一种对光线和体素进行求交的方法

申请号：201110314930.0
申请日：2011-10-17

申请(专利权)人　中国科学院自动化研究所
地址　100190 北京市海淀区中关村东路95号
发明(设计)人　田捷　杨飞　向德辉　杨鑫

主分类号　G06T15/08(2011.01)I
分类号　G06T15/08(2011.01)I
公开(公告)号　102509341A
公开(公告)日　2012-06-20
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

(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 102509341 B

(45)授权公告日 2014.06.25

(21)申请号 201110314930.0

(22)申请日 2011.10.17

(73)专利权人 中国科学院自动化研究所

地址 100190 北京市海淀区中关村东路 95 号

(72)发明人 田捷 杨飞 向德辉 杨鑫

(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 15/08(2011.01)

(56)对比文件

CN 101527031 A,2009.09.09,

CN 101593345 A,2009.12.02,

US 2011206248 A1,2011.08.25,

审查员 刘琳

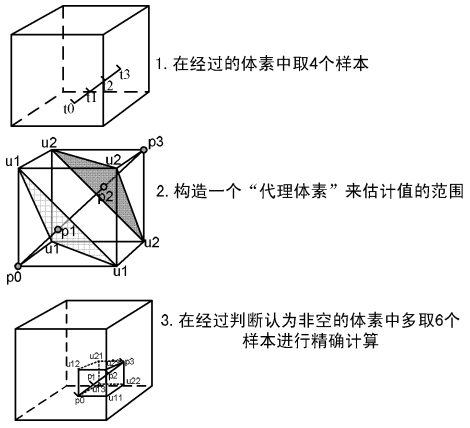
权利要求书1页 说明书8页 附图2页

(54)发明名称

一种对光线和体素进行求交的方法

(57)摘要

本发明是一种对光线和体素进行求交的方法,是通过 GPU 硬件对输入图像进行计算,得到图像数据的三线性插值并利用所述插值在光线经过的每个体素中沿光线等间距地获取四个样本;利用四个样本对体素做是否为空的判断;如果体素非空,通过 GPU 硬件提供的三线性插值重新获取六个样本,用四个样本和重新获取的六个样本计算图像数据的三线性插值沿光线变化函数的三次多项式系数,进而通过解三次方程对非空的体素进行计算,得到光线与体素的等值面的交点位置;如果体素为空,则直接跳过体素,不计算光线与体素的等值面的交点。此方法在保持精度的前提下得到更快的等值面可视化速度,可用于三维标量体数据的等值面、在科学计算和医学影像可视化领域。



CN 102509341 B

1. 一种对光线和体素进行求交的方法，其特征在于，所述求交的步骤包括：

步骤 S1：通过 GPU 硬件对输入图像进行计算，得到图像数据的三线性插值，利用图像数据的三线性插值在光线经过的每个体素中沿光线等间距地获取四个样本，所述四个样本分别位于光线的入体素点、出体素点和位于体素中的两个三等分点处；

步骤 S2：利用所述四个样本对体素做是否为空的判断，其步骤是：首先构造一个立方体结构的代理体素，将所述四个样本依次排列在代理体素的对角线上，通过代理体素中顶点值和内部值的线性关系，遵照三线性插值规则、以及代理体素的顶点值和内部值取值范围的关系估计光线段上图像数据的最大值及最小值，如果给定代理体素的等值面值在光线段上图像数据的最大值与最小值之间，则体素非空；如果给定代理体素的等值面值不在光线段上最大值与最小值之间，则体素为空；

步骤 S3：如果体素非空，通过 GPU 硬件提供的三线性插值重新获取六个样本，用所述四个样本和重新获取的所述六个样本计算图像数据的三线性插值沿光线变化函数的三次多项式系数，进而通过解三次方程对非空的体素进行计算，得到光线与体素的等值面的交点位置；如果体素为空，则直接跳过体素，不计算光线与体素的等值面的交点。

2. 根据权利要求 1 所述的对光线和体素进行求交的方法，其特征在于，所述代理体素的顶点值是：

$$\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 6 & 0 & 0 & 0 \\ -5 & 18 & -9 & 2 \\ 2 & -9 & 18 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

其中，$p_0 \sim p_3$ 是获取的所述四个样本，u_1u_2 是代理体素顶点上的图像数据值。

3. 根据权利要求 1 所述的对光线和体素进行求交的方法，其特征在于，所述的三次多项式系数表示为 A、B、C、D，用以下关系计算三次多项式系数：

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix}$$

其中获取的所述四个样本是 $p_0 \sim p_3$，代理体素顶点上的值 u_1 用 $(u_{11}+u_{12}+u_{13})/3$ 代替，代理体素顶点上的值 u_2 用 $(u_{21}+u_{22}+u_{23})/3$ 代替，所述 u_{11}，u_{12}，u_{13} 和 u_{21}，u_{22}，u_{23} 是重新获取的所述六个样本。

一种对光线和体素进行求交的方法

技术领域

[0001] 本发明属于计算机图形与可视化技术领域，涉及一种基于 GPU 的对光线和体素进行求交的方法，可用于三维标量体数据的等值面可视化。

背景技术

[0002] 三维标量体数据的等值面可视化技术可以帮助人们观察、挖掘三维标量体数据（如 CT 切片数据）中包含的信息，在科学计算可视化和医学影像可视化领域有重要的应用价值。

[0003] 在三维标量体数据的等值面可视化中，“原始数据”是需要被可视化的三维标量体数据，“体素”是由空间上相邻的 8 个原始数据点构成的立方体结构，每个原始数据点称为体素的一个顶点。在确定使用三线性插值方法的情况下，如果给定一个等值面值，则在满足一定条件的时体素中可能存在一个对应的等值面，“光线与体素求交”是光线和体素中所述等值面的求交。

[0004] 三维标量体数据的等值面可视化有多种实现方式，从计算顺序上分，有物体顺序的体素投影法 (Rottger et al.2000.Hardware-Accelerated Volume And Isosurface Rendering Based On Cell-Projection) 和图像顺序的光线跟踪法 (Parker et al.1998. Interactive Ray Tracing for Isosurface Rendering) 等，从采用的硬件平台来分，有基于 CPU 的实现和基于 GPU 的实现。由于硬件的差别，GPU 的实现和 CPU 的实现有很大差别。

[0005] 在 GPU 上实现的光线投射法有两个主要步骤，一是对光线经过的体素进行遍历（体素遍历），二是找到光线与等值面的交点（即“光线与体素求交”）。

[0006] 现有的对光线和体素进行求交的方法包括 Parker et al.1998 和 Marmitt et al.2004.Fast and Accurate Ray-Voxel Intersection Techniques for Iso-Surface Ray Tracing 等给出的精确求交方法和 Neubauer et al.2002.Cell-based first-hit ray casting. 给出的近似求交算法等。在现有的精确求交方法中，需要在每个体素中提取 8 个样本，分别位于体素的 8 个顶点处，这种方法对于中央处理器 (CPU) 计算已经比较优化，但是在图形处理器 (GPU) 环境下，由于硬件插值技术的存在，通过新的设计可以在大多数的体素中只提取 4 个样本，将效率做到更高。

发明内容

[0007] 本发明的目的是要求以使用三维数字差分分析器 (3DDDA,3D Digital Differential Analyzer) 体素遍历方法 (Amanatides et al.1987AFast Voxel Traversal Algorithm for Ray Tracing) 为前提，提出一种新的对光线和体素进行求交的方法。

[0008] 为达成所述目的，本发明提出新的基于图形处理器 (GPU) 的对光线和体素进行求交方法，所述求交的步骤包括：

[0009] 步骤 S1：通过 GPU 硬件对输入图像进行计算，得到图像数据的三线性插值，利用图像数据的三线性插值在光线经过的每个体素中沿光线等间距地获取四个样本；

[0010] 步骤 S2 :利用四个样本对体素做是否为空的判断 ;

[0011] 步骤 S3 :如果体素非空,通过 GPU 硬件提供的三线性插值重新获取六个样本,用四个样本和重新获取的六个样本计算图像数据的三线性插值沿光线变化函数的三次多项式系数,进而通过解三次方程对非空的体素进行计算,得到光线与体素的等值面的交点位置 ;如果体素为空,则直接跳过体素,不计算光线与体素的等值面的交点。

[0012] 优选实施例,所述的四个样本分别位于光线的入体素点、出体素点和位于体素中的两个三等分点处。

[0013] 优选实施例,对体素做是否为空的判断的计算步骤是 :首先构造一个立方体结构的代理体素,将四个样本依次排列在代理体素的对角线上,通过代理体素中顶点值和内部值的线性关系,遵照三线性插值规则、以及代理体素的顶点值和内部值取值范围的关系估计光线段上图像数据的最大值及最小值,如果给定代理体素的等值面值在光线段上图像数据的最大值与最小值之间,则体素非空 ;如果给定代理体素的等值面值不在光线段上最大值与最小值之间,则体素为空。

[0014] 优选实施例,所述代理体素的顶点值是 :

[0015] $$\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 6 & 0 & 0 & 0 \\ -5 & 18 & -9 & 2 \\ 2 & -9 & 18 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

[0016] 其中,$p_0 \sim p_3$ 是获取的四个样本,u_1u_2 是代理体素顶点上的图像数据值。

[0017] 优选实施例,所述的三次多项式系数表示为 A、B、C、D,用以下关系计算三次多项式系数 :

[0018] $$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix}$$

[0019] 其中获取的四个样本是 $p_0 \sim p_3$,代理体素顶点上的值 u_1 用 $(u_{11}+u_{12}+u_{13})/3$ 代替,代理体素顶点上的值 u_2 用 $(u_{21}+u_{22}+u_{23})/3$ 代替,所述 u_{11}, u_{12}, u_{13} 和 u_{21}, u_{22}, u_{23} 是重新获取的六个样本。

[0020] 本发明与现有技术比较的特点如下 :

[0021] 使用本发明的方法进行光线和体素中等值面的求交,对于大多数体素只需要利用到四个样本的值,而且实际实施中除了光线经过的第一个体素真正需要采四个样本值之外,后续的各个体素只需要采三个样本值即可,因为当前体素的出体素点是下一个体素的入体素点,在这一点上采到的样本值可以被当前体素和下一体素共用。大部分体素经过判空操作后直接跳过,而只有少数体素需要进一步的计算 ;而在以往的方法中,对于每个体素都需要取多于四个样本的值,因此本发明的方法在总的计算时间上大大缩短。

[0022] 在实现中,我们用四组不同的数据测试了此方法的速度,得到的效果如表格 1 所示。

[0023] 表格 1

[0024]

数据	传统的精确求交技术	本发明提出的技术
头部血管数据	24.6637	12.6307
腹部 CT 数据	38.8634	32.6459
头部 CTA 数据	27.9157	13.7991
腿部 CT 数据	110.673	61.8268

附图说明

[0025] 图 1 示意了一种基于 GPU 的对光线和体素进行求交的方法的主要步骤，其中包括文字描述中提到的样本点的位置。

[0026] 图 2 给出了本发明与传统精确求交方法在精度方面的比较。通过一个 3x3x3 的实验数据，进行局部放大来比较，证实在精度方面无差别。

[0027] 图 3 给出了宏观的等值面绘制效果。

具体实施方式

[0028] 为使本发明的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本发明进一步详细说明。

[0029] 本发明需要在具有可编程能力的图形处理器 (GPU) 硬件平台上实现。我们自己是在 NVIDIA GT240 显卡上通过计算统一设备架构 (CUDA) 接口实现的。在实现本方案之前，应首先参考背景技术中提到的文献实现一个基本的等值面绘制程序，其中体素遍历过程采用 3DDDA 体素遍历方法 (Amanatides et al. 1987A Fast Voxel Traversal Algorithm for Ray Tracing)。然后用本发明的方案代替其中原有的光线和体素求交部分。

[0030] 步骤 S1 ：在光线经过的体素里用 GPU 硬件提供的三线性插值功能沿光线取四个样本。

[0031] 在光线遍历过程中，使用以下的参数方程表示采样位置 $\vec{r}(t)$ ：

[0032] $$\vec{r}(t)=\vec{o}+\vec{v}t$$

[0033] 其中 $\vec{o}$ 是基本的等值面绘制中参考点坐标，$\vec{v}$ 是基本的等值面绘制中一条光线的单位方向向量，t 是基本的等值面绘制中的光线参数，$t = t_0, t_1, t_2, t_3$。

[0034] 使用 3DDDA 体素遍历方法可以依次定位出光线进入体素的光线参数值 t_0 和离开体素的光线参数值 t_3，为了获取四个样本，首先要确定四个样本所在的四个采样点位置 $\vec{r}_0,\vec{r}_1,\vec{r}_2,\vec{r}_3$。

[0035] 如图 1 中所示 1. 在经过的体素中取四个样本，利用两个所述的参数值 t_0, t_3 计算首末两个采样点的位置 $\vec{r}_0=\vec{r}(t_0)$，$\vec{r}_3=\vec{r}(t_3)$，并且以如下计算三等分点处的参数值 t_1, t_2 和坐标值 $\vec{r}_1,\vec{r}_2$ ：

[0036] $$t_1=\frac{2t_0+t_3}{3},t_2=\frac{t_0+2t_3}{3}$$

[0037] $$\vec{r}_1=\vec{r}(t_1),\vec{r}_2=\vec{r}(t_2)$$

[0038] 利用硬件提供的三线性插值功能，能取得采样点的位置$\vec{r}_0,\vec{r}_1,\vec{r}_2,\vec{r}_3$处的样本数据值 p_0，p_1，p_2，p_3，在计算统一设备架构（CUDA）接口中，采样工作可以通过图形处理器（GPU）端的 tex3D() 函数完成。

[0039] 对于每条光线，除了第一个体素以外，只需要读取 p_1，p_2，p_3 三个样本数据值，而样本数据值 p_0 能直接从上一个体素中的样本数据值 p_3 得到，它们在同一个采样点上。

[0040] 步骤 S2：对体素进行判空操作；

[0041] 如图 1 中所示 2. 构造一个代理体素来估计光线段上数据值的范围，将采样点的位置$\vec{r}_0 \sim \vec{r}_3$这段光线段置于一个代理体素当中，使光线段位于代理体素的体对角线上，此时光线段通过了代理体素的两个顶点 p_0，p_3，如图，设其余的六个顶点取两个相同的代理体素顶点上的值 u_1，u_2，由以下关系可以解出代理体素顶点上的值 u_1，u_2 使得样本数据值 p_1，p_2 的值能由三线性插值得到：

[0042] $$\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 6 & 0 & 0 & 0 \\ -5 & 18 & -9 & 2 \\ 2 & -9 & 18 & -5 \\ 0 & 0 & 0 & 6 \end{bmatrix}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

[0043] 有了代理体素顶点上的值 u_1，u_2，现在我们能确定，光线段通过区间内的体数据标量取值不会超出 [min(p_0,u_1,u_2,p_3),max(p_0,u_1,u_2,p_3)] 这个取值范围，因此，如果给定的等值面值，不妨设它为 iso，在此区间之外，则在这个区间不可能产生光线和等值面的交点，否则有可能。

[0044] 步骤 S3：在经过取值范围判断的认为非空的体素中取额外的六个样本，结合步骤 S1 中已经获取的四个样本进一步进行精确估计，计算光线与等值面的交点位置。

[0045] 如果经过上述的取值范围判断认为光线可能在当前体素与等值面产生交点，则如图 1 中所示 3. 在通过取值范围判断的体素中重新获取六个样本进一步进行精确计算，构造一个局部体素，在局部体素的顶点处采集重新获取的六个样本值 u_{11}，u_{12}，u_{13}，u_{21}，u_{22}，u_{23}，并用 $(u_{11}+u_{12}+u_{13})/3$ 代替 u_1，$(u_{21}+u_{22}+u_{23})/3$ 代替 u_2。这样做的目的是克服硬件插值可能带来的精度损失，而这些样本点的位置可以很容易地通过重新组合$\vec{r}_0\left(r_{0x},r_{0y},r_{0z}\right),\vec{r}_3\left(r_{3x},r_{3y},r_{3z}\right)$三个维度上个坐标 r_{0x}，r_{0y}，r_{0z}，r_{3x}，r_{3y}，r_{3z} 得到，如 u_{11} 的位置为 (r_{3x}, r_{1y}, r_{1z})。

[0046] 下一步，用 x 作为当前体素内光线段上的归一化参数（取值 0 ～ 1），其中：

[0047] $$x = \frac{t - t_0}{t_3 - t_0}$$

[0048] 在使用三线性插值的情况下，光线段上的数据标量值可以表示为 x 的一个三次函数 $f(x) = Ax^3+Bx^2+Cx+D$，用以下关系可以由 (p_0, u_1, u_2, p_3) 解出系数：

[0049] $$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_0 \\ u_1 \\ u_2 \\ p_3 \end{bmatrix}$$

[0050] 然后，使用割线法解三次方程 $Ax^3+Bx^2+Cx+D-iso = 0$ 得到最小的一个根 x_0，如果 $0 < x_0 < 1$ 则找到了光线与等值面的交点，位于$(1-x_0)\vec{r}_0 + x_0\vec{r}_3$处。后续工作包括使用 Phong

光照模型(Bui Tuong Phong.1975Illumination for computer generated pictures)计算光照和颜色值并返回等常规操作。

[0051] 以下是我们基于计算统一设备架构(CUDA)所作实现中的关键代码,可以在其他实现中参考,所述的关键代码如下所述:

[0052]

```
// 获取第一个体素中的号样本
float4 samples;
samples.x=d_GetScalarValue(pos1);
while(t1<tfar)
{
    float midVert1;
    float midVert2;
    float t2,t3,t4;
    while (t1<tfar)
    {
        t4=min(min(tMaxX,tMaxY),tMaxZ);
        if (t4>tfar) t4=tfar;
        // 遍历操作
        if (tMaxX<tMaxY)
        {
            if (tMaxX<tMaxZ) tMaxX+=tDeltaX;
            else tMaxZ+=tDeltaZ;
        }
        else
        {
            if (tMaxY<tMaxZ) tMaxY+=tDeltaY;
            else tMaxZ+=tDeltaZ;
        }
        // 获取号样本
        pos4=voxel_zero+voxel_unit*t4;
        samples.w=d_GetScalarValue(pos4);
        // 获取号样本
        t2=(t4-t1)*(1.0f/3.0f)+t1;
```

[0053]

```
            samples.y=d_GetScalarValue(voxel_zero+voxel_unit*t2);
            // 获取号样本
            t3=(t4-t1)*(2.0f/3.0f)+t1;
        samples.z=d_GetScalarValue(voxel_zero+voxel_unit*t3);
            // 计算代理体素顶点上的值
    midVert1=-(5.0f/6.0f)*samples.x+3.0f*samples.y-1.5*samples.z+(1.0f/3
.0f)*samples.w;
    midVert2=(1.0f/3.0f)*samples.x-1.5*samples.y+3.0f*samples.z-(5.0f/6.
0f)*samples.w;
            // 计算最大、最小值，判空
            if
(min(min(min(samples.x,samples.w),midVert1),midVert2)<c_isovalue &&
    max(max(max(samples.x,samples.w),midVert1),midVert2)>c_isovalue)
break;
            // 用当前的号样本作为下一个体素的号样本
            t1=t4;
            pos1=pos4;
            samples.x=samples.w;
            }
        if (t1>=tfar) break;
        // 对于非空体素重新取个样本
    midVert1=d_GetScalarValue(make_float4(pos4.x,pos1.y,pos1.z,1.0f));
    midVert1+=d_GetScalarValue(make_float4(pos1.x,pos4.y,pos1.z,1.0f));
    midVert1+=d_GetScalarValue(make_float4(pos1.x,pos1.y,pos4.z,1.0f));
    midVert2=d_GetScalarValue(make_float4(pos1.x,pos4.y,pos4.z,1.0f));
    midVert2+=d_GetScalarValue(make_float4(pos4.x,pos1.y,pos4.z,1.0f));
    midVert2+=d_GetScalarValue(make_float4(pos4.x,pos4.y,pos1.z,1.0f));
        // 计算三次多项式系数
        float a=-samples.x+midVert1-midVert2+samples.w;
        float b=3.0f*samples.x-2.0f*midVert1+midVert2;
```

[0054]

```
float c=-3.0f*samples.x+midVert1;
float d=samples.x-c_isovalue;
        float root;
        // 解方程求根
if (d_Solve_Cubic_Eq(a,b,c,d,root))
{
    float t=(t4-t1)*root+t1;
    return
d_GetColor(voxel_zero+voxel_unit*t,CameraDirection,increasing);
}
// 用当前的号样本作为下一个体素的号样本
t1=t4;
pos1=pos4;
samples.x=samples.w;
}
```

[0055] 图 2 给出了本发明与传统精确求交方法在精度方面的比较。通过一个 3x3x3 的实验数据，进行局部放大来比较，证实在精度方面无差别。

[0056] 图 3 给出了宏观的等值面绘制效果，这里采用的是比较接近实际应用的数据。在实际应用中，可以参考这些图像对系统的预期效果进行评估。

[0057] 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，都应涵盖在本发明权利要求书的包含范围之内。

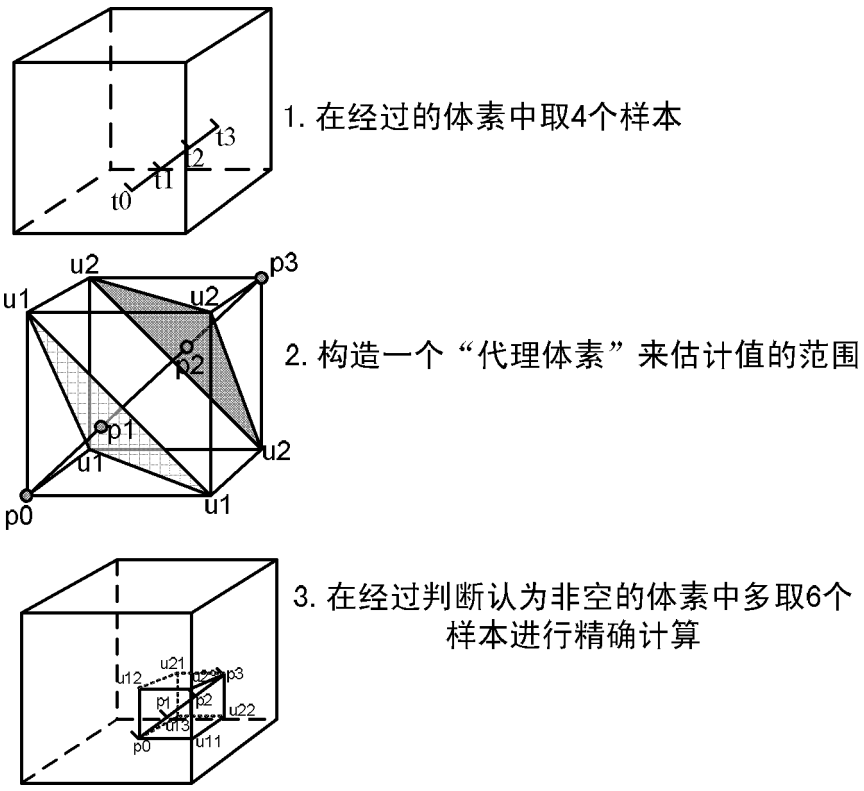

t3
t2
t1
t0
u2
p3
u1
u2
p2
p1
u2
u1
p0
u1
u21
u12
u22
u11
1. 在经过的体素中取4个样本
2. 构造一个“代理体素”来估计值的范围
3. 在经过判断认为非空的体素中多取6个样本进行精确计算

图 1

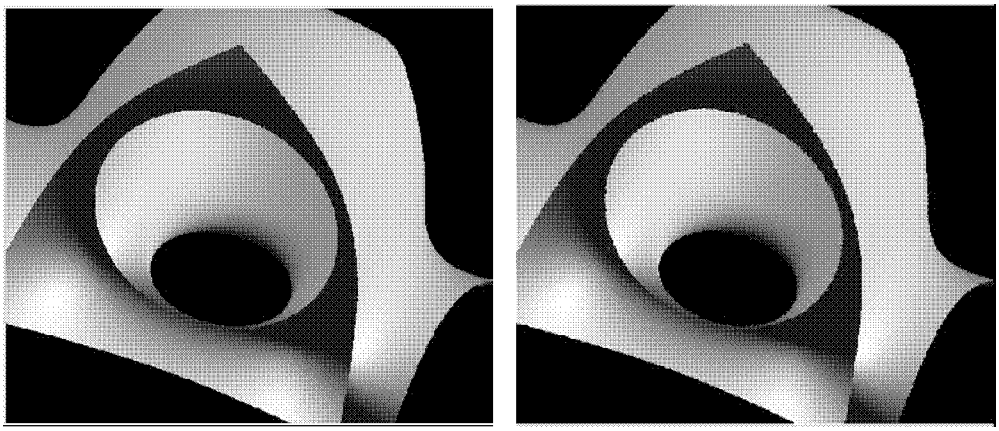


图 2

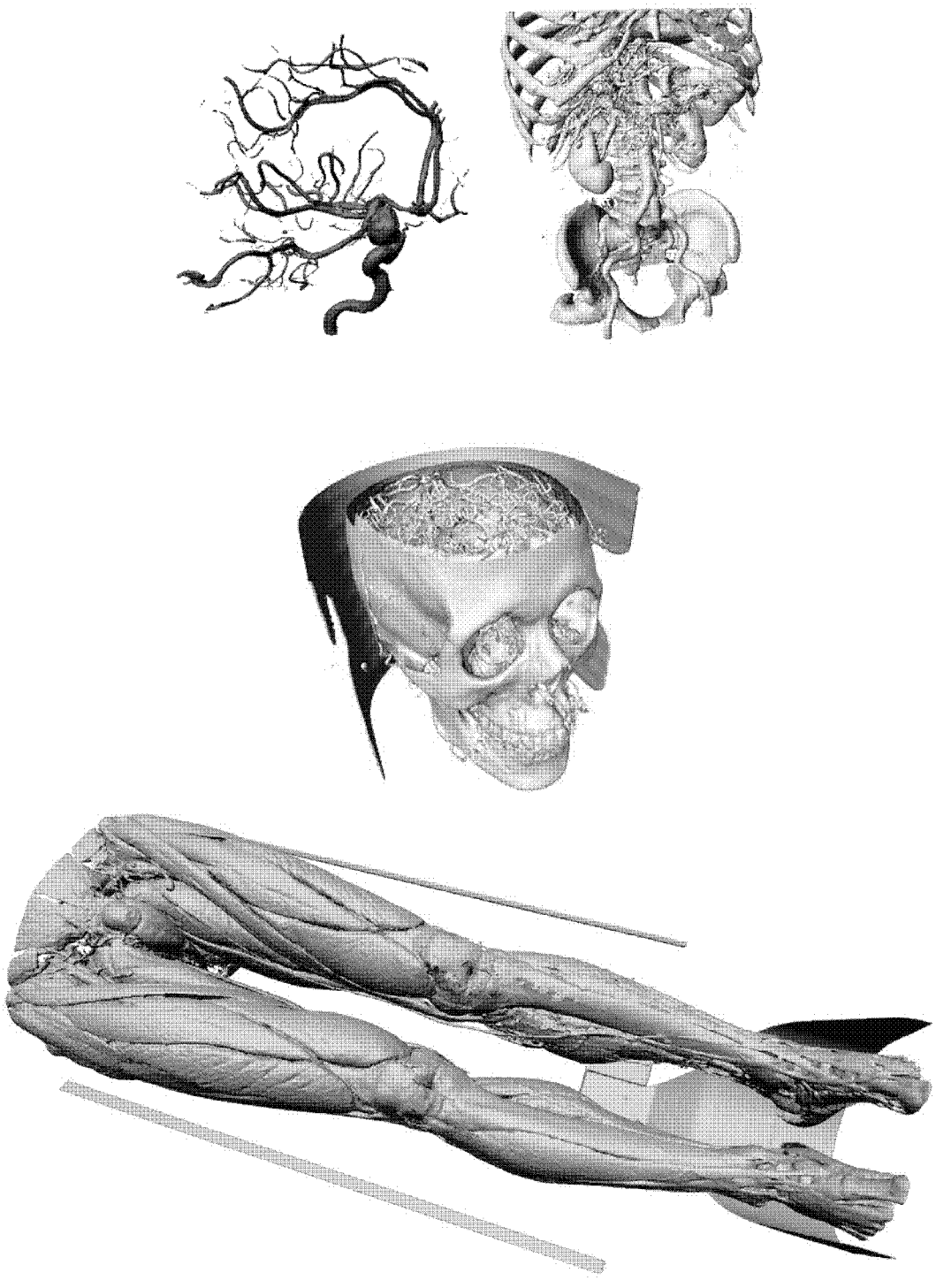


图 3