

Plans for further development of the CA Track Finder on GPU

Grigory Kozlov

FIAS group of Prof. Dr. Ivan Kisel

Plans for further developments

Phase 1 developments (October 2026 -> <...> -> December 2026)

GPU-Resident Data Management | 5-10 working days

- Upload hits once; build overlapping windows and index-based batches on GPU, eliminating CPU repacking and repeated transfers.

Inter-Window Conflict Resolution | Dec26

- Resolve duplicate tracks and shared-hit conflicts.

GPU-Driven Tracking Control | Dec26

- Minimize CPU intervention and synchronization within tracking.

GPU Lookup-Table Triplet Search | 3-15 working days

- Port lookup-window triplet construction to GPU, with configurable switching between Kalman-based and lookup-table methods.
- Maximize agreement between CPU and GPU reconstruction results.

Update and improve KF-based triplet finder | Dec26

Phase 2 developments (2027)

GPU Tracking Modernization

Refactor GPU tracking using architectural patterns established in recent GPU developments.

- Standardize XPU integration, kernel interfaces, and execution control.
- Separate algorithm logic from data management through clear, modular interfaces.
- Make GPU memory ownership explicit and reuse buffers to reduce allocations and transfers.
- Consolidate shared utilities and improve diagnostics and error handling.
- Strengthen CPU-GPU validation and simplify integration of new tracking methods.

Unified CPU-GPU Tracking Core

Share tracking mathematics and algorithmic logic while preserving efficient single-threaded CPU execution and parallel GPU processing.

- Consolidate mathematical routines and candidate-selection rules.
- Provide specialized CPU and GPU steering through a common interface.
- Adapt data layouts and memory management to each platform.
- Use compile-time specialization to minimize code duplication and runtime overhead.