

暗通道去雾方法简介：

根据先验知识，我们可以知道，在绝大多数非天空的局部区域里，某一些像素总会有至少一个颜色通道具有很低的值，该区域的光强度具有很低的值。我们定义暗通道：

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

其中暗通道的值主要由以下的景物决定：

- (1) 汽车、建筑物和城市中玻璃窗户的阴影，或者是树叶、树与岩石等自然景观的投影
- (2) 色彩鲜艳的物体或表面，在 RGB 的三个通道中有些通道的值很低（饱和度高）
- (3) 颜色较暗的物体或者表面，例如灰暗色的树干和石头

首先，在计算机视觉和计算机图形中，下述方程所描述的雾图形成模型被广泛使用：

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

$\mathbf{I}(\mathbf{x})$ 为已有的图像（待去雾的图像）， $\mathbf{J}(\mathbf{x})$ 是我们要恢复的无雾的图像， \mathbf{A} 是全球大气光成分， $t(\mathbf{x})$ 为透射率。现在的已知条件就是 $\mathbf{I}(\mathbf{x})$ ，要求目标值 $\mathbf{J}(\mathbf{x})$ 。因此我们只需要估算出 $t(\mathbf{x})$ 与 \mathbf{A} 即可。

假设 \mathbf{A} 已知，则 $t(\mathbf{x})$ 的值可以通过下列先验知识得到：

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

其中 \mathbf{A} 的值可以通过下列方法估算得到：

- (1) 从暗通道图中按照亮度的大小取前 0.1% 的像素
- (2) 在这些位置中，在原始有雾图像 \mathbf{I} 中寻找对应的具有最高亮度的点的值，作为 \mathbf{A} 值

当透射 t 值很小时，会导致 \mathbf{J} 的值偏大，从而使得图像整体偏白，因此一般可设置阈值 t_0 ，当 t 值小于 t_0 时，令 $t=t_0$ ，可以避免这样的问题，因此，最终的求解方法如下：

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

暗通道去雾方法实现详解：

(1) 首先得到该图像的暗通道，对输入的一张图像每个像素点取该像素点 RGB 最小值，然后对得到的新的灰度图进行最小值滤波，得到暗通道的灰度图，该功能由程序中的 `getDark` 函数实现

(2) 然后得到估计的 \mathbf{A} 的值，即在暗通道中按亮度大小取前 1/1000 的像素点，然后得到该像素点集的坐标，最后在原图中取该像素点集中亮度最亮的像素点的 RGB 值作为 \mathbf{A} 的值，

该功能由程序中的函数 `getLight` 实现

(3) 然后对 $t(x)$ 的值进行估计，得到整张图像的透射率的值。即对原图中的每个像素点的 RGB 的最小值，除以 A 值中的 RGB 的最小值，得到一个比值。对该图像所有的像素点进行操作后，对新的图像进行最小值滤波，然后乘以一个调整系数 ω ，最后用 1 减去该值即可得到整张图像 $t(x)$ 的值，该功能由程序中的函数 `getTransmission` 实现

(4) 最后根据公式
$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A$$
，即可求得去雾后图像的值，该功能程序中的函数 `getRadiance` 实现

暗通道去雾方法图像实例：

(1) 去雾前与去雾后图像对比如下：







通过上述的图像去雾情况表明，暗通道能够对图像实现很好的去雾效果，但是，有一些图像的去雾效果并不是很理想，而且，没有用 **soft matting** 的图像中，有一些地方的边缘并不是很协调，有一些边角并不能够将雾霾很好地去掉。

同时，对于一些极端的情况，比如场景对象与大气光有相似性的时候，或者遇到雪白的雪地

等没有影子的时候，这个时候暗通道就会失效，因此，当碰到这种情况的时候，不能够通过暗通道来进行处理。这个时候可以通过别的去雾方法对该图形进行处理，从而避免了暗通道所带来的局限性。

Soft matting（导向滤波）去雾方法：

通过上述情况可以发现，有一些图像的边缘部分去雾的时候，会出现一些没有去掉的情况。这个时候的原因是我们的 $t(x)$ ，即透射率的图像不够精确造成的，因此，我们通过导向滤波来实现个别边缘的修正，使图像去雾效果更加完美。

导向滤波实际上是对已经实现的 $t(x)$ 图像进行修正，使得 $t(x)$ 更加精细，从而使去雾能够更加精确，还原的图像更加逼真

在图像中，像素点的滤波结果是被表达成一个加权平均：

$$q_i = \sum_j W_{ij}(I)p_j$$

假设导向滤波器在导向图像 I 和滤波输出 q 之间是一个局部线性模型：

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

代价函数为：

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

其中 a 的值为：

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}$$

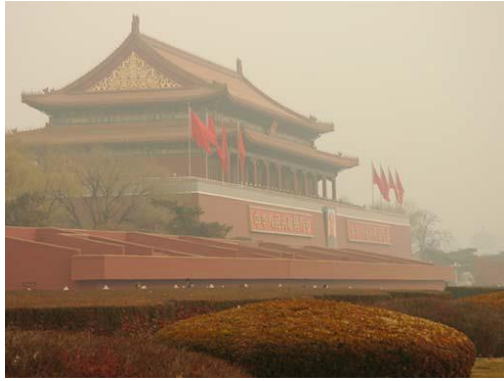
b 的值为：

$$b_k = \bar{p}_k - a_k \mu_k$$

首先定义 `boxfilter`，它能够快速计算任意大小的矩形求和运算，在程序中由函数 `boxfilter` 实现，然后根据上面的公式得到原图像的均值与协方差，最后通过求出 a, b ，从而求得精确的 $t(x)$ ，是其模型能够更加精确，在本程序中由函数 `guidedFilter` 实现

Soft matting（导向滤波）实验结果如下：

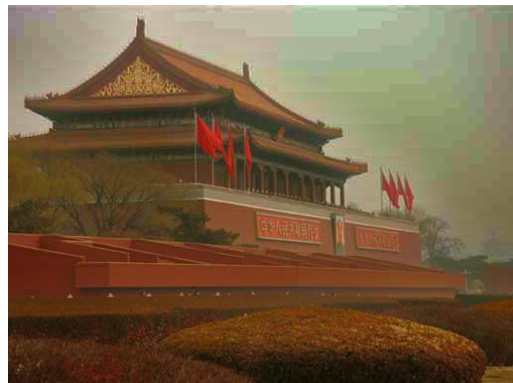
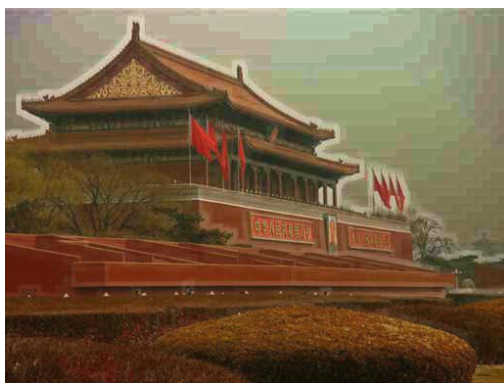
（1）原图如下：



原图与导向滤波后的图片的 $t(x)$ 透射率图对比如下：



无导向滤波与导向滤波后的图片的去雾图对比如下：



(2) 原图如下：



原图与导向滤波后的图片的 $t(x)$ 透射率图对比如下：



无导向滤波与导向滤波后的图片的去雾图对比如下：



(3) 原图如下：



原图与导向滤波后的图片的 $t(x)$ 透射率图对比如下：



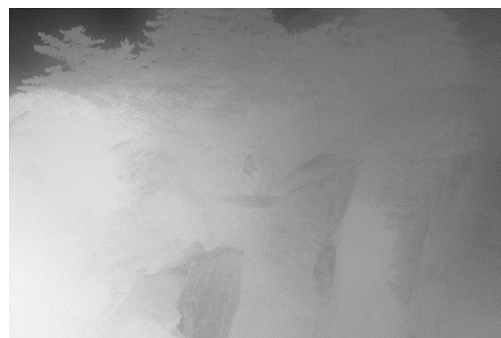
无导向滤波与导向滤波后的图片的去雾图对比如下：



(4) 原图如下：



原图与导向滤波后的图片的 $t(x)$ 透射率图对比如下：



无导向滤波与导向滤波后的图片的去雾图对比如下：



通过上面图像处理可以发现，导向滤波能够得到更精细的透射率图，从而让去雾效果更加好，但是，对于大面积的天空等，去雾效果不是很理想，还原的图像还有一些不完美的地方。总之，导向滤波能够让去雾实现更好的过渡，使图像更加自然