A MASSIVE PARALLEL COMPUTING ALGORITHM BASED ON GPU/CUDA FOR PROCESSING LARGE-SCALE MULTIFRACTAL IMAGES

J.Sh. Khursanboyev

Teacher at the Department of Pedagogy and Language Teaching, Samarkand State Pedagogical Institute

Abstract. This paper presents a massive parallel computing algorithm implemented on GPU (NVIDIA CUDA) architecture for the real-time computation of multifractal parameters in high-resolution digital images. The conventional sequential computational complexity of *O(NlogN)* has been optimized through parallel box-partitioning and dynamic shared memory allocation within CUDA threads. Experimental results demonstrate that the proposed parallel algorithm reduces processing latency by 6.0 to 9.5 times compared to CPU-based execution.

Keywords: multifractal analysis, box-counting, parallel computing, GPU, NVIDIA CUDA, speedup factor, real-time processing.

1. Introduction

High-resolution images obtained from satellite remote sensing, aerial photogrammetry, and materials microscopy often exceed several megapixels. Analyzing their internal multi-scale structure through multifractal formalism imposes heavy computational demands. In particular, sequentially evaluating the Rényi partition function across multiple box sizes $\boldsymbol{\varepsilon}$ and moment orders

$\boldsymbol{q} \in \boldsymbol{R}\ [-\mathbf{5},\mathbf{5}]$can take several minutes per image, causing significant bottlenecks in real-time automated decision-making systems. Parallelizing these operations across GPU threads is therefore of critical practical importance.

2. Parallel Algorithm Architecture

In the proposed implementation, image arrays are allocated to the GPU global memory, and processing is organized into three sequential stages:

Parallel Box-Partitioning: The image is partitioned into *B X B* CUDA thread blocks. Each thread evaluates the local pixel intensity sum $I_i(\varepsilon)$ for its assigned box ε leveraging high-speed Shared Memory;

Parallel Reduction: Evaluating the sum of q -th order probability measures $Z(q,\varepsilon) = \sum_i P_i(\varepsilon)^q$is conducted using parallel tree reduction across thread blocks;

Regression and Legendre Transform: Light GPU kernels calculate numerical derivatives for the mass exponent $\tau(q)$ and singularity spectrum $f(\alpha)$ simultaneously.

Experimental Results and Discussion

Benchmarking was conducted on an Intel Core i7 CPU and NVIDIA RTX GPU platform across varying image resolutions (*512 X 512, 1024 X 1024, and 2048 X 2048 pixels*)

Image Resolution(pixels)	CPU Sequential Time (ms)	GPU Parallel Time (ms)	Speedup Factor
512 × 512	480	78	6.15x
1024 × 1024	1850	230	8.04x
2048 × 2048	7420	780	9.51x

As image dimensions increase, the parallelization efficiency improves markedly, achieving a speedup factor of 9.51x on complex 2048 × 2048 images.

4. Conclusion

The developed GPU/CUDA parallel algorithm enables the effective deployment of multifractal analysis within high-throughput intelligent systems, ensuring low-latency execution and seamless synchronization with deep neural networks.

References

1. Potapov A.A. Fractals in Modern Radiophysics and Image Analysis. – Moscow: Radiotekhnika, 2005. – 368 p.

2. Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. – Addison-Wesley, 2010.
3. Arneodo A., Bacry E., Muzy J.F. The thermodynamics of fractals revisited with wavelets // Physica A. – 1995. – Vol. 213, No. 1-2. – P. 232-275