

文章编号: 1672-8785(2015)11-0019-06

基于硬件描述语言的图像边缘特征提取技术研究

刘宗明^{1,2,3} 张 宇^{2,3} 曹姝清^{2,3} 卢 山^{2,3} 叶 东¹

(1、哈尔滨工业大学电气工程及自动化学院, 黑龙江哈尔滨 150001;

2、上海航天控制技术研究所, 上海 200233;

3、上海市空间智能控制技术重点实验室, 上海 200233)

摘 要: 对图像边缘特征的提取是图像处理中一项非常重要的研究内容。相比于角点信息, 边缘信息对观测目标具有更丰富的表示含义, 对目标外形轮廓的精确提取具有重要的研究意义。随着探测器分辨率的提升以及对数据更新率要求的提高, 开发能快速高效地实现边缘提取的算法已成为人们研究的热点。着重研究了高精度图像处理算法的 FPGA 实现技术, 并利用 FPGA 独有的并行流水线处理机制实现了高斯滤波、sobel 算子和非极大值抑制等图像处理的基本运算。

关键词: 硬件描述语言; 图像处理; 高斯滤波; sobel 算子; 非极大值抑制

中图分类号: TP274+.2 **文献标志码:** A **DOI:** 10.3969/j.issn.1672-8785.2015.11.004

Research on Image Edge Feature Extraction Technique Based on Hardware Description Language

LIU Zong-ming^{1,2,3}, ZHANG Yu^{2,3}, CAO Shu-qing^{2,3}, LU Shan^{2,3}, YE Dong¹

(1. School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150001, China;

2. Shanghai Institute of Spaceflight Control Technology, Shanghai 201109, China;

3. Shanghai Key Laboratory of Space Intelligent Control Technology, Shanghai 201109, China)

Abstract: The extraction of image edge features is very important in the processing of images. Compared with the corner information, the edge information contains richer representation mean of the target to be observed. It is of significance to the accurate extraction of target contour. With the improvement of detector's resolution and the increase of data update rate, development of the algorithms for fast and effective edge extraction has been one of the research hotspots. The technique for implementing high-precision image processing algorithms using FPGA is studied emphatically. The basic image processing operations such as Gaussian filter, sobel operator and non-maxima suppression are implemented by using the unique parallel pipeline processing mechanism of FPGA.

Key words: hardware description language; image processing; Gaussian filter; sobel operator; non-maxima suppression

收稿日期: 2015-10-04

基金项目: 国家自然科学基金面上项目 (61374162)、上海市科技人才计划 (14XD1421400、14QB1401800) 资助项目

作者简介: 刘宗明 (1986-), 男, 山东青岛人, 硕士, 工程师, 从事计算机视觉、嵌入式图像处理、光学测量敏感器设计与研制工作。 E-mail: zongmingliu@163.com

0 引言

边缘像素是图像中灰度突变的那些像素,而边缘是连接边缘像素的几何。一条可视为边缘的线段,其两侧的背景灰度远高于或远低于该线段的像素灰度。作为一种基本的图像处理算法,边缘特征提取的运算量大,涉及到对整幅图像的多次卷积运算。对一些实时性要求较高的系统而言,图像的处理速度往往与处理效果同等重要。数字信号处理器 (Digital Signal Processor, DSP) 通常使用 C 语言进行编程,其内部采用专用硬件实现一些数字信号处理常用的运算。但从根本上来说, DSP 的指令执行体系仍然是串行的,这使得它在图像处理方面的使用受到了极大的限制。而 FPGA 独有的并行流水线处理机制可以有效避免因串行指令执行而引起的耗时过长问题,可以更好地解决由于探测器分辨率提升而引起的卷积运算时间过长,从而导致处理速度下降、数据更新率降低这一矛盾。本文针对边缘提取过程中的滤波、提取和非极大值抑制三个主要步骤所涉及的硬件描述语言实现方法展开描述,并利用并行流水处理机制实现高斯滤波、sobel 算子和非极大值抑制等图像处理的基本运算。

1 高斯滤波

在数字信号处理中,信息处理系统的目标是从带有噪声的测量信号中提取出真实有用的信息。但真实有用的信息往往无法预知,因此只能利用滤波法进行估计。在进行数学仿真或者误差评估时,往往认为传感器所引入的噪声服从正态分布(高斯白噪声),因此在滤波时可以有针对性地进行滤波器设计。高斯滤波函数为

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

高斯函数是各向同性的,其曲线为草帽状的对称图,如图 1 所示,对该曲线的整个覆盖面积求积分,结果为 1。

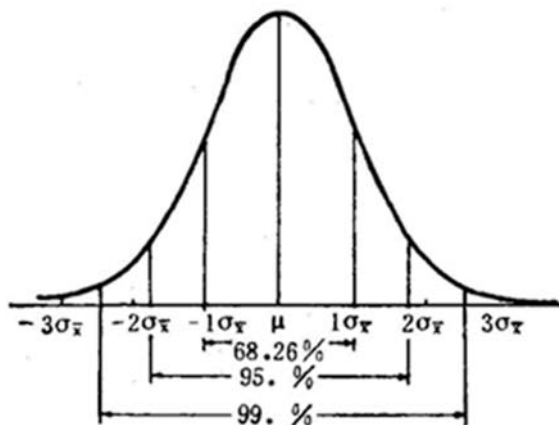


图 1 高斯函数描述曲线

高斯滤波器是一种平滑线性滤波器。平滑滤波器就是用滤波掩模确定的邻域内像素的平均灰度值去替代图像的每个像素点的值。作为一种带有权重的平均值,高斯滤波器以离散点上的函数值为权值,对采集到的灰度图像像素点作邻域一定范围内的加权平均,中心的权重比邻近像素的更大,这样就可以克服边界效应,有效消除高斯噪声。考虑到滤波算法对 FPGA 的资源需求及滤波效果,本文选用了 5×5 的高斯模板(表 1)。

表 1 Gauss 模板

1	2	3	2	1
2	5	6	5	2
3	6	8	6	3
2	5	6	5	2
1	2	3	2	1

具体公式如下:

$$g(x, y) = \left\{ \begin{aligned} & [f(x-2, y-2) + f(x-2, y+2) + f(x+2, y-2) + f(x+2, y+2)] \\ & + [f(x-1, y-2) + f(x-2, y-1) + f(x-2, y+1) + f(x-1, y+2)] \times 2 \\ & + [f(x+1, y+2) + f(x+2, y+1) + f(x+2, y-1) + f(x+1, y-1)] \times 2 \\ & + [f(x, y-2) + f(x-2, y) + f(x, y+2) + f(x+2, y)] \times 3 \\ & + [f(x-1, y-1) + f(x-1, y+1) + f(x+1, y+1) + f(x-1, y+1)] \times 5 \\ & + [f(x, y-1) + f(x-1, y) + f(x, y+1) + f(x+1, y)] \times 6 \\ & + f(x, y) \times 8 \end{aligned} \right\} / 88 \quad (2)$$

式中, $f(x, y)$ 为原图像中 (x, y) 像素点的灰度值, $g(x, y)$ 为经过高斯滤波的值。由于在 FPGA 中完成除法运算所占的资源太大,通常可

采用移位法来代替除法运算。因此, 文中将除 88 改为除 64, 即向右移 6 位。在 FPGA 中对图像进行滤波时采用流水处理方式, 即在图像数据流经过处理模块的同时完成对图像数据的操作, 与数据传输同步, 无需额外的处理时间。如果采用 5×5 的掩模算子进行滤波, 对当前像素处理时需要周围的像素数据, 因此, FPGA 内部采用行缓冲前四行的数据, 如图 2 所示。每个行缓冲模块中, 缓冲数据的个数比实际的行数据个数少 5, 当第 5 行的数据被传输到 FPGA 时就构成了 5×5 的窗口。

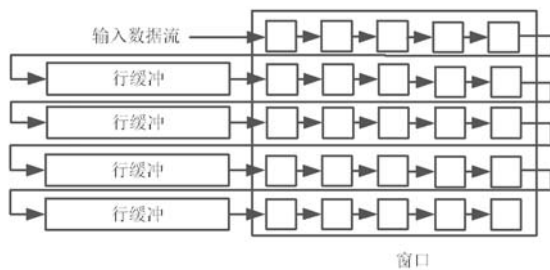


图 2 FPGA 的缓冲数据流

高斯滤波的 Verilog HDL 语言实现关键代码的过程如下:

```
always @(posedge clk)
begin
    aa11<=a11; aa12<=a12*2; aa13<=a13*3; aa14<=a14*2; aa15<=a15;
    aa21<=a21*2; aa22<=a22*5; aa23<=a23*6; aa24<=a24*5; aa25<=a25*2;
    aa31<=a31*3; aa32<=a32*6; aa33<=a33*8; aa34<=a34*6; aa35<=a35*3;
    aa41<=a41*2; aa42<=a42*5; aa43<=a43*6; aa44<=a44*5; aa45<=a45*2;
    aa51<=a51; aa52<=a52*2; aa53<=a53*3; aa54<=a54*2; aa55<=a55;
    GAUSS1<=aa11+aa15+aa51+aa55;
    GAUSS2<=aa12+aa14+aa21+aa25+aa41+aa45+aa52+aa54;
    GAUSS3<=aa13+aa31+aa35+aa53;
    GAUSS4<=aa22+aa24+aa42+aa44;
    GAUSS5<=aa23+aa32+aa43+aa34;
    GAUSS6<=aa33;
    GAUSS<=GAUSS1+GAUSS2+GAUSS3+GAUSS4+GAUSS5+GAUSS6;
    DATA_OUT<=GAUSS>>6;
end
```

其中 a_{11} 、 a_{12} 、 a_{13} 、 a_{14} 和 a_{15} , a_{21} 、 a_{22} 、 a_{23} 、 a_{24} 和 a_{25} , a_{31} 、 a_{32} 、 a_{33} 、 a_{34} 和 a_{35} , a_{41} 、 a_{42} 、 a_{43} 、 a_{44} 和 a_{45} , a_{51} 、 a_{52} 、 a_{53} 、 a_{54} 和 a_{55} 分别为 FIFO 模块缓存的五行五列数据, DATA_OUT 为高斯滤波模块的最终输出数据。高斯滤波的仿真图如图 3 所示, 从图中可以看出, 通过 FIFO 模块的缓存及移位处理, 参与运算的变量格式对齐了。当第五行的第五个元素到来时便形成了 5×5 的模板, 此时便可进行掩模运算。

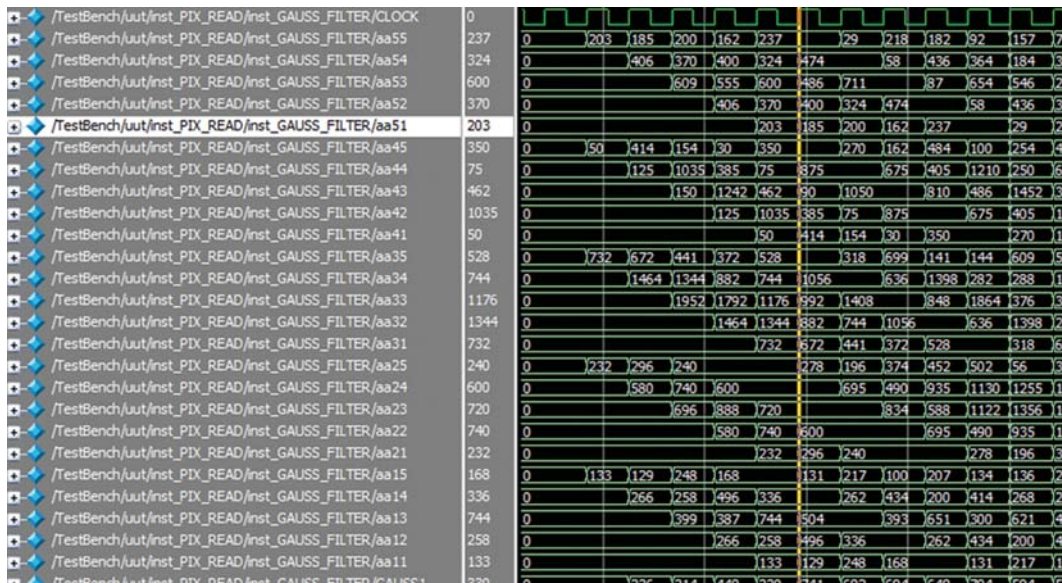


图 3 高斯模板的时序图

2 sobel 边缘提取

sobel 算子是基于一阶微分的边缘提取算法。梯度是函数变化的一种度量, 一幅图像可

以看作是图像灰度连续函数的取样点数组。因此, 图像灰度值的显著变化 (边缘) 可以用梯度的离散逼近。设 $f(x, y)$ 是图像函数, 它在 (x, y)

处的梯度矢量为

$$D[x, y] = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T \quad (3)$$

梯度方向在函数 $f(x, y)$ 变化率最大的方向上, 其幅值可以表示为

$$D[f(x, y)] = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (4)$$

经典 sobel 算子包含两个方向卷积模板, 如图 4 所示。

-1	0	1
-2	0	2
-1	0	1

(a) 水平梯度模板 D_x

1	2	1
0	0	0
-1	-2	-1

(b) 垂直梯度模板 D_y

图 4 经典 sobel 算子包含的两个方向的卷积模板

图 4 中, D_x 主要用来检测图像的垂直方向边缘, D_y 主要用来检测图像的水平方向边缘。

$$D_x = [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)] \quad (5)$$

$$D_y = [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] - [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)] \quad (6)$$

因此, 对于数字图像, 梯度的幅值和方向可分别简化为 $|D_x| + |D_y|$ 和 $\arctan(D_y/D_x)$ 。

与高斯滤波的图像处理 FPGA 实现方式同理, sobel 算子只需要作两行两列的数据缓存。当第三行的第三个数到来时便形成 3×3 的模板, 即可进行掩模处理。

sobel 算子的 Verilog HDL 语言实现关键代码的过程如下:

```
always @(posedge clk)
begin
    Mx1 <= D13 + D23; My1 <= D31 + D32;
    Mx2 <= D23 + D33; My2 <= D32 + D33;
    Mx3 <= D11 + D21; My3 <= D11 + D12;
    Mx4 <= D21 + D31; My4 <= D12 + D13;
    Mx5 <= Mx1 + Mx2; My5 <= My1 + My2;
    Mx6 <= Mx3 + Mx4; My6 <= My3 + My4;
    Dx <= Mx5 - Mx6; Dy <= My5 - My6;
end
```

其中 $D11$ 、 $D12$ 和 $D13$, $D21$ 、 $D22$ 和 $D23$, $D31$ 、 $D32$ 和 $D33$ 分别为 FIFO 模块缓存的三行三列数据, $Mx1 \sim Mx6$ 、 $My1 \sim My6$ 为中间变量, Dx 为垂直方向的边缘输出数据, Dy 为水平方向的边缘输出数据。

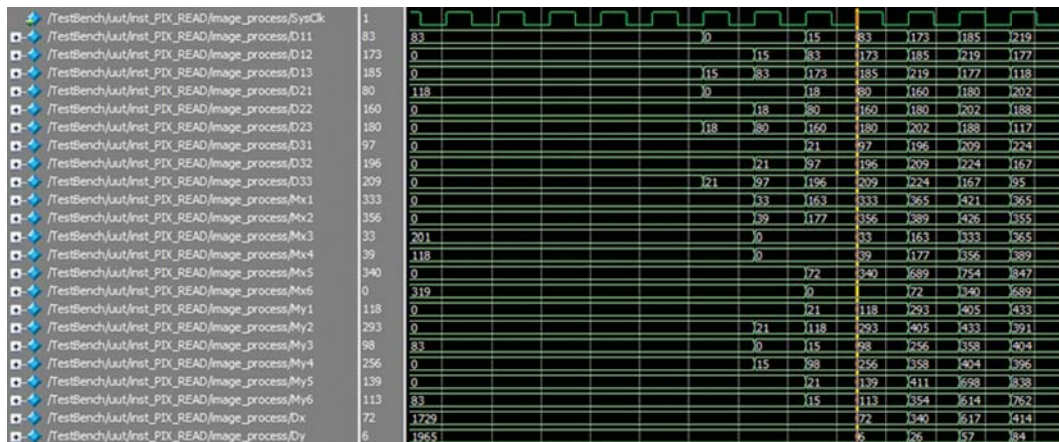


图 5 sobel 模板的时序图

3 非极大值抑制

得到图像的边缘后便可进行边缘图像的非

极大值抑制和二值化处理。非极大值抑制是将像素的梯度值与沿其梯度方向相邻的梯度值进行比较, 如果比相邻的梯度值小, 则可判定此像

素点为非边缘点, 将其梯度值置 0; 反之, 则判定为边缘点。

用 sobel 算子得到 D_x 和 D_y 后, 分别进行两行数据缓存, 得到两个 3×3 的窗口, 以进行非极大值抑制, 如图 6 所示。

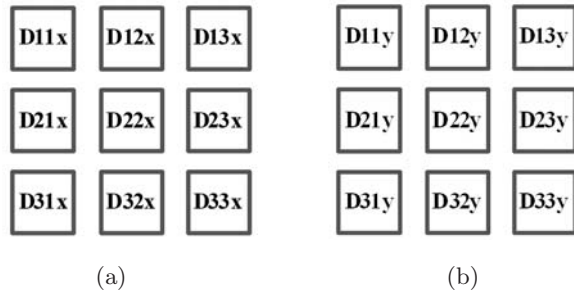


图 6 sobel 算子计算结果

对于当前的梯度值 $D_{22} = |D_{22x}| + |D_{22y}|$, 有

$$\begin{cases} d_{tmp1} = w \cdot d_1 + (1 - w) \cdot d_2 \\ d_{tmp2} = w \cdot d_3 + (1 - w) \cdot d_4 \end{cases} \quad (7)$$

若 D_{22} 大于 d_{tmp1} 和 d_{tmp2} , 则认为它是边缘点, 否则将 D_{22} 值置为 0。式 (7) 中 w 的值为

$$w = \begin{cases} \frac{|D_x|}{|D_y|} & \text{if } (|D_x| < |D_y|) \\ \frac{|D_y|}{|D_x|} & \text{if } (|D_x| \geq |D_y|) \end{cases} \quad (8)$$

d_1 、 d_2 、 d_3 、 d_4 的值和梯度方向 $\arctan(D_y/D_x)$ 的关系如图 7 所示, 将梯度方向分为 4 个区域。

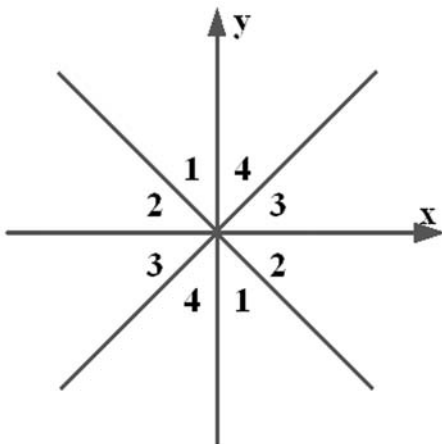


图 7 边缘梯度方向的判别

各个区域 d_1 、 d_2 、 d_3 、 d_4 的值如图 8 所示。

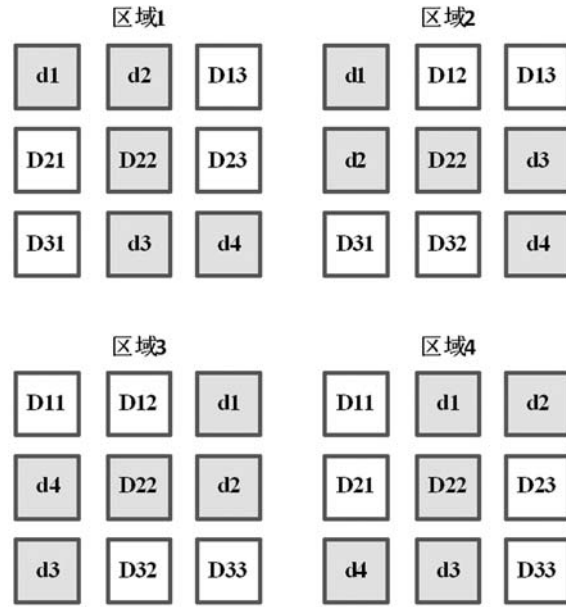


图 8 进行非极大值抑制的 d 值的选取

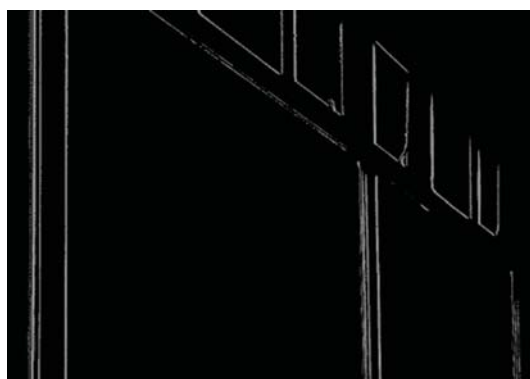
在 FPGA 内判断梯度方向时不能使用公式 $\arctan(D_y/D_x)$ 。因为只有 4 个方向区域, 可以只根据 D_y 和 D_x 的正负号是否一致以及各自绝对值的大小两项来进行判断梯度方向:

$$\begin{cases} D_x \text{ 与 } D_y \text{ 异号, } |D_x| < |D_y| & (\text{区域1}) \\ D_x \text{ 与 } D_y \text{ 异号, } |D_x| \geq |D_y| & (\text{区域2}) \\ D_x \text{ 与 } D_y \text{ 同号, } |D_x| \geq |D_y| & (\text{区域3}) \\ D_x \text{ 与 } D_y \text{ 同号, } |D_x| < |D_y| & (\text{区域4}) \end{cases} \quad (9)$$

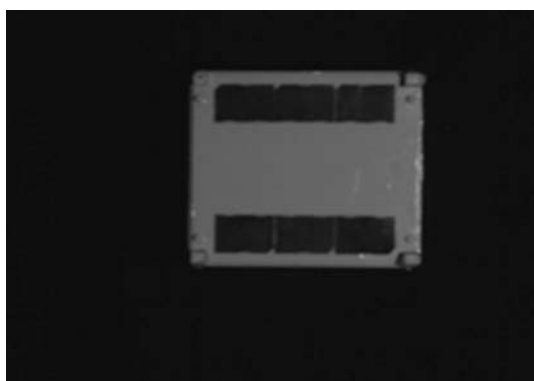
计算 w 值时也会涉及到除法运算。为了比较 D_{22} 与 d_{tmp1} 、 d_{tmp2} 的大小关系, 将此除法运算的分母同时乘以两个用来比较的数, 以避免除法运算, 提高运算速度并且达到处理效果。完成非极大值抑制后, 对得到的新值进行二值化处理; 将高于阈值的点的像素值设置为 255, 而将低于阈值的点像素值设置为 0。图 9 中, (a) 和 (c) 为用 APTINA 公司 300 万像素分辨率的探测器成像后的静态图片, (b) 和 (d) 为经过 FPGA 驱动实现滤波、边缘提取和非极大值抑制后的图像。从图中可以看出, 经过一定程度的细化处理后, 窗框和模型的边缘特征被较完整地提取了出来, 效果较明显。



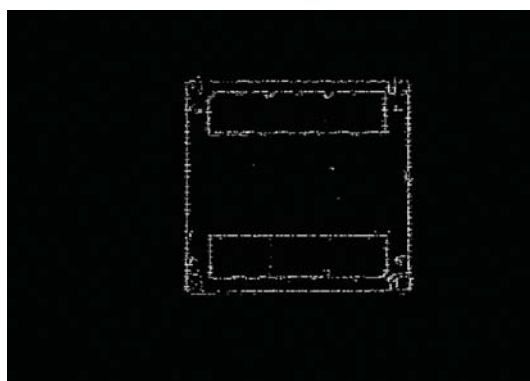
(a) 原始灰度图像 1



(b) 边缘提取后图像 1



(c) 原始灰度图像 2



(d) 边缘提取后图像 2

图 9 边缘提取实验效果图

4 结 论

在提取边缘特征时,必须对整幅图像进行多次卷积运算,且运算量大,计算耗时较长。针对这一问题,放弃了在 DSP 中完成掩模运算的传统方法,采用 FPGA 并行流水处理机制实现了高斯滤波算法、sobel 边缘检测算法和非极大值抑制算法,并将图像处理中的边缘提取流程完整地搬移到了 FPGA 平台上,有效地避免了因串行指令执行而引起的耗时过长问题,对大面阵、高分辨率探测器在图像处理方面的应用具有重要的推动作用,而且有利于提升整个系统的数据更新率。

参考文献

- [1] 位昱堡. 基于 FPGA 的图像处理算法的研究与硬件设计 [D]. 南昌: 南昌大学信息工程学院, 2006: 1-72.
- [2] 沈德海, 侯建, 鄂旭. 基于改进的 sobel 算子边缘检测算法 [J]. 计算机技术与发展, 2013, 23(11): 22-25.
- [3] 王冠, 丁友东, 魏小成. 基于改进 Sobel 算子的文物图像检索 [J]. 计算机技术与发展, 2011, 21(10): 51-54.
- [4] 孙伟, 王志坚. 基于标准梯度的模糊边缘检测算法 [J]. 红外与激光工程, 2005, 34(4): 454-458.
- [5] Health A, Sarkar S, Sanocki T, et al. Comparison of Edge Detectors: A methodology and Initial Study[J]. Computer Vision and Image Understanding, 1998, 69(1): 38-54.
- [6] Demigny D, Kamle T. A Discrete Expression of Canny's Criteria for Step Edge Detector Performances Evaluation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(11): 1199-1211.
- [7] 钟钊, 周激流, 郎方年, 等. 边缘检测滤波尺度自适应选择算法 [J]. 自动化学报, 2007, 33(8): 867-870.
- [8] 付永庆, 王咏胜. 一种基于数学形态学的灰度图像边缘检测算法 [J]. 哈尔滨工程大学学报, 2005, 26(5): 685-687.

(下转第 40 页)