

文章编号:1007-2780(2014)04-0629-08

彩色图像的 FPGA 实时增强系统实现

赵 建^{1*}, 赵 凡^{1,2}, 曲 锋¹

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院大学, 北京 100049)

摘要:为了对彩色图像进行实时增强, 本文提出了采用基于插值的分段拉伸算法。首先将彩色 RGB 图像转换为 HSV 空间, 在该空间, 对图像直方图进行基于差值的分段拉伸处理, 以达到对图像增强的目的。经过该方法增强后的图像, 细节信息更加丰富, 图像的清晰度得到了改善, 图像的视觉质量也得到了明显提高。经过该算法增强后图像的灰度平均梯度值为直方图均衡化算法的 1.95 倍。应用现场可编程门阵列(FPGA)为中央处理器, 通过并行处理结构及流水线技术, 完成图像空间的转换和图像的实时增强算法, 简化了系统设计, 使处理系统硬件更加紧凑, 运行更加可靠。给出了系统主要功能模块的实现方法, 经现场调试, 可完成每秒 30 帧 $\times 1\,024 \times 1\,024 \times 24$ bit 数据的处理, 与直方图均衡化等传统图像增强算法相比, 该算法计算时间缩短了 0.807 ms。该系统具有集成度高、图像处理速度快和实时性强等特点。

关 键 词:彩色图像增强; 分段拉伸; FPGA

中图分类号:TP391 **文献标识码:**A **doi:**10.3788/YJYXS20142904.0629

Implement of system for color image enhancement in real-time based on FPGA

ZHAO Jian^{1*}, ZHAO Fan^{1,2}, QU Feng¹

(1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;*

2. *University of Chinese Academy of Sciences, Beijing 100049, China*)

Abstract: In order to enhance the color images in real time, this article proposed an algorithm of piece-wise stretching based on interpolation. Firstly, the RGB color image is converted into HSV space. In this space, the histogram of the image is stretched based on interpolation. By using this method to enhance the image, the details of the image are more rich, the image clarity is improved obviously and the visual quality of the images has also been significantly improved. Gray average gradient value of enhanced image after this algorithm is 1.95 times than histogram equalization. In this paper, field programmable gate array (FPGA) is used as central processor. By using of parallel processing and pipelining technology, we completed the conversion of the image space and the algorithms of real-time image enhancement, simplified system design, and make the processing system more compact, the operation more reliable. The paper also gives the system functional modules. After debugging, this

收稿日期:2013-08-13; **修订日期:**2013-12-04.

基金项目:国家自然科学基金(No. 61137001)

* 通信联系人, E-mail: zhaojian6789@126.com

system could complete the processing of data $30 \times 1\,024 \times 1\,024 \times 24$ bit per second. Compared with the traditional image enhancement algorithms thus as the histogram equalization, this algorithm reduced the time about 0.807 ms. The system has the characteristics of high integration, fast image processing speed and strong real-time.

Key words: color images enhancement; piecewise linear stretching; FPGA

1 引言

图像增强^[1-2]主要是将图像转换为更适合人眼观察或者机器分析、识别的形式,改善图像的质量和视觉效果,以便获得更清楚、明晰的富含大量有用信息的可使用图像。图像增强技术在军事、遥感生物、公共安全、医学工程等方面发挥着重要作用。图像增强技术在对于灰度图像的处理得到了普遍的应用,但是随着近年来彩色图像的广泛运用,对彩色图像的增强技术^[3]有了明显需求。由于彩色图像的 R、G、B 分量相互不独立,对于彩色图像直接增强,会造成图像增强后颜色失真。而 HSV 空间更能反映人的视觉特性,将彩色图像转换成 HSV 空间,在该空间中对图像进行增强算法的处理,处理后图像效果明显优于直接对彩色图像增强。对于灰度图像的增强,常用的方法有直方图均衡化,但是运用该方法对于航拍图像进行增强,容易造成图像的纹理不清晰,细节信息也不能够最大化的显现。对图像进行分段拉伸也是一种常见算法,但是分段拉伸算法的关键点在于分段点的选择,分段点的选择稍有不慎便影响了整幅图像的增强效果。而本文中基于插值的分段线性拉伸算法,可以根据图像特点进行分段拉伸,对于原始图像的每一个像素点都可以进行变换,改变灰度的动态范围^[4-5],同时该插值点的选取,是根据直方图统计结果所选取,充分考虑到图像的特点。

在图像增强的硬件实现上,最初是用 DSP 实现,但是 DSP 在对数据接收和处理的过程中存在着延时,很难达到实时性要求。后来发展为 DSP+FPGA 这种架构,这种系统运算速度比较高,具有较强的灵活性和通用性,但是成本也较高,同时 DSP 和 FPGA 之间通信也比较复杂^[6]。FPGA 在结构上的并行处理优势使得其更适合完成高速的运算能力,同时系统结构性强,具有很大

的灵活性。基于以上背景,该系统采用采用 cyclone3 系列的 EP3C16 FPGA 为处理器,将彩色图像转换为 HSV 空间,完成图像的增强后,再将 HSV 转换成为 RGB 彩色空间,具有实现简单、集成度高、功耗低的特点^[7-8]。

2 彩色图像增强原理及算法

早期对于彩色图像的增强是将灰度域图像增强技术直接应用到彩色图像 R、G、B 三个分量上,但是这样的做法会造成计算量比较大,并且由于 R、G、B 三个分量是相关的,也造成了彩色图像的颜色失真。将彩色图像转换成 HSV 空间更能反映人的视觉特性,这也使得 HSV 空间在彩色图像的处理中得到广泛应用。在 HSV 空间采用基于插值的分段拉伸算法对每个像素点进行拉伸变换,最后将经过拉伸过的灰度图像转换为 RGB 图像。

2.1 RGB 和 HSV 空间的相互转换

在彩色成像设备和彩色显示设备中,RGB 色空间是一种简单常用的色空间,它能使颜色从一个平台映射到另外一个平台而不会严重丢失颜色信息。RGB 是一种相加成像系统,由红、绿、蓝三种基色叠加,以产生各种需要的颜色。

HSV 彩色空间是一种直观的颜色模型,H、S、V 分别代表色调、饱和度和亮度。这个颜色空间的模型对应于圆柱坐标系中的一个圆锥形子集,圆锥的顶面对应于 $V=1$ 。色彩 H 由绕 V 轴的旋转角给定,范围为 $0^\circ \sim 360^\circ$ 。在 HSV 颜色模型中,饱和度 S 取值 $0 \sim 1$,所以圆锥顶面的半径为 1。

RGB 空间向 HSV 空间的转换公式如下:

$$V = \max\{R, G, B\}, \quad (1)$$

$$S = \frac{\max\{R, G, B\} - \min\{R, G, B\}}{\max\{R, G, B\}}, \quad (2)$$

$$H = \begin{cases} \frac{1}{6} \frac{G-B}{\max\{R,G,B\}-\min\{R,G,B\}} & R = \max\{R,G,B\} \\ \frac{1}{6} \left(2 + \frac{B-R}{\max\{R,G,B\}-\min\{R,G,B\}} \right) & G = \max\{R,G,B\} \\ \frac{1}{6} \left(4 + \frac{R-G}{\max\{R,G,B\}-\min\{R,G,B\}} \right) & B = \max\{R,G,B\} \end{cases} \quad (3)$$

HSV 空间向 RGB 空间的转换公式如下:

$$\begin{cases} [R,G,B] = [V,K,M] & 0 \leq H < 1/6 \\ [R,G,B] = [V,N,M] & 1/6 \leq H < 2/6 \\ [R,G,B] = [M,V,K] & 2/6 \leq H < 3/6 \\ [R,G,B] = [M,N,V] & 3/6 \leq H < 4/6 \\ [R,G,B] = [K,M,V] & 4/6 \leq H < 5/6 \\ [R,G,B] = [V,M,N] & 5/6 \leq H < 1 \end{cases} \quad (4)$$

其中: $M = V(1-S), N = V(1-SF), K = V[1-S(1-F)], F = 6H-1$ 。

2.2 基于插值的线性拉伸算法

该方法是基于由插值函数定义的对点的转换,该函数在定义区间是线性的。分段线性拉伸是将图像灰度区间分成两段乃至多段,对图像中各灰度级进行均匀压缩或扩展,达到效果更好的灰度范围。本质上就是压缩出现概率较小的灰度级,将出现概率较大的灰度级向邻近扩展。线性拉伸算法较为柔和,拉伸范围合适的情况下,能有效地扩大对比度,明显提高视觉效果,同时也保留了图像的大量细节^[9]。

$$f(v) = \sum_{i=1}^n a_i |v - v_i|. \quad (5)$$

该函数为用于灰度级变换时的线性差值函数。 v_i 为插值点, f_i 为亮度 V 经过变换后的值。 a_i 为分段拉伸函数的拉伸系数。

首先对最优窗口的选择,将亮度小于最低阈值 $V_{\min}$ 和大于最大阈值 $V_{\max}$ 的灰度级进行压缩,增大窗口的动态范围。阈值确定后,插值点的确定过程如下:在灰度级图像中的某一范围 D 属于 R^2 内, $D \rightarrow [0, 255]$, 统计出该区域内所有像素的个数 $\text{card}(D_i)$ 。在区域 D 中,将每个像素的亮度与该亮度像素的个数相乘得到 $l(x, y)$ 。在离散区域 D 中 $l(x, y)$ 求和,根据公式(10)求得插值点 v_i ,即该区域中能量分布较密集点。插值点的确定,主要是根据对灰度图像的灰度值统计得出。

$$v_i = \frac{\sum_{(x,y) \in D_i} l(x,y)}{\text{card}(D_i)}. \quad (6)$$

根据公式(5),在 $(v_i) i=1, n$ 所对应的 $(f_i) i=1, n$ 已知的情况下,可以求得系数 a_i ,公式如下:

$$\begin{cases} a_1 = \frac{1}{2} \left(\frac{f_n + f_1}{v_n - v_1} + \frac{f_2 - f_1}{v_2 - v_1} \right) \\ a_i = \frac{1}{2} \left(\frac{f_{i+1} + f_i}{v_{i+1} - v_i} - \frac{f_i - f_{i-1}}{v_i - v_{i-1}} \right) \\ a_n = \frac{1}{2} \left(\frac{f_n + f_1}{v_n - v_1} + \frac{f_n - f_{n-1}}{v_n - v_{n-1}} \right) \end{cases} \quad (7)$$

通过对计算量和图像增强效果的综合考虑,在阈值 $V_{\min}$ 和 $V_{\max}$ 之间取两个插值点 v_2, v_3 , 根据公式(2~6),如图 1 所示,在区域 $(V_{\min}, 2(V_{\max} - V_{\min})/3)$ 中确定插值点 v_2 的值, v_3 的值确定与区域 $(1(V_{\max} - V_{\min})/3, V_{\max})$ 中。至于变换后的最小阈值 $V_{\min}$ 所对应的 f_1 ,本算法中将 f_1 拉伸到 $V_{\min}/4$ 处,最大阈值 $V_{\max}$ 对应变换后的 f_n ,拉伸到 $V_{\max}$ 后的四分之三 $255 - V_{\max}$ 处,其结果扩展了像素取值的动态范围。根据最小阈值 $V_{\min}$ 所对应的 f_1 ,最大阈值 $V_{\max}$ 对应的变换后的 f_n , f_1 取值 $(f_n - f_1)/3$, f_2 取值 $2(f_n - f_1)/3$ 。

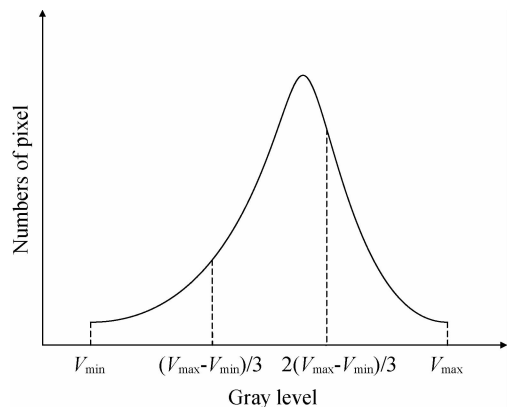


图 1 灰度级统计

Fig. 1 Gray level statistics

根据线性拉伸公式,结合本算法中所选用的插值点,公式(6)变换为:

$$f(v) = a_1 |v - v_{\min}| + a_2 |v - v_2| + a_3 |v - v_3| + a_4 |v - v_{\max}|. \tag{8}$$

3 硬件实现

系统的硬件框图及工作流程如图 2 所示,在相机控制信号的控制下,在帧有效,行有效和数据有效的前提下,一个像素时钟 FPGA 接收一个有效数据,经由空间转换模块后,只对 V 进行采集。将 V 作为 RAM 地址,不同地址中的数值表示该亮度的像素数,进行直方图统计。由于图像具有实时性的特点,在连续两帧图像变化不大的假设下,根据直方图统计结果采用插值的线性拉伸算法对原始图像数据进行处理。最后将 HSV 空间转换为 RGB 空间,在 VGA 控制模块的控制下,将增强的彩色图像显示在 VGA 上。

相机控制模块: FPGA 内部模块产生相机控制信号,如像素时钟信号、图像信号源数据、同步信号,由于 Cameralink 采用 LVDS 信号传输数据, FPGA 发出的 CMOS/TTL 控制信号经由 Cameralink 接口器件转换成 LVDS 信号,接收端配套器件进行解调,将 4 路 LVDS 数据流转换为 24 位图像数据和 4 位同步信号。同时将 LVDS 信号转换成为 FPGA 支持的 I/O 接口标准信号 LVCOMS/LVTTL 时钟输出。将 28 位数据信号和一位时钟信号传送给 FPGA 完成图像采集^[6,10]。同步信号主要有同步时钟信号,场同步信号(FVAL),行同步信号(LVAL),数据同步信号(DVAL)。在 FVAL、LVAL 和 DVAL 有效的状态下,每个同步时钟采集一个有效数据。

直方图统计模块: 系统采用双口 RAM 作为储存器来储存直方图统计结果。双口 RAM 的地址宽度设置为 8 bit。在第一个时钟根据 V_{datain} 的值,读出 RAM1 中相应地址的值 rdata,在第二个时钟把接收来的灰度值作为直方图统计的地址,将读出的值 rdata 加 1,写入 RAM1, RAM2 的该地址中。此后反复这个过程,完成一帧图像的直方图统计。系统中多采用一块 RAM2 的原因是只对直方图双端口 RAM 进行单端口写操作,另一个端口留给最优窗口确定进行

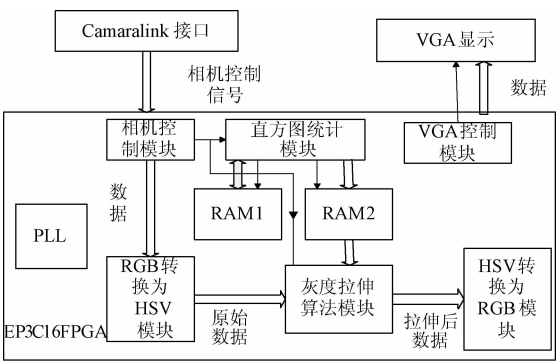


图 2 系统结构框图
Fig.2 System block diagram

异步读操作作用,这样做避免了对直方图双端口 RAM 的读端口复用。在直方图统计结束后,在场消隐区,对双口 RAM 进行清零,为下一帧图像的统计做准备。由于实时图像帧频为 30 F/s,因此可采用上一帧的直方图统计结果来完成下一帧图像的处理。

控制时钟 PLL:在一个像素时钟要完成双口 RAM 的读,并将读出结果加 1 后写回 RAM,所以将直方图统计的控制时钟设为像素时钟的 2 倍,即 PLL 二倍频。

灰度拉伸模块:步骤一:首先根据原始直方图统计图 k_i ,进行最优窗口的确定。 k_i 表示灰度级为 v_i 的像素个数。按照低地址 araddr 依次从双口 RAM2 中读出每个像素的个数 k ,进行累加。 $sum = \sum_{i=0} k_i, i \in [0, 255]$,当 sum 大于等于总像素个数的 98%时,取此时的 i 作为最大阈值 $V_{\max}$ 。同理,按照高地址 araddr,读出每个地址中像素数 k ,累加 $sum = \sum_{j=0} k_j, j \in [0, 255]$,当 sum 大于等于总像素个数的 98%时,取此时的 j 作为亮度的最小阈值 $V_{\min}$ 。步骤二:先对最小阈值 $V_{\min}$ 进行拉伸,拉伸后对应最小阈值为 f_1 ,拉伸后的最大阈值 $V_{\max}$ 拉伸后为 f_n 。在 f_2, f_3 ,确定后根据公式(7),求 a_1, a_2, a_3, a_4 。完成了在消隐区,系数和插值点的获得。在下一帧图像到来时,在每个像素时钟内,求得每个亮度 V_{datain} 在进行灰度拉伸后的 V_{change} ,进而完成每个像素点的变换。

4 实验结果及分析

设计中使用 Quartus II 中 SignalTap II 逻辑

分析仪对图像处理前的数据和处理后数据进行了观察,见图 3。 V_{datain} 是 HSV 空间未经变换的亮度值, V_{change} 为经过灰度拉伸后的亮度值。在消隐区即 FVAL 为低时,根据双口 RAM 统计结果,求得两个插值点分别为 96 和 123。在帧有效信号为高时,在每个像素时钟内,每输入一个像素,

相应输出经过变换的值。

该设计对一副大小为 $1\,024\times1\,024\times24\text{ bit}$ 大小的图像进行了增强处理。图 4 为航拍的原始彩色图像,在搭建的系统上用本算法对彩色图像处理后的图像如图 5 所示。航拍图像增强前后所对应的直方图如图 6、7 所示。

Name	-7	-6	-5	-4	-3	-2	-1	0	1	2
algorithm:inst2 FVAL										
algorithm:inst2 V_datain	B0h	B3h	B2h	B5h	B4h	B7h	B6h	B1h	B8h	BBh
algorithm:inst2 V2	96									
algorithm:inst2 V3	103									
algorithm:inst2 V_change	D2h	D8h	D9h	D8h	DAh	D0h	DFh	DCh		

图 3 灰度变换仿真结果

Fig. 3 Simulation result of gray level transformation



图 4 原始图像图

Fig. 4 Original image

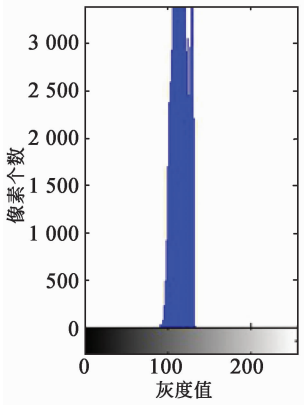


图 6 原始 RGB 图像直方图

Fig. 6 Histogram of the original RGB image



图 5 增强后图像

Fig. 5 Enhanced image

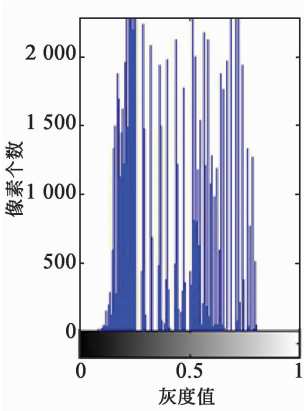


图 7 增强后 RGB 图像直方图

Fig. 7 Histogram of the enhanced RGB image

观察原始图像的直方图,可以发现,灰度值比较集中,反应到图像中就是细节不明显,对比度低。同时原始图像主要的轮廓不明显。本文方法处理后的图像,细节信息丰富,边缘清晰,视觉效果较好。直方图比较均匀,灰度动态范围宽。该算法可以很大程度上改善图像的清晰度。

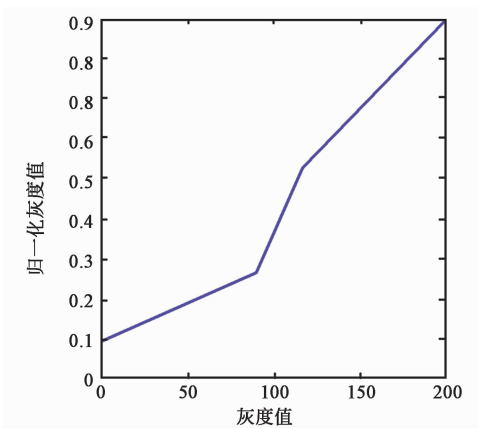


图 8 当 $n=4$ 时的插值函数图像

Fig. 8 Graphic of interpolation function for $n=4$

图 8 中为各个灰度级经过拉伸后的图像, x 轴为灰度级 $v_i(0,255)$, y 轴为经过变换后的 f_i , 分段拉伸效果的好坏,很大程度上取决于分段区间。本文中 $V_{\min}$ 取值 0, $V_{\max}$ 取 200, 两个插值点根据公式(6)求得分别为 96 和 123。

参考文献 [11] 的直方图均衡算法以及文献 [12] 的平台直方图均衡化算法, 图 9 给出了采用 3 种不同算法增强后的图像及其对应的直方图。

通过直方图均衡化后的图像^[11], 灰度值较小区域更暗, 不能显现出阴暗区域的细节信息, 且灰度值较大区域更亮, 失去原有纹理信息。平台直方图均衡化^[12]后的图像较直方图均衡化图像相比, 图像效果有了明显改进。但是该算法增强后灰度值较小处的像素值比较多, 也就造成了图像的整体效果偏暗, 不能明显显现图像细节信息。而本文增强算法将图像的亮度调节得非常均匀, 可以较为明显看出整幅图像的细节信息, 视觉效果很好。

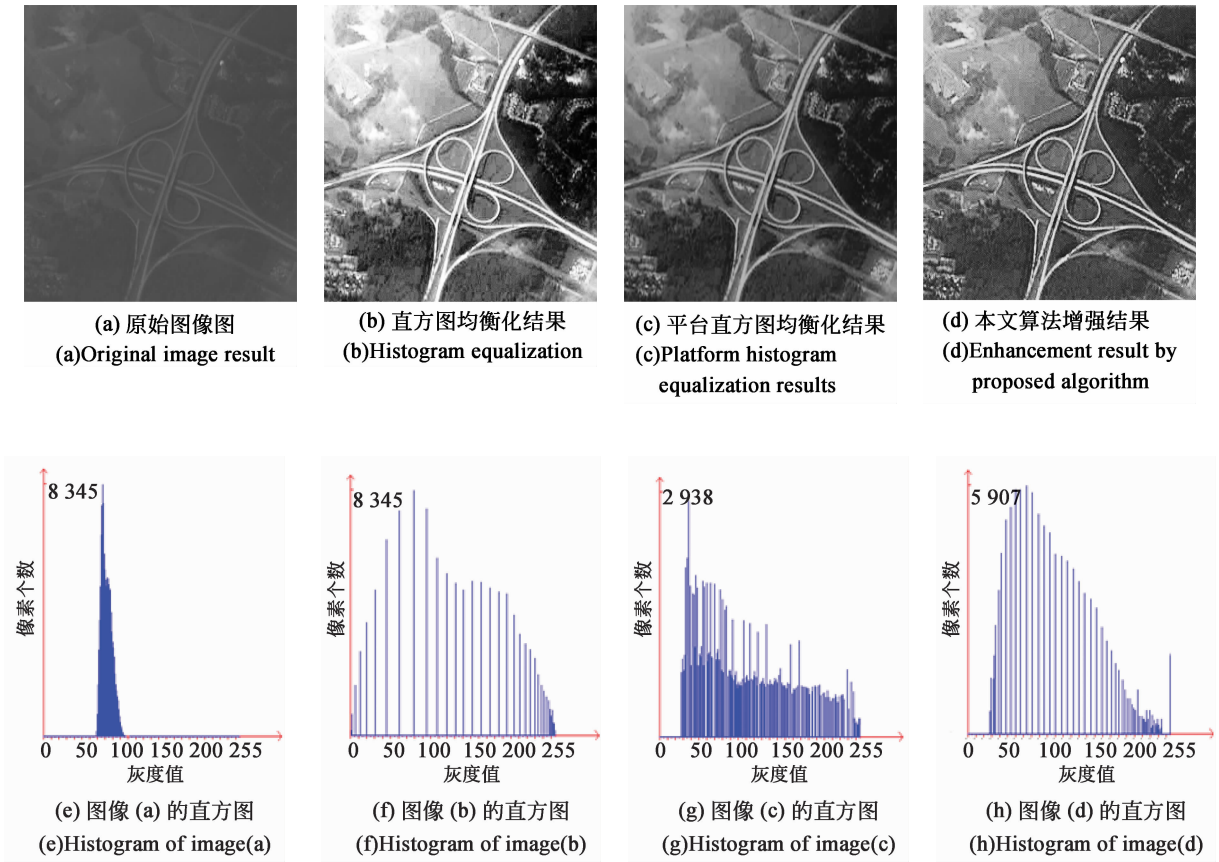


图 9 三种算法性能比较

Fig. 9 Performance comparison of three algorithms

针对航拍模糊图像,本文采用灰度平均梯度值来判断增强算法的效果和有效性。灰度平均梯度值是分别将图像长度和宽度方向上的相邻像素

灰度值做差后求平方和再求均方根值,它能较好地反映图像的对比度和纹理变化特征,其值越大表示图像越清晰,图像质量越好。公式为:

$$\text{GMG} = \frac{1}{(m-1)(N-1)} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\frac{\Delta I_x^2 + \Delta y^2}{2}} =$$
$$\frac{1}{(m-1)(N-1)} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} \sqrt{\frac{[g(i,j+1) - g(i,j)]^2 + [g(i+1,j) - g(i,j)]^2}{2}}. \tag{9}$$

表 1 给出了 3 种算法的性能评价,可以看出,基于插值的线性拉伸算法与直方图均衡化和平台直方图均衡化算法相比,处理速度更快,更能够满足实时性需求。增强后的图像更清晰,轮廓以及细节信息更明显。

的线性拉伸算法,根据插值函数对每个像素点进行灰度变换。其在增强图像对比度的同时也会很好地保留图像细节,经过该算法增强后图像的灰度平均梯度值为直方图均衡化算法的 1.95 倍。本文方法处理后的图像,细节丰富,灰度动态范围宽,该算法很大程度上改善图像了的清晰度和视觉质量。对于实时的彩色图像增强处理具有数据量大、传输速度快的特点,用 FPGA 作为整个算法的中央处理器,通过相应的图像处理模块,完成彩色图像空间和灰度空间的相互转换以及灰度图像的实时增强算法,并利用软件实现直方图统计和拉伸算法的并行执行,与直方图变换等传统图像增强算法相比,该算法计算时间缩短了 0.807 ms。该系统具有集成度高、图像处理速度快和实时性强等特点。结果表明,该算法的实现具有实时性,能够达到预期目标,基本可以实现工程应用。

表 1 三种算法性能比较

Tab. 1 Performance evaluation of three algorithms

增强方法	灰度平均梯度值	实时性/ms
直方图均衡化	8.617	3.326
平台直方图均衡化算法	15.222	4.232
基于插值的线性拉伸算法	29.650	2.519

5 结 论

对于图像的增强,本文提出了一种基于插值

参 考 文 献:

[1] 贾宏光,吴泽鹏. 基于广义线性运算和双边滤波的红外图像增强[J]. 光学精密工程,2013,12(21):3272-3282.
Jia H G, Wu Z P. Infrared image enhancement based on generalized linear operation and bilateral filter [J]. *Opt. Precision Eng.*, 2013,12(21):3272-3282. (in Chinese)

[2] 韩希珍. 结合偏微分方程增强图像纹理及对比度[J]. 光学精密工程,2012, 20(6): 1382-1388.
Han X Z. Enhancement of image texture and contrast combined with partial differential equation [J]. *Opt. Precision Eng.*, 2012,20(6) : 1382-1388. (in Chinese)

[3] 刘钦堂. 基于 Retinex 算法的彩色图像增强研究[D]. 哈尔滨:哈尔滨工程大学,2012.
Liu X T. Research of the color image enhancement based on Retinex algorithms [D]. Harbin: Harbin Engineering University,2012. (in Chinese)

[4] 武治国,王延杰. 一种基于直方图非线性变换的图像对比度增强方法[J]. 光子学报,2010,39(4):755-758.
Wu Z G, Wang Y J. An image enhancement algorithm based on histogram nonlinear transform [J]. *Acta Photonica Sinica*, 2010, 39(4) : 755 -758. (in Chinese)

[5] 庞玉林,刘珊珊. 基于硬件实现具有范围判断功能的动态线性拉伸算法[J]. 科学技术与工程,2010,10(16): 3878-3882.
Pang Y L,Liu S S. Dynamic linear extension algorithm based on hardware realization [J]. *Science Technology and Engineering*, 2010,10(16):3878-3882. (in Chinese)

[6] 何浩. 基于 DSP 和 FPGA 的图像处理平台的研究和实现[D]. 南京:南京航空航天大学,2011.
He H. Research and implementation of digital image processing card based on DSP and FPGA[D]. Nanjing :Nan-

jing University of Aeronautics and Astronautics, 2011. (in Chinese)

- [7] 于云翔, 张麟瑞. 实时图像增强算法改进及 FPGA 实现[J]. 计算机仿真, 2009, 26(2): 240-242.
Yu Y X, Zhang L R. An improved real-time enhancement algorithm and its realization with FPGA [J]. *Computer Simulation*, 2009, 26(2): 240-242. (in Chinese)
- [8] 刘春香, 郭永飞. 星上多通道遥感图像的实时合成压缩[J]. 光学 精密工程, 2013, 2(21): 445-453.
Liu C X, Guo Y F. Composition and compression of satellite multi-channel remote sensing images [J]. *Opt. Precision Eng.*, 2013, 2(21): 445-453. (in Chinese)
- [9] 牛英宇. 图像自适应分段线性拉伸算法的 FPGA 设计[J]. 现代电子技术, 2010, 10: 78-80.
Niu Y Y. Implementation of adaptive-piecewise linear stretching algorithm based on FPGA [J]. *Modern Electronics Technique*, 2010, 10: 78-80. (in Chinese)
- [10] 吴家伟, 武春风. 红外图像实时显示增强系统设计[J]. 光学 精密工程, 2009, 17(10): 2612-2619.
Wu J W, Wu C F. Design of real time infrared image enhancing system [J]. *Opt. Precision Eng.*, 2009, 17(10): 2612-2619. (in Chinese)
- [11] 胡元刚. 直方图均衡增强算法在实时图像处理中的实现[J]. 液晶与显示, 2009, 27(2): 907-909.
Hu Y G. Implementation of histogram equalization enhancement algorithms in real-time image processing [J]. *Chinese Journal of Liquid Crystals and Displays*, 2009, 27(2): 907-909. (in Chinese)
- [12] 王炳健, 刘上乾. 基于平台直方图的红外图像自适应增强算法[J]. 光子学报, 2005, 32(2): 299-301.
Wang B J, Liu S Q. Self-adaptive contrast enhancement algorithm for infrared images based on plateau histogram [J]. *Acta Photonica Sinica*, 2005, 32(2): 299-301. (in Chinese)

作者简介: 赵建(1967—), 女, 吉林长春人, 研究员, 博士生导师, 主要研究方向是数字图像处理。E-mail: zhaojian6789@126.com