

基于 B 样条的 Level-Set GPU 演化算法

袁 斌

(北京应用物理与计算数学研究所 北京 100088)

摘 要 大部分 Level-Set 演化模型基于平均曲率或梯度,这对去除 3D 数据的噪声时保持线状特征是不利的;在解 Level-Set 演化方程时一般采用迎风格式,精度较低。设计了基于曲率差的高阶演化方程以及基于 B 样条和中心差分的混合 GPU 求解器。实验结果表明,基于曲率差的演化方程能够在光顺数据时保持线状特征。

关键词 B-样条, Level-Set, GPU, 曲率

中图分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.03.015

Level-Set GPU Evolution Algorithm Based on B-spline

YUAN Bin

(Institute of Applied Physics and Computational Mathematics, Beijing 100088, China)

Abstract Most of Level-Set evolution models are based on mean curvature or gradient, which don't facilitate preserving threadlike features while removing noise in 3D dataset. Upwind scheme is often used to solve Level-Set equation, which has lower precision. This paper designed high order Level-Set evolution equations based on curvature difference and hybrid GPU solver based on B-spline and central difference. Experimental results show that the level-set equation based on curvature difference preserves threadlike features while smoothing dataset.

Keywords B-spline, Level-Set, GPU, Curvature

1 引言

数值模拟(特别是基于粒子的数值模拟)所产生的数据场经常含有噪声。无网格数据转换成均匀网格时会产生噪声。电子显微镜(Electron Microscope, EM)、磁共振成像(Magnetic Resonance Imaging, MRI)和电子计算机断层扫描(Computed Tomography, CT)等测量结果中难免带有噪声。噪声的存在,使人分辨不清物质界面和特征,给数据分析带来了困难,因此需要对数据进行去噪处理。本文将针对均匀网格研究 Level-Set。

Osher 和 Sethian 提出的 Level-Set 方法^[1-3]把曲面(曲线)嵌入到 3D(2D)数据场,通过 3D(2D)数据场的演化实现曲面(曲线)的运动。数据场演化方程的基本形式为 $\frac{\partial \phi}{\partial t} = F \|\nabla \phi\|$, ϕ 为 Level-Set 函数, F 为速度在法向量上的投影。

Level-Set 方法可以用于去除噪声。用 Level-Set 方法去除噪声时,演化方程作用于整个数据场,而不仅仅作用于零水平集(Level-Set)的邻域。

Rosenfeld 给出多尺度特征(边界和曲线)的探测结果^[4]。Witkin 采用 Gauss 滤波实现 1 维多尺度滤波^[5],并给出规范的公式 $F(z, \sigma) = f(z) * g(z, \sigma) = \int_{-\infty}^{\infty} f(u) \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(z-u)^2}{2\sigma^2}} du$, 该公式容易推广到高维^[6]。核函数称为高斯函数。事实上,它是热扩散方程的解^[6]。从物理上讲,在热扩散的过程中,温度

的分布越来越平滑。扩散方程可以光顺图像,但不能保持线状特征。

Perona 和 Malik 给出各向异性的图像滤波方法 $\frac{\partial I}{\partial t} = \nabla \cdot (g'(\|\nabla I\|^2) \nabla I)$ ^[6], 简称为 P&M 方法。如果 g' 为常数,则得到各向同性的热传导方程。解该方程等价于对能量泛函 $\int_{\Omega} g(\|\nabla I\|^2) dx dy$ 求极小值。通过对图像的各向异性扩散,能光顺数据(去除噪声)并保持特征(edge)。Rudin 等给出 TV 模型^[7],该模型可以看作 P&M 方法的特例 $g(\xi) = 2\sqrt{\xi}$ 。

Alvarez 等人给出 $\frac{\partial \phi}{\partial t} = \|\nabla \phi\| (g(\|\nabla \phi\|) (\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|}))$ ^[8], 用于图像光顺和特征探测。Caselles 等给出 GAC(Geodesic Active Contour)模型^[9],它是黎曼空间的面积或长度极小模型 $\int_{\Omega} g(\|\nabla I\|^2) da$ ^[9],若 g 为常数,则其为欧几里德空间的 $\int_{\Omega} da$ 面积(或长度)极小模型。GAC 模型主要用于数据场分割。

大部分 Level-Set 方法基于平均曲率或梯度,这对在 3D 数据场去噪过程中保持线状特征是不利的。解 Level-Set 方程时一般采用迎风格式,这会带来一定的误差。B 样条插值可以达到较高的精度。一般 $k-1$ 阶偏微分方程的解是 $k-1$

次连续可微的,而三 k 次(张量积)B样条函数也是 $k-1$ 次连续可微的。在演化计算中引入张量积B样条插值是自然的选择。

高档GPU是众核(many-core)处理器,它具有很强的并行计算能力,如Nvidia Quadro FX 5600图形卡,其显存为1.5GB,拥有128核。CUDA^[10]是通用并行计算体系结构,利用CUDA可以很方便地在GPU上实现通用计算。每个进程在GPU上采用多线程进行并行计算,从而可以大大加速Level-Set演化过程。

本文针对3D数据场设计基于曲率函数的Level-Set演化模型,在GPU上实现基于B样条和均匀网格的Level-Set方程解法器。在基于B样条函数的演化过程中,每个时间步都需要反算控制标量。本文用Cuda实现多个Level-Set演化方程的解法器,并通过实验将其与其他方法进行比对。

2 初步分析

GAC模型原来用于活动轮廓线的演化(即数据场分割),后经改造可以用于去除噪声。GAC模型的方程为:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \|\nabla \phi\| \left(\nabla \cdot \left(g(\|\nabla I\|^2) \frac{\nabla \phi}{\|\nabla \phi\|} \right) \right) \\ &= \|\nabla \phi\| \left(g(\|\nabla I\|^2) \left(\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) + \nabla g(\|\nabla I\|^2) \cdot \nabla \phi \right) \end{aligned} \quad (1)$$

这里为了保持本文记号的一致,对文献[9]的表述稍作修改。

用 $\|\nabla \phi\|$ 代替 $\|\nabla I\|$,即

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \|\nabla \phi\| \left(\nabla \cdot \left(g(\|\nabla \phi\|^2) \frac{\nabla \phi}{\|\nabla \phi\|} \right) \right) \\ &= \|\nabla \phi\| \left(g(\|\nabla \phi\|^2) \left(\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) + \nabla g(\|\nabla \phi\|^2) \cdot \nabla \phi \right) \end{aligned} \quad (2)$$

本文称其为测地模型(Geodesic Model)。如果 $g=c$ 为常数,则该模型可简化为平均曲率模型 $\frac{\partial \phi}{\partial t} = \|\nabla \phi\| \left(\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right)$,它为面积(或长度)极小模型。式(1)中去掉

$\nabla g(\|\nabla \phi\|^2) \cdot \nabla \phi$ 后可简化为 $\frac{\partial \phi}{\partial t} = \|\nabla \phi\| \left(g(\|\nabla \phi\|^2) \left(\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) \right)$,该模型在GAC提出之前由Alvarez等人提出^[8]。本文称其为简化测地模型(Reduced Geodesic Model)。

平均曲率模型可以去除Level-Set上的“皱褶”,使Level-Set更光滑,但不能很好地保持特征。对3D数据场,测地模型和简化测地模型可以在光顺数据时保持物质界面,类似于各向异性的扩散模型。这些模型虽然可以保持物质界面,但不能保持物质界面上的线状特征。因此,研究能够保持三维数据场线状特征的新的演化模型具有重要意义。

平均曲率模型可以去除Level-Set上的“皱褶”,使Level-Set更光滑,但不能很好地保持特征。对3D数据场,测地模型和简化测地模型可以在光顺数据时保持物质界面,类似于各向异性的扩散模型。这些模型虽然可以保持物质界面,但不能保持物质界面上的线状特征。因此,研究能够保持三维数据场线状特征的新的演化模型具有重要意义。

3 算法

3.1 基于曲率差的Level-Set演化模型

本文研究3D数据场保持特征去噪算法。

在式(1)中用主曲率的函数 $F(\kappa_1, \kappa_2)$ 代替 $\|\nabla I\|$ (注:三维空间点 $A(x_0, y_0, z_0)$ 的主曲率是指过 A 点的等值面

($\phi^p(x, y, z) = C$)在点 A 的主曲率),可以得到新的演化模型。适当设计主曲率函数可以突出特征。作为特例,本文用主曲率之差 $|\kappa_1 - \kappa_2|$ 代替 $\|\nabla I\|$ 得到如下模型(称为曲率差模型, Curvature Difference Model):

$$\frac{\partial \phi}{\partial t} = \|\nabla \phi\| \left(g(|\kappa_1 - \kappa_2|^2) \left(\nabla \cdot \frac{\nabla \phi}{\|\nabla \phi\|} \right) + \nabla g(|\kappa_1 - \kappa_2|^2) \cdot \nabla \phi \right) \quad (3)$$

$|\kappa_1 - \kappa_2|^2 = \kappa_1^2 + \kappa_2^2 - 2\kappa_1\kappa_2 = (\kappa_1 + \kappa_2)^2 - 4\kappa_1\kappa_2$, 取 $g(\xi) = e^{-\frac{\xi}{\sigma^2}}$, 于是 $g(|\kappa_1 - \kappa_2|^2) = e^{-\frac{(\kappa_1 + \kappa_2)^2 - 4\kappa_1\kappa_2}{\sigma^2}}$, 不难证明,它使得

黎曼空间面积 $\int e^{-\frac{(\kappa_1 - \kappa_2)^2}{\sigma^2}} da$ (它是Level-Set上的积分)达到极小。曲率差越大,面积越小。线状特征曲率差较大,式(3)能够保持线状特征,它是高阶非线性演化方程。

时间均匀离散,时间步用 p 表示,时间步长用 Δt 表示;3D空间离散成直线网格,网格结点用 (i, j, k) 表示;约定任意标量函数 f (向量函数 f)在结点 (i, j, k) 上的值用 $f_{i,j,k}$ ($f_{i,j,k}$)表示。据此,得到式(3)的离散形式:

$$\phi_{i,j,k}^{p+1} = \phi_{i,j,k}^p + (g_{i,j,k}^p \mu_{i,j,k}^p \|\nabla \phi^p\|_{i,j,k} + (\nabla g^p)_{i,j,k} \cdot (\nabla \phi^p)_{i,j,k}) \Delta t \quad (4)$$

其中, $i=d, \dots, L+1; j=d, \dots, M+1; k=d, \dots, N+1; p=0, -1, \dots; g^p = e^{-\frac{(\mu^p)^2 - 4\gamma^p}{\sigma^2}}$, $\mu^p = \kappa_1^p + \kappa_2^p$, $\gamma^p = \kappa_1^p \kappa_2^p$, κ_1^p 与 κ_2^p 分别表示第 p 步主曲率函数。

在演化过程中,采用B样条^[11-13]函数提高演化精度。在三维情况下,分别用 $N_{i,d}^z(x), i=0, L; N_{j,d}^y(y), j=0, M; N_{k,d}^x(z), k=0, N$ 表示定义于结点向量 $(x_0, x_1, \dots, x_{L+d+1}), (y_0, y_1, \dots, y_{M+d+1}), (z_0, z_1, \dots, z_{N+d+1})$ 上的 $d(d \geq 3)$ 次B样条基函数。在演化过程中,针对每个时间步 p (这里是人工时间),令 $\phi^p(x, y, z) = \sum_{k=0}^N \sum_{j=0}^M \sum_{i=0}^L c_{i,j,k}^p N_{i,d}^z(x) N_{j,d}^y(y) N_{k,d}^x(z), (x, y, z) \in [x_d, x_{L+1}] \times [y_d, y_{M+1}] \times [z_d, z_{N+1}]$ 。 $c_{i,j,k}^p$ 称为B样条系数。设 $(x, y, z) \in [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$, 根据B样条基函数的局部支撑性,得到B样条函数的分片表示形式:

$$\phi^p(x, y, z) = \sum_{k=k-d}^k \sum_{j=j-d}^j \sum_{i=i-d}^i c_{i,j,k}^p N_{i,d}^z(x) N_{j,d}^y(y) N_{k,d}^x(z) \quad (5)$$

采用式(5)进行实际计算。 ϕ^p 是定义于3D直线网格上的张量积B样条函数,它在网格结点 (i, j, k) 上的值等于 $\phi_{i,j,k}^p, i=d, \dots, L+1; j=d, \dots, M+1; k=d, \dots, N+1$ 。 ϕ^p 的一阶偏导数、二阶偏导数或更高阶导数、曲率可以精确计算。基于B样条的演化计算,计算量较大,适合于用GPU实现。

要保持 ∇g^p 的连续性,应该用三五次(张量积)B样条函数或三四次(张量积)B样条函数表示 ϕ^p 。为了减少计算量,本文用三三次B样条函数表示 ϕ^p ,预先计算每个 $g_{i,j,k}^p$,并保存在缓冲区中,这样在计算每个 $(\nabla g^p)_{i,j,k}$ 时采用中心差分(Central difference)。在具体实现时,每步演化分成3个阶段:

Step1 反算B样条系数;

Step2 基于三三次B样条计算每个 $g_{i,j,k}^p$,并保存在缓冲区中;

Step3 对网格的每个结点 (i, j, k) , 基于三三次 B 样条计算 $\mu_{i,j,k}^p \parallel (\nabla \phi^p)_{i,j,k} \parallel$, 采用中心差分计算 $(\nabla g^p)_{i,j,k}$, 在此基础上计算 $\phi_{i,j,k}^{t+1}$.

这是 B 样条和差分结合的混合方法。

在演化过程中, 采用三三次 B 样条插值, ϕ^p 可以达到四阶精度, $\phi_x^p, \phi_y^p, \phi_z^p, \phi_{xx}^p, \phi_{yy}^p, \phi_{zz}^p$ 可以达到三阶精度, $\phi_{xx}^p, \phi_{yy}^p, \phi_{zz}^p$ 可以达到二阶精度, ∇g^p 可以达到二阶精度。因此, 该算法可以达到二阶精度。

式(3)去掉 $\nabla g(|\kappa_1 - \kappa_2|^2) \cdot \nabla \phi$ 后简化为:

$$\frac{\partial \phi}{\partial t} = \parallel \nabla \phi \parallel (g(|\kappa_1 - \kappa_2|^2) (\nabla \cdot \frac{\nabla \phi}{\parallel \nabla \phi \parallel})) \quad (6)$$

本文称其为简化曲率差模型, 它也可以突出线状特征。

3.2 GPU 实现

本文针对均匀数据场用 Cuda 在 GPU 上实现式(4)的演化算法, 每一时刻的演化函数 ϕ^p 用定义于均匀网格上的三三次均匀 B 样条函数(张量积形式)表示。每步演化都分成 3 步, 下面给出实现细节。

Step1 反算 B 样条系数, 可以分别沿着 x, y, z 3 个方向反算 B 样条系数; 在沿某个方向进行计算时, 沿该方向的数据具有相关性, 在其他方向上的数据没有相关性, 具有很好的并行性, 适合用 GPU 实现; 计算 B 样条函数需要一层虚网格。在实际计算中, 初始值为原始数据, 边界条件为外边界值等于边界值。在已知样值时, B 样条系数存在线性依赖关系, 这些关系可以用三对角方程组表示, 这是强对角占优的三对角方程组。解该方程组时, 采用直接方法(追赶法)而不是迭代方法。迭代解法是理论上近似的解法, 直接解法是理论上精确的解法。在用追赶法解三对角方程时, 先进行 LU 分解, 然后计算消元和回代过程。由于 ϕ^p 用张量积形式表示, 沿同一方向所有扫描行的三对角方程的系数矩阵完全相同, 在演化过程中, 这些系数也不变化。因此, 在整个演化过程中, 每个方向只需要计算一次 LU 分解。

图 1 给出网格 $G(L \times M \times N)$, G 沿 z 方向的投影网格为 G_z (尺寸为 $L \times M$), G_z 每个结点对应 G 中 z 方向一行结点, 称为 z 扫描行。 G 沿 y 方向的投影网格为 G_y (尺寸为 $M \times N$), G_y 每个结点对应 G 中 y 方向的一行结点, 称为 y 扫描行。 G 沿 x 方向的投影网格为 G_x (尺寸为 $M \times N$), G_x 每个结点对应 G 中 x 方向的一行结点, 称为 x 扫描行。在沿某个方向反算 B 样条系数时, 相应扫描行之间具有独立性, 这种独立性称为行独立性。因此具有并行性。每个 GPU 线程可以计算一个相应方向扫描行。

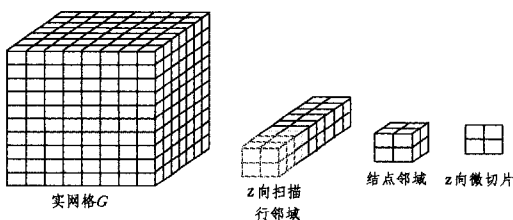


图 1 网格

Step2 在反算 B 样条系数之后, 对网格的每个结点 (i, j, k) 计算 $g_{i,j,k}^p$. G 中每个结点是有独立性, 称为(网格)结点独立性。如果每个结点对应一个线程, 则大规模数据的线程

太多, 且不利于优化计算。本文选择沿 z 方向计算, 每个 GPU 线程可以计算一个 z 扫描行。计算 $g_{i,j,k}^p$, 需要计算一阶导数、二阶导数值, 取出以该结点 (i, j, k) 为中心的 $2 \times 2 \times 2$ 邻域网格结点上的 B 样条系数。该邻域由 3 个 z 向切片组成, 每个切片是 2×2 网格, 称为 z 向微切片。对每个结点进行上述计算时, 在每个 z 向微切片内, 先沿 x 方向计算, 再沿 y 方向计算; 3 个切片计算完成后, 沿 z 方向计算。后面的结点的邻域与当前结点的邻域共享两个 z 向微切片。如果计算每个切片时把中间结果保存下来, 完成当前结点的计算后, 后面的结点只需要计算一个微切片。计算每个扫描行时, 开始计算 3 个微切片, 以后每次只计算 1 个微切片。这样计算过程便得到。

Step3 这一步在 GPU 上的具体实现类似于 Step2。

算法过程如下:

```

初始数据保存在缓冲区 A 中;
for(p=0; p<NumberOfEvolvingSteps; p++){
根据缓冲区 A 中的数据, 先沿 z 方向计算每个 B 样条系数, 结果保存在缓冲区 B 中;
在缓冲区 B 中, 沿 y 方向计算 B 样条系数;
在缓冲区 B 中, 沿 x 方向计算 B 样条系数;
在缓冲区 B 中, 沿 x 方向填充虚网格;
在缓冲区 B 中, 沿 y 方向填充虚网格;
在缓冲区 B 中, 沿 z 方向填充虚网格;
计算每个  $g_{i,j,k}^p$ , 并保存在缓冲区 C 中;
在缓冲区 C 中, 沿 x 方向填充虚网格;
在缓冲区 C 中, 沿 y 方向填充虚网格;
在缓冲区 C 中, 沿 z 方向填充虚网格;
对网格的每个结点  $(i, j, k)$ , 计算  $\mu_{i,j,k}^p \parallel (\nabla \phi^p)_{i,j,k} \parallel$  和  $\nabla g_{i,j,k}^p$ , 在此基础上计算  $\phi_{i,j,k}^{t+1}$ , 结果保存在缓冲区 A 中;
}
    
```

本文用 Cuda 实现了曲率差模型(见式(3))和简化曲率差模型(见式(6))的基于 B 样条的 GPU 解法器。为了便于比较, 还用 Cuda 实现了测地模型、简化测地模型、平均曲率模型、扩散模型和各向异性扩散模型的基于 B 样条的 GPU 解法器。

4 测试结果

所提算法在装有 NVidia nvs 4200M 显卡的 Intel i5 机器上进行测试。绘制采用基于曲率的 GPU 光线投射算法, 颜色色调是曲率的函数, 色调由蓝色经绿色变到红色表示曲率由小变大^[14]。不透明度是标量的函数, 必要时采用梯度量调制, 测试中, 对各项异性扩散模型, $g'(\xi) = e^{-\frac{\xi}{\sigma^2}}$; 对其他与 g 相关的模型, $g(\xi) = e^{-\frac{\xi}{\sigma^2}}$; 每个图采用统一的转换函数和视点。在以下测试结果中, 演化步数 $n = 100$, t 为演化时间, 单位为毫秒(ms)。

图 2 包含多个演化模型对 MRL_Head $256 \times 256 \times 256$ 数据的演化结果, 颜色色调表示主曲率平方和。采用梯度量调制。演化步数均为 100。图 2(a)为原始数据, 表面不光顺; 由于扩散作用, 图 2(b)线状特征变粗, 表面不光顺; 图 2(c)为当 $\sigma = 1$ 时各向异性扩散的结果, 表面不光顺, 几乎与原始数据相同; 图 2(d)为当 $\sigma = 50$ 时各向异性扩散的结果, 表面有较

大区域不光顺,放大特征,线状特征两端变粗,中间变弱,不能很好地保持线状特征(正确的情况是,应该尽可能保持原特征的尺寸),与图 2(a)相比增加了噪声;图 2(e)为平均曲率模型的演化结果,水平线消失;图 2(f)为当 $\sigma=1$ 时简化测地模型的结果,表面不光顺,几乎与原始数据相同;图 2(g)为当 $\sigma=50$ 时简化测地模型的结果,水平线消失;图 2(h)为当 $\sigma=1$ 时测地模型的结果,表面不光顺,有许多颗粒状特征;图 2(i)为当 $\sigma=50$ 时测地模型的结果,大部分区域不光顺,有许多颗粒状特征,水平线状特征两头变粗,中间变弱,与图 2(a)相比增加了噪声;图 2(j)为当 $\sigma=1$ 时简化曲率差模型的演化结果,表面比较光顺,而且保持线状特征;图 2(k)为当 $\sigma=50$ 时简化曲率差模型的结果,表面比较光顺,线状特征消失;图 2(l)为当 $\sigma=1$ 时曲率差模型的结果,表面比较光顺,而且保持线状特征;图 2(m)为当 $\sigma=50$ 时曲率差模型的结果,表面比较光顺,线状特征消失。从图 2(e)、图 2(g)、图 2(j)、图 2(k)、图 2(l)、图 2(m)可以看出,基于 B 样条的 Level-Set 演化算法是有效的。

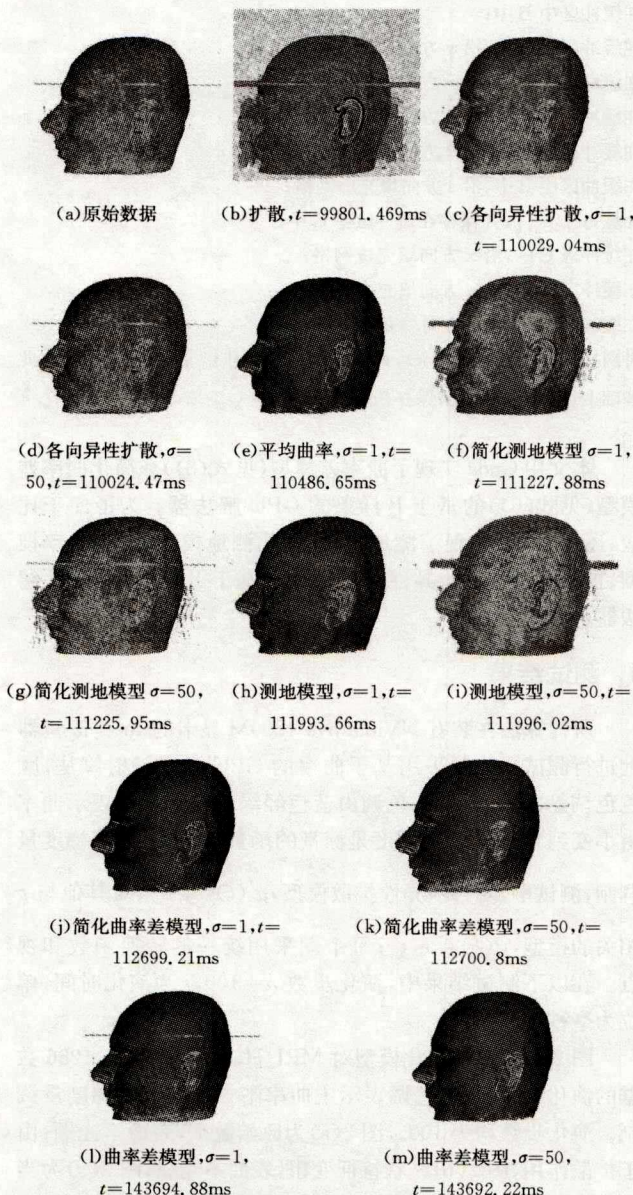


图 2 MRI_Head $256 \times 256 \times 256$ 数据

结束语 本文给出基于曲率函数的 Level-Set 高阶演化模型,设计实现基于 B 样条的 GPU 解法器,并与其他模型进行比对。实验证明,基于 B 样条的 Level-Set 演化算法是有效的;基于扩散的演化方程容易放大特征尺寸;测地模型会增加新的特征;这些新的特征可能遮挡原来的特征,基于平均曲率的演化模型可能淡化或丢失一些特征;基于曲率差的演化模型既可以光顺数据,又可以保持一些线状特征。

参考文献

- [1] OSHER S, SETHIAN J. Fronts propagating with curvature dependent speed; Algorithms based on hamilton-jacobi formulation [J]. Journal of Computational Physics, 1988, 79(1): 12-49.
 - [2] MALLADI R, SETHIAN J A, VEMURI B C. Shape modeling with front propagation; A Level-Set approach [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(2): 158-175.
 - [3] OSHER S, FEDKIW R. Level-Set methods; an overview and some recent results [J]. Journal of Computational Physics, 2001, 69(2): 463-502.
 - [4] ROSENFELD A. Edge and Curve Detection for Visual Scene Analysis [J]. IEEE Transactions on Computers, 1971, C-20(5): 562-569.
 - [5] WITKIN A P. Scale-space filtering; a new approach to multi-scale description [C] // IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'84). Piscataway, NJ, USA: IEEE Press, 1984: 150-153.
 - [6] PERONA P, MALIK J. Scale space and edge detection using anisotropic diffusion [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(7): 629-639.
 - [7] RUDIN L, OSHER S, FATEMI E. Nonlinear total variation based noise removal algorithms [J]. Physica D Nonlinear Phenomena, 1992, 60(1-4): 259-268.
 - [8] ALVAREZ L, LIONS P L, MOREL J M. Image selective smoothing and edge detection by nonlinear diffusion. II [J]. SIAM Journal on Numerical Analysis, 1992, 29(3): 845-866.
 - [9] CASELLES V, KIMMEL R, SAPIRO G. geodesic active contours [J]. International Journal of Computer Vision, 1997, 22(1): 61-79.
 - [10] NVIDIA. NVIDIA_CUDA_Programming_Guide [OL]. http://developer.nvidia.com/object/cuda_3_2downloads.html.
 - [11] DE BOOR C. On Calculation with B-spline [J]. Journal of Approximation Theory, 1972, 6: 50-62.
 - [12] COX M G. The numerical evaluation of B-Spline [J]. IMA Journal of Applied Mathematics, 1972, 10(2): 134-149.
 - [13] 施法中. 计算机辅助几何设计与非均匀有理 B 样条 [M]. 北京: 北京航空航天大学出版社, 1994.
 - [14] YIAN B. GPU Raycasting based on curvature [J]. Journal of Graphics, 2012, 33(6): 24-30 (in Chinese)
- 衰减. 基于曲率的 GPU 光线投射 [J]. 图学学报, 2012, 33(6): 24-30.