

实验 1-基于暗通道先验的图像去雾

一、实验环境

Python version 3.9.13
opencv-python version 4.6.0.66
numpy version 1.23.3
matplotlib version 3.6.2
MATLAB R2021a

二、实验步骤

1、计算输入带雾图像的暗通道

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

其中， J^c 是 J 图像的一个颜色通道， $\Omega(x)$ 是一块以 x 为中心具有一定滤波半径的区域点的集合。根据暗通道先验的假设，除天空区域外，一张自然图像的暗通道总是非常低甚至趋于零。

2、估计大气亮度

何恺明博士提出了一种新的估计方法：首先选择在暗通道图像中前 0.1% 的像素，由暗通道先验条件保证了挑选出来的像素不会受到自然图像景物的干扰。然后找到在原始图像中 I 中对应的这 0.1% 的像素每个通道的最亮点，将其作为大气光强度 A 的估计值。

3、计算传导图像

大气亮度参数 A 已知，并且假定在 $\Omega(x)$ 区域内大气透射率 $t(x)$ 为常数，用 $\tilde{t}(x)$ 表示。根据暗通道先验条件，可以得到下式结果：

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

其中， $\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right)$ 且是对有雾图像标准化之后的暗通道，可以理解为模糊率，由此可以直接得到大气透射率 $t(x)$ 的估计值。

考虑到天空等场景对于暗通道先验条件的不适用性，无法利用上述的公式来

估计大气透射率，需要另作分析。由于天空亮度与大气光亮度 A 非常接近，我们可以得到：

$$\min_c \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \rightarrow 1, \text{ and } \tilde{t}(x) \rightarrow 0$$

然而，在实际生活中，即使是非常晴朗的天气，大气的透射率也不会完全等于 1。所以为了还原自然的图像，需要保留一定的模糊度，我们可以在上面的式子中加入一个系数 $\omega (0 < \omega < 1)$ ：

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

其中， ω 表示去除模糊的程度，在文章中作者选取了 0.95。

4、导向滤波或 Soft Matting 方法优化传导图像

(1) 导向滤波法：

导向滤波的假设是导向图像 I 与滤波输出图像 q 之间是一个线性模型，假设 q 是导向图像 I 在以像素 k 为中心的窗口上的线性变换：

$$q_i = a_k I_i + b_k, \quad \forall i \in \omega_k$$

其中， (a_k, b_k) 是在窗口 ω_k 下的线性系数。同时，假设需要滤波的输入图像记为 p_i ， p_i 为 q_i 叠加噪声后的结果。

为了求出线性系数 (a_k, b_k) ，首先需要定义一个损失函数，评价滤波输出图像与输入图像的相似程度，然后最小化该损失函数，从而求出线性系数。损失函数 $E(a_k, b_k)$ 可以定义为：

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2 \right)$$

其中， εa_k^2 为惩罚项，用于防止过拟合。

通过最小化上式，可以求得：

$$a_k = \frac{\text{Cov}(I_i, p_i)}{\text{Var}(I_i) + \varepsilon}, \quad b_k = \bar{p}_k - a_k \mu_k$$

其中， $\text{Cov}(I_i, p_i)$ 表示 I_i 与 p_i 的协方差， $\text{Var}(I_i)$ 表示 I_i 的方差， \bar{p}_k 是窗口 ω_k 内 p_i 的均值， μ_k 是窗口 ω_k 内 I_i 的均值。

对于窗口的滑动，像素 k 会落入不同的窗口中，从而会有不同的 (a_k, b_k) 系数，因此，需要对包含像素的所有窗口所求得 (a_k, b_k) 的进行平均处理：

$$\bar{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} a_k$$

$$\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$$

(2)Soft Matting 法:

Image matting 通常用马尔科夫随机场(MRF)来得到 alpha matting。在去雾任务中, 也可以通过 MRF 公式来得到优化后的传导图像:

$$E(\mathbf{t}) = \lambda \|\mathbf{t} - \tilde{\mathbf{t}}\|^2 + \mathbf{t}^T \mathbf{L} \mathbf{t}$$

其中, \mathbf{L} 是 matting 的拉普拉斯矩阵, $\tilde{\mathbf{t}}$ 是优化前的传导图像, \mathbf{t} 是优化后的传导图像。

通过拉格朗日乘子法可以得到线性系统如下:

$$(\mathbf{L} + \lambda \mathbf{U}) \mathbf{t} = \lambda \tilde{\mathbf{t}}$$

线性系统一般可以通过 Conjugate Gradients 方法迭代计算得到 \mathbf{t} 。

5、估计原始场景图

根据有雾图像的合成模型公式, 可得图像复原公式为:

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

考虑到当 $t(x)$ 趋于 0 时导致等式失效, 因此, 我们强制给透射率设置一个下界 t_0 。文章中 t_0 取 0.1。

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

将之前得到的传导图像与大气光强代入到上式即可得到复原后的去雾图像。

三、实验成果

3.1 带雾图像暗通道计算结果

将滤波核形状设置为矩形, 滤波核大小设置为 5*5。得到的暗通道计算结果如图 3.1 所示。



(a) rainy1 暗通道计算结果

(b) rainy2 暗通道计算结果



(c) foggy1 暗通道计算结果

(d) foggy2 暗通道计算结果

图 3.1 暗通道计算结果（滤波核大小 5×5 ）

替换最小值滤波的顺序,得到的暗通道计算结果与之前对比如图 3.2 所示(实验中选取 rainy1 与 foggy2 图片进行对比实验)。



(a) rainy1 暗通道计算结果（顺序替换前）

(b) rainy1 暗通道计算结果（顺序替换后）



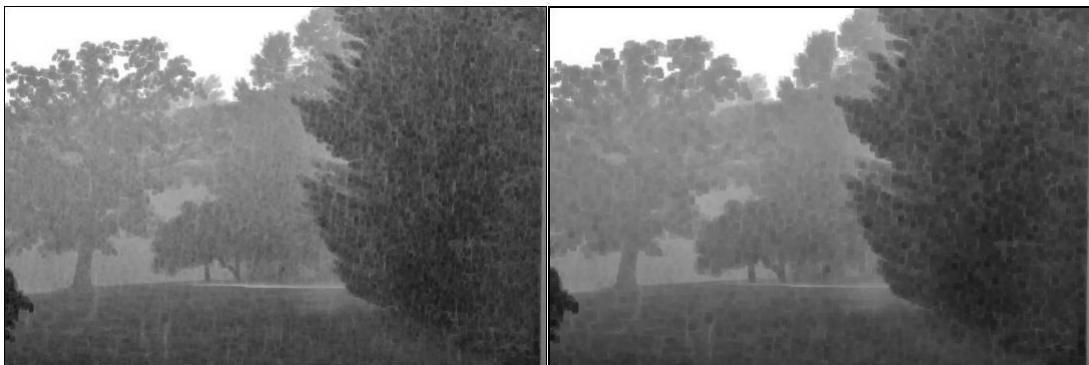
(c) foggy2 暗通道计算结果（顺序替换前） (d) foggy2 暗通道计算结果（顺序替换后）
图 3.2 替换最小值滤波顺序得到的暗通道计算结果对比（滤波核大小 5×5 ）

观察图 3.2 可知，替换最小值滤波的顺序对最终暗通道计算的结果并没有比较大的影响，得到的暗通道计算结果基本一致。

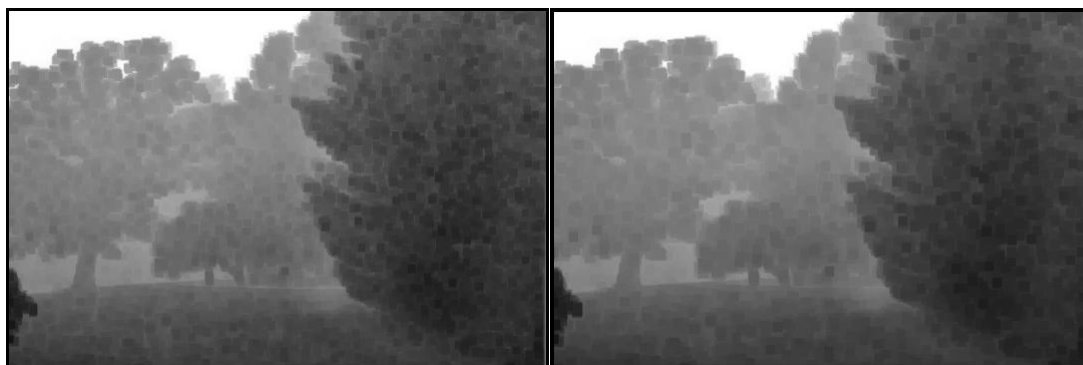
暗通道优先算法旨在利用自然图像中经常观察到的一种统计特性，即在图像的非朦胧区域通常有许多暗像素（即强度值较低的像素）。无论在每个局部斑块中应用最小值滤波器的具体顺序如何，这一特性都是成立的。

出现上述实验结果可能的原因是：虽然调换局部斑块中最小值滤波器的顺序可能会导致个别像素的暗通道值略有不同，但图像清晰区域中存在暗像素的整体统计特性仍然是一致的。因此，在实际应用中，该算法的结果往往是相似的，而且该算法对每个补丁内最小值运算顺序的微小变化具有相对的鲁棒性。

修改不同的最小值滤波半径，得到的暗通道计算结果与之前对比如图 3.3 所示（实验中选取 rainy1 进行对比实验）。

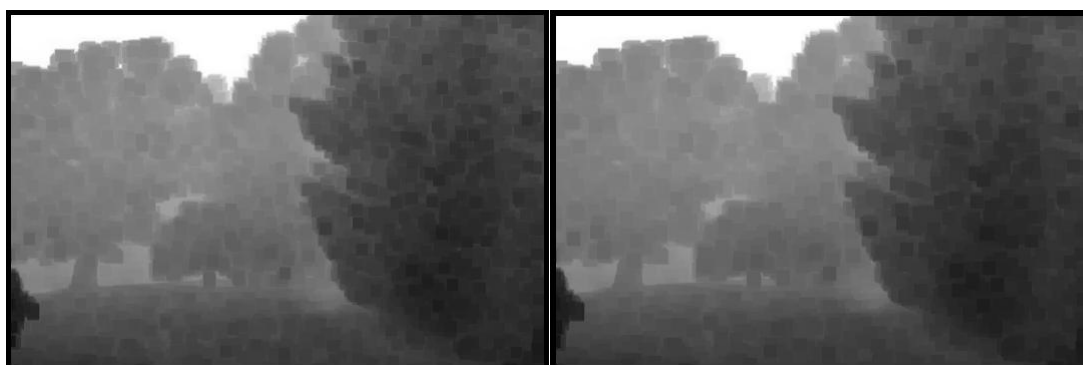


(a) rainy1 暗通道计算结果 (3×3) (b) rainy1 暗通道计算结果 (5×5)



(c) rainy1 暗通道计算结果 (7*7)

(d) rainy1 暗通道计算结果 (9*9)



(e) rainy1 暗通道计算结果 (11*11)

(f) rainy1 暗通道计算结果 (13*13)

图 3.3 修改最小值滤波半径得到的暗通道计算结果

观察图 3.3 可知，滤波半径越小，暗通道图像就可以捕捉到更精细的细节。因为其考虑的是每个像素周围较小的邻域。但也导致对图像中的噪点或微小变化比较敏感；滤波半径越大，暗色通道的计算越全面。它考虑的邻域更大，有助于消除噪点并提供更可靠的雾度估计。但是从图中可以看出越大的滤波半径，图中的细节越模糊。

进一步选取不同的最小值滤波半径观察最终的去雾效果，如图 3.4 所示。



(a) foggy1 去雾结果 (5*5)

(b) foggy1 去雾结果 (21*21)

图 3.4 修改最小值滤波半径得到的去雾结果

观察图 3.4 可知，滤波半径越小，越可能会引入伪影，例如图像中高对比度边缘的晕影或环形效果；滤波半径较大，则可以改善伪影的情况，效果更加平滑。

3.2 传导图像计算结果

得到的传导图像计算结果如图 3.5 所示，系数 ω 的值实验中取 0.95。

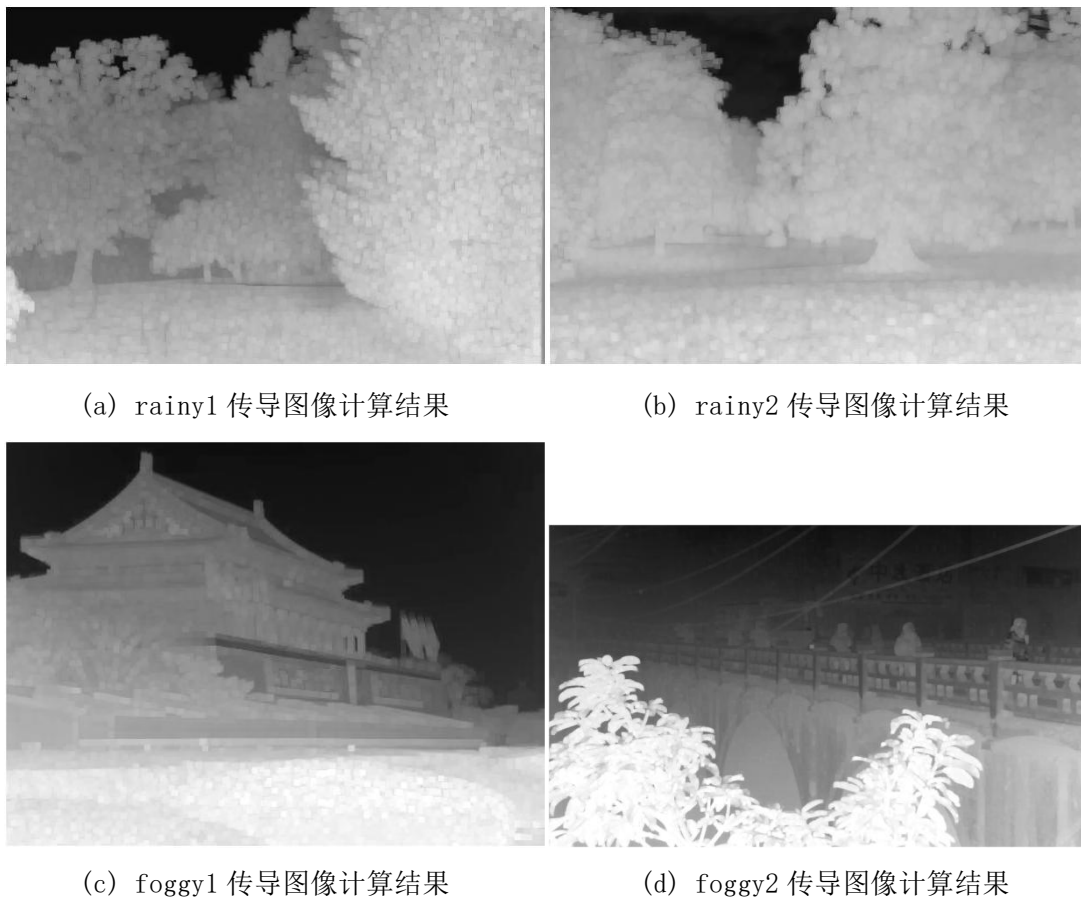


图 3.5 传导图像计算结果（滤波核大小 5*5）

3.3 传导图像优化结果

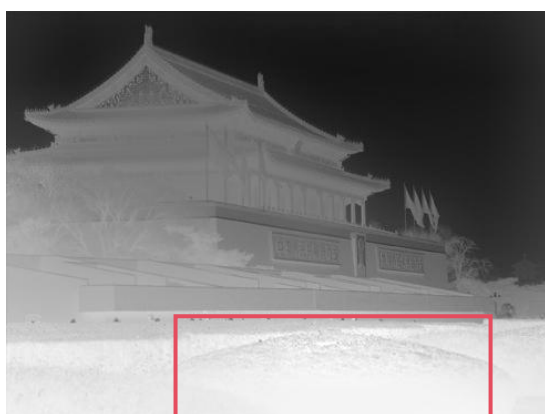
先采用导向滤波的方法进行优化，得到的传导图像结果如图 3.6 所示。



(a) rainy1 传导图像优化结果



(b) rainy2 传导图像优化结果



(c) foggy1 传导图像优化结果



(d) foggy2 传导图像优化结果

图 3.6 传导图像导向滤波优化结果

再采用 Soft Matting 的方法进行优化，得到的传导图像结果如图 3.7 所示。



(a) rainy1 传导图像优化结果



(b) rainy2 传导图像优化结果



(c) foggy1 传导图像优化结果

(d) foggy2 传导图像优化结果

图 3.7 传导图像 Soft Matting 优化结果

对比图 3.6 与图 3.7 可知，在雨天场景下，导向滤波的优化结果要优于 Soft Matting 优化结果。同时，Soft Matting 优化的传导图像保留了更多的细节和边缘，因为其考虑了图像的结构和像素之间的关系；而引导滤波边缘信息相对较少，其主要优势在于保持引导图像和传输图之间的局部色。

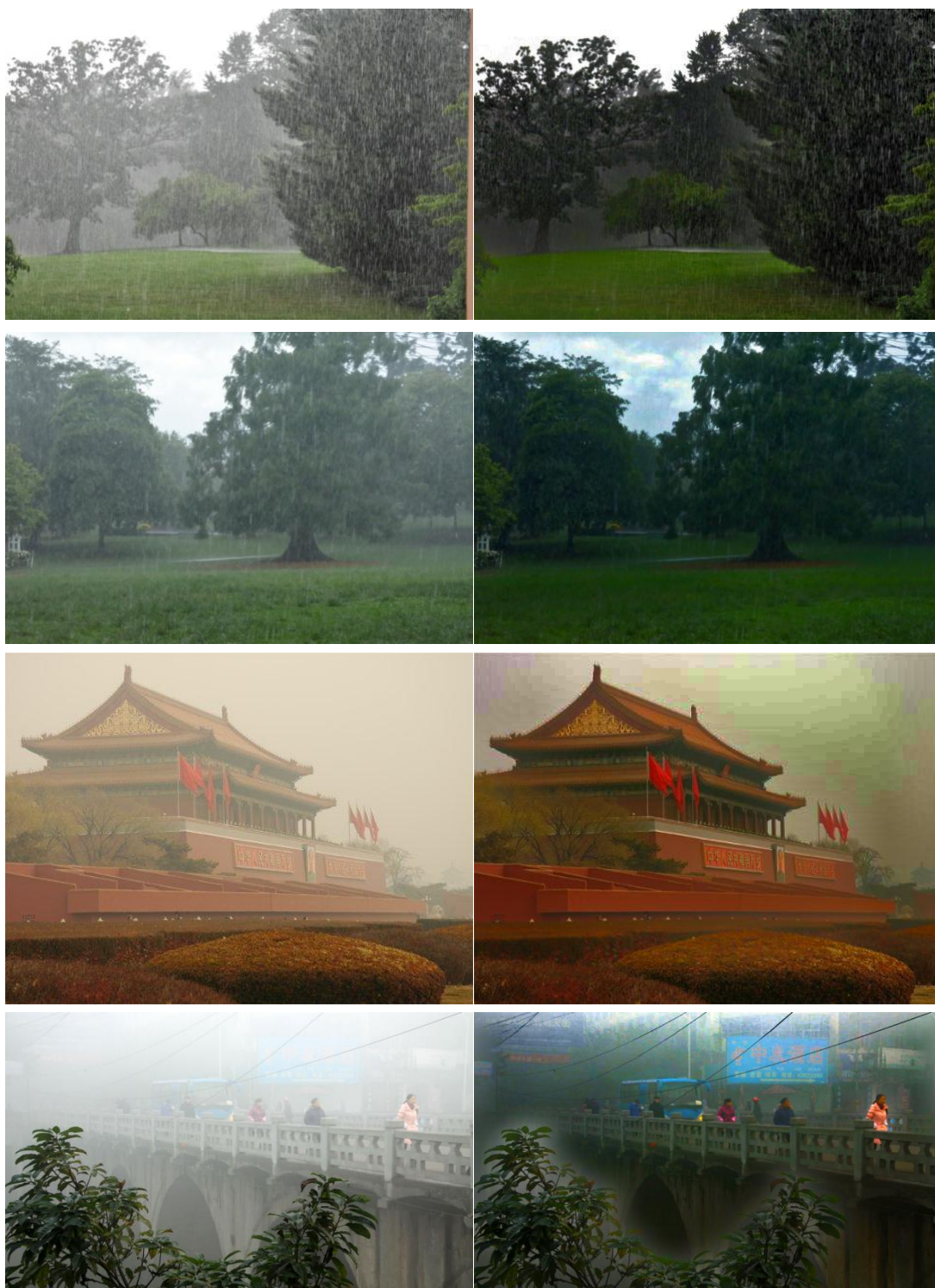
由于求解线性方程组的计算复杂性，Soft Matting 算法运行比导向滤波法慢。因此，导向滤波器通常速度更快，更适合实时或接近实时的应用，两者平均运行时间对比如下表所示：

表 3.1 导向滤波与 Soft Matting 平均用时对比

方法	rainy1 用时	rainy2 用时	foggy1 用时	foggy2 用时
导向滤波	0.020s	0.021s	0.015s	0.035s
Soft Matting	4.42s	4.59s	4.08s	15.22s

3.4 原始场景图估计

根据计算的传导图像恢复原始的场景的图像如图 3.8 所示，传导图像的优化方案为引导滤波。



(a) 恢复前

(b) 恢复后

图 3.8 原始场景图像估计结果（滤波核大小 5*5）

四、实验总结

本实验的目的是复现并评估暗通道先验算法在图像去雾和改善灰度图像可见度方面的有效性。实验中主要使用了 NumPy 和 OpenCV 等库在 Python 中实现了暗通道先验去雾算法。该算法涉及几个关键步骤，包括暗通道计算、大气光估计、透射图估计和去雾。

在暗通道图像计算的步骤，先尝试了替换最小值滤波的顺序，最后发现替换最小值滤波的顺序对最终暗通道计算的结果并没有比较大的影响，得到的暗通道计算结果基本一致，原因是图像清晰区域中存在暗像素的整体统计特性仍然是一致的。接着尝试不同的最小值滤波半径，得出了以下几点结论：

（1）半径越小，暗色通道的计算就越本地化。它考虑的是每个像素周围较小的邻域，可以捕捉到精细的细节，但也可能对图像中的噪点或微小变化比较敏感；半径越大，暗色通道的计算越全面。它考虑的邻域更大，有助于消除噪点并提供更可靠的雾度估计。不过，也可能会模糊图像中的细节。

（2）较小的半径可能会引入伪影，例如图像中高对比度边缘的晕影或环形效果；半径越大，效果越平滑，可以改善伪影的情况，但可能会使细节过度平滑。

（3）计算成本方面，小半径值需要的计算量更少，计算速度更快。适合实时或资源有限的应用；由于邻域规模较大，较大的半径值需要更多的计算。计算成本较高，尤其是对于高分辨率图像。不同滤波半径下暗通道图像计算平均运行时间对比如表 4.1 所示：

表 4.1 不同滤波半径下暗通道图像计算平均运行时间对比

滤波核大小	rainy1 用时	rainy2 用时	foggy1 用时	foggy2 用时
5*5	1.04s	1.01s	0.96s	3.09s
11*11	1.08s	1.06s	0.99s	3.30s
21*21	1.17s	1.13s	1.11s	3.82s
41*41	1.53s	1.51s	1.41s	4.68s
61*61	1.97s	2.01s	1.85s	5.99s

（4）较小的半径值可能对雾霾和小尺度场景特征更敏感。可能更适用于具有明显局部雾度的图像，例如具有强烈光散射的场景；滤波半径值越大，对局部雾霾的抗干扰能力越强，但可能无法很好地捕捉到精细的场景细节，更适用于雾霾比较均匀或弥漫的图像。

因此，暗通道先验算法中最小滤波半径的选择取决于输入图像的具体特征，以及在保留精细细节和实现稳健的雾度去除之间所需的权衡。在实践中，通常会尝试不同的半径值，以找到最适合特定图像或数据集的值。该算法的某些实现方法还可以使用自适应方法，根据本地图像内容调整滤波半径。

在大气光强估计的步骤，需要注意不能简单地将最亮的像素点作为大气光强。应该首先选择在暗通道图像中前 0.1% 的像素，由暗通道先验条件保证了挑选出来的像素不会受到自然图像景物的干扰。然后找到在原始图像中 I 中对应的这 0.1% 的像素每个通道的最亮点。

在计算传导图像的步骤，注意需要在算法中引入去除模糊的程度参数 ω 。

在优化传导图像的步骤，实验中尝试了导向滤波算法与 Soft Matting 算法，得出了以下几点结论：

(1) 在雨天场景下，导向滤波的优化结果要优于 Soft Matting 优化结果。

(2) Soft Matting 倾向于有效地保留传输图中的细节和边缘，它考虑了图像的结构和像素之间的关系；导向滤波也能在一定程度上保留边缘，但其主要优势在于保持引导图像和传输图之间的局部色彩一致性。

(3) 由于求解线性方程组的计算复杂性，Soft Matting 算法运行比导向滤波法慢；导向滤波器通常速度更快，更适合实时或接近实时的应用。

(4) Soft Matting 在某些情况下可能会产生伪影，尤其是在没有仔细调整参数的情况下；导向滤波器处理伪影和产生更平滑效果的能力更强。

因此，Soft Matting 和导向滤波都能有效改善暗通道除雾方法的传导图像。两者之间的选择取决于计算资源、所需的控制水平以及所处理图像的具体特征等因素。Soft Matting 可以提供更多的控制和细节保留，而导向滤波器则更简单、高效和对伪影具有稳健性。

最后，根据有雾图像的合成模型公式，得到恢复后的图像。然而，恢复后的图像仍然存在一些问题，如图 3.8 中 foggy1 的去雾图像存在大片的伪影，在调整滤波半径后仍然存在一部分；图 3.8 中的 rainy1 与 rainy2 图像中仍然存在雨点噪点。因此，算法可能还需要一些后处理操作以及进一步的优化。