引文格式：胡森,高苏,蔡忠亮,等. 协同CPU和GPU的核密度估计及其可视化算法[J]. 地理空间信息, 2024, 22(6): 29-33. doi:10. 3969/j. issn. 1672-4623.2024.06.007

协同CPU和GPU的核密度估计及其可视化算法

胡　森[1]，高　苏[2]，蔡忠亮[1*]
（1. 武汉大学 资源与环境科学学院，湖北 武汉 430079；
2. 云南省地图院，云南 昆明 650034）

摘　要：大数据时代背景下，空间数据点规模越来越大，图像分辨率越来越高，使用CPU计算核密度估计结果并对其可视化的效率越来越低，难以满足应用对实时性的需求。针对该问题，提出了一种协同CPU和GPU的核密度估计及其可视化算法，该算法结合CPU的控制能力、GPU的并行计算能力以及OpenGL中的核心模式，并借助显存映射，同时优化了核密度估计的计算和可视化2方面。实验结果表明，相较于CPU并行和串行算法，该算法的执行效率分别提高了约5倍和20倍，且随着图像分辨率的提高，加速比呈现逐步上升的趋势。

关键词：核密度估计；可视化；GPU；OpenGL；统一计算架构
中图分类号：P208　　**文献标志码：**B　　**文章编号：**1672-4623（2024）06-0029-05

Kernel Density Estimation and Its Visualization Algorithm Combining CPU and GPU

HU Sen[1], GAO Su[2], CAI Zhongliang[1]
(1. School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China;
2. Yunnan Provincial Mapping Institute, Kunming 650034, China)

Abstract: In the context of the big data era, the scale of spatial data points is getting larger and larger, and the image resolution is getting higher and higher, thus, the efficiency of using CPU to calculate kernel density estimation result and visualize it is getting lower and lower. It's difficult to meet the real-time requirements of applications. Aiming at this problem, we proposed a kernel density estimation and visualization algorithm combining CPU and GPU. By utilizing the control ability of CPU, the parallel computing ability of GPU and the core mode in OpenGL, this algorithm optimizes both computational and visual aspects of kernel density estimation. The experimental results show that compared with CPU based parallel and serial algorithms, the efficiency of this algorithm is increased by about 5 times and 20 times respectively, and with the improvement of image resolution, the acceleration ratio shows a gradual upward trend.

Key words: kernel density estimation, visualization, GPU, OpenGL, CUDA

核密度估计基于有限的样本数据估算总体数据的分布情况，其符合地理学第一定律[1]，对计算结果进行可视化分析，可以直观地获取空间点的分布特征，是研究空间点分布中最常用地方法之一[2]。目前已被广泛应用于院前急救时空热点分析、交通事故热点分析、城市设施兴趣点分布热点检测、轨迹线热力图生成等领域[2-5]。

人类社会已进入大数据时代，社交媒体如微博、推特，定位系统如GPS、北斗以及医疗卫生等产生的海量地理空间数据[6]，向研究人员提供了新的机遇和挑战。如何快速甚至实时的处理并分析海量的时空数据，洞察数据背后的意义，在很大程度上依赖于对计算资源的充分利用[7]。计算机中用于计算的两大元件是中央处理器（CPU）和图形处理器（GPU），其中CPU[8]是计算机的大脑，能够处理很复杂或需要用户交互的工作，但只能同时运行少量任务，GPU具备成百上千个计算核心，能够同时运行大量任务，这种强大的并行计算能力是游戏引擎、人工智能、自动驾驶、计算机视觉、高性能计算等技术的根基。实际证明，采取CPU与GPU相结合的主机端—客户端架构，使用CPU控制程序的运行，使用GPU执行具体的计算，能够最大限度的利用计算资源[9]。

核密度估计及其可视化研究，包括核密度估计的计算和可视化两方面。目前，空间数据点规模越来越

收稿日期：2023-02-07。
项目来源：国家重点研发计划资助项目 (2021YFB2501101)。
第一作者简介：胡森（1999—），硕士研究生，研究方向为高性能地理计算，E-mail: 2309780282@qq.com。
通信作者：蔡忠亮（1971—），博士，教授，研究方向为地图制图自动化，E-mail：zlcai@whu.edu.cn。

大、图像分辨率越来越高，现有方法难以满足应用对实时性的要求，针对该问题，国内外研究者提出了一些解决方案，核密度估计的计算方面，郭宇达[6]等使用Spark并行计算框架，大幅提高了路网核密度估计的计算效率；Zhang[11]等将空间索引引入核密度估计，并使用GPU对带宽的选择进行了加速；Panagiotis D[12]等使用CUDA（compute unified device architecture，统一计算架构）框架加速了核密度估计的计算；可视化方面，禹文豪[9]等使用核密度估计法对网络空间下POI点的可视化与分析进行了讨论。对以上文献分析可知，目前国内在核密度估计的GPU并行算法方面的研究尚不充分，同时，国内外研究者多聚焦于核密度估计及其可视化中的一个方面，而对整体流程的研究较少。因此，本文提出了一种协同CPU和GPU的核密度估计及其可视化全流程算法，以在最大程度上提高执行效率。

1 算法设计

核密度估计的基本思想是，对于数据点 p_i，使用核函数（或距离衰减函数） K 计算其对带宽 h 范围内所有目标点的分量 v_i，遍历全部 n 个数据点，目标点 x 的核密度估计值即为所有分量的和 $\sum_{i=1}^{n} v_i$。根据以上分析，空间中目标点 x 的核密度估计值 $\hat{f}(x)$ 的计算公式为：

$$\hat{f}(x)=\frac{1}{nh^2}\sum_{i=1}^{n}K\left(\frac{x-p_i}{h}\right) \tag{1}$$

核密度估计计算属于计算密集型问题，执行时间与计算目标点的核密度估计值中的遍历过程密切相关，可视化方面，现有算法往往需要CPU与GPU之间频繁地进行数据拷贝，使用协程或CPU多线程技术难以大幅提高计算效率。针对核密度估计及其可视化中的问题，本文提出了一种协同CPU和GPU的核密度估计及其可视化算法。该算法使用CPU控制算法的执行，使用GPU并行计算得到核密度估计结果，使用OpenGL核心模式进行可视化，并借助CUDA与OpenGL的互操作，大幅减少了CPU与GPU之间的数据拷贝次数，从而最大限度提高效率。

首先，使用CPU读取数据，然后使用GPU结合CUDA框架并行计算得到核密度估计结果，并将计算结果存储在GPU中；然后，创建着色器及相关缓存对象，并建立顶点缓存对象（vertex buffer object，VBO）与CUDA的显存映射关系，使CUDA可以在核函数中访问或更新顶点缓存对象（VBO）中的数据。接着，根据建立的显存映射关系，在CUDA核函数中；根据核密度估计结果，计算坐标和颜色值数据，存储至VBO；最后，使用OpenGL核心模式对VBO进行可视化，停止渲染后，释放GPU中的存储空间。

1.1 协同CPU与GPU的核密度估计计算

CPU是计算机的大脑，负责计算、逻辑控制和输入输出等操作，图1为基于CPU的串行核密度估计计算流程，该算法需要串行计算每个目标点的核密度估计结果，随着数据点的规模和图像分辨率的提高，效率会愈来愈低，同时，受计算核心数量的物理限制，即使使用CPU多线程并行技术，也难以执行大规模并行计算，无法满足应用对实时性的要求。

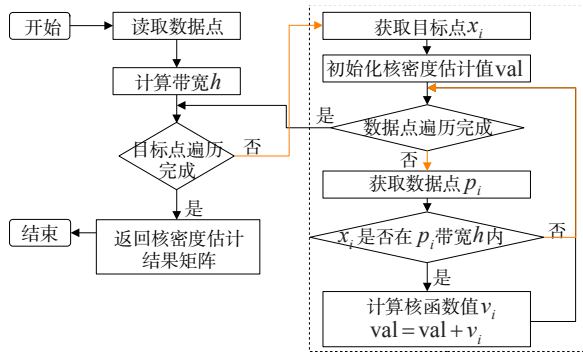


图1 基于CPU的核密度估计计算算法

GPU在设计之初主要用于图形和视频的渲染，而随着关于通用图形处理器（general-purpose computing on graphics processing units，GPGPU）的研究愈来愈深入，尤其OpenCL[14]、Nvidia CUDA[15]等GPU编程模型的出现，以GPU为代表的众核计算架构愈来愈受到重视，逻辑上看，线程是GPU计算的基础组件，多个线程组成线程块，多个线程块组成线程网格，而从硬件的角度看，以Nvidia Turing架构为例，CUDA核心是GPU计算的基础组件，多个CUDA核心组成流式多处理器（stream multiprocessor，SM），实际执行时，SM负责线程块的调度，会将线程块进一步划分为线程束，线程束的执行时间取决于其中耗时最久的线程，因此，为最大化GPU的执行效率，需要选择合适的线程块尺寸[16]。图2为CPU并行计算与GPU并行计算的区别，图中的每个方格代表一个任务，其中CPU依赖多核心执行并行计算，能够同时运行的线程数量很少，因此每次只能执行少量任务（图中CPU每次执行3个任务），而GPU依赖SM执行并行计算，现代GPU能够同时运行成千上万个线程，因此每次可以执行大量任务（图中每个SM执行25个任务，因此一次就可以完成所有任务）。

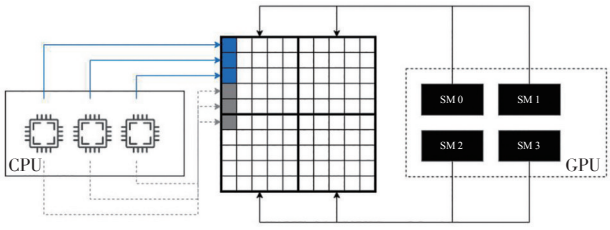


图2　CPU与GPU并行计算的区别

依据以上分析，设计协同CPU与GPU的核密度估计算法流程，如图3所示，具体步骤如下：

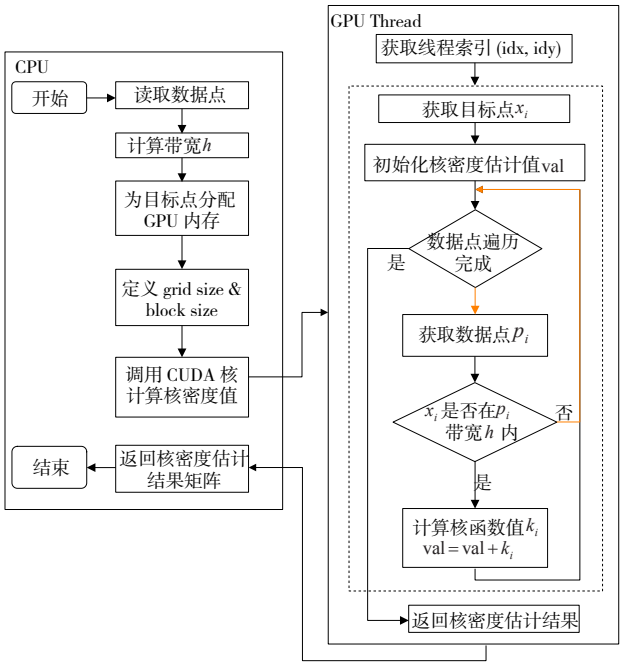


图3　CPU与GPU协同的核密度估计计算算法

1）读取所有数据点的坐标信息，根据所生成的图像分辨率，计算核密度估计相关参数。

2）根据目标点的数量，确定CUDA的线程网格参数，包括线程块尺寸和网格尺寸，接着在GPU中为目标点分配内存，将核密度估计的相关参数以及点数据从CPU拷贝至GPU后，执行核密度计算核函数计算核密度估计结果。

核密度计算核函数的执行逻辑与CPU算法中对目标点遍历的一次循环相对应，首先获取线程的索引(idx, idy)，然后根据该索引获取对应的目标点 x_i，对该目标点，遍历所有数据点，使用核函数计算每一个数据点对该目标点的分量，分量的和为目标点 x_i 的核密度估计结果。这样，GPU中的每一个线程计算得到相应目标点的核密度估计结果，所有线程并行执行结束后，就得到了所有目标点的核密度估计结果。

3）将该结果存储在GPU中，利用CUDA与OpenGL的互操作，可以使OpenGL直接访问存储于GPU中的数据，因此，不需要将数据从客户端拷贝至主机端。

1.2　核密度估计结果的可视化

数据可视化技术是指运用计算机图形学和图像处理技术，将数据转变为直观的图形图像信息，从而帮助决策者理解复杂的数据，洞察数据背后的意义[17]。本文使用OpenGL研究对核密度估计结果的高效率可视化方法。

OpenGL支持立即渲染模式和核心模式，其中立即渲染模式简单易用，但操作GPU的方法十分有限，渲染效率很低，难以满足大数据或需要实时渲染的应用场景。核心模式大大减少了CPU与GPU的内存交换次数，因此渲染效率极高。本算法使用核心模式对核密度估计的结果进行可视化，流程如下。

首先，将核密度估计结果矩阵中的元素值进行归一化，然后建立核密度值与颜色值的映射关系，从而获取目标点的颜色数据，同时，根据核密度估计结果，计算投影矩阵；接着，使用OpenGL着色器语言编写顶点着色器和片段着色器，并将投影矩阵传递给顶点着色器；然后，将目标点坐标数据和颜色数据缓存至VBO，将目标点索引数据缓存至IBO，将VBO和IBO缓存至VAO；最后，OpenGL内部执行顶点着色、添加纹理、几何着色、栅格化、片段着色等阶段后，即可得到渲染图像。与立即渲染模式不同的是，核心模式只需要从CPU拷贝一次数据至GPU，之后的每一次渲染，只需要绑定相应的VAO即可，这极大的提高了渲染效率，能够满足应用对实时性渲染的要求。

2　实验及结果分析

2.1　核函数及带宽选择

实验使用ArcGIS中核密度分析工具所使用的核函数，其计算公式为：

$$K(x)=\frac{3}{\pi}\left(1-x^2\right)^2 \tag{2}$$

使用ArcGIS中核密度分析工具所使用的带宽计算策略，其计算公式为：

$$h=0.9\times\min\left(SD,\sqrt{\frac{1}{\ln(2)}}\times D_m\right)\times n^{-0.2} \tag{3}$$

$$SD=\sqrt{\frac{\sum_{i=1}^{n}\left|p_i-\bar{P}\right|}{n}} \tag{4}$$

$$\bar{P}=\sum_{i=1}^{n}\frac{p_i}{n} \tag{5}$$

式中，SD 为标准距离；p_i 为数据点 i；$\bar{P}$ 为数据点的平均中心；D_m 为所有数据点距平均中心距离的中值。

2.2 数据集

使用武汉市中心城区综合医院POI共1 002个点数据作为测试集，该数据来源于高德数据开放平台，其空间分布如图4所示，实验以该点数据作为测试数据，通过设置不同的图像分辨率，分组对比算法执行效率。

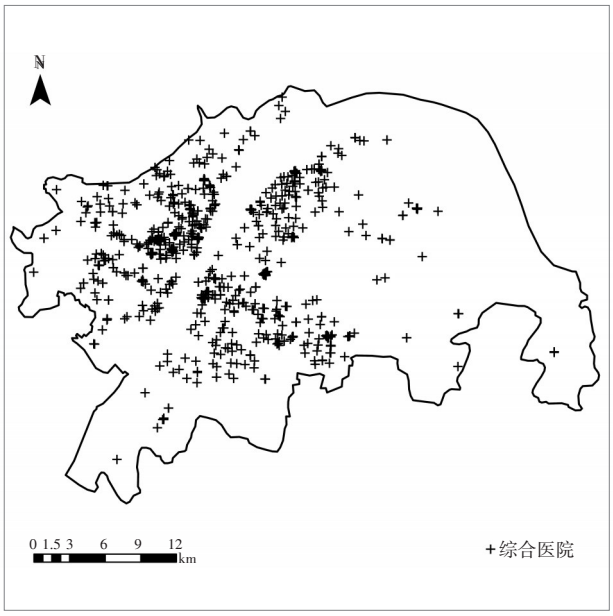


图4 测试数据集空间分布

审图号：鄂S（2024）010号

2.3 实验设计

实验所用CPU的型号是Intel® Core ™ i5-10400F，具有6核处理器，支持12个线程。GPU型号是NVIDIA GeForce RTX 2060 SUPER，该显卡使用Turing架构，共包含34个SM，2 176个CUDA计算核心，每个SM的最大并发线程数是1 024，支持OpenGL 4.6。

除空间点数据量外，算法的执行效率还与图像分辨率强相关，对核密度估计结果可视化，分别生成500×500和1 000×1 000像素尺寸的图像，结果如图5所示，其中上半部分为全区域500×500像素尺寸的可视化结果以及局部放大图，下半部分为全区域1 000×1 000像素尺寸的可视化结果以及局部放大图，可以看出，图像的分辨率越高，虽然计算量越大，但包含的信息量会更大，可视化效果也会更好。

为比较OpenGL立即渲染模式和核心模式在显示效率上的差异，对测试数据集，首先使用CPU串行计算核密度估计结果，然后分别使用立即渲染模式和核心模式进行渲染，分别生成1 000×1 000，5 000×5 000，10 000×10 000三种像素尺寸（像素点分别为10^6，2.5×10^7，10^8个）的图像，对每一像素尺寸，

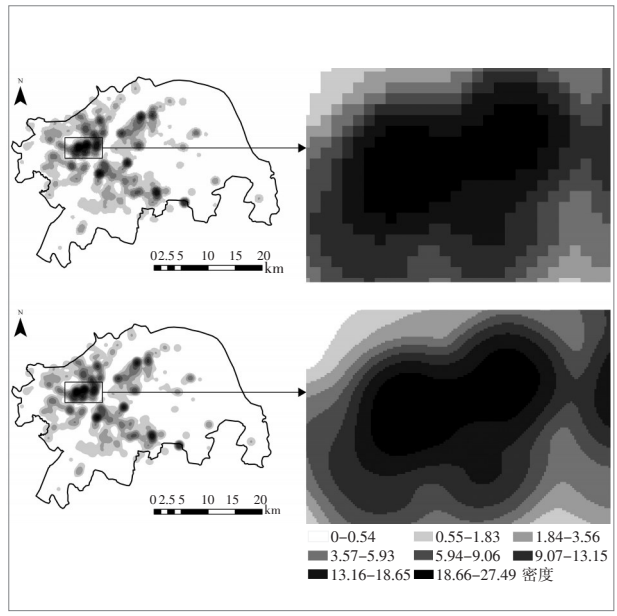


图5 不同图像分辨率下的可视化结果对比

审图号：鄂S（2024）010号

执行10次实验，记录渲染时间，获取平均值，结果如表1所示。由于立即渲染模式在每一帧渲染之前都需要重新计算所需的数据，因此渲染时间较长，随着分辨率的提高，卡顿感也越来越明显，而核心模式提前计算了渲染所需的数据，大幅度降低了每一帧的渲染时间，每一帧的渲染时间也比较稳定，不会有明显的卡顿感。

表1 OpenGL立即渲染模式与核心模式渲染效率对比

	OpenGL模式		1 000×1 000	5 000×5 000	10 000×10 000
执行时间	立即渲染模式	平均值	0.007 867 s/帧	0.283 679 s/帧	1.170 202 s/帧
	核心模式	平均值	0.018 s +0.000 002 s/帧	0.416 s +0.000 002 s/帧	1.650 s +0.000 003 s/帧

为确定CUDA的最佳线程块尺寸，分别使用32、64、128、256、512、1 024六种线程块尺寸，对上述测试集，使用协同CPU和GPU的核密度估计及其可视化算法，计算核密度估计结果，然后将其渲染为1 000×1 000像素尺寸的图像，对每一线程块尺寸，分别执行10次实验，记录所需的时间，计算平均值和标准差，结果如图6所示。结果表明：当线程块尺寸为256时，运行时间最短，该参数将应用于与CPU串行/并行的对比实验中。

本文提出的协同CPU与GPU的核密度估计和可视化算法，包括核密度估计计算以及可视化2个方面。OpenGL立即渲染模式与核心模式渲染效率对比实验证实了核心渲染模式的高效性，为验证核密度估计

表2　核密度估计及其可视化算法效率对比

算法			1 000×1 000	5 000×5 000	10 000×10 000
执行时间/s	CPU串行	平均值	1.619	39.814	158.331
		标准差	0.022	0.532	1.106
	CPU并行	平均值	0.357	8.104	32.077
		标准差	0.009	0.069	0.217
	协同CPU和GPU	平均值	0.203	1.319	4.892
		标准差	0.005	0.021	0.028
加速比	CPU串行/CPU并行		4.533	4.913	4.936
	CPU并行/协同CPU和GPU		1.756	6.146	6.557
	CPU串行/协同CPU和GPU		7.958	30.197	32.366

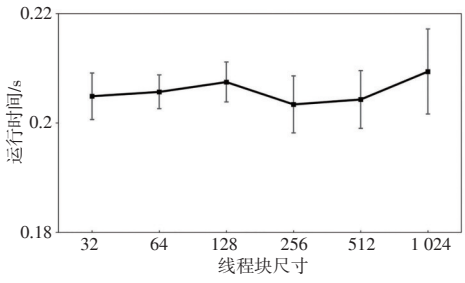


图6　线程块尺寸对运行时间的影响

的计算以及CUDA与OpenGL互操作的高效性，对上述测试集，分别使用基于CPU串行的核密度估计算法、基于CPU并行[18]的核密度估计算法以及协同CPU和GPU的核密度估计算法，计算核密度估计结果，然后统一采用核心模式，对结果进行渲染，分别生成1 000×1 000，5 000×5 000，10 000×10 000三种像素尺寸的图像，同样对每一像素尺寸，执行10次实验，记录时间，求得平均值和标准差，结果如表2所示，算法A相对于算法B的加速比表示为算法A的执行时间与算法B执行时间的比值，实验结果显示，相比于CPU并行和串行算法，本文提出的算法，在效率上分别提升了约5倍和20倍，同时，随着所生成图像分辨率的提高，加速比呈现逐步增大的趋势，因此，认为本文提出的算法显著提高了整体的执行效率。

3　结　语

大数据时代背景下，空间数据点规模和图像分辨率越来越高，传统的核密度估计及其可视化算法的执行效率愈来愈低，难以满足应用对实时性的需求，因此，针对该计算密集型问题，本文结合GPU的大规模并行计算能力以及CUDA与OpenGL的互操作，提出一种应对地理空间大数据的协同CPU与GPU的核密度估计及其可视化算法，实验证明，该算法大幅度提高了核密度估计及其可视化的效率。

参考文献

[1] Miller H J. Tobler's First Law and Spatial Analysis[J]. Annals of the Association of American Geographers, 2004, 94(2):284-289

[2] Xie Z, Yan J. Kernel Density Estimation of Traffic Accidents in a Network Space[J]. Computers, Environment and Urban Systems, 2008, 32(5):396-406

[3] 胡森，李松廉，蔡忠亮. 院前急救时空热点分析:醉酒为例[J]. 地理空间信息，2022，20(6):20-23

[4] 禹文豪，艾廷华，杨敏，等. 利用核密度与空间自相关进行城市设施兴趣点分布热点探测[J]. 武汉大学学报(信息科学版)，2016，41(2):221-227

[5] 朱峥嵘，黄亚锋，赵立营. 一种利用GPU加速的轨迹线热力图生成显示方法[J]. 武汉大学学报(信息科学版)，2022，47(7):1 035-1 042

[6] Ghani N A, Hamid S, Targio H I A, et al. Social Media Big Data Analytics: A Survey[J]. Computers in Human Behavior, 2019, 101:417-428

[7] Jo J, Lee K W. High-performance Geospatial Big Data Processing System Based on Map Reduce[J]. ISPRS International Journal of Geo-Information, 2018, 7(10):399

[8] Brodtkorb A R, Hagen T R, Stra M L. Graphics Processing Unit (GPU) Programming Strategies and Trends in GPU Computing[J]. Journal of Parallel and Distributed Computing, 2013, 73(1):4-13

(下转第47页)