# Study #1 — Deterministic Normalization in Managed Libraries

*File-Level Media Normalization at Scale with Adaptive Geocoding Caching*

## 1. Executive Summary

This work started with a simple practical problem: organizing a large personal archive without losing structural truth. What emerged was a real-world benchmark of file-level media normalization at scale.

Across 363,575 media files spanning 25 years, one pattern became clear: archives become structurally simpler, operationally more predictable, and progressively easier to normalize when organization begins at file level.

**Study #1 isolates the most structured segment of that benchmark — 10 managed photo libraries totaling 116,445 files — and highlights five central findings.**

### 1.1 Hidden duplication is significant

**29.6% duplicate ratio**

Even curated libraries carry silent structural duplication created by normal archive lifecycle events — replication, migration, and export workflows.

### 1.2 Metadata quality is high

**Only 3.8% NoGPS**

Managed libraries are overwhelmingly metadata-rich, creating strong normalization confidence and cleaner deterministic outcomes.

### 1.3 Knowledge accumulation dominates

**73.4% accumulated reuse**

Location resolution gradually becomes reuse-driven, shifting normalization from dependency-driven to knowledge-driven.

### 1.4 Cost converges

**~1.05 sec/file stabilized**

Despite mixed workload composition, normalization converges toward a narrow and predictable operating regime.

### 1.5 Structure creates efficiency

**Well-structured archives normalize better**

The strongest finding: structural coherence naturally improves normalization behavior without changing normalization logic.

**Normalize first. Catalog later.**

# 2. Why Organized Libraries Still Drift Structurally

At first glance, modern photo libraries feel organized. Albums, dates, locations, faces, search, and polished browsing experiences create the impression of structural order.

For everyday use, that is usually enough. But underneath that experience layer, archives continue evolving — quietly, incrementally, and structurally.

**Archives grow. Devices change. Libraries get copied. Backups get replicated. Exports create derivatives.**

Most of this happens naturally — not because users are disorganized, but because digital archives evolve over years, often over decades.

**What looks clean on the surface may already contain hidden structural complexity:**

- duplicated assets created during migrations
- inconsistent metadata between sources
- partial geolocation coverage
- orphaned exports and derived copies
- fragmented chronology spread across libraries
- archive layers accumulated through normal device replacement cycles

Inside the catalog, much of this remains invisible. Albums still work. Search still works. Faces still work. Memories still appear beautifully organized.

But structural drift becomes visible the moment someone tries to merge libraries, move platforms, build a long-term archive, or remove duplicates safely.

**A library that looks organized is not necessarily structurally organized.**

Catalogs are excellent at helping people browse media. But browsing is different from normalization.

**Browsing optimizes experience. Normalization optimizes structure.**

**The files remain. And when they remain, structural quality matters.**

# 3. Normalize Before You Catalog

This benchmark started with a deceptively simple question: What changes when media organization begins at file level instead of catalog level?

## That small shift changes everything.

Traditional workflows begin inside software: files are imported, metadata is indexed, albums are created, tags are added, and organization becomes part of the application experience.

That works — until archives become large, fragmented, duplicated, or historically layered. At that point, the catalog begins carrying structural complexity that was never resolved at file level.

## Normalize first. Catalog later.

**In practical terms, normalization means giving each file a deterministic archival outcome based on what the file intrinsically knows about itself:**

- when it was created
- where it was captured, when location metadata exists
- what type of media it is
- whether it is unique or duplicated
- whether its metadata is complete or incomplete
- how it should be structurally placed inside the archive

**This is not about beautifying folders. It is about establishing structural truth.**

**Instead of asking:**
How should this photo appear in a library?

the better question becomes:
**What is the correct canonical representation of this file in the archive?**

Normalization and cataloging both matter — but they solve different problems.

**Normalization → establishing structural truth at file level**
**Cataloging → browsing, editing, tagging, searching, and consuming media**

## Catalogs organize experiences. Normalization organizes archives.

# 4. The Archive Behind This Study

**This was not a synthetic benchmark. It was a living archive.**

To understand how file-level normalization behaves in practice, this study was built on a real archive accumulated over many years through normal use.

**Its composition reflects how personal archives actually evolve:**

- personal photography
- smartphone capture
- video recording
- library migrations
- backup replication
- exported media
- recovered files
- long-term storage accumulation across devices and systems

What makes this important is not only scale — but realism. This archive carries the uneven accumulation, layered history, and operational messiness of decades of perfectly normal use.

**Normal archives are historical systems.**

That is exactly why they are meaningful benchmarks for normalization.

## Benchmark Snapshot

| 363,575 | 2.1 TB | 25 years | Local-first | Deterministic |
|---|---|---|---|---|
| Files | Normalized Output | Media History | Execution | Normalization |

### Execution Environment

Machine — MacBook Pro M2 Pro / 32 GB unified memory
Source — encrypted WD My Passport HDD (4 TB)
Destination — encrypted Samsung Portable SSD T7 Shield (4 TB)

**No cloud. No distributed compute. No remote indexing. No upload pipeline.**

**Consumer hardware. Real conditions. Real archives.**

**Study #1 isolates the most structured segment: 10 managed photo libraries / 116,445 files.**

# 5. What the Archive Revealed

Raw throughput tells only part of the story. The deeper findings come from how the archive behaves structurally over time — how it accumulated, how workload composition shaped cost, and how normalization became progressively more efficient as archive knowledge grew.

**Study #1 reveals five structural patterns.**

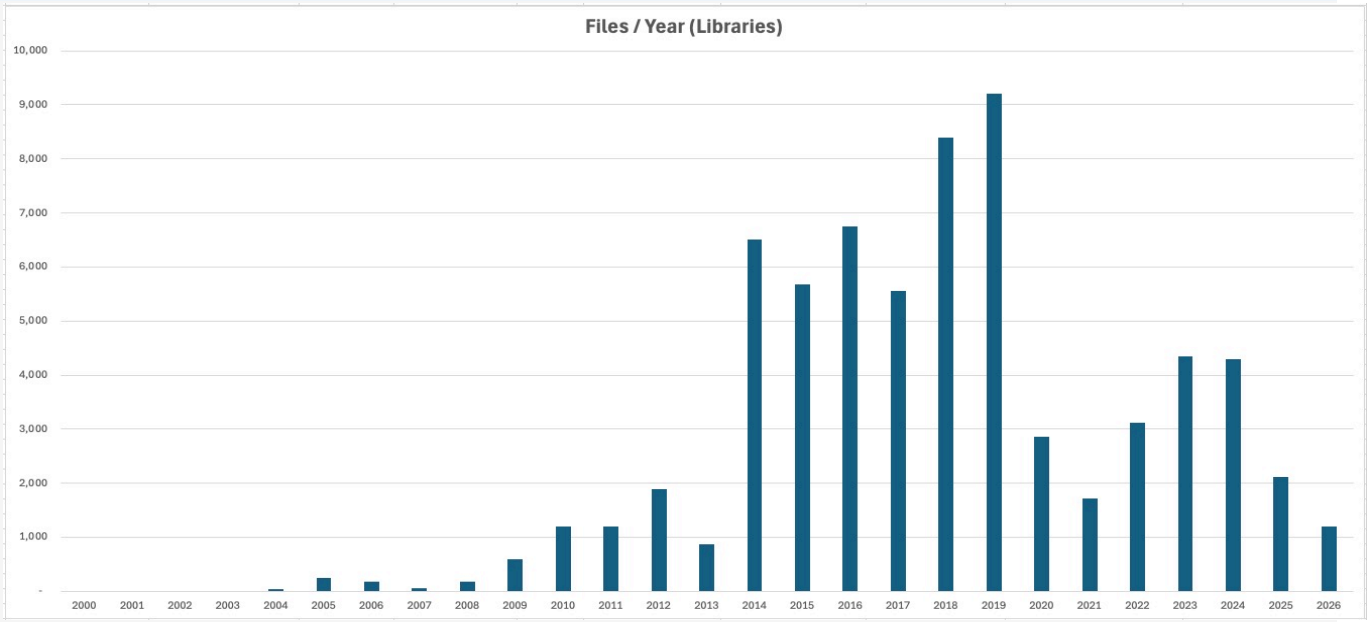
| 116,445 | 29.6% | 3.8% | 73.4% | ~1.05 sec/file |
|---|---|---|---|---|
| Files | Duplicates | NoGPS | Accumulated Reuse | Stable Cost |

**What this shows — A highly structured archive produces predictable normalization behavior.**
Why it matters — Structure becomes operational advantage.

**Hidden duplication is significant. Metadata quality is high. Knowledge accumulation dominates. Cost converges.**

# Archives Are Historical Systems

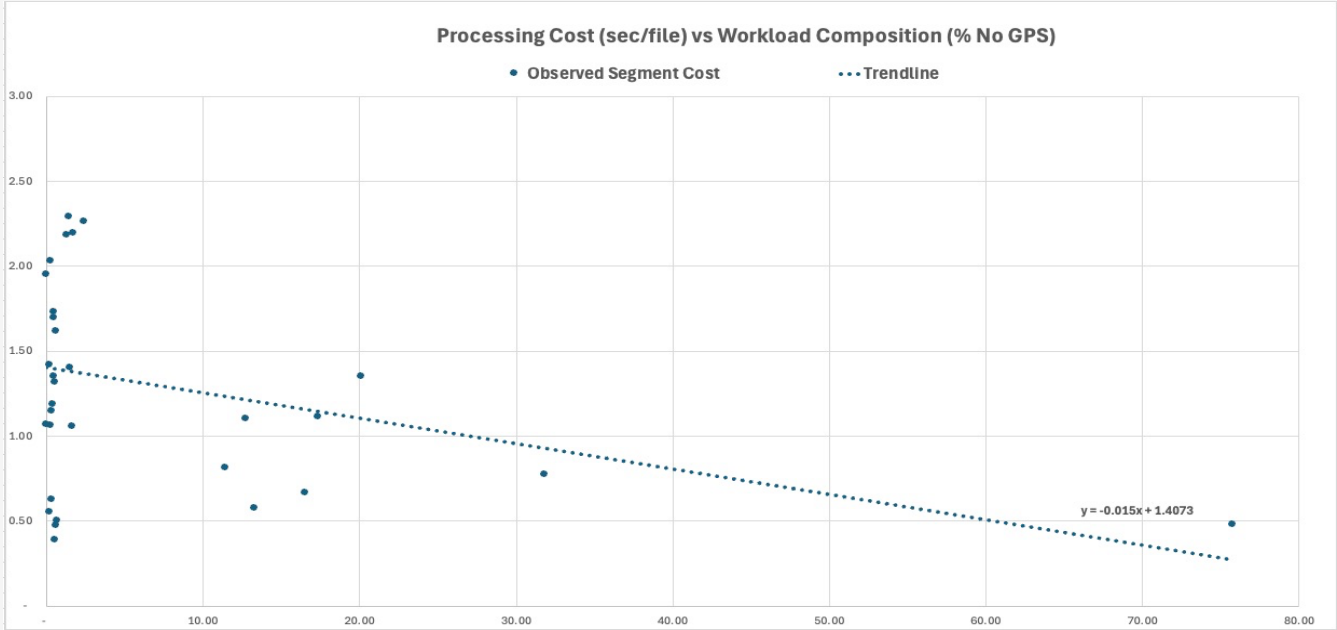
**Large archives are not static datasets. They are historical systems.**




**What this shows — Media accumulation happens in distinct historical phases, not as uniform growth.**
Why it matters — Normalization must operate across changing devices, formats, and metadata regimes.

**Growth is not purely additive.**
Personal archives often expand in structural jumps triggered by normal lifecycle events — device replacement, library migration, backup consolidation, and ecosystem changes. These transitions

frequently introduce silent duplication layers that remain hidden inside otherwise well-managed libraries.



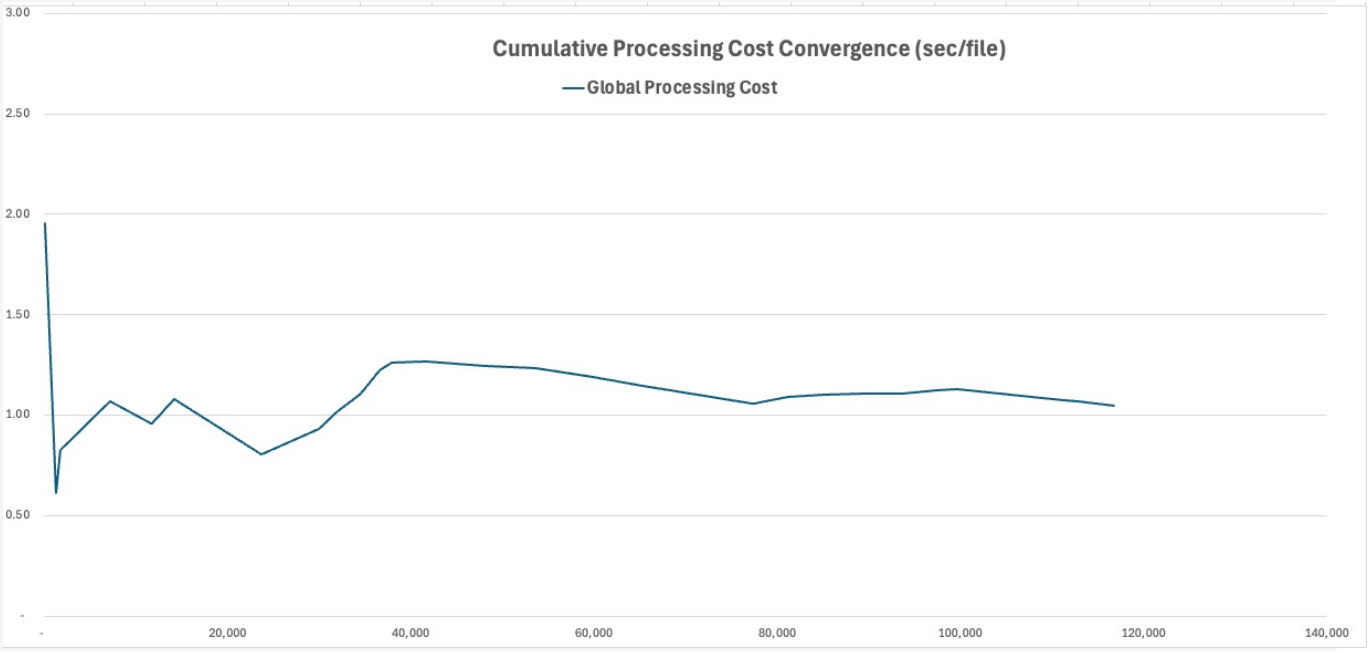

**What this shows — Processing cost varies meaningfully with workload composition.**

Why it matters — Archive composition shapes cost more than archive size.

**Processing cost is shaped more by archive composition than by archive size.**

## Cost Stabilizes Over Time

Normalization moved through three recognizable phases: Warm-up, Adaptation, and Steady Regime.

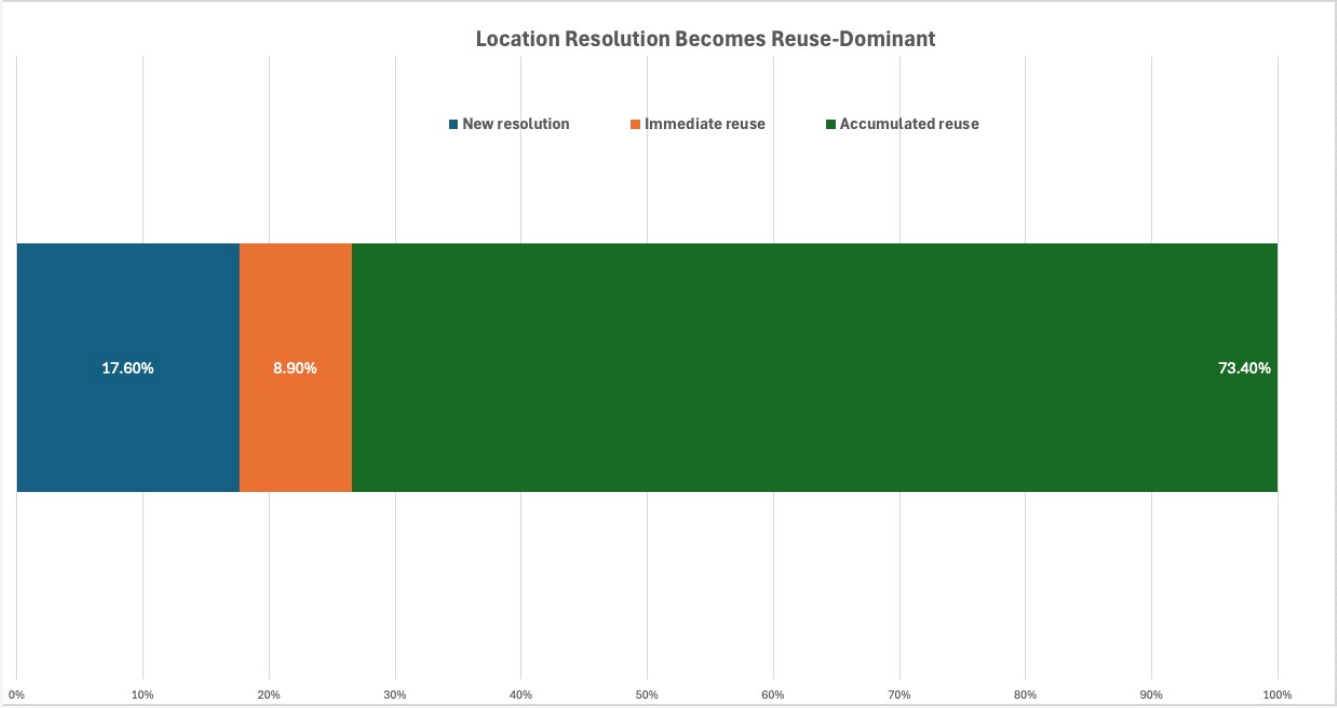



**What this shows — Operating cost narrows toward a stable range over time.**

Why it matters — Predictability increases as archive knowledge accumulates.

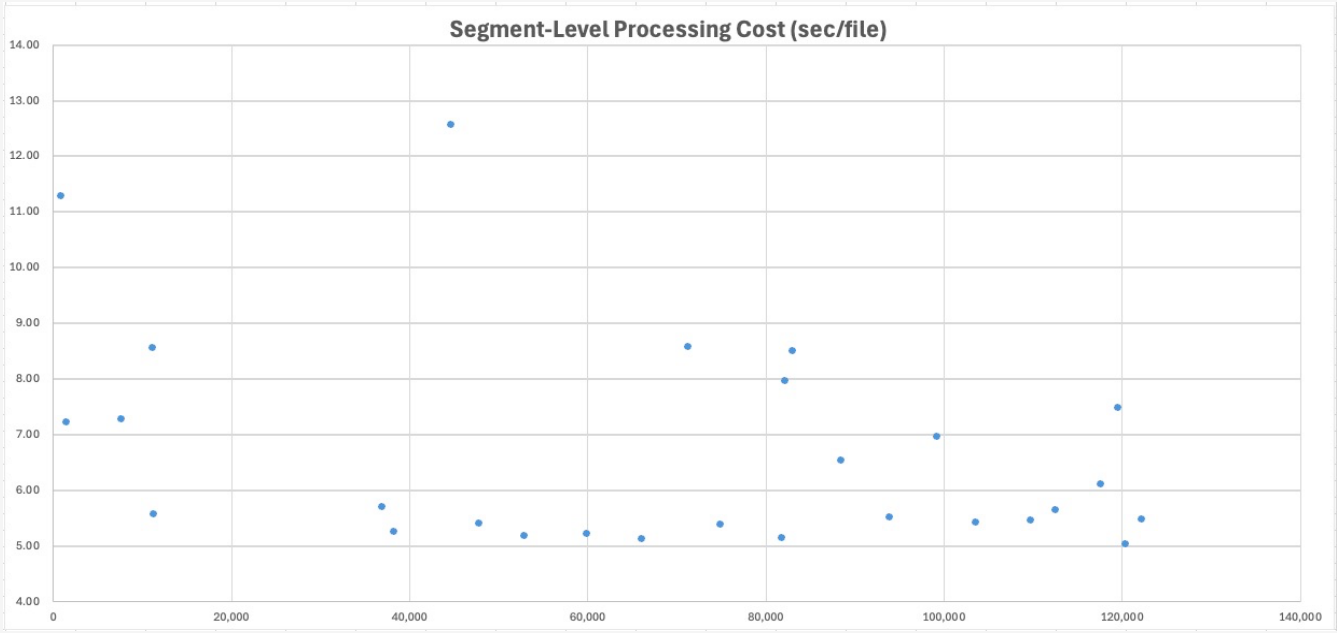
**Normalization efficiency accumulates over time.**

# Normalization Becomes Knowledge-Driven




**What this shows — New resolution becomes minority workload while accumulated reuse becomes dominant.**

Why it matters — Operational dependency declines as archive intelligence grows.




**What this shows — Local segments vary, but global operating behavior converges.**

Why it matters — Mature normalization combines local variability with global predictability.

**Local variability. Global predictability.**

**That is what mature normalization looks like.**

# 6. What This Changes

**File-level normalization is not simply a cleanup step. It changes what an archive becomes.**

### Cleaner archives become portable archives

Normalization makes archives portable, vendor-neutral, easier to validate, easier to back up, and easier to preserve long-term.

### Duplicate propagation stops

Structural duplication stops silently multiplying across years of backups, migrations, exports, and device transitions.

### Uncertainty remains preserved

Incomplete metadata remains visible, preserved, and available for future review — nothing disappears into ambiguity.

### Local-first processing preserves ownership

Keeping normalization local preserves practical ownership, structural ownership, and long-term trust in personal archives.

### A structural layer beneath every catalog

Catalogs organize experience. Normalization preserves archive truth.

**Normalize first. Catalog later.**

# 7. From Principle to Practice

The ideas explored in this study did not begin as theory. They emerged from practical normalization work on a large real-world archive.

**What began as organization became normalization.**

**From that work emerged a practical model:**

- originals remain preserved
- duplicates stop propagating
- unresolved media remains visible
- archives remain portable
- processing remains local
- organization becomes deterministic

Built from that work, MediaOrganizer was developed around those principles in practice — not as a catalog replacement, but as the structural layer beneath future organization.

**Step Zero of media organization. Everything else can come after.**

# 8. What Remains

What began as an effort to organize a large personal archive revealed something broader: archives do not become complex all at once — they become complex gradually, through years of perfectly normal use.

**Hidden structural drift is normal.**

**File-level normalization changes that:**

- hidden structure becomes visible
- duplication becomes measurable
- uncertainty remains preserved
- organization becomes progressively easier
- archives become structurally trustworthy

**Catalogs help people experience memories. Normalization helps archives remain trustworthy.**

Technologies change.
Platforms evolve.
Applications come and go.

## The files remain.

**Normalize first. Catalog later.**

Built from real-world normalization work developed through MediaOrganizer Studio.