(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101606854 B
(45)授权公告日 2010.11.17

(21)申请号 200910033262.7

(22)申请日 2009.06.10

(73)专利权人 无锡祥生科技有限公司
地址 214142 江苏省无锡市新区硕放镇香楠路 8 号

(72)发明人 赵明昌 莫善珏

(74)专利代理机构 无锡市大为专利商标事务所 32104
代理人 曹祖良

(51)Int.Cl.
A61B 8/14(2006.01)
A61B 8/00(2006.01)
G06F 19/00(2006.01)

审查员 胡亚婷

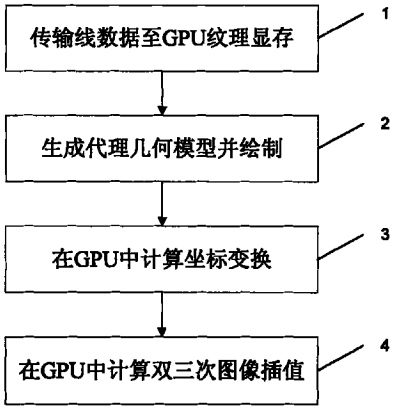
权利要求书 2 页 说明书 8 页 附图 3 页

(54)发明名称

一种高精度实时超声图像扫描变换方法

(57)摘要

本发明公开了一种高精度实时超声图像扫描变换方法，使用基于 GPU 的并行计算架构和 SIMD 矢量运算，通过绘制合适的代理几何模型，将扫描变换模块中的坐标变换和图像插值算法全部放在 GPU 中作 32 位浮点数高精度计算，同时使用优化的高精度的双三次插值算法，获得更平滑的图像质量。本发明可以在目前普通中低档显卡上，对 512 条线数据，每条线上 1024 个采样点这样大规模的数据，达到每秒钟 60 帧以上的扫描变换速度，可以充分挖掘利用当代 GPU 所提供的强大的并行计算能力，从而将传统的扫描变换模块升级为下一代高保真的扫描变换模块。



CN 101606854 B

1. 一种高精度实时超声图像扫描变换方法，其特征在于，步骤如下：

步骤 1：获取超声探头所采集的每帧图像，由 LN 条线数据组成，形成一个扇形，内径为 r，外径为 r+1，第 1 条线数据和第 LN 条线数据所夹的圆心角为 2θ，LN 条线数据为等角度采样，每条采样线上有 PN 个采样点，为等距离采样；将所述每帧图像作为二维纹理传输至 GPU 显存，并保存为线数据纹理；

步骤 2：生成代理几何模型并绘制，包括：

使用六边形 ABQRHG 作为代理几何模型，其顶点坐标分别为：

$(A_x, A_y) = (-r\sin\theta, r\cos\theta)$

$(B_x, B_y) = (-(r+1)\sin\theta, (r+1)\cos\theta)$

$(Q_x, Q_y) = (-(r+1)\sin\theta, r+1)$

$(R_x, R_y) = ((r+1)\sin\theta, r+1)$

$(H_x, H_y) = ((r+1)\sin\theta, (r+1)\cos\theta)$

$(G_x, G_y) = (r\sin\theta, r\cos\theta)$

其中 A 和 G、B 和 H、Q 和 R 的 x 坐标相反，y 坐标相同，因此只需计算 A、B、Q 三个点的坐标，其它三个点的坐标根据上述关系求得；

将六边形 ABQRHG 使用三角带形式分解为四个三角形，顶点顺序为 G，H，A，R，B，Q，三角形分别为：GHA、HAR、ARB 和 RBQ；

使用 OpenGL 或者 DirectX 的 API 函数绘制三角带；

步骤 3：在 GPU 中对代理模型绘制后的每个光栅点计算坐标变换，包括：

将代理模型绘制后的每个光栅点从显示器的直角坐标 (x，y) 转换为所述线数据纹理所在的极坐标 (s，t)，计算公式为：$s = (\sqrt{x^2 + y^2} - r)(PN-1)/l$，$t = (\arctan(x/y) + \theta)(LN-1)/2\theta$，其中 t 代表在第 t 条线数据上，s 代表在线数据的第 s 个采样点上；

判断 (s，t) 是否满足 $0 \leqslant t \leqslant (LN-1)$ 且 $0 \leqslant s \leqslant (PN-1)$，如果不满足则终止计算；

步骤 4：在 GPU 中计算双三次图像插值，包括：

在 $-2 < u < 2$ 的区间上取 NX 个离散的三次多项式 f(u) 的非零值，NX 为 2 的正整数次方，用一维纹理传输至 GPU 显存，并保存为三次多项式纹理，其中 f(u) 的定义如下：

$$f(u) = \begin{cases} \frac{7}{6}|u|^3 - 2|u|^2 + \frac{8}{9}, & |u| < 1 \\ -\frac{7}{18}|u|^3 + 2|u|^2 - \frac{10}{3}|u| + \frac{16}{9}, & 1 \leq |u| < 2 \\ 0, & |u| \geq 2 \end{cases};$$

分别使用纹理坐标 (floor(s)-1，floor(t)-1)，(floor(s)，floor(t)-1)，(ceil(s)，floor(t)-1)，(ceil(s)+1，floor(t)-1) 访问所述线数据纹理，得到 4 个长度为 4 的矢量 v1，v2，v3，v4，其中，函数 floor(v) 代表不大于 v 的最大的整数，函数 ceil(v) 代表不小于 v 的最小的整数；

分别使用纹理坐标 floor(t)-t-1 和 floor(s)-s-1 访问所述三次多项式纹理，得到 2 个长度为 4 的矢量 f1，f2；

利用 GPU 的 SIMD 矢量计算指令，计算得到一个长度为 4 的矢量 temp = (<v1 • f1>，<v2 • f1>，<v3 • f1>，<v4 • f1>)，<•> 表示矢量点积计算；

利用 GPU 的 SIMD 矢量计算指令，计算待求的采样值 I_{val} = <temp • f2>；

最后，将采样值 I_{val} 输出至显示器的 (x，y) 坐标处显示。

2. 根据权利要求 1 所述的高精度实时超声图像扫描变换方法，其特征在于，所述线数据纹理采用 RGBA 四个通道，每个通道为 32 位浮点数的格式；对于第 i 条线上第 j 个采样点，其 R 通道存放 (i，j) 处的采样值，G 通道存放 (i+1，j) 处的采样值，B 通道存放 (i+2，j) 处的采样值，A 通道存放 (i+3，j) 处的采样值。

3. 根据权利要求 1 所述的高精度实时超声图像扫描变换方法，其特征在于，所述三次多项式纹理采用 RGBA 四个通道，每个通道为 32 位浮点数的格式；对于在区间 $-2 < u < 2$ 的 u，R 通道存储 f(u) 的值，G 通道存储 f(u+1) 的值，B 通道存储 f(u+2) 的值，A 通道存储 f(u+3) 的值。

一种高精度实时超声图像扫描变换方法

技术领域

[0001] 本发明涉及一种高精度实时超声图像扫描变换方法，用于医用超声诊断设备。

背景技术

[0002] 超声波回波成像技术目前已经被广泛应用于军事、医疗等领域，通过向目标区域发射超声波，然后使用接收装置接收反射回来的回波信号，并通过信号处理技术和图像处理技术，抑制回波信号中的无用部分，最终形成目标区域的图像。

[0003] 在跟我们每个人的日常生活息息相关的医疗领域，超声波回波成像技术更是获得了长足的发展，目前各种医用超声诊断设备如 B 超等已经广泛应用于各个医院的临床诊断中，除了传统的黑白超可以观察病人的内部组织和器官的解剖结构外，彩超通过使用多普勒效应可以对血管内的血流成像，大大提高了超声诊断设备的临床应用范围。

[0004] 在目前的超声诊断设备中，扫描变换是其中非常重要的一个模块。超声诊断设备上目前大部分使用的探头，一次是沿着某个方向发射和接收超声数据的，通过多次发射和接收，形成了一帧完整的图像。对于常用的凸阵探头，采集到的数据形成了一个扇形的几何形状，如附图 1 所示，探头第一次沿着线段 AB 的方向发射超声数据，然后接收的时候在线段 AB 上进行等间隔采样，图中的空心点即为采样点。探头第二次沿着 CD 方向发射、接收并采样，第三次沿着 EF 方向放射、接收并采样，第四次沿着 GH 方向发射、接收并采样。这四个方向的采样数据，形成了一帧图像，其几何形状为一个扇形 ACEGHFDB，圆心在 O 点。但是对于目前的显示设备而言，都是光栅设备，只能显示一个一个离散的像素点，并且这些像素点只能排列在矩形网格上，如图中的实心点所示。因此要把超声诊断设备采集到的扇形数据，显示在 PQRS 这样一个矩形区域里面，必须通过一个变换过程，依据已知的空心点采样值，计算出每个实心点处的像素值，这个变换过程叫做扫描变换。

[0005] 扫描变换分为两个部分，一个部分为极坐标和直角坐标之间的坐标变换，这一部分主要的运算量在于每次变换都需要计算反三角函数和开根号函数；另外一个部分为图像插值，因为空心点和实心点的坐标大部分情况下都不重合，因此需要通过插值算法来从多个空心点的采样值计算一个实心点的像素值。传统上来讲，扫描变换是超声诊断设备中非常耗费资源的计算密集型的模块，为了简化起见，图 1 中假设只扫描了四次，或者说只获取了四条线数据，每条线上只采样了四个点，对于实际的情况，往往是要采集 512 线以上，每条线上采样点的个数也在 512 个以上。对于这么大的数据量，计算坐标变换和图像插值需要耗费很大量的计算资源，而这些计算又必须实时地完成才能达到临床检查的要求。为了满足实时扫描变换的需求，大部分超声诊断设备上都专门用 FPGA 或者 DSP 芯片来完成扫描变换，但是受硬件资源的限制，这些计算都是采用定点数计算，反三角函数和开根号函数等都是用的近似算法或者查表算法，计算精度受到很大的影响，另外图像插值算法用的是最近邻插值算法或者双线性插值算法，更高阶的插值算法因为运算量太大，无法达到实时的计算速度。

[0006] 从 2000 年以来，随着三维游戏产业的迅猛发展，普通的民用显卡的计算能力越来

越强，并且允许开发人员使用类似 C 语言的语法直接对显卡进行编程，如美国 NVidia 公司的 Cg、CUDA 编程语言，Microsoft 公司的 HLSL 编程语言，OpenGL 架构委员会的 GLSL 编程语言等。跟 CPU 相类似，显卡的计算核心被叫做 GPU(Graphics Processing Unit)，目前 GPU 除了应用于三维游戏当中的计算之外，也越来越多地被应用于通用计算。西门子公司的专利“Volume Renderingin the Acoustic Grid Methods and Systems for Ultrasound Diagnostic Imaging”(美国专利号 :6852081) 提出了一种基于 GPU 的扫描变换方法，首先在 CPU 中生成一个几何模型，把 AC、CE、EG、HF、FD、DB 用直线连接起来，用多边形 ACEGHFDB 来近似地逼近扇形 ACEGHFDB，如图 1 所示，多边形 ACEGHFDB 叫做代理几何模型。AB、CD、EF、GH 四条线上的采样数据被作为一个二维纹理，传输给 GPU 的纹理显存。代理几何模型上的每个顶点都被赋予一个纹理坐标，然后通过 GPU 内部硬件实现的纹理映射和双线性插值，来完成扫描变换过程。虽然这个方法可以在 GPU 上实现实时的扫描变换，但是仍然存在一些不足 :第一，其代理几何模型只是对原始扇形的一个逼近，会造成一定的计算误差 ;第二，只是代理几何模型的顶点 (A、C、E、G、H、F、D、B) 作了精确的坐标变换计算，代理几何模型内部的点均是通过 GPU 内部的纹理坐标插值机制来计算坐标变换的，是一个近似算法 ;第三，代理几何模型的顶点坐标和纹理坐标仍然需要 CPU 来计算，其中会牵涉到反三角函数和开根号函数，当采样的线数和帧频非常高时，会对 CPU 造成一定的计算负担 ;第四，图像插值是利用的 GPU 内置的双线性插值算法，插值所得到的函数曲面只是分段连续的，精度不够高。

[0007] 考虑到上述问题，在超声诊断设备上提供一种高精度的，能够实时完成计算的扫描变换方法，是非常有意义的。

发明内容

[0008] 本发明的目的是解决目前超声诊断设备中所使用的扫描变换模块精度不够高的问题，提供一种高精度实时超声图像扫描变换方法，使用基于 GPU 的计算架构，通过绘制合适的代理几何模型，将扫描变换模块中的坐标变换和图像插值算法全部放在 GPU 中作 32 位浮点数高精度计算，同时使用高精度的双三次插值算法，获得优化的图像质量。

[0009] 按照本发明提供的技术方案，一种高精度实时超声图像扫描变换方法步骤如下 :

[0010] 步骤 1 :获取超声探头所采集的每帧图像，由 LN 条线数据组成，形成一个扇形，内径为 r，外径为 r+l，第 1 条线数据和第 LN 条线数据所夹的圆心角为 2θ，LN 条线数据为等角度采样，每条采样线上有 PN 个采样点，为等距离采样。将所述每帧图像作为二维纹理传输至 GPU 显存，并保存为线数据纹理 ;

[0011] 步骤 2 :生成代理几何模型并绘制，包括 :

[0012] 使用六边形 ABQRHG 作为代理几何模型，其顶点坐标分别为 :

[0013] $(A_x, A_y) = (-r\sin\theta, r\cos\theta)$

[0014] $(B_x, B_y) = (-(r+1)\sin\theta, (r+1)\cos\theta)$

[0015] $(Q_x, Q_y) = (-(r+1)\sin\theta, r+1)$

[0016] $(R_x, R_y) = ((r+1)\sin\theta, r+1)$

[0017] $(H_x, H_y) = ((r+1)\sin\theta, (r+1)\cos\theta)$

[0018] $(G_x, G_y) = (r\sin\theta, r\cos\theta)$

[0019] 其中 A 和 G、B 和 H、Q 和 R 的 x 坐标相反，y 坐标相同，因此只需计算 A、B、Q 三个点的坐标，其它三个点的坐标根据上述关系求得；

[0020] 将六边形 ABQRHG 使用三角带形式分解为四个三角形，顶点顺序为 G，H，A，R，B，Q，三角形分别为：GHA、HAR、ARB 和 RBQ；

[0021] 使用 OpenGL 或者 DirectX 的 API 函数绘制三角带；

[0022] 步骤 3：在 GPU 中对代理模型绘制后的每个光栅点 (Fragment) 计算坐标变换，包括：

[0023] 将代理模型绘制后的每个光栅点从显示器的直角坐标 (x，y) 转换为所述线数据纹理所在的极坐标 (s，t)，计算公式为：$s=(\sqrt{x^2+y^2}-r)(PN-1)/l$，t = (arctan(x/y)+θ)(LN-1)/2θ，其中 t 代表在第 t 条线数据上，s 代表在线数据的第 s 个采样点上；

[0024] 判断 (s，t) 是否满足 0 ≤ t ≤ (LN-1) 且 0 ≤ s ≤ (PN-1)，如果不满足则终止计算；

[0025] 步骤 4：在 GPU 中计算双三次图像插值，包括：

[0026] 在 -2 < u < 2 的区间上取 NX 个离散的三次多项式 f(u) 的非零值，NX 为 2 的正整数次方，用一维纹理传输至 GPU 显存，并保存为三次多项式纹理，其中 f(u) 的定义如下：

[0027] $$f(u)=\begin{cases}\frac{7}{6}|u|^3-2|u|^2+\frac{8}{9}, & |u|<1\\ -\frac{7}{18}|u|^3+2|u|^2-\frac{10}{3}|u|+\frac{16}{9}, & 1\le|u|<2\\ 0, & |u|\ge 2\end{cases};$$

[0028] 分别使用纹理坐标 (floor(s)-1，floor(t)-1)，(floor(s)，floor(t)-1)，(ceil(s)，floor(t)-1)，(ceil(s)+1，floor(t)-1) 访问所述线数据纹理，得到 4 个长度为 4 的矢量 v1，v2，v3，v4，其中，函数 floor(v) 代表不大于 v 的最大的整数，函数 ceil(v) 代表不小于 v 的最小的整数；

[0029] 分别使用纹理坐标 floor(t)-t-1 和 floor(s)-s-1 访问所述三次多项式纹理，得到 2 个长度为 4 的矢量 f1，f2；

[0030] 利用 GPU 的 SIMD 矢量计算指令，计算得到一个长度为 4 的矢量 temp = (<v1 • f1>，<v2 • f1>，<v3 • f1>，<v4 • f1>)，<•> 表示矢量点积计算；

[0031] 利用 GPU 的 SIMD 矢量计算指令，计算待求的采样值 I_{val} = <temp • f2>；

[0032] 最后，将采样值 I_{val} 输出至显示器的 (x，y) 坐标处显示。

[0033] 所述线数据纹理采用 RGBA 四个通道，每个通道为 32 位浮点数的格式；对于第 i 条线上第 j 个采样点，其 R 通道存放 (i，j) 处的采样值，G 通道存放 (i+1，j) 处的采样值，B 通道存放 (i+2，j) 处的采样值，A 通道存放 (i+3，j) 处的采样值。

[0034] 所述三次多项式纹理采用 RGBA 四个通道，每个通道为 32 位浮点数的格式；对于在区间 -2 < u < 2 的 u，R 通道存储 f(u) 的值，G 通道存储 f(u+1) 的值，B 通道存储 f(u+2) 的值，A 通道存储 f(u+3) 的值。

[0035] 本发明的优点是：本发明使用目前普通民用显卡提供的 GPU 计算架构，可以对 512 条线数据，每条线上 1024 个采样点这样大规模的数据，达到每秒钟 60 帧以上的扫描变换速度。相比于已有的 GPU 方法，本发明使用了更简化的代理几何模型，并且无需指定每个顶点的纹理坐标，因此大大降低了 CPU 的负担，而将所有的计算全部放在 GPU 中以最高精度完

成，没有使用任何的近似算法。同时，图像插值算法并没有利用 GPU 内置的双线性插值算法，而是使用更高精度的双三次插值算法，保证了最终图像的质量。本发明所提供的方法可以充分挖掘利用当代 GPU 所提供的强大的并行计算能力，从而将传统的扫描变换模块升级为下一代高保真的扫描变换模块。

[0036] 附图说明

[0037] 图 1 是传统扫描变换的示意图。

[0038] 图 2 是本发明的实施流程图。

[0039] 图 3 是代理几何模型的示意图。

[0040] 图 4 是代理几何模型的坐标系及参数示意图。

[0041] 图 5 是双三次插值的示意图。

[0042] 具体实施方式

[0043] 下面结合附图和实施例详细说明本发明技术方案中所涉及的各个细节问题。应指出的是，所描述的实施例仅旨在便于对本发明的理解，而对其不起任何限定作用。

[0044] 如图 2 所示，本发明的实施流程分为 4 个步骤。在步骤 1 中，首先将探头采集到的线数据传输至 GPU 中的纹理显存。探头一次沿着某个方向发射和接收超声数据，在一个方向上接收到的数据叫做一条线数据，比如图 1 中的线段 AB。一条线数据可以用一个一维数组在计算机中表示，假设一条线上采样点的个数是 PN，那么这个一维数组的长度就是 PN。假设形成一帧图像需要采集的线数为 LN，则总共得到 LN 个一维的数组，或者说一个二维的数组，这个二维数组的大小为 PN×LN。假设我们用 lineArray 来代表这个二维数组，则 lineArray[i][j] 表示第 i 条线上第 j 个采样点的数值，这里使用基于 0 的索引，即 $0 \leqslant i \leqslant (LN-1)$，$0 \leqslant j \leqslant (PN-1)$。以图 1 为例，因为只有四条采样线，所以 LN ＝ 4，每条采样线上有四个采样点，所以 PN ＝ 4，lineArray[1][2] 代表 CD 采样线上 T 采样点的数值。为了进行扫描变换，必须首先把 lineArray 作为一个二维纹理传输到 GPU 的纹理显存中，然后 GPU 的计算单元才能访问到它。目前对 GPU 进行编程的工具主要有 OpenGL 和 DirectX，可以调用它们所提供的 API 函数来完成上述的纹理传输工作。

[0045] 需要注意的是，本发明中使用 RGBA 四通道的纹理格式，每通道是 32 位的浮点数，这个一方面是为了保证高精度的计算，另外一方面也是为了加快图像插值算法的速度，具体会在步骤 4 中给出详细的说明。

[0046] 在步骤 2 中，生成代理几何模型，然后通过 OpenGL 或者 DirectX 等图形 API 的绘制命令将代理几何模型传送至 GPU，由 GPU 对其进行光栅化操作，生成一个一个的光栅片段 (Fragment)，再由 GPU 的 Fragment Shader 对每一个 Fragment 进行处理。代理几何模型的选择非常重要，不能过于复杂，否则会对 CPU 造成很大的计算代理几何模型顶点坐标的负担，也会导致通过 AGP 总线或者 PCIE 总线给 GPU 传输数据太多，从而形成传输瓶颈。与之前的技术不同，本发明中使用非常简单的代理几何模型，如图 3 所示，六边形 ABQRHG 完全包含了 ABHG 扇形成像区域，可以选择它作为代理几何模型。直线 QR 是 BH 弧的切线，直线 BQ 和 HR 垂直于直线 QR。生成六边形 ABQRHG 只需计算六个顶点的坐标，并且顶点 A 和 G、B 和 H、Q 和 R 都是关于六边形的中心线对称的，更进一步简化了 CPU 所需的计算。因为 GPU 对由三角形组成的几何模型的绘制效率最高，因此还需把六边形分解成一系列的三角形。为了减少传输给 GPU 的几何数据，这里使用三角带 (Triangle Strip) 来对六边形进行分解，所谓

三角带，就是指的一系列的三角形，相邻两个三角形必须共享一条边，或者说两个顶点。三角带对顶点顺序是有规定的，假设给定 1，2，3，.....，n+2 总共 n+2 个顶点，则 1，2，3 三个顶点组成一个三角形，2，3，4 三个顶点组成一个三角形，依次类推，直到 n，n+1，n+2 三个顶点组成一个三角形，因此可以用 n+2 个顶点来表示 n 个三角形 (n ≥ 1)，而如果用普通的三角形表达方式，则需要 3n 个顶点才能表示 n 个三角形。分别连接 AH、AR、RB 三条直线段，将六边形分解为四个三角形，用三角带的形式表达，顶点顺序为 G，H，A，R，B，Q，分别代表三角形 GHA、HAR、ARB 和 RBQ。

[0047] 用三角形来对代理几何模型进行分解后，还需要计算出每个顶点的坐标，如图 4 所示，首先建立坐标系，坐标原点定在扇形 ABHG 的圆心 O 处，x 轴水平从左到右，y 轴竖直从上到下。直线段 OA 的长度为 r，AB 的长度为 1，OB 与 y 轴正向的夹角为 θ，这些参数都是已知值。从上面的这些条件，可以计算出顶点 A 的坐标为：$(A_x, A_y) = (-r\sin\theta, r\cos\theta)$，顶点 G 的坐标为：$(G_x, G_y) = (r\sin\theta, r\cos\theta)$，顶点 B 的坐标为：$(B_x, B_y) = (-(r+1)\sin\theta, (r+1)\cos\theta)$，顶点 H 的坐标为：$(H_x, H_y) = ((r+1)\sin\theta, (r+1)\cos\theta)$，顶点 Q 的坐标为：$(Q_x, Q_y) = (-(r+1)\sin\theta, r+1)$，顶点 R 的坐标为：$(R_x, R_y) = ((r+1)\sin\theta, r+1)$。从上述坐标计算公式可以看出，实际上只需计算 A、B、Q 三个顶点的坐标，G、H、R 分别和 A、B、Q 关于 y 轴对称，因此无需计算，另外 B 和 Q、H 和 R 的 x 坐标相等，又可以节省昂贵的三角函数计算。计算得到代理模型的顶点坐标之后，就可以使用 OpenGL 或者 DirectX 的 API 函数将代理几何模型绘制并交给 GPU 进行后续处理。

[0048] 与之前已有的技术相比，本发明所提出方法除了使用的代理模型更为简单，计算量更小之外，坐标变换的精度也更高。传统的技术在生成代理几何模型时，对于每个顶点都要计算并分配一个纹理坐标，这一方面增加了 CPU 的计算负担，另外一方面依赖纹理坐标的双线性插值去计算坐标变换，是一个粗糙的逼近。本发明中在生成代理几何模型时，并没有给顶点分配纹理坐标，而是在步骤 3 中，直接在 GPU 的 Fragment Shader 中用 32 位浮点数进行高精度的坐标变换计算。需要注意的是，步骤 1 和步骤 2 都是在 CPU 中完成的，从步骤 3 开始，所有的计算都是在 GPU 中完成。通过步骤 2，六边形 ABQRHG 被绘制成为一系列离散的像素点，这些像素点的集合组成了六边形 ABQRHG 的内部和边界，不过在屏幕的分辨率下肉眼看是连续的。其中每个像素点叫做 Fragment，记录有自己的坐标、颜色等信息，会送给 GPU 的 Fragment Shader 作进一步的处理。GPU 的 Fragment Shader 是一个可编程的器件，可以使用 GLSL、Cg、HLSL 或者 CUDA 对其进行编程，以达到自己特定的计算目的。每个 Fragment 所记录的坐标是在屏幕坐标系的 (x，y) 坐标，是直角坐标系，而探头所采集的线数据用极坐标系更容易表达，如图 5 和图 1，任取一个 Fragment，假设为 I，其直角坐标为 (x，y)，线段 OI 的长度为 ρ，OI 与 y 轴正向的夹角为 Φ，则 I 可以用极坐标 (ρ，Φ) 来表示，其中 $\rho = \sqrt{x^2 + y^2}$，$\Phi = \arctan(x/y)$，$r \leqslant \rho \leqslant (r+1)$，$-\theta \leqslant \Phi \leqslant \theta$。另外，还必须计算出 I 是处于第几条采样线上，是在采样线的第几个采样点上。假设采集一帧图像的总采样线的条数为 LN，一条采样线上采样点的个数是 PN，I 处于第 t 条采样线的第 s 个采样点，$0 \leqslant t \leqslant (LN-1)$，$0 \leqslant s \leqslant (PN-1)$，则从极坐标 (ρ，Φ) 可以很容易地推出来 (s，t) 的取值：$s = (\rho - r)(PN-1)/l$，$t = (\Phi + \theta)(LN-1)/2\theta$，最终得到的 (s，t) 和 (x，y) 之间的坐标变换关系为：$s = (\sqrt{x^2 + y^2} - r)(PN-1)/l$，$t = (\arctan(x/y) + \theta)(LN-1)/2\theta$。

[0049] 通过上述坐标变换计算得到的 s 和 t 不一定是整数，但是探头实际采样得到的数据，s 和 t 必须是整数，因此需要使用图像插值算法得到任意 (s,t) 坐标处的采样值。传统的技术为了计算速度，往往采用最近邻插值或者双线性插值算法。所谓最近邻插值算法，是对 s 和 t 分别四舍五入，取最近的整数坐标处的采样值，这种方法速度最快，但是图像质量也最差。用的最为广泛的是双线性插值算法，在 I 的周围取最近的四个邻域采样点 J、K、L、M，其 (s,t) 坐标分别为 :(floor(s),floor(t))、(ceil(s),floor(t))、(ceil(s),ceil(t))、(floor(s),ceil(t))，其中 floor(x) 函数代表不大于 x 的最大的整数，ceil(x) 函数代表不小于 x 的最小的整数。I 点处的采样值由 J、K、L、M 四个整数坐标点的采样值加权平均计算得到，具体计算公式为 :$I_{val} = J_{val}(1-ds)(1-dt)+K_{val}ds(1-dt)+L_{val}dsdt+M_{val}(1-ds)dt$，其中 ds = s-floor(s)，dt = t-floor(t)，I_{val}，J_{val}，K_{val}，L_{val}，M_{val} 分别是 I、J、K、L、M 点处的采样值。由于双线性插值算法的计算复杂度不高，在 GPU 硬件内部也内置了对双线性插值算法的支持，能达到实时的要求，并且图像质量也可以接受，因此目前所有主流的超声诊断设备上都使用其作为扫描变换模块的插值算法。

[0050] 本发明使用的简单的代理几何模型，大大简化了前面几个步骤的计算负担，因此为使用更为复杂、精度更高的图像插值算法提供了可能。在步骤 4 中，不同于传统技术使用双线性插值算法，本发明在 GPU 中计算双三次图像插值。如图 5 所示，在 I 的周围取最近的 16 个邻域采样点 A、C、E、G、W、J、M、N、V、K、L、U、B、D、F、H，其 (s,t) 坐标分别为 :

[0051] $(A_s,A_t) = (floor(s)-1,floor(t)-1)$

[0052] $(C_s,C_t) = (floor(s)-1,floor(t))$

[0053] $(E_s,E_t) = (floor(s)-1,ceil(t))$

[0054] $(G_s,G_t) = (floor(s)-1,ceil(t)+1)$

[0055] $(W_s,W_t) = (floor(s),floor(t)-1)$

[0056] $(J_s,J_t) = (floor(s),floor(t))$

[0057] $(M_s,M_t) = (floor(s),ceil(t))$

[0058] $(N_s,N_t) = (floor(s),ceil(t)+1)$

[0059] $(V_s,V_t) = (ceil(s),floor(t)-1)$

[0060] $(K_s,K_t) = (ceil(s),floor(t))$

[0061] $(L_s,L_t) = (ceil(s),ceil(t))$

[0062] $(U_s,U_t) = (ceil(s),ceil(t)+1)$

[0063] $(B_s,B_t) = (ceil(s)+1,floor(t)-1)$

[0064] $(D_s,D_t) = (ceil(s)+1,floor(t))$

[0065] $(F_s,F_t) = (ceil(s)+1,ceil(t))$

[0066] $(H_s,H_t) = (ceil(s)+1,ceil(t)+1)$

[0067] I 点处的采样值由上述 16 个整数坐标点的采样值加权平均计算得到，具体计算方法为 :先由 A、C、E、G 四个采样点处的采样值 A_{val}，C_{val}，E_{val}，G_{val} 插值计算得到一个临时的采样值 temp1，计算公式为

[0068] $temp1 = A_{val}f(A_t-t)+C_{val}f(C_t-t)+E_{val}f(E_t-t)+G_{val}f(G_t-t)$，

[0069] 其中 f(x) 为一个三次多项式，其定义为 :

[0070] $$f(x)=\begin{cases}\frac{7}{6}|x|^3-2|x|^2+\frac{8}{9}, & |x|<1\\ -\frac{7}{18}|x|^3+2|x|^2-\frac{10}{3}|x|+\frac{16}{9}, & 1\le|x|<2\\ 0, & |x|\ge 2\end{cases}$$

[0071] 接着分别由 W、J、M、N 四个采样点处的采样值 W_{val}，J_{val}，M_{val}，N_{val} 插值计算得到第二个临时的采样值 temp2，计算公式为

[0072] $temp2 = W_{val}f(W_t-t)+J_{val}f(J_t-t)+M_{val}f(M_t-t)+N_{val}f(N_t-t)$ ；

[0073] 由 V、K、L、U 四个采样点处的采样值 V_{val}，K_{val}，L_{val}，U_{val} 插值计算得到第三个临时的采样值 temp3，计算公式为

[0074] $temp3 = V_{val}f(V_t-t)+K_{val}f(K_t-t)+L_{val}f(L_t-t)+U_{val}f(U_t-t)$ ；

[0075] 由 B、D、F、H 四个采样点处的采样值 B_{val}，D_{val}，F_{val}，H_{val} 插值计算得到第四个临时的采样值 temp4，计算公式为

[0076] $temp4 = B_{val}f(B_t-t)+D_{val}f(D_t-t)+F_{val}f(F_t-t)+H_{val}f(H_t-t)$ ；

[0077] 最后由四个临时采样值 temp1 ～ temp4 再次插值计算得到 I 点处的采样值，计算公式为 $I_{val} = temp1 \cdot f(A_s-s)+temp2 \cdot f(W_s-s)+temp3 \cdot f(V_s-s)+temp4 \cdot f(B_s-s)$。

[0078] 从上面的计算过程可以看出双三次插值需要大量的运算，对于每个 Fragment，光是计算三次多项式就需要 20 次，每个三次多项式至少需要 7 次乘法，加上加权平均需要的 20 次乘法，每个 Fragment 至少需要 160 次乘法，另外还需要 16 次内存访问。假设扫描变换后的图像为 800×600，为了达到实时超声检查的需求，成像速度至少要高于 30 帧 / 秒，因此光是插值这一个步骤就至少需要 $800\times600\times30\times160 = 2.304\times10^9$ 次乘法运算，再加上 $800\times600\times30\times16 = 2.304\times10^8$ 次内存访问和一些加法运算，这在目前的 CPU 上是不可能的，如果用专用的 FPGA 或者 ASIC 硬件电路来实现，如此规模的电路需要很高的成本，因此目前高精度的双三次图像插值算法尚没有广泛应用。即使对于目前拥有高度并行化架构的 GPU，其一次能够并行处理多个 Fragment，但是其对显存的访问速度仍然远远落后于其计算速度，对于如此大的运算量和显存访问次数，仍然很难达到实时的要求。

[0079] 要使得双三次插值在目前中低档的显卡上能够达到超声实时成像的需求，必须进行优化。本发明中对最耗时的两个部分结合 GPU 的特点进行了优化：第一个部分是三次多项式的计算，第二个部分是纹理显存的访问优化。对于三次多项式 f(x) 来说，其只在 $-2 < x < 2$ 的范围内取非零值，在其它区间都是零，因此可以将 $-2 < x < 2$ 的区间离散化成 NX 个点，预先算好这些离散点上的 f(x) 取值，然后将其作为一维纹理传输并储存在 GPU 的纹理显存中。GPU 的纹理单元一般都是对长度为 2 的整数次方的纹理作了高度优化，因此在具体实施中，NX 可以取值为 512、1024、2048 等，取值越大，三次多项式的计算精度就越高，但是所占用的纹理显存也越大，因此要根据所用显卡的实际情况来决定 NX 的取值。另外，GPU 所采用的都是 SIMD 指令集，对于矢量运算提供了最优化的支持，仔细观察双三次插值中的 5 次插值计算，发现它们都是两个长度为 4 的矢量的点积计算，可以统写为：

[0080] $\langle(val1, val2, val3, val4) \cdot (f(x_1), f(x_2), f(x_3), f(x_4)\rangle$，

[0081] 其中 val1 ～ val4 分别代表四个采样值，x_1 ～ x_4 代表对应的自变量取值，另外它们满足：$x_4 = x_3+1 = x_2+2 = x_1+3$，对于前四次插值计算，自变量只牵涉到 t 坐标，并且这四次所用到的 x_1 ～ x_4 是对应相等的，第 5 次插值计算，自变量是用的 s 坐标，其所用 x_1 ～ x_4

跟前面四次是不同的，因此只需两套 $x_1 \sim x_4$ 即可。对于目前支持 OpenGL 2.0 或者 DirectX 9.0 的比较新的 GPU 来讲，其纹理可以使用 RGBA 四个通道，每个通道 32 位浮点，一个纹理元素总共 128 位，因此可以同时放四个采样值或者 f(x) 的值。在步骤 1 生成线数据纹理时，第 i 条线上第 j 个采样点的 R 通道放 (i，j) 处的采样值，G 通道放 (i+1，j) 处的采样值，B 通道放 (i+2，j) 处的采样值，A 通道放 (i+3，j) 处的采样值；同样，在生成三次多项式纹理时，对于任意一个 x，R 通道存储 f(x) 的值，G 通道存储 f(x+1) 的值，B 通道存储 f(x+2) 的值，A 通道存储 f(x+3) 的值。

[0082] 在具体实现优化的双三次插值算法时，要首先通过四次纹理显存访问，得到 16 个采样点 A、C、E、G、W、J、M、N、V、K、L、U、B、D、F、H 的采样值，这个可以通过使用纹理坐标

[0083] (floor(s)-1，floor(t)-1)，

[0084] (floor(s)，floor(t)-1)，

[0085] (ceil(s)，foor(t)-1)，

[0086] (ceil(s)+1，floor(t)-1)

[0087] 访问线数据纹理，得到 4 个长度为 4 的 RGBA 矢量 v1，v2，v3，v4，其中

[0088] $v1 = (A_{val}, C_{val}, E_{val}, G_{val})$，$v2 = (W_{val}, J_{val}, M_{val}, N_{val})$，

[0089] $v3 = (V_{val}, K_{val}, L_{val}, U_{val})$，$v4 = (B_{val}, D_{val}, F_{val}, H_{val})$；

[0090] 再通过使用纹理坐标 floor(t)-t-1 和 floor(s)-s-1 对三次多项式纹理进行两次访问，得到 2 个长度为 4 的 RGBA 矢量 f1，f2，其中

[0091] $f1 = (f(A_t-t), f(C_t-t), f(E_t-t), f(G_t-t)) = (f(W_t-t), f(J_t-t), f(M_t-t), f(N_t-t)) = (f(V_t-t), f(K_t-t), f(L_t-t), f(U_t-t)) = (f(B_t-t), f(D_t-t), f(F_t-t), f(H_t-t))$，$f2 = (f(A_s-s), f(W_s-s), f(V_s-s), f(B_s-s))$。然后利用 GPU 内部提供的 SIMD 矢量点积指令，计算 $temp = (\langle v1 \cdot f1\rangle, \langle v2 \cdot f1\rangle, \langle v3 \cdot f1\rangle, \langle v4 \cdot f1\rangle)$，最后再计算出待求的采样值 $I_{val} = \langle temp \cdot f2\rangle$，并将 I 点处的采样值 I_{val} 输出至显示器。

[0092] 通过上述的优化算法，一方面大大减少了访问纹理显存的次数，从原来的需要 16 次减少到现在的 16/4+2 = 6 次，其中包括 4 次访问线数据纹理和 2 次访问三次多项式纹理；另外一方面可以充分利用 GPU 提供的 SIMD 矢量运算，将原来的 160 次乘法减少到 5 次矢量点积运算。通过这两个方面的优化，可以充分挖掘 GPU 的潜力，实现实时的双三次插值运算。

[0093] 除了上述的优化之外，还可以通过减少要处理的 Fragment 的数量来进一步加快算法的速度。如图 3 所示，在绘制代理几何模型时，图中的阴影区域实际上并不在实际成像区域中，因此需要忽略掉阴影区域光栅化得到的那些 Fragment。对于每个 Fragment，在步骤 3 通过坐标变换计算得到它的 (s，t) 坐标，如果不满足 $0 \leqslant t \leqslant (LN-1)$，$0 \leqslant s \leqslant (PN-1)$ 这两个条件，就可以说明此 Fragment 处于阴影区域中，不需要进行步骤 4 的插值运算，可以提前终止此 Fragment 的处理。

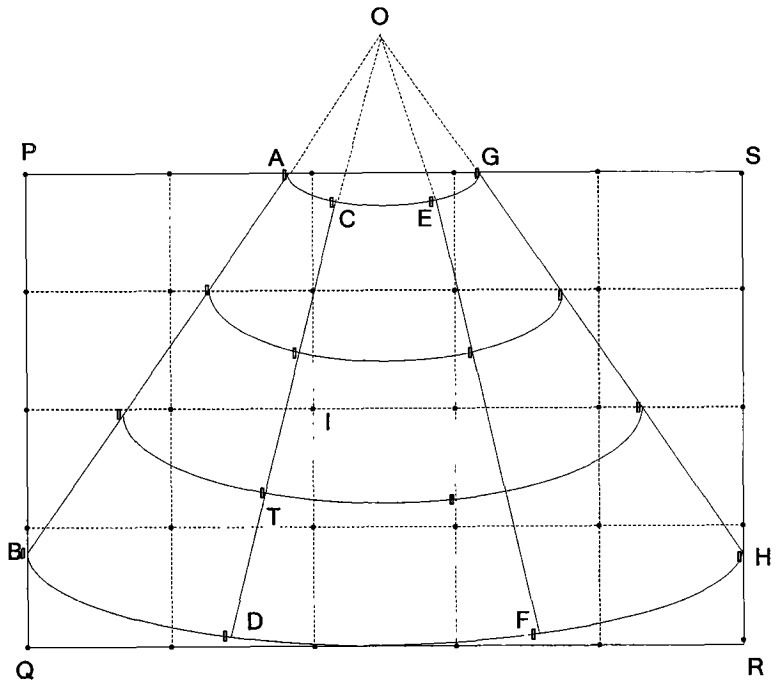

O
P
A
G
S
C
E
I
T
B
H
D
F
Q
R

图 1

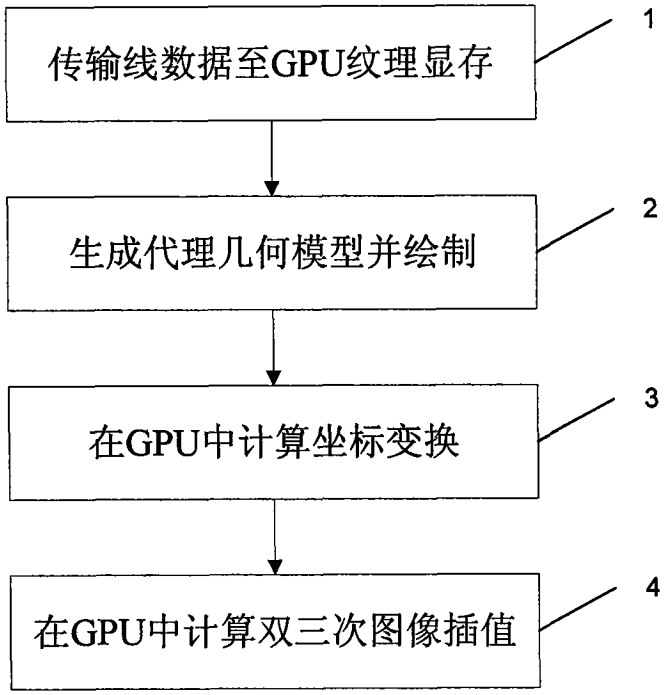

传输线数据至GPU纹理显存
1
生成代理几何模型并绘制
2
在GPU中计算坐标变换
3
在GPU中计算双三次图像插值
4

图 2

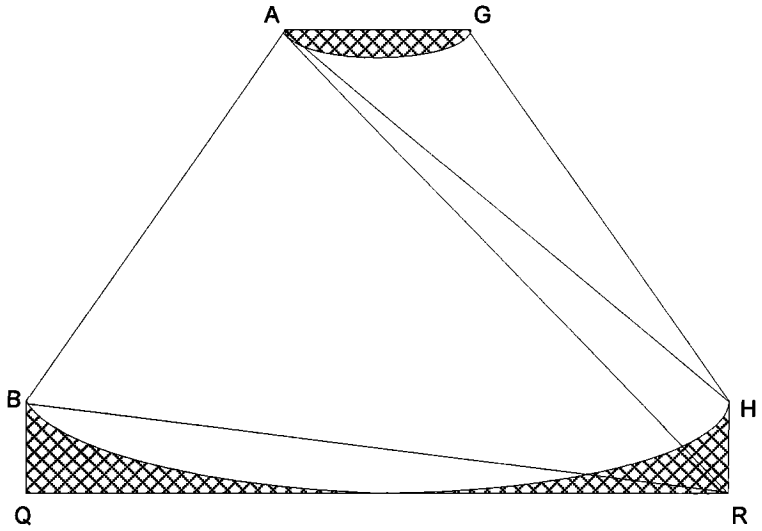


图 3

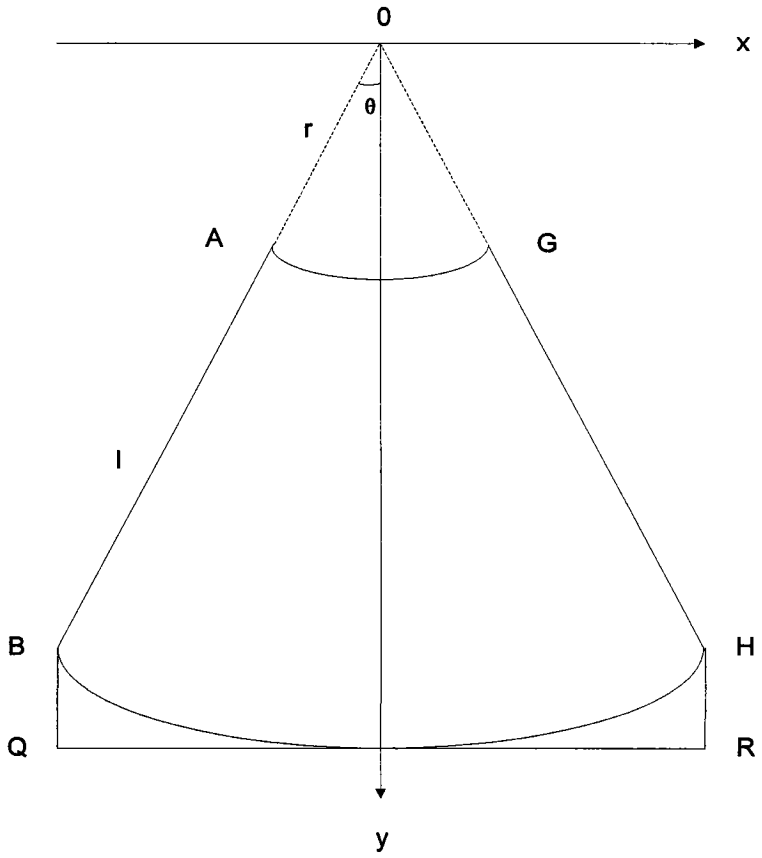


图 4

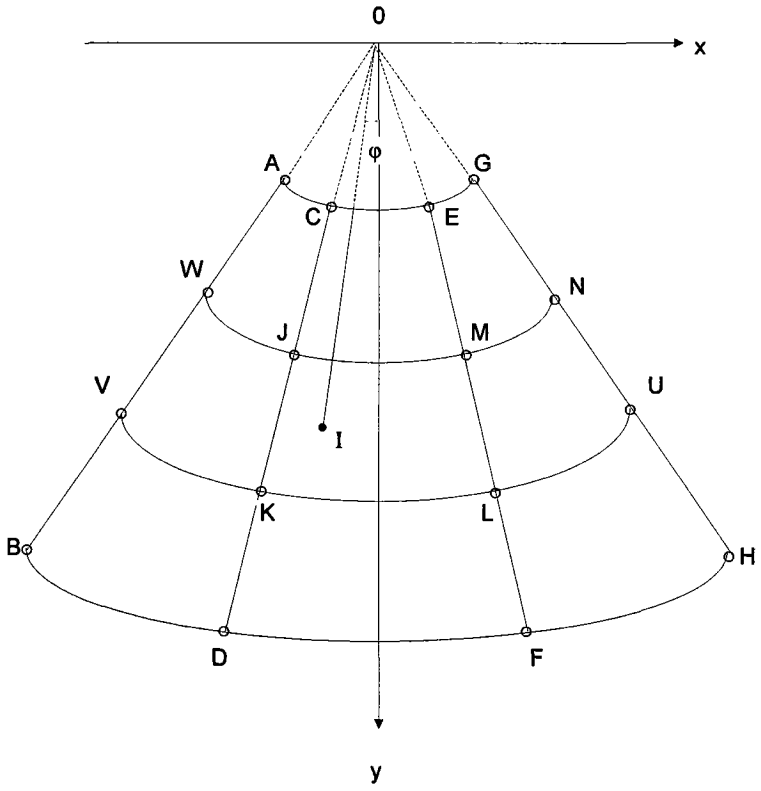


图 5

专利名称(译)	一种高精度实时超声图像扫描变换方法		
公开(公告)号	CN101606854B	公开(公告)日	2010-11-17
申请号	CN200910033262.7	申请日	2009-06-10
[标]发明人	赵明昌 莫善珏		
发明人	赵明昌 莫善珏		
IPC分类号	A61B8/14 A61B8/00 G06F19/00		
审查员(译)	胡亚婷		
其他公开文献	CN101606854A		
外部链接	Espacenet SIPO		

摘要(译)

本发明公开了一种高精度实时超声图像扫描变换方法，使用基于GPU的并行计算架构和SIMD矢量运算，通过绘制合适的代理几何模型，将扫描变换模块中的坐标变换和图像插值算法全部放在GPU中作32位浮点数高精度计算，同时使用优化的高精度的双三次插值算法，获得更平滑的图像质量。本发明可以在目前普通中低档显卡上，对512条线数据，每条线上1024个采样点这样大规模的数据，达到每秒钟60帧以上的扫描变换速度，可以充分挖掘利用当代GPU所提供的强大的并行计算能力，从而将传统的扫描变换模块升级为下一代高保真的扫描变换模块。