :, 1) = (Image(:, :, 1) - Airlight(1)) ./ tmap_ref + Airlight(1);
10. J(:, :, 2) = (Image(:, :, 2) - Airlight(2)) ./ tmap_ref + Airlight(2);
11. J(:, :, 3) = (Image(:, :, 3) - Airlight(3)) ./ tmap_ref + Airlight(3);
```

### 6. Patch 大小选取 (Patch size)

size 选取较大的结果

- (1) 每个 patch 包含暗像素的概率更大，暗通道先验效果可能更好
- (2) 透射率在一个 patch 内为常数，靠近 depth edge 的光晕会更强，影响恢复图像的效果

综上两点，我们选取折中的 patch size 为  $15 \times 15$