Haze Removal

1. Introduction

在计算机视觉与计算机图形学中，一般用下面的式子描述有雾图像

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

其中 **I** 代表观察到的图像（即输入的有雾图像），**J** 代表去雾之后的图像，**t** 代表透射率，**A** 代表全球大气光成分。然后我们的去雾模型的构建就是基于这个表达式。然后，作者提出来暗通道先验这个理论：对于一个像素点而言就是 RGB 分量重最小的那个通道，然后在一定大小窗口中做最小值滤波，即取此窗口内各点 RGB 分量最小值得最小值，就找到了暗通道。作者通过对随机选取的 5000 张图片（手动裁剪掉天空部分）进行分析，发现对于没有雾的图像，他们的暗通道趋于全黑（即值趋于 0）；而有雾的图像，暗通道则是相对较亮的。既有如下公式：

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

J^{dark} 表示 X 点处的暗通道值，Ω(X)为以 X 为中心的窗口，C 表示 RGB 分量。因此，由 $J^{dark} \rightarrow 0$ 经过推到和固有经验，可以求出式(1)中未知的变量，从而求出去噪后的图像 **J(X)**.

根据论文中的推理，我们可以得到以下公式：

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

这里 ω 用来根据不同图形进行微调，其取值区间为(0,1)

A 的值可以根据暗通道的值估计出来，方法是取暗通道中亮度在前 0.1%的像素点，然后选取这些像素点中在对应原图中强度最大的点的 RGB 值，作为 A 的值。

最终，我们得到下面的公式：

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

这里 t_0 是阀值，用来限制 t(X)不会太小，一般取 0.1

.这样，一幅去噪之后的图像就生成了，但是这样的做法，由于暗通道和透射率视图不够精细，所以会在生成后的图像中出现明显的边缘现象（具体

现象见实验结果图），于是论文中提出了 soft matting 的方法对透射率图进行优化，使其更加精细。但是这种方法的算法比较复杂，而且复杂度高，图像处理时间长，所以，这里我们使用该作者另一篇论文介绍的 Guided Image Filter 的方法进行来优化透射率视图。具体公式如下：

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

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

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

其中 q_i 代表滤波之后输出的图像，I_i 代表输入图像，ω_k 为以 K 为中心的窗口，μ_k 是在窗口中 I 的均值，σ^2_k 是在窗口中 I 的方差。

算法：

Algorithm 1. Guided Filter.

Input: filtering input image p, guidance image I, radius r
regularization ϵ

Output: filtering output q.

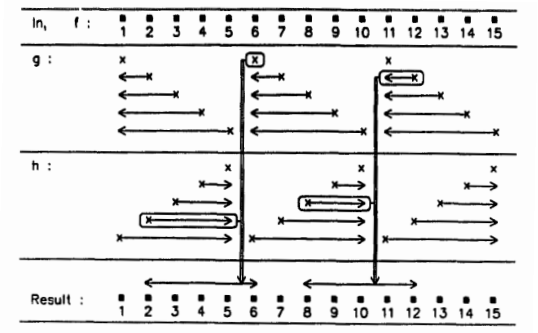
1: $\text{mean}_I = f_{\text{mean}}(I)$
$\text{mean}_p = f_{\text{mean}}(p)$
$\text{corr}_I = f_{\text{mean}}(I. * I)$
$\text{corr}_{Ip} = f_{\text{mean}}(I. * p)$
2: $\text{var}_I = \text{corr}_I - \text{mean}_I. * \text{mean}_I$
$\text{cov}_{Ip} = \text{corr}_{Ip} - \text{mean}_I. * \text{mean}_p$
3: $a = \text{cov}_{Ip}./(\text{var}_I + \epsilon)$
$b = \text{mean}_p - a. * \text{mean}_I$
4: $\text{mean}_a = f_{\text{mean}}(a)$
$\text{mean}_b = f_{\text{mean}}(b)$
5: $q = \text{mean}_a. * I + \text{mean}_b$

/* f_{mean} is a mean filter with a wide variety of O(N) time methods. */

这里使用了一种与窗口半径无关的均值滤波算法，大概思想是利用已经算过的和，减去前一个，加上后一个来进行计算（分行列计算），这里不详细阐述。

2. Implementation details

首先，求出图像的暗通道值，首先循环求出原图每一点 RGB 最小值通道，然后对整幅图使用最小值滤波，即可求出原图的暗通道图像。这里的使用了最小值滤波的快速算法，即用两个辅助数组存储前后的最小值，实现与滤波半径无关的算法，这个算法的复杂度是 O(n)的（对于整幅图片而言）。如图：



然后根据论文中的原理，取暗通道中亮度前 0.1%的像素点，找到其中在原图中强度最大的值作为 A 的值。一开始，我的第一反应时对暗通道中所有的值按照亮度由大到小排序，然后取出前 0.1%的值。这样算法的复杂度为 O(nlg(n))。后来我想起来好像可以用堆来实现：即我先读取暗通道的 0.1%个值构建一个最小堆，然后每次读取暗通道中下一个数据，如果小于堆顶值，则不做任何操作，循环继续；如果大于堆顶值，则替换堆顶值，维护堆，这样最后堆中的数据就是所有数据中最大的 0.1%个。这里堆中的每一个节点是一个类，保存了当前取出像素的值，以及对应的坐标。然后遍历这 0.1%个节点，找到在原图中强度最大的那个，即求出了 A 的值。这样算法的复杂度大概是 O(nlg(k))k 为 0.1%个暗通道点的数量。

接下来是求透射率图。首先遍历一次全部像素，求出每个点三个通道与 A 对应通道值商的最小值，然后对求得的结果进行最小值滤波，然后将滤波结果标定到[0,1]之间，然后对每个点乘以 ω 值，并用一减去其积。这样就求得了没有经过优化的透射率视图。下一步就是去用导向滤波求更加精细的透射率图。导向滤波的算法前面已经详细阐述过了，这里不再赘述。要提两点，一是导向性滤波中，所有耗时的过程都在均值滤波上，而在使用了与窗口半径无关的算法后，使得整个算法控制在 O(n)的全局复杂度，并没有增加整个去雾算法的复杂度，所以用了取代 soft matting 是十分合适的；二是用作导向性图的图片，可以选择原图或者原图的灰度图，后者比较简单，这里使用的是后者，至于彩色图传灰度图，这里用的是 R*0.3+G*0.59+B*0.11 的算法，然后再进行滤波之前要将其标定到[0,1]。

这样我们就计算好了 A（全球大气光含量）值和 t(x)（透射率图）值，已经可以计算去雾之后的图像了。具体方法是循环遍历整幅图片，用原图

的每个点分别减去 A 的值（这里是 RGB 对应相减），然后除以 t(x)和 t0 的最大值，最后加上 A 的值（这里是 RGB 分别相加），然后对小于 0 的通道，直接复制为 0，大于 255 的直接赋值为 255。这样就完成了对整幅图片的去雾处理。

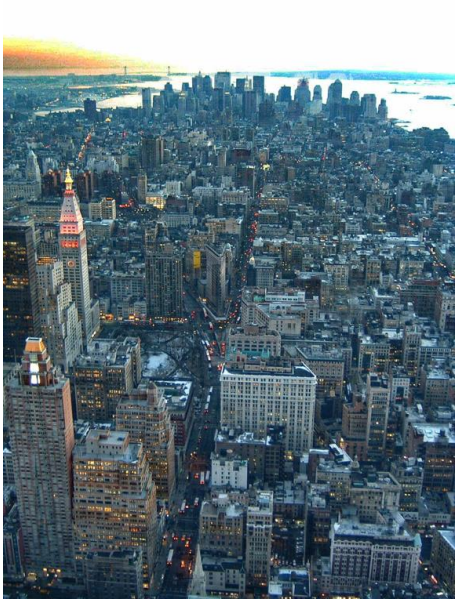
论文中有一句话指明，在做完去雾处理后，图像会变暗，因此可以进行曝光补偿。一开始我是用的直方图均衡化，图像确实变亮了很多，但是边缘部分明显出现失真的情况，后来在网上查到一种叫自动色阶的算法，使用之后，虽然还会有失真，但是效果好了很多，可以根据具体图片来使用。

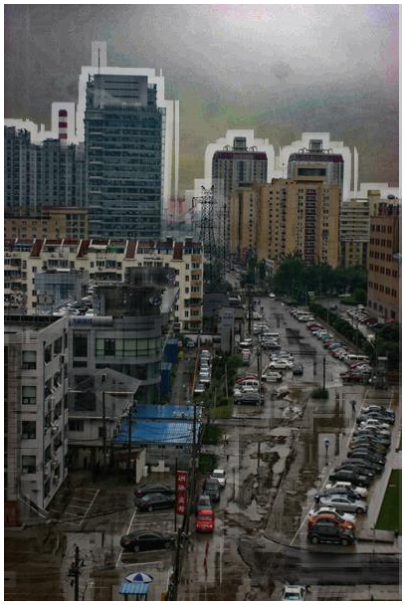
3. Results and analyses

未使用导向滤波优化透射率的去雾结果：









注：这里为了文档的显示效果缩小了图片，原图请查看文件夹“未用导向滤波\去雾图”

上面这几张图片是没有用导向性滤波来优化，因此我们可以明显在物体的边缘看到模糊的方块形状的半透明现象。这是因为此时的透射率图是相对比较粗糙的，没有处理好边缘的情况，所以就会出现这种边缘现象。

使用导向性滤波之后的效果：



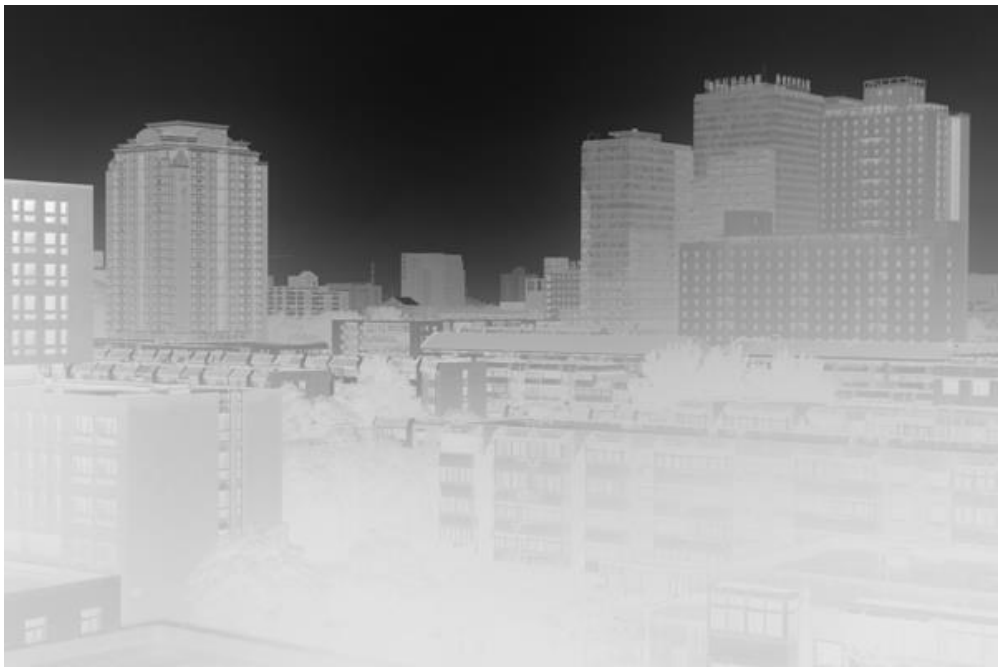


注：这里为了文档的显示效果缩小了图片，原图请查看文件夹“使用导向滤波\去雾图”

这些图片，由于使用了导向性滤波，图片变得很光滑，即没有明显的边缘现象的出现。这里还有一点需要注意的就是 A 的值，是要根据不同图片进行不同的调整的，若始终使用同一个值，有些图片的效果会很差。

对比透射率图：

未使用导向滤波的透射率图



使用导向滤波的透射率图

明显在使用导向性滤波后图像清晰了很多。具体全部透射率图可在文件夹“未用导向滤波\透射率图”和“使用导向滤波\透射率图”中查看到

4. Disadvantages and improvements

作为一个学渣，能基本实现人家教授的方法，已经“元气大伤”，再要去找人家的不足，真的很难，只能是谈谈拙见。

1、 在计算透射率的时候，是先算出每个点与 A 的商值得最小值，然后进行最小值滤波。这里我认为可以用最小值滤波的结果去除以 A 的值，应该可以得到近似的结果（由于时间紧迫，并没有进行实验验证相似度），这样就可以使用前面计算好的暗通道的值来直接除以 A,从而省去了对 A 做最小值滤波的时间，加快了图像的处理速度。

2、 在对 A 值进行估计的时候，本来只是估计一个参数的值，却成为真个算法中复杂度最高最耗时的部分，所以我在想既然 A 是估计值，那么有没有更好的更快的方法去估算 A 呢（何况文中的方法估算出的 A 的值也不能很好的适用于所有的图片），我想过可不可以对暗通道进行按一定比例缩小，然后从缩小的图中找到前 0.1% / 缩放比例的亮度的值，然后将这些值得坐标按照比例缩放回去，获取 0.1%的值中对应原图中最大强度的值得 RGB 值。

3、 和第二点一样，在获取原图的透射率图中，是否也可以使用原图按比例缩小的图片来进行计算，从而提高效率呢？

4、 以上说的都是关于提高算法效率的方法，但是对于如何提高去雾图像的质量，一直没有想法，但是应该可以再去雾之后使用一些增强图片对比度和亮度的方法来是完善去雾的效果吧。