



去雾概述

报告人： 谷鸢翔

中国科学院自动化研究所

November 17, 2011





说明

- 讲述内容只是个人的、局部的理解，仅供参考，恳请各位老师和师兄师姐指正。
 - ① 大四毕设做的一点工作；
 - ② 时间冲突，**slides** 准备不足；
 - ③ 研二，经验不足。
- 主要目标：1、去雾是个什么问题？ 2、有哪些算法？好不好？快不快？
 - ① 侧重去雾模型；
 - ② 侧重于实时性；
 - ③ 面向具体问题：无人驾驶车、路牌识别。



内容

- ① 问题引出
- ② 物理建模
- ③ 研究意义与现状
 - 研究意义
 - 方法分类
 - 国内现状
- ④ 典型去雾算法
 - Dark Channel Prior (DCP) 去雾算法
 - DCP 改进算法
 - 基于半求反的检测去雾算法 (Semi-Inverse)
- ⑤ 总结与建议





问题引出



(a) 输入有雾图像



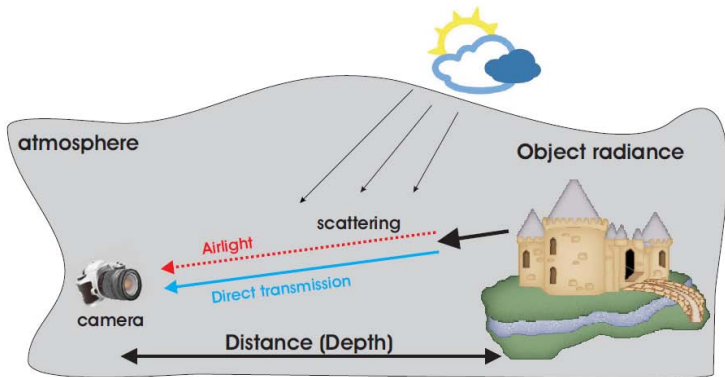
(b) 输出图像

- 雨、雾、雪等自然现象很容易导致户外图像降质；
- 普通的图像增强算法不能有效的处理这些降质图像。
- 通过对大气传输模型的研究，**Narasimhan** 等提出了目前通用的去雾物理模型，去雾效果大大提升。





物理模型



(c) 光照模型





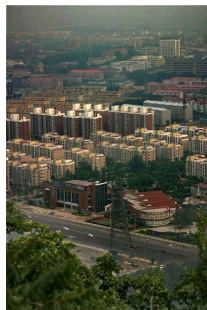
数学表达

● 常用模型

$$I = Jt + A(1 - t) \quad (2)$$

其中：

$$t = e^{-\beta d(x)} \quad (3)$$





补充说明

- 常用模型

$$I = Jt + A(1 - t) \quad (4)$$

其中：

$$t = e^{-\beta d(x)} \quad (5)$$

- 模型的物理意义：左边第一项代表物体本身的辐射经大气衰减后进入相机的光强，第二项代表太阳光经大气反射或折射后进入相机的光强。
- 去雾是一个欠约束问题：通常要添加不同的先验信息或者约束条件才能求解。
- 求解上述模型的关键在于： t ，或者说 $d(x)$ 。





研究意义

● 图像处理角度

- ① 预处理：可以显著提高图像的对比度、还原真实信息，直接影响接下来的图像分割、目标识别等；
- ② 附加值：可以从雾的浓度估计出景物的深度信息；

● 实际应用角度

- ① 军用系统：航空航天、目标精确定位与打击、对地观测等；
- ② 民用系统：**智能交通系统**、户外监视系统、地形侦测系统等。



方法分类

- 多幅图像去雾

- ① 同一场景点不同偏振光条件下多幅图像——Yoav Y. Schechner CVPR 2001
- ② 同一场景不同季节的图像——Srinivasa G. Narasimhan PAMI 2003 ;
- ③ 已知深度 (卫星提供或者已知三维模型)——Kpof Siggraph Asia 2008
- ④ 红外和可见光两幅图像融合——Lex Schaul ICIP 2009

- 单幅图像去雾

- ① 极大化局部对比度 (梯度和)——Tan CVPR 2008;
- ② ICA——Fattal Siggraph 2008
- ③ **Dark Channel Prior** () ——He CVPR 2009 ;
- ④ **MDCP**——Kristofor OCEANS 2010
- ⑤ **Semi Inverse**——Kristofor Gibson Siggraph Asia 2010





多幅图像去雾

- 利用相机附带的偏振光滤波器，以及同一场景点在不同偏振光条件下获得的图像，实现多幅图像去雾。缺点：是对动态场景处理效果不好。
- 利用同一场景在不同雾强度下进行拍摄所得的多幅图像来去雾。缺点：需要等到天气条件的改变，才能拍摄多幅图像，有很强的时间限制；不能很好的处理运动场景。
- 利用单幅图像和附加的场景深度或者场景结构信息来实现去雾、在同一场景不同角度拍摄多幅图像去雾、用户有限的交互操作下去雾等。





单幅图像去雾

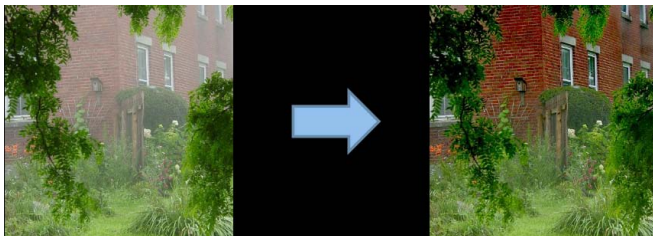


- Tan 基于无雾图像的对比度比有雾图像的对比度要高的事实，通过最大化局部对比度，然后基于 MRF 框架对结果进行规整化，也实现了单幅图像的去雾。优缺点：该方法能极大的增强图像的对比度，但是容易导致图像的颜色失真，并且在场景深度不连续的地方会产生光圈效应。





单幅图像去雾



- **Fattal** 运用独立成份分析的方法，基于图像表面阴影和大气传递函数在图像局部块上统计不相关的假设，利用雾天图像退化模型，在 **MRF** 框架的约束下，实现了单幅图像的去雾。优缺点：该方法去雾效果显著。不过对于图像各个分量变化不大的情况，处理效果不好。



国内现状

● 侧重研究

- ① 孙剑、汤晓鸥老师——微软亚洲研究院 -何凯明博士的 Dark Channel Prior 去雾；
- ② 中科院自动化所、天津大学等；

● 侧重项目

- ① ClaireVue 团队——清华大学——指导老师：温江涛，iPhone4 上 192×144 实时去雾；
- ② 北航薛斌党老师——军方项目。



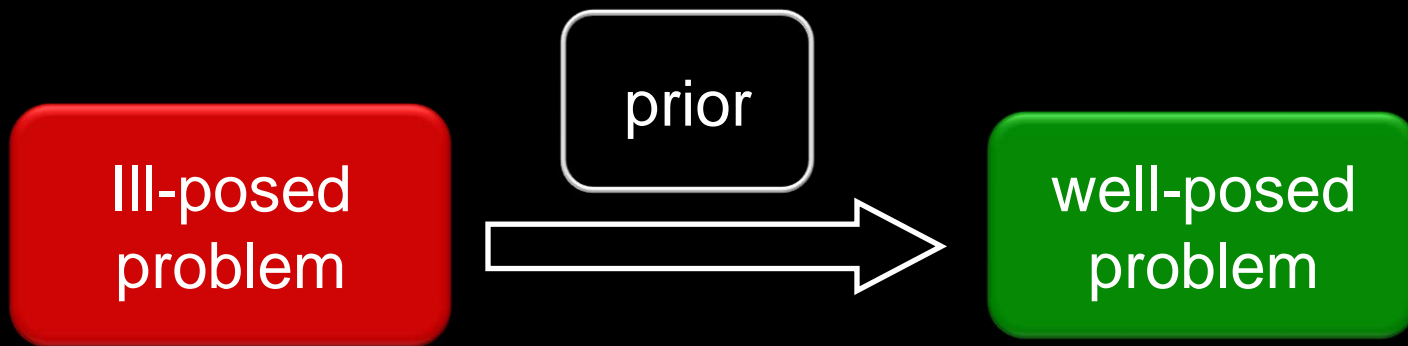


去雾经典算法——Dark Channel Prior (DCP) 算法

- 何凯明博士在 2009 年 CVPR 上面提出一中 DCP 先验，简单却强大；
- 通过 DCP 先验，能够快速估计出大气传输函数 t ，进而求解清晰图像 J 。
- 去雾效果很理想，但是平滑大气传输函数 t 时复杂度过高，导致去雾速度较慢。
- 很多算法都是基于 DCP 的改进算法。



Priors in Computer Vision



- Smoothness prior
- Sparseness prior
- Exemplar-based prior

Dark Channel Prior

Dark Channel

- $\min(\text{rgb}, \text{local patch})$

Dark Channel

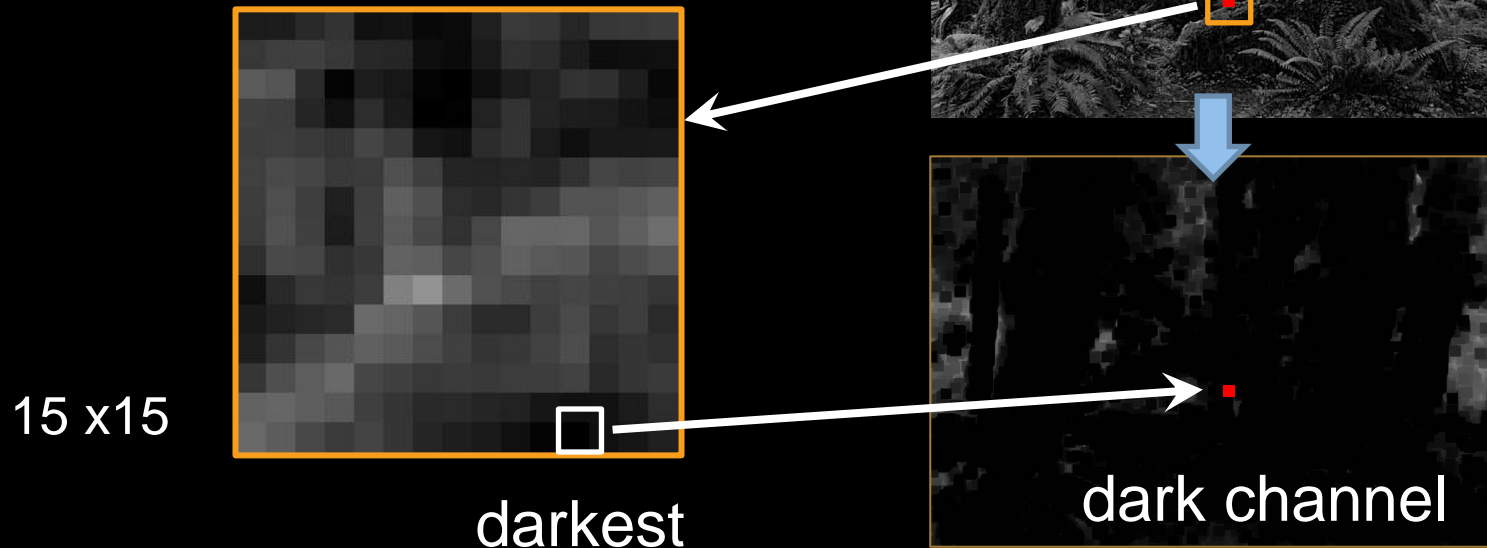
- $\min(\text{rgb}, \text{local patch})$
 - $\min(r, g, b)$



$\min(r, g, b)$

Dark Channel

- $\min(\text{rgb}, \text{local patch})$
 - $\min(r, g, b)$
 - $\min(\text{local patch}) = \text{min filter}$

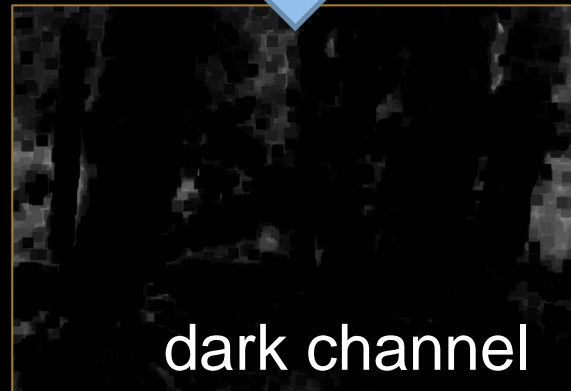
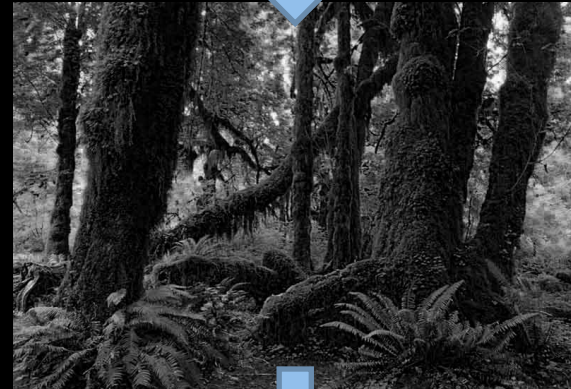


Dark Channel

- min (rgb, local patch)
 - min (r, g, b)
 - min (local patch) = min filter

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

- J^c : color channel of \mathbf{J}
- J^{dark} : dark channel of \mathbf{J}

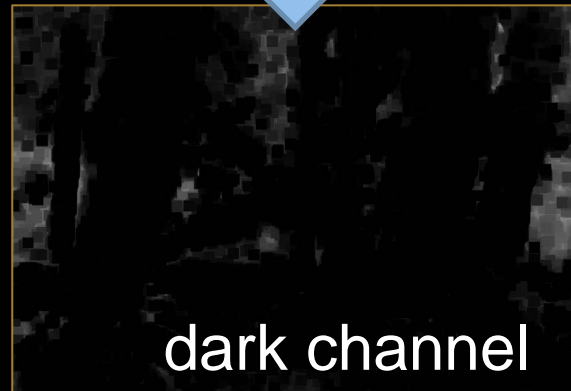


Dark Channel

- min (rgb, local patch)
 - min (r, g, b)
 - min (local patch) = min filter

$$J^{dark} = \min_{\Omega} (\min_c J^c)$$

- J^c : color channel of J
- J^{dark} : dark channel of J



A Surprising Observation

Haze-free



A Surprising Observation

Haze-free



A Surprising Observation

Haze-free



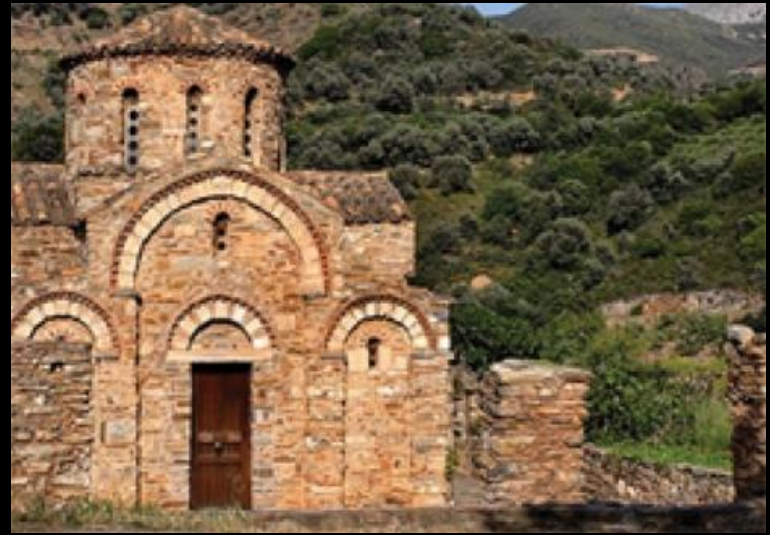
A Surprising Observation

Haze-free



A Surprising Observation

Haze-free

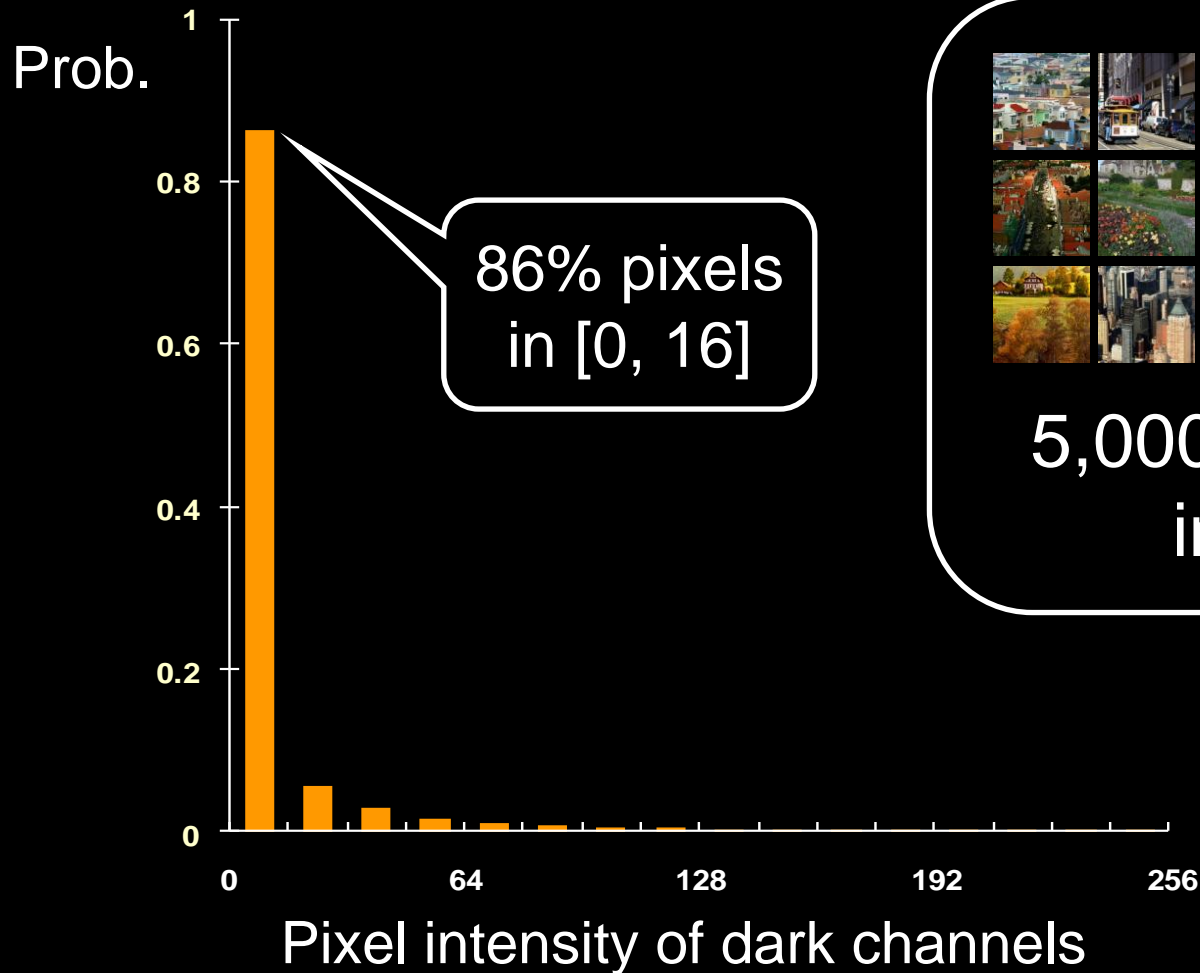


A Surprising Observation

Haze-free



A Surprising Observation



5,000 haze-free
images

Dark Channel Prior

- For outdoor haze-free images

$$\min_{\Omega} (\min_c J^c) \rightarrow 0$$

What makes it dark?

- Shadow



- Colorful object



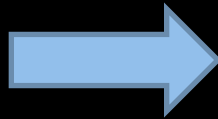
- Black object



Dark Channel of Hazy Image



hazy image



dark channel

- The dark channel is no longer dark.

Transmission Estimation

Haze imaging model $\mathbf{I} = \mathbf{J} \cdot t + \mathbf{A} \cdot (1 - t)$

Normalize
$$\frac{I^c}{A^c} = \frac{J^c}{A^c} t + 1 - t$$

Compute dark channel

$$\min_{\Omega} \left(\min_c \frac{I^c}{A^c} \right) = \left\{ \min_{\Omega} \left(\min_c \frac{J^c}{A^c} \right) \right\} t + 1 - t$$

Transmission Estimation

Dark Channel Prior

$$\min_{\Omega} (\min_c J^c) \rightarrow 0$$



Compute dark channel

$$\min_{\Omega} (\min_c \frac{I^c}{A^c}) = \left\{ \min_{\Omega} (\min_c \frac{J^c}{A^c}) \right\}^t + 1 - t \rightarrow 0$$

Transmission Estimation

Estimate transmission

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



Compute dark channel

$$\min_{\Omega} \left(\min_c \frac{I^c}{A^c} \right) = \left\{ \min_{\Omega} \left(\min_c \frac{J^c}{A^c} \right) \right\} t + 1 - t$$

Transmission Estimation

Estimate transmission

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



input I



estimated t



Transmission Optimization

Haze imaging model $\mathbf{I} = \mathbf{J} \cdot t + \mathbf{A} \cdot (1 - t)$

Matting model $\mathbf{I} = \mathbf{F} \cdot \alpha + \mathbf{B} \cdot (1 - \alpha)$



+

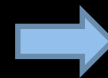


tri-map

α



+



Refined
transmission

Transmission Optimization

$$E(\mathbf{t}) = \lambda \underbrace{\|\mathbf{t} - \tilde{\mathbf{t}}\|^2}_{\text{Data term}} + \underbrace{\mathbf{t}^T \mathbf{L} \mathbf{t}}_{\text{Smoothness term}}$$

Data term

Smoothness term

- **L** - matting Laplacian [Levin et al., CVPR '06]
- Constraint - soft, dense (matting - hard, sparse)

Transmission Optimization



before optimization

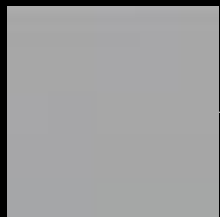
Transmission Optimization



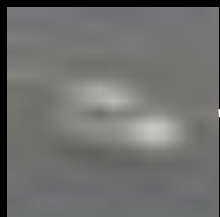
after optimization

Atmospheric Light Estimation

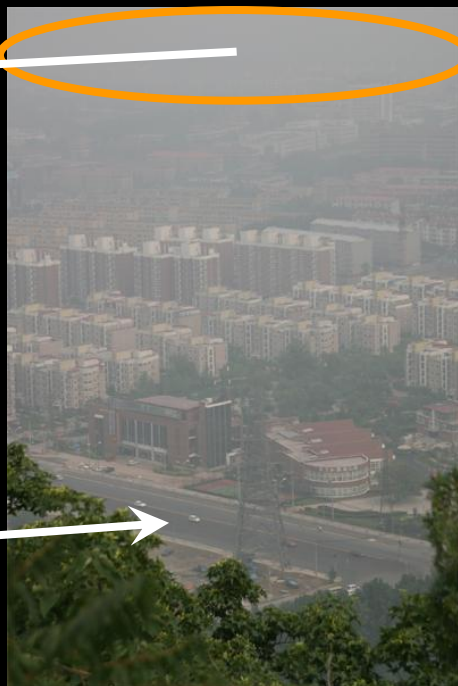
A : most hazy



brightest pixels



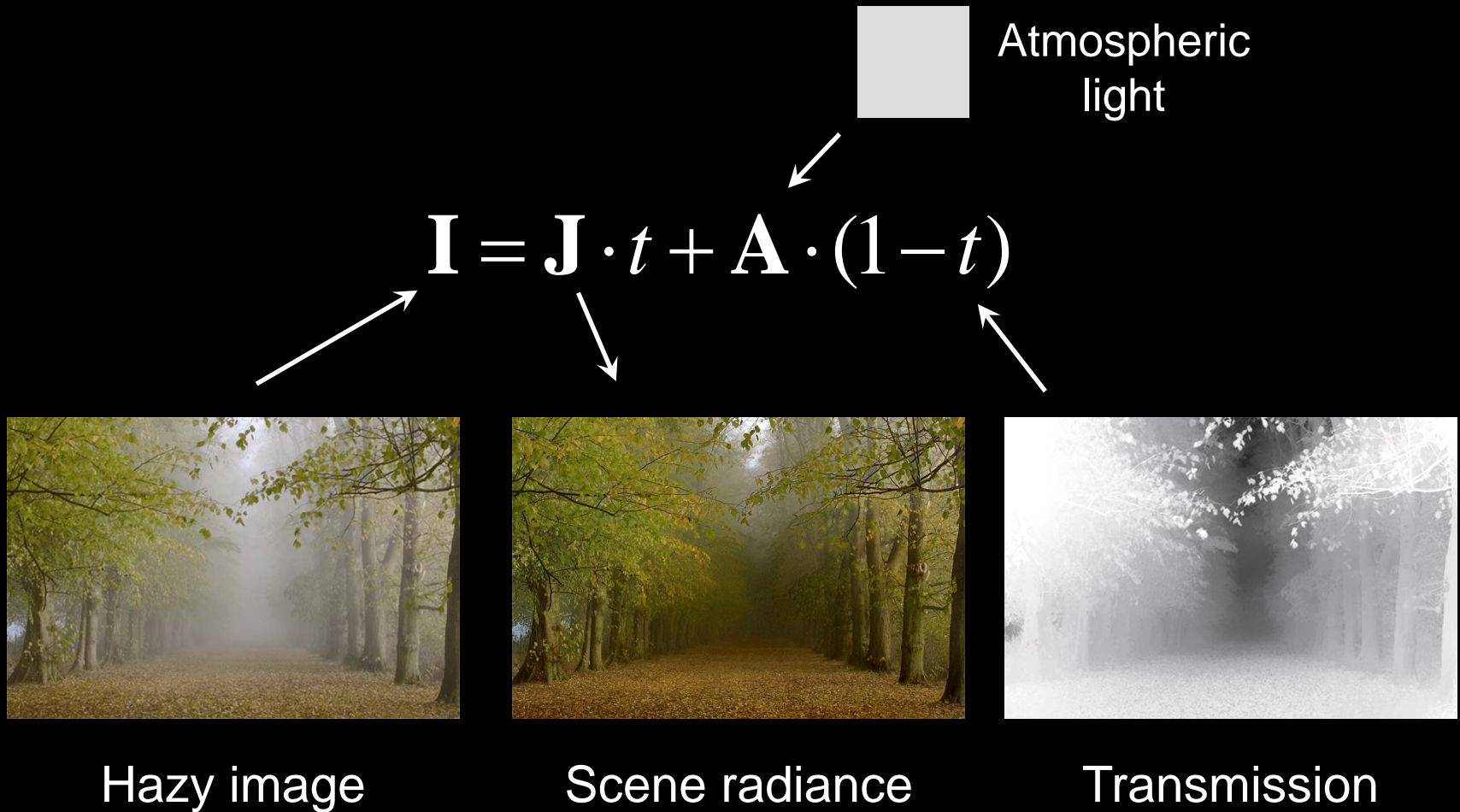
brightest pixel



hazy image

dark channel

Scene Radiance Restoration

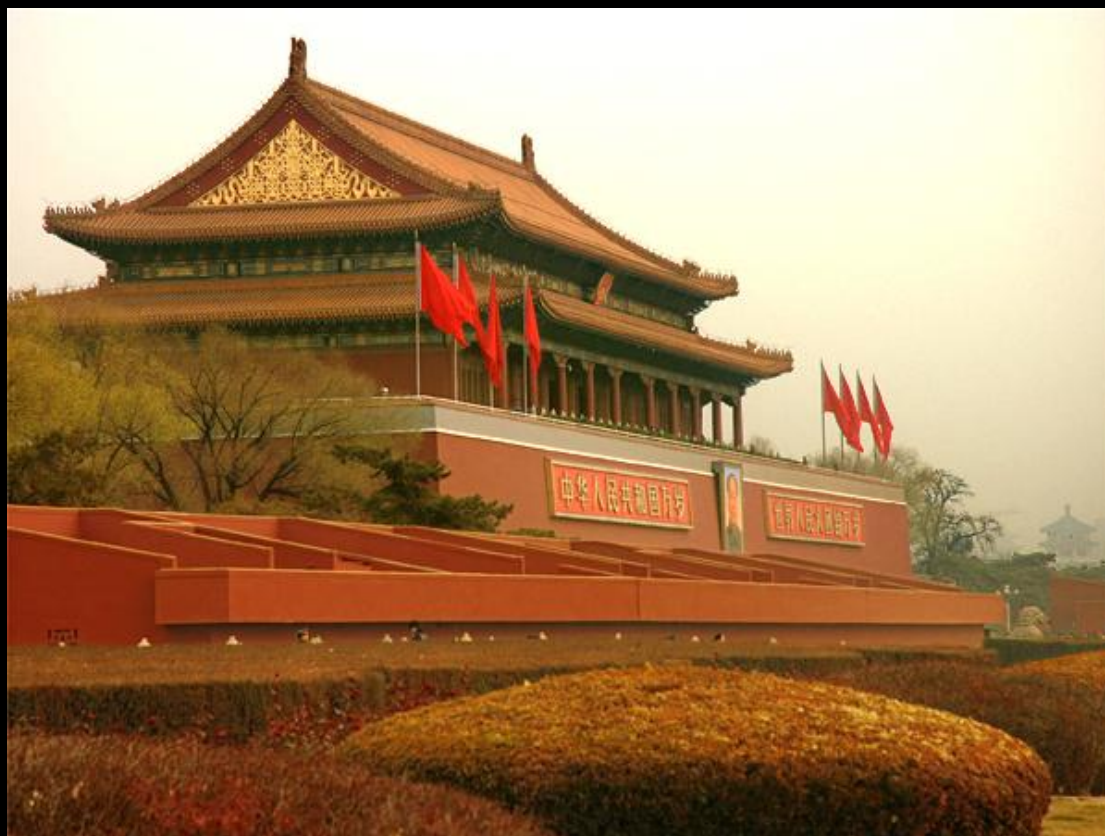


Results



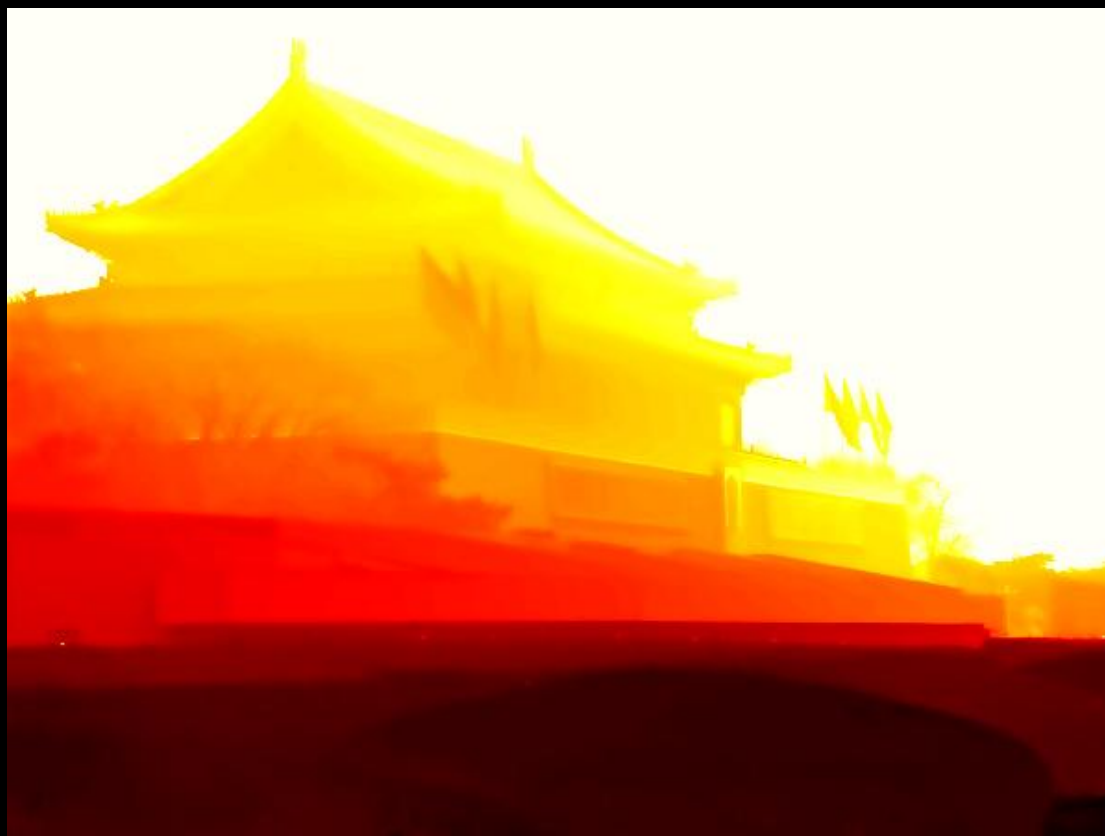
input

Results



recovered image

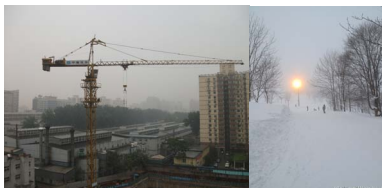
Results



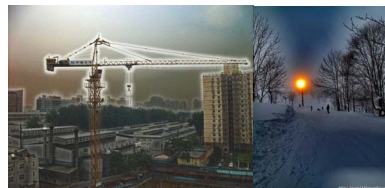
depth



DCP 算法失效场景



(g) 输入图像 (I)



(h) 输出图像





DCP 算法分析总结

- 先验信息非常简单有效，去雾效果较为理想；
- 平滑大气传输函数采用 **matting** 的思路，效果较好但是复杂度过大，为 $O(NK^3)$ ，其中 N 为图像大小， K 为所取窗口大小；
- 处理一幅 600×400 的图像需要 2s (C++)，远远达不到实时的标准。





DCP 改进算法

- 基于快速滤波的去雾算法——Tarel, ICCV 2009
 - ① 效果一般，全局颜色和局部窗口的细节有一定失真，
 - ② 480×600 用时 0.17s，利用帧间信息和多核加速算法可以达到实时。
- 基于中值滤波的改进去雾算法（MDCP）——Kristofor OCEANS 2010
 - ① 算法非常简单，效果一般；
 - ② 算法复杂度为 $O(\log(N))$ ，可以分块并行处理，能达到实时的要求；
- 基于半求反的检测去雾算法——Codruta, SIGGRAPH ASIA 2010
 - ① 算法非常简单，能够有效检测并且去雾；
 - ② 600×800 图像 C++ : 0.013s, 已达到实时要求。





基于中值滤波的改进去雾算法 (MDCP)

- MDCP 核心思想：逐像素求取局部窗口内的中值，来替代 DCP 算法中的窗口内最小值。
- 两者主要区别：
 - ① DCP 算法：

$$\theta_D(m, n) = \min_{k, l \in \Omega(m, n)} \left(\min_{c \in \{r, g, b\}} \frac{x(k, l, c)}{A(c)} \right), \quad (6)$$

- ② MDCP 算法：

$$\theta_M(m, n) = \text{med}_{k, l \in \Omega(m, n)} \left(\min_{c \in \{r, g, b\}} \frac{x(k, l, c)}{A(c)} \right), \quad (7)$$

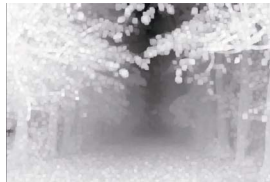




MDCP VS DCP



(i) 输入图像 (I)



(j) 输出图像 (J)



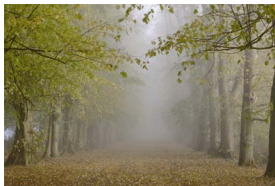
(k) 估计景深 (t)



(l) 估计景深 (t)



MDCP VS DCP



(m) 输入图像 (I)



(n) 输出图像 (J)



(o) 估计景深 (t)





MDCP 总结



Fig. 1. (a) The original hazy image from [11] with size 600×400 . (b) The dehazed image using He's DCP method [11]. 589.1 seconds to process. (c) Dehazed using our proposed MDCP method. 1.2 seconds to process. (d) Dehazed using Tarel's method [12]. 2.9 seconds to process. Note: All simulations are done using Matlab code. Processing times can be reduced considerably if implemented in C or C++.

(p) 算法对比

- 与 DCP 算法相比：MDCP 算法去雾效果稍差，但是速度快，可以采用快速中值滤波算法，而且可以分块并行运算，适合视频去雾。





Semi-Inverse 算法的先验信息

- 清晰户外图像的局部窗口中至少有一个 R,G,B 通道的值非常小 (dark channel prior)；
- 天空或者有雾图像的局部窗口中每一个 R, G,B 通道的值都比较大；
- 有雾图像的局部窗口中的值变化很平缓，雾越浓越平缓。





Semi-Inverse 检测算法

- 基于上述先验信息，定义一组求反算子：

$$\begin{aligned}I_{si}^r(x) &= \max_{x \in I} \{I^r(x), 1 - I^r(x)\} \\I_{si}^g(x) &= \max_{x \in I} \{I^g(x), 1 - I^g(x)\} \\I_{si}^b(x) &= \max_{x \in I} \{I^b(x), 1 - I^b(x)\}\end{aligned}\tag{8}$$

- 得到的 **Semi-Image** 图像 I_{si} 的动态范围为 $[0.5, 1]$ ，所以需要重新规整到 $[0, 1]$ ；
- 规整后图像的有雾区域的动态范围得到拉伸。
- 将 I, I_{si} 转换到 **CIE L^*c^*h** 颜色空间，规定 $h(x) - h_{si}(x) < 10^\circ$ 的像素为有雾像素。





Semi-Inverse 雾的检测实验

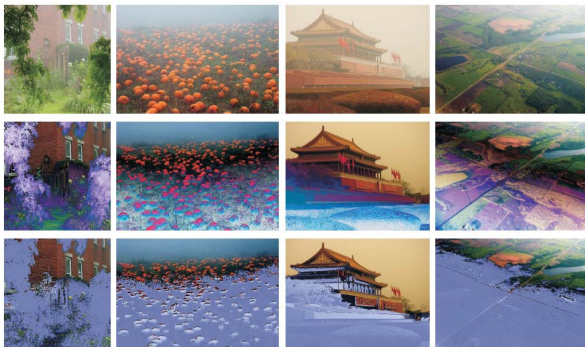


Fig. 3. Haze detection. The first row shows the original hazy images \mathcal{I} . In the second row, we show the yielded semi-inversed image \mathcal{I}_{si} . Finally, in the third row, we label the pixels identified as not under the influence of haze with a blue mask. In these regions, the intensity of the blue color is proportional with the hue disparity.





Semi-Inverse 雾检测算法的有效性验证



Fig. 4. Results from applying our haze detection procedure on a large data set of images. Overall, haze-free images contain 96% pixels labeled as haze-free (masked in blue), while hazy and sky images are characterised by a significant decrease in haze-free pixels (less than 13%).





Semi-Inverse 去雾算法

- 先求 **A**：由于算法能够较为准确的定义有雾区域，所以 **A** 可以从雾的最浓区域中选取最亮的像素值作为常值 **A**。
- 计算清晰图像 **J**：采用分层的思路。
 - ① 从输入有雾图像中减去不同分量的常值 **A**

$$I_i = I - c_i \cdot A \quad (9)$$

其中： $c_i = [0.2; 0.4; 0.6; 0.8; 1]$ 。

- ② 对不同的 I_i 应用去雾检测算法， $h(x) - h_{si}(x) < 10^\circ$ 的像素标记为 L_i (not very clearly)；
- ③ 清晰图像 **J** 为各层 L_i 的求和：

$$J = \sum_{i=1}^k \lambda_i L_i.$$





Semi-Inverse 去雾算法图像演示

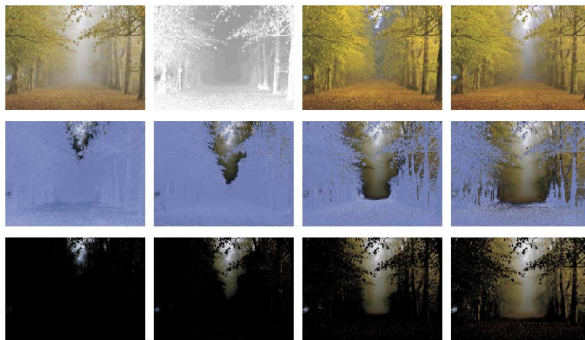


Fig. 5. *Layer-based dehazing.* **Top line:** the initial foggy image; the rough transmission map that corresponds to \mathcal{I}_0 ; the result of a naive method which simply pastes all layers \mathcal{L}_i upon each other, introducing artifacts; the result of our method, which applies soft blending of the layers. **Middle line:** the mask regions for each layer. **Bottom line:** the computed layers \mathcal{L}_i .





Semi-Inverse 检测去雾算法总结

- Semi-Inverse 算法能够快速有效的检测有雾区域；
- Semi-Inverse 算法去雾算法简单，效果较好，速度快。



(t) 算法对比



总结

- 介绍了经典的去雾模型和求解思路；
- 重点讲述了何博士的 **DCP** 算法去雾算法（效果较为理想，速度慢）；
- 简单介绍了两种基于 **DCP** 改进的 **MDCP** 和 **Semi Inverse** 算法（满足实时要求）。





建议 (1)——从问题角度

- 以上算法都是从研究的角度出发的，怎样得到视觉效果更好的全局清晰图像；
- 从无人驾驶车识别路牌的角度出发，也许只需要得到便于识别的清晰路牌即可；



(u) 算法对比

- 全局去雾也许不是最好的解决思路，面向路牌的局部区域去雾算法效果也许更好、更快。





建议 (2)——从硬件角度

$$I = Jt + A(1 - t) \quad (11)$$

- 假如能有深度测量设备，比如小型雷达，则

$$t = e^{-\beta d(x)} \quad (12)$$

其中： $\beta = \frac{1}{\lambda r}$ 是与波长和粒子直径相关的常数，

- 假如有红外摄像头（红外图像基本上无雾）——可见光与红外融合去雾——Lex Schaul ICIP 2009。





建议 (3)——从软件角度

- 推荐使用：半求反的检测去雾算法，去雾效果较好，速度快，
- 半求反的检测去雾算法中的检测算法是亮点，去雾算法也许可以进一步改进，
- CAVE 实验室的视频去雨算法演示效果较为理想。





谢谢！：)

