

互相关运算在银河飞腾 DSP 上的实现及优化

杨 琳 吴家铸 扈 啸 田 希

(国防科学技术大学计算机学院 长沙 410073)

摘 要 在信号处理领域中,互相关运算对于寻找未知信号中的特性具有重要的作用。互相关常被用于图像匹配、粒子图像测速等领域。针对互相关运算计算量大而实时系统中要求较快运算速度的问题,利用银河飞腾多核 DSP(YHFT-QDSP)的 FFT 加速器完成二维 FFT 运算,在 YHFT-QDSP 上实现了互相关运算,从而提高了互相关运算在实时系统中的性能,并在此基础上针对输入数据大小的不同提出了若干优化的方法,进一步提高了互相关运算的性能。

关键词 二维 FFT,互相关,优化,银河飞腾 DSP

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.11.009

Realization and Optimization of Cross-correlation Based on YHFT-QDSP

YANG Lin WU Jia-zhu HU Xiao TIAN Xi

(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract In the field of signal processing, cross-correlation plays an important role in finding an unknown signal's feature. Cross-correlation is often used in areas such as image matching and particle image velocimetry. For large cross-correlation calculation and requirements of faster computing speed in real-time systems, in order to improve cross-correlation's performance in real-time systems, we used the FFT accelerators of YHFT-QDSP to achieve 2D-FFT. Cross-correlation was realized based on YHFT-QDSP and this paper proposed some optimizations based on the size of input to improve the performance of cross-correlation.

Keywords 2D-FFT, Cross-correlation, Optimization, YHFT-QDSP

1 互相关

在信号处理领域中,互相关是用来表示两个信号相对于时间的一个相似性度量,通常通过与一个已知信号比较来寻找未知信号中的特性,有时也称为滑动点积,在模式识别、密码分析学、检测以及图像匹配等领域中都有应用^[1]。对提高互相关算法的运算效率进行研究具有重要意义。基于 FFT 的互相关算法简化了互相关计算,极大地提高了互相关运算效率,成为了互相关算法中最为典型的算法^[2]。

对两个矩阵进行频域的互相关运算的一般流程如图 1^[3]所示。

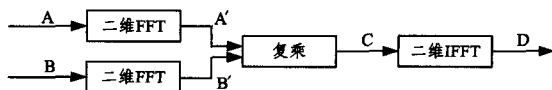


图 1 互相关流程

其中 A 、 B 、 A' 、 B' 、 C 、 D 为矩阵。 A 、 B 、 D 为单精度浮点实数矩阵, A' 、 B' 、 C 为复数矩阵。

对输入矩阵进行二维 FFT 变换,采用按行、按列进行一维 FFT 变换的方式^[4]。两个矩阵的二维 FFT 运算长度相同。二维 FFT 的运算过程如图 2 所示。

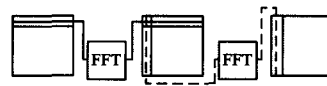


图 2 二维 FFT 过程

(1)首先逐行对矩阵进行 FFT 变换,并且将变换的结果逐行保存,如实线所示;

(2)对(1)的结果矩阵逐列进行 FFT 变换,并且将变换的结果逐列进行保存,如虚线所示。

对两个二维 FFT 结果矩阵进行复乘运算。设矩阵 A 中的元素为 $a(m,n)$, 矩阵 B 中的元素为 $b(m,n)$, 两个矩阵的大小都为 $N \times N$, A 、 B 的表达式如下:

$$a(m,n) = ax + ayi$$

其中, ax 为 $a(m,n)$ 的实部, ay 为 $a(m,n)$ 的虚部。

$$b(m,n) = bx + byi$$

其中, bx 为 $b(m,n)$ 的实部, by 为 $b(m,n)$ 的虚部。

则复乘的结果矩阵 C 的元素 $c(m,n)$ 为:

$$c(m,n) = (ax \times bx - ay \times by) + (ax \times by + ay \times bx)i$$

其中, $(ax \times bx - ay \times by)$ 为 $c(m,n)$ 的实部, $(ax \times by + ay \times bx)$ 为 $c(m,n)$ 的虚部。

二维 IFFT 类似于二维 FFT,采用 IFFT 来分别实现行列操作。

到稿日期:2014-10-02 返修日期:2014-11-13

杨 琳(1989—),男,硕士生,主要研究领域为 DSP 应用,E-mail:yanglin0507@163.com;吴家铸(1963—),男,博士,副研究员,主要研究领域为嵌入式实时系统和多核体系结构;扈 啸(1977—),男,博士,助理研究员,主要研究领域为嵌入式多核处理器系统、嵌入式处理器仿真调试、图像识别与跟踪;田 希(1986—),男,硕士,工程师,主要研究领域为 DSP 算法应用。

输出的矩阵 D 为二维 IFFT 输出结果的实数部分。

从互相关运算的流程中可以看出,整个互相关中大部分计算是二维 FFT(或 IFFT)运算,二维 FFT(或 IFFT)的运算时间成为制约互相关运算效率的主要因素。提高二维 FFT(或 IFFT)运算的效率将大大提高互相关的运算效率。

2 银河飞腾多核 DSP

银河飞腾多核 DSP(YHFT-QDSP)是一款高性能浮点多核 DSP,内含 4 个高性能 DSP 核,其应用范围涉及雷达信号处理、图像处理、电子对抗、声纳探测、通信和飞行控制等领域,具有高性能、低功耗、易于扩展等特点。

其中,可配置 FFT 加速器(简称为 FFT 加速器)主要完成不同规模的 IEEE-754 标准单精度浮点或 24 位定点 FFT 运算,可以满足各种信号处理应用需求。FFT 加速器支持复数 FFT 运算和实数 FFT 运算;支持 FFT 运算和 IFFT 运算;同时还支持实向量与复向量交叉存储或实向量与复向量分开存储两种数据存储方式。另外该加速器还能够实现批量 FFT 运算,相比于软件方法实现,FFT 效率更高。FFT 加速器通过 DDR 总线和 SMC 总线与 DDR2 和共享 SRAM 进行数据通信,通过外部总线与 DSP0123 内核进行命令通信,如图 3 所示。

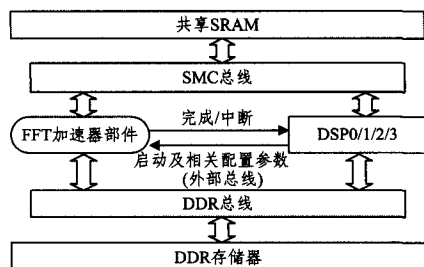


图 3 FFT 加速器交互模块

FFT 加速器接收来自 DSP 内核的参数配置及启动命令,然后从 DDR 或共享 SRAM 存储器中读取数据进行指定规模的 FFT 运算,并将结果写入到存储器的指定位置,最后通过查询方式或者中断方式返回完成信号及完成情况。FFT 加速器的核心计算模块直接支持最大规模为 1024 点的 FFT 运算,对于更大规模 FFT,需要将大规模 FFT 分解为多个小规模 FFT,通过多次调用 FFT 加速器完成。FFT 加速器可以通过配置选择 FFT 或 IFFT 运算^[5]。

使用 YHFT-QDSP 的 FFT 加速器来实现互相关运算的二维 FFT(或 IFFT)运算可以大大提高互相关运算的效率。

3 互相关在 YHFT-QDSP 上的实现过程及优化

3.1 互相关在 YHFT-QDSP 上的实现

互相关运算根据流程可以分成两部分:

- (1) 二维 FFT(或 IFFT)运算
- (2) 复乘运算

因为 YHFT-QDSP 的 FFT 加速器只能运算行 FFT,因此实现二维 FFT(或 IFFT)运算中的列 FFT 运算时,需要将矩阵在转置后做行 FFT 运算再转置得到列 FFT 运算的结果。因此在 YHFT-QDSP 上实现互相关运算只需要实现以下 3 个模块即可:

- (1) 矩阵的行 FFT(或 IFFT)运算模块
- (2) 矩阵的转置模块
- (3) 矩阵的复乘运算模块

对于矩阵的行 FFT(或 IFFT)运算部分, $N \times N$ 矩阵的行 FFT(或 IFFT)运算即是进行 N 次 N 点的 FFT(或 IFFT)运算,使用 YHFT-QDSP 的 FFT 加速器的批量 FFT 运算可以直接实现这一功能。

对于矩阵的转置运算,可将矩阵上三角的数据存放在一个中间变量中,再将下三角的数据搬移到上三角,然后将中间变量中的数据放在下三角来实现。

对于复乘部分,使用一个 N 次循环内嵌套一个 N 次循环完成对应矩阵元素的复乘。

3.2 互相关在 YHFT-QDSP 上的初始性能

对于以上互相关运算的实现方法,以输入矩阵 1024×1024 为例,测试得到其各个模块与总体的运行时间,如表 1 所列。

表 1 输入为 1024×1024 时的运算时间

模块	时间(ms)	运行次数	总时间(ms)
FFT(IFFT)(DDR 中)	19.2	6	115.2
转置	381.7	6	2290.2
复乘	3402.8	1	3402.8
总用时			5808.2

通过对互相关各模块的时间测试,可以看出限制互相关运算效率的关键是转置模块和复乘模块。因此在优化程序时,将重点对这两个模块进行优化。

YHFT-QDSP 拥有一个 3M Byte 的片内共享存储(共享 SRAM)空间和一个 512M Byte 的片外共享存储(共享 DDR)空间。当输入矩阵小于 512×512 时,可以将所有数据存放在共享 SRAM 中;当输入矩阵大于或等于 512×512 时,运算需要超过 3M Byte 的空间,因此数据只能放在 DDR 中。根据数据的存储位置的不同,优化时将有少许不同。

3.3 数据存放于片外的优化实现

(1) 矩阵行 FFT(IFFT)模块。因为 FFT 加速器的运算效率只与存储带宽有关,可以适当增加 DDR 的频率以增加 DDR 的带宽。另外增加 DDR 的带宽,数据在 DDR 中的运算效率也会提高,其他模块的运算效率也会相应提高。

(2) 矩阵转置模块。由于使用 CPU 进行矩阵的转置时,因数据量大,CPU 搬移数据速度较慢,导致转置时间较长,效率不高,因此改用其他的方法完成转置运算。

注意到 YHFT-QDSP 中 DMA 模块可以根据配置的帧索引与单元索引,对相邻两个数据单元和相邻两个帧的首数据单元进行地址跳变的读或写,因此可以使用 DMA 进行按索引值搬移,通过适当配置 DMA 的索引值,实现矩阵的转置。如使用 DMA 将矩阵 A 中第一行搬到矩阵 B 中, $A[0][0]$ 搬到 $B[0][0]$,DMA 配置中索引值配置为 B 每行首元素的地址差,则 DMA 会将 $A[0][1]$ 根据索引值间隔搬移到 $B[1][0]$ 中, $A[0][2]$ 搬移到 $B[2][0]$ 中,依此将矩阵 A 中第一行转置到矩阵 B 中第一列。以同样的方法可以将矩阵 A 全部转置到矩阵 B 中。

由于数据在共享 SRAM 中的处理速度比在 DDR 中更快,因此可以使用 DMA 将数据从 DDR 搬移并转置到共享 SRAM 中,在共享 SRAM 中做 FFT(或 IFFT)运算。因为共享 SRAM 只有 3MB 大小,而数据量较大,因此根据共享 SRAM 的大小将部分数据搬移到共享 SRAM 中,完成 FFT(或 IFFT)运算后搬出并转置回原地址,照此方法重复直到将矩阵中所有数据全部转置并完成 FFT(或 IFFT)运算。因为 DDR 按列访问时间比按行访问时间长,因此使用 DMA 转置

时在 DDR 中按行读或写,而在共享 SRAM 中按列写或读。

在进行二维 FFT(或 IFFT)运算时,先进行行 FFT(或 IFFT)变换,再进行列 FFT(或 IFFT)变换与先进行列 FFT(或 IFFT)变换,再进行行 FFT(或 IFFT)变换的结果是一样的,因此在进行二维 FFT(或 IFFT)变换时,可以只进行一次转置,然后将数据按顺序搬出。A、B 矩阵转置一次做二维 FFT 变换得到的结果是原结果的转置矩阵,A、B 复乘后得到的 C 为原 C 矩阵的转置矩阵,对 C 进行一次转置的二维 IFFT 变换,相当于先进行列 IFFT 变换,再进行行 IFFT 变换,得到的结果与原结果一致。经测试,按顺序搬出比搬出并转置用时更短。因此在将数据从共享 SRAM 中搬移到 DDR 中时,只进行搬移操作,不再进行转置操作。

转置模块实现过程:1)DMA 搬移部分数据并转置到 SRAM 中;2)在 SRAM 中对这部分数据做 FFT 运算;3)将数据直接搬移出 SRAM;4)搬移下一部分数据进行相同操作直至所有数据运算完毕。

(3)复乘模块。YHFT-QDSP 是一个多核 DSP,因此在运算复乘时,可以使用多核并行运算,用 YHFT-QDSP 的 4 个核分别并行计算 1/4 的数据,以减少复乘运算时间。

使用编译关键字来标明一个指针是指向一个特定对象的唯一指针,可以帮助编译器确定访问存储器的指令是否不相关,以便使程序的指令进行更多的并行处理,提高程序性能。

使用指令流水。指令流水使循环的多次迭代能够并行执行,提高程序性能。

(4)其他的优化方法。可以使用 Cache 对程序进行加速,但需注意应当在适当的地方对 Cache 进行写回操作以确保数据的正确性。

使用最高级别编译优化,实施如循环展开和 SIMD(单指令多数据流)等优化方法。

3.4 数据存放于片内的优化实现

数据存放在片内的优化与数据存放在片外时互相关的优化基本相同,主要不同之处如下。

由于数据的处理全是在共享 SRAM 中进行,因此在实现二维 FFT(或 IFFT)时,对转置后的矩阵做完行 FFT(或 IFFT)后,可以不将数据搬移出共享 SRAM 来给其他数据留出空间,因此可以省去在矩阵做完行 FFT(或 IFFT)后再用 DMA 搬移的步骤。因此转置模块实现过程为:1)DMA 搬移数据并转置;2)对数据做 FFT 运算。

4 互相关在 YHFT-QDSP 上的测试结果及分析

(1)当数据存放在片外时,以输入矩阵 1024×1024 为例,测得互相关运算各模块运行的时间及总时间,如表 2 所列。

表 2 输入为 1024×1024 时优化后的运算时间

模块	时间(ms)	运行次数	总时间(ms)
FFT(FFT)(DDR 中)	12	3	36
FFT(FFT)(SRAM 中)	2.9	3	8.7
DMA	10.6	3	31.8
复乘	29.8	1	29.8
总用时			106.3

优化后的总用时为 106.3ms,对比优化前的时间,可以看到优化后的性能比优化前提高了 50 多倍。优化后的效率提高十分明显。

(2)当数据存放在片内时,以输入矩阵 256×256 为例,测得互相关运算各模块运行的时间及总时间,如表 3 所列。

表 3 输入为 256×256 时优化后的运算时间

模块	时间(ms)	运行次数	总时间(ms)
FFT(FFT)(SRAM 中)	0.16	6	0.96
DMA	0.39	3	1.17
复乘	0.52	1	0.52
总用时			2.65

优化后的总用时为 2.65ms,优化前的总用时为 83.897ms。对比优化前的时间,可以看到优化后的性能比优化前提高了 30 多倍。

(3)互相关计算时间分析与评估。

互相关的时间由矩阵行 FFT(或 IFFT)时间、矩阵转置时间、复乘时间组成。

根据 YHFT-QDSP 的说明手册,对于 M 次 N 点小规模 FFT($N \leq 1024$),存储带宽为 T (B/s),执行时间为 $16M \times N / T$ (s)。对于 M 次 N 点中等规模 FFT($1024 < N \leq 65536$),存储带宽为 T_1 ,共享 SRAM 带宽为 T_2 ,执行时间为 $M \times (16N / T_1 + 32N / T_2)$ (s)。对于 M 次 N 点大规模 FFT($65536 < N \leq 1048576$),DDR 存储带宽为 T_1 ,共享 SRAM 带宽为 T_2 ,执行时间为 $M \times (32N / T_1 + 16N / T_2)$ (s)。对于 DMA 时间,可以根据搬移数据大小与带宽计算得出。以单核计算复乘时间的 1/4 作为四核并行计算复乘的理论时间。

对于不同规模的互相关,计算得到的理论用时及实际测试得到的实测时间如表 4 所列。

表 4 不同规模的互相关运算的理论时间与实测时间

计算规模	理论时间(ms)	实测时间(ms)
128×128	0.628	1.373
256×256	1.928	2.65
512×512	15.772	24.563
1024×1024	61.647	106.3
2048×2048	1297.226	2050.4
4096×4096	5445.615	9367.392

可以看到实测时间与理论时间有一定的差别。造成这一差别的原因是计算理论时间时没有包含 FFT 加速器与 DMA 的启动时间;另外在多核并行计算复乘时,多个核同时对存储器的读写竞争也会造成很大的延迟。其中读写竞争是造成实测时间与理论时间差异的主要因素。

结束语 本文在 YHFT-QDSP 上实现了互相关运算,并根据 YHFT-QDSP 的体系及存储结构,针对输入数据的规模设计实现了两种优化方案,大大提高了互相关运算在 YHFT-QDSP 上的运算效率。本文提出的若干优化方法也可以使用在其他程序的优化中。

参 考 文 献

- [1] Adiran R J. Twenty years of particle image velocimetry [J]. Experiments in Fluids, 2005, 39(2): 159-169
- [2] Adric E, Vlachos P P. Assessment of advanced windowing techniques for digital particle image velocimetry [J]. Measurement Science and Technology, 2009, 20(7): 1-9
- [3] Bao X L. An Improved PIV Cross-correlation Algorithm Based on FFT[J]. Journal of Dalian University of Technology, 2011, 51(3): 417-421
- [4] Chen L L. Contrast Research on FPGA and DSP in Application of 2D-FFT[J]. Modern Electronics Technique, 2012, 35(20): 80-83
- [5] 扈啸,李杰,陈莉丽,等.多核处理器 YHFT-QDSP 的调试系统[J].计算机工程与科学,2008,30(9):116-119
Hu X, Li J, Cheng L L, et al. The Debug Systems of the Multi-core YHFT-QDSP[J]. Computer Engineering & Science, 2008, 30(9): 116-119