DOI:10.11721/cqnuj20130319

Fermi 平台下的医学超声实时扫描转换并行算法*

何兴无[1,2], 张 霞[2]
(1. 成都师范学院 网络与信息管理中心, 成都 611130; 2. 成都农业科技职业学院 电子信息分院, 成都 611130)

摘要:在超声成像系统中扫描转换是让超声图像较好地在屏幕上进行显示所必须的处理步骤。由于这一处理步骤中存在运算复杂的插值运算,特别是在横向使用的三次方插值,使其成为临床实时成像系统中提供帧速率的一大性能提升瓶颈,为此本文研究并提出了一种基于新兴的高性能并行计算平台 Fermi 架构 GPU(Graphics processing unit)的并行处理算法,该算法基于 GPU 并行处理平台,主要包括初始化阶段、图像插值以及图像显示这 3 个处理环节。该算法不仅保持了与现有计算平台系统的计算精度,而且显著地提高了这一处理环节的计算速度。数据测试结果显示,采用 Fermi 架构的 GPU 处理在得到与基于 CPU 的实现完全一致的扫描转换效果的同时,取得了较大的加速效果。对于 3 121×936 的图像数据能够达到 1 558 fps 的帧率,速度提高了大约 664 倍。

关键词:高性能并行计算; 扫描转换;Fermi;图像并行处理算法
中图分类号:TP391.41 **文献标志码**:A **文章编号**:1672-6693(2013)03-0094-05

近年来,随着计算机技术的高速发展以及现代先进的医学影像检测技术的出现和图像工程领域应用的不断延伸,医学超声诊断技术因为其使用简单方便、对人体无侵害性、实时性、廉价性等特点和优势,在现在的医学诊断学中有着难以取代的作用[1]。在医学数字彩超系统上,实时扫描转换是一种让超声图像可视化的关键技术,也是任何一个超声成像系统必须拥有的处理模块。因为要将超声图像在显示屏上进行正常显示,就需要将数据从极坐标系转换到标准显示设备的笛卡尔坐标系。这个坐标转换需要通过求反正切来决定扫描角度和计算平方根得到沿半径方向的像素的位置[2]。虽然在现在的实时系统中已经有相关应用,但这个模块的这些核心运算对于超声这种高实时性系统来讲,仍然希望使用廉价的硬件得到更高的性能。同时扫描转换模块又是超声成像系统所有功能模式必须经过的环节。这一功能模块的性能提升将对整个系统性能的提高产生重要的影响。

如何加速扫描转换处理环节,有许多的研究小组提出了很多方法。文献[3-5]中,研究者为了消除极坐标处理方式的大量冗余计算,提出在进行插值运算前把所有极坐标计算结果存储起来,然后进行查表,以加快计算速度。还有一些研究者研究的是对于超声成像设备使用定制的数字信号处理器(DSP)[4]和现场可编程门阵列(FPGA)[6-7]。无论是扩大存储空间,还是使用定制硬件,基于硬件的方式都会使系统造价变得太高。因此,本文通过研究新近出现的具有通用计算能力的 Fermi 架构 GPU 并行处理平台,提出了一种利用 GPU 通用计算能力的超声系统实时扫描转换并行算法。

随着多核时代的到来,并行处理技术是解决实时系统计算需求问题的最有效途径之一。与多核 CPU 相比,近年来出现的具有通用计算能力的 GPU,特别是英伟达公司新近设计完成的 Fermi 架构 GPU,在通用计算方面,表现出更为明显的优势。在保持图形处理性能的前提下,Fermi 架构将通用计算技术提升到前所未有的高度。与传统的处理器不同的是,在硬件架构方面 GPU 上多数的晶体管被用来设计为计算单元,而用于指令流的控制单元较少。CUDA2.0 以上的设备都是采用 Fermi 并行处理架构的 GPU,它主要在 GPU 处理能力、显存缓存机制和计算精度等方面使 GPU 更加适合于数据的并行计算[8]。在航天航空、地理信息分析等各种应用领域中,GPU 都已经得到了成功应用,在超声成像系统中的应用才刚刚起步[9]。

CUDA 程序设计实现方式与一般的 CPU 处理是有所不同的。程序流程一般包括数据由主机端到设备端的传输、CUDA 核函数执行配置和处理结果的返回(由设备端到主机端)。CUDA 程序的设计关键是发挥并行处理平台的计算能力。在设计中一般需要考虑这几个方面:首先是存储器操作,CPU 的内存存取延迟主要是通过多级缓存来消除,而 CUDA 主要通过高

* **收稿日期**:2012-05-21 **修回日期**:2012-11-17 **网络出版时间**:2013-05-20 18:04
作者简介:何兴无,男,副教授,博士生,研究方向为超声图像处理,E-mail:hexw981@126.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20130520.1804.201303.94_018.html

度并行化和合并访问的方式来达到高的带宽利用率，即使在新一代 Fermi 架构下，让 CUDA 程序尽可能满足合并访问，也会明显地提高性能；另外一个方面是线程结构设计，要让 GPU 的占用率保持在一个比较高的水平上，并行度尽可能高；还有一个方面就是运算指令的优化，包括尽可能使用高吞吐率的运算指令和合理使用快速函数等。本文依据 Fermi 架构的这些性能特性要求设计完成了一种并行扫描转换算法。

另外，GPU 通用计算源于传统的图形学绘制管线，数据需要流经顶点着色器、光栅化器、片段着色器再到帧缓存等部分，一般由图形学的 D3D 或者 OpenGL 来完成整个渲染过程。CUDA 采用了统一的渲染架构，以通用的渲染单元代替了原先分离的顶点着色器和片段着色器。同时 CUDA 开放了图形学管线资源共享的 API，可以通过互操作将 CUDA 计算得到结果直接利用图形学的绘制接口完成显示。

本文提出的这一并行算法，是在通用计算机环境中利用 CUDA 并行处理平台，并结合新一代的并行体系架构 Fermi 实现超声成像系统实时扫描转换的并行处理算法。在下面的第 1 部分中将详细阐述本文所设计的并行处理算法。实验数据的结果及相关分析讨论放在本文的第 2 部分。最后在本文的第 3 部分给出了工作总结和对未来工作的展望。

1 基于 Fermi 的超声实时扫描转换并行处理算法

本文所研究的扫描转换算法是为了解决系统在多角度发射声束时让数据从极坐标系转换到笛卡尔坐标系，便于显示。设计的并行算法是基于 GPU 并行处理平台进行的，主要包括初始化阶段、图像插值以及图像显示这 3 个处理环节。

1.1 初始化阶段

在 GPU 上实现通用计算，首要就是将数据从主机端传送到设备端，在主机端和设备端均有多种具有不同性能特性的存储器。因此在实现设计时需要选择合理的使用策略，最大化提高带宽利用率，最小化数据在主机与设备间传输的开销。在设备端主要有 4 种具有不同性能特性的存储器：全局存储器、常量存储器、纹理存储器和共享存储器。其中常量存储器和纹理存储器对于一个核函数的所有线程是只读的。在 Fermi 架构下全局存储器拥有缓存机制，而共享存储器与 L1 缓存共享同一个大小为 64 KB 的片内高速存储空间。在初始化阶段主要完成参数的计算、装载和图像数据的传输。由于在插值运算中数据的访存大都是随机但具有局部性的，同时考虑二维纹理存储器一次可使用的空间大小满足图像数据规模要求。对于边界区域，由于纹理存储器提供了自动滤波线性填值功能，可以减少运算次数。因此，在设备端选择用二维纹理存储器来绑定显存中的图像数据。与此同时，因为常量存储器具有当相邻两个 warp 线程块同时访问时达到最大化带宽的性能特性，所以可以将算法中用到的参数放入其中，既达到了最优的常量存储器带宽利用率，又节约了寄存器空间，让核函数执行的并行度更高。

另一方面在主机端，主要进行处理的数据是每一帧扫描转换前的图像数据。CUDA 平台上为内存设置了两种类型的存储器：分页内存和页锁定内存。页锁定内存的传输带宽会比分页内存高出 2 倍多。页锁定的内存一共提供了 4 种工作模式。其中，写结合模式可以使数据在通过 PCI-e 总线传输时不会被监视，并能获得高达 40％的传输加速，最适合 CPU 是只写的情况。由于这里的输入图像数据对于 CPU 而言是只写的，因此可以使用页锁定内存中的这种写结合模式。需要说明的是，如果使用的 GPU 平台为集成显卡，采用零拷贝模式直接实现内存与显存地址映射的方式会更好[10]。具体的实现如下：

1）图像数据传输与纹理绑定。使用 cudaHostAlloc 分配主机端页锁定内存，设置写结合模式 cudaHostAllocWriteCombined。

设备端使用 cudaMallocArray 分配 CUDA Array 存储空间存放图像数据。这是专用于纹理拾取而优化的显存存储空间。

传输到设备端后，利用 cudaBindTexture 绑定到物理存储器。注意在绑定前需要设置纹理通道参数来决定纹理拾取的工作条件。

2）参数数据传输。使用 cudaMemcpyToSymbol 将系统参数由主机端传入常量存储器。这里为了让 Cache 命中，将下一步扫描转换所需要的参数都打包为一个结构，这样就只占用 1 行 Cache。

1.2 插值计算（轴向的线性插值和横向的三次方插值）

1）轴向的线性插值。在扫描转换处理中，首先计算沿横向方向每一条声束的角度。沿着每一条声束的径向网格像素是均匀空间，任意相邻两个像素间的距离是一个常数，只与声束有关。基于这种特性，径向网格的像素值可以通过沿声束方向的线性插值运算得到[11]。

扫描转换算法的处理是从两边最外面的声束开始，在第 k 条声束的新的像素值 $V[i][k]$ 可以通过使用采样点数为 P 的线性插值得到，其中 i 为深度方向的位置索引。对于插值，需要计算决定新采样点对于原先位置的偏移量，计算过程如下

$$r=\lfloor D_a-R_p \rfloor \tag{1}$$

$$offset=(D_a-R_p)-r \tag{2}$$

其中，D_a 是第 k 条声束轴向的从中心点到当前目标图像位置的距离，R_p 是探头半径。r 是当前目标点的到

这条声束第一个点的距离。由此就可以定义插值后的输出

$$V[i][k]=offset*P[r][k]+(1-offset)*P[r+1][k] \tag{3}$$

2) 横向的三次方插值。类似于轴向的情况,在横向使用另一个一维重采样滤波器去产生与图像横向方向一致的显示像素值。这里的插值计算核是采用一个三次方卷积的方式实现的,定义为

$$h(x)=\begin{cases}(\alpha+2)|x|^3-(\alpha+3)|x|^2+1, & |x|<1\\ \alpha|x|^3-5\alpha|x|^2+8\alpha|x|-4\alpha, & 1\leqslant|x|<2\\ 0, & \text{否则}\end{cases} \tag{4}$$

其中,α 是一个常数,一般可取值为-1,-0.75 等。如图1所示,新的样本值是通过它周围4个相邻点插值得到的。具体做法是先计算偏移量,然后通过三次方插值核计算4个相邻点的和。

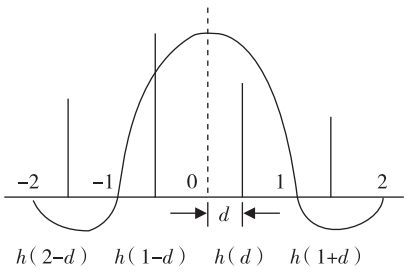

图1 三次方插值核心部分的示意图

三次方插值计算从图像水平线两端开始向中心进行,从较浅的深度连续到较深的深度。插值用到的第一个最左边的像素是通过第 k 条声速与中心线的角度 θ 来得到的。在这种情况下,中心与第一个插值像素之间的宽度就为 $z*\tan\theta$,其中 z 是深度。三次方插值是通过每一个深度完成的,因此需要通过计算决定第一个需要插值的像素点的 x 坐标,计算方式为

$$\theta_x=halfLine-\frac{\theta}{\theta_0} \tag{5}$$

其中,θ_0 是最边上的声束与最中心声束的夹角,$xdim$ 是数据的横向宽度。$halfLine$ 表示方向角范围内一半的声束数目。

由对称原理知道,插值从最右边开始的计算公式应为

$$\theta_x=halfLine+\frac{\theta}{\theta_0} \tag{6}$$

为了确定新采样点相对于旧采样点的位置偏移,新采样点到中心线的距离 x,可以通过沿水平方向每两条声束间的距离得到,并且进而得到偏移量

$$x=\lfloor\theta_x\rfloor \tag{7}$$

$$offset=\theta_x-x \tag{8}$$

三次方插值的输出像素值为

$$\begin{aligned}Vnew[i][x]=\ & h(offset)\ V[i][x]+\\ & h(offset+1)\ V[i][x-1]+\\ & h(1-offset)\ V[i][x+1]+\\ & h(2-offset)\ V[i][x+2]\end{aligned} \tag{9}$$

由原算法的运算过程可以看到,扫描转换的处理具有计算无关性,可以直接并行化。为了提高GPU的占用率,本文采用一种细粒度的并行处理方式。这里的细粒度并行方式是让一个线程负责计算一个目标图像的像素值,同时完成坐标转换和插值计算。

为了在图像数据地址计算时避免整数求商求模这样的低吞吐率的运算指令,笔者设计了一种二维网格和二维线程块的线程结构,定义为

$$B_x=\left\lceil\frac{width}{T_{xDim}}\right\rceil \tag{10}$$

$$B_y=\left\lceil\frac{height}{T_{yDim}}\right\rceil \tag{11}$$

其中 B_x,B_y 分别表示栅格的 x 方向和 y 方向上的线程块数目。T_{xDim} 和 T_{yDim} 分别表示同一个线程块内部 x 方向和 y 方向上线程数目,在Fermi架构下 T_{xDim} 为32的倍数为宜,本文设置为32。$Width$ 和 $height$ 分别为目标输出图像的宽度和高度。

在这个计算过程中,存在大量运算指令很耗时的计算函数,比如平方根函数、反正切函数等,因此需要保持精度的情况下最小化使用这些低吞吐率的数学函数。

1.3 图像显示

经过扫描转换处理后的超声图像显示数据存放在显存上。要进行显示,既可以写回主机端,然后进入图形学管线进行绘制,又可以直接利用CUDA技术与图形学管线的互操作接口如D3D(Direct 3-Dimension)进行资源共享。采用后者的方式,可以将显存中存放的数据直接写入到图形学纹理渲染器中进行图像显示,这就避免内存和显存进行数据传输的时间消耗。

通过CUDA与D3D的互操作,D3D资源可以被映射到CUDA空间,这样D3D设备就可以访问到由CUDA计算生成的图像数据。在CUDA平台上实现这一功能,主要包含初始化D3D设备,D3D资源在CUDA上注册,D3D资源映射到CUDA,体素渲染和资源释放。具体实现如下:

第1步,初始化。在任何运行时CUDA函数执行之前,必须首先初始化D3D设备并将其指定在一个CUDA上下文中,并且CUDA上下文和D3D设备都必须是在同一GPU上创建的。D3D的体素空间和属性也需要被设置。

第2步,资源注册。在使用D3D和CUDA互操作之前,必须将D3D资源注册到CUDA上。这个注册函数的执行非常耗时,因此一般在实际应用时对于每一个D3D设备资源仅注册一次,但可以多次映射不同的数据。

第3步,资源映射。当D3D资源注册成功后,就

可以使用 API 实现资源映射，让 CUDA 核函数可以访问体素空间。

第 4 步，渲染。D3D 使用由 CUDA 核函数写入到体素空间的数据，利用图形学管线进行显示渲染。渲染过程与传统图形学渲染过程完全一致。

第 5 步，资源释放。在实际超声成像系统中进行显示更新时，仅仅需要让核函数更新体素空间的经扫描转换处理后的图像数据，再重复执行步骤 4 的渲染过程。而在显示结束时，就需要先解映射后从当前 CUDA 上下文中释放 D3D 资源。

在实际应用本算法时，步骤 1～3，在程序初始化阶段执行一次，步骤 5 在程序结束时进行。扫描转换所使用的核函数在步骤 4 的绘制管线中更新体素空间用于显示实时图像序列。

2 实验结果与分析

本文实验平台为 2.81 GHz 的 AMD Althlon(tm) II X2 240，2GB DDR2 RAM，操作系统为 Windows 7。Fermi 架构的 GPU 为 NVIDIA Geforce GTX 560 Ti，显存为 1GB，核心频率 1.645 GHz，14 个多处理器，使用 4.0 版本的 CUDA Toolkit 及对应的开发包。编程环境为 Visual Studio 2010。同时，采用 GTS250 作为非 Fermi 架构的 GPU 进行对比，GTS 250 拥有 1GB 的显存，核心频率为 1.8 GHz，16 个多处理器。

为了测试本文提出的并行实现算法的处理效果和运行效率，本文使用由数字超声扫描器在超声系统 iMago C21 上采集得到的人体数据作为本文研究工作的实验数据。图 2(a)显示的是由 3.5 MHz 凸阵扫描器采集得到的经过中端处理人体肝脏组织未经扫描转换的超声图像。图 2(b)和图 2(c)分别是扫描转换后 CPU 和 GPU 的处理结果。通过做差运算知道 CPU 和 GPU 的处理结果完全保持一致，图像像素值的误差为 0。

使用 CUDA 与 D3D 互操作的渲染方式和传统的渲染管线是一样的，区别在于 CUDA 使用统一的着色器生成了用于渲染的纹理。而传统基于 CPU 的绘制管线，需要在这个地方把纹理从内存读入显存。GPU 或 CPU 分别在插值运算和像素渲染运算这两个阶段的计算性能比较见表 1。可以看到，直接由 GPU 完成计算与渲染因为减少了数据的 PCI-E 传输，性能有明显提升，表中选用数据尺寸为 3 121×936。

由于 D3D 渲染绘制管线是成像必须的过程，因此本文的重点在扫描转换的核心处理插值计算部分的计算速度。表 2 给出了本文所研究的扫描转换算法 CPU 和 Fermi 架构和非 Fermi 架构 GPU 处理的性能比较。程序的运行时间是图像数据进入显存后进行扫描转换处理的运算时间。从表 2 可以明显看出，本文所提出的基于 Fermi 架构 GPU 并行处理算法相比较于传统的 CPU 串行处理提高了大约 664 倍。加速比也说明了数据计算规模越密集，GPU 并行加速效果越明显。另一方面 与非 Fermi 架构的具有相近显存空间、多处理器个数与核心频率的 GPU 相比较，Fermi 架构的 GPU 显示出了更强大的计算能力，而且计算越密集，GPU 计算资源利用情况越好。

表 1　算法处理环节性能比较

计算环节	CPU 运行时间/ms	GTS 250 运行时间/ms	GTX 560 运行时间/ms	加速比
扫描转换	426.37	2.087	0.642	664.12
像素渲染	28.67	15.65	12.33	2.35
总时间	455.04	17.737	12.972	35.08

(a) 超声 B 模式扫描转换前的图像

(b) CPU 处理的结果

(c) GPU 处理的结果

图 2　超声系统扫描转换前后的超声图像

表 2 扫描转换性能比较

尺寸大小	CPU 运行时间/ms	GTS 250 运行时间/ms	GTX 560 运行时间/ms	加速比
789×540	54.42	0.371	0.119	457.31
1 249×618	108.11	0.735	0.231	468.01
3 121×936	426.37	2.087	0.642	664.12

3 结语

本文提出了一种基于 Fermi 架构 GPU 的超声实时扫描转换并行实现方法，实验结果显示了用本文所提出的并行实现方法得到的显示图像和通用 CPU 处理的结果保持完全一致，而在时间性能方面取得了很好的提升效果，得到大约 664 多倍的加速效果。在这样的 GPU 处理效率下，超声成像系统中的各种模式下实时扫描转换帧速率大幅度提高，不仅保证了处理质量，而且使超声系统上可以应用更多的高级处理功能。

参考文献：

[1] 李治安. 临床超声影像学[M]. 北京：人民卫生出版社，2003.
Li Z A. Clinical ultrasound imaging[M]. Beijing: The People's Medical Publishing House, 2003.

[2] 刘广东，陈阿林. 基于 curvelet 变换的图象去噪[J]. 重庆师范大学学报：自然科学版，2009，26(4)：1-4.
Liu G D, Chen A L. Image denoising based on curvelet transform[J]. Journal of Chongqing Normal University: Nature Science, 2009, 26(4): 1-4.

[3] HouL L, Liu D C. A filtering based scan conversion method in ultrasound system[C]//Sun L K. Proceeding of world congress on medical physics and biomedical engineering. Seoul: FMBE, 2006: 3494-3497.

[4] Berkhoff A, Huisman H, Thijssen J, et al. Fast scan conversion algorithms for displaying ultrasound sectorimages[J]. Ultrasonic Imaging, 1994, 16(3): 87-108.

[5] Lee M, Jim J, Park S. Analysis of a scan conversion algorithm for a real-time sector scanner[J]. IEEE Trans Med Imaging, 1986, 5(2): 96-105.

[6] Kassem A, Sawan M, Boukadoum M. A new digital scan conversion architecture for ultrasonic imaging system[J]. Journal of Circuits, Systems and Computers, 2005, 14(2): 367-382.

[7] Chang J H, Yen J T, Shung K K. High-speed digital scan converter for high-frequency[J]. Ultrasoundsector Scanners Ultrasonics, 2008, 48(5): 444-52.

[8] NVIDIA Corporation. CUDA programming guide4. 0[S/OL]. (2012-7-3)[2011-05-06]. http:// www. nvdia. com.

[9] 夏春兰，石丹，刘东权. 基于 CUDA 的超声 B 模式成像[J]. 计算机应用研究，2011，28(6)：2011-2015.
Xia C L, Shi D. Liu D Q. Ultrsound B-mode image on CUDA[J]. Application Research of Computers, 2011, 28(6): 2011-2015.

[10] NVIDIA Coporation. CUDA API reference manual 4. 0[M]. California: Santa Clara University, 2011.

[11] Richard W D, Arthur R M. Real-time ultrasonic scan conversion via linear interpolation of oversampled vectors[J]. Ultrason Imaging, 1994, 16: 109-23.

A Parallel Algorithm of Real-time Scan Conversion for Medical Ultrasound Imaging on Fermi

HE Xing-wu[1,2] ZHANG Xia[2]

(1. Network & Information Management Center, Chengdu Normal University, Chengdu 611130;
2. Department of Electronics and Information, Chengdu Vocational College of Agricultural Science and Technology, Chengdu 611130, China)

Abstract: In the ultrasound imaging system, scan conversion is a necessary step for better imaging display quality on the monitor. Because of the massive computation involved in this filter technique, especially for the cubic interpolation along the lateral direction, it has been the bottleneck for the clinical real-time imaging system. However, in this paper, a new parallel algorithm of scan conversion based on Fermi GPU (graphics processing unit) is presented. It contains three main procedures on GPU platform, that is, initialization, image interpolation and image display. This algorithm not only keeps the precision with one of CPU, but also significantly accelerates the scan conversion function. Test results show the output of graphics processing unit (GPU) is definitely the same as one of CPU, and also demonstrate the obvious speedup using GPU, that is, it can achieve 1558 fps for the image size (3121×936) which is about 664 times faster than the CPU implementation.

Key words: high performance parallel processing; scan conversion; Fermi; parallel algorithm for image processing

(责任编辑 游中胜)