

一种基于幂指数拉伸的去雾算法

李忠国¹, 吴昊宸¹, 付启高², 席 茜¹, 吴金坤¹

(1. 江苏科技大学机械工程学院, 镇江 212000; 2. 贵州风雷航空军械有限责任公司, 安顺 561000)

摘 要: 比较同一场景无雾和有雾时图像 RGB (Red-green-blue) 三通道和 HSV (Hue-saturation-value) 三通道的变化, 提出一种基于幂指数拉伸的去雾算法。首先将图像从 RGB 变换到 HSV 空间, 将饱和度分量和亮度分量分别作 1~3 的幂指数拉伸和调整, 将拉伸变换后分量生成 HSV 图像再变换到 RGB 空间, 生成增强后的去雾图像。以饱和度均值、亮度指标、信息熵和对比度作为去雾评价的指标, 确定最优的拉伸幂指数组合。然后使用最优幂指数完成去雾处理, 同时根据图像饱和变化的阈值或时间间隔长度决定是否重新寻找最优拉伸幂指数。最后使用 Python 软件, 借助多进程编程实现本文去雾算法。当图像分辨率为 400 像素 × 300 像素时, 树莓派上运行时幂指数参数寻优用时为 5.077~6.160 s, 单帧图像去雾用时第 1 帧时间长为 0.308 s, 其余时间为 0.077~0.168 s, 结果验证了本文算法的实时性。

关键词: 去雾增强; HSV 空间; 幂指数拉伸; 去雾评价; 多进程编程

中图分类号: TP391 **文献标志码:** A

Defogging Algorithm Based on Power Exponent Stretching

LI Zhongguo¹, WU Haochen¹, FU Qigao², XI Qian¹, WU Jinkun¹

(1. College of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212000, China; 2. Guizhou Fenglei Aviation Armament Co., Ltd., Anshun 561000, China)

Abstract: After comparing three channels of RGB (Red-green-blue) and three channels of HSV (Hue-saturation-value) in the same scene between clear and fog pictures, a haze removal algorithm based on power exponent stretching is proposed. Firstly, the image is transformed from RGB to HSV space. Then the saturation component and the brightness component are exponentially stretched with power of 1—3, and then they are both adjusted to their suitable range. After stretching transformation of saturation and brightness, the image is transformed from HSV to RGB space to generate enhanced defogging images. Taking the mean value of saturation, brightness index, information entropy and contrast as defog evaluation indexes, the optimal stretching power index combination is determined. The optimal power index combination is used to complete the defogging process. At the same time, it is decided whether to find the optimal power index again according to the change of image average saturation or the length of time interval. Finally, the fog removal algorithm is implemented by multi-process programming with the Python

基金项目: 国家重点研发计划 (2018YFC0309100)。

收稿日期: 2021-03-12; **修订日期:** 2021-11-01

software. When the image resolution is $400 \text{ pixel} \times 300 \text{ pixel}$, it takes 5.077—6.160 s to optimize the power index parameters on the raspberry PI. For one frame defogging, the first frame takes longer time of 0.308 s. The other frames take 0.077—0.168 s to removal haze for a single frame.

Key words: haze removal enhancement; HSV space; power exponential stretching; haze removal evaluation; multiprocessing programming

引言

近年来由于空气污染加剧,雾霾天气出现越来越频繁,对交通构成较大安全隐患^[1]。由于大气薄雾导致图像清晰度降低,甚至模糊^[2],各类计算机视觉应用,包括监视、目标分类、跟踪和识别等都会受到影响^[3],研究应用于便携装置的去雾算法具有一定的工程意义。

现有的图像去雾技术大致可分为两类:一类是基于大气散射的物理模型,另一类是基于图像的增强^[4]。常规的图像增强方法有 Gamma 校正、直方图均衡、同态增析以及基于小波的多尺度增强等,这些方法主要增强图像的边缘等高频信息,针对图像对比度低、视觉效果差的问题,通过增强模糊图像的颜色和对比度,使处理后的图像具有良好的视觉效果^[5]。基于图像成像物理模型的算法是基于大气成像模型,有基于暗通道的图像去雾以及导向滤波去雾等方法,涉及到透射率估计、雾浓度及景深估计等,用时较长^[6]。该方法对大多数室外图像具有良好的去雾效果,但对于未建立暗区的明亮区域,如天空或白色物体,则会造成颜色失真^[7]。基于深度学习的方法去雾也是一个新的思路,但需要建立大量的样本训练学习,并且能否满足不同浓度、不同光照等的自适应去雾要求尚未可知。

近年来图像去雾研究较多。邢世宏等^[8]在海上薄雾条件下研究了快速去雾增强方法,对亮度通道进行了高斯卷积,对饱和度进行拉伸得到了增强图像,针对海上图像背景相对简单的场景取得了较好的效果。郭青山等^[9]基于大气散射模型采用 Dehaze Net 与边缘检测引导滤波图像去雾,效果较好;但由于涉及到网络的卷积运算等,计算量较大,实时性得不到保证。林宇轩等^[10]选择现场可编程门阵列(Field-programmable gate array, FPGA)作为去雾计算的处理器。辛佳佳^[11]利用 FPGA 驱动网口完成 4 路图像信息同步传输,利用浮点数字信号处理器(Digital signal processor, DSP)完成复杂运算实现图像去雾。Ju 等^[12]分析了大气散射模型的固有缺陷,提出了一种改进的大气散射模型的快速图像去雾算法。图像去雾处理在航空、航海等交通领域有较大的应用需求^[13]。在交通领域的去雾应用都有较高的实时性和去雾质量的要求,使得许多有效的去雾算法,如暗原色、暗通道理论等遇到困难^[14]。

目前图像去雾算法的硬件实现大致可以分为基于 PC 机的实现方案、基于 DSP 的实现方案和基于 FPGA 的实现方案。基于 PC 机的图像去雾处理成本高、功耗大,且 PC 机体积较大,不适用于车载场合;基于 DSP 的实现方案应用较多,开发难度稍大;基于 FPGA 的方案由于通信带宽有限,容易出现速度瓶颈。

树莓派是一款运行 Linux 系统的微型电脑主板,采用了一个 4 核处理器,具备所有 PC 的基本功能,并且芯片性能好、功耗小、体积小以及性价比高^[15],适合车载使用。同时,树莓派是开源硬件,可以运行开源的 Python 软件,而 Python 软件又可以使用功能强大的 OpenCV 图像处理库,适用于当前交通领域的需要。因此,本文以树莓派作为处理器,研究一种基于幂指数拉伸的图像去雾算法,实现简单、速度

快、效果好。

1 有雾图像和无雾图像差异比较

景物的图像受到雾的影响,会发生哪些变化是增强去雾算法设计的依据。为了寻找有雾图像和无雾图像的差异,在有雾和无雾两种情况下采集了相同的操场场景的图像,如图1所示,图像分辨率为550像素 \times 650像素。图1中有雾和无雾图像顶部都是天空区域,色彩较为单一;下面是树木和塑胶场地,色彩较为丰富;后面有建筑物和汽车等,层次较多。为了统计天空区与和下方景物区域的数值差异,在无雾图像楼顶上方某一位置划一横线(图像纵轴像素坐标为280像素),见图1(a)中间位置。以图1两幅图像为例,从上到下统计每1行图像在RGB(Red-green-blue)空间三分量数值的区别,如图2所示,图1中横线为图2中垂直的点画线。

由图2可见:在天空区域,无雾图像的 R 和 G 两分量数值小于有雾图像, B 分量各有大小;在下面景物的大部分区域中,无雾图像 R 、 G 、 B 三分量均小于有雾图像;而横线下方近距离区域内,无雾图像 R 、 G 、 B 三分量大于有雾图像,这是由于横线下方有从左到右的云层。分别从云层和天空无云区域取像素坐标为(150, 150)和(450, 150)两点,其 R 、 G 、 B 数值分别为(215, 225, 231), (87, 158, 210), 因而有云层的情况下三分量数值较高,没云层时 R 、 G 分量下降较大, B 分量稍有下降。在有雾图像天空区域取一个点,其 R 、 G 、 B 三分量为(36, 48, 36)。有雾区域的三分量的数值较小,并且三分量数据之间差距较小。

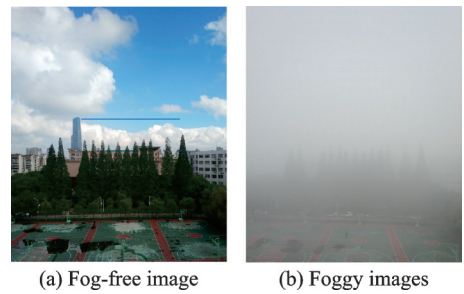


图1 操场场景的无雾和有雾图像

Fig.1 Fog-free and foggy images of the playground scene

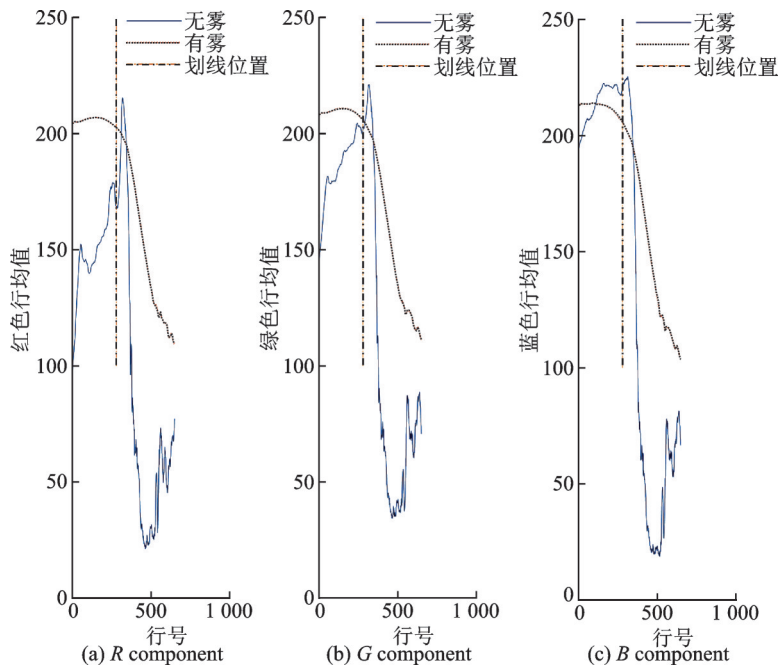


图2 无雾和有雾图像 R 、 G 、 B 三分量的比较

Fig.2 Comparison of three components R , G and B of fog-free and foggy images

将图像从RGB空间转换到HSV(Hue-saturation-value)空间。同样统计每一行的色度、饱和度和亮度的均值,如图3所示(图像数据为Double型)。

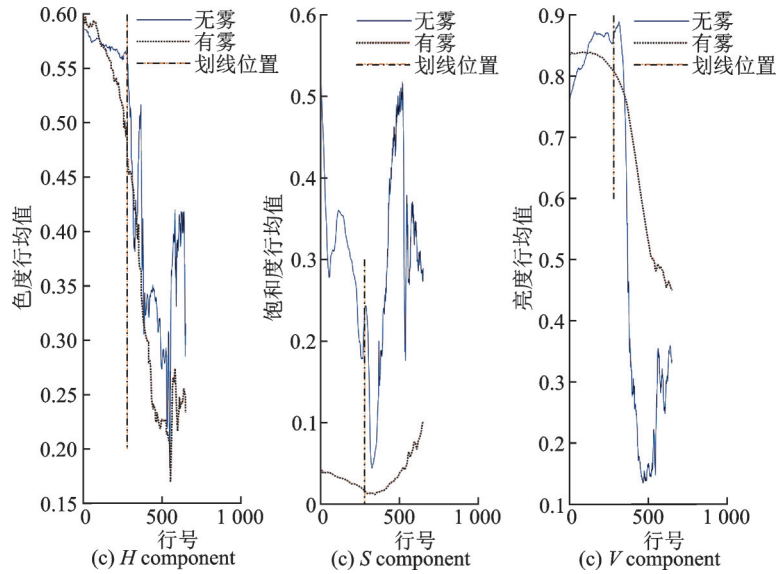


图3 无雾和有雾图像H、S、V三分量比较

Fig.3 Comparison of three components H, S and V of fog-free and foggy images

由图3可见,有雾图像相对无雾图像色度H分量总体上数值稍有降低,但差异不明显,且无规律性。有雾图像的饱和度S分量在整个统计范围内均明显小于无雾图像,且无雾图像色度S分量的最小值是云层区域。有雾图像在景物区域的亮度V分量大于无雾图像,而在划线区域周边无雾图像V分量高于有雾图像,其原因是云层亮度较高。无雾图像和有雾图像亮度V分量的差异曲线和RGB空间中B分量的差异曲线高度类似。无雾图像和有雾图像的原图和H、S、V三分量图如图4所示。

由图4可以看出,无雾图像的云朵饱和度S分量较低,和有雾图像天空的S分量接近;无雾图像的蓝



(a) Original images (b) Effect of H component (c) Effect of S component (d) Effect of V component

图4 无雾图像和有雾图像的H、S、V分量效果比较

Fig.4 Effect comparison of H, S, V components of fog-free and foggy images

色天空区域亮度 V 分量和有雾图像接近, 云朵的亮度大于有雾图像天空的亮度。无雾图像云朵中取 1 点, 其 H 、 S 、 V 分量为 (0.576 9, 0.056 3, 0.905 9); 蓝色天空中取 1 个点, 其 H 、 S 、 V 分量为 (0.570 5, 0.585 7, 0.823 5)。有雾图像天空中取 1 点, 其 H 、 S 、 V 三分量为 (0.533 3, 0.023 7, 0.827 5)。根据上述数据特点可对有雾图像的天空区域做出大致判断。为了验证变化规律的普遍性, 现选取另一对高楼场景的无雾图像和浓雾图像, 如图 5 所示。对图 5 中的高楼、天空、草坪和路灯 4 个物体分别取点 A 、 B 、 C 、 D , 比较这 4 个点的 RGB 和 HSV 空间分量在无雾与有雾时的区别, 如表 1 所示。

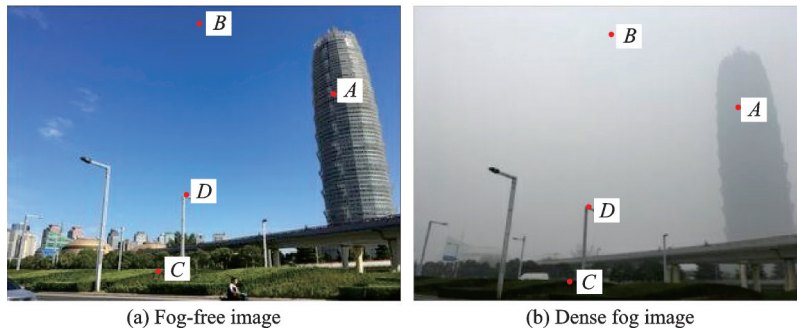


图 5 高楼场景的无雾图像和浓雾图像

Fig.5 Fog-free image and dense fog image of the building scene

比较有雾图像和无雾图像的 R 、 G 、 B 分量可以发现, B 分量变化不大, 但高楼和天空的 R 和 G 分量在有雾时增加, 草坪和路灯 R 和 G 分量在有雾时减小 (无雾时有光照)。因而 R 、 G 、 B 三分量在去雾算法设计中的信息规律性小。而在 HSV 空间中 H 的变化呈无规律性, 有雾图像 S 和 V 两分量低于无雾图像, 所以含有较多可以恢复的信息。本文在 HSV 空间通过变换 S 和 V 分量设计去雾增强算法。

表 1 高楼场景不同点的 RGB 和 HSV 空间分量值

Table 1 Values of RGB and HSV space components of different points in the building scene

| 物体 | | R | G | B | H | S | V |
|-------|----|-----|-----|-----|--------|---------|---------|
| 高楼(A) | 无雾 | 66 | 78 | 82 | 0.3216 | 0.196 1 | 0.443 1 |
| | 有雾 | 173 | 176 | 181 | 0.7098 | 0.043 1 | 0.427 4 |
| 天空(B) | 无雾 | 56 | 93 | 164 | 0.6431 | 0.658 8 | 0.431 4 |
| | 有雾 | 230 | 234 | 237 | 0.9294 | 0.031 4 | 0.403 9 |
| 草坪(C) | 无雾 | 90 | 101 | 43 | 0.4588 | 0.478 4 | 0.141 2 |
| | 有雾 | 33 | 34 | 28 | 0.1373 | 0.176 5 | 0.133 3 |
| 路灯(D) | 无雾 | 149 | 188 | 217 | 0.8510 | 0.313 7 | 0.403 9 |
| | 有雾 | 141 | 139 | 139 | 0.5765 | 0.011 8 | 0.117 6 |

2 幂指数拉伸的图像去雾增强算法

从图 2 中可以看出, 在 HSV 空间, 有雾图像和无雾图像的 H 分量差异不明显, 说明颜色基本保持不变。饱和度 S 在有雾图像中受距离影响较大, 近距离的饱和度较大, 距离远的饱和度较小, 同时也和景物的原始饱和度有关。亮度 V 分量和 RGB 空间中三分量变换相似, 有雾图像的亮度由远到近 (雾由浓

到薄)数值降低,且受景物影响较小。而且有雾图像相对无雾图像其饱和度和亮度分量的变化斜率变小,即相邻行间差异变小。若要复原成原来的无雾图像,需要增加2个分量的变化斜率。由于2个分量都小于1,故将2个分量作幂指数运算可以增加分量之间的差异。本文基于幂指数拉伸原理提出一种图像去雾增强算法,算法流程如图6所示。

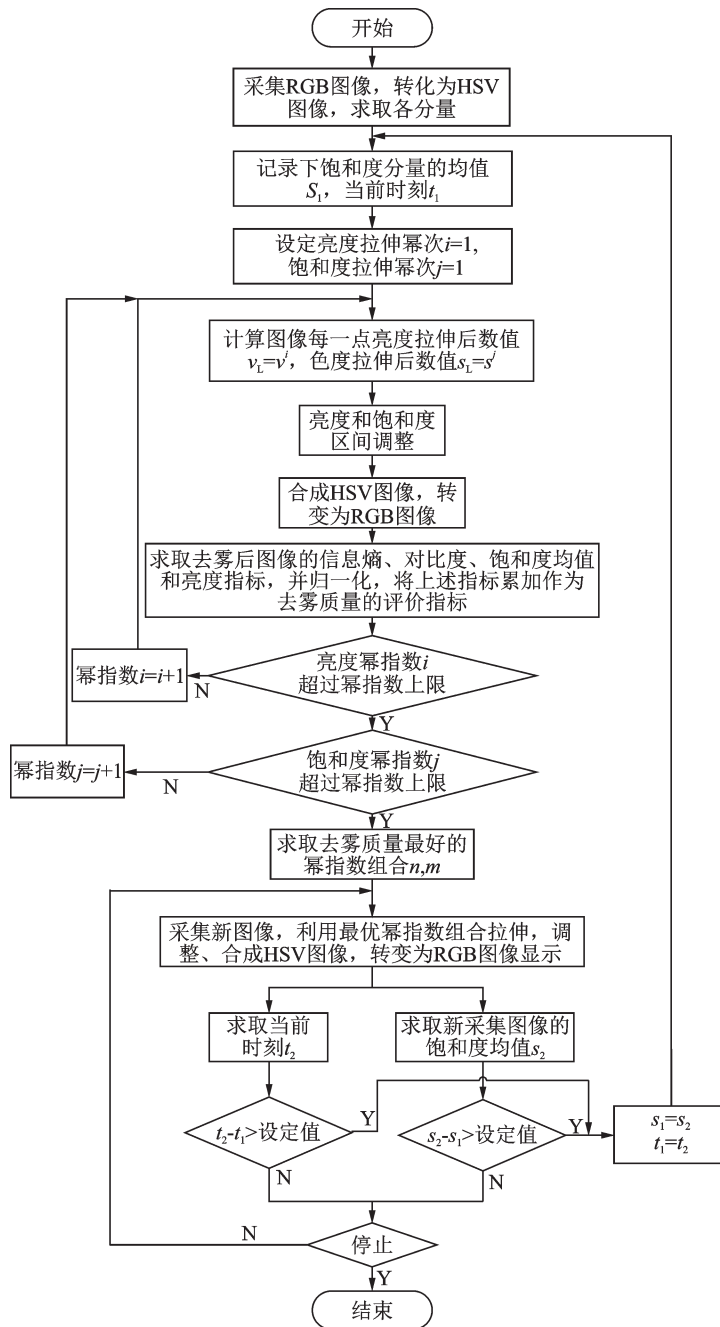


图6 幂指数拉伸增强去雾算法流程图

Fig.6 Flow chart of power exponential stretching based dehazing enhancement

算法具体步骤如下:

(1) 颜色空间转换。将采集的 RGB 有雾图像转化为 HSV 图像,提取色度分量 H 、饱和度分量 S 和亮度分量 V ,除以 255 转换为双精度数据,计算图像饱和度的均值 S_1 ,记录当前时间 t_1 。

(2) 幂指数拉伸。对景物区域的饱和度分量 S 和亮度分量 V 分别作幂运算 S^i 和 V^j ,饱和度幂指数 i 和 j 从 1 到 3 变化,然后调整饱和度和亮度数值分布区间拉伸为 $[0,1]$ (后面三分量数据都按照 $[0,1]$ 区间描述);接着将每个饱和度和亮度拉伸组合分别结合原色度分量 H ,乘以 255 恢复到原来区间,生成 HSV 图像,再转换为 RGB 图像,得到去雾后图像。

(3) 拉伸去雾评价和寻优。本文采用信息熵、对比度、饱和度均值和亮度指标的累加和作为标准评价去雾质量的好坏。

信息熵可以用来表示图像所含的信息量多少。所含信息多,则熵值往往越大;反之,所含有的信息少则熵值较小。雾天图像,物体的轮廓边缘信息会变得模糊不清,各像素的亮度值更接近于大气光值,整幅图像的亮度值会更加集中,图像的信息熵熵值小。相反,无雾图像,图像的熵值较大。信息熵计算公式为

$$E = - \sum_{i=1}^L p_i \log p_i$$

式中: L 为亮度值的个数; p_i 为第 i 个亮度值在图像中出现的概率。

图像对比度指的是一幅图像中最暗的黑和最亮的白之间的级别数。级别越多代表黑白之间的等级越多;相反,级别越少则代表黑白之间的分级越少。通常来说,黑白差异越大,分级越多,则整幅图像的清晰度越高,即对比度数值越大,图像就拥有更高的清晰度。去雾后的图片相较于雾图的对比度应该要得到提升。对比度的计算公式为

$$\text{contr} = \sum_{\delta} \delta(i,j)^2 P_{\delta}(i,j)$$

式中: $\delta(i,j)$ 表示相邻像素间的灰度差; $P_{\delta}(i,j)$ 表示相邻像素间的灰度差为 δ 的像素分布概率。

图像亮度均值太大或太小都不合适,因此求取亮度均值为 0.5(使得图像不太亮也不太黑,后续更加便于处理得到精确清晰的图像)、亮度方差为 0.157 的正态分布函数值 HS 作为亮度指标,然后将每个指标分别进行归一化,相加后作为每个组合去雾效果的评价指标。饱和度均值也反映去雾效果。

因为寻优参数为整数,组合数有限,为了提高算法速度,最优幂指数的选取采用栅格寻优方法。将饱和度拉伸幂指数和亮度拉伸幂指数都从 1 变化到 3,共 9 种组合拉伸。从这 9 种组合中根据去雾评价指标选取最优的一组幂指数组合,得到亮度拉伸的幂指数 n 和饱和度拉伸幂指数 m 。

(4) 新采集 RGB 有雾图像转化为 HSV 图像,提取色度、饱和度和亮度分量,并计算饱和度的均值 S_2 。如果 $S_2 - S_1$ 大于规定阈值 δ (本文选择 δ 为 0.039),则针对新采集的 RGB 有雾图像重复步骤(2~5),即重新计算最优幂指数;如果 $S_2 - S_1 \leq \delta$,则利用步骤(3)所得最佳亮度拉伸的幂指数 n 和饱和度拉伸幂指数 m 完成新采集图像的去雾。记录当前时间 t_2 ,如果 $t_2 - t_1$ 超过规定时间间隔(本文选择时间间隔为 30 s),也启动寻找最优参数组合程序。

3 数据处理和分析

3.1 去雾效果分析

为验证幂指数组合算法的去雾效果,将图 1(b)浓雾图像的饱和度分量进行 S^3 运算后作 imadjust 调整,亮度进行 V^2 运算后作 imadjust 调整,去雾效果如图 7 所示。

由图 7 可见,去雾后图像在天空区稍有马赛克效应,下方的景物区色度和清晰度都有较大提升。为了检验算法的适应性,采用该方法和暗通道去雾算法及导向滤波去雾算法对树林图像进行去雾处理,结果如图 8 所示,可见本文方法去雾效果有提升。

因为图 8 层次较多,将其作为典型图像进行效果分析,去雾后的评价指标如表 2 所示。从表 2 可以看出,相对暗通道去雾算法和导向滤波去雾算法,本文算法亮度指标有所减小,但衡量图像鲜艳程度的饱和度均值、信息熵和对比度都有明显提高,证明本文算法具有一定的优越性和适应性。如果对亮度有较高的要求,可增加评价指标中的亮度指标的权系数,筛选出满足要求的幂指数来实现。为了验证算法的适应性,另取 4 幅有雾图像,去雾效果如图 9 所示,总体而言本文算法去雾效果明显,图像质量有较大提升。



图 7 图 1(b)幂指数拉伸后去雾效果图
Fig.7 Dehazing image of Fig.1(b) after power exponent stretching

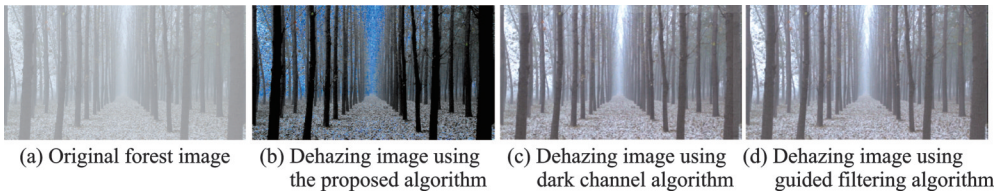


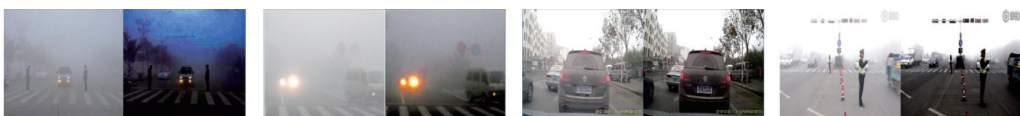
图 8 3 种去雾方法效果比较

Fig.8 Dehazing effect comparison of three methods

表 2 3 种去雾算法指标对比

Table 2 Comparison of defogging indicators of three algorithms

| 指标 | 本文算法 | 暗通道算法 | 导向滤波算法 |
|-------|---------|---------|---------|
| 亮度指标 | 0.626 0 | 0.754 6 | 0.714 5 |
| 饱和度均值 | 0.140 1 | 0.076 1 | 0.072 0 |
| 信息熵 | 2.916 1 | 2.693 3 | 2.550 1 |
| 对比度 | 0.034 7 | 0.026 2 | 0.024 1 |
| 累加和 | 3.716 9 | 3.550 2 | 3.360 7 |



(a) Dehazing effect of scene 1 (b) Dehazing effect of scene 2 (c) Dehazing effect of scene 3 (d) Dehazing effect of scene 4

图 9 本文算法对于不同场景去雾效果比较(左边为有雾图像,右边为去雾图像)

Fig.9 Comparison of defogging effects in different scenes (fogging images on the left, defogging images on the right)

3.2 去雾实时性分析

在 Intel(R)Core(TM) i7-8565 CPU @1.80 GHz 1.99 GHz, 16.0 GB RAM 的笔记本电脑上运行相应的 Python3.7 软件程序, 摄像头采用蓝色妖姬 USB 摄像头, 图像分辨率为 400 像素 \times 300 像素。为了提高程序计算效率, 幂指数寻优采用多进程编程, 多进程寻优程序流程图如图 10 所示。

幂指数寻优用时在 4.129~4.577 s 之间, 单帧图像去雾用时在 0.022~0.120 s 之间, 多数情况是 0.051 s 左右, 具有较好的实时性。针对图 7 中有雾图像, 暗通道去雾算法由于需要暗通道估计, 用时最

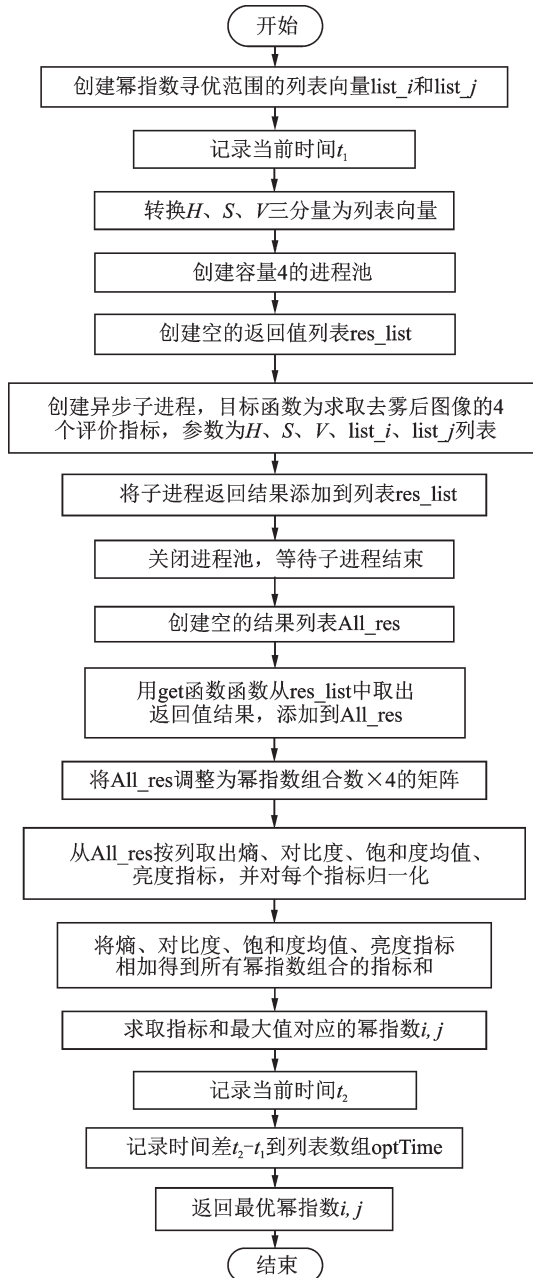


图 10 幂指数寻优多进程编程流程图

Fig.10 Multi-process programming flowchart for power exponent optimization

长,约0.045 s。导向滤波算法速度最快,约0.026 s。幂指数参数最优组合为(1,2)时,本文算法用时0.028 s;幂指数组合为(3,3)时,用时稍有增加,为0.0318 s。尽管如此,本文算法和导向滤波算法用时都远远低于暗通道去雾算法。

树莓派相对PC机处理能力较弱,其实时性问题不能忽视。本文选用的树莓派为4B型,4 GB内存,具有4核处理器。运行相同的程序,幂指数寻优用时为5.077~6.160 s。单帧图像的去雾第1帧时间较长,为0.308 s,其余帧时间为0.077~0.168 s,多数帧去雾处理时间为0.09 s左右,各帧去雾时间如图11所示。由于人的视觉暂留作用,除了第1帧外感觉不到明显的卡顿,故认为能满足多数情况下的去雾实时性要求。若改用树莓派广角CSI视频接口摄像头,与蓝色妖姬USB摄像头相比实时性没有明显区别。

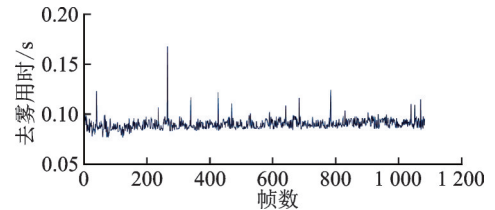


图11 树莓派每帧图像去雾用时统计

Fig.11 Raspberry Pi defogging time statistics for each frame of image

4 结束语

本文提出了在HSV空间通过对饱和度和亮度分量进行幂指数拉伸的方法进行去雾。幂指数是在一定范围内通过多进程编程按照去雾评价指标寻优得到。根据图像饱和度的变化和时间间隔的长度,运行幂指数寻优程序,确定最佳幂指数组合,因而该去雾算法可以适应一定范围内雾浓度的变化和景物层次的变化。确定了最优幂指数后的树莓派图像去雾用时约0.09 s,能够满足多数情况下的车载装置实时性要求。幂指数寻优用时还可通过优化评价指标、减小图像的像素尺寸以及去掉去雾无关的输出和计算等方法进一步降低。本文的研究结果表明,通过对有雾图像的饱和度和亮度分量的简单变换可以快速改善图像的质量,取得较好的去雾效果。本文的去雾算法可为海陆空交通运输安全保障提供技术支持。

参考文献:

- [1] 刘桂丽,黄文博,陈培文,等.高速公路雾、霾净化技术研究[J].公路交通科技,2016,33(2):143-150.
LIU Guili, HUANG Wenbo, CHEN Peiwen, et al. Research of fog and haze purification technology for expressway[J]. Journal of Highway Communication Technology, 2016, 33(2): 143-150.
- [2] LI J W, LIU L, JIANG J W, et al. Image fast dehazing algorithm based on dark channel prior improvement[J]. ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020. DOI: 10.5194/isprs-archives-XLII-3-W10-585-2020.
- [3] SABIR A, KHURSHID K, SALMAN A. Segmentation-based image defogging using modified dark channel prior[J]. EURASIP Journal on Image and Video Processing, 2020(8): 492-505.
- [4] XIN Jiajia, WANG Yu. Adaptive demisting algorithm based on Polari metric image[J]. Materials Science and Engineering, 2018, 394(5): 52-71.
- [5] DU Hongchun. Research on image de-disturbing algorithm based on dark channel prior and anisotropic Gaussian filtering[J]. Concurrency and Computation: Practice and Experience, 2018, 30(24): e4933.1-e4933.11.
- [6] JIANG Yutong, SUN Changming, ZHAO Yu, et al. Fog density estimation and image defogging based on surrogate modeling for optical depth[J]. IEEE Transactions on Image Processing, 2017, 26(7): 3397-3409.
- [7] MA Ruiqiang, ZHANG Shanjun. An improved color image defogging algorithm using dark channel model and enhancing saturation[J]. Optik, 2019, 180: 997-1000.
- [8] 邢世宏,杨晓东,王小海.海上薄雾条件下退化图像的快速Retinex增强算法[J].导航与控制,2015,14(3):112-115,97.
XING Shihong, YANG Xiaodong, WANG Xiaohai. A rapid Retinex enhancement algorithm of images degraded in mist [J]. Navigation and Control, 2015, 14(3): 112-115, 97.

- [9] 郭青山, 黄玉清. 基于DehazeNet与边缘检测均值引导滤波图像去雾算法[J]. 传感器与微系统, 2020, 39(1): 150-153.
GUO Qingshan, HUANG Yuqing. Image dehazing algorithm based on dehaze net and edge detection mean-guided filtering[J]. Transducer and Microsystem Technologies, 2020, 39(1): 150-153.
- [10] 林宇轩. 基于FPGA的实时视频去雾系统研究[D]. 大连: 大连理工大学, 2019.
LIN Yuxuan. Research on real-time video defogging system based on FPGA[D]. Dalian: Dalian University of Technology, 2019.
- [11] 辛佳佳. 基于DSP的同时偏振图像去雾系统设计[D]. 长春: 长春理工大学, 2019.
XIN Jiajia. Design of simultaneous polarization imaging defogging system based on DSP[D]. Changchun: Changchun University of Science and Technology, 2019.
- [12] CHEN Zhixiang, OU BINNA, TIAN Qianyi. An improved dark channel prior image defogging algorithm based on wavelength compensation[J]. Earth Science Informatics, 2019, 12(4): 501-512.
- [13] 王博. 无人艇光视觉感知研究发展综述[J]. 舰船科学技术, 2019, 41(23): 44-49.
WANG Bo. Review of development in perception of unmanned surface vehicle based on optical vision[J]. Ship Science and Technology, 2019, 41(23): 44-49.
- [14] 李红云, 高银. 基于天空区域分割和边界限制 L_0 梯度最小化滤波的图像去雾算法[J]. 南京理工大学学报, 2020, 44(2): 236-245.
LI Hongyun, GAO Yin. Image dehazing based on sky region segmentation and boundary constraint L_0 gradient minimization filtering[J]. Journal of Nanjing University of Science and Technology, 2020, 44(2): 236-245.
- [15] Huayucong. 树莓派详细介绍[EB/OL]. (2015-08-30)[2021-04-19]. https://blog.csdn.net/huayucong/article/details/48106677?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.nonecase.

作者简介:



李忠国(1975-), 男, 博士研究生, 研究方向: 图像测量技术, E-mail: lzg1975@163.com。



吴昊宸(1995-), 通信作者, 男, 硕士研究生, 研究方向: 计算机视觉, E-mail: 1054473846@qq.com。



付启高(1996-), 男, 学士, 研究方向: Python编程。



席茜(1993-), 女, 硕士研究生, 研究方向: 计算机视觉。



吴金坤(1997-), 男, 硕士研究生, 研究方向: 数字图像处理。

(编辑: 张黄群)