·信号与信息处理·

基于 FPGA 实时红外图像自适应线性增强算法的实现

徐世伟[1],张　威[2],刘严严[1],高文清[1]

(1.光电系统信息控制技术国家级重点实验室,河北　三河　065201;2.东北电子技术研究所,辽宁　锦州　121000)

摘　要:提出了灰度直方图统计方法,利用 FPGA 完成一帧图像直方图统计并且采用自适应线性增强算法完成对下一帧图像数据的转换.针对实时处理的要求,应用现场可编程门阵列(FPGA)构造高速图像处理器,完成红外跟踪图像预处理的实时计算.经现场调试,代码简单,占用 FPGA 系统资源少,可以完成实时数据转换,结果证明是可行的.

关键词:直方图;线形变换;实时处理;图像增强

中图分类号:TN911.73;TN929.11　　**文献标识码**:A　　**文章编号**:1673－1255(2009)06－0066－04

Realization of Real-Time Infrared Image Accommodation Linear Enhancement Algorithm Based on FPGA

XU Shi-wei[1], ZHANG Wei[2], LIU Yan-yan[1], GAO Wen-qing[1]

(1. *National Laboratory of Electro-Optics System Technology*, *Sanhe* 065201, *China*;
2. *Northeast Research Institute of Electronics Technology*, *Jinzhou* 121000, *China*)

Abstract: A gray-scale histogram statistic method was proposed, a frame image histogram statistics was completed using FPGA and the next frame of image data conversion was made by the adaptive linear enhanced algorithm. According to the needs for real time processing, field programmable gate arrays (FPGA) was applied to construct the high-speed image preprocessor and to finish the real-time calculation for infrared tracking image preprocessing. The simulation and experimental results demonstrate that this method is feasible, and it has the advantages such as: the simple code, less system resources occupied by FPGA, and the capability of real-time data conversion.

Key words: histogram; linear transform; real-time processing; image enhancement

实时红外成像涉及的数据量非常大,因此要求图像处理系统具有高速强大的数据处理能力.可以采用多线程技术去实现图像数据的读取和显示,并及时响应用户操作,使用多线程技术可以提高软硬件资源的利用率,优化程序的动态性能,达到图像数据读取和显示的动态性和交互性效果,提高图像显示的连续性和稳定性[1].实现图像数据的读取和显示,并及时地响应用户操作.

1　直方图统计流程及硬件的实现

1.1　利用 FPGA 实现直方图统计流程

灰度级的直方图反映图像灰度级与出现这种灰度的概率之间的关系图形.一幅均匀量化的自然图像的灰度直方图通常在低灰度区域上频率较大,这样的图像较暗,区域中的细节常看不清楚.为使图像变清楚,一个自然的想法就是使图像的灰度动态范围变大,并且让频率小的灰度级经过变换后其频率变得大一些,即将变换后的图像灰度直方图在较大的动态范围内趋于均衡,这就是直方图修正技术.直方图修正技术是一种新的数字图像增强处理方法,

收稿日期:2009－09－15

作者简介:徐世伟(1979－),男,吉林农安人,硕士,工程师,主要研究方向为图像处理、目标探测与识别.

能够有效地改善灰度分布集中,动态范围狭窄,图像模糊等现象,提高图像的辨析程度. 直方图统计在图像处理中经常用到,但需要统计的数据太多,耗费了实时系统里的宝贵时间,这里讨论一种利用 FPGA 完成实时统计直方图的实现方法.

FPGA 首先将输入的每一个有效像素接收进来,并根据像素的灰度进行归类.相同灰度的像素放在一起,最后得出每一个灰度级上的所有像素数的数量,形成以灰度为横坐标,以像素数量为纵坐标的灰度分布图,也就是图像直方图,其统计流程如图 1 所示.

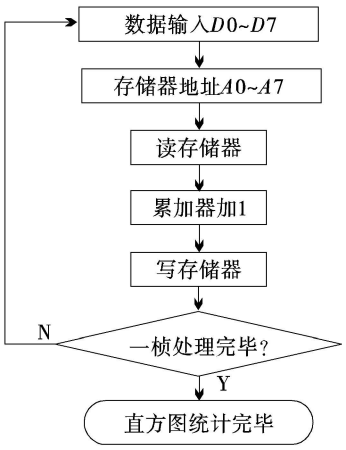


图 1　利用 FPGA 实现直方图统计流程

1.2　利用 FPGA 直方图统计的硬件实现

一个红外图像大小为 640×512 的系统,考虑到极端情况下,即所有像素都分布在一个灰度级上,则每个存储单元的宽度应为 19 bit,像素的灰度级用 8 bit表示.则存放直方图的存储器大小应该为256×19 bit,其地址就是输出、输入的数据灰度值.存储器用双口 RAM 最为合理,Altera 公司的 StatixⅡ系列

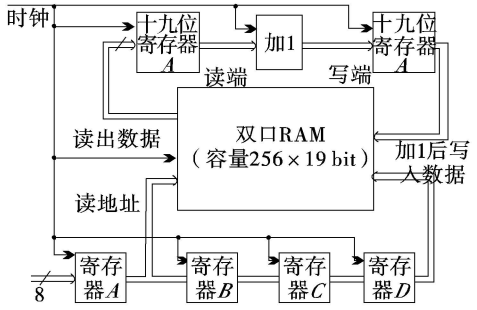


图 2　利用 FPGA 直方图统计的实现

EP2S601024C4 包含 2 544 192 bit 的可利用 RAM,所以可以用 FPGA 内的(EAB)来构造.直方图统计原理图如图 2 所示.

图像数据以系统工作时钟为节拍,被依次存放在 4 个八位移位寄存器中,第 1 个移位寄存器的内容为双端口 RAM 的读地址,第 4 个移位寄存器的内容为其写地址.第 1 个时钟脉冲,数据按寄存器 A 中的地址读出;第 2 个时钟脉冲,把从 RAM 里读出来的数据放在十九位的寄存器中;第 3 个时钟脉冲对十九位寄存器加 1,并把结果放在缓冲器中;第 4 个时钟脉冲时把结果按寄存器 D 中的地址写入 RAM 中.这样每个像素数据输入后的第 4 个时钟之后就能完成对像素的计数统计.如此,每场图像的数据结束的第 4 个脉冲后就能提供该场图像数据的直方图,从而实现实时直方图统计.

2　自适应线性 FPGA 实现

在许多实际应用中,许多图像增强的算法由于复杂度、运算量或缺乏硬件支持而难以实现实时处理.一种好的算法必须结合图像的具体特征,对绝大多数图像都有效,这就要求算法具有很强的适应性[2].因此下面就是基于一些学者提出的自适应灰度分段线形变换算法用于 FPGA 实现.

首先作如下定义[3]: 灰度最频值为直方图中具有最大像素数的灰度级; 频数为灰度重复次数,即图像中具有某灰度值的像素总数; $\{a_i, n_i\}$ 为灰度级 a_i 对应的频数 n_i.

如果存在 $\{a_0, n_0\}$,其中 a_0 为灰度频数值; n_0 为最频值对应的频数.令 $n_T = n_0 \times 10\%$,那么在[0, a_0]的灰度区间,必然存在 $\{a_L, n_L\}$ 使得[0, a_L]区间所有的 $n_i < n_T$;同样对于[a_0,255]的灰度区间,必然存在 $\{a_R, n_R\}$ 使得[a_R,255]区间所有的 $n_i < n_T$;令

$$\begin{aligned} &G(X,Y)=0 \qquad F(X,Y)<F_1 \\ &G(X,Y)=255\times(F(X,Y)-F_1)/(F_2-F_1) \\ &\qquad\qquad F_2\leqslant F(X,Y)\leqslant F_1 \qquad (1) \\ &G(X,Y)=255 \qquad F(X,Y)>F_2 \end{aligned}$$

式中,$F(X,Y)$是原始图像的灰度值; $G(X,Y)$是增强后的灰度值.

自适应分段线形变换算法的实现过程如下,为了实现对图像的实时显示,FPGA 实现需要分为 2 个阶段.

第一阶段是统计当前帧数据,用以实现直方图的统计,找到 a_L、a_R.以下是找到 a_L、a_R 的过程:

(1)统计灰度直方图,找到灰度最频值 a_0 和对应的频数 n_0;

(2)令 $n_T = n_0 \times P$;

(3)从直方图 0 级开始向右搜索,直到找到 a_L,满足对应的 $n_L < n_T$,且 $n_{L+1} > n_T$ 记下 a_L;

(4)从直方图 255 级开始向左搜索,直到找到 a_R,满足对应的 $n_R > n_T$,且 $n_{R+1} < n_T$,记下 a_R.

第二阶段是对下一帧数据进行判决附值.根据式(1):如果像素值小于 a_L(式中 F_1)的将其重新附值为 0,如果像素值大于 a_R(式中 F_2)的将其重新附值为 255,如果在两者之间的数据根据式(1)进行数据变换.在此同时也要对下一帧数据进行直方图的统计,找到 a_L、a_R.完成第一阶段的操作.

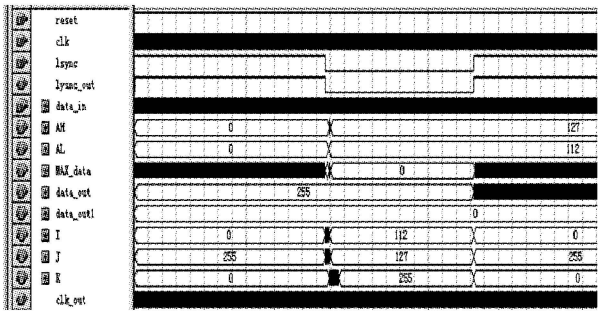
对于 640×512 的红外图像,进行 PAL 制式显示需要统计 640×256 个像素;根据 PAL 显示协议,采用系统时钟频率为 12.5 MHz.完全可以完成对实时图像送显[4].第一阶段中(2)、(3)步的实现,必须在图像显示的场消隐时间来完成,对于 $G(X,Y) = 255 \times (F(X,Y) - F_1)/(F_2 - F_1)$:乘 255 可以用左移 8 位来实现,$F_2 - F_1$(即 $a_R - a_L$)也要在场消隐时间来完成.这样对于数据的实时变换只是完成了一个减法操作、一个除法操作和移位操作,它们的操作时间延迟完全不影响在 80 ns(频率 12.5 MHz)的时钟周期下的数据传输.

3 软件的实现及仿真结果

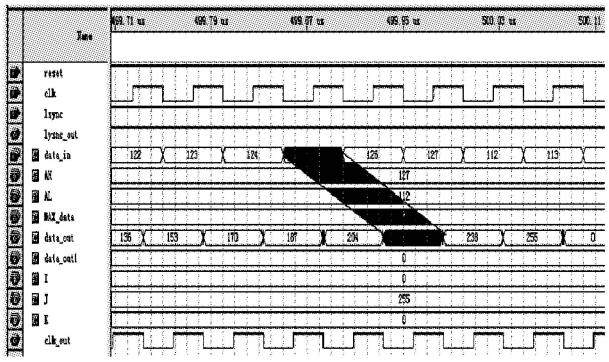
为了完整地显示出图像的内容,代码包括两部分,一是图像增强算法的实现;二是图像显示的送显程序,其中包括调用 FIFO 芯片的时序配置,ADV7123 显示芯片的控制.这里只给出了图像增强算法实现的代码核心算法部分(去掉了管脚定义及变量类型声明),此代码在 QUARTUS II 仿真软件采用 EP2S60C 可以顺利实现[5].

```
……
assign clk_out = ~clk;
image_divide (HL,FL,out,rem);//调用除法器
//- - - - - - - - - - - - - - - - - - - - - - - - - -
always @(posedge clk or negedge reset)
    if(! reset)
    begin
      I<=0;
      J<=255;
      K<=0;
      MAX_data<=1;
      state<=0;
      data_out<=0;
    end
  else if(lsync==1)
    begin
      I<=0;
      J<=255;
      K<=0;
      state<=0;
      FL<=((data_in-AL)<<8);
      data<=data_in;
      ram[data_in]<=ram[data_in]+1;
      if(MAX_data<ram[data_in])MAX_data<=ram[da-
ta_in];
      else MAX_data<=MAX_data;
      p<=(MAX_data/6);          //可调因子
      if(data<=AL)data_out[13:6]<=0;
      else if(data<AH)data_out[13:6]<=out[7:0];
      else data_out[13:6]<=255;
    end
  else
    begin
      case(state)
        1'b0:
          begin
            if(ram[I]<p)I<=I+1;
            else AL <= I;
            if(ram[J]<p)J<=J-1;
            else begin state<=1;AH<=J;HL<=J-I;end
          end
            1'b1:
              begin
                ram[K]<=0;
                if(K<254)K<=K+1;
                else K<=255;
                if(K==10)begin p<=0;data_out[13:6]<
              =0;MAX_data<=1;end
                  else begin p<=p; MAX_data<=MAX_
data;end
              end
            endcase
      end
      ……
```

图 3 是具体的仿真图,图 3a 是两帧仿真图在 lysnc 为场消隐高电平为数据送显有效时间,低电平为场消隐时间.在 lysnc 等于 1 时间段进行像素计数统计,从 lysnc 等于 0 开始进行搜索其中的 I、J,看出在不到 256 个时钟时间内,可以找到 a_L = AL 为 112 和 a_R = AH 为 127.搜索到 a_L = AL 为 112 和 a_R = AH 为 127 后,第二帧进行数据运算,对于小于等于 112 的附值为 0,大于等于 127 的附值为 255,在 112 和 127 之间的数根据式(1)进行变换.以 125 为例:$G = 255(125 - 112)/(127 - 112)$,计算结果为 121.比较仿真图,计算需要经过一个半时钟的延迟后得到的结果完全正确.



(a)像素统计搜索仿真结果



(b)数据运算的仿真结果

图 3　Quartus II 软件仿真结果

4　结　束　语

在对整个系统的试验表明,对于采用此算法进行图像增强,效果明显,整个图像无法看清楚的一些细节能够很好的显现出来.通过理论分析与实验研究表明,该算法可以获得理想红外图像的实时动态显示视觉结果,且图像清晰,实验结果较为满意.

参考文献

[1]　姜勇,刘广文,赵丹培,等.基于 FPGA 的图像增强以及滤波技术[J]. 长春理工大学学报,2002(12):20 - 22.

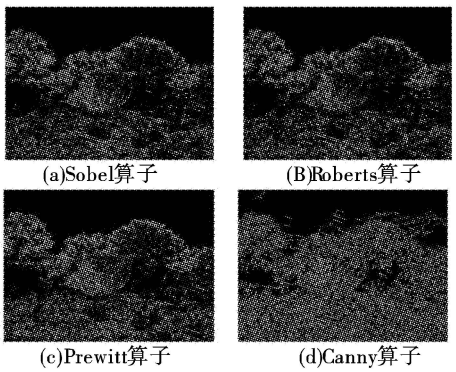
[2]　任爱峰,初秀琴.基于 FPGA 的嵌入式系统设计[M].西安:西安电子科技大学出版社,2004:260.

[3]　常本康,蔡毅.红外成像阵列与系统[M].北京:科学出版社,2006:243 - 250.

[4]　王超,叶中付.红外图像的变分增强算法[J].红外与毫米波学报,2006,25(4):306 - 310.

[5]　王辉,殷颖,陈婷,等. MAX + plus II 和 Quartus II 应用与开发技巧[M]. 北京:机械工业出版社,2007.

(上接第 3 页)



(a)Sobel算子　(B)Roberts算子

(c)Prewitt算子　(d)Canny算子

图 8　将变形支撑杆 1、2、4、5 撑起后图像边缘检测

4　结　束　语

利用变形遮障设计方法,改变目标外形轮廓,使目标具有 2 种或 2 种以上的伪装形态.通过边缘检测的图像评价方法,发现采用变形遮障后目标外形可以与背景很好地融合.

参考文献

[1]　刘凯龙.国外伪装器材及性能测试的新进展[J].装备环境工程,2006,3(5):92 - 95.

[2]　林仲贤,孙秀如.视觉及测色应用[M].北京:科学出版社,1987:38 - 46.

[3]　杨宜禾,周维真.成像跟踪技术导论[M].西安:西安电子科技大学出版社,1992:184 - 198.

[4]　李补莲.自动目标识别 ATR 技术发展述评[J].Modern Defence Technology,2000, 28(2):10 - 15.

[5]　李运华.机电控制[M].北京:北京航空航天大学出版社,2003.