

# 图像增强去雾系统设计与实现

朱 烨, 王 婷, 魏 敏, 苟艺频

成都信息工程大学, 计算机学院, 四川 成都

收稿日期: 2022年3月13日; 录用日期: 2022年4月14日; 发布日期: 2022年4月21日

---

## 摘 要

雾霾天气在中国冬季的大部分城市都长期存在, 导致能见度降低, 给人们的日常生活和出行带来极大影响。本文设计了一种基于python使用Django框架所开发的web图像去雾系统, 通过对暗通道、导向滤波去雾和自动色阶去雾算法的实现, 用户可以选择不同的去雾功能来实现去雾, 并且对同一算法能在界面上调制参数来实现去雾功能。在去雾时可在系统中通过设置参数的方式, 自由选择合适的去雾程度和去雾方案, 让用户摆脱在雾天所拍的图片不清晰的困扰。

## 关键词

去雾, 暗通道, 导向滤波, 自动色阶, Django框架

---

# System Design and Realization of Web Image Defogging

Ye Zhu, Ting Wang, Min Wei, Yipin Gou

School of Computer Science, Chengdu University of Information Technology, Chengdu Sichuan

Received: Mar. 13<sup>th</sup>, 2022; accepted: Apr. 14<sup>th</sup>, 2022; published: Apr. 21<sup>st</sup>, 2022

---

## Abstract

Smog has long existed in most cities of China in winter, resulting in reduced visibility and greatly affecting People's Daily life and travel. This paper designs a web image defogging system based on Python and Django framework. Through the realization of dark channel, guided filtering and automatic color level defogging algorithm, users can choose different defogging functions to realize the defogging function, and the same algorithm can modulate parameters on the interface to realize the defogging function. When defogging, the appropriate defogging degree and defogging scheme can be freely selected by setting parameters in the system, so that users can get rid of the problem of unclear pictures taken in foggy days.

## Keywords

Haze Removal, Dark Channel, Guided Filter, Automatic Color Level, Django Framework

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

雾霾天气所拍摄的图片大部分都不是人们所想看到的，尤其是对一些户外监控领域，图像拍摄的质量严重影响正常工作的运行。为了更好地恢复雾霾图像的场景信息，需要对雾霾图像进行复原处理。因此，图像去雾技术的不断探索与研究是计算机视觉领域的持续研究热点。

图像去雾是以满足特定条件下应用需求为目的，通过对有雾图像进行分析和预处理，突出图像中的细节信息使之更加适合人机识别的一种图像预处理方法。目前图像去雾的方法主要为以下两类：第一类为雾化图像增强算法，第二类是雾化图像复原。对于第一类又可在细分为全局化增强算法以及局部化增强，对于第二类可分为基于偏微分方程、基于先验理论和基于深度关系。

1) 基于图像增强技术主要是针对图像自身特征，如对比度、色彩进行图像增强。该方法是直接针对图像来增强其像质，用于提高用户的视觉观察性。其中最常见的方法是平布直方图均衡化的方法。其算法比较简单，运行的速率高，时性较高，但仅针对黑白图像进行处理。局部直方图均衡化技术的提出，可以避免有全图对比度越大使用直方图均衡化会变得更加模糊的问题，但该方法运算量较大，耗时长，效率低下，效果不太理想。

2) 基于物理模型的复原技术，是基于大气散射角度建立图像的退化模型，Oakley [1]等人考虑大气的散射因素以及实际环境的图像复原，在基于先验知识的基础上实习场景复原。南京理工大学计算机学院模式识别与智能系统实验室[2]通过对大气散射强度、图像场景深度以及图像对比度损失程度之间关系的研究，提出了基于偏微分方程的去雾算法。何恺明等人提出一种基于暗通道先验理论的图像去雾算法，不仅可以还原图像的颜色和能见度，还能利用雾的浓度来估计物体的距离[3]。随后又提出了其优化版本—导向滤波去雾[4]，其比之前的暗通道先验去雾算法清楚许多，并且在运行速度上有很大的提升，对于图像中景深处的景物有明显的改进。

本系统通过对暗通道算法、导向滤波去雾，自动色阶去雾算法以及直方图均衡化去雾算法进行实现，并且对不同的去雾算法的效果进行比较来确定最佳去雾算法。本系统采用 python 语言进行开发，并且使用 Django 框架来设计。在进行去雾效果检测和对比采用的是控制变量法，将其中的几个参数设置为常数，再对其进行一个参数的对比。再对不同天气下的图像的去雾效果，采用的是多种算法来进行去雾的补充。

## 2. 相关知识

### 2.1. 暗通道先验算法加上导向滤波的优化算法

暗通道先验算法是何恺明提出的去雾方法[3]。暗通道[5] [6] [7] [8]的概念是一个基本假设：即在大多数非天空的局部区域中，某一像素总是会至少存在一个颜色通道很低的值。则用数学表达式表示暗通道：

$$J^{dark}(x) = \min_{y \in \Omega(x)} \left( \min_{c \in \{r, g, b\}} J^c(y) \right) \quad (1)$$

在(1)式中  $J^c$  表示彩色图像的每一个通道,  $\Omega(x)$  表示的是以像素  $x$  为中心的一个局部块。根据式(1)首先求出每个像素中 RGB 分量中的最小值, 再将其存入到一张与原图大小相同的灰度图中, 并对其灰度图进行最小值滤波。若  $J$  是无雾霾图像, 除了天空区域, 式(1)的值趋近与 0。

同性滤波在去除图像噪声的同时也会使得一些细节被去除, 减少图像边缘的表现。而导向滤波器[9][10]是从局部线性模型中, 通过导引图像来计算滤波输出。与双边滤波器一样, 引导滤波器使得图像边缘的细节特点保留的更好, 可以将引导图像的结构传输到滤波输出, 从而实现去雾。其最为关键的便是其自适应权重原理。根据其线型滤波公式:

$$q_i = a_k I_i + b_k \quad (2)$$

$$\text{其中 } a_k = \frac{1}{|w|} \sum_{i \in w_k} \frac{I_i p_i - u_k \bar{P}_k}{\sigma_k^2 + \epsilon}, \quad b_k = \bar{P}_k - a_k u_k。$$

在公式(2)中,  $I$  为引导图像,  $q$  为输出图像,  $a_k$  与  $b_k$  两个系数是根据  $I$  和  $q$  共同决定的。对于输入图像, 引导图可分为单通道和三通道的图像。当引导图为单通道即为灰度图时, 由于本文中的测试均为彩色图, 则不予考虑。当引导图为三通道时, 即彩色图, 本文在求  $a$  时将其中的方差  $\sigma$  替换为协方差。

## 2.2. 自动色阶算法

色阶[11][12]就是指用直方图描述的整个图像的明暗信息, 而自动色阶算法主要便是要去除图像中像素值最高的比例以及图像中像素最低的比例, 之后再再将图像所剩下的像素值进行线性映射或者是伽马校正到[0, 255]区间。本文是基于 python 实现自动色阶去雾的, 则先通过 python 中的 image, shape 获取图片的高和宽, 再用 numpy 中的 np.histogram 方法来得到相应通道的直方图。

## 2.3. Django 框架

Django 框架它是一款基于 python 开发的全栈式一体化 web 应用框架, 其在 2005 年便完成了开源。其目的便是削减代码量, 简单迅速地搭建以数据库为主题的复杂 web 站点。且 Django 还具有集成访问组件、强大的 URL 映射技术、后台管理自动生成等特点。其组成结构包括管理工具、模型、视图、模板、管理站。

# 3. 基于 Django 框架的 web 图像去雾系统设计

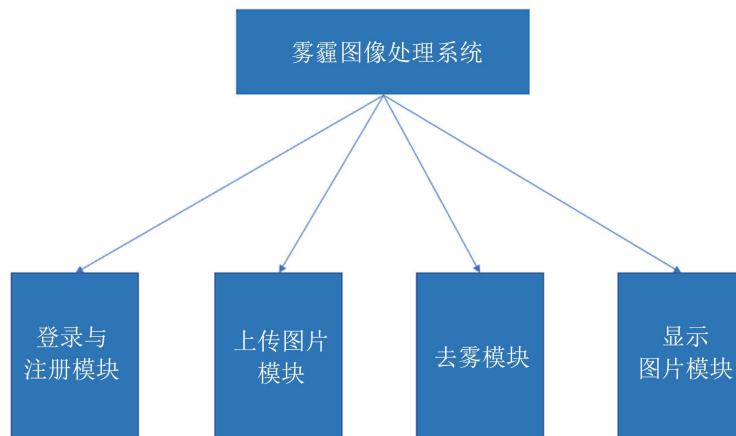
## 3.1. 系统设计

本系统使用 PyCharm 作为开发工具, 用 python 作为开发语言, 并且使用 Django 框架进行 web 开发。其中, 对于 Django 框架还是第一次用, Django 框架使用 python 语言编写的开源的 web 开发框架, 采用 MVT 模式, 其中 M 为模型, V 时视图。

系统实现的是多种去雾方法的比较, 并且能让用户自己调制参数。设计了一个 index 主页面, 以及三个按钮, 分别是通往暗通道去雾, 导向滤波去雾, 以及自动色阶去雾的界面, 并且还设计了显示的页面、有参数框以及去雾按钮。

## 3.2. 模块设计

本系统设计有登录注册模块、上传图片模块、显示图片模块和去雾模块。对于显示模块增加了其对去雾后图片的显示, 更利于对其去雾效果的一个比较。其中系统的大致模块结构图如下图 1:



**Figure 1.** System architecture diagram  
**图 1.** 系统总体架构图

### 3.3. 界面设计

界面设计依据界面设计准则，分为 6 个步骤：1) 用户需求的分析：通过需求模式和概况分析来综合得到本系统的一个大概的模式和概况。2) 分析系统：根据其复杂性以及难易程度等，进行任务的分解确定合理的用户交互方式。3) 环境的分析：就是确定所开发系统的软、硬件所支持的环境和接口。4) 布局设计：先确定屏幕上显示信息的内容以及几面显示的次序和各种跳转按钮和链接，然后进行总体布局设计，和显示结构的设计。5) 帮助信息和出错信息的设计：先决定有哪些错误信息和帮助信息，在确定其显示内容，之后进行出错信息和帮助信息的显示结构设计。6) 确定界面：根据系统用户的特性和需求进行产品分析，确定开发环境得出界面设计机构图，从而确定界面原型。

### 3.4. 数据库设计

本系统主要的功能是对图片的去雾操作，基于 Django 框架链接 MySQL 数据库，其设计包含的表格有存储图片表、用户表和视图表。对于存储图像表，主要有两个字段：图片的名字及格式，所存储图片在服务器上的绝对路径。对于用户表便是传统的用户名和密码。

用户表、图像表和视图表的详细信息分别见表 1~3。

**Table 1.** Users list

**表 1.** 用户表

中文名称	用户表			
物理表名	Users			
主键	ID			
业务主键	无			
索引	IND_TBL_USER_USERCODE(普通索引)			
字段列表				
中文名称	列名	数据类型	主键	非空
姓名	Username	Char	无	√
密码	Password	Char	无	√

Table 2. Images list

表 2. 图像表

中文名称	图像表			
物理表名	Img			
主键	ID			
业务主键	无			
索引	IND_TBL_USER_USERCODE (普通索引)			
字段列表				
中文名称	列名	数据类型	主键	非空
图像名	imgname	Char	无	√
图像地址	HeadImg	ImageField	无	√

Table 3. View list

表 3. 视图表

序号	中文名称	物理名称	备注
1	登录视图	Login	登录系统
2	注册视图	Register	注册界面, 实现账号的注册
3	上传图片视图	Upload	点击上传图片到服务器
4	显示图片视图	Show_img	显示所有图像, 并且还有显示已经去雾的图像
5	处理图片视图	Quwu	通过点击显示图像界面中的按钮实现去雾

## 4. 系统的实现

### 4.1. 模块实现

#### 1) 登录注册模块。

对于注册模块, 用户能通过系统中的注册界面填入自己的账号和密码, 点击注册按钮后来将自己的账号信息存储到数据库中去。先获取到所填文本框中的用户名和密码, 在通过与数据库中的数据进行比较若有相同的用户名, 便通过返回 message 到注册界面提示用户该账号已经被注册过了, 请用户换账号名, 在其注册成功后返回注册成功信息, 并返回到登录界面。

对于登录模块, 同样先获取用户输入到输入框中的账号和密码, 先验证其用户账号是否在数据库中, 若在数据库中, 在验证其所输入的密码, 若密码也相同, 则提示登录成功, 并且进入主界面, 若密码不同, 则返回 message 到登录界面, 提示用户密码输入不正确, 若在数据库中未找到该账号, 则返回 message 到登录界面, 提示用户该账号未注册, 请注册或重新确认输入账号。

#### 2) 上传图片模块

在创建数据库表时, 对存储图片路径的后面应加上 `upload_to = "文件名"`, 在文件的 `setting.py` 中应加入静态变量位置说明。在界面中可通过点击按钮选中要上传的图片, 或者将要上传的图片拖到指定区域, 再点击上传按钮, 后台会返回信息。在点击了上传按钮时, 后台会通过 `request.FILES` 来获取表单中的信息, 其中 `request.FILES` 是一个字典, 之后再将其与已经建立的数据库表相对创建一个数据模型对象, 最后存储到数据库中, 并且存到本地服务器指定路径中去。该模块是通过将其前端所传的图片信息

放到一个字典中去,并传入后端存入数据库中。

### 3) 显示图像模块显示

由于去雾算法的参数不同,本系统有两个不同的界面显示图片。暗通道导向滤波的显示界面有三个参数,参数一是内核矩阵的尺寸(即将图像转换为矩阵处理时的窗口值),通过 `cv2.getStructuringElement(cv2.MORPH_RECT, (sz, sz))` 来实现其向矩阵转化的过程。参数二为透射率的大小,由于导向滤波的透射率会对图像整体像质产生影响,若透射率过大,会使得图像整体向白场过渡,使得其整体模糊,因此需用户先设定一个值,并取算出的值与用户输入值的最小值。参数三为滤波半径,滤波半径即在图像滤波时所用到的半径值。暗通道导向滤波的显示界面如图 2 所示。



Figure 2. Guided filter display interface

图 2. 导向滤波显示界面展示

自动色阶去雾的功能中没有引入参数,而是直接将其中的像素值通过其算法映射。自动色阶的显示界面如图 3 所示。



Figure 3. Automatic level algorithm defogging interface

图 3. 自动色阶算法显示图片界面

## 4.2. 去雾模块

在本系统中,暗通道导向滤波用的是局部线性滤波。根据用户在界面输入的内核矩阵的尺寸,透射率,滤波半径三个参数,计算出大气光学、透射率、去雾后的图。将得到的去雾图片存储到服务器以及数据库中,并将去雾图片的信息通过字典的方式传到前端显示界面去,以便读取图片。自动色阶算法是将其所有像素进行映射的过程。实验中将三通道彩色图像先转换为直方图形式进行映射。暗通道导向滤波和自动色阶算法去雾过后显示结果分别如图 4 和图 5 所示。



**Figure 4.** Dark channel guided filter defogging effect

**图 4.** 暗通道导向滤波去雾显示展示图

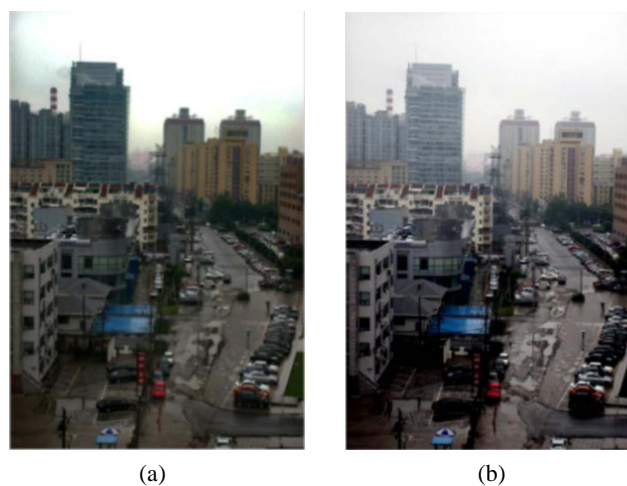


**Figure 5.** Automatic level algorithm defogging effect

**图 5.** 自动色阶去雾效果展示图

### 4.3. 结果测试及性能分析

该实验测试是以系统测试，黑盒测试与功能测试为主。其中测试以本系统展开，测试环境为笔记本电脑与 pycharm。测试范围为系统的去雾、显示图片、上传图片以及登录注册功能。图 6(a)和图 6(b)所示为不同算法的去雾效果对比。



**Figure 6.** (a) Guided filter defogging effect; (b) Automatic level algorithm defogging effect

**图 6.** (a) 导向滤波去雾; (b) 不同色阶去雾

两种去雾算法在性能上, 导向滤波要更为优。对于导向滤波, 当设置图片转换为矩阵的内核半径时, 其窗口值越大, 去雾效果会变差。实验表明, 当其将图片转换为矩阵时的规格在 15~25 之间, 去雾效果最佳。

## 5. 小结

根据图像去雾的实际需求, 在 python Django + bootstrap 框架下设计开发而成的系统, 以美观友好的人机交互界面和较为完善的功能体系, 为用户对图像去雾提供一个方便、快捷的操作。由于每个用户对清晰度的要求不同, 在去雾时可在系统中通过设置参数的方式, 自由选择合适的去雾程度和去雾方案。此外, 该系统引入了平布直方图的做法, 目的是与其他算法进行比较, 便于用户选择出最佳的去雾方法, 和最佳去雾参数, 让用户摆脱在雾天所拍的图片不清晰的困扰。目前基于深度卷积神经网络的图像去雾的理论技术日益成熟, 如何将深度学习技术应用到实际的系统中, 从而提高图像去雾的清晰度和时效性, 是今后研究的方向。

## 基金项目

四川省科技计划项目(重点研发项目)2020YFG0442, 四川省科技计划项目(重点研发项目)2020YFG0453。

## 参考文献

- [1] Oakley, J. and Hong, B. (2007) Correction of Simple Contrast Loss in Color Images. *IEEE Transactions on Image Processing*, **16**, 511-522. <https://doi.org/10.1109/TIP.2006.887736>
- [2] 孙玉宝, 肖亮, 韦志辉, 等. 基于偏微分方程的户外图像去雾方法[J]. 系统仿真学报, 2007, 19(16): 3739-3744.
- [3] He, K., Sun, J., et al. (2011) Single Image Haze Removal Using Dark Channel Prior. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **33**, 2341-2353. <https://doi.org/10.1109/TPAMI.2010.168>
- [4] He, K., Sun, J. and Tang, X. (2013) Guided Image Filtering. *IEEE Transactions on Software Engineering*, **35**, 1397-1409. <https://doi.org/10.1109/TPAMI.2012.213>
- [5] 马啸, 邵利民, 徐冠雷. 基于暗原色先验的雾霾天气图像清晰化算法[J]. 计算机工程与科学, 2018, 40(12): 119-126.
- [6] 吴国鹏, 李盾, 何宇龙, 周纪元. 一种基于暗通道理论的海上图像去雾算法[J]. 光学与光电技术, 2019, 17(6): 45-50.
- [7] 谢立, 熊刚, 于波, 朱凤华, 胡斌. 基于暗通道先验的大气退化图像去雾新方法[J]. 控制工程, 2020, 27(2): 207-211.
- [8] 肖钟捷, 余文森, 程仁贵, 刘长勇. 基于暗通道先验的快速图像去雾算法[J]. 吉林师范大学学报(自然科学版), 2014, 35(3): 106-110.
- [9] 胡晨辉, 吕伟杰, 张飞. 导向滤波优化的单幅去雾算法[J]. 传感器与微系统, 2017(10): 135-138.
- [10] 董宇飞, 杨燕, 曹碧婷. 基于导向图优化的单幅图像深度去雾算法[J]. 计算机应用, 2017, 37(1): 268-272+277.
- [11] 胡学龙. 数字图像处理[M]. 第三版. 北京: 电子工业出版社, 2014.
- [12] 冯红波, 李萍, 李波. 基于自动色阶和多尺度 Retinex 彩色图像增强算法[J]. 无线电工程, 2019, 49(10): 910-914.