

文章编号:1671-6833(2020)02-0019-06

基于 FPGA 的高分辨率视频图像实时增强去雾系统

刘华军¹, 张瑞珏², 刘建锋², 王 盛², 夏巧桥¹

(1.华中师范大学 物理科学与技术学院,湖北 武汉 430079; 2.上海航天电子技术研究所,上海 201109)

摘要:传统去雾系统大多针对单幅图像进行处理,其稳定性和实时性不足,不仅算法复杂度较高,还存在亮光区域直方图过曝的缺陷,难以应用于实时视频图像修复中。针对该问题,设计了一种基于 FPGA 的 1080P 全高清视频图像实时增强去雾系统。首先,在 RGB 颜色空间采用分块分通道进行直方图统计;其次,通过预设阈值对多通道合并后的直方图进行削减,限制对比度过度放大,而后进行双线性插值计算;最后,对改进后的限制对比度自适应直方图均衡算法(contrast limited adaptive histogram equalization, CLAHE)进行仿真验证,并对系统进行有雾视频、图像以及户外场景的去雾测试。实验结果表明:本系统能实时处理帧速率为 30 f/s、分辨率为 1 920×1 080 的图像,比传统方法去雾效果更好、更稳定,系统功耗低于 5 W,可应用于低功耗领域。

关键词:图像去雾; 视频图像; FPGA; 实时; 低功耗

中图分类号: TP391.41

文献标志码: A

doi:10.13705/j.issn.1671-6833.2019.03.014

0 引言

由于大气散射使得图像质量严重退化、对比度下降、层次不丰富^[1],雾天图像去雾处理成为了国内外的研究热点。根据不同的图像处理方式将去雾算法分为基于物理模型方法和基于非物理模型方法^[2]。睢丹等^[3]提出了基于混沌性的加权滤波算法,有效地降低了雾化、减少了浓雾噪点。郭伟^[4]提出了基于多种先验知识的单幅雾天图像复原算法;何凯明等^[5]提出了“基于暗通道优先的单幅图像去雾”方法,两种方法主要针对单幅图像,去雾效果明显,但算法的时间和空间复杂度较高。何凯明等^[6]进一步提出导向滤波算法,以提高暗通道优先算法的处理速度,但其计算复杂度仍较高。魏先民^[7]提出了自适应直方图均衡算法(adaptive histogram equalization, AHE),该算法自适应好、鲁棒性强,但是其存在过度放大均匀区域噪声的缺陷。基于此,Zuiderveld^[8]提出了限制对比度自适应直方图均衡(contrast limited adaptive histogram equalization, CLAHE)算法,提升了图像的局部对比度并突出图像细节。由于 CLAHE 算法有优异的去雾效果,计算简单且易于

并行化实现,因此得到研究者青睐。杨光等^[9]重点研究了 CLAHE 算法框架及其 FPGA 实现,实验表明该算法具有良好的去雾效果。周卫星等^[10]提出的方法改善了图像的整体视觉效果。文献[11-12]以 FPGA 为硬件平台实现了 CLAHE 算法,系统满足实时性且图像对比度明显增强。孔壮等^[13]利用 CLAHE 算法设计基于高分辨率的视频图像去雾系统,但是图像分辨率仅为 1 004×1 004。周晓波等^[14]利用 FPGA 设计了支持多种分辨率的图像高速实时去雾系统,但其算法复杂度以及系统实时性仍有改善的空间。虽然现有去雾系统取得了不错的去雾效果,但大多数系统针对单幅图像进行处理,稳定性和实时性不足,且存在亮光区域过曝的缺陷。针对这些问题,笔者设计并构建了一种基于 FPGA 的 1080P 全高清视频图像实时增强去雾系统。测试结果表明:系统对分辨率为 1080P 视频图像去雾效果明显、图像层次感强,同时具有较强的实时性和稳定性,系统功耗小于 5 W。

1 系统硬件设计

图 1 为系统硬件框图。高速相机输出图像分辨率为 1 920×1 080,帧频为 30 f/s。相机通过

收稿日期:2019-03-21;修订日期:2019-06-10

基金项目:国家自然科学基金资助项目(61501334);中央高校基本科研业务费(CCNU16A05028)

通信作者:夏巧桥(1987—),男,湖北咸宁人,华中师范大学讲师,博士,主要从事高速实时数据处理平台、LDPC 等方向的研究,E-mail:xqq2947759@163.com。

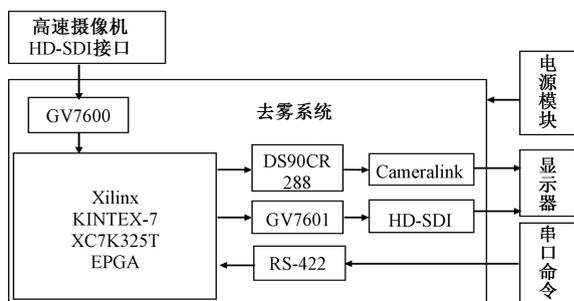


图1 系统框图

Figure 1 System architecture

HD-SDI 接口传输至去雾系统,接口芯片为GV7600,用于视频图像配置,系统可实现图像自检、镜像、电子变焦以及电十字叠加等功能,并完成图像增强去雾。通过 RS-422 串口命令对系统功能进行控制,其经过 FPGA 软核解析并转发给逻辑实现控制 FPGA 内部功能。图 2 为 FPGA 内部功能图,图 3 为系统实物图,其尺寸为 103 mm×60 mm,质量小于 100 g。

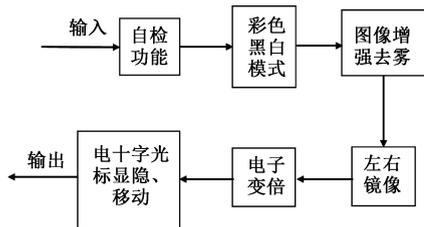


图2 FPGA 功能模块

Figure 2 The functional modules of FPGA



图3 系统实物图

Figure 3 The photo of the system

2 CLAHE 算法原理

CLAHE 算法将图像进行分块,对每个子块采用对比度限幅操作来克服 AHE 算法过度放大噪声的缺陷。CLAHE 算法步骤如下。

步骤 1 将图像划分为 4×4 个大小相同的子块(非重叠图像, $1 \leq i, j \leq 4$),分别对子块进行直方图统计,即统计各灰度级的像素数。

步骤 2 利用截断阈值 β 对直方图进行削峰填谷操作,截断阈值 β 的计算如式(1):

$$\beta = \frac{N}{M} \left(1 + \frac{\alpha}{100} (s - 1) \right), \quad (1)$$

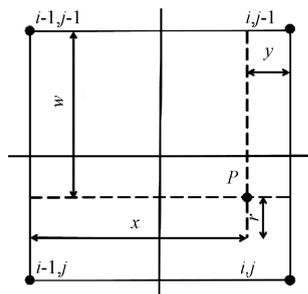
式中: N 为每个子块的总像素数; M 为像素灰度级数 $[0, 255]$;参数 s 为控制指定像素值周边的对比度放大程度,取值范围为 $1 \sim 4$ 的整数; α 为截断系数,取值范围为 $[0, 100]$ 。当 $\alpha = 100$ 时, β 取最大值 $\frac{N}{M}s$,此时的增强效果达到最大。

步骤 3 对削减后的直方图进行均衡,将多出的像素数分配到像素数小于 β 的灰度级中,以 1 作为像素分配步长,通过迭代的方法进行像素重分配,直至像素分配完毕,得到均衡后的结果。

步骤 4 根据直方图均衡结果可得到每个子块直方图映射 f 以及其中心像素点的灰度值,其他点的像素通过与其邻近的 4 个子块的直方图映射 f 采用双线性插值算法获取。图 4 为图像第 $j-1$ 行至第 j 行、第 $i-1$ 列至第 i 列的局部示意图 ($i, j \geq 2$),其中 P 为待增强像素点, x, y, w, r 数值关系确定 P 点和参考点的位置关系,根据其处于角落、内部、边界来确定各相邻子块的权重关系,加权计算得到最终的像素值,如式(2)所示:

$$P_{\text{new}} = \frac{r}{r+w} \left(\frac{y}{x+y} f_{i-1,j-1} P_{\text{old}} + \frac{x}{x+y} f_{i,j-1} P_{\text{old}} \right) + \frac{w}{r+w} \left(\frac{y}{x+y} f_{i-1,j} P_{\text{old}} + \frac{x}{x+y} f_{i,j} P_{\text{old}} \right), \quad (2)$$

式中: P_{old} 为原始像素点的灰度值; P_{new} 为增强去雾后新灰度值。

图4 第 (i, j) 子块中像素 P 及其相邻的 4 个子块Figure 4 The four adjacent blocks of the sub-block (i, j)

3 改进 CLAHE 算法的 FPGA 实现

3.1 FPGA 实现框架

图 5 为改进 CLAHE 算法的 FPGA 实现流程图。整个系统划分成 6 个模块,分别为输入模块、直方图统计模块、直方图均衡模块、灰度值映射模块、双线性插值模块以及输出模块。其中,直方图统计模块采用分块分通道统计,将多通道直方图合并于一个直方图中,用于后面的

削峰填谷;针对直方图均衡模块,原 CLAHE 算法以 1 作为分配步长,该过程需要多次的迭代,像素重分配操作需要进行多次迭代,使得图像处理延时较大。而笔者将该过程利用映射算法进行计算,将多出的像素灰度值个数均匀的分配给 256 个灰度级,且仅需要进行一次削峰填谷操作,提高了算法的效率。各个模块均采用标准 axi_stream 接口进行互联。

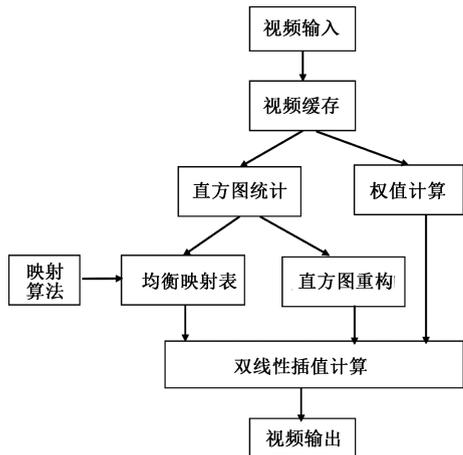


图 5 FPGA 实现流程图

Figure 5 The FPGA program block diagram

3.2 各模块原理以及实现

输入模块:输入模块通过 SDI 接口采集相机输出的高速图像数据,将图像数据从 YUV 颜色空间转换为 RGB 颜色空间,并将数据流转换为标准的 axi_stream 接口,方便与后续模块互联。

直方图统计模块:本系统将视频图像划分为多个非重叠子块,对每个子块分 RGB 通道进行直方图统计,并将多通道直方图进行合并,合并后的直方图数据存入乒乓 RAM 中。如图 6 为子块 1 的直方图统计,分 RGB 通道统计各分块的灰度直方图信息,采用边输入边计算的方法,将每个通道灰度值对于像素数存入对应的 RAM1_R、RAM1_G、RAM1_B 存储器中。当该子块所有像素输入完成之后,读出该子块每个通道像素值并将其进行叠加统计,最后将整合的直方图写入乒乓 RAM1_RGB 中。截断阈值 β 可以由上位机通过 RS-422 设置。当某个灰度值的个数 M' 大于或者等于 β 时,由 β 代替该像素的个数 M' 存入到 RAM_RGB 中,并将超过门限值的像素点个数 $M' - \beta$ 累计到对应子块的寄存器 REG_cnt 中,用于后续的直方图均衡操作。

直方图均衡模块:直方图统计完成后,通过削峰填谷对其进行均衡操作,如图 7 所示。 β 为输入阈值,由于实际环境的多变性,单一阈值无法对

所有情况都达到良好的去雾效果,因此本系统提供了多个可供选择的 β 值。

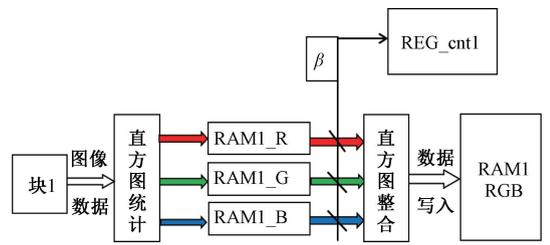


图 6 子块 1 的直方图统计

Figure 6 The histogram statistics

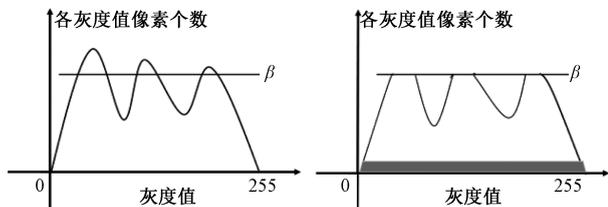


图 7 直方图均衡

Figure 7 The histogram equalization

均衡算法步骤如下:

步骤 1 初始化参数。计算分配步长 $L = REG_cnt/256$, 设置截断阈值 β , 初始化当前灰度级 $i = 0$ 的像素个数 M' , 均衡后的灰度级个数 $K = 0$ 以及灰度级范围 $i \in [0, 255]$ 。

步骤 2 计算均衡后的灰度级像素个数 $K = L + M'[i]$ 。

步骤 3 若 $K > \beta$, 则当前灰度级像素个数 $M'[i]$ 为 β ; 否则灰度级像素个数 $M'[i] = M'[i] + L$ 。

步骤 4 更新灰度级 $i = i + 1$ 。

步骤 5 停止准则, 判断 i 是否达到最大灰度级 255。若满足则输出均衡后各灰度级的像素数; 否则转步骤 2。

原始 CLAHE 算法像素重分配操作需要进行多次迭代, 使得图像处理延时较大, 将变量 REG_cnt 与灰度级的个数作商, 将多出的像素灰度值个数一次性均匀地分给各个未饱和灰度级, 提高了实时性的同时节约了系统资源。

灰度值映射模块:利用映射算法, 通过原始图像灰度值映射出增强后的灰度值。

映射算法步骤如下:

步骤 1 初始化每个子块的总像素数 N 、灰度级 $i = 0$ 以及其对应的原始灰度级像素值 $Y_{i=0}$ 。

步骤 2 计算中间过度值 $Y'_i = \sum_{i=0}^i Y_i$ 。

步骤 3 计算增强后的灰度级像素值 $Y_{i_NEW} =$

$256 \times Y'_i/N$ 。

步骤 4 更新灰度级 $i = i + 1$ 。

步骤 5 判断停止条件,判断 i 是否达到最大灰度级 255。若满足则输出增强后各灰度级的数值;否则转步骤 2。

将均衡后的灰度值直方图数据采用乒乓的方式按奇偶帧存入 RAM 中。

双线性插值模块:图 8 为双线性插值模块功能框图。双线性插值模块包括数据选取、权值计算以及插值计算。从 RAM 中读取像素数据时,同时计算权值因子,需对像素时钟做一定的延时来保证两者的同步性。位于角落、内部、边界位置的像素对应不同的插值计算。最后,利用视频图像相邻两帧直方图的高度相似性,以前一帧直方图均衡后的映射灰度值作为当前帧图像处理的数据,可避免缓存一帧的数据,避免了大容量外部存储器的使用,节约了资源,提高了实时性。

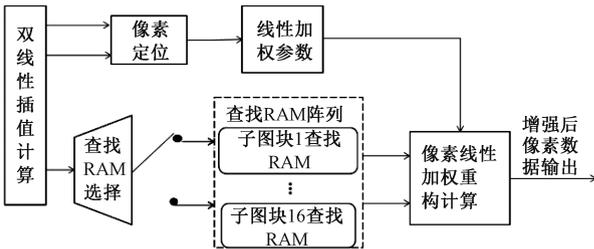


图 8 插值计算模块

Figure 8 The interpolation calculation

像素点在图像中的位置由像素时钟和行同步信号(LVAL)计数获得,即确定其所属子块中的位置。

输出模块:输出数据通过芯片 GV7600/DS90CR288 解析,通过 SDI/Cameralink 接口传送给显示端。

4 系统测试结果

为验证本文算法的有效性,将本文算法与原始 CLAHE 算法进行对比。图 9 为两种去雾算法效果图。仅对原始 CLAHE 算法中 Y 方向分量做均衡使得图片的整体亮度有一定的增强,但其色彩暗淡,对比度较低;而本文算法处理效果明显,图片色彩均匀,图片的层次感较强。

采用 Xilinx Vivado 软件对 FPGA 程序进行布局布线,FPGA 资源消耗如表 1 所示。其中,共消耗了 53 064 个查找表(LUT),占总查找表的 26.04%;消耗 103 个块存储单元(BRAM),占总 BRAM 的 23.15%;消耗了 60 个 DSP 资源,占总 DSP 的 7.14%。



(a) 原始图像



(b) CLAHE Y分量均衡算法



(c) 本文算法

图 9 两种算法去雾效果对比

Figure 9 The fogging effect of the two algorithms

表 1 FPGA 资源消耗

Table 1 The resource consumption of FPGA

项目	资源占用 资源/个	可用资 源/个	所占百分 比/%
LUT	53 064	203 800	26.04
LUTRAM	3 924	64 000	6.15
FF	50 077	407 600	12.29
BRAM	103	445	23.15
DSP	60	840	7.14
IO	138	400	34.50
BUFG	12	32	37.50
MCMC	3	10	30.00
PLL	1	10	10.00

图 10 为直方图统计模块 MODELSIM 仿真的结果图。将像素灰度值按一定的时序写进 RAM 中,图中:rgb_vld 为数据有效标志(高有效);rgb_data 为灰度值(如 0×71 、 0×72 等);ram_wr_cnt 为灰度值个数(如 $0 \times 4 100$ (16 进制))。

图 11 为双线性插值模块的 MODELSIM 仿真结果图。按一定的时序从 RAM 中读取直方图均衡后的灰度值,通过读取原始像素值查找新映射列表来索引增强后的像素值。以读取 B 通道数据为例,其中:ram1_en_pi_b 为读 B 值使能(高有

效),ram1_addr_pi_b 为 B 的数据地址(如 0x85、0x84),ram1_din_pi_b 为 B 的数据值(如 0x92),pipe1_row_cnt 与 pipe1_line_cnt 为子块 1 的行列计数(如 0x20、0x3f),以确定该值所在子块 1 中位置。

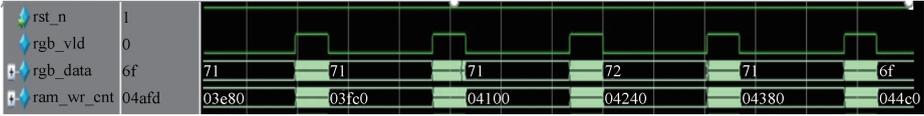


图 10 直方图统计仿真结果

Figure 10 The simulation of histogram statistics

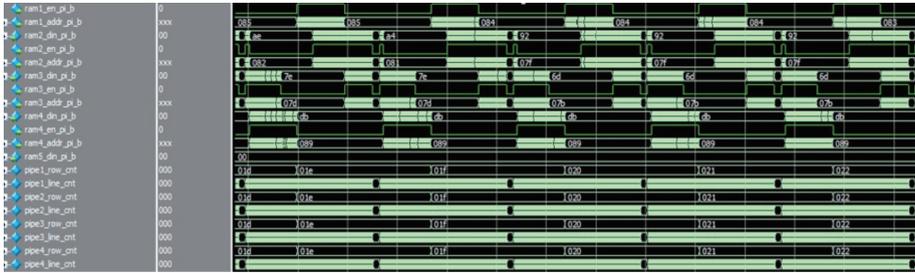


图 11 双线性插值仿真图

Figure 11 The simulation of bilinear interpolation

图 12 为本文算法 MATLAB 实现和 FPGA 实现效果图。FPGA 与 MATLAB 实现结果图对应位置像素点的 RGB 值一致,说明了 FPGA 实现的正确性。

图 13 为系统外场测试结果。可以看出,原始有雾图像图片不清晰、对比度较低,很难识别图中的目标(图中房屋轮廓);而去雾图像对比度增强、突出细节、层次感增强。结果表明,该系统具有较强的实时性和稳定性,系统功耗测试仅为 4.848 W,可应用于低功耗领域。



(a) 原始图像



(b) MATLAB实现



(c) FPGA实现



(a) 测试原始图像



(b) 去雾效果图

图 13 系统去雾效果图

Figure 13 The fogging effect of this system

图 12 MATLAB 和 FPGA 实现效果图
 Figure 12 The fogging effect of this algorithm in MATLAB and FPGA

5 结论

以 FPGA 为硬件平台,设计并构建了一种基于 1080P 高分辨率视频图像实时增强去雾系统,对

CLAHE 算法进行一定的改进,完成了改进后的 CLAHE 算法各个模块的设计和硬件实现。系统测试结果表明,系统具有极好的稳定性和实时性,图像的对比度明显增强,图像内容清晰,层次感强,能够达到很好的去雾效果,同时还具有低功耗特性。

参考文献:

- [1] 王欣威,李颖,董慧颖,等.基于大气模型的天气退化图像复原方法及应用[J].沈阳理工大学学报,2005,24(1):32-35.
- [2] 郭璠,蔡自兴,谢斌,等.图像去雾技术研究综述与展望[J].计算机应用,2010,30(9):2417-2421.
- [3] 睢丹,张亚利,杨杰.雾霾污染下基于混沌性的图像去雾算法改进[J].郑州大学学报(工学版),2016,37(4):91-96.
- [4] 郭伟.基于多种先验的单幅雾天图像复原算法研究[D].厦门:厦门大学,2014:58-61.
- [5] HE K M, SUN J, TANG X O. Single image haze removal using dark channel prior[J]. IEEE transactions on pattern analysis & machine intelligence, 2011, 33(12):2341-2353.
- [6] HE K M, SUN J, TANG X O. Guided image filtering[J]. IEEE transactions on pattern analysis & machine intelligence, 2013, 35(6):1397-1409.
- [7] 魏先民.自适应直方图均衡方法研究与应用[J].信息技术与信息化,2005(4):124-125.
- [8] ZUIDERVELD K. Contrast limited adaptive histogram equalization[M]. San Diego, CA, USA: Academic Press Professional Inc. 1994:474-485.
- [9] 杨光,李晶,吴钟建,等.基于FPGA的实时视频增强[J].电视技术,2013,37(19):19-23.
- [10] 周卫星,廖欢.基于高频强调滤波和CLAHE的雾天图像增强算法[J].电视技术,2010,34(7):38-40.
- [11] KENTARO K, MARUYAMA T. Real-time processing of contrast limited adaptive histogram equalization on FPGA[C]//International Conference on Field Programmable Logic & Applications. Milano, Italy: IEEE, 2010:155-158.
- [12] FERGUSON P D, ARSLAN T, ERDOGAN A T, et al. Evaluation of contrast limited adaptive histogram equalization (CLAHE) enhancement on a FPGA[C]//2008 IEEE International SOC Conference. Newport Beach, CA, USA: IEEE, 2008:119-122.
- [13] 孔壮,何衡湘,代俊,等.高分辨率视频图像实时增强系统的设计[J].电视技术,2015,39(20):115-118.
- [14] 周晓波,何魁华,周聪.基于FPGA的图像高速去雾实时系统设计实现[J].电视技术,2018,42(4):67-72.

High Resolution Video Image Real-time Enhancement System Based on FPGA

LIU Huajun¹, ZHANG Ruijue², LIU Jianfeng², WANG Sheng², XIA Qiaoqiao¹

(1.School of Physical Science and Technology, Central China Normal University, Wuhan 430079, China; 2.Shanghai Institute of Space Electronics Technology, Shanghai 201109, China)

Abstract: Most defogging systems mainly focused on single image, lacking in stability and real-time performance. Besides, the traditional image haze removal algorithm had high complexity and the histogram of bright area was overexposed, it was difficult to be applied to real-time video image restoration. Facing this problems, a 1080P full-hd video image real-time enhanced defogging system based on FPGA was designed and constructed. At first, the histogram statistics was conducted in RGB color space by using blocks and dividing the channels. And then, the combined histogram of multi-channel was clips at a predetermined value, thereby limiting the contrast amplification. Finally, the bilinear interpolation is performed between the new histogram. We simulated the improved CLAHE algorithm, and tested the system for foggy video, image and outdoor scenes. Experimental results showed that the system can real-time process 30 frames per second 1 920×1 080 resolution images, compared with the mainstream methods, the defogging effect was better and more stable. The system power was less than 5 watts, so it could be applied to low power consumption.

Key words: image dehazing; video image; FPGA; real-time; low-power dissipation