The University Of Sheffield.

Improved GPU Near Neighbours Through Modular Bin Strips

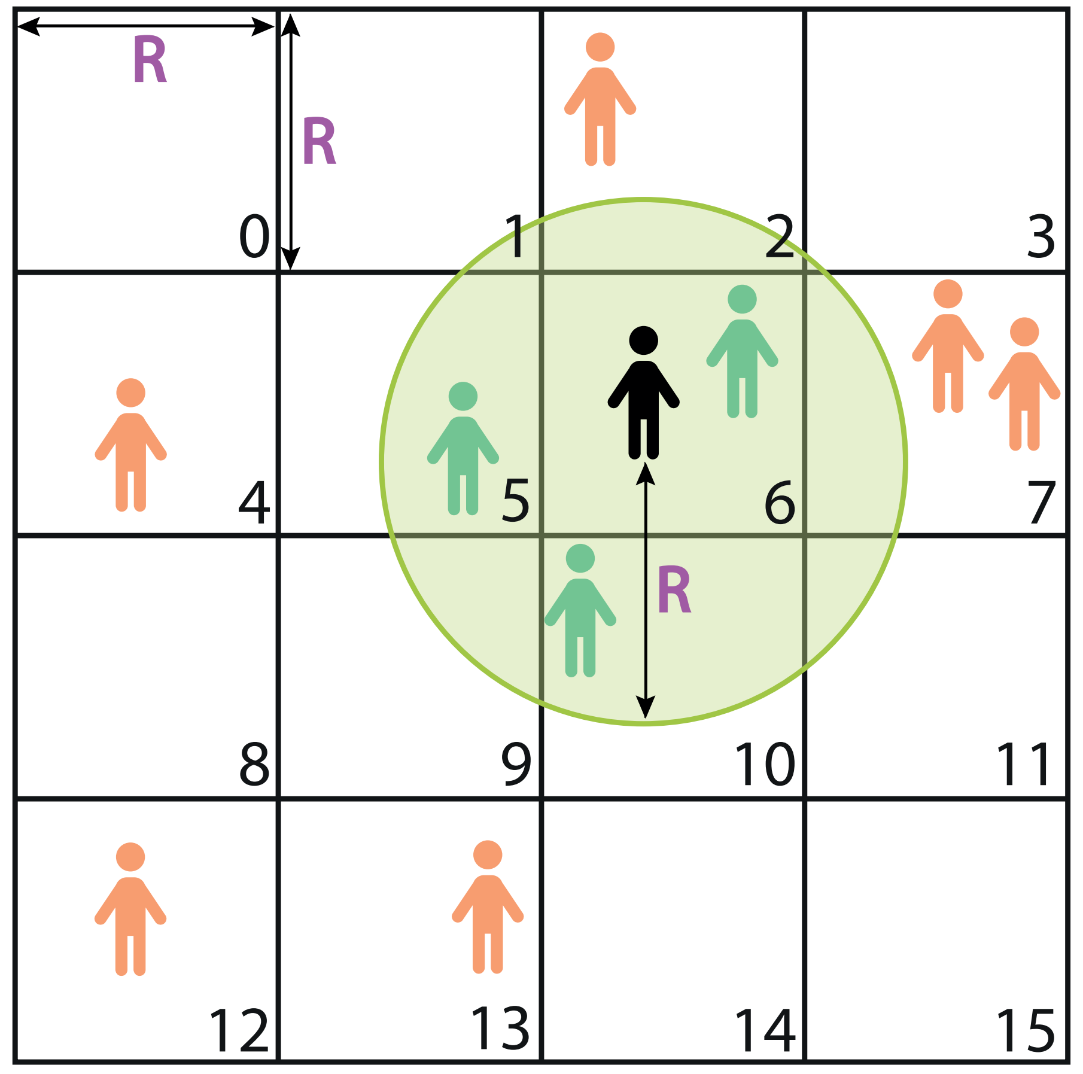
R. Chisholm, Dr P. Richmond, Dr S. Maddock. {R.Chisholm, P.Richmond, S.Maddock}@sheffield.ac.uk
Department of Computer Science, University of Sheffield, UK

Motivation

- Many complex systems contain mobile spatial actors which are influenced by their neighbours: particles, people, vehicles, etc.
- Uniform spatial partitiniong is used to manage these actors in simulations.
- Accessing this data structure is often more costly than model logic due to the high level of scattered memory accesses creating cache contention.

This research focuses on techniques for optimising GPU near neighbours to facilitate larger complex systems simulations.

Uniform Spatial Partitioning



1. Decide the interaction radius: R
2. Partition the environment:
   - Uniform cells of height/width R
   - Clamp agents to environment bounds
3. Search the Moore neighbourhood:
   - Ignoring agents outside of the Euclidean distance R

Figures: Top-Right (1-2), Centre-Right (3)

Implementation: Construction

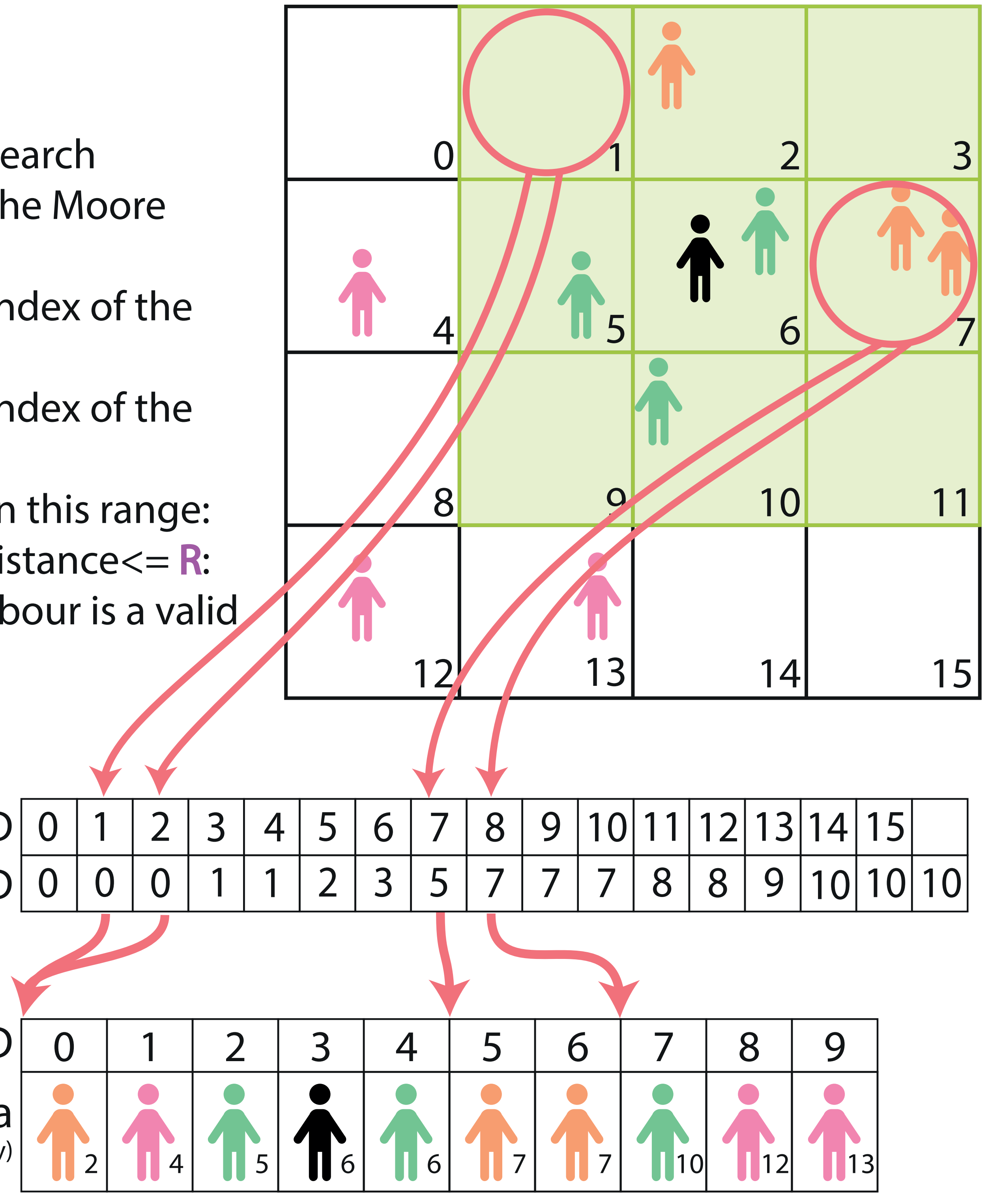
1. Sort the agent data according to their cell indexes into an array
2. Build a boundary index of where each cell's agent storage begins

Figure: Bottom (1-2)

Implementation: Search

1. For each cell in the Moore neighbourhood
2. Locate the first index of the cell's data
3. Locate the first index of the next cell's data
4. For each agent in this range:
   - If Euclidean distance<= R:
     The neighbour is a valid

Figure: Bottom (1-3)



Boundary Index

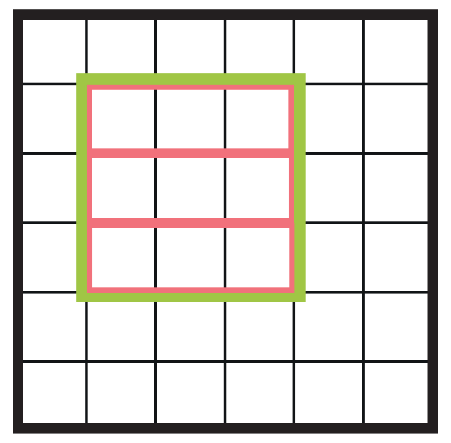
Cell ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Cell Data Start ID	0	0	0	1	1	2	3	5	7	7	7	8	8	9	10	10	10

Agent Storage

Array ID	0	1	2	3	4	5	6	7	8	9
Agent Data (Cell ID included for clarity)	2	4	5	6	6	7	7	10	12	13

Optimisations

Strips



- We search the Moore neighbourhood of bins about a target location, this consists of a 3x3 grid

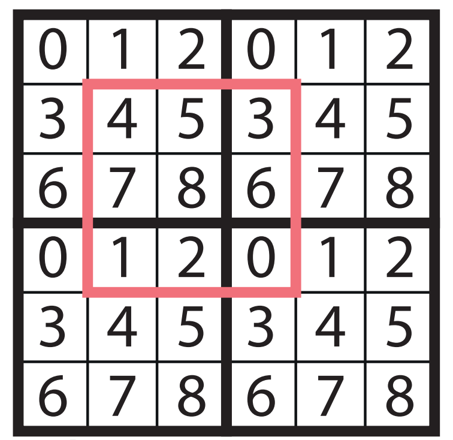
Figure: Right, Green

- Each row of this grid contains 3 contiguous bins (as stored in memory), therefore we can treat each 3-bin strip as a single bin

Figure: Right, Pink

- This modification decreases the number of bin changes from 9 to 3. This reduces redundant memory reads and the branch divergence that occurs when warp threads are operating on bins of differing sizes.

Modular



0	1	2	0	1	2
3	4	5	3	4	5
6	7	8	6	7	8
0	1	2	0	1	2
3	4	5	3	4	5
6	7	8	6	7	8

- In the existing technique each thread starts searching from the same point in the Moore neighbourhood, traversing the remainder of bins in the same order.
- This creates a potential for every bin to be accessed simultaneously during a single iteration of the neighbourhood search.
- Instead the environment is subdivided into bins in 3x3 groups.
  - Each bin within these groups is labelled 0-8, such that any 3x3 Moore neighbourhood in the environment will encapsulate 9 bins labelled 0-8.
  - Bins of each Moore neighbourhood are now iterated in label order.

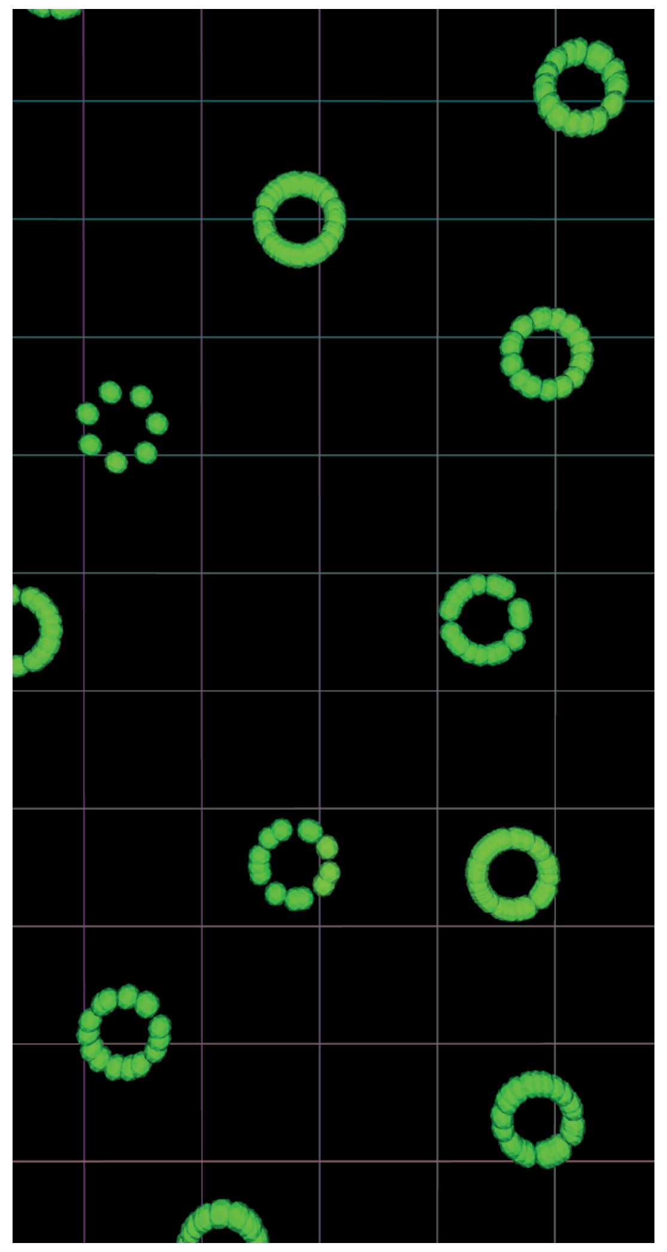
Figure: Right, Pink

- This modification reduces the initial memory accesses to only access 1/9th of the bins.

Hybrid

- The Strips technique only affects the first dimension.
- The Modular technique can be extended to higher dimensions. This reduces initial memory scatter to 1/27 in three dimensions.
- It is possible to apply the Modular technique to the remaining dimensions after Strips.

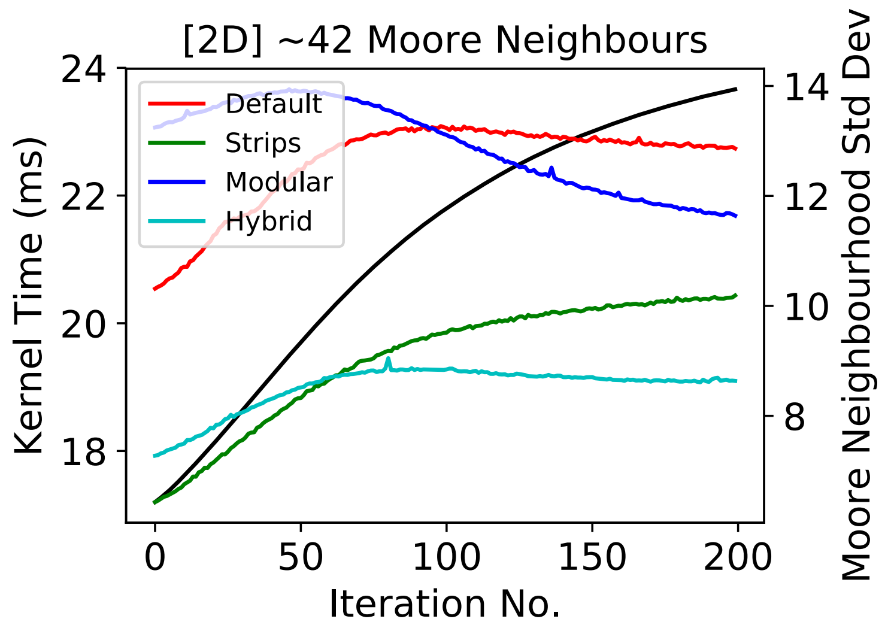
Experiments

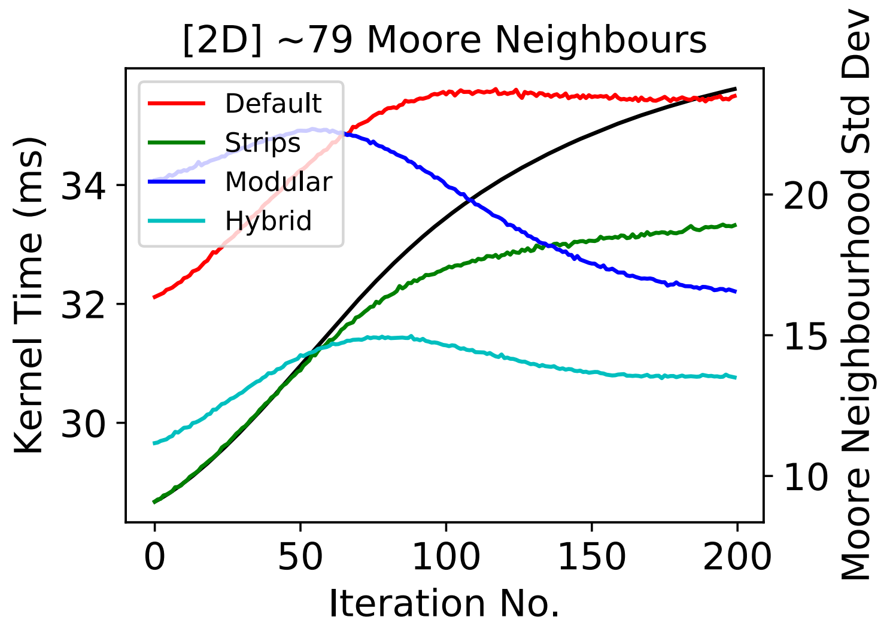


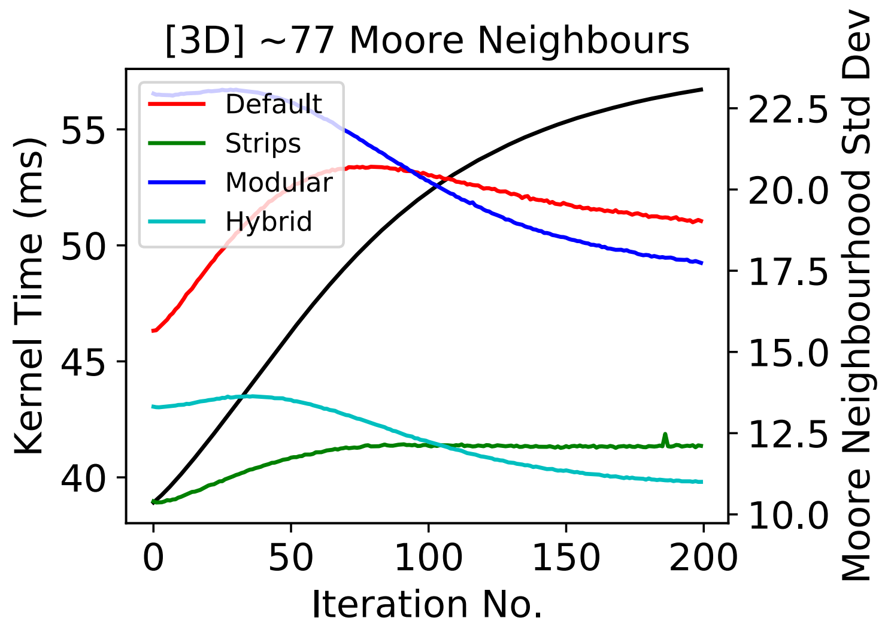
Complex systems simulations see spatial actors transition through a variety of distributions. To capture the influence these distributions have on the performance of each technique, experiments utilised the 'Circles' benchmark model.

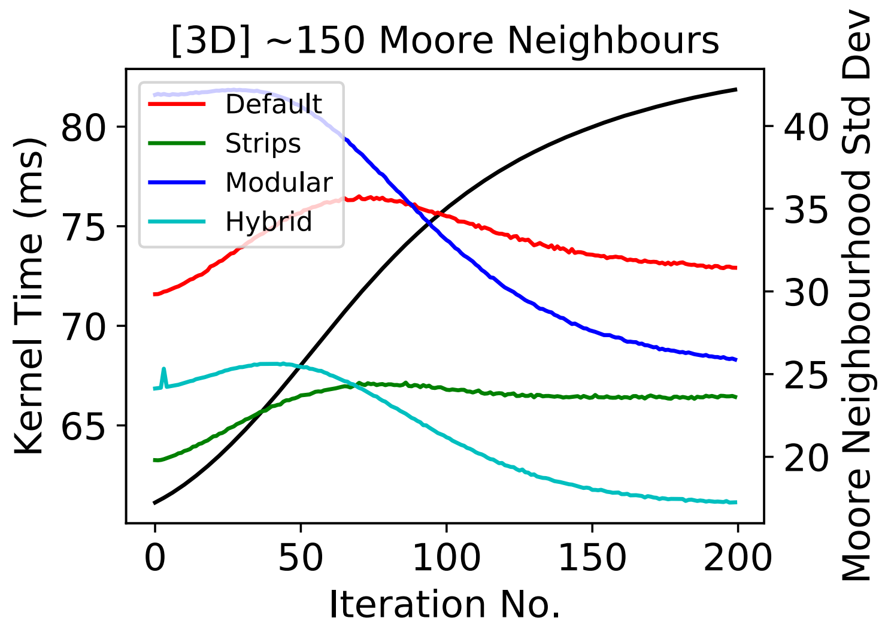
The Circles benchmark model is an analogue of general N-Body models. Actors are initialised in a uniform random distribution. As the simulation progresses the actors move towards a steady state whereby they are arranged in circles in two-dimensions, or spheres in three-dimensions. A force modifier can be adjusted to change the time taken to reach the steady state. Performance is captured alongside the standard distribution of neighbourhood sizes at each timestep of the simulation. This enables visualisation of how the uneven distribution of actors affects performance.

Results









Each graph (right) shows the performance of the four techniques during execution.

- Strips consistently outperforms the existing technique (Default).
- The performance of Strips and Default, both diminish as the standard deviation of agent distribution increases.
- In each case, Strips is initially the fastest technique, whilst the population is uniformly distributed.
- The performance of Modular and Hybrid both improve towards the end of the benchmark when the population has diverged. This improvement is emphasised more in higher density actor populations.
- In each case, Hybrid is the fastest technique by the end of the benchmark, when the population has formed clusters.
- The peak improvement measured in 2D was 1.19x.
- The peak improvement measured in 3D was 1.28x.
- The optimisations provide greater speedup in 3D due to the increased number of cells in each Moore neighbourhood.
- These results show that the original technique is less capable of handling imbalanced workloads created by a non-uniformly distributed actor population.

Conclusions & Future Research

The Strips technique provides a near constant time speed up. The standard technique, lacking the presented optimisations is unfavourable for handling actor distrbutions lacking uniformity. The Modular and Hybrid techniques performances improve under the same circumstances.

We hope to classify the distributions found in complex systems, such as crowd and fluids, so that we can further explore their impact and whether they can be detected efficiently to select the most performant search technique. Similarly, due to differences in memory bandwidth between GPUs we will explore how the performance of each optimisation is affected by alternate GPU models and whether this can be estimated from available device properties.