

Compute the Spatial Distance Histogram of a Set of 2D Points using Parallel Computing

Yu Liang, Master Student in CSE
Jiabo Liang, Master Student in CSE
Jiayi Wang, Master Student in CSE

Abstract — as parallel computing has evolved, all areas of computing are faced with rebuilding to get faster processing speed. This article focuses on one important issue in scientific simulation data analysis: the spatial distance histogram (SDH). When calculating the spatial distance from the histogram when the calculation is very huge. Usually these calculations should be kept for a short period of time, in the event of failure, but also to start again. We propose a more efficient algorithm that uses parallel computing and prove that this algorithm has a good fault tolerance. The core of our algorithm is to decompose the space of a particle into a quadtree-based structure and then obtain the number of particles in each node of the tree to determine the spatial distribution of the particle. Our research shows how we can implement these algorithms in a graphics processing unit (GPU) to achieve a certain level of fault tolerance. The efficiency of our proposed algorithm will have a wide range of effects on the data produced by the actual simulation study

Index Terms — spatial distance histogram, parallel computing, 2D Points

I. INTRODUCTION

Every year through computer simulation experiments will produce a large amount of data. Despite the use of a database management system to process this data, data management software is not sufficient to handle such a large amount of data [1-3]. Traditional database management system is established for business applications, not suitable for managing big data. So we need parallel computing to redesign the data management system. These programs usually require parallelism methods that use large numbers of GPUs for computations to significantly improve operational efficiency [4]. In this way, these scientific big data can be effectively used for scientific simulation experiments.

In order to analyze the data of 2D points, traditionally, scientists have to do a lot of complex calculations to create a spatial distance histogram [5-7]. In general, the queries used in this analysis are based on the location of each particle: the function of all m-tuple subsets involving data is called the m-body correlation function. One such analysis query discussed in this article is the so-called Spatial Distance Histogram (SDH). SDH is a histogram of the distances between all pairs of particles in the system, which represents a continuous probability distribution of distances (called the radial distribution function (RDF).) This type of query is very important in an MS database as one of the basic building blocks to describe a series of key quantities (such as total pressure and energy) required by physical systems.

With the development of GPU-based parallel computing in recent years, GPU parallel computing has become a fantastic choice when the CPU cannot handle large amounts of social networking data quickly[8, 9]. Through the GPU's large-scale processing power, can quickly construct SDH relationship between notes. In this study, we use parallel computing to construct SDH relationship in 2D points.

II. METHOD

The SDH problem can be formally described as follows: given the coordinates of N particles and a user-defined distance w , we need to compute the number of particle-to-particle distances falling into a series of ranges (named buckets) of width w : $[0, w)$, $[w, 2w)$, ..., $[(l-1)w, lw]$. Essentially, the SDH provides an ordered list of non-negative integers $H = (h_0, h_1, \dots, h_{l-1})$, where each $h_i (0 \leq i < l)$ is the number of distances falling into the bucket $[iw, (i+1)w)$. We also use $H[i]$ to denote h_i in this paper.

To capture the variations of system states over time, there is a need to compute SDH for a large number of consecutive frames. We denote the count in bucket i at frame j as $H_j[i]$.

Each node has 4 child nodes, down until it reaches a threshold, the last layer is the leaf node. The nodes above the leaf nodes need only store the number of internal points of the Cell (each square is a Cell). The leaf nodes not only store the number but also store the index of all the points in the cell, so as to directly calculate the distance between the points distance.

the minimum distance are not in the same bucket, compute their child node until it meets this condition or calculate the distance between the leaf nodes directly.

III. RESULT

```

-----
The time spent on SDW query: 4.267(sec)
The number of distances fall into No.1 bucket: 72958
The number of distances fall into No.2 bucket: 121698
The number of distances fall into No.3 bucket: 344978
The number of distances fall into No.4 bucket: 469453
The number of distances fall into No.5 bucket: 585542
The number of distances fall into No.6 bucket: 695644
The number of distances fall into No.7 bucket: 794131
The number of distances fall into No.8 bucket: 889374
The number of distances fall into No.9 bucket: 974262
The number of distances fall into No.10 bucket: 1054512
The number of distances fall into No.11 bucket: 1127703
The number of distances fall into No.12 bucket: 1195985
The number of distances fall into No.13 bucket: 1256512
The number of distances fall into No.14 bucket: 1308940
The number of distances fall into No.15 bucket: 1357247
The number of distances fall into No.16 bucket: 1395996
The number of distances fall into No.17 bucket: 1431645
The number of distances fall into No.18 bucket: 1459501
The number of distances fall into No.19 bucket: 1482999
The number of distances fall into No.20 bucket: 1501710
The number of distances fall into No.21 bucket: 1511927
The number of distances fall into No.22 bucket: 1517963
The number of distances fall into No.23 bucket: 1515277
The number of distances fall into No.24 bucket: 1511360
The number of distances fall into No.25 bucket: 1499271
The number of distances fall into No.26 bucket: 1482020

```

```
Time to generate: 701.7936 ms

Time: 72568 | 251380 | 344976 | 466423 | 565543 |
655643 | 794137 | 883872 | 974002 | 1054512 |
10117704 | 11595985 | 1256511 | 13089840 | 13572748 |
13359596 | 14316444 | 14935002 | 14833001 | 15017079 |
15131925 | 16135470 | 16355762 | 15511360 | 14499700 |
14812018 | 14585002 | 14343378 | 14070000 | 13712823 |
1235623 | 1251143 | 1244791 | 1156882 | 1141496 |
1085940 | 1063459 | 964887 | 901398 | 836125 |
7593015 | 682576 | 601969 | 521021 | 437320 |
354549 | 279494 | 223475 | 181406 | 146180 |
114963 | 92743 | 56767 | 34567 | 27828 |
30949 | 21579 | 14567 | 9073 | 5192 |
2761 | 1380 | 427 | 81 | 13 |
0 | 0 | 0 | 0 | 0 |
T:45955000
```

Second, we generate 100,000 points and build SDH with bucket width 1,000. The results for CPU (Figure 4) and GPU (Figure 5) are as below.

```

-----
The time spent on SDH query: 249.630(sec)
The number of distances fall into No.1 bucket: 28626442
The number of distances fall into No.2 bucket: 81637251
The number of distances fall into No.3 bucket: 120413139
The number of distances fall into No.4 bucket: 169233261
The number of distances fall into No.5 bucket: 204316427
The number of distances fall into No.6 bucket: 233717572
The number of distances fall into No.7 bucket: 257666701
The number of distances fall into No.8 bucket: 276138766
The number of distances fall into No.9 bucket: 289725427
The number of distances fall into No.10 bucket: 298538906
The number of distances fall into No.11 bucket: 302763689
The number of distances fall into No.12 bucket: 302713145
The number of distances fall into No.13 bucket: 298678417
The number of distances fall into No.14 bucket: 290709241
The number of distances fall into No.15 bucket: 278819768
The number of distances fall into No.16 bucket: 263546945
The number of distances fall into No.17 bucket: 244882832
The number of distances fall into No.18 bucket: 222913150
The number of distances fall into No.19 bucket: 198547312
The number of distances fall into No.20 bucket: 171635381
The number of distances fall into No.21 bucket: 142508957
The number of distances fall into No.22 bucket: 111329453
The number of distances fall into No.23 bucket: 78073808
The number of distances fall into No.24 bucket: 48660875
The number of distances fall into No.25 bucket: 31511923
The number of distances fall into No.26 bucket: 20011577
The number of distances fall into No.27 bucket: 12144193
The number of distances fall into No.28 bucket: 6861788
The number of distances fall into No.29 bucket: 3467411
The number of distances fall into No.30 bucket: 1460912
The number of distances fall into No.31 bucket: 434594
The number of distances fall into No.32 bucket: 59971
The number of distances fall into No.33 bucket: 736
The total number of distances in the entire histogram: 499995000

```

Fig 4. Time cost 249s for 100,000 points with bucket width 1,000 in CPU

```

Time to generate: 346.72236 ms
00: 72558 | 212590 | 344978 | 469453 | 585543 |
05: 695443 | 794132 | 893373 | 974262 | 1054512 |
10: 1127704 | 1195995 | 1256511 | 1308940 | 1357248 |
15: 1395996 | 1431644 | 1459502 | 1483901 | 1501707 |
20: 1511242 | 1519665 | 1515276 | 1511360 | 1499279 |
25: 1482019 | 1459202 | 1434379 | 1407090 | 1374283 |
30: 1329823 | 1291143 | 1244791 | 1196882 | 1143496 |
35: 1045841 | 1025494 | 964927 | 901996 | 834125 |
40: 739615 | 692516 | 603469 | 521021 | 437329 |
45: 354349 | 277949 | 223425 | 181406 | 146188 |
50: 116809 | 92749 | 72611 | 56326 | 42846 |
55: 38948 | 21579 | 14987 | 9078 | 5182 |
60: 2791 | 1180 | 427 | 81 | 13 |
65: 0
T:499995000
[1]jaywang@cs.cmu.edu treeSDH

```

Fig 5. Time cost 12.894s for 100,000 points with bucket width 1,000 in GPU

Third, we generate 100,000 points and build SDH with bucket width 500. The results for CPU (Figure 6) and GPU (Figure 7) are as below.

```

-----
The time spent on SDH query: 452.593(sec)
The number of distances fall into No.1 bucket: 7286465
The number of distances fall into No.2 bucket: 21339977
The number of distances fall into No.3 bucket: 34600325
The number of distances fall into No.4 bucket: 47036926
The number of distances fall into No.5 bucket: 58740267
The number of distances fall into No.6 bucket: 69672872
The number of distances fall into No.7 bucket: 79873019
The number of distances fall into No.8 bucket: 89360262
The number of distances fall into No.9 bucket: 98140915
The number of distances fall into No.10 bucket: 106175512
The number of distances fall into No.11 bucket: 113515718
The number of distances fall into No.12 bucket: 120201854
The number of distances fall into No.13 bucket: 126186358
The number of distances fall into No.14 bucket: 131480343
The number of distances fall into No.15 bucket: 136045561
The number of distances fall into No.16 bucket: 140073205
The number of distances fall into No.17 bucket: 143474421
The number of distances fall into No.18 bucket: 146251006
The number of distances fall into No.19 bucket: 148465625
The number of distances fall into No.20 bucket: 150073261
The number of distances fall into No.21 bucket: 151132839
The number of distances fall into No.22 bucket: 151630850
The number of distances fall into No.23 bucket: 151603751
The number of distances fall into No.24 bucket: 151109394
The number of distances fall into No.25 bucket: 150106703
The number of distances fall into No.26 bucket: 148571714
The number of distances fall into No.27 bucket: 146599109
The number of distances fall into No.28 bucket: 144110132
The number of distances fall into No.29 bucket: 141094534
The number of distances fall into No.30 bucket: 137725234
The number of distances fall into No.31 bucket: 133895236
The number of distances fall into No.32 bucket: 129651709

```

Fig 6. Time cost 452s for 100,000 points with bucket width 500 in CPU

```

Time to generate: 10969.74707 ms
00: 7286466 | 21339978 | 34600329 | 47036929 | 58740274 |
05: 69672882 | 79873006 | 89360377 | 98140931 | 106175534 |
10: 113515712 | 120201863 | 126186350 | 131480343 | 136045577 |
15: 140073210 | 143474459 | 146251021 | 148465659 | 150073257 |
20: 151128440 | 151630825 | 151630799 | 151109324 | 150106705 |
25: 146571712 | 146599098 | 144110130 | 141094545 | 137725221 |
30: 133895245 | 129651676 | 125028450 | 119854480 | 114330019 |
35: 108582133 | 102497608 | 96049469 | 89325419 | 82309541 |
40: 75012208 | 67484725 | 59746483 | 51622527 | 43238931 |
45: 34814953 | 27143870 | 21716591 | 17485640 | 14026249 |
50: 11185753 | 8825813 | 6873882 | 5276292 | 3961925 |
55: 2899858 | 2060494 | 1466912 | 910224 | 548891 |
60: 297484 | 136997 | 48691 | 11289 | 736 |
65: 0
T:499995000

```

Fig 7. Time cost 10.969s for 100,000 points with bucket width 500 in GPU

Finally, we generate 100,000 points and build SDH with bucket width 5,000. The results for CPU (Figure 8) and GPU (Figure 9) are as below.

```

-----
The time spent on SDH query: 1.634(sec)
The number of distances fall into No.1 bucket: 6093552
The number of distances fall into No.2 bucket: 13518238
The number of distances fall into No.3 bucket: 14712601
The number of distances fall into No.4 bucket: 11018716
The number of distances fall into No.5 bucket: 4185410
The number of distances fall into No.6 bucket: 462001
The number of distances fall into No.7 bucket: 4402
The total number of distances in the entire histogram: 49995000

SDH resolution = 5000.000 start level = 3 stop levels = 5
Level 3 4065.86 Recursions(Pairs) = 2016 Resolved =
Level 4 2032.93 Recursions(Pairs) = 32256 Resolved =
Level 5 1016.47 Recursions(Pairs) = 370752 Resolved =

Inter-node distances resolved: 31832901 (63.672%)
Intra-node distances resolved (64 nodes): 779978 (1.560%)
The number of distances resolved by Brute-Force: 17382121 (34.768%)
The total number of recursions have been called: 405024

```

Fig 8. Time cost 1.634s for 100,000 points with bucket width 5,000 in CPU

```

Time to generate: 3911.46753 ms
00: 6093552 | 13518238 | 14712602 | 11018715 | 4185410 |
05: 462001 | 4402 |
T:49995000
[jaywang@cs.cmu.edu treeSDH]$ vi output3

```

Fig 9. Time cost 3.911s for 100,000 points with bucket width 5,000 in GPU

Points; bucket width	CPU	GPU
10,000; 500	4.267s	0.386s
100,000; 1000	249s	12.894s
100,000; 500	452s	10.969s
10,000; 5,000	1.634s	3.911s

Table 1. Overall result

III. CONCLUSION

We found that for 10,000 dots, the GPU performed better when the width was small. When the width is large, the CPU performs better. For

100000 points, the GPU behaves better than the CPU, regardless of the width.

III. REFERENCE

- [1] R. Bhatti, A. Samuel, M. Y. Eltabakh, H. Amjad, and A. Ghafoor, "Engineering a policy-based system for federated healthcare databases," (in English), *Ieee Transactions on Knowledge and Data Engineering*, vol. 19, no. 9, pp. 1288-1304, Sep 2007.
- [2] M. Gray and J. Kalpers, "Ranger based monitoring in the Virunga-Bwindi region of east-central Africa: A simple data collection tool for park management," (in English), *Biodiversity and Conservation*, vol. 14, no. 11, pp. 2723-2741, Oct 2005.
- [3] M. H. Ng et al., "BioSimGrid: Grid-enabled biomolecular simulation data storage and analysis," (in English), *Future Generation Computer Systems-the International Journal of Grid Computing Theory Methods and Applications*, vol. 22, no. 6, pp. 657-664, May 2006.
- [4] A. Kumar, V. Grupcev, Y. K. Yuan, J. Huang, Y. C. Tu, and G. Shen, "Computing Spatial Distance Histograms for Large Scientific Data Sets On-the-Fly," (in English), *Ieee Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2410-2424, Oct 2014.
- [5] V. Grupcev et al., "Approximate Algorithms for Computing Spatial Distance Histograms with Accuracy Guarantees," (in English), *Ieee Transactions on Knowledge and Data Engineering*, vol. 25, no. 9, pp. 1982-1996, Sep 2013.
- [6] J. Narwade and B. Kumar, "Comparison of Spatial Color Histograms Using Quadratic Distance Measure," (in English), 2015 *International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 995-998, 2015.
- [7] S. P. Chen, Y. C. Tu, and Y. N. Xia, "Performance analysis of a dual-tree algorithm for computing spatial distance histograms," (in English), *Vldb Journal*, vol. 20, no. 4, pp. 471-494, Aug 2011.
- [8] C. Chen, K. L. Li, A. J. Ouyang, Z. Tang, and K. Q. Li, "GPU-Accelerated Parallel Hierarchical Extreme Learning Machine on Flink for Big Data," (in English), *Ieee Transactions on Systems Man Cybernetics-Systems*, vol. 47, no. 10, pp. 2740-2753, Oct 2017.
- [9] F. Han and S. Z. Sun, "Petroleum Geoscience Big Data and GPU Parallel Computing," (in English), 2015 *1st Ieee International Conference on Multimedia Big Data (Bigmm)*, pp. 292-293, 2015.