



(12) 发明专利申请

(10) 申请公布号 CN 101739661 A

(43) 申请公布日 2010.06.16

(21) 申请号 200910242341.9

(22) 申请日 2009.12.09

(71) 申请人 中国科学院自动化研究所

地址 100080 北京市海淀区中关村东路 95 号

(72) 发明人 赵明昌 何晖光

(74) 专利代理机构 中科专利商标代理有限责任公司 11021

代理人 梁爱荣

(51) Int. Cl.

G06T 1/00 (2006.01)

A61B 8/12 (2006.01)

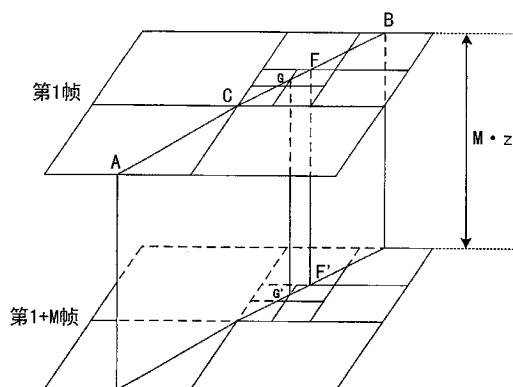
权利要求书 2 页 说明书 7 页 附图 2 页

(54) 发明名称

一种快速高保真地构造血管内超声长轴影像的方法

(57) 摘要

本发明公开了一种快速高保真地构造血管内超声长轴影像的方法,使用基于图形处理器的并行计算架构,通过四叉树数据结构对大数据量三维 IVUS 数据进行高效管理,递归遍历四叉树并且计算长轴影像所在平面与四叉树节点的交点坐标,将长轴影像分解为一系列小的平面,然后再送至图形处理器绘制并作插值计算。解决数据量太大不能全部加载到图形处理器显存中的问题,提高图形处理器纹理缓存的利用效率。将三维样条曲线插值使用图形处理器的单指令多数据矢量点积来计算,提高了运算速度和图像的质量。本发明在普通中高档显卡上,对 $512 \times 512 \times 4000 \times 16\text{bit}$ 这样大规模的 IVUS 影像数据,达到每秒钟 30 帧以上的高保真长轴影像构造速度。



1. 一种快速高保真地构造血管内超声长轴影像的方法,其特征在于,步骤如下:

步骤 1:加载三维血管内超声数据,其宽度和高度分别为 W 像素、H 像素,影像帧数为 N,相邻像素 x 向物理间距为 x_s 毫米, y 向物理间距为 y_s 毫米,相邻两帧影像之间 z 向物理间距为 z_s 毫米;

步骤 2:获取用户指定的直线段 AB,用于构造血管内超声长轴影像,直线段 AB 与 x 轴的夹角为 θ ;

步骤 3:沿着三维血管内超声数据的第一帧影像所在平面进行四叉树均匀划分,使得叶子节点的大小为 $M \times M$;

步骤 4:对四叉树从根节点开始递归遍历,并处理每个节点,包括:

判断当前节点是否与直线段 AB 相交,如果当前节点与直线段 AB 相交并且当前节点不为叶子节点,则继续递归遍历;如果当前节点与直线段 AB 相交并且当前节点为叶子节点,则执行步骤 5;如果当前节点与直线段 AB 不相交则直接返回父节点继续遍历;

步骤 5:计算叶子节点与直线段 AB 的交点 F、G 的 x、y 坐标,并同时得到第 M+1 帧影像上对应点交点 F'、G' 的 x、y 坐标,使得交点 F' 的 x 坐标等于交点 F 的 x 坐标,交点 G' 的坐标等于交点 G 的坐标,交点 F'、G' 的 z 坐标分别等于交点 F、G 的 z 坐标加上 $M \times z_s$;绘制平面 FF' G' G,同时将叶子节点所在的数据块以三维纹理的方式传输至图形处理单元处理;将上述过程重复 $\lfloor N/M \rfloor + 1$ 次,每次只需将交点 F、G、F'、G' 的 z 坐标分别加上 $M \times z_s$,对应的块数据随之更新即可;

步骤 6:在图形处理单元的片段着色器中对绘制平面 FF' G' G 所产生的每个片段进行三维的样条曲线插值。

2. 根据权利要求 1 所述的快速高保真地构造血管内超声长轴影像的方法,其特征在于,所述叶子节点与直线段 AB 相交的计算方法,包括:

从叶子节点中获取其对应的长方形的四个顶点坐标为: (x_0, y_0) 、 (x_1, y_0) 、 (x_1, y_1) 和 (x_0, y_1) ;

计算直线段 AB 中心点 C 点的坐标 C_x, C_y 为: $(C_x, C_y) = (W \cdot x_s / 2, H \cdot y_s / 2)$;

计算直线段 AB 的参数方程:

$$\begin{cases} x(t) = C_x + t \cos \theta \\ y(t) = C_y + t \sin \theta \end{cases},$$

其中 t 为参数, $x(t)$ 、 $y(t)$ 为直线段 AB 上任意一点的 x、y 坐标;

计算直线段 AB 与直线 $x = x_0$ 的交点: $(x_0, C_y + (x_0 - C_x) \tan \theta)$;

计算直线段 AB 与直线 $x = x_1$ 的交点: $(x_1, C_y + (x_1 - C_x) \tan \theta)$;

计算直线段 AB 与直线 $y = y_0$ 的交点: $(C_x + (y_0 - C_y) \cot \theta, y_0)$;

计算直线段 AB 与直线 $y = y_1$ 的交点: $(C_x + (y_1 - C_y) \cot \theta, y_1)$;

判断以下四个条件:

$$\begin{cases} y_0 \leq C_y + (x_0 - C_x) \tan \theta \leq y_1 \\ y_0 \leq C_y + (x_1 - C_x) \tan \theta \leq y_1 \\ x_0 \leq C_x + (y_0 - C_y) \cot \theta \leq x_1 \\ x_0 \leq C_x + (y_1 - C_y) \cot \theta \leq x_1 \end{cases},$$

是否满足,如果四个条件都不满足,那么直线段 AB 和此叶子节点不相交,如果四个条件至少有一个满足,则直线段 AB 和此叶子节点相交,并且每满足一个条件,就可以得到一个交点坐标,当满足的条件大于 2 个时,必然有重合的顶点,去掉重合的顶点,可以得到两个交点 F、G 的坐标。

3. 根据权利要求 1 所述的快速高保真地构造血管内超声长轴影像的方法,其特征在于,所述三维样条曲线插值算法,包括:

将传统的插值计算公式:

$$I(x_v, y_v, z_v) = \sum_{k=-1}^2 \left\{ \sum_{j=-1}^2 \left[\sum_{i=-1}^2 I(\lfloor x_v \rfloor + i, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k) \cdot f(\lfloor x_v \rfloor + i - x_v) \right. \right. \\ \left. \left. \cdot f(\lfloor y_v \rfloor + j - y_v) \right] \cdot f(\lfloor z_v \rfloor + k - z_v) \right\}$$

改写为基于长度为 4 的矢量点积的运算:

$$I(x_v, y_v, z_v) = \sum_{k=-1}^2 \left[\sum_{j=-1}^2 \langle \bar{I}(j, k), \bar{f}_{x_v} \rangle \cdot f(\lfloor y_v \rfloor + j - y_v) \right] \cdot f(\lfloor z_v \rfloor + k - z_v)$$

其中 (x_v, y_v, z_v) 分别为待处理的片段映射到数据空间的 x、y、z 坐标, i、j、k 为求和索引变量, $I(x, y, z)$ 为加载的三维血管内超声影像数据在第 z 帧上 (x, y) 坐标处的像素值, f 为样条曲线函数, $\bar{I}(j, k)$ 是长度为 4 的矢量:

$$(I(\lfloor x_v \rfloor - 1, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k), I(\lfloor x_v \rfloor, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k), \\ I(\lfloor x_v \rfloor + 1, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k), I(\lfloor x_v \rfloor + 2, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k))$$

$\bar{f}_{x_v}$ 也是长度为 4 的矢量:

$$(f(\lfloor x_v \rfloor - 1 - x_v), f(\lfloor x_v \rfloor - x_v), f(\lfloor x_v \rfloor + 1 - x_v), f(\lfloor x_v \rfloor + 2 - x_v))$$

$\langle, \rangle$ 是矢量点积运算;

使用图形处理单元的单指令多数据矢量点积运算,一次完成一个长度为 4 的矢量点积,通过 21 次矢量点积运算来完成所述三维样条曲线插值。

一种快速高保真地构造血管内超声长轴影像的方法

技术领域

[0001] 本发明涉及医学影像处理技术领域,具体地说涉及一种快速高保真地构造血管内超声长轴影像的方法。

背景技术

[0002] 在全世界范围内,目前冠状动脉性心脏病(简称冠心病)是导致人类死亡的首要原因之一,而冠状动脉介入治疗手术是治疗冠心病的最主要手段。最近新兴起的血管内超声(IVUS)影像被认为是辅助冠状动脉介入治疗手术最好的一种影像方式。IVUS 影像的获取是微创式的,通过在导管上加一个非常小型的高频超声探头,通过病人的血管将其放置到医生怀疑有病变的冠脉中,通过马达控制,匀速回拉探头,从而得到一系列的血管内窥影像。

[0003] 目前医生对于 IVUS 影像的阅片方式主要是长轴成像法,假设总共获取了 N 帧影像,医生可以在第一帧影像上用鼠标定义一条直线段 AB (请参看附图 1),因为 IVUS 影像的中心一般是血管内壁等医生感兴趣的区域,加上导管的高回波区,所以直线段 AB 是通过图像的中心点 C 的,医生通过鼠标操作,可以绕中心点 C 旋转直线段 AB ,直至得到比较满意的角度,此时系统会在第一帧影像上沿着直线段 AB 进行图像采样,得到一条线数据并映射到长轴影像的 1 号线上,其中 A 、 B 、 C 分别映射到 A' 、 B' 、 C' 上,因为 N 帧影像的大小都是相等的,因此 A 、 B 的坐标可以直接应用到第 2 ~ N 帧影像上,通过同样的过程,可以得到长轴影像的第 2 ~ N 号线的图像数据。医生通过观察和测量长轴影像,可以得到血管壁、斑块、支架放置等信息,从而辅助手术或者评价手术后支架释放贴壁情况。

[0004] 因为 IVUS 影像的获取是通过马达控制匀速回拉获取的,目前典型的回拉速率是 0.5mm/s ,假设要获取 5cm 的数据,那么需要采集 100 秒,以 25帧/s 的速率来计算,那么总共获取 $N = 2500$ 帧影像,实际上临床中一次获取的数据量经常为 $N > 4000$ 帧。每帧影像的分辨率一般是 512×512 ,每个像素 8bit 或者 16bit,因此临床中典型的 IVUS 数据量为: $512 \times 512 \times 4000 \times 16 \div 8 = 2000\text{MB}$ 。这么庞大的数据量给长轴影像的快速构造带来了很大的挑战,目前已有的方法都是在图像质量上作了折中以便达到实时生成长轴影像的目的,比如考虑到 N 远远大于目前显示器的分辨率,因此在生成长轴影像时,可以采用下采样的方法,每隔若干条线计算一次,这样可以成倍地减少计算量,但是生成的长轴影像的质量也随之下降很多。另外,目前的算法只是在一帧影像上沿着一条直线进行采样,只采用了图像的二维插值,没有利用相邻帧进行三维插值,为了追求计算速度,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,而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,因此目前没有被采用。

[0005] 最近几年,高性能计算的趋势发生了很大的变化,普通民用显卡所包含的图形处理单元(GPU,Graphics Processing Unit)芯片的计算能力越来越强,逐渐地超越了传统的中央处理器(CPU)的计算能力,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。目前 GPU 越来越多地被应用于通用计算,在图像处理、视频处理、三维游戏等领域显示了强大的并行计算能力,使得原来一些非常难以实时处理的算法变得可能,比如三维医学影像数据的实时体绘制 (Volume Rendering),在目前的主流显卡上可以 30 帧/s 的速度绘制 $512 \times 512 \times 512$ 规模的数据集。另外一个发展趋势是利用 GPU 强大的计算能力,将传统的算法中为了提高绘制速度而对图像质量所作的折中给消除,使用最高精度的计算方法,从而得到最高质量的图像,比如美国的 Fovia 公司 (<http://www.fovia.com/index.php>) 在 2007 年提出了高保真体绘制 (High Definition Volume Rendering) 的算法,从而将传统体绘制的图像质量提高了一个档次。

[0006] 考虑到 GPU 相比于 CPU 计算的优势以及在体绘制方面取得的成功,在 IVUS 影像设备或者软件中提供一种基于 GPU 的针对大数据量 IVUS 影像的高保真长轴影像构造算法,是非常有意义的。

发明内容

[0007] 本发明的目的是解决目前 IVUS 影像设备或者软件中所使用的长轴影像构造算法精度不够高的问题,为此,提供一种高保真的实时大数据量 IVUS 长轴影像构造方法。

[0008] 为达成所述目的,本发明提供一种快速高保真地构造血管内超声长轴影像的方法,包括步骤如下:

[0009] 步骤 1:加载三维血管内超声数据,其宽度和高度分别为 W 像素、H 像素,影像帧数为 N,相邻像素 x 向物理间距为 x_s 毫米, y 向物理间距为 y_s 毫米,相邻两帧影像之间 z 向物理间距为 z_s 毫米;

[0010] 步骤 2:获取用户指定的直线段 AB,用于构造血管内超声长轴影像,直线段 AB 与 x 轴的夹角为 θ ;

[0011] 步骤 3:沿着三维血管内超声数据的第一帧影像所在平面进行四叉树均匀划分,使得叶子节点的大小为 $M \times M$;

[0012] 步骤 4:对四叉树从根节点开始递归遍历,并处理每个节点,包括:

[0013] 判断当前节点是否与直线段 AB 相交,如果当前节点与直线段 AB 相交并且当前节点不为叶子节点,则继续递归遍历;如果当前节点与直线段 AB 相交并且当前节点为叶子节点,则执行步骤 5;如果当前节点与直线段 AB 不相交则直接返回父节点继续遍历;

[0014] 步骤 5:计算叶子节点与直线段 AB 的交点 F、G 的 x、y 坐标,并同时得到第 M+1 帧影像上对应点交点 F'、G' 的 x、y 坐标,使得交点 F' 的 x 坐标等于交点 F 的 x 坐标,交点 G' 的坐标等于交点 G 的坐标,交点 F'、G' 的 z 坐标分别等于交点 F、G 的 z 坐标加上 $M \times z_s$;绘制平面 FF' G' G,同时将叶子节点所在的数据块以三维纹理的方式传输至图形处理单元处理;将上述过程重复 $\lfloor N/M \rfloor + 1$ 次,每次只需将交点 F、G、F'、G' 的 z 坐标分别加上 $M \times z_s$,对应的块数据随之更新即可;

[0015] 步骤 6:图形处理单元的片段着色器中对绘制平面 FF' G' G 所产生的每个片段进行三维的样条曲线插值。

[0016] 本发明的优点是:按照本发明提供的技术方案,使用基于 GPU 的计算架构,通过四叉树数据结构对三维 IVUS 数据进行分块处理,一方面解决数据量太大不能全部加载到 GPU 显存中的问题,另外一方面提高对 GPU 纹理缓存的利用效率,另外通过使用三维的样条曲

线插值,大大提高了图像的质量。本发明使用目前普通民用显卡提供的 GPU 计算架构,可以对 $512 \times 512 \times 4000 \times 16\text{bit}$ 这样大规模的 IVUS 影像数据,达到每秒钟 30 帧以上的高保真长轴影像构造速度。相比于已有的方法,本发明没有使用任何图像下采样方法来降低图像分辨率,并且完全在 GPU 中计算三维的样条曲线插值,相比于二维的最近邻插值或者双线性插值,可以得到更高的计算精度,同时仍然能够达到实时的计算速度。另外,本发明提供了一种新的基于四叉树对三维数据进行空间分块的方法,相比于传统的基于八叉树的方法,可以大大减少需要的额外存储量,另外也可以加速遍历树的速度。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 IVUS 长轴影像构造算法升级为下一代高保真的算法,将图像质量提升一个档次。

附图说明

[0017] 图 1 是传统 IVUS 长轴影像构造的示意图。

[0018] 图 2 是构造全局坐标系和视点坐标系的示意图。

[0019] 图 3 是四叉树分块处理的示意图。

[0020] 图 4 是判断指定直线段是否与某个叶子节点相交的示意图。

具体实施方式

[0021] 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。

[0022] 与传统的方法对每一帧加载三维血管内超声 (IVUS) 影像分别处理不同,本发明采用了基于三维图像插值的方法,因此把所有的 N 帧影像当做一个三维的体数据,把 IVUS 长轴影像构造看做一个三维平面对三维体数据的采样过程,这个需要合适的坐标系才能保证正确的计算。

[0023] 附图 2 给出了全局坐标系和视点坐标系的构造示意图,为了方便起见,全局坐标系的坐标原点 O 与第 1 帧影像的左下角重合,全局坐标系的 xoy 平面与第 1 帧影像所在的平面重合, x 轴正向由左至右, y 轴正向由下向上,其它 $N-1$ 帧影像垂直于 z 轴顺序排列, z 轴正向由第 1 帧指向第 N 帧,相邻两帧之间的 z 向距离: $z_s = \text{马达回拉速度} / \text{帧频}$ 。虽然 N 帧影像是离散的,但是通过图像插值算法可以重构出一个连续的三维体数据,因此在图中并没有画离散的每一帧,而是用了一个连续的长方体来表示这个三维的体数据。医生在第 1 帧上指定的直线段 AB 垂直于 z 轴投影可以确定一个平面 $ABDE$,沿着 $ABDE$ 平面均匀采样,就可以构造出长轴影像。

[0024] 确定了全局坐标系之后,还需要构造一个合适的视点坐标系,如果视点坐标系选择不合适的话,那么 $ABDE$ 平面投影到视平面上将会是一个平行四边形,而达不到传统的长轴构造的效果。本发明中选择 A 点作为视点坐标系的原点 o' , AB 作为视点坐标系的 x' 轴, AE 作为视点坐标系的 y' 轴,视点坐标系的 z' 轴按照左手定则,选择垂直于 $x' o' y'$ 平面且通过 o' 点的直线。另外,投影方式选择平行投影,视锥体分别通过 A 、 B 、 D 、 E 四个顶点。

[0025] 在上述构造坐标系的基础上,可以有一个很简单的方法来实现 IVUS 影像的长轴构造:首先将 N 帧 IVUS 影像作为一个三维纹理传输至图形处理单元 (GPU) 显存,然后分别

将 A、B、D、E 四个点的全局坐标作为它们的纹理坐标,通过绘制平面 ABDE,可以利用 GPU 硬件中的光栅化单元和三线性纹理插值单元来得到 IVUS 长轴影像。这个算法的原理和传统 CT、MRI(核磁共振)影像处理中的多平面重建(MPR)算法的原理是相同的,但是很遗憾,在大数据量的 IVUS 长轴影像构造中并不适用,因为典型的 IVUS 影像的数据量往往大于 2GB,而目前除了少数价格非常昂贵的专业级显卡之外,大部分显卡的显存都远远少于 2GB,目前主流的中高端显卡的显存一般在 256MB 到 1GB 之间,并且不能全部用于三维纹理。

[0026] 为了解决这个问题,本发明中把一个大的 IVUS 三维数据分成若干个 $M \times M \times M$ 的小块, M 一般选择为 2 的整数次方,并且尽量使得一个小块的数据能够完全被 GPU 的三维纹理缓存所容纳,因为 GPU 硬件的纹理缓存往往都比较小,因此 M 不能太大,如果 M 取值太大以至于纹理缓存不能包含一小块的数据,那么 GPU 在做计算的时候,就要大量地访问速度比较慢的纹理显存,速度会受到很严重的影响。在本发明中, M 取值 32,但是具体实施的时候,可以根据所用显卡纹理缓存的大小来适当地调节 M 的取值,以达到最快的运算速度。

[0027] 将数据分块以后,可以减少很多不必要的纹理显存,因为在所有的数据块中,和平面 ABDE 相交的只占很少一部分,而只是这很少的一部分需要传输到 GPU 中,去作进一步的计算。这样虽然解决了显卡显存不足的问题,但是另外一个问题又浮现出来:因为每个数据块都要和平面 ABDE 作相交计算,按照 $M = 32$ 来计算,对于一个 $512 \times 512 \times 4000$ 的数据集来说,总共需要的块数是 32000,计算 32000 个平面-长方体的求交计算需要非常大的计算量。传统上解决这类问题的办法是使用八叉树进行递归的分层次分块,首先将整个数据集从空间上分为大小相等的八个块,每个块是其父块的子节点,然后对每个块再递归地八等分,直到块大小小于等于 M 为止才停止划分。在绘制的时候,从根节点开始,自顶向下遍历八叉树,对于每个节点,如果其不跟平面 ABDE 相交,那么其所有的子节点也不可能跟平面 ABDE 相交,因此能够很快速地把不相关的块在很高的层次上就抛弃掉,大大减少需要进行求交计算的块数。

[0028] 但是基于八叉树的方法存在一些缺点:首先,八叉树需要很大的额外存储空间,还按照 $M = 32$ 为例,八叉树的叶子节点为 32000 个,以满八叉树计算,加上所需的中间节点,总共约需 37000 个节点的存储空间;其次,八叉树的递归遍历比较复杂和耗时;最后,使用八叉树就必然要进行平面-长方体的求交运算,而这个运算是个三维几何运算,也需要一定的复杂度。

[0029] 本发明中提供了一个新的基于四叉树递归分块的方法,四叉树只能用于二维空间的递归划分,之所以能够适用于三维空间的划分,完全是由于利用了 IVUS 长轴影像构造算法的特殊性所得到的。请参看附图 3,可以仅在第 1 帧影像上应用四叉树递归地分块,然后计算直线段 AB 和每个数据块所对应的二维的长方形区域求交,所求得的交点坐标可以直接应用于 z 向上下一个块,而无需在 z 向上递归分块。如图中 AB 和某个块相交于 F 点,那么垂直投影 F 点到第 $1+M$ 帧上,可以得到 F' 点,其 x, y 坐标和 F 点 x, y 坐标相同,仅 z 坐标不同: $z_{F'} = z_F + Z_s \times M$,其中 $z_{F'}$ 为 F' 点的 z 坐标, z_F 为 F 点的 z 坐标。通过这样的处理,每个块的大小仍然为 $M \times M \times M$,只不过通过遍历四叉树所计算得到的每个交点的坐标,在 z 向上都可以重复利用 $\lfloor N/M \rfloor + 1$ 次。

[0030] 使用基于四叉树的递归分块方法,具有如下优点:首先,四叉树所需要的额外存储空间非常小,还用上述 $M = 32$ 的例子,四叉树的叶子节点只需要 $(512/32)^2 = 256$,以满四叉

树计算,加上所需的中间节点,总共需要 341 个节点;其次,因为 341 个节点所占的存储空间很小,可以很容易地加载到 CPU 的 L1 Cache(一级缓存)中,另外四叉树的深度只有 4,这两个因素使得遍历小型的四叉树效率非常高;最后,使用四叉树避免了复杂的三维平面-长方体求交,而只需计算比较简单的二维直线段-长方形求交。

[0031] 附图 4 给出了计算直线段 AB 和任意一个四叉树的叶子节点所对应的长方形求交的示意图,长方形的四个顶点的坐标由四叉树的叶子节点所记录,不失一般性,假设分别为 (x_0, y_0) 、 (x_1, y_0) 、 (x_1, y_1) 、 (x_0, y_1) 。另外,假设一帧 IVUS 影像的宽度为 W 个像素,高度为 H 个像素(目前临床中使用最多的影像,其 W 和 H 均为 512 像素),相邻像素 x 向物理间距为 x_s , y 向物理间距为 y_s ,直线段 AB 和 x 轴的夹角为 θ ,则图像中心点 C 点的坐标为: $(C_x, C_y) = (W \cdot x_s/2, H \cdot y_s/2)$,直线段 AB 的参数方程可以表示为:

$$[0032] \quad \begin{cases} x(t) = C_x + t \cos \theta \\ y(t) = C_y + t \sin \theta \end{cases}$$

[0033] 其中 t 为参数, $x(t)$ 、 $y(t)$ 为直线段 AB 上任意一点的 x 、 y 坐标;其与直线 $x = x_0$, $x = x_1$ 的交点分别为: $(x_0, C_y + (x_0 - C_x) \tan \theta)$, $(x_1, C_y + (x_1 - C_x) \tan \theta)$,与直线 $y = y_0$, $y = y_1$ 的交点分别为: $(C_x + (y_0 - C_y) \cot \theta, y_0)$, $(C_x + (y_1 - C_y) \cot \theta, y_1)$,接着判断以下四个条件:

$$[0034] \quad \begin{cases} y_0 \leq C_y + (x_0 - C_x) \tan \theta \leq y_1 \\ y_0 \leq C_y + (x_1 - C_x) \tan \theta \leq y_1 \\ x_0 \leq C_x + (y_0 - C_y) \cot \theta \leq x_1 \\ x_0 \leq C_x + (y_1 - C_y) \cot \theta \leq x_1 \end{cases}$$

[0035] 如果四个条件都不满足,那么直线段 AB 和此叶子节点不相交,如果四个条件都满足,则直线段 AB 和此叶子节点相交,并且交点坐标也可由上述计算得到。当直线段和长方形相交并且直线段不经过长方形的四个顶点之一时,上述四个判断条件中有且只有两个满足,此时图 4 中的 F、G 两个点分别位于长方形的两条不同的边上;当直线段仅通过长方形的四个顶点之一,与其它的边都不相交时,上述四个判断条件中有两个满足,必然有重合的顶点,去掉重合的顶点,此时 F、G 两个点重合于相交的顶点;当上述四个判断条件中有三个满足时,此时必然是直线段通过长方形的某个顶点,但是同时又经过长方形的某条边,此时 F、G 两个点中有一个和相交的顶点重合;当上述四个判断条件同时满足时,只可能是直线段通过长方形的两个对角顶点,此时 F、G 两个点即为相交的两个顶点。不管在上述哪种情况之下,都可以算出 F、G 两个点的 x 、 y 坐标,而 F' 、 G' 两个点(参看附图 3)的 x 、 y 坐标是分别等于 F、G 的 x 、 y 坐标的,其 z 坐标分别是 F、G 的 z 坐标加上 $M \times z_s$ 得到的,此时将包含平面 $FF'G'G$ 的大小为 $M \times M \times M$ 的块数据作为三维纹理传输至 GPU 纹理显存,并在三维空间中绘制平面 $FF'G'G$,就完成了对一个数据块的处理,也得到了整个长轴影像的一部分。上述 F、G 的 x 、 y 坐标要重复利用 $\lfloor N/M \rfloor + 1$ 次,每次 z 坐标加上 $M \times z_s$,对应的块数据也随之更新,这样处理一个四叉树的叶子节点求交计算,就可以在 z 向上处理 $\lfloor N/M \rfloor + 1$ 个数据块,通过遍历整个四叉树,就可以处理完所有的 $(\lfloor W/M \rfloor + 1)(\lfloor H/M \rfloor + 1)(\lfloor N/M \rfloor + 1)$ 个数据块,每个数据块所包含的平面 $FF'G'G$,拼在一起看好组成了整个平面 ABDE,因此可以得到完整的长轴影像。

[0036] 上述四叉树遍历和求交运算都是在 CPU 中完成的,绘制平面 $FF'G'G$ 是利用

GPU 的光栅化硬件,把平面光栅化成为一系列离散的像素点,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 的片段着色器 (Fragment Shader) 作进一步的处理。GPU 的 FragmentShader 是一个可编程的器件,可以使用 GLSL、Cg、HLSL 或者 CUDA 等语言对其进行编程,以达到自己特定的计算目的。目前 GPU 硬件中同时存在多个 Fragment Shader,可以并行地对多个片段 (Fragment) 处理,因此可以大大提高计算速度。本发明中使用 GPU 的 Fragment Shader 对平面光栅化后的每个像素点进行三维样条插值计算,以同时保证图像的高质量和运算的高性能。

[0037] 在 GPU 的 Fragment Shader 中可以得到当前处理的像素点的屏幕坐标 (x_f, y_f, z_f) , 以及视点变换投影矩阵 M_{mvp} , 由它们可以计算出当前处理的像素点所对应的全局坐标 (x_g, y_g, z_g) , 计算公式如下:

$$[0038] \quad [x_g, y_g, z_g]^T = M_{mvp}^{-1} [x_f, y_f, z_f]^T$$

[0039] 因为图像插值需要在数据空间中进行,因此还需要计算数据空间的坐标 (x_v, y_v, z_v) , 计算公式如下:

$$[0040] \quad [x_v, y_v, z_v]^T = [x_g/x_s, y_g/y_s, z_g/z_s]^T$$

[0041] 并且满足:

$$[0042] \quad \begin{cases} 0 \leq x_v < W \\ 0 \leq y_v < H \\ 0 \leq z_v < N \end{cases}$$

[0043] 通过上述坐标变换计算得到的 x_v, y_v, z_v 不一定是整数,但是 IVUS 影像数据只提供了 $W \times H \times N$ 个离散的采样值,因此需要使用图像插值算法得到任意浮点数 (x_v, y_v, z_v) 坐标处的采样值。传统的算法为了保证计算速度,往往采用最近邻插值或者双线性插值算法,得到的图像质量比较差,本发明中使用精度更高的三维样条曲线插值算法,其计算公式为:

[0044]

$$I(x_v, y_v, z_v) = \sum_{k=-1}^2 \sum_{j=-1}^2 \sum_{i=-1}^2 I(\lfloor x_v \rfloor + i, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k) \cdot f(\lfloor x_v \rfloor + i - x_v, \lfloor y_v \rfloor + j - y_v, \lfloor z_v \rfloor + k - z_v)$$

[0045] 其中 $I(x, y, z)$ 为三维 IVUS 影像数据在第 z 帧上 (x, y) 坐标处的像素值, f 为三维样条曲面函数,为了计算简单,一般选择 f 使得其满足:

$$[0046] \quad f(x, y, z) = f(x) \cdot f(y) \cdot f(z)$$

[0047] 这样上述三维插值运算将能够分离成三个单独的一维插值运算,如下式所示:

[0048]

$$I(x_v, y_v, z_v) = \sum_{k=-1}^2 \left\{ \sum_{j=-1}^2 \left[\sum_{i=-1}^2 I(\lfloor x_v \rfloor + i, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k) \cdot f(\lfloor x_v \rfloor + i - x_v) \right] \right. \quad (1)$$

[0049]

$$\left. \cdot f(\lfloor y_v \rfloor + j - y_v) \right\} \cdot f(\lfloor z_v \rfloor + k - z_v)$$

[0050] 通过这样的分离计算,可以将三维的样条曲面函数求值化简为一维的样条曲线函数求值,并且可以按照先行后列最后帧这样的顺序来计算一维的插值,大大减少了计算量。不过即使这样,对一个 Fragment 的插值计算仍然需要 672 次乘法,64 次内存访问,如果所需的数据不在 CPU 的缓存内的话,那么内存访问是一个代价非常高的操作,因为 IVUS 影像数据的大容量,并且每次插值都需要访问连续四帧内的数据,因此 CPU 缓存机制在大

部分情况下是失效的。假设 IVUS 影像的大小为 $512 \times 512 \times 4000$, 为了高保真地构造长轴影像, 其分辨率至少应该为: $512 \times \sqrt{2} \times 4000$, 也就是说包含至少 2896310 个 Fragment, 而为了达到实时的更新速度, 一秒钟至少应该能够更新 30 次, 因此 1 秒钟需要的乘法个数为 $672 \times 2896310 \times 30$, 大于 580 亿次, 内存访问次数大于 50 亿次, 这在目前的个人电脑上是无法实现的, 因此尽管其图像质量的优越性, 现有的算法都没有使用三维样条曲线插值。

[0051] 本发明中充分利用 GPU 的单指令多数据 (SIMD) 运算能力, 去解决这一问题。GPU 硬件内部在存储纹理时, 往往对于 RGBA 四通道格式的纹理处理最有效, 每通道是 32 位的浮点数, 因此每个纹理像素 (Texel) 由 128bit 组成, 在计算时可以并行地同时处理 4 个浮点数的乘法和加法。通过仔细观察式 (1), 其可以重新写成长度为 4 的矢量点积的求和操作:

[0052]

$$I(x_v, y_v, z_v) = \sum_{k=-1}^2 \left[\sum_{j=-1}^2 \langle \vec{I}(j, k), \vec{f}_{x_v} \rangle \cdot f(\lfloor y_v \rfloor + j - y_v) \right] \cdot f(\lfloor z_v \rfloor + k - z_v) \quad (2)$$

[0053] 其中 (x_v, y_v, z_v) 分别为待处理的 Fragment 映射到数据空间的 x、y、z 坐标, i, j, k 为求和索引变量, 为加载三维血管内超声影像数据在第 z 帧上 (x, y) 坐标处的像素值, f 为样条曲线函数, $\vec{I}(j, k)$ 是长度为 4 的矢量:

[0054]

$$(I(\lfloor x_v \rfloor - 1, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k), I(\lfloor x_v \rfloor, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k),$$

[0055]

$$I(\lfloor x_v \rfloor + 1, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k), I(\lfloor x_v \rfloor + 2, \lfloor y_v \rfloor + j, \lfloor z_v \rfloor + k))$$

[0056] $\vec{f}_{x_v}$ 同样也是长度为 4 的矢量:

[0057]

$$(f(\lfloor x_v \rfloor - 1 - x_v), f(\lfloor x_v \rfloor - x_v), f(\lfloor x_v \rfloor + 1 - x_v), f(\lfloor x_v \rfloor + 2 - x_v))$$

[0058] $\langle, \rangle$ 是矢量点积运算。通过同样的操作, 式 (2) 中对 j 和 k 的求和也可以分别改写成成长度为 4 的矢量点积计算, 因此要计算式 (1), 总共需要 21 次长度为 4 的矢量点积, 使用图形处理单元的单指令多数据矢量点积运算, 一次完成一个长度为 4 的矢量点积, 可以通过 GPU 的 SIMD 指令在 21 个时钟周期中完成。关于内存访问, 因为之前已经将数据分成 $M \times M \times M$ 的小块, 其中每个小块都能加载到 GPU 的纹理缓存中, 因此理论上对每一个数据块, 只需要一次比较慢的显存访问, 以后所有的纹理访问, 都直接从缓存中获取。

[0059] 以上所述, 仅为本发明中的具体实施方式, 但本发明的保护范围并不局限于此, 任何熟悉该技术的人在本发明所揭露的技术范围内, 可理解想到的变换或替换, 都应涵盖在本发明的包含范围之内, 因此, 本发明的保护范围应该以权利要求书的保护范围为准。

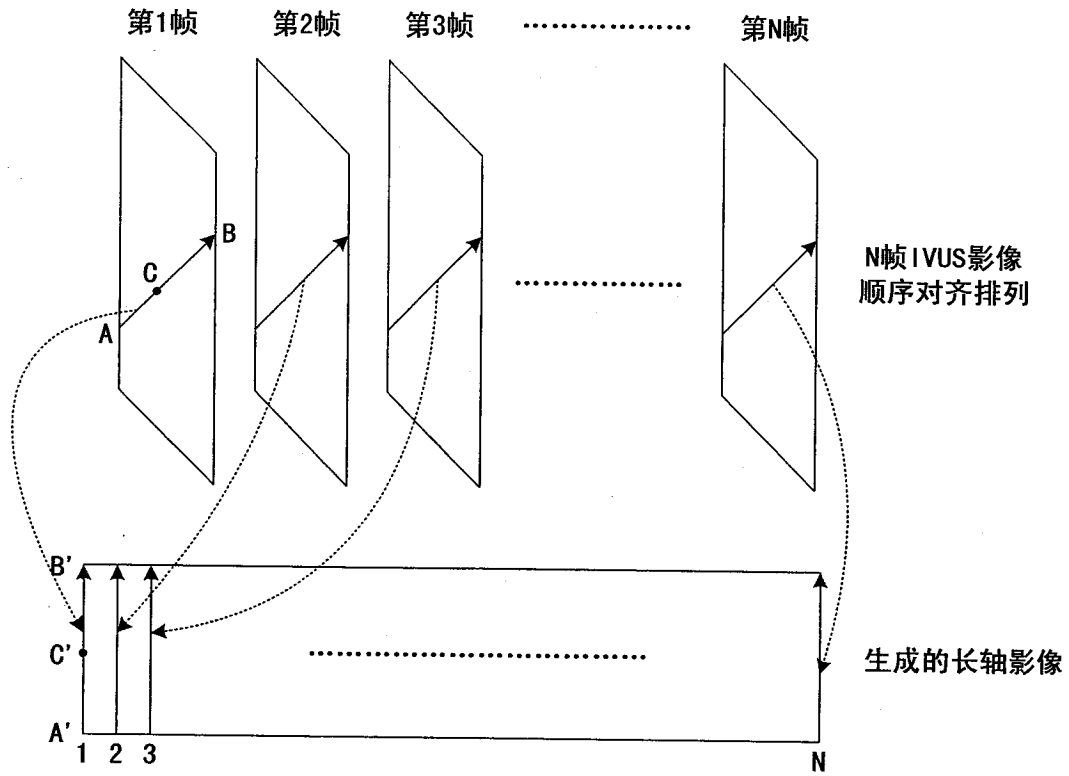


图 1

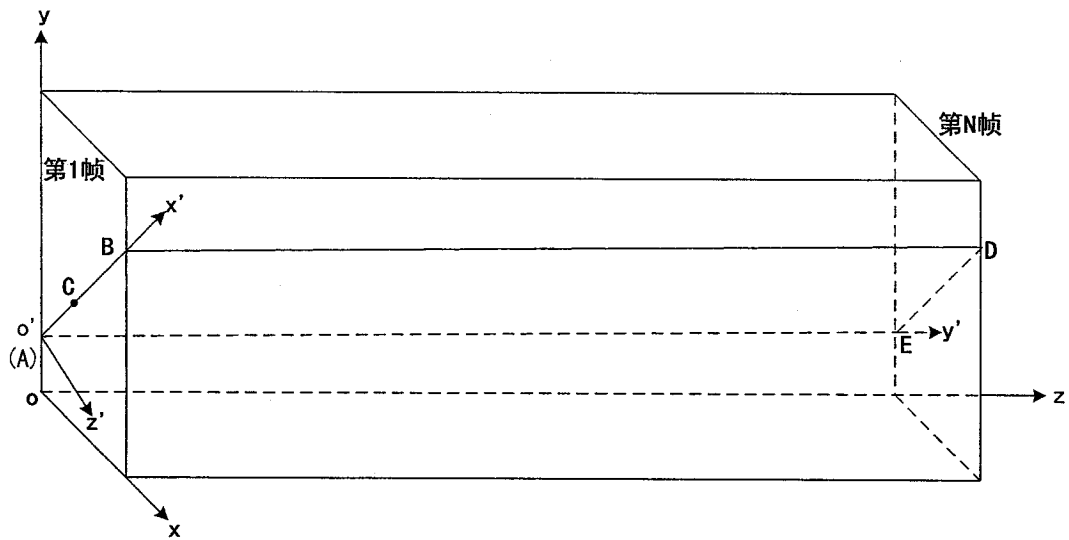


图 2

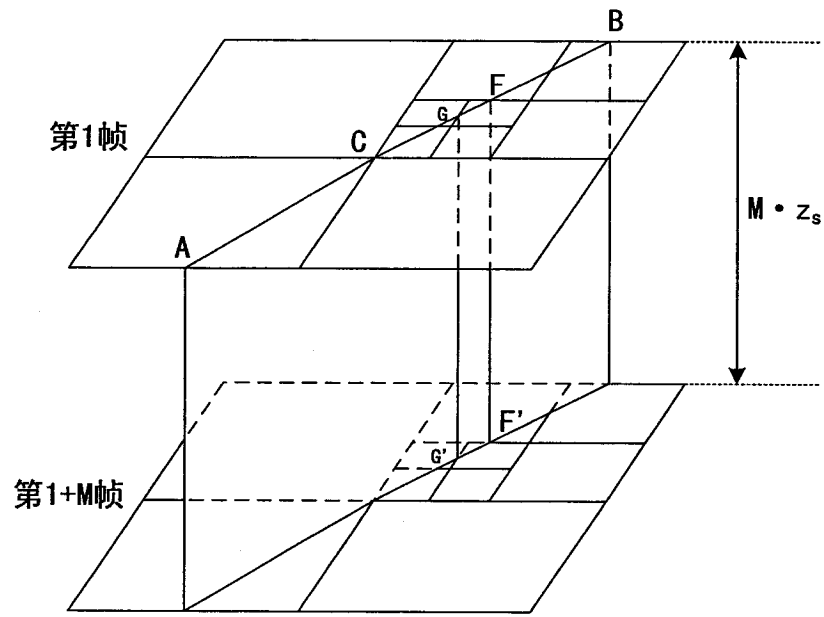


图 3

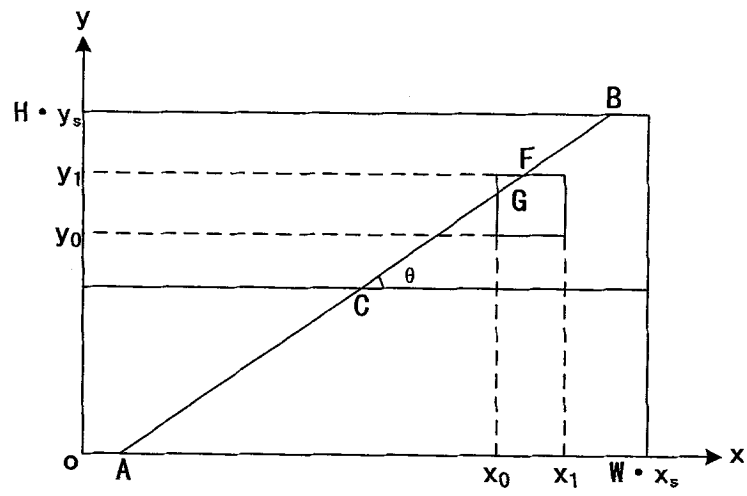


图 4

专利名称(译)	一种快速高保真地构造血管内超声长轴影像的方法		
公开(公告)号	CN101739661A	公开(公告)日	2010-06-16
申请号	CN200910242341.9	申请日	2009-12-09
[标]申请(专利权)人(译)	中国科学院自动化研究所		
申请(专利权)人(译)	中国科学院自动化研究所		
当前申请(专利权)人(译)	中国科学院自动化研究所		
[标]发明人	赵明昌 何晖光		
发明人	赵明昌 何晖光		
IPC分类号	G06T1/00 A61B8/12		
其他公开文献	CN101739661B		
外部链接	Espacenet SIPO		

摘要(译)

本发明公开了一种快速高保真地构造血管内超声长轴影像的方法，使用基于图形处理器的并行计算架构，通过四叉树数据结构对大数据量三维IVUS数据进行高效管理，递归遍历四叉树并且计算长轴影像所在平面与四叉树节点的交点坐标，将长轴影像分解为一系列小的平面，然后再送至图形处理器绘制并作插值计算。解决数据量太大不能全部加载到图形处理器显存中的问题，提高图形处理器纹理缓存的利用效率。将三维样条曲线插值使用图形处理器的单指令多数据矢量点积来计算，提高了运算速度和图像的质量。本发明在普通中高档显卡上，对512×512×4000×16bit这样大规模的IVUS影像数据，达到每秒钟30帧以上的高保真长轴影像构造速度。

