

In-Memory, Multi-Camera, Real-time Sports Analytics at the Edge

Project Theme: 2.3: New Computing: Memory-Centric Autonomous Agent Computing

PIs and PhD students: Lakshminarayanan Subramanian (PI), Anirudh Sivaraman (Co-PI), Denis Rybkin (PhD student). Courant Institute of Mathematical Sciences, New York University, Emails: {lakshmi,anirudh}@cs.nyu.edu,denisrybkin@nyu.edu

Research Abstract and Goals: Video analytics systems for real-time contexts like live sports analytics inherently rely on expensive cloud infrastructure to process complex queries. To address these challenges, this project introduces EdgeGrid, a memory-centric architecture where edge devices operate as autonomous agents with episodic memory capabilities, enabling complex video analytics directly at the edge. Unlike conventional systems that stream raw video to the cloud, EdgeGrid treats each edge device as an autonomous agent that builds and maintains semantic memory representations, evolving from raw sensory input to queryable knowledge structures as demonstrated in Figure 1. Edge agents continuously construct in-memory object trajectories and episodic storage, supporting SQL-like queries for complex interactions between moving objects while sharing only lightweight metadata with cloud resources. This approach directly addresses the challenge of memory-centric computing for autonomous systems, where memory is not just passive storage but an active substrate for reasoning and learning.

Duration: Fall 2025 - Fall 2026

Research Context and Objectives: Imagine a multi-camera environment at a live sports event, where the goal is to derive a summarized and condensed video highlight as a response to a user query involving a specific play or the interaction between a set of players across multiple camera feeds in real-time with no human intervention. In essence, given an input query from a user, we need a distributed query processing system that can perform in-memory, real-time analytics at each camera, that can perform high-fidelity object detection and tracking, and provide a condensed video response for the query that can be collectively summarized as a video highlight by a query server (demonstrated in Query 1). Addressing this problem is a very challenging proposition with the need to address four fundamental challenges in memory-centric autonomous agent computing:

Lightweight, In-Memory, On-Device Query Processing: Edge devices, including cameras, have limited memory and limited compute abilities. Given a complex query, an important first challenge is to decompose the query for each edge device and enable complex reasoning directly on memory-resident video data in real-time. We will create an SQL-based query abstraction where users can specify edge queries on object trajectories to derive compact video summaries that operate efficiently over different forms of device memory representations (see Table 1 for performance comparison).

Autonomous Memory Management: How can edge agents organize, adapt, and reason over episodic memory under physical constraints? We will develop computational frameworks for managing context-aware memory where agents semantically fuse redundant interaction history into compact representations that enable complex query processing at the edge over live data, compact historical video data, and interaction user query responses (illustrated by Query 2).

Multi-Camera Identity Stitching: To enable distributed query processing across multiple autonomous agents, we need to maintain a consistent semantic memory of object identities and trajectories across distributed views. To achieve this, we will enable memory sharing protocols where agents hand off object identities seamlessly as tracked entities move between camera views controlled via a central query server that enables abstract user-defined object definitions, live object detection, and continuous tracking of objects across multiple camera views (as shown in Figure 2).

Multi-Camera Video Response Synthesis: Given an abstract user query, the final goal is to derive a condensed video highlight that best synthesizes multiple video responses from individual edge camera devices that capture the best snippets relevant to the user query.

Significance of Research: Current video analytics systems are not well-suited to address the in-memory, multi-camera, real-time sports analytics challenge outlined above. Most existing video analytics systems transmit raw video streams to cloud infrastructure, requiring high bandwidth (>10 Mbps per camera) and introducing latency (>500ms), unsuitable for real-time applications (see Table 2). Sports venues, robotics applications, and surveillance systems increasingly demand immediate insights without cloud dependency. EdgeGrid's memory-centric approach reduces network usage by 90% while maintaining sub-100ms latency (demonstrated in Table 4), enabling new classes of applications: real-time tactical analysis in sports, autonomous robot navigation in disconnected environments, and privacy-preserving analytics where video never leaves the premises. By treating edge devices as autonomous agents with rich episodic memory, we enable a future where intelligence resides at the data source rather than distant data centers.

Research Plan and Technical Approach: We are currently building EdgeGrid as an early prototype system of a multi-camera, in-memory, real-time sports analytics system. EdgeGrid supports execution of expressive, low-latency video queries directly on resource-constrained edge devices, close to where the data is generated. Given a query, EdgeGrid provides edge nodes the ability to process video streams and share highly condensed meta information on object trajectories and vision features relevant to the query with cloud resources to enable complex video analytics over highly resource-constrained environments. EdgeGrid marries an on-edge camera object tracking pipeline with an interactive query engine, enabling users to search live video streams via high-level queries. Edge nodes (smart cameras or edge servers) continuously detect and track objects, generating a stream of metadata – including object identities, classes, trajectories, and timestamps, which is immediately indexed for querying (shown in Query 2 and Table 3). EdgeGrid supports an SQL-based querying pipeline: as new metadata is produced, it is made available through a declarative SQL-like interface that supports complex predicates on object attributes, spatial relationships, and temporal patterns. In this project, we aim to extend EdgeGrid to be a fully functional end-to-end multi-camera, in-memory, real-time sports analytics system addressing all four key research challenges outlined earlier. In addition, we aim to support several in-device memory-centric optimizations to the system (relevant to the Samsung GRO), including:

Memory-type Aware Meta-data tracking and Storage: We aim to extend EdgeGrid to support a novel memory hierarchy where edge devices maintain multiple levels of semantic representation of videos captured at the edge in different forms: (1) Short-term trajectory memory via ByteTrack, maintaining in-memory object states, (2) Medium-term episodic memory storing metadata streams as SQL tables, and (3) Long-term semantic memory utilizing local storage as ring buffers for frame retrieval.

Advanced Query Processing Pipeline: We aim to enable an advanced, declarative query interface that enables EdgeGrid to process more complex query specifications directly on evolving on-device memory structures. A key challenge is to enable real-time query response with compact video or metadata responses in limited memory and bandwidth settings.

Multi-Agent Memory Evolution: We will extend EdgeGrid to scenarios with multiple cameras as distributed autonomous agents. Each agent maintains its own episodic memory while participating in a distributed protocol for identity consensus. When an object moves between camera views, agents negotiate handoff through shared semantic descriptors, maintaining consistent identity without centralized coordination. This requires developing memory fusion algorithms that reconcile overlapping observations into unified semantic representations.

Low-Power Heterogeneous Memory: We will also explore hardware-software co-design opportunities, particularly leveraging emerging persistent memory technologies. We envision

hybrid SRAM-DRAM-NVM architectures where hot trajectory data resides in SRAM, recent episodes in DRAM, and historical patterns in NVM, with in-memory compute at each tier.

Preliminary Results: We outline some of our early results of EdgeGrid in the Supplementary Materials. As demonstrated in the Figures and the query examples, as on-device tracking generates metadata (object IDs, classes, positions, timestamps), this information immediately populates SQL tables (Detections, Frames, Objects), and users can compose queries to find complex patterns across multiple object trajectories (as demonstrated in Query 2). Our current query engine leverages temporal and spatial indexes to execute complex predicates in 5-20ms, enabling interactive exploration of live streams. This elevates memory from passive storage to an active computational substrate. Our early prototype on an Nvidia Jetson Orin achieves:

- *Tracking Performance:* 10 FPS with YOLOv8+ByteTrack, MOTA ~72% accuracy on soccer videos (Table 1)
- *Query Latency:* Simple selections in 1-5ms, complex spatial joins in 5-20ms
- *Network Efficiency:* Metadata streams use <1KB/frame vs 5-50KB for compressed video (Table 2)
- *Storage Retention:* 2TB enables ~63 days of 720p video in the ring buffer (Table 3)
- *End-to-end Latency:* <100ms from query to frame retrieval (Table 4)

Milestones

Q1 (Months 1-3): Implement core memory-centric architecture with SQL query engine on a single-camera setup. Evaluate on sports analytics benchmarks.

Q2 (Months 4-6): Develop multi-agent coordination protocols for cross-camera identity stitching. Deploy a 4-camera testbed covering a soccer field.

Q3 (Months 7-9): Integrate heterogeneous memory management with semantic compression algorithms. Achieve 100-day retention at 1080p.

Q4 (Months 10-12): Complete system evaluation on real deployments. Submit findings to top-tier venues (SIGCOMM, NSDI, IMWUT, Mobicom, Mobisys, Sensys).

Expected Outcomes and Results: Tangible outcomes include: (i) EdgeGrid platform supporting memory-centric sports video analytics; (ii) Demonstration system processing 10+ camera feeds with <100ms query latency; (iii) Published protocols for distributed episodic memory management. Additional intangible outcomes include: (i) A new distributed edge computing paradigm treating edge devices as autonomous agents with diverse memory specifications; (ii) New data frameworks for in-memory query processing on continuously evolving data; (iii) New systems foundation for privacy-preserving analytics where raw data never leaves the edge. This research will fundamentally reshape how we architect intelligent edge systems, moving from passive sensors to active memory-centric agents capable of complex reasoning.

Expected Budget: \$150K (PhD student support for one year, limited PI summer support, hardware, compute resources and limited budget for travel, along with associated indirect costs)

Proposal Appendices - Resources & Others

- Supplementary Figures and Tables
- CVs of PIs: Lakshminarayanan Subramanian, Anirudh Sivaram
- Proposed Graduate Students CV: Denis Rybkin

Current External Funding for this Project: None

Pre-Existing IP: None

Supporting Figures and Tables

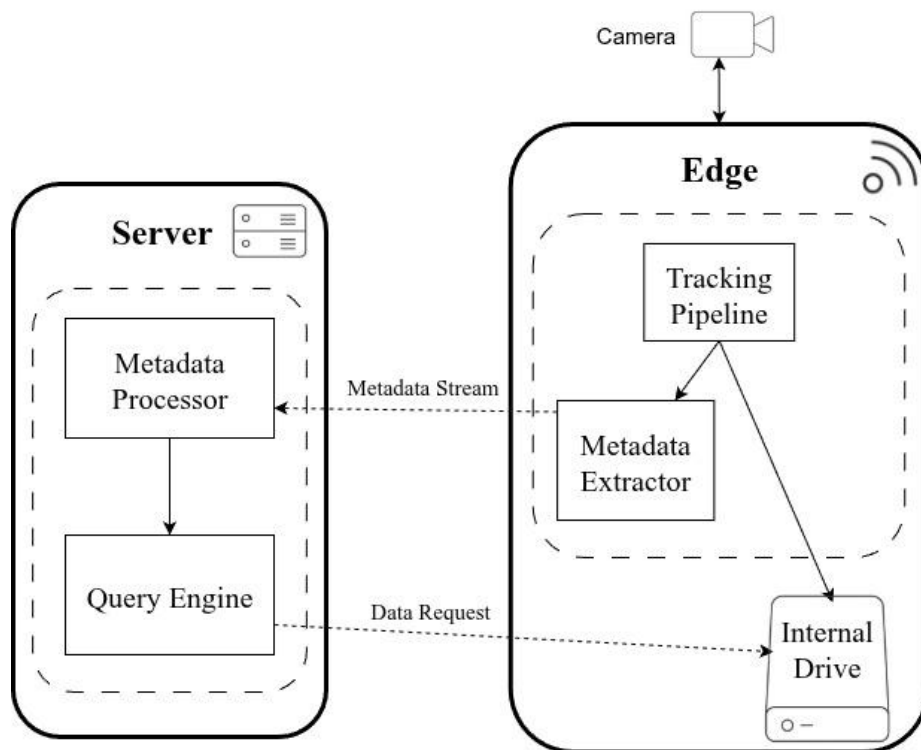


Figure 1: EdgeGrid architecture: server-edge split

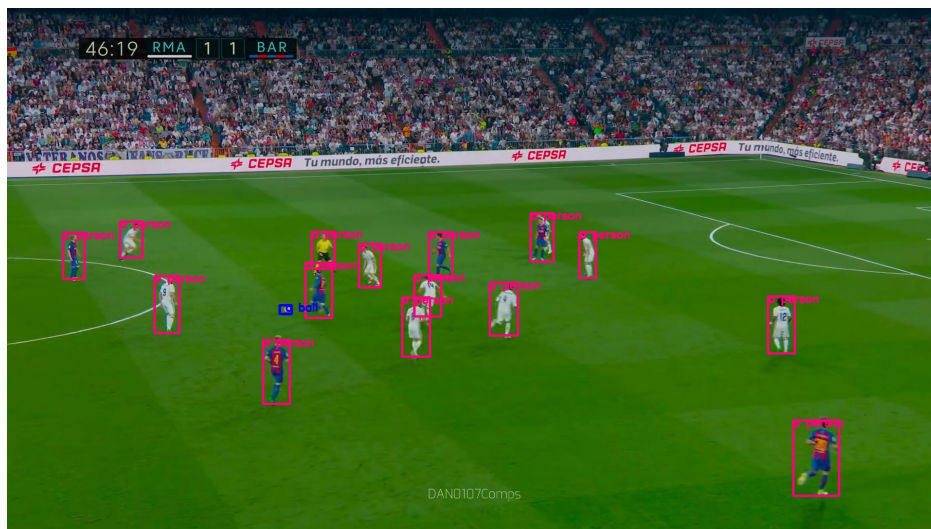


Figure 2: Example of nanoOWL performing real-time open-vocabulary detection on the edge.

```
SELECT timestamp , x , y
FROM Detections
WHERE object_id = 7;
```

Query 1: Retrieving all timestamps and positions of a particular player by ID

```
SELECT T.frame_id, T.object_id AS ball_id, U.object_id AS player_id
FROM Detections T JOIN Detections U ON T.frame_id = U.frame_id
WHERE T.label='ball' AND U.label='player'
AND sqrt((T.x-U.x)^2 + (T.y-U.y)^2) < 5;
```

Query 2: Find player-ball interactions within 5 meters

Method	Capabilities	Throughput	Accuracy
YOLOv8 + ByteTrack	Detect & track objects	~10 FPS	MOTA ~72% (soccer video)
NanoOWL (OWL-ViT)	Open-vocab detect (no ID)	~24 FPS	mAP ~40% (zero-shot)
TAM + SAM	Segment & track anything	<1 FPS	mIoU >90%

Table 1: Performance of tracking pipelines on Jetson Orin Nano (720p video).

Operation	Latency (ms)
SQL execution time	~10
Server → Edge request (over TCP)	~1-2
Edge fetch & read frames (disk buffered)	~5-10
Edge → Server Image Transfer	~10
Reconstruction & delivery	~10
Total estimated end-to-end latency	35-42

Table 2: Server-side latency breakdown.

Resolution	Bitrate	Storage Use	Retention
720p	3 Mbps	1.35 GB/hour	~63 days
1080p	5 Mbps	2.25 GB/hour	~38 days
4K	25 Mbps	11.25 GB/hour	~7.6 days

Table 3: Days of video that can be stored in the local frame storage of Jetson Orin Nano (2 TB capacity).

Operation	Latency (ms)
Client registration	34
YOLOv8 + ByteTrack inference (edge)	50
NanoOWL inference (edge)	42
Metadata serialization	2
Metadata transmission (edge → server)	2
Total end-to-end (capture → result)	60-70

Table 4: Edge-side latency breakdown.