

文章编号: 1002-2082(2007)06-0712-04

基于FPGA的实时中值滤波器设计

刘健鹏, 陈卫东, 钱 钧

(西安应用光学研究所, 陕西 西安 710065)

摘 要: 在图像生成和采集过程中引入的各种噪声会使图像质量变差。为了实现图像的实时预处理, 首先介绍中值滤波器的基本原理和算法, 然后在现场可编程门阵列(FPGA)上根据中值滤波的根基数算法, 采用流水技术设计一种快速实时中值滤波器, 给出按行输出的图像处理过程中存储前2行图像数据的方法。仿真结果表明: 该中值滤波器可实时完成CCD输出图像的预处理, 达到了抑制噪声及保持图像细节的目的。

关键词: 图像预处理; 中值滤波器; 现场可编程门阵列; 实时性

中图分类号: TN713

文献标志码: A

Design of real-time median filter based on FPGA

LIU Jian-peng, CHEN Wei-dong, QIAN Jun

(Xi'an Institute of Applied Optics, Xi'an 710065, China)

Abstract: Since all kinds of noises introduced in the process of the video forming and sampling inevitably lead to a bad quality of video images, the basic principle and algorithm of the median filter are presented to realize the real-time image pre-processing for succeeding processing. The rapid real-time median filter was designed with "flowing technology" according to an algorithm in the field-programmable gate array (FPGA). The method to store the first two lines of the image data in the line output based image processing is given. The simulation result shows that the median filter can implement the real-time pre-processing of CCD output images, and achieve the goal of noise suppression and image details keeping.

Key words: image pre-processing; median filter; FPGA; real-time median filtering

引言

在图像生成、采集及传输过程中, 不可避免的引入各种噪声使图像质量变差。因此必须对图像进行预处理, 抑制图像噪声, 以利于后续处理。但由于图像预处理的计算量庞大, 使用软件实现图像预处理比较慢, 达不到实时处理的要求, 因此选用硬件实现。

现场可编程门阵列(FPGA)是基于查找表结构的可编程器件, 具有丰富的寄存器资源, 而且可通过系统的重新配置来改变逻辑功能, 使得设计的灵活性大大提高。随着FPGA器件容量、功能以及

可靠性的提高, 它已经成为数字电路系统领域的主要设计方式之一。使用FPGA设计不但能大大缩减电路的体积, 提高电路的稳定性, 而且其先进的开发工具使整个系统的设计调试周期大大缩短。因此FPGA技术已经广泛应用在通信、电子、信号处理、工业控制等领域。

中值滤波是一种常用的图像预处理方法, 它的响应基于图像滤波所截取的窗口中像素的排序, 然后由排序得到的中值代替窗口中心像素的值。对于随机噪声, 中值滤波不仅具有优良的去噪能力, 而且比小尺寸的线性平滑滤波的图像模糊程度明显

收稿日期: 2006-12-13; 修回日期: 2007-04-12

作者简介: 刘健鹏(1981—), 男, 陕西扶风人, 硕士, 主要从事图像处理方面的研究。

要低,因此得到了普遍的应用。但是传统的中值滤波算法需要进行大量的数据比较运算,数据排序费时较多,不利于图像的实时处理。

1 中值滤波的基本算法及其传统实现方法

信号中值是信号值按大小排序后排在中间位置的数值,对于图像信号 $\{x_{i,j} : i, j \in N\}$,中值定义为

$$y_{i,j} = M\{x_{i,j} : i, j \in W\}$$

式中 N 表示自然数集; W 表示截取的图像窗口。窗口 W 通常有线性窗、方形窗、十字窗和菱形等多种形式。

传统的中值滤波算法实现过程如下:选择一个 $n \times n$ 的窗口(n 通常取奇数),并使该窗口沿图像数据移动;每次移动后,对窗口内的像素值进行排序,并用排序得到的中值替换窗口中心位置的像素值。

对于一幅 $M \times N$ 的图像,如果不考虑边界滤波,图像数据需要进行 $(M-n+1) \times (N-n+1)$ 次对窗口像素灰度值的排序。因此当图像数据量比较大时,中值滤波的传统算法实时性较差。

2 一种快速算法

本文首先介绍中值滤波的根基数算法。它是一种便于硬件实现的中值滤波快速算法,能实时完成图像的中值滤波。假设数字图像的像素集合为 $\{x_{i,j} : i, j \in W\}$,其像素 $x_{i,j}$ 用二进制表示为 $(b_{i,j}^1, b_{i,j}^2, \dots, b_{i,j}^L)$,中值 $y_{i,j} = M\{x_{i,j} : i, j \in W\}$ 用二进制表示为 (u_1, u_2, \dots, u_L) ,其中 L 为像素值二进制表示的位数。

如果多数元素的 $b_{i,j}^1$ 位等于1(0),根据中值的定义,不难得到 $u_1=1(0)$ 。即如果已知集合中各元素的最高位,就可以确定中值的最高位。为了确定中值的前 K 位,只需对像素集合 $\{x_{i,j} : i, j \in W\}$ 中各像素的前 K 位进行运算即可。

因此把像素集合 $\{x_{i,j} : i, j \in W\}$ 分成2组:第1组是像素最高位 $b_{i,j}^1$ 等于1的像素集合,定义此集合为 $S(b_{i,j}^1=1)$ 。第2组为最高位是0的像素集合,该集合定义为 $S(b_{i,j}^1=0)$ 。如果在像素集合 $\{x_{i,j} : i, j \in W\}$ 中,子集 $S(b_{i,j}^1=1)$ 中元素的个数多于子集 $S(b_{i,j}^1=0)$ 中元素的个数, $y_{i,j}$ 显然在 $S(b_{i,j}^1=1)$ 中,而 $S(b_{i,j}^1=0)$ 中的元素都小于 $y_{i,j}$,子集 $S(b_{i,j}^1=0)$ 中的元素变成无效数据。为了保持 $y_{i,j}$ 在像素集合 $\{x_{i,j} : i, j \in W\}$ 中仍然处在第 $(N+1)/2$ 的位置,将子

集 $S(b_{i,j}^1=0)$ 中的元素的所有位都置为0。这样处理后,虽然子集 $S(b_{i,j}^1=0)$ 中元素数值改变,但是它们仍然都小于 $y_{i,j}$,并不改变 $y_{i,j}$ 的数值,也没有改变 $y_{i,j}$ 在像素集合 $\{x_{i,j} : i, j \in W\}$ 中的位置。同理,如果子集 $S(b_{i,j}^1=1)$ 中元素的个数少于子集 $S(b_{i,j}^1=0)$ 中元素的个数, $y_{i,j}$ 显然在 $S(b_{i,j}^1=1)$ 中,将子集 $S(b_{i,j}^1=1)$ 中元素的所有位都置为1。如此处理后,虽然子集 $S(b_{i,j}^1=1)$ 中元素数值改变,但是它们仍然都大于 $y_{i,j}$,并不改变 $y_{i,j}$ 的数值,也没有改变 $y_{i,j}$ 在像素集合 $\{x_{i,j} : i, j \in W\}$ 中的位置。按照这样的算法依次计算下去,就可以计算出 $y_{i,j}$ 的 u_1, u_2, \dots, u_L ,最终求得 $y_{i,j}$ 。

3 硬件设计实现

按照上述快速算法,笔者在Xilinx公司的XC2S300E上设计并实现了 3×3 十字窗口的实时中值滤波处理。由于视频图像是按行输出的,因此在进行 3×3 十字窗口的中值滤波时,需要对前2行的图像数据进行存储。如何存储前2行图像数据是实现实时中值滤波器的一个关键因素。图1所示是实时中值滤波器的硬件框图,它由3部分组成:

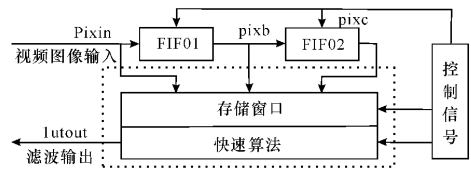


图1 中值滤波器的硬件框图

Fig. 1 The hardware block diagram of median filter

1) 快速FIFO单元

为了缩短电路的延时时间,有效地存储前2行数据,设计中利用Xilinx ISE7.1的IP核生成2个快速FIFO来负责分别缓存一行的数据。CCD输出的PAL制式的视频图像经过同步信号分离,A/D采样量化后,在读写信号的控制下被写入FIFO和从FIFO中读出。A/D输出的第 $(i-2)$ 行数据在同步时钟作用下,先按序存入FIFO1,然后按时间顺序存到FIFO2中,与此同时第 $(i-1)$ 行数据存入FIFO1,它们和第 i 行有相同列坐标的采样输入数据同时送到中值滤波模块中进行处理。

2) 中值滤波单元

图1中虚线框中的部分是本设计的核心中值滤波单元。这部分的输入信号是视频数据连续3行有相同列坐标的3个数据 $pixin, pixb, pixc$ 和时钟

clk,输出是滤波处理结果中值lutout。中值滤波采用VHDL语言设计,利用FPGA的硬件资源,首先生成3×3的矩形存储窗口,将3行数据分别按行移位后,存入存储窗口;然后根据滤波窗口的形式,将参与计算的5个数据取出;最后利用上述快速算法求出十字滤波窗口的中值。部分代码如下:

//3×3 矩形存储窗口

```
signal apixa: std_logic_vector(7 downto 0);
signal apixb: std_logic_vector(7 downto 0);
signal apixc: std_logic_vector(7 downto 0);
signal bpixa: std_logic_vector(7 downto 0);
signal bpixb: std_logic_vector(7 downto 0);
signal bpixc: std_logic_vector(7 downto 0);
signal cpixa: std_logic_vector(7 downto 0);
signal cpixb: std_logic_vector(7 downto 0);
signal cpixc: std_logic_vector(7 downto 0);
//定义新变量并赋值(选取十字窗口)
pix1:=apixb; pix2:=bpixa; pix3:=bpixb;
pix4:=bpixc; pix5:=cpixb;
```

//VHDL语言实现快速算法

```
for i in 7 downto 0 loop
    qq:="0000";
    //统计第i位是1的数据个数
    if(pix1(i)='1') then
        qq:=qq+1; end if;
    if(pix2(i)='1') then
        qq:=qq+1; end if;
    if(pix3(i)='1') then
        qq:=qq+1; end if;
    if(pix4(i)='1') then
        qq:=qq+1; end if;
    if(pix5(i)='1') then
        qq:=qq+1; end if;
    //如果多数数据的第i位是0,则将首位是1
    的数据各位都置为1
    if(qq<3)then
```

```
data(i):='0';
if(pix1(i)='1') then
    pix1:="11111111"; end if;
if(pix2(i)='1') then
    pix2:="11111111"; end if;
if(pix3(i)='1') then
    pix3:="11111111"; end if;
if(pix4(i)='1') then
    pix4:="11111111"; end if;
if(pix5(i)='1') then
    pix5:="11111111"; end if;
//反之,如果多数数据的第i位是1,则将首位
是0的数据各位都置为0
data(i):='1';
if(pix1(i)='0') then
    pix1:="00000000"; end if;
if(pix2(i)='0') then
    pix2:="00000000"; end if;
if(pix3(i)='0') then
    pix3:="00000000"; end if;
if(pix4(i)='0') then
    pix4:="00000000"; end if;
if(pix5(i)='0') then
    pix5:="00000000"; end if;
end loop;
lutout<=data;
```

3) 控制电路单元

控制电路单元完成对数据流向的控制,为了保持输入数据的顺序性,数据输入流向始终控制为A/D→FIFO1→FIFO2。数据读出则是同时读出三个有相同列坐标的数据到中值滤波单元。

4 时序仿真

使用Xilinx ISE Simulator进行行为级仿真,仿真的时序图如图2所示。

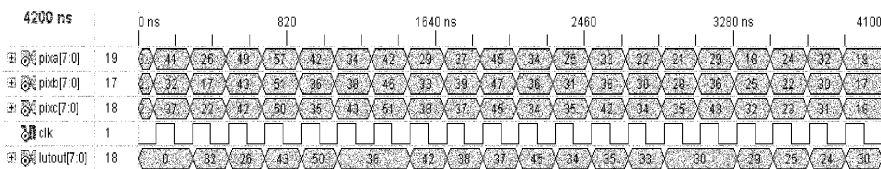


图2 时序仿真图

Fig. 2 Time sequence simulation diagram

5 结论

根据中值滤波的根基数算法,采用流水技术实现了实时中值滤波器的设计。通过对CCD输出的视频信号进行 3×3 十字窗口快速中值滤波处理,图像噪声得到有效抑制,质量显著改善。本设计在Xilinx公司的FPGA XC2S300E上利用VHDL得以实现,并且只需修改VHDL代码就可实现任意形式窗口的中值滤波器设计。

参考文献:

- [1] 冈萨雷斯,阮秋琦,阮宇智. 数字图像处理[M]. 2版. 阮秋琦,阮宇智,译. 北京:电子工业出版社,2005.
GONZALEZ R G. Digital image processing[M]. 2nd ed. translated by RUAN Qiu-qi, RUAN Yu-zhi. Beijing:Publishing House of Electric Industry,2005. (in Chinese)
- [2] ATAMAN E, AATER V K, WONG K M. A fast method for real-time median filtering [J]. IEEE

Transactions on ASSP,1980,28(4):415-421.

- [3] 李刚强,田斌,易克初. FPGA设计中关键问题的研究[J]. 电子技术应用,2002(6):68-71.
LI Gang-qiang, TIAN Bin, YI Ke-chu. Key problem research on FPGA design [J]. Application of Electronic Technology Technique, 2002(6): 68-71. (in Chinese)
- [4] 曾桂林. 电视跟踪在光电火控系统中的应用及发展趋势[J]. 应用光学,2001,22(2):1-5.
ZENG Gui-lin. Applications of TV tracking in electrooptic systems and its development[J]. Journal of Applied Optics,2001,22(2):1-5. (in Chinese)
- [5] 陈家祯,郑子华,叶锋. 基于可编程芯片的图像实时中值滤波实现[J]. 长沙大学学报,2006,20(2):62-64.
CHEN Jia-zhen, ZHENG Zi-hua, YE Feng. Realization of medium filtering of image data based on programmable chip [J]. Journal of Changsha University,2006,20(2):62-64. (in Chinese)