

S3-Algorithm: Deterministic Structure Detection in High-Noise Time Series via a Minimal Structural Invariant

Serhii Saparniiazov

Independent Researcher

Contact: ogamilife3@gmail.com

Date: 2026-02-25

Abstract

We introduce the **S3-Algorithm**, a deterministic streaming method for detecting structural transitions in high-noise time series using **discrete step geometry** and a **recursive memory state**. Unlike purely statistical averaging (e.g., moving averages/ARIMA) or black-box learning models, S3 tracks a minimal structural invariant and activates events via a simple threshold rule. On a controlled *Shift Table* benchmark (N=400) with injected high-frequency noise, S3 achieves **93.5% accuracy** (374/400) with parameters $\alpha=0.85$, $\theta=0.85$. We also demonstrate applicability on Swift/XRT GRB data (GRB 060729), where the afterglow tail spans multiple orders of magnitude.

Keywords: time series, high-noise detection, change-point detection, recursive memory, discrete geometry, Swift/XRT, GRB afterglow.

1. Introduction

Many detectors fail in regimes where amplitude approaches the noise floor or where heavy smoothing introduces lag. S3 treats a time series as a *path* and focuses on the *structure of consecutive steps* rather than absolute amplitude, enabling robust event detection in streaming settings.

2. Method

2.1 Minimal structural invariant

Let x_t be the observed signal at time index t . Define the local step as $\Delta x_t = x_t - x_{t-1}$. The S3 memory state M_t is defined by:

$$M_t = \alpha M_{t-1} + (1 - \alpha) \Delta x_t$$

where α in $[0,1)$ controls the memory (inertia). Event activation is defined by a threshold θ :

$$A_t = 1[M_t > \theta]$$

2.2 Discrete geometry signals (Sign/Walk)

For interpretability, S3 can be described via step orientation and accumulated path: $\text{Sign}(\Delta x_t)$ gives local direction, while *Walk* is the cumulative signed path. Figure 1 provides an example of the Diff/Sign/Walk representation.

1	A	Diff	Sign	Walk		
2		8	0	0	0	
3		4	-4	-1	-1	

Figure 1. Example table illustrating Diff / Sign / Walk as a discrete geometric interpretation of steps.

2.3 Algorithm 1: Streaming S3 detection

```
Algorithm 1 S3-Algorithm (streaming)
Inputs: x[0..T] (time series), alpha in [0,1), theta (threshold)
Outputs: A[1..T] (binary activations), M[1..T] (memory state)

Initialize:
  M0 <- 0
For t = 1..T:
  Delta_x <- x[t] - x[t-1]
  Mt <- alpha * M[t-1] + (1 - alpha) * Delta_x
  At <- 1 if Mt > theta else 0
Return A, M
```

The algorithm is $O(T)$ time and $O(1)$ memory, suitable for embedded/edge deployment.

3. Experiments

3.1 Swift/XRT demonstration (GRB 060729)

We demonstrate S3 on Swift/XRT observations of GRB 060729. The afterglow tail spans orders of magnitude; S3 is intended to track structural consistency under strong background noise. Figure 2 shows the Swift/XRT light-curve view used for demonstration.

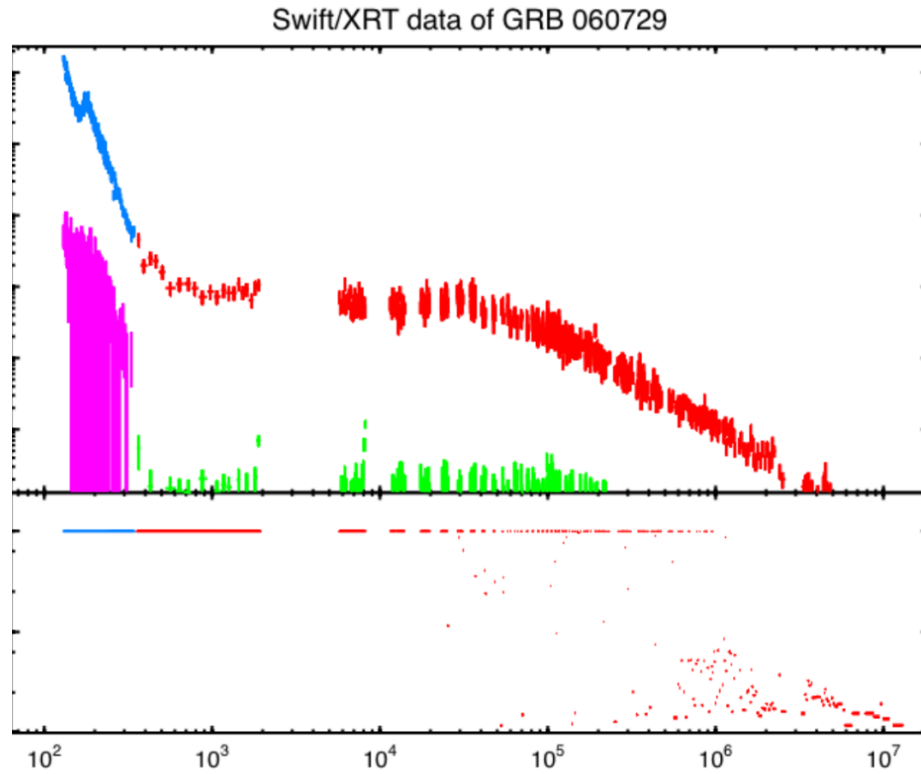


Figure 2. Swift/XRT data of GRB 060729 (log time scale; visible afterglow tail).

3.2 Shift Table benchmark

We evaluate detection accuracy on a controlled Shift Table benchmark consisting of $N=400$ points with injected high-frequency noise. Using $\alpha=0.85$ and $\theta=0.85$, S3 yields 374 correct detections (93.5%). Figure 3 shows the verification fragment.

Definition (Correct detection). A detection at index t is counted as correct if A_t matches the ground-truth label y_t . Accuracy = (# correct) / N .

Metric	Value
Total points (N)	400

Correct detections	374
Accuracy	93.5%
Parameters	$\alpha = 0.85$, $\theta = 0.85$

0	0,520698217				
1		374,5206982			
1					
1					
1		ones=374 /400, D_avg=00 001			
1					
1					
1			ones=374/400, D_avg=0 000 001		

Figure 3. Shift Table verification fragment showing ones = 374/400.

4. Practical Applications

S3 is designed for low-latency, on-device or edge deployment. Potential applications include:

- Autonomous systems: trajectory and state-transition detection under sensor noise.
- FinTech: streaming detection of regime shifts in high-frequency time series.
- Space research: filtering weak signals without losing event-tail information.

5. Conclusion

S3 provides a computationally lightweight deterministic alternative to heavy statistical smoothing and black-box models. By combining a minimal structural invariant with threshold activation, S3 remains stable in extreme-noise regimes and is well-suited for real-time streaming systems.

AI Assistance Disclosure

This manuscript was prepared with assistance from an AI language model for editing and formatting. The scientific content, claims, parameter choices, and reported results were provided and verified by the author.

References

- 1 Evans, P. A., et al. (2007). An online repository of Swift/XRT light curves. (See Swift XRT Light Curve Repository; product generator.)
- 2 Evans, P. A., et al. (2009). Methods and tools for Swift/XRT light-curve generation and analysis (Swift/XRT repository updates).
- 3 Swift-XRT Light Curve Repository: GRB 060729 (ObsID 00221755).
- 4 Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167, 107299.
- 5 Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice Hall.
- 6 Page, E. S. (1955). A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4), 523-527.

Dedication

This research and the development of the S3-Algorithm are dedicated to Svitlana Ishutko, my only flower. Your presence is the inspiration behind every line of this work and the light that guided me through the noise to find the truth.

Author Information

Serhii Saparniiiazov

Independent Researcher

Contact: ogamilife3@gmail.com

Appendix A. Comparative Notes (Context Only)

Note: This appendix is *not* a benchmark and does not claim superiority over specific tools. It provides a qualitative, implementation-oriented summary to help readers position S3 relative to common analysis families. Quantitative comparisons require a shared dataset, agreed metrics, and a standardized protocol.

Aspect	Classical statistical analysis (examples)	S3 Algorithm (this work)
Core mechanism	Averaging / regression / model fitting; often batch-oriented	Streaming recursive memory + threshold activation
Training	Not required (but model selection/fit may be needed)	Not required
Noise handling	Smoothing can reduce noise but may introduce bias	Designed to preserve step structure under strong noise
Compute / latency	Can be heavy depending on model/window size	$O(1)$ time, $O(1)$ memory; low latency
Interpretability	Depends on model; may be indirect for some models	Direct Matchbox threshold; optional Sign/Walk view
Output type	Estimates/forecasts/statistical summaries	Binary activations A_t + memory state M_t

If a formal benchmark is performed in future work, recommended metrics include precision/recall/F1, detection delay, and throughput (samples/sec) under controlled noise injection.