

文章编号: 2096-1618(2021)02-0138-05

基于FPGA的实时图像去雾系统

陈 龙, 陈子为, 朱美吉, 吴正正, 舒秉礼

(成都信息工程大学电子工程学院, 四川 成都 610225)

摘要:自动驾驶汽车要求准确的环境信息来执行对场景的理解,从而做出准确无误的判断和严格的执行。雾作为影响汽车驾驶最危险的天气现象之一,会导致视频图像信息中的颜色和对比度发生很大的改变。通过估计大气光幕的值和大气光强的值,采用基于环境光照估计的去雾算法,设计开发了基于FPGA的实时去雾系统。该系统充分发挥FPGA并行处理的优势,通过硬件描述语言完成该算法在FPGA平台上的实现,能够较为真实地恢复无雾、清晰的图像,并能够达到实时不卡顿的效果。

关键词:FPGA;去雾;环境光照估计;图像增强

中图分类号:TP391.41

文献标志码:A

doi:10.16836/j.cnki.jcuit.2021.02.002

0 引言

图像信息作为人类感知世界最直观的信息之一,为认识世界提供了极大的帮助。然而,在雾天环境中,由于大气中混浊的介质对可见光的散射,使得光学传感器接收到的辐射强度被衰减,导致图像对比度降低,亮度与色度的动态范围缩小,图像细节模糊不清,同时出现严重的颜色失真,获取的图像视觉效果不能令人满意,难以用于后续的图像识别^[1]。对于视频信息占据主导地位的视频监控、无人驾驶等领域,如果采集到的视频是有雾的,将对后面的图像处理产生极大的不便。因此,雾天条件下图像的恢复具有非常重要的研究意义。

以前对图像去雾增强处理以事后分析居多,难以满足无人驾驶、及时跟踪等实时性要求较高的应用需求。随着FPGA的发展,FPGA被广泛应用于实时去雾领域。文献[2]在FPGA上实现了MSR去雾算法,但去雾效果一般;文献[3]提出了一种快速的去雾算法,并将该算法在FPGA上进行实现,实际效果能达到实时去雾,但还是容易出现色彩失真的问题;随着ARM+FPGA架构的兴起,文献[4]利用HLS(高层次综合)工具实现了暗通道去雾算法在FPGA平台上的移植,能够达到实时去雾的效果,但是在雾的浓度比较大时会出现比较明显的光晕现象。针对暗通道先验去雾算法处理雾天图像高亮度区域(如天空区域)时出现的彩色失真以及复杂场景时输入图像目标边缘会出现光晕现象等问题,提出了一种基于环境光照的去雾算法,该

算法不用估计透射率的大小,而是通过估算大气光幕值和大气光强值,通过计算而去雾,并采用Verilog硬件描述语言在FPGA上实现了该去雾算法。

1 基于环境光照估计的去雾算法

1.1 Koschmieder 定律

1924年,Koschmieder研究了目标物亮度经过大气衰减后的变化规律,提出了关于目标物在距离 d 时的亮度 L 和在靠近目标物时的亮度 L_0 之间的关系,其表达式为

$$L = L_0 \cdot e^{-\beta d} + L_\infty \cdot (1 - e^{-\beta d}) \quad (1)$$

其中, L_∞ 是大气背景光亮度, β 是消光系数, d 为能见度。表明:在雾中,目标物的亮度 L_0 被指数因子 $e^{-\beta d}$ 衰减,背景光通过雾衰减之后的亮度为 $L_\infty \cdot (1 - e^{-\beta d})$ 。

同样的原理运用于摄像头采集到的图像 I ,修改Koschmieder公式后得到的表达式为

$$I = R e^{-\beta d} + A_\infty (1 - e^{-\beta d}) \quad (2)$$

其中, R 表示要得到的无雾图像, A_∞ 表示无穷远时的大气光强,也就是在雾天条件下天空的亮度。

1.2 算法流程

当运用单目摄像头采集图像时,深度信息 d 不可知,并且一般情况下不能通过测量直接得到。因此,定义大气光幕 V ^[6]为

$$V = A_\infty (1 - e^{-\beta d}) \quad (3)$$

通过估算图像中大气光幕 V 的大小计算最后无雾的图像 R 。根据式(3),将式(2)变形后可以得

收稿日期:2020-03-28

基金项目:四川省教育厅科学研究重点资助项目(18ZA0088);四川省科技计划重点研发资助项目(2019YFG0126)

$$R = \frac{I - V}{1 - \frac{V}{A_\infty}} \quad (4)$$

式(4)表明:在没有深度信息的情况下,可以通过估算大气光幕 V 和大气光强 A_∞ 的值来计算最后无雾的图像 R 。因此,整个去雾算法流程如图1所示。

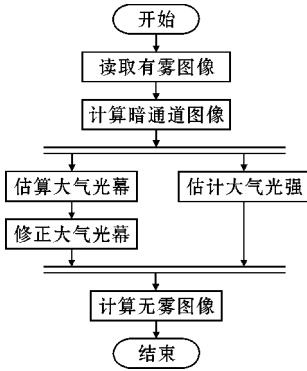


图1 基于环境光照估计的去雾算法流程图

1.3 暗通道先验理论

何凯明等^[7]提出的暗通道先验理论是一个统计规律,通过对大量无雾的户外清晰图像统计时发现:在户外无雾的情况下,绝大多数非天空的场景目标区域里,某些像素点总会存在至少一个颜色通道具有很低的值,即该区域光强度的最小值是个很小的数。因此,对于任意的输入图像 J ,其暗通道可以表示为

$$J^{\text{dark}}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{r, g, b\}} (J^c(y)) \right) \quad (5)$$

其中, J^c 表示 J 的三个颜色通道 RGB 中的某一个颜色通道; $\Omega(x)$ 表示以像素 x 为中心的矩形区域。

暗通道先验理论指出:

$$J^{\text{dark}}(x) \rightarrow 0 \quad (6)$$

1.4 估算大气光幕

为了估算大气光幕 V 的大小,设置如下两个约束条件:

第一,大气光幕必须比零大、比暗通道先验图像 W 小^[8],即

$$0 \leq V \leq W \quad (7)$$

其中, W 为暗通道先验图像,为一灰度图。当输入图像是灰度图像时, W 即为原输入灰度图像;当输入图像是彩色图像时, W 表示的是输入图像 RGB 三个颜色通道中的最小值。

第二,文献^[9]提出了一个新的约束大气光幕的条件——NBPC (no black pixel constraint)。这个约束条件表明:图像去雾后,任一点的局部标准偏差,必须小于该像素点的局部平均值,即

$$f \cdot \text{std}(R) \leq \bar{R} \quad (8)$$

其中, f 是一个常量,通常被设置为 1。第二个约束困难在于它是作用于未知的图像 R 上的。因此,回到式(4),发现它是线性变化的。所以,对等式两边同时求平均值,就有式

$$\bar{R} = \frac{\bar{I} - V}{1 - \frac{V}{A_\infty}} \quad (9)$$

然后再同时求标准偏差,就有

$$\text{std}(R) = \frac{\text{std}(I)}{1 - \frac{V}{A_\infty}} \quad (10)$$

于是,整理后得到的表达式如式:

$$V \leq \bar{I} - f \cdot \text{std}(I) \quad (11)$$

最后,加入常量 P ,使得去雾得到的图像更具真实性。于是有了 V 的约束公式:

$$V = P \cdot \min(W, \bar{I} - f \cdot \text{std}(I)) \quad (12)$$

由于均值滤波不能保留图像边缘信息^[10],而中值滤波不仅保留了原图的边缘信息,而且还会随雾的浓度变化而变化,提升了图像恢复之后的清晰度,因此,采用中值滤波代替均值滤波。此外,为了减少因图像滤波而引起的光晕效应,同时对输入有雾图像的暗通道图像计算标准偏差,得到了大气光幕的约束公式:

$$V = \max(\min(p|M - \text{std}|, W), 0) \quad (13)$$

其中, P 是用来控制去雾强度的量,通常设置为 85%~99%; W 是暗通道图像, M 是中值, std 是标准偏差,两者都是在先获得暗通道图像后,再对暗通道图像的每一列求中值 M 和标准偏差 std 。

1.5 修正大气光幕

对整幅图像而言,大气光幕的值并不完全相同。众所周知,雾并不是均匀的,距离不同、雾的浓度不一样,它会随距离变远而迅速呈指数增加。因此,在实际应用时,必须将大气光幕进行有效修正,减少其与真实大气光幕之间的偏差。

雾天情况下图像具有如下特点^[11]:雾在图像的最远端也就是天空区域会达到一个最大值,此时雾的强度最大,去雾程度也必须达到最大;雾在图像的最近端也就是靠近摄像头处会达到一个最小值(接近 0),此时雾的强度最小,去雾程度也必须达到最小;从天空区域到摄像头之间,雾是非线性指数衰减变化的。

根据雾天图像的指数衰减效应,设置一个校正函数 $K(0 \leq K \leq 1)$,使越靠近天空的地方 K 越大,图像恢复得越多;越靠近摄像头的地方 K 越小,图像恢复得越少^[12]。由于雾是按照指数关系成非线性衰减,因此

该校正函数也应是非线性形式。校正函数的具体求取方法如下。

首先,分别求得输入图像的最大灰度值和最小灰度值,校正阈值 T 设置为

$$T = (I_{\max} + I_{\min}) / 2 \quad (14)$$

然后,根据输入图像的灰度值 $I_g(x)$ 的大小进行校正,取 $\rho = I_g(x) - T$,大气光幕校正函数 K 为

$$K = \begin{cases} \frac{\rho}{|100 - \rho|} & I_g(x) > T \\ 1 & I_g(x) \leq T \end{cases} \quad (15)$$

最后,对大气光幕进行校正:

$$V_{\text{final}} = V \cdot K \quad (16)$$

1.6 估算大气光强

关于大气光强 A_{∞} 的估算,方法有很多,有直接选择输入雾图中亮度最亮的值作为大气光强 A_{∞} 的值,这种方法虽然简单但是不太合理;也有在获得暗通道图像的基础上,从暗通道图像里面按照亮度大小取亮度最大的前0.1%的像素点,然后再从原输入雾图中选取与这些像素点对应的像素值,最后取像素点的最大值作为大气光强 A_{∞} ,这个方法比较合理,但是需要在设计时预留两块不同的图像存储区域,并且需要高速地在两块存储区域内完成数据的交换,这增大了系统开销,影响了算法的实时性能。

为了简化大气光强 A_{∞} 的估算,同时考虑到方便FPGA的实现,本文采用的方法是:先计算输入有雾图像的暗通道图像 $H_{\text{dark}}(x, y)$,然后选择暗通道图像中最亮的值作为大气光强 A_{∞} 值。同时,考虑到实际情况下可能存在白光物体或者原输入图像中可能存在光源,为了防止这些情况的干扰,设置阈值 240。因此,大气光强 A_{∞} 可以表示为

$$A_{\infty} = \begin{cases} 240 & \text{if } (\max(H_{\text{dark}}(x, y)) > 240) \\ \max(H_{\text{dark}}(x, y)) & \text{else} \end{cases} \quad (17)$$

2 基于FPGA的实时图像去雾系统设计

以 Artix7 系列的FPGA为核心硬件平台进行设计,整个系统主要包括:通过IIC接口对OV5640摄像头进行配置,完成图像的采集;由于数据速率不匹配,采集到的图像数据不能直接存储在DDR3中,而是先经过写FIFO缓存区,然后再缓存到DDR3,整个数据的存储是由DDR3控制器来进行控制;同理,存储后的图像也因为数据速率不匹配的问题,也要先经过读FIFO后才能

进行图像去雾算法处理;最后将处理后的图像经过HDMI时序控制模块输出至显示器上进行显示。通过使用Verilog硬件描述语言,将基于环境光照估计的去雾算法移植到FPGA中并封装成IP核,实现实时图像去雾处理。其系统功能结构框图如图2所示。

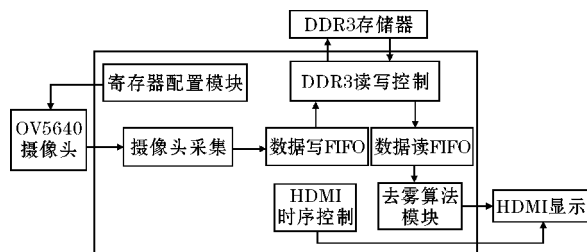


图2 系统功能结构框图

2.1 获取暗通道图像

暗通道图像的获取是对雾天图像做去雾算法处理的重要步骤,根据式(5)表示的暗通道定义,需要对以像素 x 为中心的矩形区域(本文设该区域大小为 5×5)分两次求取最小值,而且这两次求取最小值的顺序可以互换。通过移位寄存器来实现 5×5 滑动矩阵的功能。其最小值比较分为两步实现:首先对RGB三个通道的像素值分别使用通道内的比较,取每个颜色通道中的最小值,即取得R通道像素最小值、G通道像素最小值和B通道像素最小值;然后是通道之间的比较,对前一步得到的最小值图像进行局部区域内的像素最小值排序,求得每个区域的最小值像素点,最后组合就得到暗通道图像。

2.2 大气光幕计算

根据式(13),考虑到标准偏差和方差的关系,通过先求输入图像的方差,再来得到其标准偏差的值。对于中值滤波,只需要将求取暗通道的最小值改成中值滤波即可。然后,再取与暗通道图像进行比较时两者中较小的那一个,得到初步估算的大气光幕 V 。对于大气光幕修正,先求整幅输入图像的灰度最大值和最小值,得到阈值 T ,再将每个像素点的灰度值与 T 进行比较并相减得到 ρ ,通过判断每个像素点的灰度值与阈值 T 大小的比较来确定 K 的值,最后与前面粗略估算的 V 值相乘得到最终的大气光幕值 V_{final} 。

2.3 大气光强计算

根据式(17),对前面得到的暗通道图像的每一个像素点进行排序得到最大值,再将该最大值与设置的阈值 240 进行比较,最后得到大气光强 A_{∞} 。

3 算法仿真与系统测试

3.1 FPGA 算法仿真平台介绍

由于很难使用 Modelsim 软件对一帧或者几帧图像直接读入或者使用 Modelsim 直接输出一帧或者几帧图像,但是 Modelsim 软件可以通过 Verilog 代码对 txt 文件进行读取或写出。考虑到 Matlab 具有强大的图像分析与显示功能,可以结合 Matlab 和 Modelsim 来搭建 FPGA 测试平台,完成对去雾算法的仿真及测试,如图 3 所示。首先通过 Matlab 将图像转换为 txt 文件,其次通过 Verilog 语言编写 Testbench,在 HDMI_CTL 模块的 HDMI 时序控制下使用 Imread 模块读取 txt 文件(图像数据),再经过图像处理算法模块处理一帧或者连续几帧图像数据,通过 Imwrite 模块将算法处理后的数据写入到 txt 文件中。最后通过 Modelsim 对去雾算法模块进行时序分析,同时利用 Matlab 将 txt 文件还原成图像,在 Matlab 中进行进一步分析、显示并保存。这样做的好处是可以通过仿真看到图像处理各个环节的时序、数据变化以及最后算法处理后的图像,从而快速找到算法中的问题。

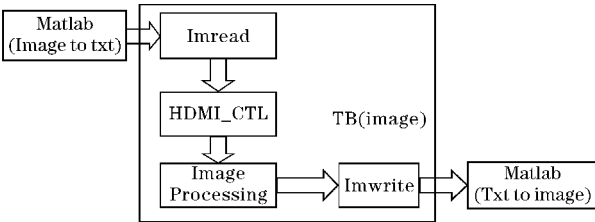
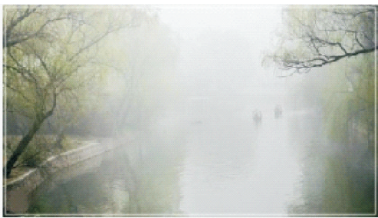
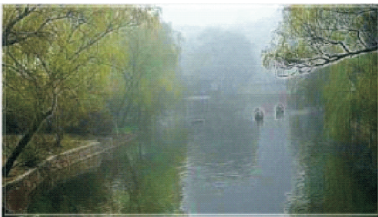


图 3 FPGA 算法仿真平台

利用 Matlab 可以查看去雾算法在 FPGA 中处理前、后的图像,其结果如图 4 所示,其中 image 1 为原始有雾图像, image 2 为去雾后的图像。仿真结果表明,与 PC 端用 Matlab 软件的仿真结果一致,达到了预期的去雾效果。



(a) image 1



(b) image 2

图 4 Matlab 环境下查看算法处理的结果

3.2 系统测试

在有雾的天气条件下,通过 OV5640 摄像头将采集到的有雾图像传输到 FPGA 开发平台上,经过去雾算法模块处理后,将去雾后的图像通过 HDMI 接口外接显示器来显示。

如图 5 所示,分别采集 3 种不同浓度的有雾图像,通过本系统去雾处理后,结果如图 6 所示。对比图 5 和图 6,可以看出基于环境光照的去雾算法有很好的去雾作用,能够较为真实地还原出无雾、清晰的图像,并在硬件开发系统上能够实时不卡顿地运行。



(a)薄雾



(b)中雾



(c)大雾

图 5 原始有雾图像



(a)薄雾去雾后图像



(b)中雾去雾后图像



(c)大雾去雾后图像

图 6 去雾处理后的图像结果

系统占用的 FPGA 资源情况如图 7 所示。由图 7 可以看到,系统占用资源并不是太多,FPGA 仍然剩下不少资源。

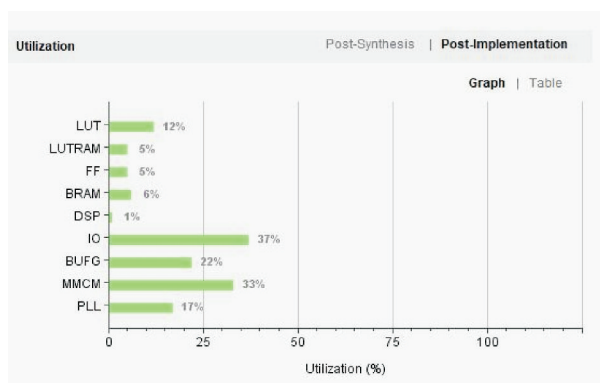


图 7 系统消耗 FPGA 资源情况图

4 结束语

基于环境光照的去雾算法有很好的去雾效果,使得天空区域不再产生色彩上的失真,也不会有光晕现象的产生。通过使用硬件描述语言 Verilog 在 FPGA 上用硬件实现去雾算法,能够较为真实地还原无雾清晰的图像,并能够达到实时不卡顿的效果。对去雾算法的仿真和在 FPGA 上的实现能为无人驾驶等领域后期的图像处理提供更加清晰准确的图像,帮助视频监控系统在面对雾天情况时也能有很好的适应能力。

参考文献:

- [1] 叶崧,杨万扣.改进的基于大气光幕单幅图像去雾算法[J].电子测量技术,2016,39(3):50-55.
- [2] 魏苗.基于FPGA的1080P低质视频实时增强系统的研究[D].南京:南京邮电大学,2017.
- [3] 高全明.基于FPGA架构的视频图像去雾算法研究与实现[D].长春:长春理工大学,2017.
- [4] 齐乐,张小刚,姚航.基于HLS的实时图像去雾实现[J].计算机工程,2016,42(5):224-229.
- [5] Middleton W. Vision Through the Atmosphere [M]. Toronto:University of Toronto Press,1952.
- [6] Tarel J P, Hautière N. Fast visibility restoration from a single color or gray level image [C]. IEEE International Conference on Computer Vision, 2010.
- [7] K He, J Sun, X Tang. Single image haze removal using dark channel prior [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12):2341-2353.
- [8] 余朋,文小琴,汪兆栋,等.改进的异质大气光估计的图像去雾算法[J].计算机工程与设计,2018,39(3):792-797.
- [9] Tarel J P, N. Hautière, Cord A, et al. Improved visibility of road scene images under heterogeneous fog [C]. Intelligent Vehicles Symposium. IEEE, 2010.
- [10] 陈露.基于滤波的暗原色先验图像云雾算法[D].成都:西南交通大学,2015.
- [11] 王玉坤,贾娜.雾天图像清晰化方法研究[J].信息技术与信息化,2019(6):215-216.
- [12] 高健.基于Zynq7000平台的去雾算法研究及实现[D].南京:南京理工大学,2015.

Real-time Image Defogging System based on FPGA

CHEN Long, CHEN Ziwei, ZHU Meiji, WU Zhengzheng, SHU Bingli

(College of Electronic Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Autonomous vehicles are required by accurate environmental information that help to finish an understanding of the scene, then make accurate judgment and implement strict. Fog is one of the most dangerous weather phenomena which results in the change of color and contrast in video image information. Therefore, a real-time defogging system based on FPGA was designed and developed in this paper, which used defogging algorithm based on environmental illumination estimation by estimating the atmospheric veil and sky intensity in the image. The system makes full use of the characteristics of FPGA parallel processing data, and implements the algorithm on the FPGA platform through the hardware description language, and restores the fog-free and clear image more realistically, and can achieve the effect of real-time non-stuttering.

Keywords: FPGA; defogging; environmental illumination estimation; image enhancement