

高效且对网格分辨率不敏感的点在多边形内的判定方法

王文成*, 王盛春, "Efficient Point-in-Polygon Tests by Grids without the Trouble of Tuning the Grid Resolutions," in *IEEE Transactions on Visualization and Computer Graphics*, doi: 10.1109/TVCG.2021.3073919, 2021.

*E-mail: whn@ios.ac.cn, *Tel:010-62661611

问题描述

点在多边形内的判定检测是计算几何中的一个基本问题，在计算机图形学、卫星定位系统及地理信息系统等领域有着广泛的应用。

该问题可描述为：给定一个多边形P与一个任意的点q，判断点q是否位于多边形P内。

- 目前已有大量的点在多边形内的判定检测算法，如射线法、面积法、均匀网格法等，但寻找更加高效的解决方案仍然是重要的研究内容。
- 该判定问题是很多算法的基础，如多边形裁剪和光线跟踪等，其效率在很大程度上影响着上层应用的效率。

相关工作

射线法：通过从测试点向某方向作一条射线，根据该射线与多边形边的交点数目的奇偶性进行判定^[1]。若交点数目为奇数，则该点位于多边形内；否则位于多边形外。

均匀网格法：将多边形的包围盒划分为均匀网格，进而根据点与所在网格内的多边形边的位置关系，判定点的位置属性。文献[2]是目前处理速度最快的算法(GCP)，其通过使用位置属性已知的网格中心点来加快被测点的判定，但它依旧存在许多问题，限制了该算法的处理效率。

- 使用叉积运算确定边是否相交，计算量比较大。
- 预处理阶段对网格分辨率的大小非常敏感，当网格分辨率设置过大或过小时，其算法的处理效率都将受到很大的影响。

工作思路

提出一种通过使用与Y轴平行的条带来增强网格单元结构，以进一步解决上述问题，称之为R-GP方法。该方法的创新点是引入了一种条带来管理每个网格单元中的多边形边，其具体构建过程如下所述。

- 计算多边形边与网格线的横线的所有交点；
- 在每个网格单元中，根据得到的交点生成与Y轴平行的边缘线，以将网格单元划分为条带，如图1所示。

R-GP方法包括2个阶段：预处理阶段和判定计算阶段。

预处理阶段：首先建立均匀网格；计算多边形边与网格线的横线的所有交点，并在每个网格单元中构造条带，如图2所示；最后，预计算条带顶部边界的包含属性。最终每个条带都记录着与它相交的边的相关信息以及条带顶部边界的包含属性。

测试阶段：首先确定待测点所在的网格单元，之后再确定其所处的条带；然后从测试点向上发出一条平行Y轴的测试线，并将其延伸至条带的顶部边界；最后通过计算测试线与条带内多边形边的交点个数，同时根据条带顶部边界的内外属性，进而确定测试点的内外属性。

图1 单元网格中的条带示意图

图2 多边形内条带生成示意图

创新点

- 引入条带结构管理每个网格单元中的多边形边，条带之间相互独立，有利于使用GPU进行并行加速。
- 提出一种对网格分辨率变化不敏感的点在多边形内的判定方法，避免了调整网格分辨率的麻烦。

参考文献

- [1] Feito F, Torres J C, Ureña A. Orientation, simplicity, and inclusion test for planar polygons[J]. Mathematical Gazette, 1995, 19(4):595-600.
- [2] 李静, 王文成. 基于网格中心点的点在多边形内的高效判定[J]. 软件学报, 2012, 23(9):2481-2488.

实验结果

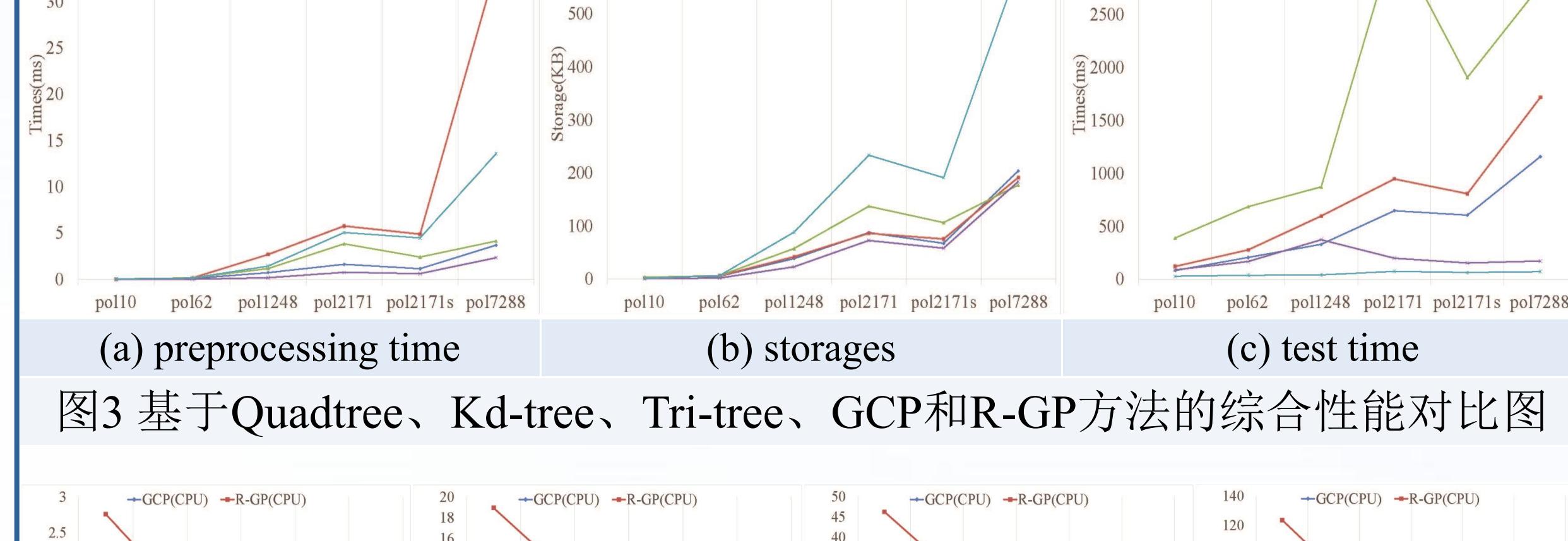


图3 基于Quadtree、Kd-tree、Tri-tree、GCP和R-GP方法的综合性能对比图

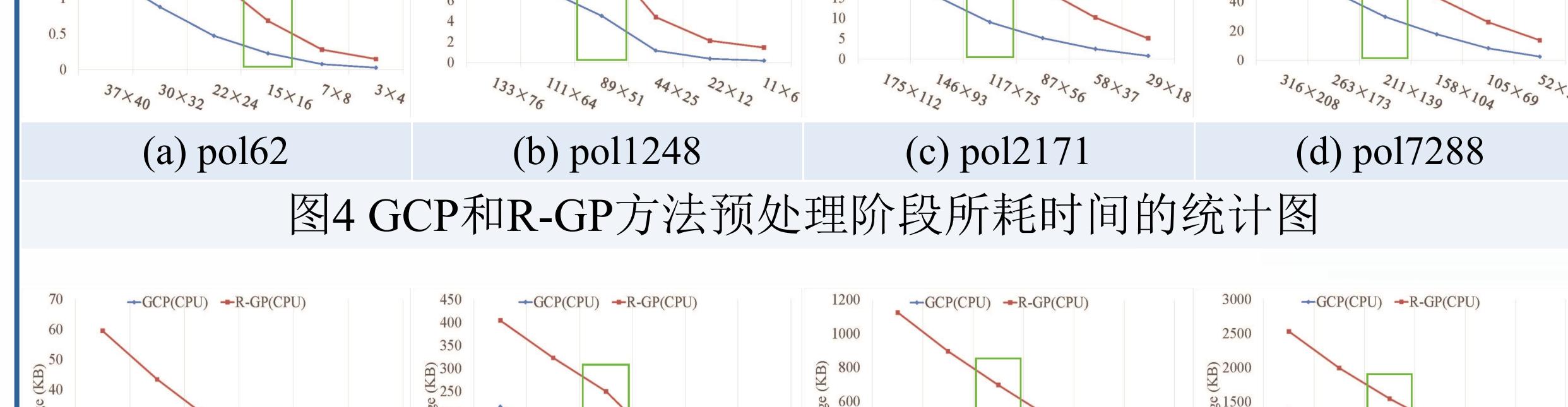


图4 GCP和R-GP方法预处理阶段所耗时间的统计图

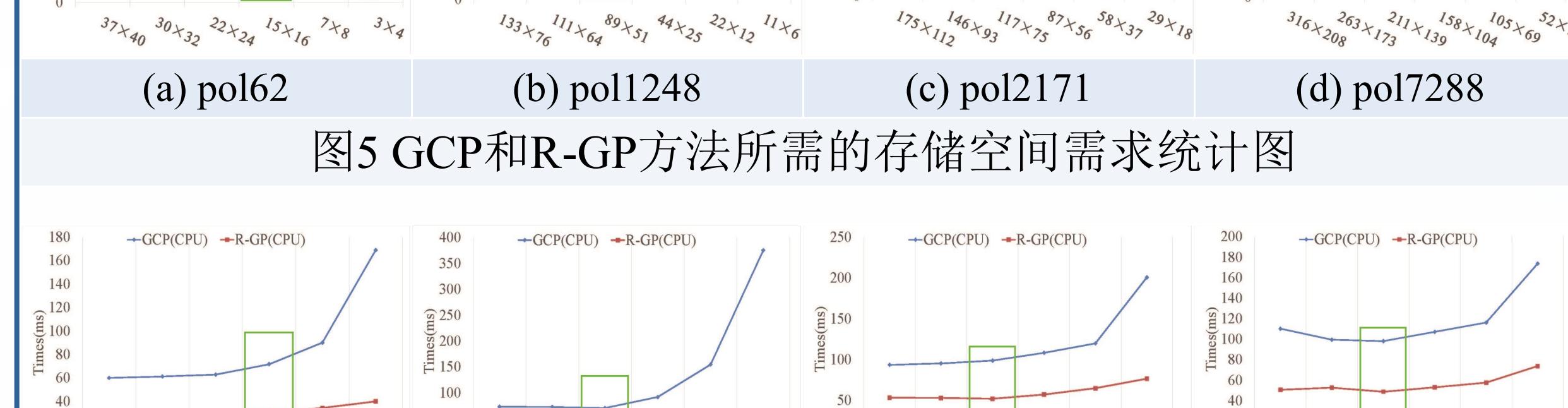


图5 GCP和R-GP方法所需的存储空间需求统计图

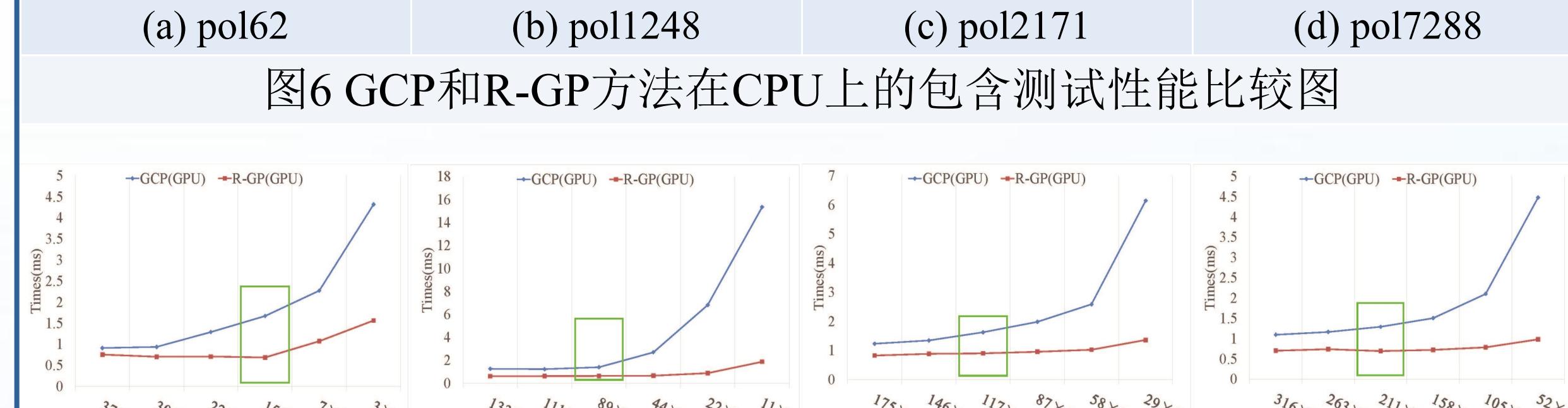


图6 GCP和R-GP方法在CPU上的包含测试性能比较图

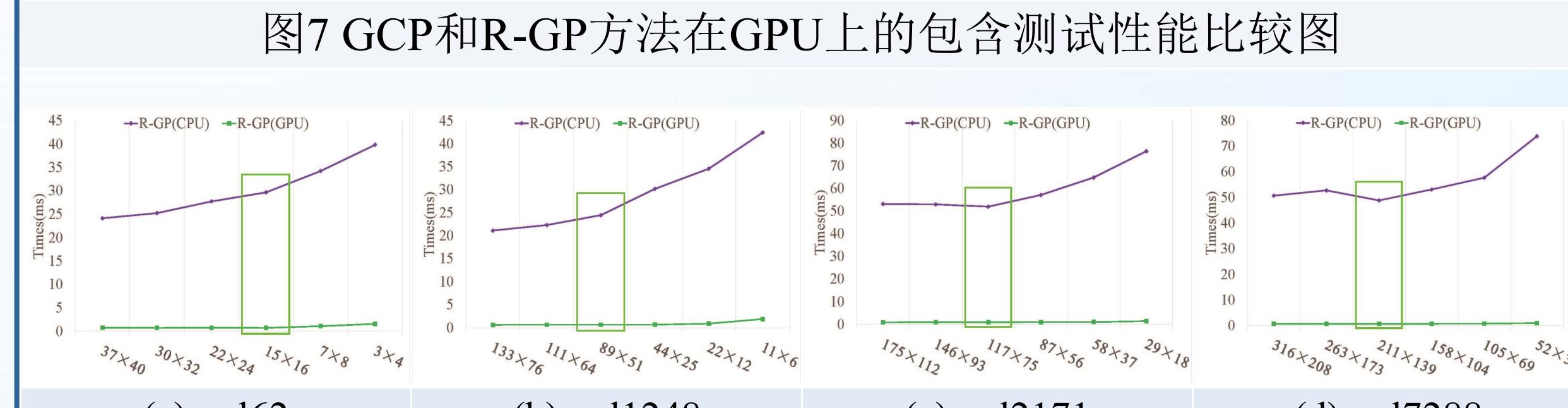


图7 GCP和R-GP方法在GPU上的包含测试性能比较图

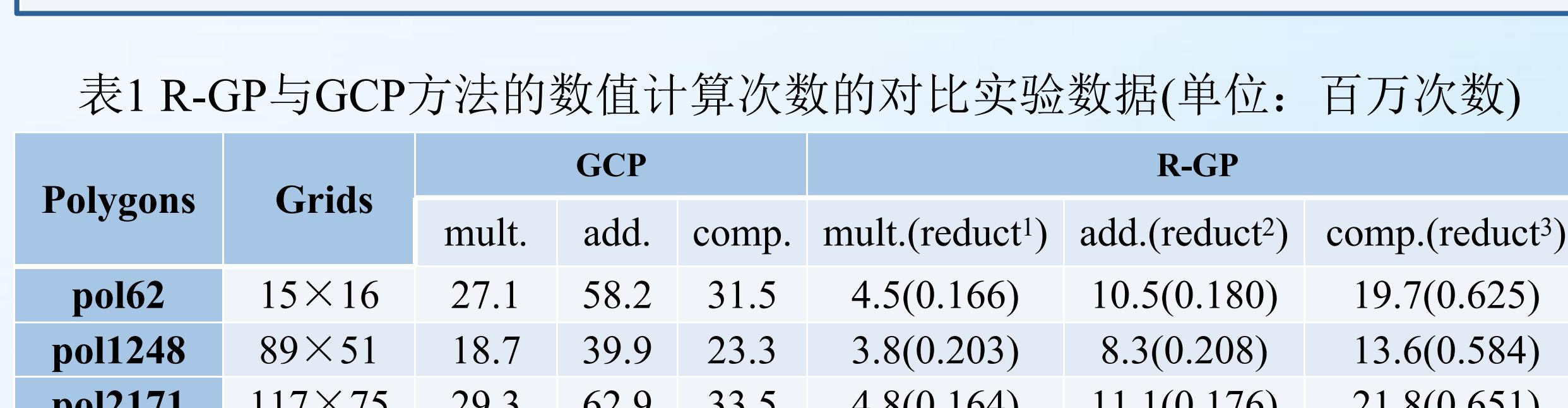


图8 R-GP方法在CPU和GPU上的性能比较图

表1 R-GP与GCP方法的数值计算次数的对比实验数据(单位: 百万次数)

Polygons	Grids	GCP			R-GP		
		mult.	add.	comp.	mult.(reduct ¹)	add.(reduct ²)	comp.(reduct ³)
pol62	15×16	27.1	58.2	31.5	4.5(0.166)	10.5(0.180)	19.7(0.625)
pol1248	89×51	18.7	39.9	23.3	3.8(0.203)	8.3(0.208)	13.6(0.584)
pol2171	117×75	29.3	62.9	33.5	4.8(0.164)	11.1(0.176)	21.8(0.651)
	316×208	13.0	27.3	17.8	3.6(0.277)	7.6(0.278)	11.3(0.635)
	263×173	15.0	31.6	19.7	3.7(0.247)	7.9(0.250)	12.2(0.619)
	211×139	18.2	38.7	22.8	3.8(0.209)	8.4(0.217)	13.6(0.596)
	158×104	24.7	52.9	29.1	4.0(0.162)	9.1(0.172)	15.8(0.543)
	105×69	41.0	88.3	44.7	4.4(0.107)	10.4(0.118)	19.5(0.436)
	52×34	118.1	256.7	118.9	5.0(0.042)	14.0(0.055)	30.5(0.257)

mult., add.和comp分别表示GCP和R-GP在测试过程中所需执行的乘法、加法和比较运算的总次数。

reduct¹, reduct²和reduct³，分别表示R-GP相对于GCP在乘法、加法和比较运算次数上的缩减率，其中，reduct¹= multiplications(R-GP)/multiplications(GCP), reduct²= additions(R-GP)/additions(GCP)和reduct³= comparisons(R-GP)/comparisons(GCP)。

结论

从上述图例中可以得出，本文提出的R-GP方法明显优于现有方法的处理性能。虽然R-GP方法需要更多的预处理时间和存储空间，但是该方法可以采用较小的网格分辨率进行处理，在降低预处理时间和存储空间的同时，依旧能够保持较高的计算效率。

从表1中可以看出，R-GP可以大大减少所需计算操作(乘法、加法和比较运算)的数量。同时该方法能够充分利用GPU的并行计算能力，因此相对于现有的方法，本文所提出的方法取得了较高的处理速度。

R-GP方法具有较高的处理速度，可适用于需要实时处理的实际应用，如洪水泛滥中安全区域的实时检测，该方法的处理速度大于1000fps。

图9 洪水泛滥中安全区域的实时检测示例图