

Run-time analysis of spatial structure: A Cuda implementation of Minkowski Functionals (MKF)

Ruth E Falconer
Alasdair N Houston



School of Design & Informatics
Abertay University

What

Fast analysis of 3D spatial structures that evolve over time beyond means. From imaged (confocal and Computed Tomography) or simulated data (Computational Fluid Dynamics, Turing Patterns Reaction Diffusion Systems). This is widely applicable in the sciences e.g. material science, healthcare, cosmology and environmental sciences.



Why

A known limitation of GPGPU applied to spatially discretised simulation models is the frequency of device-host data transfers necessary to capture events of interest or otherwise provide sufficiently dense data for the robust analysis of evolving state. To reduce this burden, run-time analysis tools (i.e. executing on the GPU) are developed to characterise the spatio-temporal evolution of structure within 3D scalar fields. The Minkowski Functionals (Matheron, 1967) are fundamentally important measures of spatial structure and are used in diverse science and engineering domains. The 3D Minkowski Functionals (henceforth MF's) consist of four measures: Volume, Surface-area, Integral Mean Curvature and Total (Gaussian) Curvature. These measures together describe the geometric and topological properties of objects in 3D space.

Minkowski Functionals - Diffusion

Minkowski functionals constitute a family of order parameters which discriminate spatial patterns according to size, shape and connectivity. Commodity GPUs to perform real-time estimates from large 3D datasets often scalar fields. MF's are computed efficiently from a binary image (where each binary element defines set membership: object vs. background) this image being determined from one or more scalar fields using, in many cases, a simple threshold operation. From the binary image a Binary Pattern Frequency Distribution is determined: this compactly represents the spatial structure information as 256 patterns and has the advantage of being additive such that an image can be subdivided, BPFDF's calculated independently for the parts and then summed to give the complete information. Therefore we turned to CUDA to achieve an efficient GPU implementation. Our test case is a single chemical diffusion from a central point source (initial mass=1.0) on 256^3 uniform lattice for 150 iterations.

Multicore Implementation

We initially implemented the analysis using OpenACC to achieve parallelisation and found that a "multicore" target (executing on CPU) performed as expected but a "tesla" target (executing on GPU) was unsatisfactory. The diffusion code is implemented using the PGI2019 OpenACC compiler suite and we make use of interop features to allow CUDA kernels to access OpenACC controlled buffers on the GPU. The results presented below were generated on NVidia GTX-970M (NVidia Maxwell "tesla5.2") using CUDA10 runtime libraries.

GPGPU Implementation

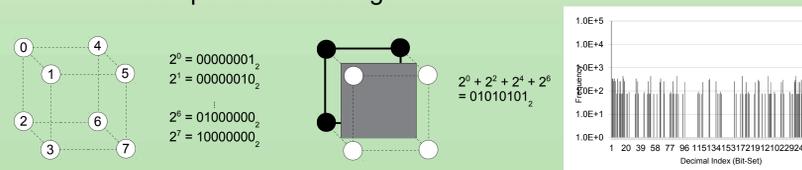
MKF is computed firstly by two CUDA kernels that execute in sequence (non-overlapped in current version) on the GPU and then a final computation on the host CPU:

1. BinMap Kernel: flexible relational operator applied to threshold scalar fields having arbitrary planar/interleaved organisation forming a packed bitmap.

- + High bandwidth requirement of source data mitigated by compact Binary Map (packed bitmap) result

- + Warp SHFL primitives achieve efficient bit packing.

- ? Descriptive statistics/moments might be efficiently computed at this stage.



2. BPFDF Kernel: assemble adjacent elements of the packed bitmap to form binary patterns, generate Binary Pattern Frequency Distribution (256 possible patterns).

- + Small memory footprint, very modest bandwidth requirement.

- + Row-oriented warp threads achieve efficient (shift-mask) processing of packed bitmap.

- + Data privatisation (many partial 16bit BPFDF's temporarily held in shared memory) maximises thread independence.

- + Block-wide coalesced read/write merging of 16bit partial BPFDF's with final atomic add to global 64bit BPFDF.

- Full independence limits block size: 64 threads * 256 bins * 16bits = 32KBytes shared memory.

Performance Results

MKF analysis is carried out at 1, 5, 10 and 25 iteration intervals. 20 replicates of each category

Host-MKF (OpenACC/multicore)- **HFHT**

CUDA-MKF (CUDA10, sequential Kernel execution & buffer copy to pinned mem.) - **CFHT**

Measure interval:	1	5	10	25
OpenACC-multicore scalar field transfer	11.75	5.25	4.43	3.94
stdev	1.48	0.30	0.15	0.06
CUDA-kernels BPFDF transfer	6.58	4.25	3.94	3.77
stdev	3.26E-03	2.84E-03	4.79E-03	4.47E-03

Table 1 Von-Neumann neighbourhood diffusion from central point source (initial mass=1.0) on 256^3 uniform lattice, 150 iterations using OpenACC/tesla5 (multicore) and Cuda 10 on NVidia GTX-970M. Average time in seconds given with variance in brackets n = 20.

Box plots of OpenACC (HFHT) and Cuda (CFHT) performance based on run time analysis frequency



Evolution of Minkowski Functionals



Future Work

Investigate effect of larger blocksize (reduced thread independence) for BPFDF kernel.

Using asynchronous streams to allow some overlap in kernel processing and buffer transfer, possible to hide latency.

Addressing host-CUDA synchronisation latency should further improve performance.

We Anticipate the overhead of the run time analysis becoming small once the communication latency is properly addressed.

Apply run-time analysis to GPGPU version of a well-established microbial model based on a reaction-diffusion system (Falconer, Houston, Portell, & Otten, 2019)

Repo: <https://github.com/DrAI-HFS/MKF>

References: Falconer, R. E., Houston, A. N., Portell, X., & Otten, W. (2019). Hardware Acceleration of Reaction-Diffusion Systems: A Guide to Optimisation of Pattern Formation Algorithms Using Openacc. 2019 Spring Simulation Conference (SpringSim), 1–12. Houston, A. N., Otten, W., Falconer, R., Monga, O., Baveye, P. C., & Hapca, S. M. (2017). Quantification of the pore size distribution of soils: Assessment of existing software using tomographic and synthetic 3D images. Geoderma, 299, 73–82. Houston Alasdair. (2013). Analysis of tomographic images — Abertay University. Retrieved from <https://rke.abertay.ac.uk/en/studentTheses/analysis-of-tomographic-images>. Mantz, H., Jacobs, K., & Mecke, K. (2008). Utilising Minkowski Functionals for Image Analysis: a marching square algorithm. J. Stat. Mech. P12015 J. Stat. Mech, 12. Savage, A., Katz, E., Eberst, A., Falconer, R. E., Houston, A., Harrison, D. J., & Bown, J. (2013). Characterising the tumour morphological response to therapeutic intervention: an ex vivo model. Disease Models & Mechanisms, 6(1), 252–260. Serna, H., Muñuzuri, A. P., & Barragán, D. (2017). Thermodynamic and morphological characterization of Turing patterns in non-isothermal reaction–diffusion systems. Physical Chemistry Chemical Physics, 19(22), 14401–14411.

Acknowledgements: RF acknowledges funding from NERC- NE/S004920/1