

基于暗通道优先的去雾算法

09014325 孙新凯
09014323 郑思豪
09014333 黄子尧

关键词：

暗通道 去雾

去雾被广泛地应用于计算机视觉与图像处理等方面，但是去雾通常是一个很有挑战的问题。基于何恺明教授的 *Single Image Haze Removal Using Dark Channel Prior*，我们采用暗通道优先的方法来识别雾区，因为根据统计发现，对于没有雾的图片(去除掉天空)，他rgb三个通道中至少有一个值会比较的低，所以根据这种特性，采用了暗通道方式进行去雾处理。

1、原论文算法

1.1 暗通道

论文中提出的暗通道优先算法是基于以下观察规律：对于一张无雾的户外图像，在大部分非天空区域，至少有一个色彩通道在某些点有着很低的像素值。下面给出暗通道的定义：

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

上式中 J^c 表示彩色图像的每个通道， $\Omega(x)$ 表示以 x 为中心的一个窗口。根据暗通道理论，对于无雾图像， J_{dark} 趋近于0。

1.2 估计大气光值

之前说过暗通道值比较高的位置，比较有可能是雾比较大的位置，这些地方可能是天空等，这时，选取暗通道中最高的0.1%的点，然后从原图中挑选这些像素点中亮度最高的点作为相应的大气光值。

1.3 估计透射率

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：

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

其中， $I(x)$ 就是我们现在已经有的图像（待去雾的图像）， $J(x)$ 是我们要恢复的无雾的图像， A 是全球大气光成分， $t(x)$ 为透射率。现在的已知条件就是 $I(x)$ ，要求目标值 $J(x)$ 。根据论文中的推导，可以得到如下的结果：

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

在现实生活中，空气中存在一定的微粒，因此看远处的景物仍然能感觉到雾的存在。此外，雾也使人们感觉到景深的存在。因此有必要在去雾时保留一定的雾。可以通过引入值在[0,1]之间参数 ω 来实现。因此最终的透射率估算公式如下：

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

1.4 软磨砂

之前得到的估计透射率发现，只是一个一个比较明显的像素块，用这种的透射率来计算会导致比较明显的色区，所以根据之前的理论，计算了Matting Laplacian matrix。

$$L = \sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} (1 + (I_i - \mu_k)^T \left(\sum_k + \frac{\epsilon}{w_k} \right)^{-1} (I_i - \mu_k)) \right)$$

然后根据下面的式子

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

重新计算透射率 t

1.5 还原图像

根据常规的公式直接可得到原图。

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

这里之所以使用 $\max(t(x), t_0)$ 是为了避免 $t(x)$ 趋近于0而造成的问题。

1.6 实验结果



1.7 问题分析

大部分的实验结果表现比较好，但是发现有些雾比较厚的天空部分就会出现一些明显的过渡区域，但是天空的颜色应该会比较均匀，不会有这么明显的过渡，这个是比较糟糕的。上述情况的图片还有一个问题，那就是图片偏暗，这个情况在何恺明教授的论文中提到过这个情况，但是并没有给出比较详细的说明。另外就是soft matting这个方法导致一张600×400的图片处理可能要达到40s——50s的时间，处理速度是比较慢的。

2、改进

基于上面的问题，我们增加了曝光，然后对于天空部分统一进行处理，最后采用新的方法来取代原来的soft matting的方法，以提高效率。

2.1 增强曝光

为了提高曝光，我们在这使用简单的MATLAB中的方法，增加了曝光和对比度，让有些图片看上去不在那么暗淡。

```
recover_img = recover_img + (1-recover_img) .* recover_img * 0.7;  
recover_img = imadjust(recover_img, [0.1 0.83], []);
```

2.2 天空处理

对于有天空的图片，我们认为他们的亮度应该是一致的，但是原文中并没有做出很好的处理，所以这边我们设置了一个阈值K，用原图的像素与我们得到的大气光照的像素进行对比，当大于等于这个部分的像素，按统一进行处理，这样可以明显改善天空部分的过渡情况。

```
J(:, :, i) = A(i) + (images(:, :, i) - A(i)) ./ min(max(K./abs(images(:, :, i) - A(i)), 1) .* max(trans, t0), 1);
```

2.2 导向滤波

针对原来处理比较慢的情况，何恺明教授在后边提出了一个比较的方法**Guided Image Filtering**。

导向滤波是一种图像滤波技术，比软磨砂效率高很多，并且能达到相似的效果。通过一张引导图G（本实验中为原图像），对目标图像P（本实验中为初始的透射率图像）进行滤波处理，使得最后的输出图像在像素值上与目标图像P相似，但是纹理上与引导图G相似。论文所给的算法如下：

输入：目标图像 p ，导向图 I ，窗口宽度 w ，调整系数 ϵ ，按照abcbe步骤操作。

输出：滤波后的图像 q

a)

$$mean_I = f_{mean}(I)$$

$$mean_p = f_{mean}(p)$$

$$corr_I = f_{mean}(I .* I)$$

$$corr_{I_p} = f_{mean}(I .* p)$$

b)

$$var_I = corr_I - mean_I .* mean_I$$

$$cov_{I_p} = corr_{I_p} - mean_I .* mean_p$$

c)

$$a = cov_{I_p} ./ (var_I + \epsilon)$$

$$b = mean_p - a .* mean_I$$

d)

$$mean_a = f_{mean}(a)$$

$$mean_b = f_{mean}(b)$$

e)

$$q = \text{mean}_a .* I + \text{mean}_b$$

其中 f_{mean} 表示均值滤波。可以通过方框滤波来实现快速均值滤波，复杂度仅为 $O(n)$ 。

2.3 实验结果对比

