



Canny 边缘检测算法在飞腾平台上的实现与优化

郭恒亮¹, 柴晓楠², 韩林¹, 赫晓慧³, 商建东¹

(1. 郑州大学 河南省超级计算中心, 郑州 450000; 2. 郑州大学 信息工程学院, 郑州 450000;

3. 郑州大学 地球科学与技术学院, 郑州 450000)

摘要: 为实现国产飞腾 DSP 平台对底层图像库的支持, 针对原始 Canny 边缘检测算法计算时间过长的问题, 设计一种面向 FT-M7002 平台的 Canny 梯度计算并行算法。基于 FT-M7002 高性能处理架构, 采用单指令流多数据流向量化方式增强 DSP 内核指令的并行处理能力, 根据 FT-M7002 平台向量存储器的层次结构特征, 分析 Canny 梯度计算并行算法的访存模式, 通过首地址偏移取址解决不连续访存问题, 并结合双缓冲方式完成数据传输与数据计算。实验结果表明, 在与原始 Canny 算法具有相同检测精度的情况下, 该算法在卷积核大小为 3×3 、 5×5 、 7×7 时整体运行速度提升了 1.490~2.112 倍, 缩小了与主流加速器件在数字图像处理领域的性能差距。

关键词: FT-M7002 处理器; Canny 边缘检测; 梯度计算并行; 访存优化; 双缓冲方式

开放科学(资源服务)标志码(OSID):



中文引用格式: 郭恒亮, 柴晓楠, 韩林, 等. Canny 边缘检测算法在飞腾平台上的实现与优化[J]. 计算机工程, 2021, 47(7): 37-43.

英文引用格式: GUO H L, CHAI X N, HAN L, et al. Implementation and optimization of Canny edge detection algorithm on FT platform[J]. Computer Engineering, 2021, 47(7): 37-43.

Implementation and Optimization of Canny Edge Detection Algorithm on FT Platform

GUO Hengliang¹, CHAI Xiaonan², HAN Lin¹, HE Xiaohui³, SHANG Jiandong¹

(1. Henan Province Supercomputing Center, Zhengzhou University, Zhengzhou 450000, China;

2. School of Information Engineering, Zhengzhou University, Zhengzhou 450000, China;

3. School of Earth Science and Technology, Zhengzhou University, Zhengzhou 450000, China)

[Abstract] In order to support the underlying image library on the FT DSP platform, and reduce the time consumed by the calculation in the Canny edge detection algorithm, an algorithm for parallel Canny gradient computing based on FT-M7002 is proposed. On the basis of FT-M7002 high-performance processing architecture, Single Instruction Multiple Data (SIMD) is vectorized to enhance the parallel processing of the instructions of DSP cores. According to the hierarchical structure features of the vector memory of FT-M7002, the mode of data memory access of the Canny parallel gradient computing algorithm is analyzed. The first address offset is used to deal with discontinuous data memory access, and data transmission and data calculation is completed by means of double buffering mode. Experimental results show that when reaching the same detection accuracy as the original Canny algorithm, the proposed algorithm improves the overall running speed by 1.490~2.112 times when the size of convolution core is 3×3 , 5×5 , and 7×7 , bridging the performance gap with the mainstream accelerators in digital image processing.

[Key words] FT-M7002 processor; Canny edge detection; parallel gradient computing; memory access optimization; double buffering mode

DOI: 10.19678/j.issn.1000-3428.0059943

0 概述

边缘检测是数字图像处理领域的关键技术, 近

年来在道路车辆识别^[1]、遥感影像处理^[2-3]、医学影像识别^[4]等任务中得到广泛应用。数字图像处理中的边缘是周围像素极大变化点的集合。边缘检测是指

基金项目: 国家重点研发计划(2018YFB0505000)。

作者简介: 郭恒亮(1971—), 男, 副教授, 主研方向为智慧城市、地理信息系统、高性能计算; 柴晓楠, 硕士研究生; 韩林(通信作者), 副教授; 赫晓慧、商建东, 教授。

收稿日期: 2020-11-09 修回日期: 2020-12-10 E-mail: hanlin@zzu.edu.cn

一种保留图像结构属性点、大幅度减少图像数据量并剔除不相关信息的图像处理方法。经典的边缘检测算法主要基于边缘算子,并且随着人工智能技术的发展,又出现了基于形态学、小波变换和机器学习等的边缘检测算法^[5-7]。由于图像处理技术在人工智能、大数据等领域得到广泛应用,因此对于硬件加速处理器的计算力要求越来越高。DSP具有快速、高效、低功耗和便于集成等特点,适用于语音处理、图像处理^[8]、信息通信等应用场景。面对新应用对计算机性能提出的更高要求,目前DSP趋向于通过单指令流多数据流(Single Instruction Multiple Data, SIMD)位长和超长指令字(Very Long Instruction Word, VLIW)等技术来增强DSP内核指令的并行处理能力^[9-11]。DSP底层架构的不断优化升级带来性能提升的同时,也对上层软件提出了更高的要求,因此优化软件处理算法以充分利用高性能的DSP架构也是DSP应用领域亟需解决的问题。

针对不同应用场景提取的边缘精度不同问题,文献[12-14]通过根据不同的应用场景选择不同的平滑去噪方式或增强边缘轮廓来获取更精准边缘。文献[15-16]选取不同边缘阈值以提取更精准边缘特征。Canny边缘检测是一种广泛使用的边缘检测算法,在此基础上提出的改进滤波、叠加其他增强边缘算法和改进阈值^[17-19]等优化方案均是基于算法应用层面,而面向加速并行的优化方案目前还较少。由国防科技大学研制的FT-M7002是一款具有512位长SIMD的高性能并行处理器,DSP内核基于超长指令字结构,具有标量以及向量处理单元,32位峰值性能可达200GFLOPS和100GMACS^[20-21]。由于国内对DSP方面的研究起步较晚,硬件与软件支持不匹配,面向国产平台加速并行优化应用较少,而FT-M7002具有高度并行处理特性,因此本文基于FT-M7002架构进行Canny边缘检测算法的实现与优化。

1 Canny边缘检测算法原理与硬件平台架构

常见的经典边缘检测算子包括Sobel算子、Robel算子、Priwitt算子、Laplacian算子及Canny算子。边缘检测算子的差异主要为计算边缘点时的图片结构特征提取方式不同。根据不同的图片结构特征提取方式,提取的边缘效果也不同。Canny边缘检测算子是由

John F. Canny于1986年提出的具有附加响应最优检测、检测边缘位置偏差最小以及单个边缘点变化定位误差小等优点的算子,是目前应用较广泛的算子。

1.1 Canny边缘检测算法

Canny边缘检测是一种基于模板卷积计算保留图片特征信息的算法,检测流程如图1所示。

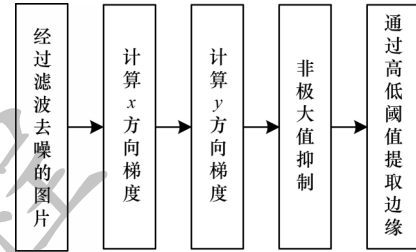


图1 Canny边缘检测流程

Fig.1 Procedure of Canny edge detection

Canny边缘检测算法的具体步骤如下:

1) Canny边缘检测属于先平滑后求导的算法,通常使用高斯滤波平滑图像。 $G(x, y)$ 表示二维高斯函数,即图像去噪平滑使用的卷积操作数,如式(1)所示。 $f_s(x, y)$ 为滤波卷积平滑后的图像像素数据,其中 $f(x, y)$ 表示原图像像素数据,即输入的原数据,如式(2)所示。

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

$$f_s(x, y) = f(x, y) * G(x, y) \quad (2)$$

2) 计算图像每个像素的点梯度, x 方向和 y 方向偏导求梯度的具体计算过程为通过一阶有限差分近似求取灰度的梯度值,如式(3)和式(4)所示。梯度值为矢量,即包含幅值和方向,根据式(5)求出幅值 $M(x, y)$ 。变化率又称为幅值,幅值越大,像素区域灰度值变化越明显,而像素灰度值变化越明显的点越能标识图片结构特征。根据式(6)计算得到判断其方向变化的 $\alpha(x, y)$ 。

$$g_x = \frac{\partial f}{\partial x} \quad (3)$$

$$g_y = \frac{\partial f}{\partial y} \quad (4)$$

$$M(x, y) = \sqrt{g_x^2 + g_y^2} \quad (5)$$

$$\alpha(x, y) = \arctan \frac{g_x}{g_y} \quad (6)$$

3) 对幅值图像进行非极大值抑制,根据象限两个对角线的4个方向进行划分,如图2(a)和图2(b)所示,

共分为8个部分,即上(Up)、下(Down)、左(Left)、右(Right)、左上(LU)、左下(LD)、右上(RU)和右下(RD),对幅值进行非极大值抑制。

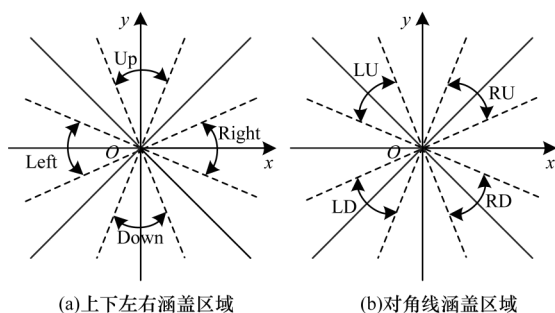


图2 4个方向的角度图

Fig.2 Angle diagram of four directions

4)选取高阈值 T_H 和低阈值 T_L ,判断经过极大值抑制之后的像素梯度,若幅值小于 T_L 的点则抛弃,若

大于 T_H 的点则标记,而大于 T_L 且小于 T_H 的点根据周围 N 连通区域判断其是否为边缘点。

1.2 飞腾平台架构

飞腾平台是一款高性能数字信号处理器,搭载自主设计的高性能处理芯片,芯片主频为1 GHz,峰值性能为200GFLOPS,包含1个RISC CPU核和2个DSP核,如图3所示,具有全局共享Cache(SubGC)、核间同步、DDR3存储器以及PCIE等多个设备。内核与核外设备为环形连接,环形互连包含双向读写环路和单项配置命令环路,其中单项数据位宽为256位。内核采用超长指令字结构,包含向量处理单元(Vector Processing Unit, VPU)与标量处理单元(Scalar Processing Unit, SPU)。飞腾平台架构的主要特点是支持512位SIMD操作以及半字(16位数据)和字(32位数据)大小的数据变量。

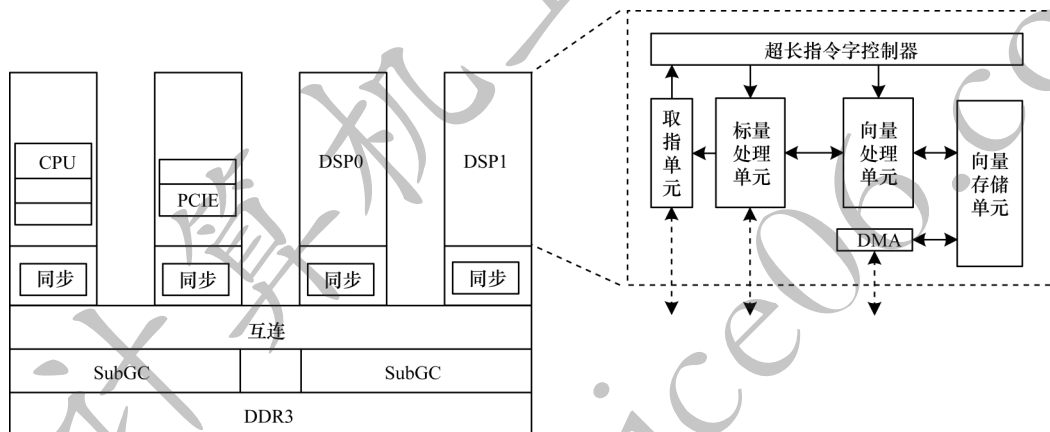


图3 飞腾平台架构

Fig.3 FT platform architecture

2 基于飞腾平台的Canny算法实现与优化

2.1 Canny边缘检测算法的实现

飞腾平台具有相应的FT-M7002 IDE,借助IDE生成工程后,C程序通过M7002编译器生成对应的汇编文件,经M7002汇编器将汇编文件转换为ELF目标文件,最终通过M7002链接器将目标文件以及对应相关库链接生成一个可执行文件。可执行文件经过IDE下载到飞腾对应的内存地址空间处执行程序。在飞腾平台上实现Canny算法时,考虑到实际检测边缘过程中采用的去噪平滑方式不同,因此本文将高斯平滑去噪部分略去,在滤波平滑去噪后进行梯度计算以及边缘阈值提取。基于飞腾平台的Canny边缘检测算法实现流程如图4所示。

在实际代码的实现过程中,首先Canny接口直接进行图片的梯度计算,图片读入默认为8位数据变量类型(char),然后进行梯度计算,按照式(3)和

式(4)分别计算 x 方向梯度与 y 方向梯度,每个方向的梯度计算过程为:参考具有中心对称的二维卷积核可以分解为两个互相垂直的一维卷积,即将先计算出的行卷积结果与其方向垂直的一维卷积核进行卷积,得到二维卷积结果。因此,在实现每个方向的梯度计算时,将再次划分为横向与纵向的卷积计算。

在梯度计算完成后,根据高低阈值以及梯度计算后的幅值与方向来标记边缘,根据横轴、纵轴以及对角线划分8个方向区域,将每个像素计算出的幅值变化方向 $\alpha(x,y)$ 对应到方向角度为 $[0.0^\circ, 22.5^\circ]$ 、 $[22.5^\circ, 67.5^\circ]$ 和 $[67.5^\circ, 90.0^\circ]$ 。若 α 值在 $[0.0^\circ, 22.5^\circ]$ 范围内,则对比区域为左右邻域,若在 $[22.5^\circ, 67.5^\circ]$ 范围内,则对比区域为对角线所有邻域,若在 $[67.5^\circ, 90.0^\circ]$ 范围内,则对比区域为上下邻域。对比获得的结果值,若其为上下、左右、对角线邻域内的最大幅值,则为边缘点。

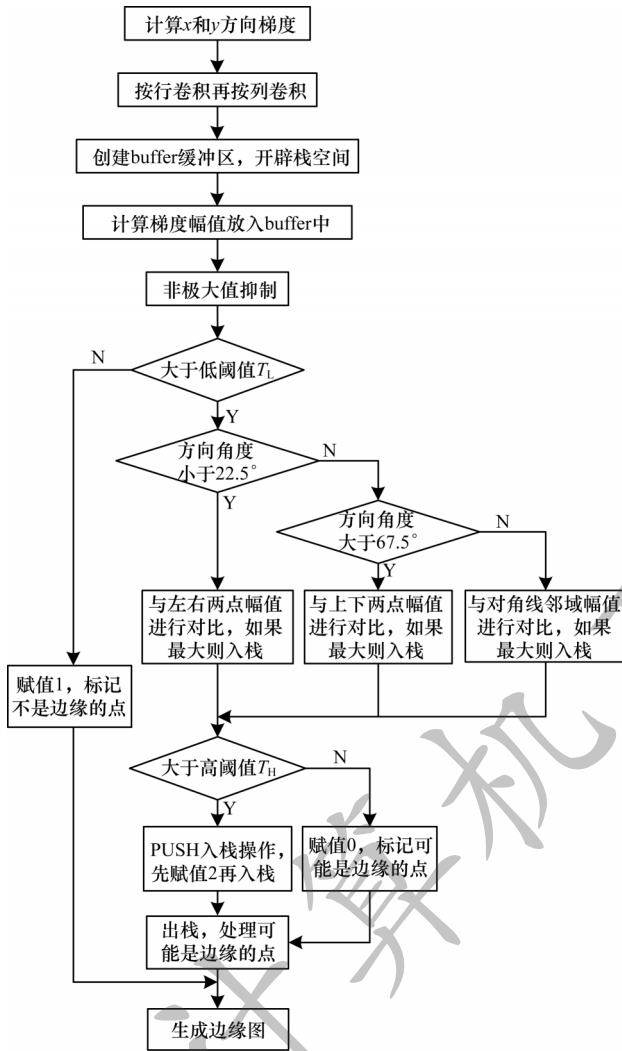


图4 基于飞腾平台的Canny边缘检测算法实现流程

Fig.4 The implementation procedure of Canny edge detection algorithm based on FT platform

在处理每个像素点时,判断高低阈值以及梯度幅值,根据不同分支情况对图片进行标记入栈操作来区分是否为边缘点。标记分为边缘点、可能是边缘点、不是边缘点等3种情况,3种情况对应的入栈赋值为2、0、1。当所有像素处理完毕后,针对栈中值为0,即可能是边缘点的情况进行单独处理,查看该像素周围是否含有边缘,若有则标记为边缘点。

2.2 Canny边缘检测算法的优化

针对Canny边缘检测算法的实现过程进行各部分计算时间的分析,如图5所示,根据常见的3种卷积核测试得到计算梯度的时间约占总时间的2/3,而剩余的1/3主要是提取边缘处理图片的计算时间。根据上述分析结果可知,梯度计算时间是Canny边缘检测算法实现过程中占用时间最多的计算过程,因此根据飞腾平台架构特点针对梯度计算进行优化。

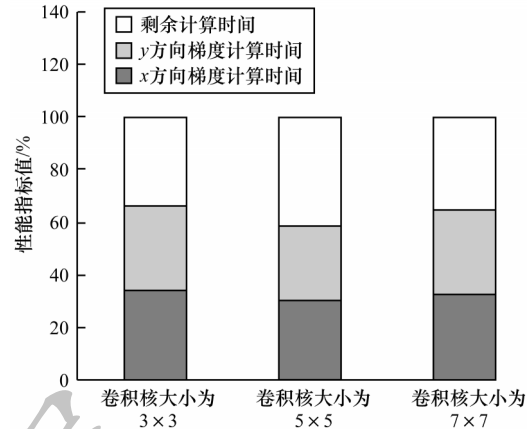


图5 Canny边缘检测算法实现过程中各部分的计算时间占比

Fig.5 The proportion of calculation time of each part in the implementation process of Canny edge detection algorithm

2.2.1 梯度计算并行化

由于梯度计算算法存在计算量大、局部数据连续访问和计算过程重复等特点,而飞腾平台具有长向量位支持特性,因此可以利用SIMD对其进行并行化处理。在将图片读入转换为矩阵时,默认图片矩阵数据类型为8位数据变量类型(char),但是在计算梯度时卷积核为浮点数,即32f,因此利用飞腾内嵌的混洗指令、移位指令和打包指令封装生成数据转换并行接口,以便调用实现8u数据类型变量图片读入与32f类型数据计算,同时增加了int转为short类型的并行接口。在解决数据类型问题后,利用其他内嵌指令进行并行化,具体实现的核心代码如算法1所示,首先使用图片vec_ld指令分别取出3行连续的16个数据,然后利用向量乘加指令进行计算。在梯度计算完成后的计算过程中使用的数据类型为半字类型,因此通过vec_fdstru指令将计算结果的数据类型由float转为int类型,这样就可以使用int_to_short并行接口将数据类型由float转为short类型。同时,由于float类型数据转换为short类型数据,在理论上内存空间要缩减一半,因此并行接口每转换一次short类型的数据变量,float类型数据需计算两次(即循环展开一次)才可以获得一次的转换数据量。

算法1 Canny梯度计算并行算法

```

1.M7002_vector float s0 = vec_ld(i,src_0);
2.M7002_vector float s1 = vec_ld(i,src_1);
3.M7002_vector float s2 = vec_ld(i,src_2);
4.M7002_vector float x0 = vec_muli(s0,f0);
5.s0 = vec_add(x0,d4);
6.M7002_vector float x1 = vec_add(s1,s2);
7.M7002_vector float x2 = vec_muli(x1,f1);
8.s0 = vec_add(s0,x2);
9.M7002_vector int st = vec_fdstru(s0);
10.M7002_vector float s00 = vec_ld(i+16,src_0);
  
```

```

11.M7002_vector float s10 = vec_ld(i+16,src_1);
12.M7002_vector float s20 = vec_ld(i+16,src_2);
13.M7002_vector float x00 = vec_muli(s00,f0);
14.s00 = vec_add(x00,d4);
15.M7002_vector float x10 = vec_add(s10,s20);
16.M7002_vector float x20 = vec_muli(x10,f1);
17.s00 = vec_add(s00,x20);
18.M7002_vector int st0 = vec_fdstru(s00);
19.M7002_vector unsigned int tt = int_to_short((M7002_
vector unsigned int)st,(M7002_vector unsigned int)st0);

```

2.2.2 梯度计算存储优化

梯度计算分为横向梯度与纵向梯度计算。横向梯度是连续访问的,其左右偏移也为固定通道数。纵向梯度因为不是不同行之间的访问计算,每次计算均需跨行访问,因此数据访问是不连续的。

根据飞腾平台架构分为标量存储空间与向量存储空间,在每次进行SIMD计算前需要将数据传输到向量存储空间,利用DMA数据传输机制并结合双缓冲的方式传输数据,即将数据划分为两个区域,一部分用来计算,另一部分用来存取下一次计算所需的数据。在每次计算横向梯度时,传输需要计算的一行数据,以便减少数据传输次数,达到数据重用的目的。在计算纵向数据时,使用SIMD指令取值是通过定义的首地址进行偏移取值,在实际计算过程中按照矩阵跨行访问的方式,每次计算结果得到一行数据,共需跨卷积核乘列的内存单元。因此,在计算纵向数据时需根据数据访问特性,设置固定地址起始存储计算涉及的行数数据来减少数据访问的不连续性问题,并使用DMA数据传输将其在计算之前存储到固定地址,达到连续访问的效果。

2.2.3 其他优化

本文结合Canny并行梯度算法的实现程序,深入了解飞腾的编译工具链对核心计算部分代码进行函数内联、消除冗余代码段、删除不使用的参数、指令重排等编译优化。使用循环展开和软流水使得指令重新排序,解决程序指令处理时隙问题,并利用循环置换及不变量外提减少计算过程中数据的访问次数,增加数据重用性,提升程序执行性能。

3 实验结果与分析

3.1 实验环境

实验平台为FT-M7002的单个DSP内核,使用680像素×480像素的经典边缘检测测试图片,评价指标为运行时间。实验主要对卷积核大小为3×3、5×5、7×7情况下Canny边缘检测算法优化前后的性能进行对比分析。

3.2 Canny算法优化前后性能对比

Canny边缘检测算法经过横向梯度计算优化后的性能如表1所示。当卷积核大小为3×3时,x方向梯度与y方向梯度计算的加速效果为1.316与1.346;当卷积核大小为5×5时,x方向梯度与y方向梯度计算的加速比为1.430与1.459;当卷积核大小为7×7时,x方向梯度与y方向梯度计算的加速比为1.512与1.540。实验数据表明,x方向梯度与y方向梯度计算在相同实验环境下的加速效果相差不大,随着卷积核尺寸的增大,加速效果也随之增加,但是该加速效果与预期加速效果不太符合,根据飞腾平台架构的512位SIMD计算同时叠加其他优化方式,应该达到更好的加速效果。这是因为在计算过程中,虽然数据是可重用的并且传输数据是双缓冲的,但是数据传输仍然需要占用时间,同时在整个计算过程中的外层循环控制过程的计算较复杂,使用向量部件也需要消耗计算时间,因此加速效果不明显,而只有计算量足够大才可能忽略向量部件的计算时间消耗,达到预期的加速效果。

在表1中,在卷积核大小为3×3、5×5与7×7时,Canny横向梯度计算优化之后算法的整体加速效果分别为1.138、1.173、1.221。可以看出,整体的加速效果要比单独计算x方向梯度与y方向梯度差,由算法整体计算时间分布可以看出,x方向梯度与y方向梯度计算约占整体计算时间的2/3,并且本文主要针对梯度计算进行优化,因此剩余约1/3的计算时间优化较少,整体加速效果不明显。在结合纵向梯度计算优化后,由表2可以看出,在不同卷积核尺寸下,x方向梯度与y方向梯度计算加速效果提升明显,这充分证明了数据量与加速效果在飞腾并行处理部件存储空间可承受范围内是成正比的。

根据飞腾平台的周期置换时间公式将周期乘以16再除以 10^6 可置换为ms,基于横向梯度计算优化、横向与纵向梯度计算优化后与整体优化的Canny边缘检测算法运行时间对比结果如图6所示。可以看出,在计算优化前卷积核大小为7×7的运行时间几乎为3×3的2倍,但是优化后的运行时间相差较小,这充分证明了Canny边缘检测算法本身不容易优化的计算基数所占的运行时间是固定的,在卷积核大小为3×3时可优化的计算时间占比较小,因此整体优化效果不明显,而卷积核大小为7×7时的计算量大,因此优化效果最明显。

表1 横向梯度计算优化前后的Canny算法性能对比

Table 1 Performance comparison of Canny algorithm before and after transverse gradient calculation optimization

性能指标	卷积核大小为3×3			卷积核大小为5×5			卷积核大小为7×7		
	x方向 梯度计算	y方向 梯度计算	整体计算	x方向 梯度计算	y方向 梯度计算	整体计算	x方向 梯度计算	y方向 梯度计算	整体计算
优化前的运行时间/周期	11 903 206	11 204 854	34 940 277	15 921 106	15 216 117	52 954 108	19 918 760	19 220 845	60 682 281
优化后的运行时间/周期	9 043 067	8 325 368	30 708 887	11 126 942	10 425 893	45 151 206	13 176 340	12 481 337	49 684 714
加速比	1.316	1.346	1.138	1.430	1.459	1.173	1.512	1.540	1.221

表2 横向梯度计算与纵向梯度计算优化前后的Canny算法性能对比

Table 2 Performance comparison of Canny algorithm before and after transverse gradient calculation and vertical gradient calculation optimization

性能指标	卷积核大小为3×3			卷积核大小为5×5			卷积核大小为7×7		
	x方向 梯度计算	y方向 梯度计算	整体计算	x方向 梯度计算	y方向 梯度计算	整体计算	x方向 梯度计算	y方向 梯度计算	整体计算
优化前的运行时间/周期	11 903 206	11 204 854	34 940 277	15 921 106	15 216 117	52 954 108	19 918 760	19 220 845	60 682 281
优化后的运行时间/周期	5 824 170	5 846 273	23 458 856	6 322 591	6 166 031	34 550 292	7 025 007	6 656 267	28 729 384
加速比	2.044	1.917	1.490	2.518	2.468	1.533	2.835	2.888	2.112

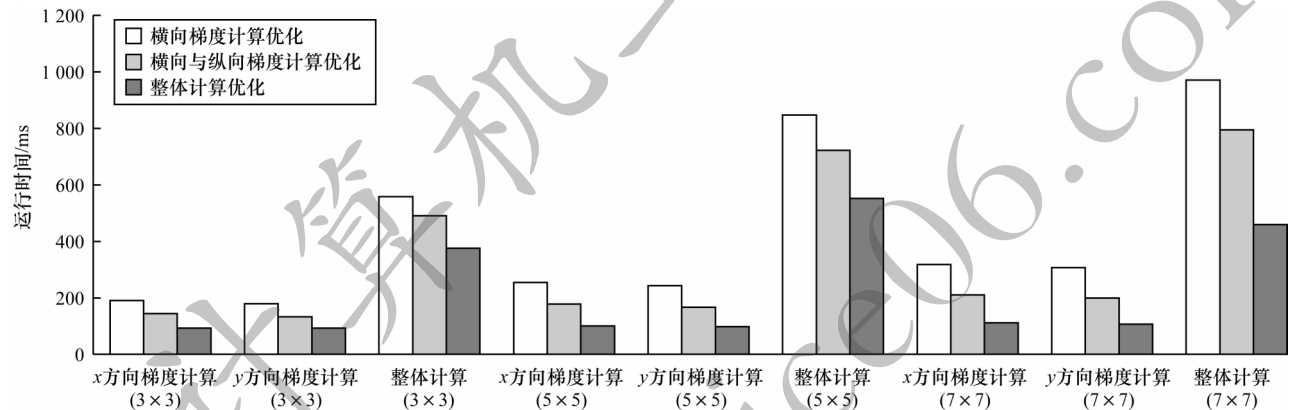


图6 基于3种计算优化方式的Canny算法运行时间对比

Fig.6 Comparison of running time of Canny algorithm based on three calculation optimization modes

4 结束语

本文基于FT-M7002高性能处理架构,提出Canny梯度计算并行算法,采用手工向量化、访存优化、DMA数据双缓冲、循环展开和软件流水等技术,提升该算法在FT-M7002平台下的并行性与运行效率。实验结果表明,与原始Canny边缘检测算法相比,该算法整体运行速度提升了1.490~2.112倍,缩小了与主流加速器件在图像处理领域的性能差距。然而由于本文研究仅局限于单个FT-DSP核上Canny边缘检测算法的优化,因此后续将进一步提升其在多DSP核硬件平台架构中的运行速度,增强Canny梯度并行算法在国产高性能DSP中的适用性。

参考文献

[1] 李飞,宾洋,罗文广,等. 基于TMSDM6437的车道线检测[J]. 广西科技大学学报,2018,29(1):43-48.
LI F, BIN Y, LUO W G, et al. Lane line detection based on TMSDM6437[J]. Journal of Guangxi University of Science and Technology, 2018, 29(1):43-48. (in Chinese)

[2] 卢俊,张保明,黄薇,等. 基于GPU的遥感影像数据融合IHS变换算法[J]. 计算机工程,2009,35(7):261-263.
LU J, ZHANG B M, HUANG W, et al. IHS transform algorithm of remote sensing image data fusion based on GPU[J]. Computer Engineering, 2009, 35(7):261-263. (in Chinese)
[3] LIU F, GUAN G X, ZHAO H M, et al. Fast processing of UAV remote sensing image based on DSP hardware simulation platform [EB/OL]. [2020-10-04]. https://www.researchgate.net/publication/340737558_Fast_Processing_of_UAV_Remote_Sensing_Image_Based_on_DSP_Hardware_Simulation_Platform.
[4] OKAMOTO T, KOIDE T, YOSHIDA S, et al. Implementation of computer-aided diagnosis system on customizable DSP core for colorectal endoscopic images with CNN features and SVM [C]//Proceedings of the 28th International Conference on Microelectronics. Washington D. C., USA: IEEE Press, 2019: 1663-1666.
[5] AGGARWAL V, GUPTA A. Integrating morphological edge detection and mutual information for nonrigid registration of medical images [J]. Current Medical Imaging Reviews, 2019, 15(3):292-300.

- [6] ZHOU X L, XU L M, WANG J. Road crack edge detection based on wavelet transform [EB/OL]. [2020-10-04]. <https://iopscience.iop.org/article/10.1088/1755-1315/237/3/032132/pdf>.
- [7] CUI J C, TIAN K. Edge detection algorithm optimization and simulation based on machine learning method and image depth information [J]. IEEE Sensors Journal, 2020, 20(20): 11770-11777.
- [8] 宋琦. DSP芯片在实时图像处理系统中的应用分析 [J]. 电子技术与软件工程, 2015(9): 86.
SONG Q. Application analysis of DSP chip in real-time image processing system [J]. Electronic Technology and Software Engineering, 2015(9): 86. (in Chinese)
- [9] PR Newswire. CEVA announces industry's first high performance sensor hub DSP architecture [EB/OL]. [2020-10-04]. <https://www.baidu.com/link?url=GLlzb7iS0JKumoTpnYQW370As9qsRZg90UHlRtSN1-wBhtnNK9-L1dISRS8qFymXV8ZgoetG4dIDDHfuLoHrq&wd=&eqid=db5b583e000301630000000460a759c3>.
- [10] MODY M, HARIYANI H, BALAGOPALAKRISHNAN A, et al. GPU assist using DSP pre-processor [C]//Proceedings of 2020 IEEE International Conference on Electronics, Computing and Communication Technologies. Washington D. C., USA: IEEE Press, 2020: 1-4.
- [11] NAJOU M, BAHTAT M, HATIM A. VLIW DSP-based low-level instruction scheme of givens QR decomposition for real-time processing [J]. Journal of Circuits Systems and Computers, 2017, 26(9): 478-504.
- [12] KUMAR A, RAHEJA S. Edge detection using guided image filtering and enhanced ant colony optimization [J]. Procedia Computer Science, 2020, 173: 8-17.
- [13] SENGUPTA S, MITTAL N, MODI M. Improved skin lesion edge detection method using ant colony optimization [J]. Skin Research and Technology, 2019, 25(6): 846-856.
- [14] DAGAR N S, DAHIYA P K. Edge detection technique using binary particle swarm optimization [J]. Procedia Computer Science, 2020, 167: 1421-1436.
- [15] 陈超. 多阈值优化的运动图像轮廓特征提取方法 [J]. 沈阳工业大学学报, 2019, 41(3): 315-319.
CHEN C. Extraction method of contour features by multi-threshold optimization for motion images [J]. Journal of Shenyang University of Technology, 2019, 41(3): 315-319. (in Chinese)
- [16] WANG S Q, LIANG S, PENG F, et al. Image edge detection algorithm based on fuzzy set [J]. Journal of Intelligent & Fuzzy Systems, 2020, 38(4): 3557-3566.
- [17] HOU Z R. Dual threshold and edge image optimization of Canny algorithm in the road extraction from remote sensing image [J]. International Journal of Earth Sciences and Engineering, 2015, 8(1): 188-194.
- [18] WU G H, YANG D Y, CHANG C, et al. Optimizations of Canny edge detection in ghost imaging [J]. Journal of the Korean Physical Society, 2019, 75(3): 223-228.
- [19] SHI Q N, AN J C, GAGNON K K, et al. Image edge detection based on the Canny edge and the ant colony optimization algorithm [C]//Proceedings of the 12th International Congress on Image and Signal Processing. Washington D. C., USA: IEEE Press, 2019: 12-18.
- [20] 孙广辉, 扈啸, 王蕊. 基于FT-M7002的OpenCV移植与优化 [C]//第二十二届计算机工程与工艺年会暨第八届微处理器技术论坛论文集. 长沙: 中南大学出版社, 2018: 113-121.
SUN G H, HU X, WANG R. OpenCV transplantation and optimization based on FT-M7002 [C]//Proceedings of the 22nd Annual Conference of Computer Engineering and Technology and the 8th Microprocessor Technology Forum. Changsha: Central South University Press, 2018: 113-121. (in Chinese)
- [21] 宋贵环. YHFT-Matrix2编译器SIMD优化技术研究与实现 [D]. 长沙: 国防科技大学, 2014.
SONG G H. Research and realization of YHFT-Matrix2 compiler SIMD Optimization technology [D]. Changsha: National University of Defense Technology, 2014. (in Chinese)