TED University

Real-Time Parallel Drone Swarm Coordination & Simulation

Duygu Can Gündoğan, Ali Kendir, Alp Eren Çakmak, Cem Serimer

ABSTRACT

Simulating coordinated behavior of unmanned aerial vehicle (UAV) swarms becomes a severe bottleneck on traditional CPU-based approaches as the number of agents grows.

This work develops a GPU-parallel simulation framework, built with CuPy, capable of running swarm scenarios with hundreds to thousands of UAVs at near real-time rates.

Within the project conducted with TUSAŞ, swarm dynamics, collision avoidance and task allocation algorithms are ported to the GPU, yielding significant speedup over the NumPy-based CPU reference.

PROBLEM & METHOD

Problem Definition & Motivation

- UAV swarms play a critical role in defense, search-and-rescue and surveillance missions.
- Inter-agent interaction is $O(N^2)$; on CPU, real-time performance breaks down for N>500.
- TUSAŞ requires rapid iteration in its swarm concept development process.
- Commercial simulators (Gazebo, AirSim) focus on single-agent models and are not optimized for large swarms.

System Architecture / Method

- Swarm state tensor (position, velocity, orientation) → transferred to GPU memory
- Boids-based flocking rules (separation, alignment, cohesion) vectorized with CuPy
- **Collision avoidance:** GPU-friendly grid hashing in place of KD-tree
- **Task allocation:** parallel Hungarian / auction algorithm variant
- **Render & telemetry:** asynchronous loop decoupled from the main loop

TECHNOLOGIES USED

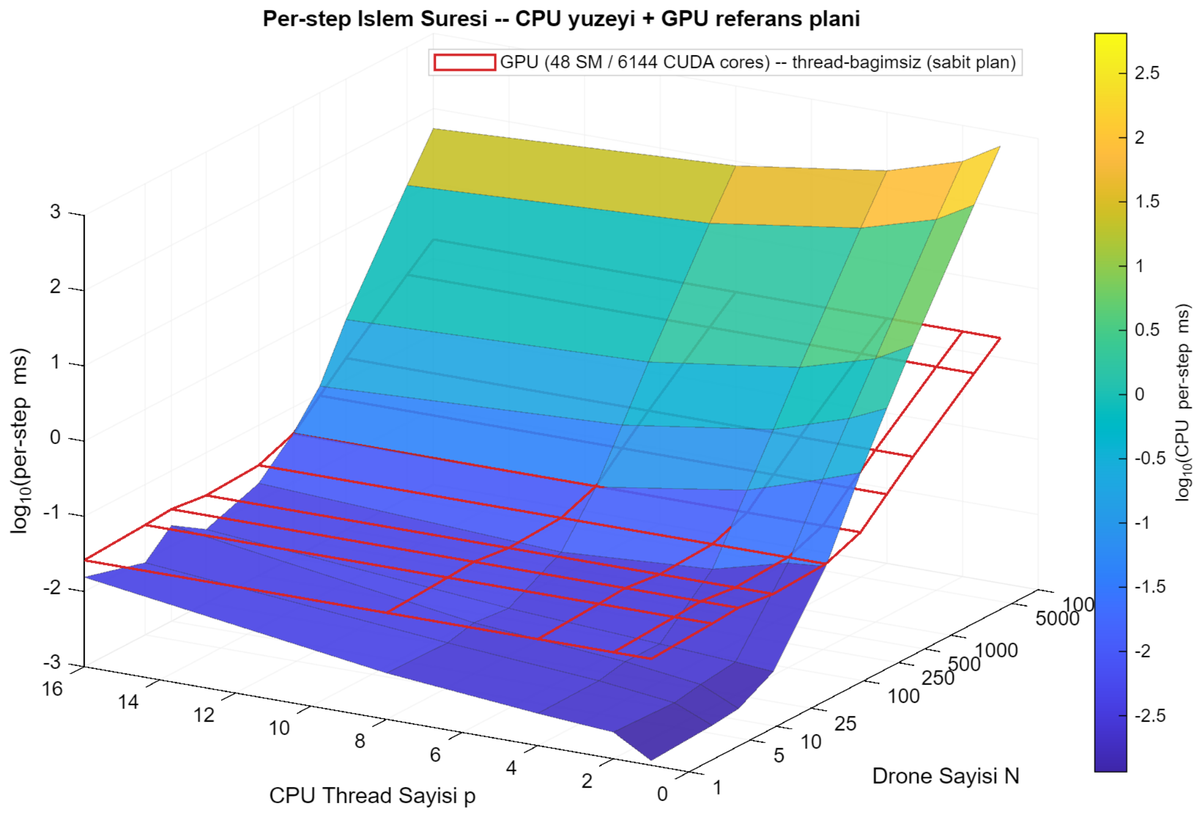
RESULTS & CONTRIBUTION

Experimental Results

Per-step time — best CPU (Numba parallel, 16 threads) vs GPU (CuPy):

- N=1000: CPU 0.63 ms / GPU 0.19 ms → 3.3× speedup
- N=5000: CPU 14.52 ms / GPU 0.94 ms → 15.4× speedup
- N=10000: CPU 54.25 ms / GPU 1.82 ms → 29.8× speedup
- **vs. serial baseline:** GPU reaches 25,875× at N=10000; parallel CPU (t=16) reaches 941×.

Hardware: GPU 48 SM / 6144 CUDA cores; CPU 16 threads. Serial fit R^2=1.0000.



Social Responsibility & Community Impact

- Search-and-rescue after earthquakes, floods and wildfires: large UAV swarms scan disaster zones in minutes instead of hours, helping locate survivors faster and saving lives.
- Public safety for civilians: realistic swarm rehearsals reduce collision and mission-failure risk before any drone flies over populated areas.
- Accessible domestic R&D: an open, lightweight Python/CuPy framework lowers the barrier for Turkish universities, students and SMEs to contribute to national swarm technology.
- Civilian dual-use: same framework supports agricultural monitoring, environmental surveys and traffic/crowd safety — direct public-good applications beyond defense.

Novelty & Contribution

- Lightweight, extensible alternative to commercial simulators, built in pure Python/CuPy.
- GPU implementations of swarm algorithms, validated against a CPU reference.
- Rapid prototyping capability for TUSAŞ swarm concept studies.

Conclusion & Future Work

GPU parallelization brings large-scale UAV swarm simulation to near real-time on desktop hardware. Next steps: heterogeneous swarms, realistic flight-dynamics models, and multi-GPU scaling.

References

Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model.

Okuta et al. (2017). CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations.

Genç Beyinler Yeni Fikirler-ANKARA
Proje Pazarı ve Bitirme Projeleri Ortak Sergisi