

基于单幅图像的快速去雾算法

刘倩, 陈茂银, 周东华

清华大学信息学院自动化系, 北京 100084

E-mail: zdh@mail.tsinghua.edu.cn

摘要: 雾霾等天气会导致室外图像可视性降低, 进一步导致室外图像处理系统性能下降. 本文给出一种基于单幅彩色或灰度图像的快速去雾算法. 基于物理模型分析, 仅利用均值滤波对环境光和全局大气光进行估计. 本文算法简单有效, 能够用于实时计算. 实验表明, 与其他算法相比较, 结果具有很好的可视性.

关键词: 去雾, 实时, 物理模型, 可视性

Fast Haze Removal from a Single Image

Qian Liu, Maoyin Chen, Donghua Zhou

Department of Automation, Tsinghua University, Beijing 100084, P. R. China

E-mail: zdh@mail.tsinghua.edu.cn

Abstract: Images of outdoor scenes show poor visibility in presence of haze. This results in poor performance of many outdoor images progressing systems. In this paper, we present a fast haze removal method using a single color or gray level image. Based on the analyses of physical-based model, we estimate airlight and atmospheric light utilizing the simple average filter. Our method is very simple and efficient and achieves real-time computation. Experiments demonstrate that our results achieve as good visibility as a few state-of-art algorithms'.

Key Words: Haze removal, Real-time, Physical-based model, Visibility

1 引言

在视频监控, 导航控制, 目标跟踪, 遥感和车辆自主驾驶等技术领域, 都涉及到图像处理步骤. 这些室外图像处理系统的性能极易受到恶劣天气的影响. 在雾, 霾等恶劣天气下, 物体的反射光会受到大气中悬浮颗粒的吸收和散射, 致使被获取的图像可视性差, 细节模糊, 色彩淡化, 严重影响后续图像处理结果. 因此, 图像去雾算法的研究非常具有实际意义. 很多实际应用系统对实时性要求极高, 快速去雾算法的研究极为重要.

很多学者对雾天图像清晰化算法进行了研究. 如, Narasimhan在[1]中提出一种基于物理模型的去雾算法, 该算法从同一场景在不同天气条件下的多幅图像中获取景物的结构信息; 在文[2]中, Shwartz等人利用多幅偏振图像来进行环境光的估计; 文[3]给出一种基于用户交互的算法; 文[4]利用已知的景物深度模型进行去雾. 尽管这些算法都获得了很好的去雾效果, 但是都需要输入单幅图像以外的信息, 使得实际应用受限.

在基于单幅图像的去雾算法中, Tan[5]以增强图像对比度为目的, 提出了一种同时适用于灰度和彩色

图像的去雾算法. Fattal在[6]中基于景物表面阴影与介质透射率具有局部统计不相关性这一假设, 提出一种只能处理彩色图像的算法. He[7]利用暗原色先验来求取透射率, 然后利用软抠算法进行修正, 最终恢复出无雾图像. 上述算法都取得了极好的去雾效果, 其中He的算法去雾效果尤为显著. 但这些算法的去雾速度都不够理想, 如, 文[5]处理大小为 600×400 的图像需要大约5到7分钟(double processors of Pentium 4 and 1 GB memory), [6]处理一幅 512×512 的图像时用时约35秒(C++, a 1.6 GHz Intel Pentium dual core processor), [7]的算法处理一幅 600×400 像素的图像需要大约10到20秒(a 3.0 GHz Intel Pentium 4 Processor). Tarel[8]通过对环境光取值范围的分析, 提出了一种快速去雾算法, 该算法对图像进行白平衡后, 利用多次中值滤波估计环境光. 但该算法的运算速度也有待进一步提高.

很多新的快速去雾算法源于对文[7]和文[8]中算法的改进. 文[9]先求出部分像素点的透射率值, 然后利用多元线性回归求出整幅图像的透射率, 以加快算法速度. 但这种方法求出的透射率较为平滑, 只适用于景深变化平缓的特定应用场合, 处理景物深度呈现剧烈跳变的图像则效果较差. 文[10]利用快速双边滤波对透射率进行细化, 代替文[7]中的软抠步骤. 文[11]对图像进行白平衡后, 求取颜色通道最小值作为环境光的粗估计, 然后进行边界保持滤波来细化环境光.

此项工作得到国家973项目(项目编号: 2010CB731800, 2009CB320602)和国家自然科学基金(项目批准号: 61210012, 61021063, 61290324)资助.

上述方法中的滤波器仍然较为耗时,快速性有待提高.本文给出的算法将仅利用一次简单的均值滤波来估计环境光和全局大气光,具有更为快速的执行速度.本文算法是受到文[8]中对环境光取值范围分析的启发而提出的,但本文算法与文[8]相比在快速性方面有极为显著提高,同时,本算法的去雾效果与文[8]的去雾效果相当,甚至更优于文[8]的去雾效果.

本文旨在保证一定去雾效果的基础上,尽可能降低去雾算法的复杂度,使算法能够满足实时应用系统的要求.本文将通过对雾天图像退化物理模型的分析,逐步分析出透射率的估计方法,仅利用一次简单的均值滤波,给出环境光和全局大气光的估计方式.且本算法能够自动调节去雾后图像的亮度,防止去雾后图像出现过度偏暗或偏亮的情况.

2 本文的算法

2.1 雾天图像退化物理模型

在计算机视觉领域,通常用下述物理模型来描述雾霾等恶劣天气条件对图像造成的影响.该模型包括衰减模型和环境光模型两部分,模型表达式为

$$H(x) = F(x)e^{-rd(x)} + A(1 - e^{-rd(x)}) \quad (1)$$

其中, x 为图像像素的空间坐标; H 是观察到的有雾图像(Haze image); F 是待恢复的无雾图像(Haze-free image); r 表示大气散射系数; d 代表景物深度; A 是全局大气光(Atmospheric light),通常情况下假设为全局常量,与空间坐标 (x) 无关.

式(1)中, $F(x)e^{-rd(x)}$ 被称为直接衰减(Direct attenuation),表示景物表面的反射光在介质中传播时因散射等作用而衰减; $A(1 - e^{-rd(x)})$ 被称为环境光(Airlight)或大气光幕(Atmospheric veil),它导致图像色彩和亮度的偏移.

对于式(1),用 $L(x)$ 代替 $A(1 - e^{-rd(x)})$,即

$$L(x) = A(1 - e^{-rd(x)}) \quad (2)$$

则式(1)被改写为

$$H(x) = F(x) \left(1 - \frac{L(x)}{A} \right) + L(x) \quad (3)$$

可见,我们需要从图像 $H(x)$ 中估计出环境光 $L(x)$ 和全局大气光 A ,以恢复出清晰的无雾图像 $F(x)$.

2.2 估计环境光

用介质透射率(Medium transmission) $t(x)$ 来替换指数衰减项 $e^{-rd(x)}$,即

$$t(x) = e^{-rd(x)} \quad (4)$$

则式(1)简化为

$$H(x) = F(x)t(x) + A(1 - t(x)) \quad (5)$$

由式(5),得出

$$A(1 - t(x)) \leq H(x) \quad (6)$$

对式(6)两侧求取三个颜色通道的最小值,并记

$$M(x) = \min_{c \in \{r, g, b\}} (H^c(x)) \quad (7)$$

$$A = [A_o \quad A_o \quad A_o]^T \quad (8)$$

则式(6)改写为 $A_o(1 - t(x)) \leq M(x)$,即

$$t(x) \geq 1 - \frac{M(x)}{A_o} \quad (9)$$

由式(4)知, $t(x)$ 随着景物深度 d 的增大而减小. $t(x)$ 值越小表示像素 (x) 处景物反射光衰减的程度越大.当 $t(x)$ 为0时,表示景物反射光衰减为0,此时 $H(x) = A$,当 $t(x)$ 为1时,表示景物反射光完全没有衰减,此时 $H(x) = F(x)$.透射率 $t(x)$ 具有和景物深度相似的趋势,即在深度缓变处平滑,在深度不连续处随之跳变.

本文通过以下方式近似估计透射率 $t(x)$.

对式(9)右侧进行均值滤波

$$\begin{aligned} average_{s_a} \left(1 - \frac{M(x)}{A_o} \right) &= 1 - \frac{average_{s_a}(M(x))}{A_o} \\ &= 1 - \frac{\sum_{y \in \Omega(x)} M(y)}{A_o s_a^2} \end{aligned} \quad (10)$$

其中, s_a 表示均值滤波窗口的尺寸, $\Omega(x)$ 为像素 (x) 的 $s_a \times s_a$ 邻域.均值滤波后的结果能反应 $t(x)$ 的大致趋势,但与真实的 $t(x)$ 相差一定的绝对值,因此对该均值滤波的结果弥补一个偏移值,并记

$$M_{ave}(x) = average_{s_a}(M(x)) \quad (11)$$

于是,得出透射率的粗估计值

$$\tilde{t}(x) = 1 - \frac{M_{ave}(x)}{A_o} + \varphi \frac{M_{ave}(x)}{A_o} \quad (12)$$

上式中右侧最末项为弥补的偏移值,且 $0 \leq \varphi \leq 1$.

记 $\delta = 1 - \varphi$,则

$$\tilde{t}(x) = 1 - \delta \frac{M_{ave}(x)}{A_o} \quad (13)$$

对 δ 的取值进行分析如下,当 δ 取值较小时,透射率取值较大,最后恢复出的图像残留雾气较多,图像整体偏白;当 δ 取值较大时,透射率取值较小,最终恢复出的图像整体较为暗淡.为了防止去雾后图像出现整体画面过于偏暗或过于偏亮的情况,这里设置

$$\tilde{\delta} = \rho m_{av} \quad (14)$$

其中, ρ 为可调节的参数且 $0 \leq \rho \leq 1/m_{av}$, m_{av} 是 $M(x)$ 中所有元素的均值. (注意, 需要约束 m_{av} 的值在 $[0 \ 1]$ 之间, 若输入图像的范围为 $[0 \ 1]$, 则 m_{av} 直接取 $M(x)$ 中所有元素的均值; 若输入图像的范围为 $[0 \ 255]$, 则需要在求取 $M(x)$ 中所有元素的均值后相应缩小255倍). 由于 m_{av} 是一个和原图像的整体灰度分布情况有关的值, 使得 $\tilde{\delta}$ 具有自动调节去雾后图像整体亮度的作用. 原图像越暗, m_{av} 值越低, 透射率相应增大, 这使得去雾后的图像亮度不会过于暗淡, 反之亦然.

为了保证 φ 为正值且并不过小, 需要对 δ 的上限进行设置, 这里设定上限为0.9, 则

$$\delta = \min(\rho m_{av}, 0.9) \quad (15)$$

联合式(9), 式(13)和式(15), 可以得出

$$t(x) = \max\left(1 - \min(\rho m_{av}, 0.9) \frac{M_{ave}(x)}{A_o}, 1 - \frac{M(x)}{A_o}\right) \quad (16)$$

记 $L(x) = [L_o(x) \ L_o(x) \ L_o(x)]^T$, 由式(2), 可以得出 $L_o(x) = A_o(1 - t(x))$, 则环境光(Airlight)估计如下

$$L_o(x) = \min(\min(\rho m_{av}, 0.9) M_{ave}(x), M(x)) \quad (17)$$

可以看出, 我们利用均值滤波估计出的环境光具有深度缓变处平滑, 深度跳变处随之跳变的性质. 这是因为, 当图像 $H(x)$ 深度发生大的跳变时, $M(x)$ 和 $M_{ave}(x)$ 也随之有大的跳变, 最终体现在 $L(x)$ 上. (如图1所示.)

2.3 估计全局大气光

文[5]中选取输入图像中亮度最大的点的灰度值作为全局大气光 A , 虽然步骤简单但不太合理. 文[7]先从暗原色通道中选取最亮的0.1%比例的像素, 然后选取原输入图像中这些像素具有的最大灰度值作为全局大气光 A , 这种处理方式更为合理有效, 但暗原色通道的获取过程较为耗费时间.

本文将采用下述方式来估计全局大气光 A . 容易看到, 式(10)中第一个等号左侧的表达式取值范围应为 $[0 \ 1]$, 由此, 可以得出 $A_o \geq \max(M_{ave}(x))$, 一般情况下, 又认为: $A_o \leq \max(\max_{c \in \{r, g, b\}}(H^c(x)))$, 于是 $\max(M_{ave}(x)) \leq A_o \leq \max(\max_{c \in \{r, g, b\}}(H^c(x)))$. 因此, A_o 可以表示为

$$A_o = \varepsilon \max\left(\max_{c \in \{r, g, b\}}(H^c(x))\right) + (1 - \varepsilon) \max(M_{ave}(x)) \quad (18)$$

其中, $0 \leq \varepsilon \leq 1$. 由于 ε 的准确取值很难获取, 综合考虑算法的快速性, 这里, 我们取 $\varepsilon = 0.5$, 则有

$$A = \frac{1}{2} \left(\max\left(\max_{c \in \{r, g, b\}}(H^c(x))\right) + \max(M_{ave}(x)) \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (19)$$

这种估计方式既避免了直接选取输入图像中最亮点的灰度值作为大气光值的不合理, 又兼顾了算法的快速性.

3 算法流程

估计出环境光 $L(x)$ 和全局大气光 A 后, 利用物理模型恢复出图像 $F(x)$

$$F(x) = \frac{H(x) - L(x)}{1 - \frac{L(x)}{A}} \quad (20)$$

本文中算法的实施步骤如表1所示.

表 1. 算法步骤

步骤:
1: 输入: 有雾图像 $H(x)$.
2: 求取 $M(x)$.
$M(x) = \min_{c \in \{r, g, b\}}(H^c(x))$
3: 对 $M(x)$ 进行均值滤波, 得出 $M_{ave}(x)$.
$M_{ave}(x) = \text{average}_{s_s}(M(x))$
4: 求取 $M(x)$ 中所有元素的均值 m_{av} .
5: 利用 $M_{ave}(x)$ 求出 $L(x)$.
$L_o(x) = \min(\min(\rho m_{av}, 0.9) M_{ave}(x), M(x))$
6: 利用 $M_{ave}(x)$ 和 $H(x)$ 求出 A .
$A = \frac{1}{2} \left(\max\left(\max_{c \in \{r, g, b\}}(H^c(x))\right) + \max(M_{ave}(x)) \right) [1 \ 1 \ 1]^T$
7: 输出: 去雾后图像 $F(x)$.
$F(x) = \frac{H(x) - L(x)}{1 - \frac{L(x)}{A}}$

当输入图像 $H(x)$ 是灰度图像时, 用 $H(x)$ 代替 $M(x)$ 进行 $L_o(x)$ 的估计, 并用下式求出 A_o

$$A_o = \frac{1}{2} \left(\max(H(x)) + \max(M_{ave}(x)) \right) \quad (21)$$

进而求出 $F(x)$

$$F(x) = \frac{H(x) - L_o(x)}{1 - \frac{L_o(x)}{A_o}} \quad (22)$$

可见, 本文通过一次均值滤波来估计环境光和局部大气光, 进而恢复图像, 与已有算法相比, 极大的简化了算法步骤.

图1为本文算法的彩色图像处理示例($\rho=1.3$), 从左到右依次为输入图像 $H(x)$, 均值滤波结果 $M_{ave}(x)$, 环境光 $L(x)$, 输出图像 $F(x)$. 图2显示了对于同一幅图像, ρ 取值的改变对实验结果的影响,

从左到右 ρ 的取值依次为0.8, 1.2, 1.6和2.0. 当 $0 \leq \rho \leq 0.9/m_{av}$ 时, 随着 ρ 值的增大, 环境光取值随着变大, 去雾后图像中残留的雾量相应减少, 图像整体越暗淡; 当 $0.9/m_{av} < \rho$ 时, ρ 的调节作用失效. 注意, 对于同一幅输入图像, m_{av} 保持不变, 只有 ρ 发挥调节作用; m_{av} 的调节作用体现在不同图像上.



图1: 彩色图像处理示例



图 2: ρ 的取值对去雾效果的影响





图 3: 彩色图像去雾效果比较

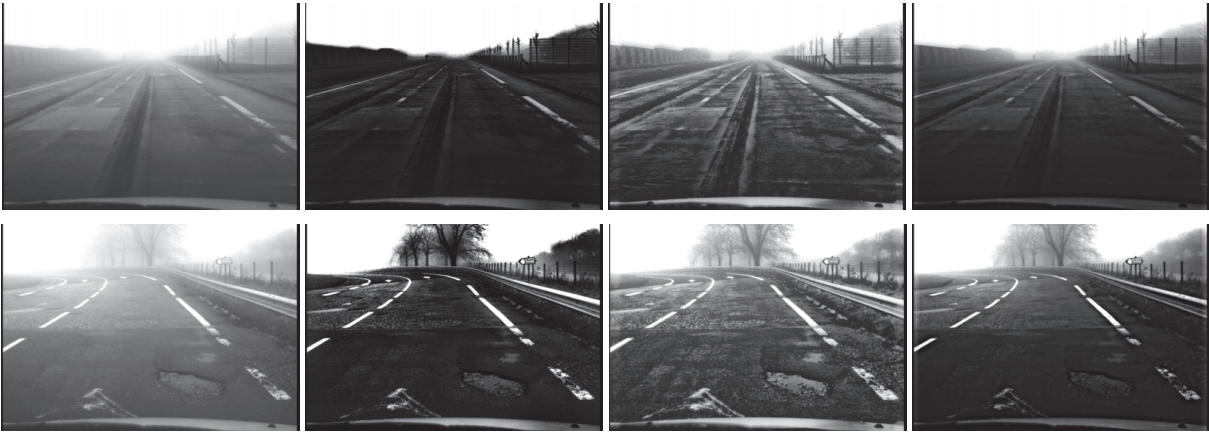


图 4: 灰度图像去雾效果比较

4 实验比较与分析

4.1 去雾效果比较与分析

图3和图4为本算法($\rho=1.3$)与其他文献中算法的去雾效果比较. 在图3和图4中, 从左到右依次为原图, 文[7]的处理结果, 文[8]的处理结果和本文算法的处理结果. 其中, 文[8]的处理结果通过运行网址[12]上提供的源码得到, 原图和文[7]的处理结果下载于网址[12]或[13].

在图3中, 本文的处理结果能较好地恢复出景物的色彩和结构, 亮度适中, 色彩深浅适宜且无光晕. 图4中, 本文的处理结果能很好地恢复出景物结构, 且画面更为平滑. 由此可见, 本文的去雾效果与文[7], 文[8]的去雾效果相差不大.

4.2 运算速度比较

本文选取文[8]中的快速去雾算法进行快速性比较. 本文算法与文[8]算法得出图3和图4中所对应的无雾图像时所需时间如表2和表3所示. 表中数据通过Matlab7.11仿真得到, 运行环境为: 2.33Ghz的Intel Core2 (Q8200) CPU, 2GB物理内存, 64位win7系统. 滤波半径均取图像尺寸(长, 宽)的最大值的1/50(向下取整). 由表2和表3可见, 本文算法的运行效率远远大于

文[8]. 这是由于本文算法仅进行一次滤波, 且使用的均值滤波器执行较为快捷, 耗时较少.

表 2. 彩色图像去雾时间比较(单位/s)

	本算法	文[8]
图3上	0.063	27.83
图3中	0.032	5.25
图3下	0.075	31.11

表 3. 灰度图像去雾时间比较(单位/s)

	本算法	文[8]
图4上	0.033	39.78
图4下	0.032	43.50

5 总结

本文通过对物理模型的分析, 提出一种快速单幅图像去雾算法, 该算法仅利用一次简单的均值滤波对环境光和全局大气光进行估计, 步骤简单, 运行速度快, 且去雾效果良好, 并具有自动调节去雾后图像亮度的能力.

参考文献

- [1] S. G. Narasimhan, S. K. Nayar, Contrast Restoration of Weather Degraded Images, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, 713-724, 2003.
- [2] S. Shwartz, E. Namer, Y. Schechner, Blind Haze Separation, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1984–1991, 2006.
- [3] S. G. Narasimhan, S. K. Nayar, Interactive Deweathering of an Image Using Physical Models, in *Proc. Workshop on Color and Photometric Methods in Computer Vision*, 2003.
- [4] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, D. Lischinski, Deep photo: Model-based photograph enhancement and viewing, *ACM Trans. Graphics*, Vol. 27, No. 5, 116:1-116:10, 2008.
- [5] R. T. Tan, Visibility in Bad Weather from a Single Image, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1-8, 2008.
- [6] R. Fattal, Single Image Dehazing, in *Proc. ACM SIGGRAPH*, 1-9, 2008.
- [7] Kaiming He, Jian Sun, Xiaoou Tang, Single Image Haze Removal Using Dark Channel Prior, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1956-1963, 2009.
- [8] J.-P. Tarel, N. Hautière, Fast Visibility Restoration from a Single Color or Gray Level Image, in *Proc. IEEE Conf. Computer Vision*, 2201–2208, 2009.
- [9] Fang Li, Haoxian Wang, Xingpeng Mao, Yunlei Sun, Huiyao Song, Fast Single Image Defogging Algorithm, *Computer Engineering and Design*, Vol. 32, No. 12, 4129-4132, 2011.
- [10] Haoran Xu, Jianming Guo, Qing Liu, Lingli Ye, Fast Image Dehazing Using Improved Dark Channel Prior, in *Proc. IEEE Conf. Information Science and Technology*, 663-667, 2012.
- [11] Jing Yu, Chuangbai Xiao, Dapeng Li, Physics-based Fast Single Image Fog Removal, in *Proc. IEEE Conf. Signal Processing*, 1048-1052, 2010.
- [12] <http://perso.lcpc.fr/tarel.jean-philippe/visibility/>
- [13] <http://research.microsoft.com/en-us/um/people/kahe/>