

Multiresolution Time

The Fractal Temporal Operator and Glyph Tape for exact history, motifs, branches, and foveated recall

Hayden Lindley · Glyphd Labs

2026-07-27

Contents

Multiresolution Time	1
Abstract	1
1. The problem of one-scale time	2
2. Exact event tape	2
3. Nested temporal cells	3
4. Stable addresses in changing time	3
5. Temporal foveation	3
6. State deltas versus narrative summaries	4
7. Motif discovery and Glyph Macros	4
8. Branches, tombstones, and supersession	5
9. Relationship to content-addressable video	5
10. Relationship to compression	5
11. Temporal queries	5
12. Fractal Temporal Operator	6
13. Model generation as tape	6
14. Evidence state	6
15. Benchmarks	7
Correctness	7
Retrieval	7
Compression	7
Cognition	7
16. Security and privacy	7
17. Limitations	8
18. Conclusion	8
Architecture illustration briefs	8
Source register	8
Public-disclosure and IP boundary	8

Multiresolution Time

Abstract

Time in software is usually represented as an ordered sequence: log entries, frames, messages, events, commits, or samples. Sequential order is essential, but it makes multiscale reasoning expensive. The same stream must support questions about the current instant, the surrounding episode, recurring motifs, long-term drift, branch history, and exact provenance.

The **Fractal Temporal Operator (FTO)** and **Glyph Tape** propose a multiresolution temporal representation in which events retain exact order while also belonging to nested, addressable temporal neighborhoods. Fine-scale detail can be unfolded near an active event; coarser summaries and motifs remain visible farther away. Repeated subgraphs or sequences can be represented through deterministic macros, but the tape never replaces exact history.

The contribution is a disciplined composition of event sourcing, temporal pyramids, stable event identity, branch lineage, motif dictionaries, and foveated access. Geometry and recurrence organize time; they do not establish truth or guarantee compression. Performance claims remain gated behind comparison with ordinary indexes, segment trees, log compaction, dictionary compression, and competent lazy retrieval.

1. The problem of one-scale time

A flat event stream answers:

- what happened next?
- what was the event at this offset?
- which events fall in this interval?

It is weaker at:

- what changed during this episode?
- which prior episode resembles the current one?
- what is the coarse state around this exact event?
- how did this motif recur across branches?
- what detail can be omitted without losing orientation?
- how can a model retrieve one local temporal neighborhood without loading the entire log?

Summaries can answer some questions, but a freeform summary often loses exact event links and changes with the model.

FTO preserves exact events and builds derived temporal structure around them.

2. Exact event tape

The base tape is append-preserving:

```
Event {
  event_id
  event_type
  logical_time
  observed_time
  recorded_time
  branch_id
  actor
  target_refs[]
  payload_ref
  causal_parents[]
  authority
  evidence_refs[]
  receipt_ref
}
```

Exact event identity and order belong to the canonical ledger. The multiresolution structures are derived indexes.

Deleting a temporal pyramid or motif cache must not erase the base tape.

3. Nested temporal cells

Events are grouped into nested intervals:

```
tick
→ beat
→ episode
→ session
→ phase
→ project era
```

The names are domain-specific. A video may use frame, shot, scene, sequence, act. A project may use event, run, phase, milestone, release. A conversation may use message, exchange, topic, session, decision era.

A temporal cell includes:

```
cell_id
level
start_event
end_event
child_cells[]
summary_ref?
state_delta_ref?
motif_refs[]
proof_root
```

Cell boundaries may be deterministic from event count, time, semantic transition, explicit milestone, or a hybrid contract. The boundary method is versioned.

4. Stable addresses in changing time

An event ID remains stable. A cell address may include branch, level, and interval identity.

```
tape://project/P7/branch/main/phase/12/episode/4
```

If later events extend the phase, the cell may gain children or a new version. Historical addresses continue to resolve through lineage.

A branch can fork from an event and create a new sequence without rewriting the parent.

5. Temporal foveation

An active event can retrieve:

- exact current event;
- immediate predecessors and successors;
- parent episode;
- neighboring episodes;
- active causal parents;
- unresolved consequences;
- coarse summaries of distant phases;
- recurring motifs;
- branch alternatives.

The retrieval budget expands detail near the active point and preserves coarser context farther away.

For a model, the resulting context might be:

```
exact current work order
last three relevant events
current phase summary
prior accepted decision
one analogous prior episode
coarse project history
explicit omissions
```

The compiler records which levels and events were included.

6. State deltas versus narrative summaries

A temporal cell can store a deterministic state delta:

```
before_state_refs
events_in_cell
after_state_refs
changed_objects[]
satisfied_constraints[]
new_questions[]
failures[]
```

A model-generated narrative summary is optional and marked derived. The state delta preserves operational meaning.

This prevents the system from treating a fluent phase summary as the source of truth.

7. Motif discovery and Glyph Macros

Repeated event or graph structures can become motifs:

capture → classify → approve → execute → verify

A deterministic dictionary can assign a macro:

```
macro_id
pattern_version
event_type_pattern
parameter_slots
expansion_digest
occurrences[]
```

The macro helps:

- compression;
- retrieval;
- visualization;
- anomaly detection;
- process comparison;
- training data construction.

Expansion remains deterministic and bounded. The exact event tape remains accessible.

A motif is not a causal law simply because it repeats.

8. Branches, tombstones, and supersession

The tape supports:

- fork;
- merge proposal;
- supersession;
- rollback;
- cancellation;
- tombstone;
- replay.

A tombstone preserves that an object or branch existed and was intentionally retired. A merge records contributing branches and conflict resolution.

Temporal structure can show not only the winning history but the alternatives that were considered.

9. Relationship to content-addressable video

Video is one natural application.

frame → shot → scene → sequence → act

An active shot can expose exact frames, nearby events, scene state, object continuity, and branch alternatives. Repeated production motifs can become shot or edit macros.

The same temporal operator can address project history, agent runs, learning progress, economic simulations, and model generation streams.

10. Relationship to compression

FTO may support real storage savings through:

- repeated motif dictionaries;
- deltas between nested state summaries;
- shared event schemas;
- branch deduplication;
- prior-version references;
- fold/2 routes.

The hierarchy itself does not compress arbitrary events. It creates candidate structure and access paths.

Any compression result must count:

- base tape;
- indexes;
- macro dictionary;
- summaries;
- dependencies;
- cold-start reconstruction.

11. Temporal queries

The operator should support:

look(event)
neighbors(event, radius)

```
parent(cell)
children(cell)
branch_at(event)
compare(cell_a, cell_b)
trace_cause(event)
trace_effect(event)
motifs(cell)
state_before(event)
state_after(event)
replay(interval)
```

These are semantic contracts, not mandatory API names.

12. Fractal Temporal Operator

The operator can be understood as:

```
FT0(active_event, depth_budget, proof_policy)
→ {
    exact_neighborhood,
    nested_context,
    motifs,
    branch_state,
    omissions,
    proof_refs
}
```

“Fractal” here refers to recursive multiscale organization and repeated structural motifs. It does not imply a natural fractal dimension or magical compute.

13. Model generation as tape

A language-model run produces:

- prompt and context;
- token emissions;
- segment boundaries;
- tool calls;
- structured deltas;
- verification events;
- returns.

Typed segment boundaries can become independent tape cells. M-flight can entropy-code emissions during generation. The tape preserves exact output and execution context.

This makes model activity inspectable at several scales: token, segment, action, run, project phase.

14. Evidence state

The project corpus includes fractal-tape and Glyph Core concepts, event-sourced token lifecycle, recursive macro inheritance, FOVEA multiscale retrieval, OOC context shelves, content-addressable video, and local model trace artifacts.

A complete normative FTO implementation and benchmark suite is not represented as accepted production evidence in this edition. The paper is a synthesis of specified mechanisms and research direction.

15. Benchmarks

Correctness

- exact event order;
- deterministic cell construction;
- cross-language golden vectors;
- branch lineage;
- macro expansion;
- replay;
- tombstone behavior;
- state-before/state-after.

Retrieval

- bytes/tokens moved;
- query latency;
- active-neighborhood quality;
- omitted critical event rate;
- cache hit;
- branch restore.

Compression

- macro/delta savings versus dictionary and log-compression baselines;
- metadata overhead;
- dependency-adjusted bytes;
- cold-start reconstruction.

Cognition

- human understanding of long histories;
- time to find a prior decision;
- recognition of repeated failure;
- ability to compare branches;
- perceived continuity.

16. Security and privacy

Temporal structure can reveal sensitive patterns even when event payloads are hidden. The system must protect:

- activity timing;
- relationships;
- private branches;
- location histories;
- model prompts;
- health or behavioral sequences.

Controls include scoped views, aggregation, redacted addresses, time-window permissions, deletion/tombstone policy, and no cross-project motif mining without consent.

Macro expansion requires recursion and size limits. Replay must not repeat external side effects.

17. Limitations

Semantic cell boundaries may be unstable. A hierarchy can oversimplify events that belong to several contexts. Motif detection can produce false patterns. Derived summaries can drift. Branch graphs can become complex. Temporal indexes add storage and migration cost.

Human users may prefer ordinary timelines for many tasks. Multiresolution navigation must prove value.

18. Conclusion

Exact history and multiscale time should coexist.

The tape preserves what happened. The temporal operator makes local detail, surrounding episode, recurring structure, and long-term change addressable without asking a model to summarize the entire past on every request.

The core rule is:

Compress and project the path to history; never discard the history that makes the path verifiable.

Architecture illustration briefs

- **P12-F01 — Nested temporal tape:** Exact events at bottom grouped into episodes, phases, and eras, with active event foveation.
- **P12-F02 — Branch-aware history:** Main tape forks into alternatives and later merge proposal, preserving tombstones and exact event ancestry.
- **P12-F03 — Temporal motif macro:** Repeated event sequences mapped to one deterministic macro with parameter slots and exact expansion.

Source register

- **P12-S01** — Fractal Tape and Glyph Core project corpus (2026-07). `bohselecta/fractal-tape / glyph-core`
- **P12-S02** — GlyphCAS Design Report (2026-07-24). `How_we_can_make_CAS_lightning_fast_for_any_LL`
- **P12-S03** — GlyphDrive FOVEA Specification (2026-07-12). `GLYPHDRIVE-FOVEA-SPEC-v0.2.md`
- **P12-S04** — Content-addressable video and OOC-V corpus (2026-07). `fractal-video.zip`
- **P12-S05** — Local model trace export (2026-03-27). `run.export.md`

Public-disclosure and IP boundary

This paper publishes the public-safe multiresolution temporal architecture and benchmark hypotheses. It excludes confidential model traces, private project histories, unpublished algorithms, and patent-claim language. No storage or retrieval advantage is claimed until measured against competent baselines.