



The 2001 IEEE International Conference on Data Mining

An Online Algorithm for Segmenting Time Series

Eamonn Keogh*, **Selina Chu**, David Hart,
and Michael Pazzani.

University of California, Irvine
Information and Computer Science
Irvine, CA 92697-3425

*Computer Science & Engineering Department
University of California - Riverside
Riverside, CA 92521

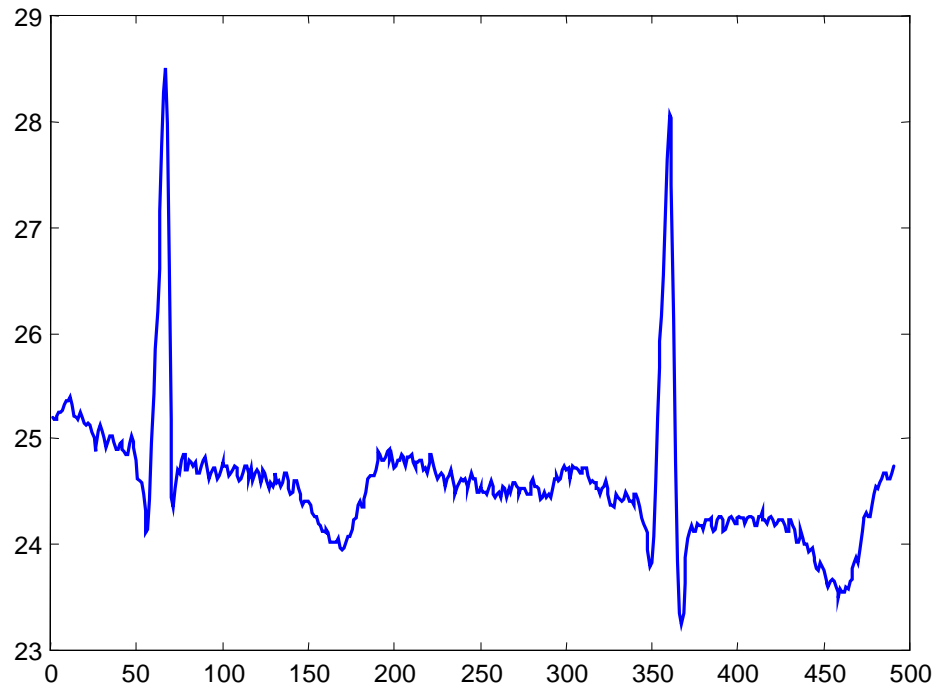
Email the authors for an expanded version of the paper.

Outline of Talk

- What are time series, why mine them.
- Choosing a high-level representation of the data.
- The Piecewise Linear Approximation (PLA)
- Obtaining the PLA
 - Top-Down
 - Bottom-Up
 - Sliding-Window
- The SWAB algorithm.
- Experimental Results.
- Conclusions/Future work.

What are Time Series?

A time series is a collection of observations made sequentially in time.

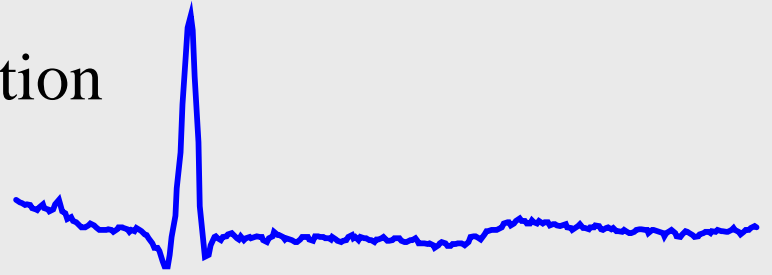


25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750
••
••
24.6250
24.6750
24.6750
24.6250
24.6250
24.6250
24.6750
24.7500

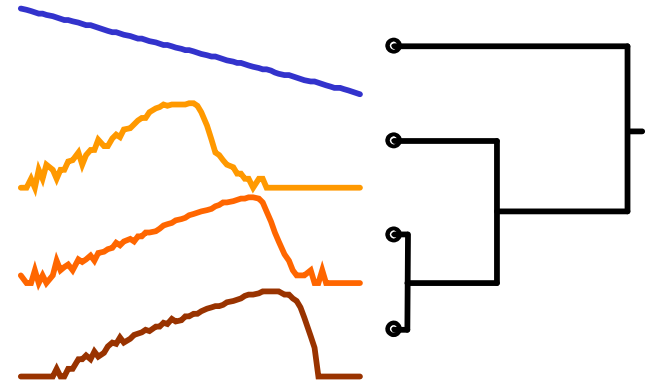
People measure things, and things (with rare exceptions) change.

Time Series Date Mining

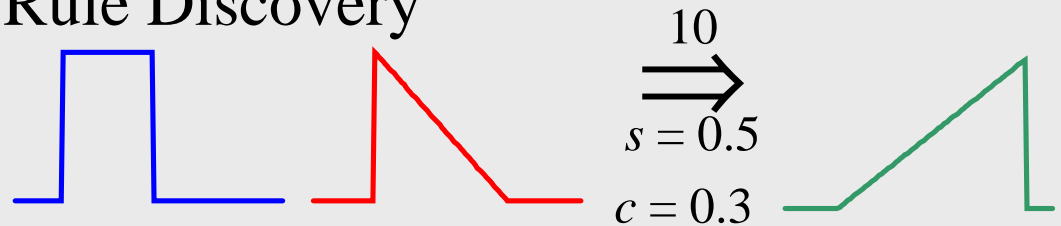
Classification



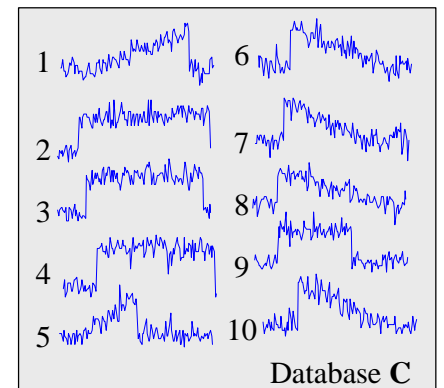
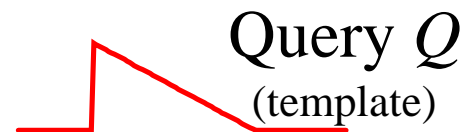
Clustering



Rule Discovery



Query by Content

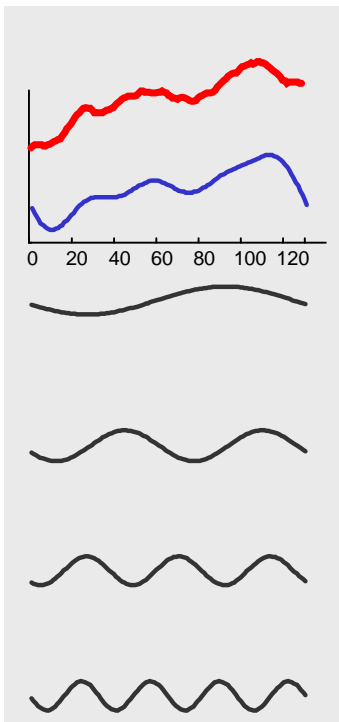


Why an Approximate Representation of the Data?

- One Hour of ECG data: 1 Gigabyte.
- Typical Web-Log: 5 Gigabytes per week.
- Space Shuttle Database: 158 Gigabytes and growing.
- Macho Database: 2 Terabytes, updated with 3 gigabytes per day.

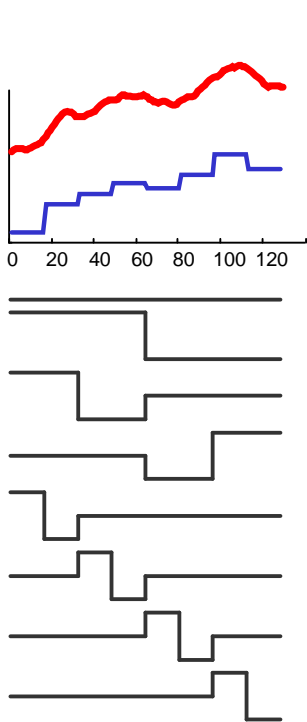
We need a representation of the data that we can efficiently manipulate.

In most cases, patterns, not individual points, are of interest.



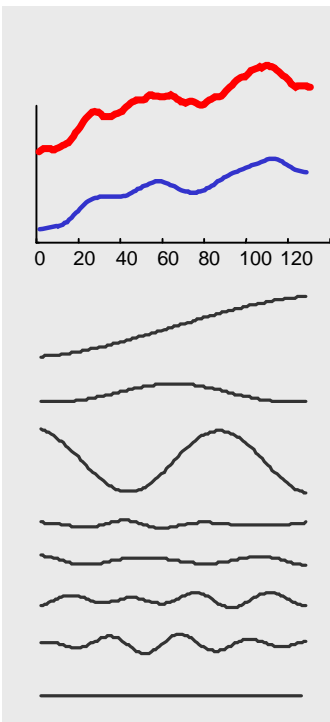
DFT

Agrawal, Faloutsos, &
FODO 1993
Faloutsos, Ranganathan, &
Manolopoulos. SIGMOD 1994



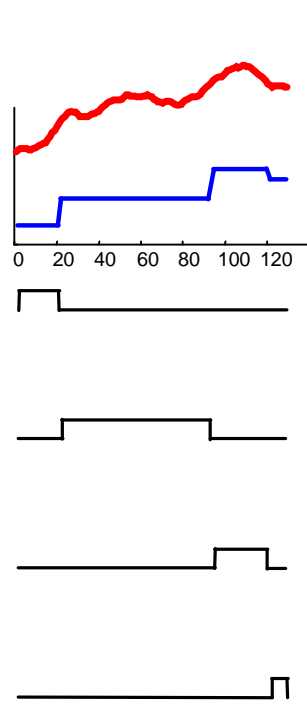
DWT

Chan & Fu. ICDE 1999



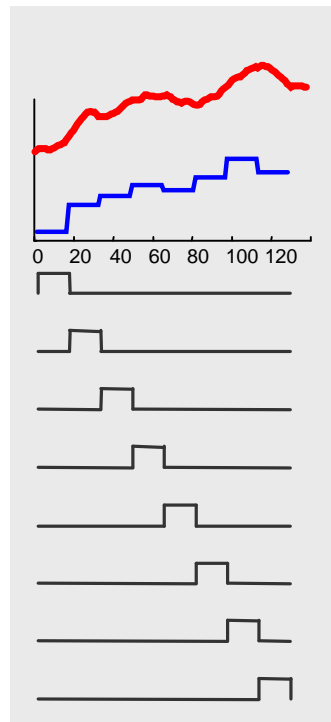
SVD

Kom, Jagadish &
Faloutsos. SIGMOD 1997



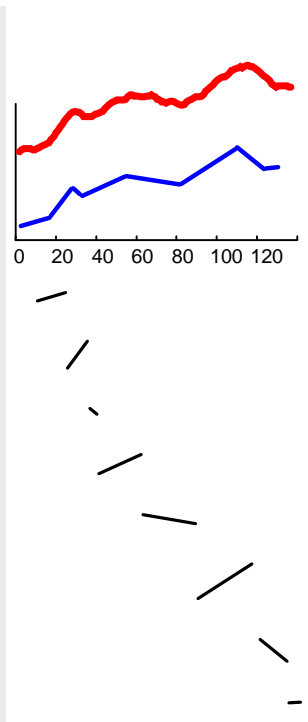
APCA

Keogh, Chakrabarti, Pazzani &
Mehrotra SIGMOD 2001



PAA

Keogh, Chakrabarti, Pazzani &
Mehrotra KAIS 2000
Yi & Faloutsos VLDB 2000

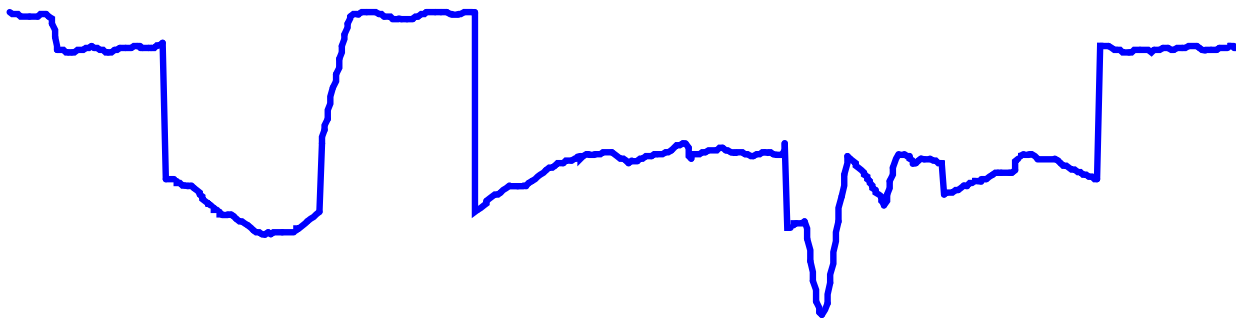


PLA

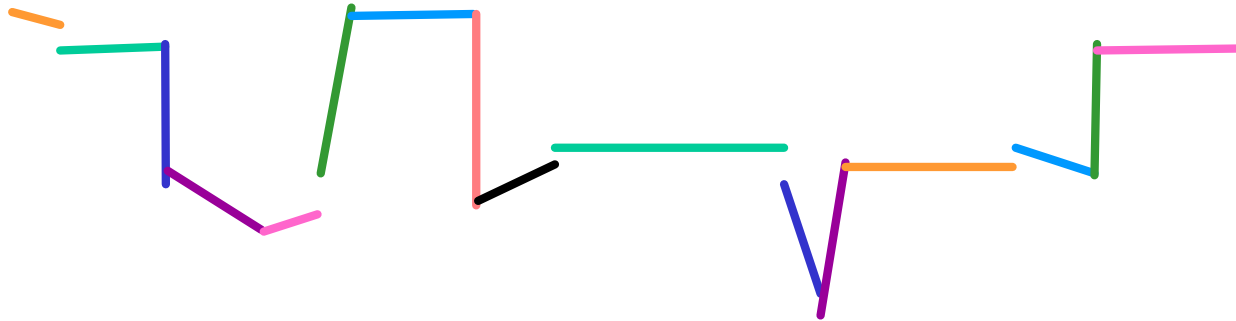
Morinaka, Amagasa, &
Yoshikawa, PAKDD 2001
Uemura, PAKDD 2001

Piecewise Linear Approximation

Here is a time series with n points:



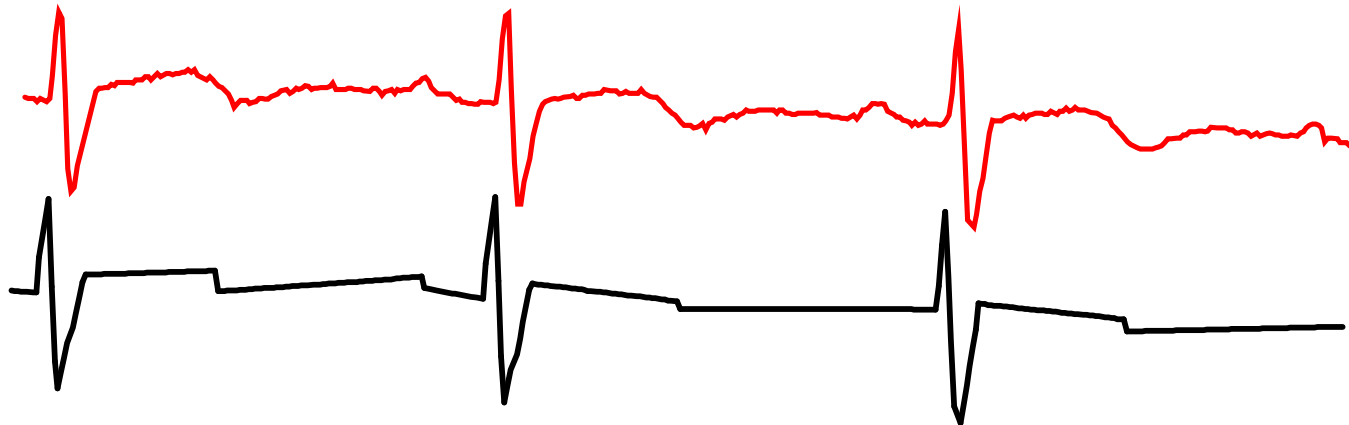
Here is the PLA with K segments:



Why Piecewise Linear Approximation?

Supports:

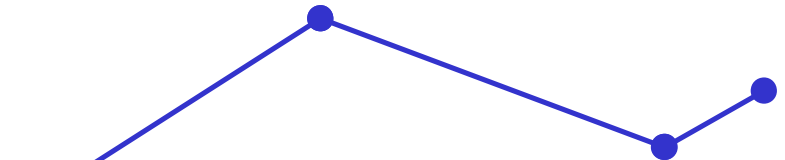
- Fast exact similarity search [13].
- Novel distance measures for time series, including “*fuzzy queries*” [27, 28].
- Weighted queries [15].
- Multi-resolution queries [31, 18], dynamic time warping [22] and relevance feedback [14].
- Concurrent mining of text and time series [17].
- Novel clustering and classification algorithms [15].
- Change point detection [29, 8].



We have to make a representational choice...

- **Linear Interpolation**

Segments are **connected**; it takes two numbers to represent a segment.



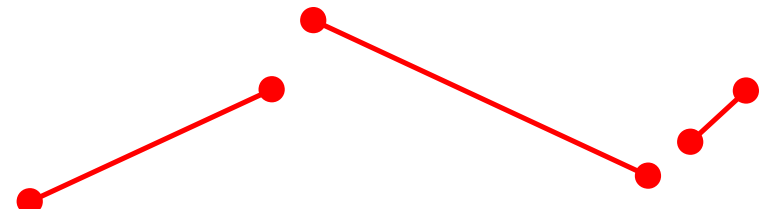
Each line segment has

- length
- left_height

(right_height can be inferred by looking at the next segment)

- **Linear Regression**

Segments are **disconnected**; it takes three numbers to represent a segment.



Each line segment has

- length
- left_height
- right_height

Obtaining the PLA

Exact algorithm (based on dynamic programming) is $O(n^2K)$ in the worst case, which is too slow for data mining with massive datasets.

Researchers have worked on faster heuristic solutions to the problem. They can be classified into the following classes:

- Top Down $O(n^2K)$
- Bottom up $O(n^2/K)$
- Sliding Window $O(n^2/K)$
- Others (Genetic Algorithms, randomized algorithms, B-spline wavelets, MDL)

Considerations we must take into account:

- Time / Space complexity
- Quality of approximation
- Batch vs. Online

Our Contribution

- Conducted the first large scale comparison of the 3 major segmentation techniques.
- Showed that the most popular algorithm, Sliding Window, produces poor quality approximations.
- In contrast, the best overall algorithm in terms of approximation accuracy, Bottom-Up, is not an online algorithm.

We introduce a new algorithm with these properties:

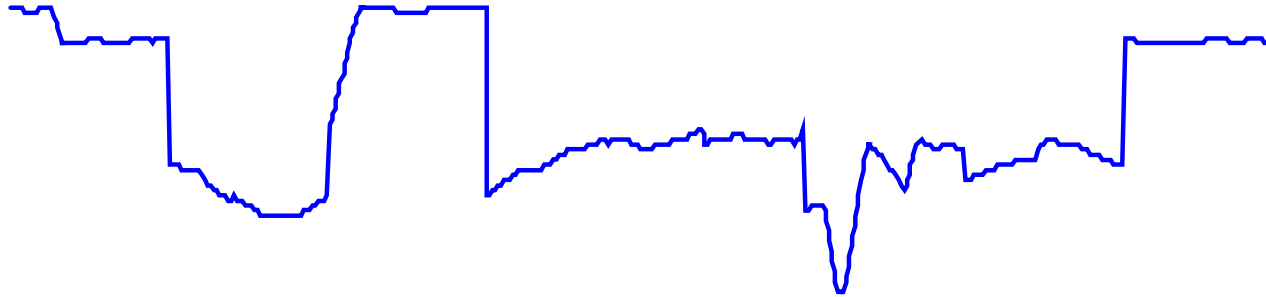
- It's **online**
- It's both **time and space efficient**
- It produces **high quality** results

We will show the three major algorithms in
their simplest forms...

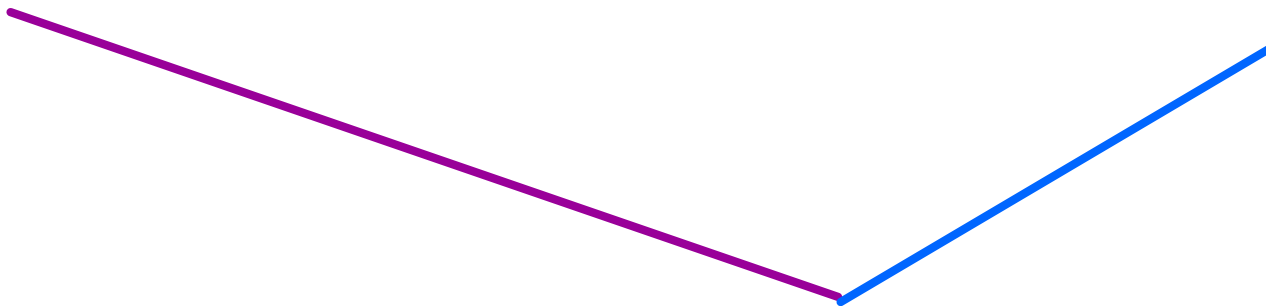
Top-Down

- **Top-Down:** The time series is recursively partitioned until some stopping condition is met.

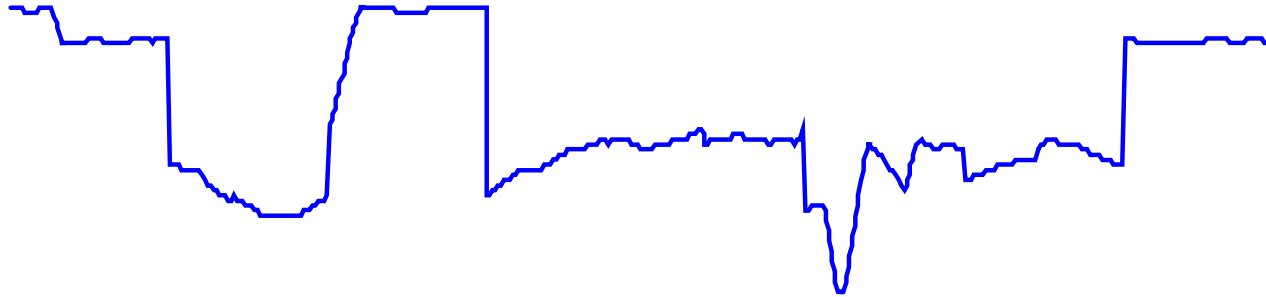
Top-Down



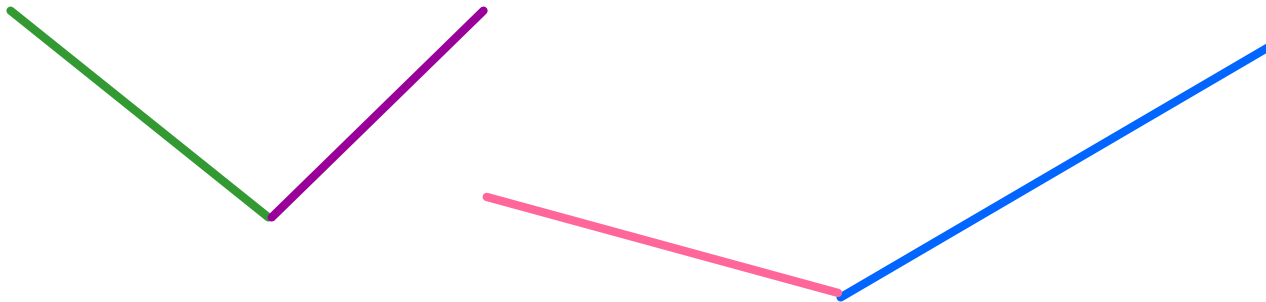
- **Considers every possible partition.**
- **Attempts to split at the best location.**



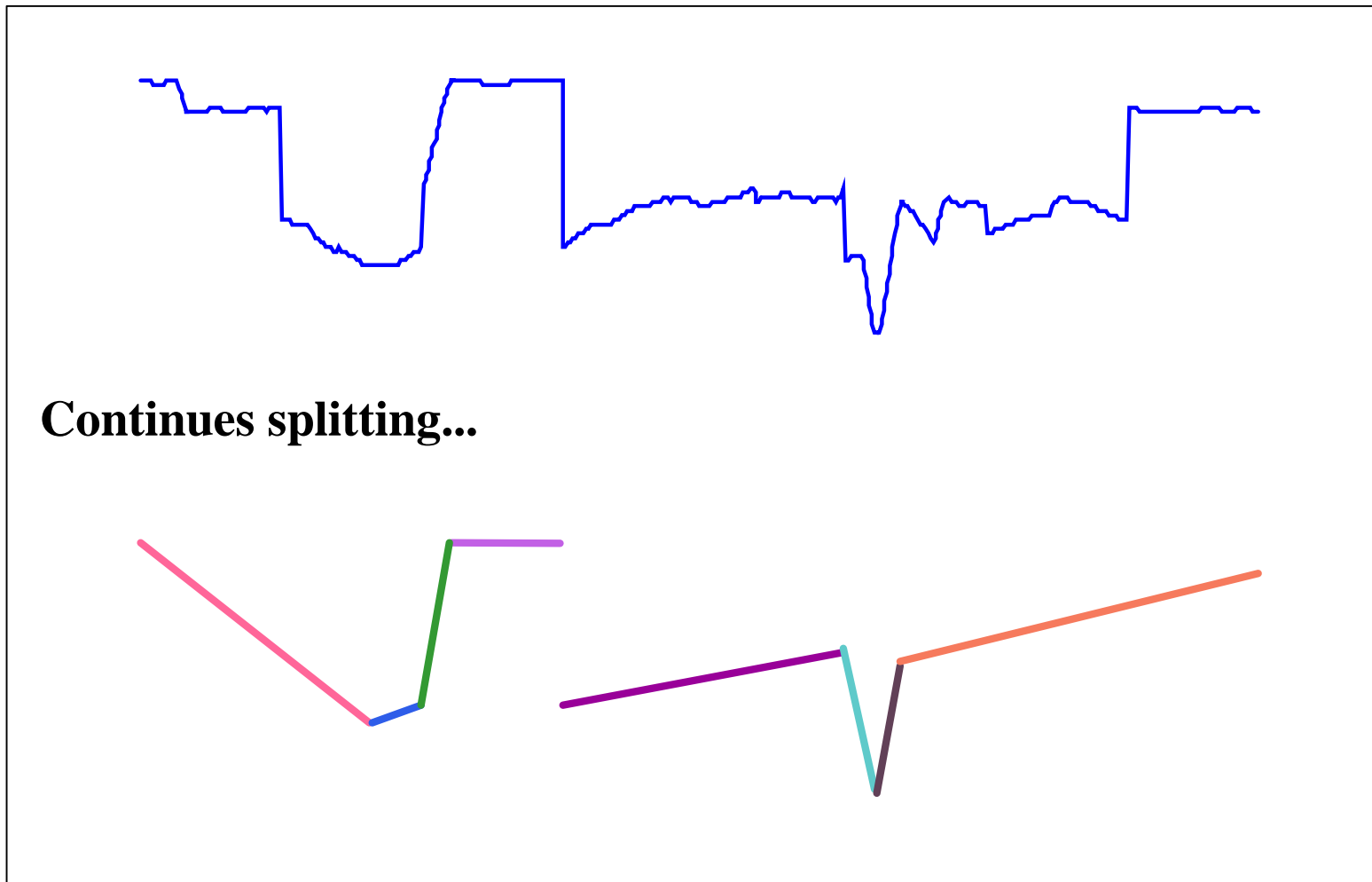
Top-Down



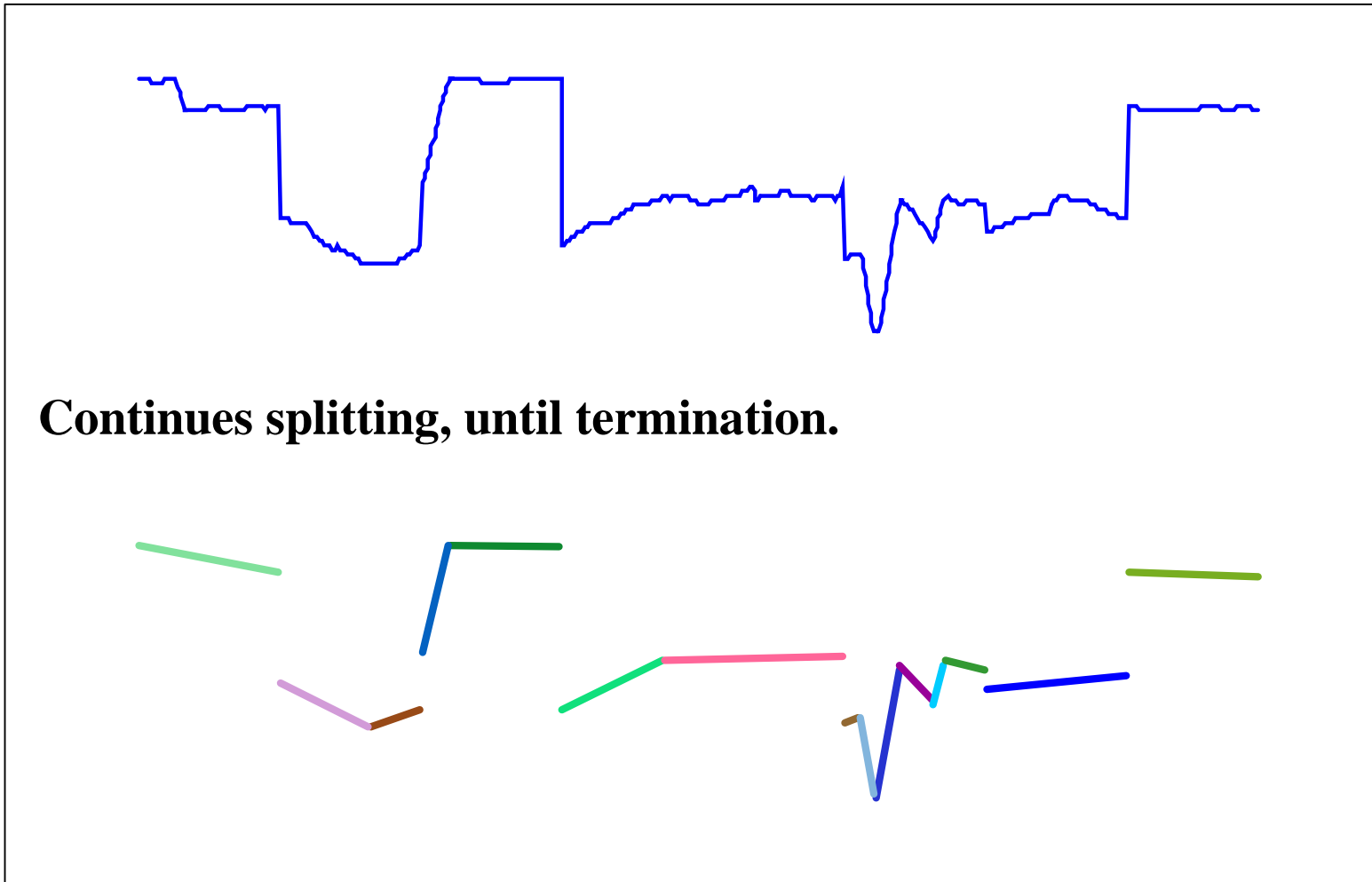
- **Continues to split until all segments have residual error less than a user-specified threshold.**



Top-Down



Top-Down

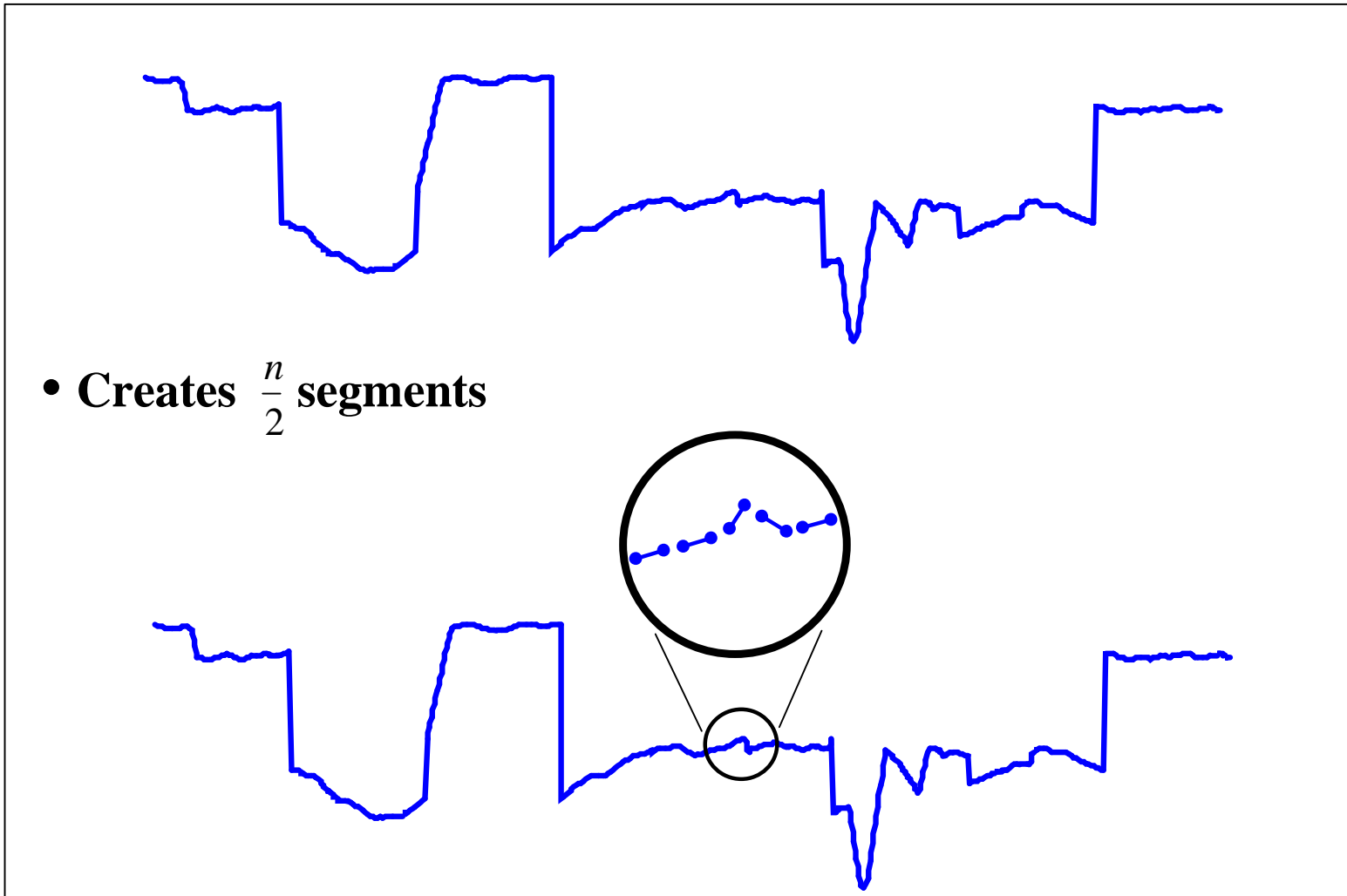


Bottom-Up

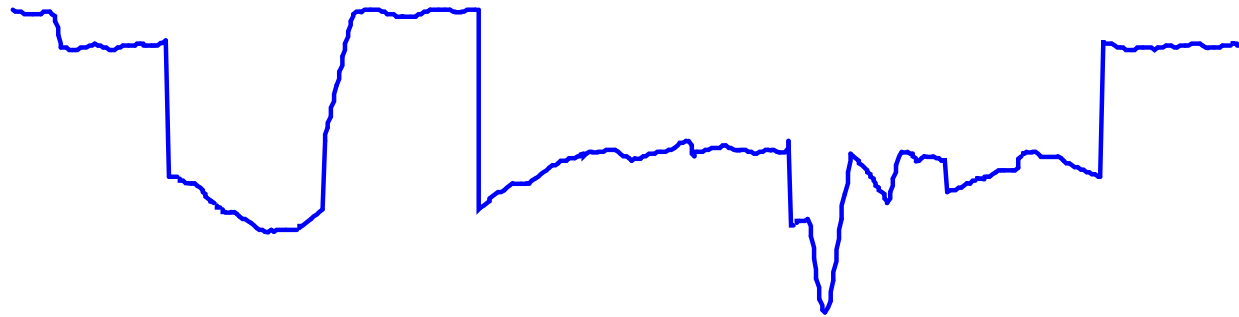
- **Bottom-Up:** Starting from the finest possible approximation, segments are merged until some stopping condition is met.

Bottom-Up

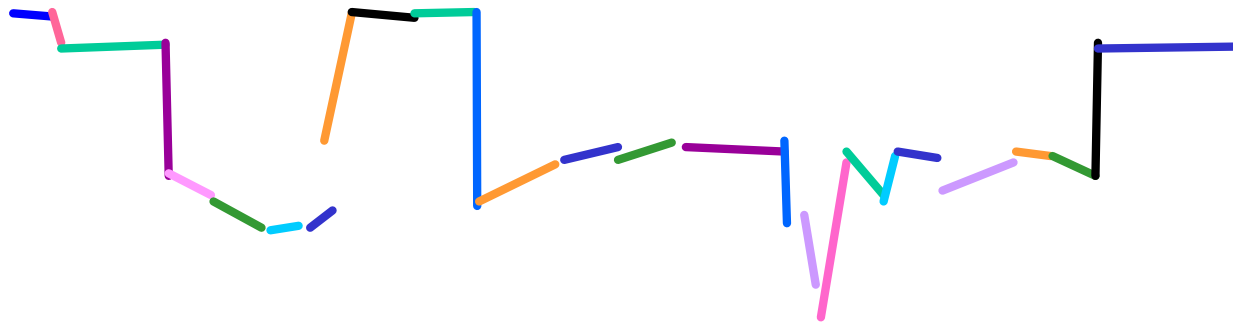
- Creates $\frac{n}{2}$ segments



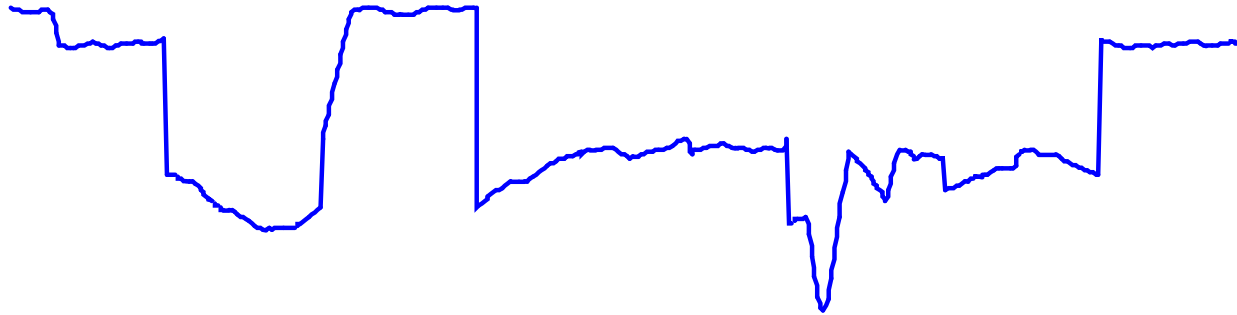
Bottom-Up



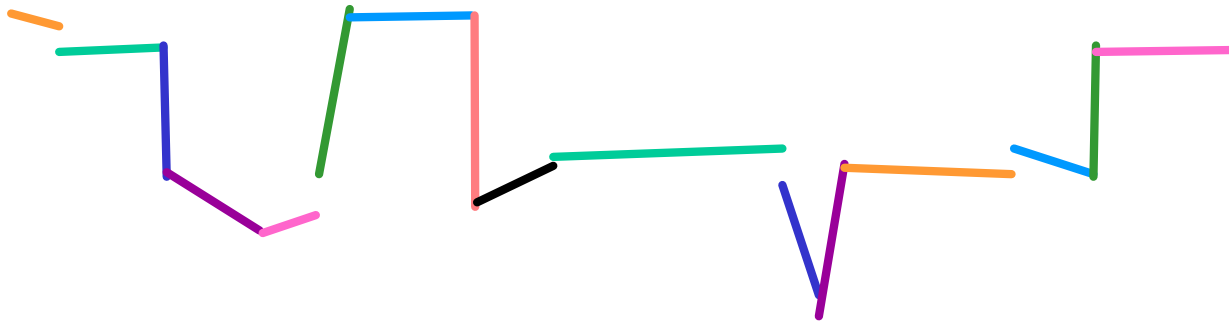
- Finds the lowest-cost pair to merge until the stopping condition is met.



Bottom-Up



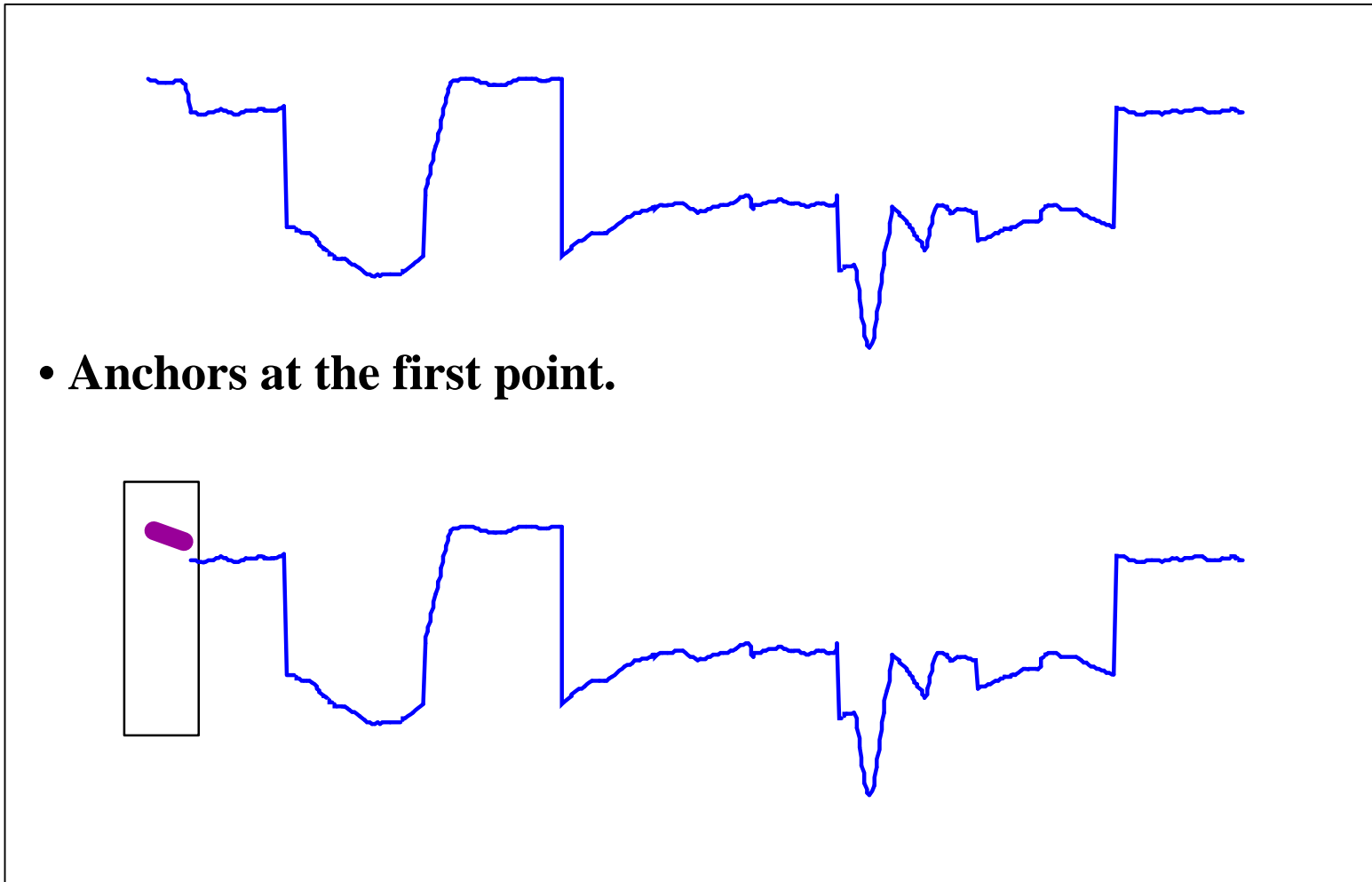
Continues merging, until termination.



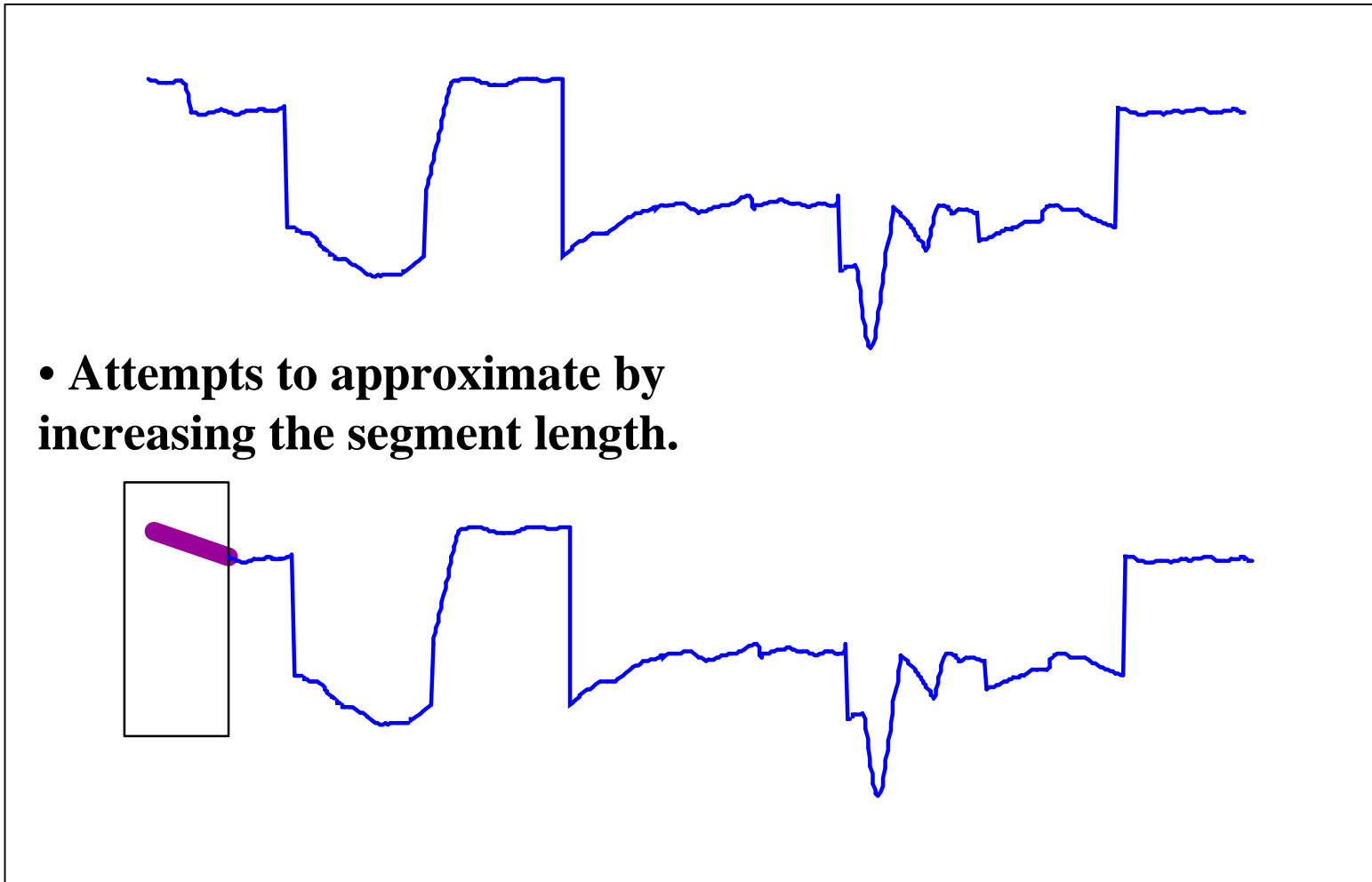
Sliding Window

- **Sliding Window:** A segment is grown until it exceeds some error bound. The process repeats with the next data point not included in the newly created segment.

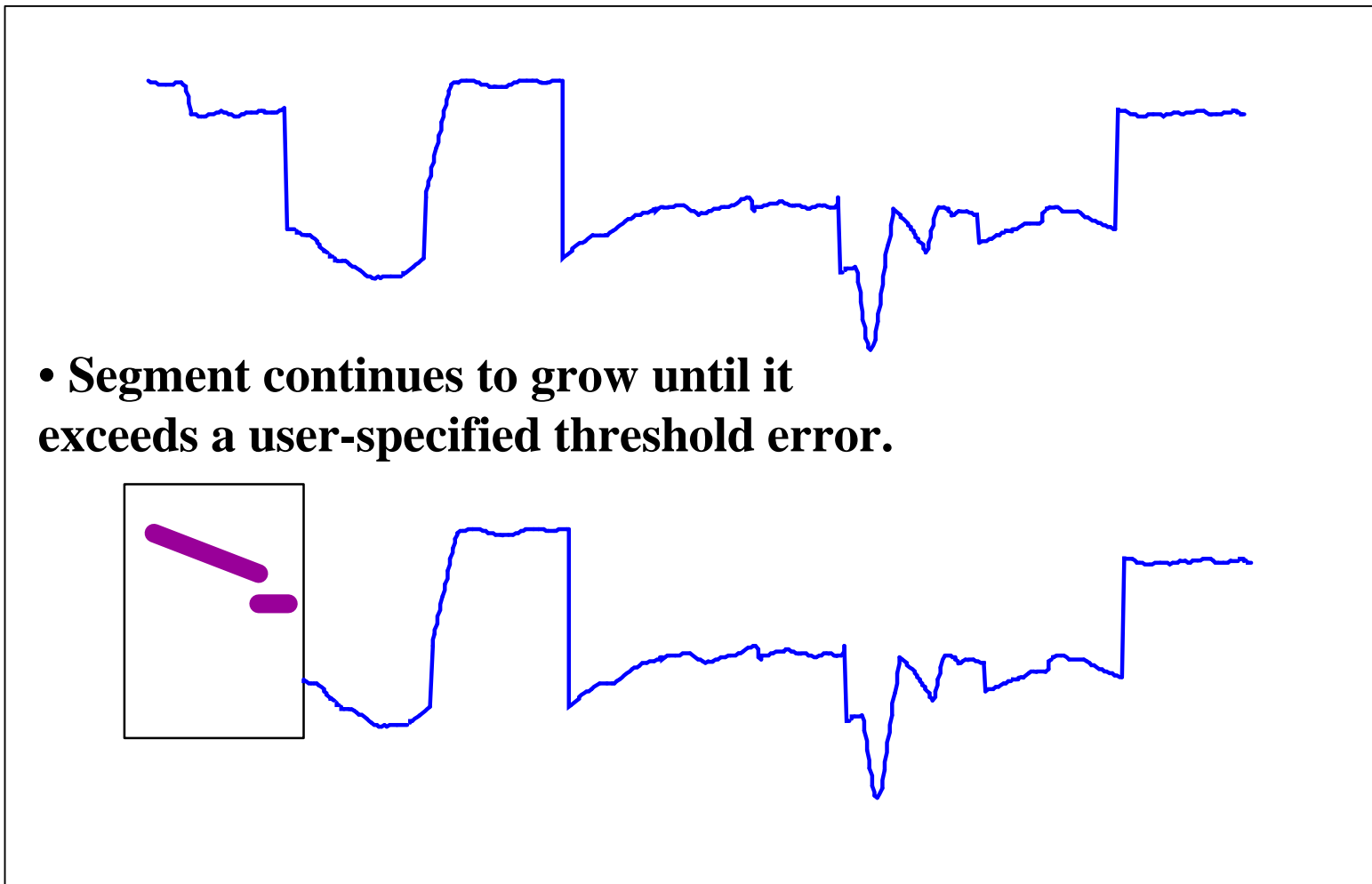
Sliding Window



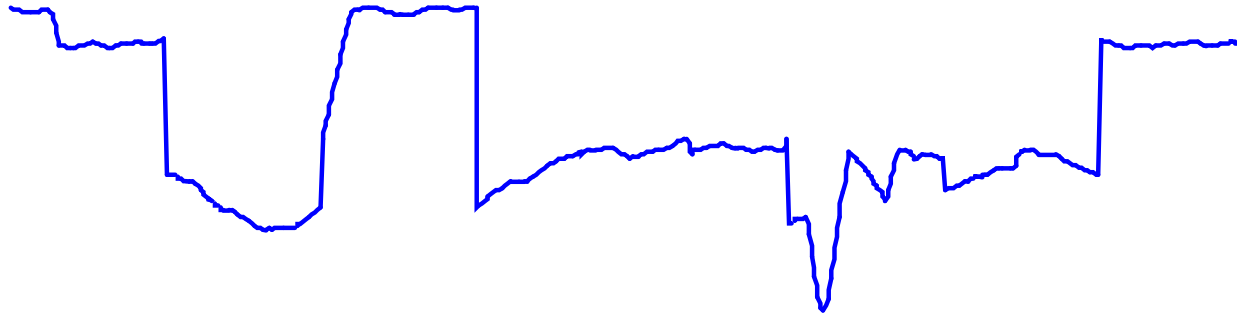
Sliding Window



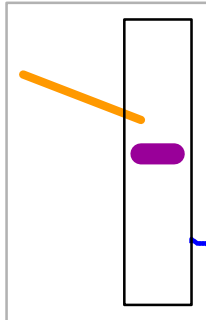
Sliding Window



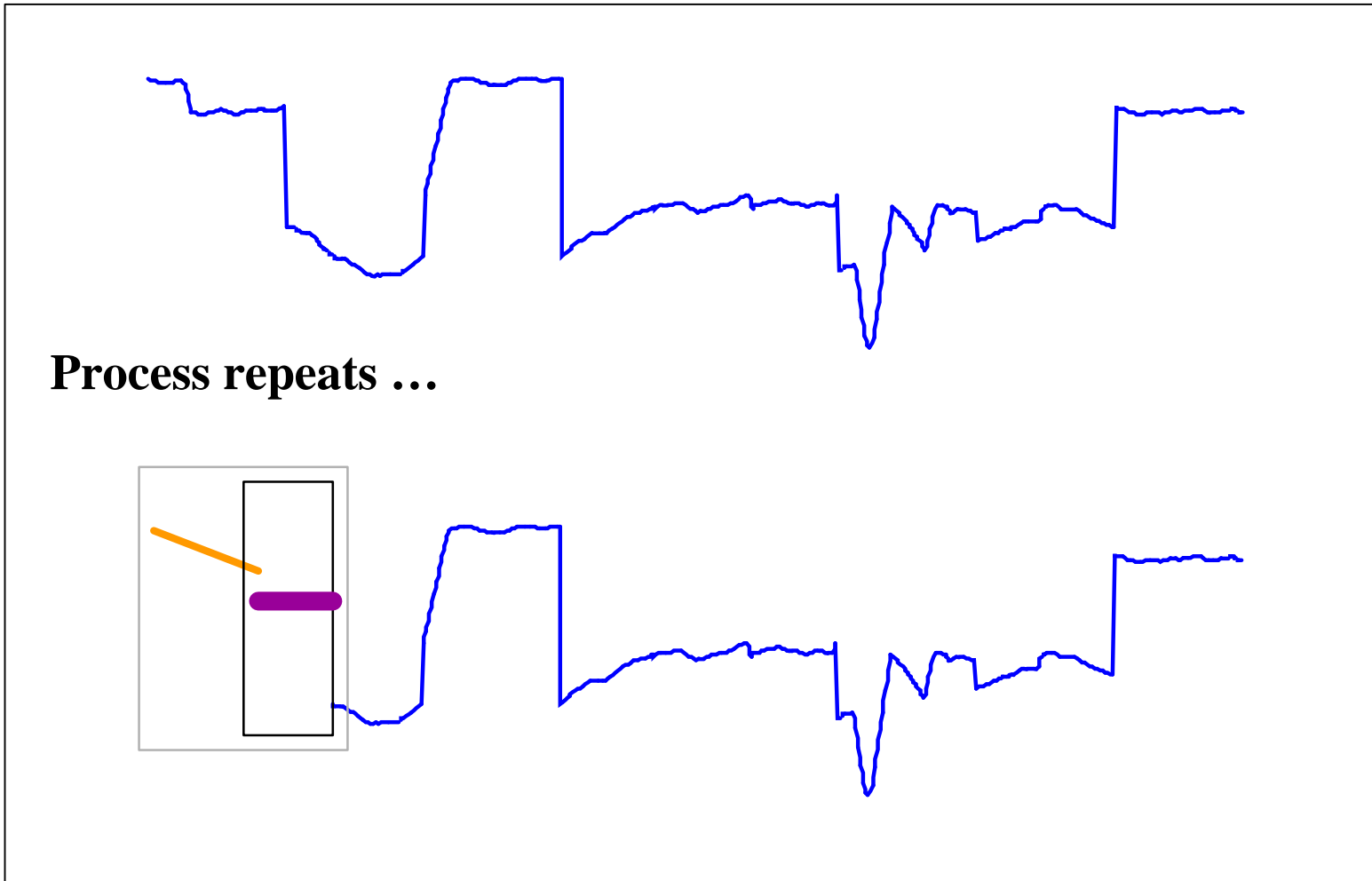
Sliding Window



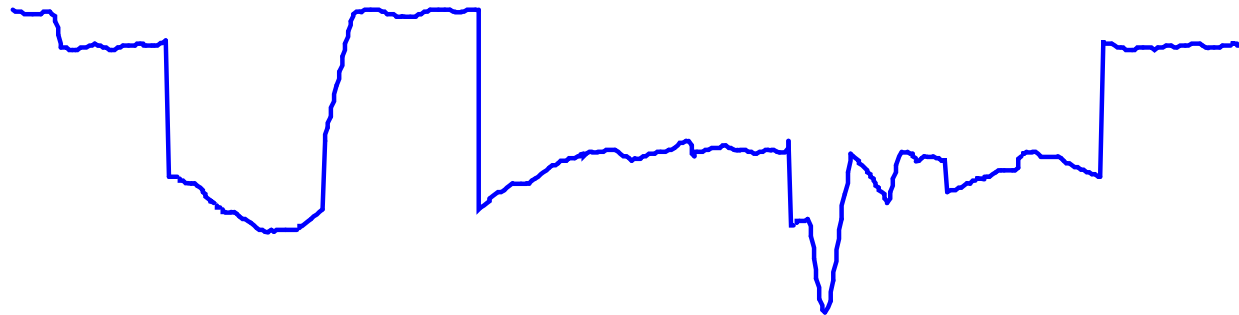
- Points already visited are made into a segment.



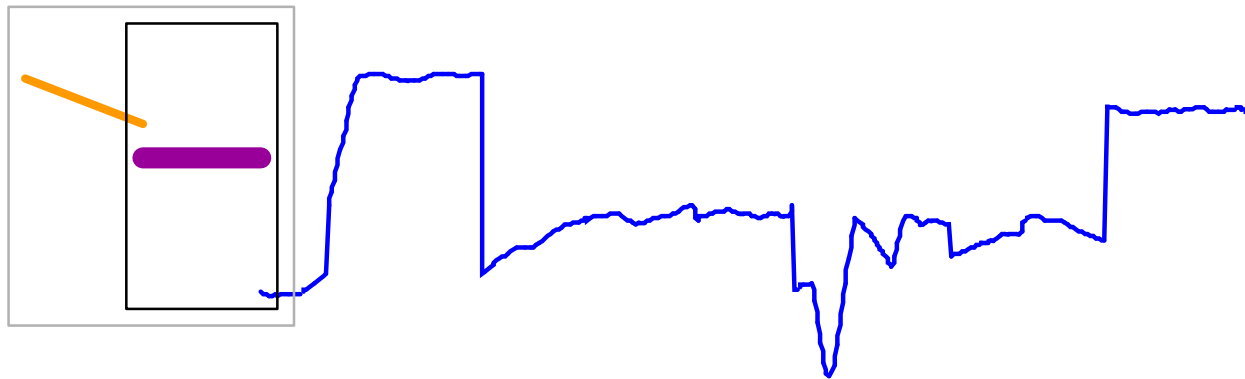
Sliding Window



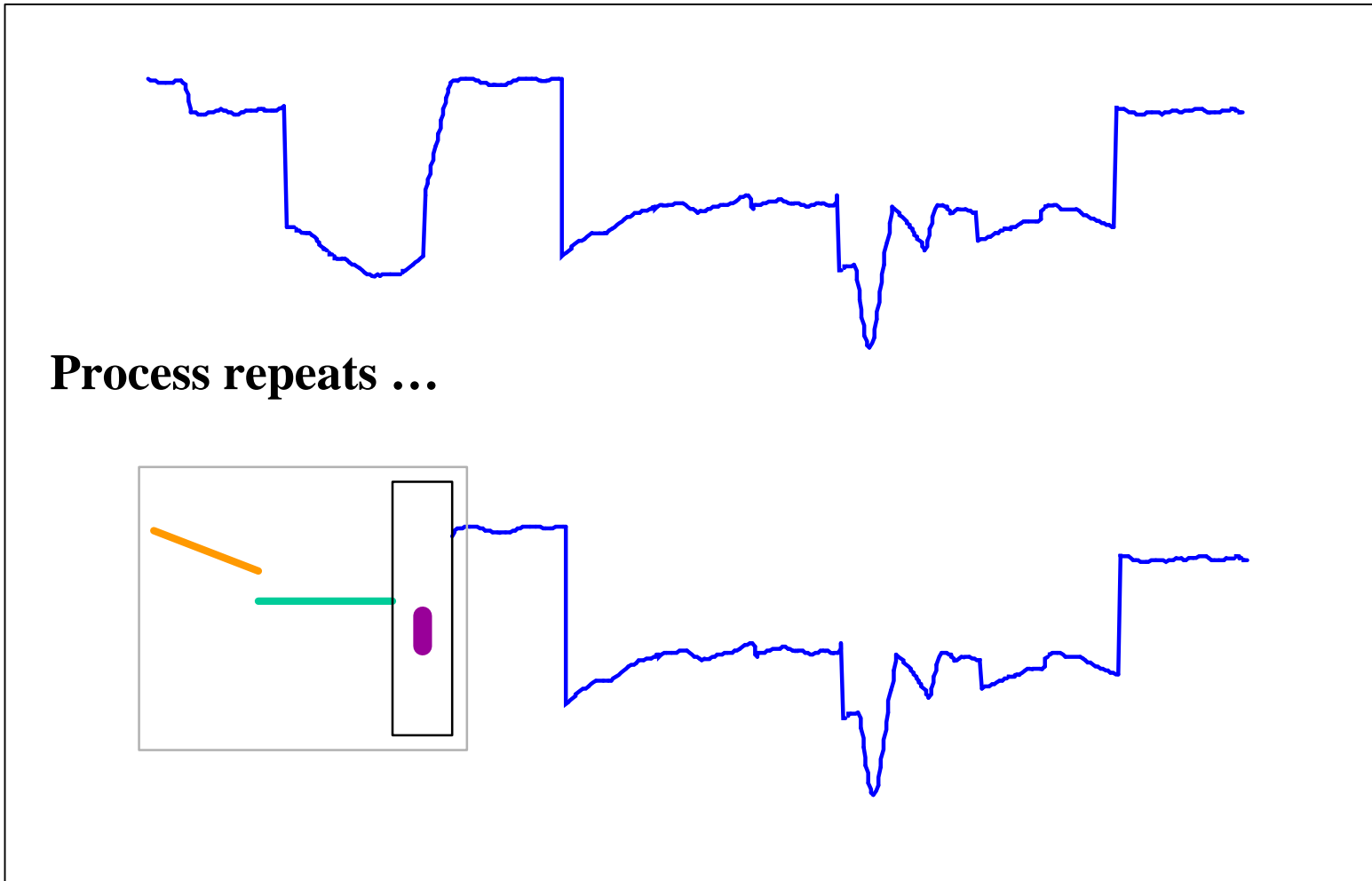
Sliding Window



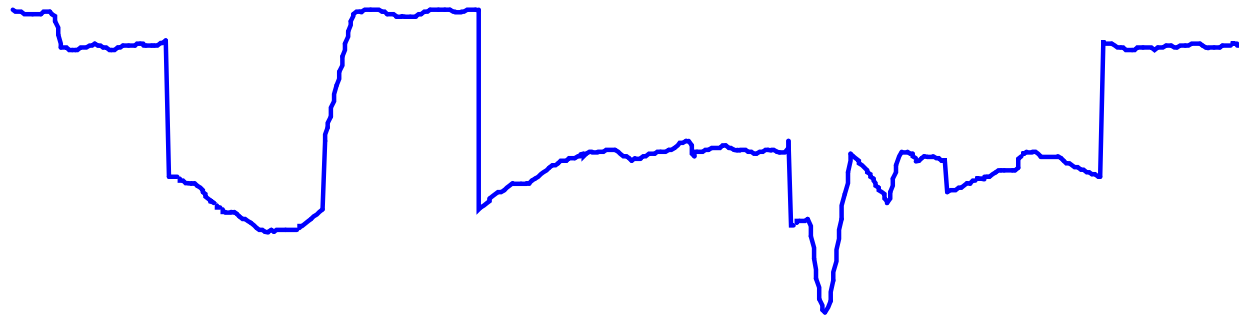
Process repeats ...



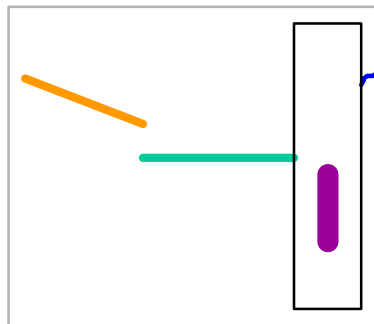
Sliding Window



Sliding Window

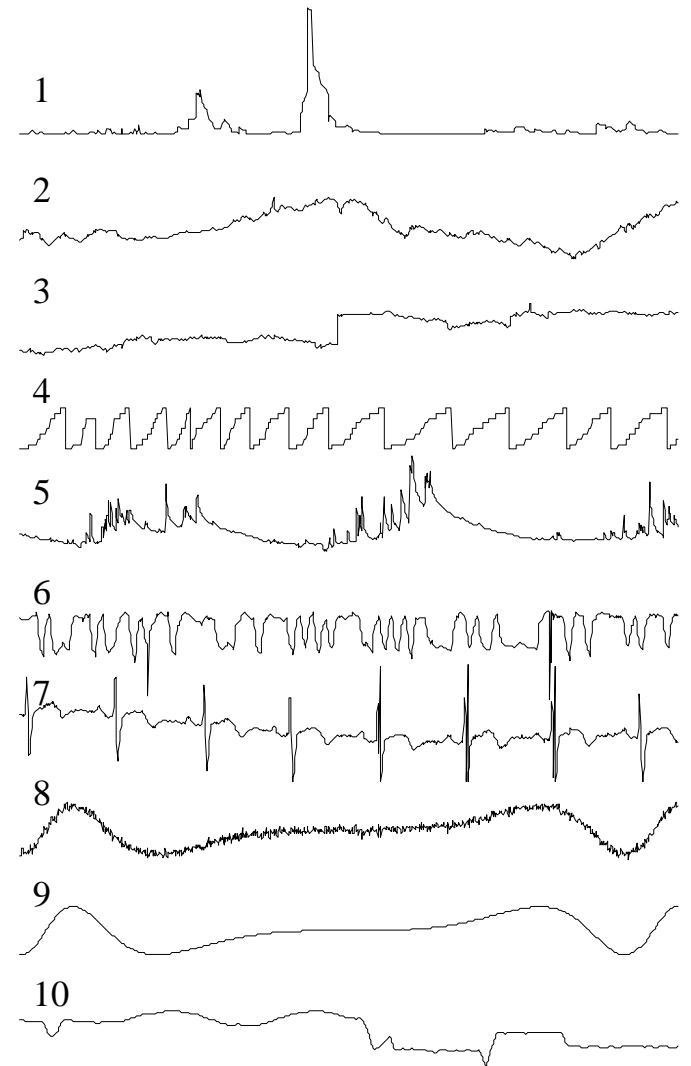


Process repeats ...



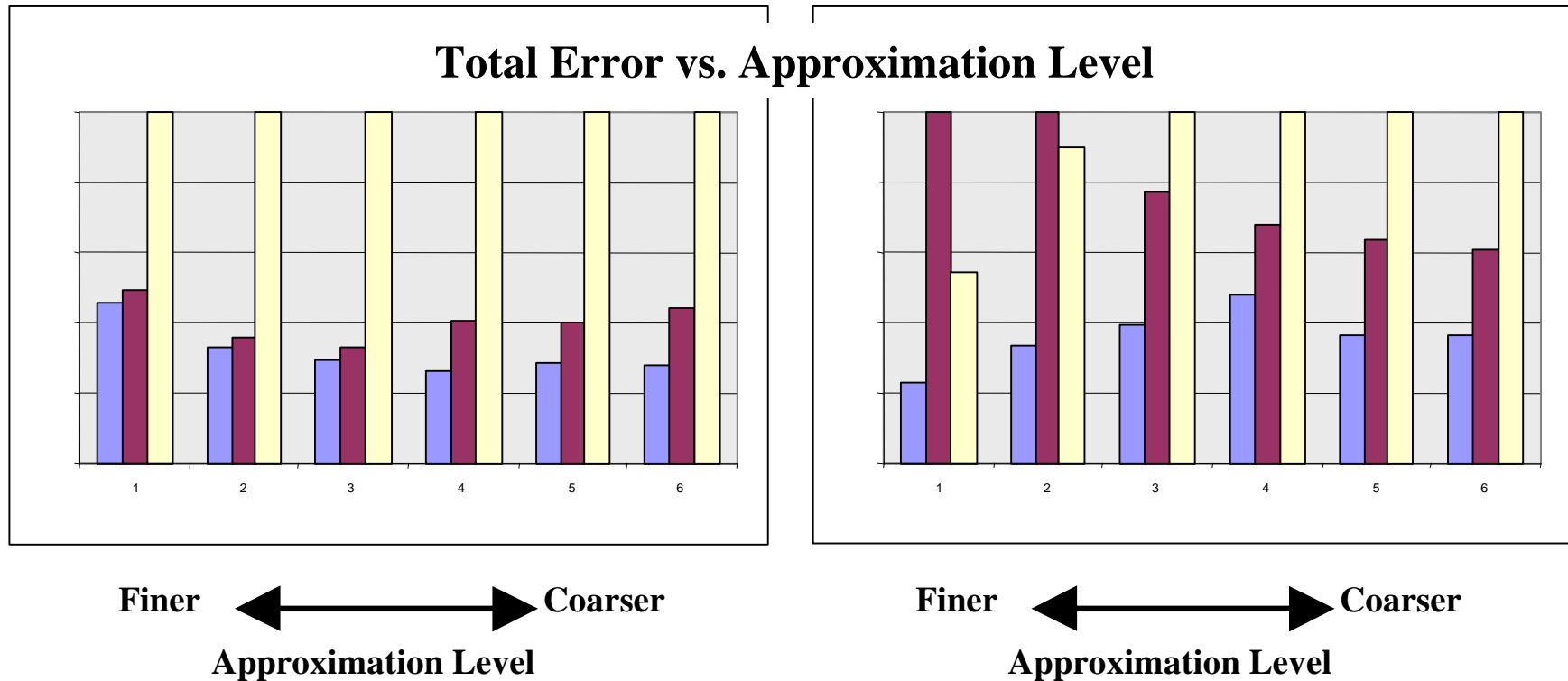
Experimental Comparison

- Used a very diverse set of time series (10 sets).
- Compared each set at six different compression levels. Since some algorithms might be better at certain compression levels.
- Measured total error of the entire approximation (Residual Error or Sum of Squared Error).



We will only show a few representative example results...

Experimental Results



Number of winners (lowest total approximation error):

- Bottom Up:** 40 out of 60 trials
- Top Down:** 20
- Sliding Window:** 0 (worst performance, 55 out of 60)

Problems with Current Techniques

- Online algorithm gives very poor approximation.
- Best algorithm is batch.

Is there an algorithm that will provide quality results while retaining the online robustness?

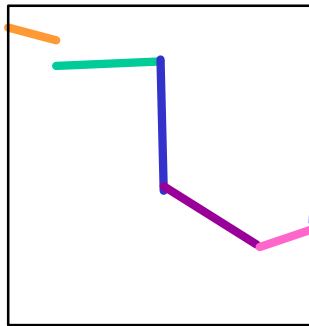
SWAB

Sliding Window And Bottom-Up

- **SWAB**: Uses a buffer to create a “semi-global” view of the dataset, allowing the usage of the Bottom Up method.

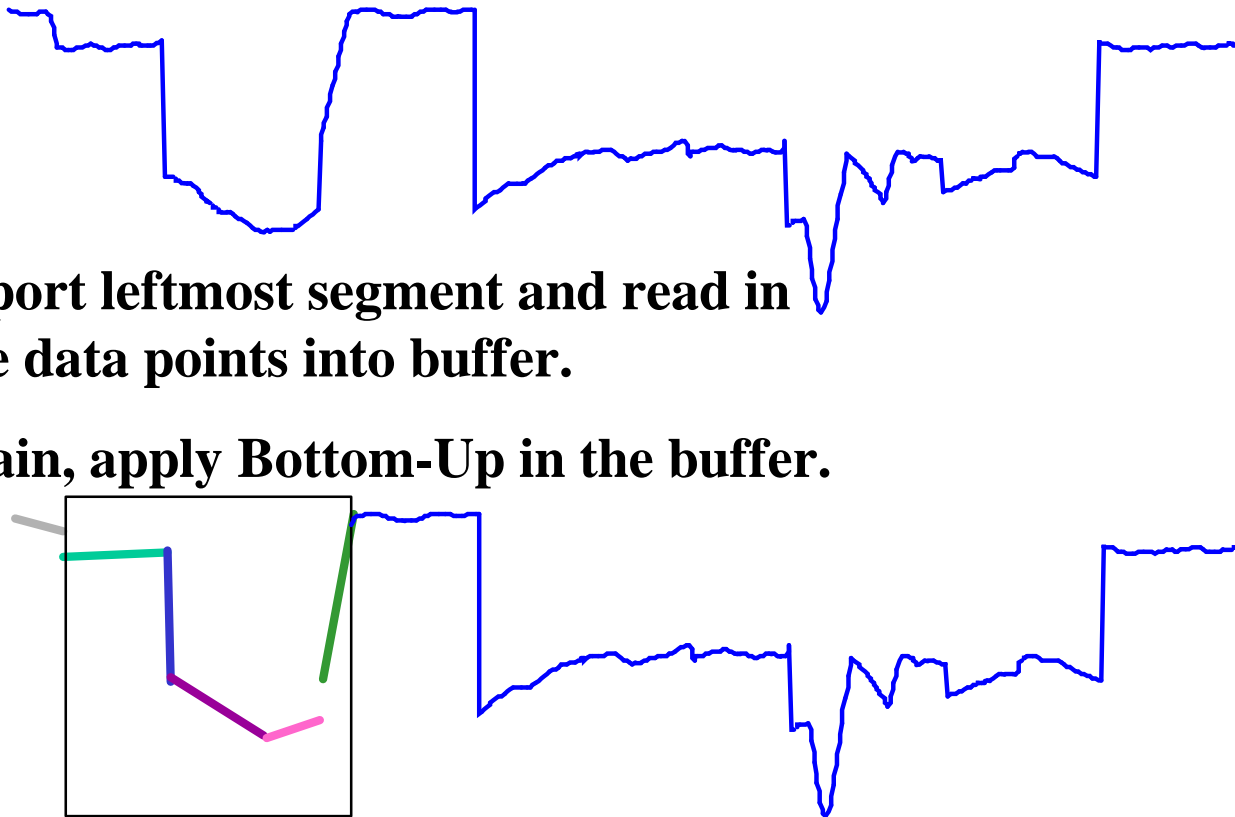
Sliding Window Bottom-Up

- **Buffer size is initialized to a small amount of data, enough to create 5 or 6 segments.**
- **Apply Bottom-Up in the buffer.**

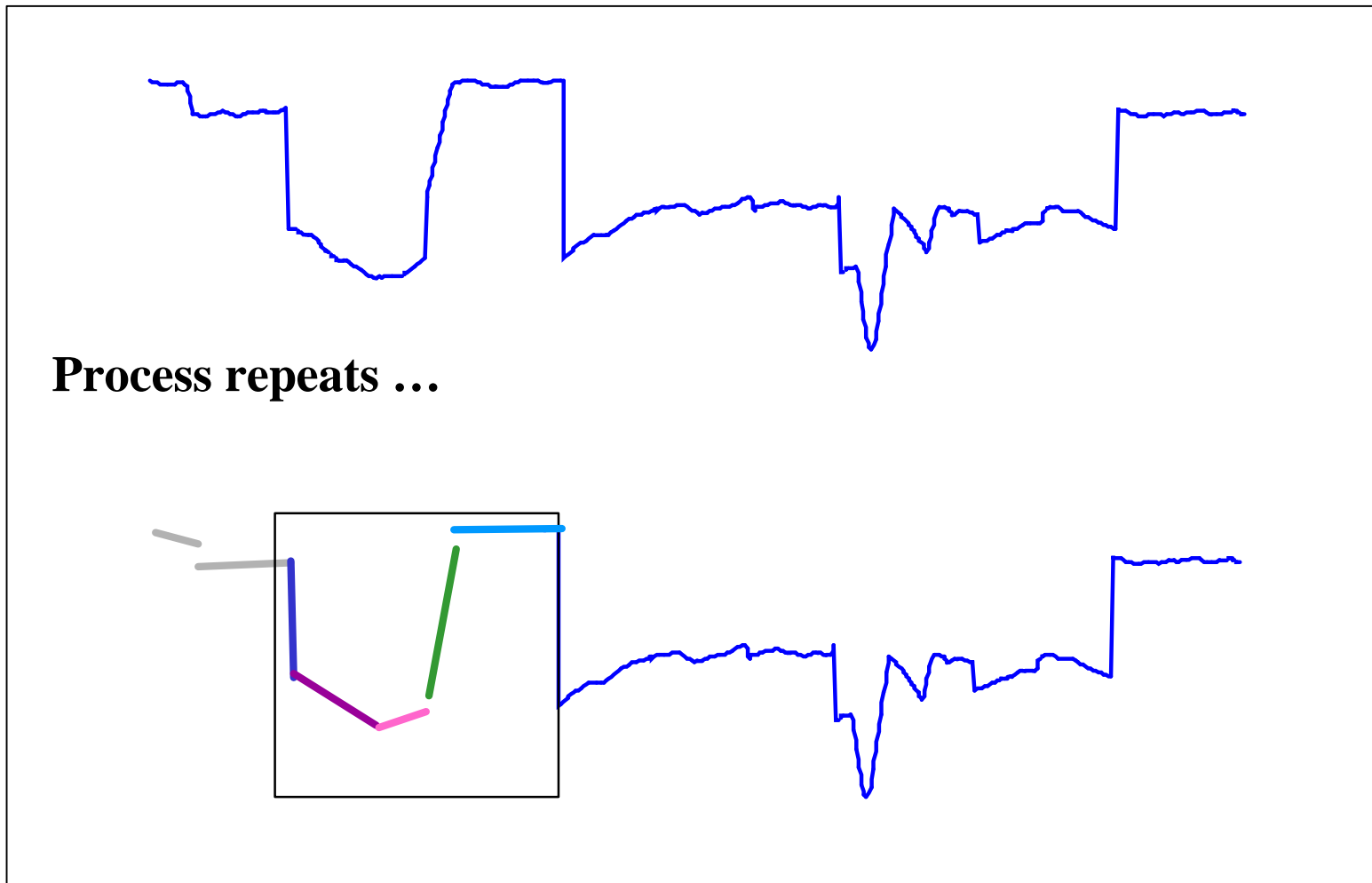


Sliding Window Bottom-Up

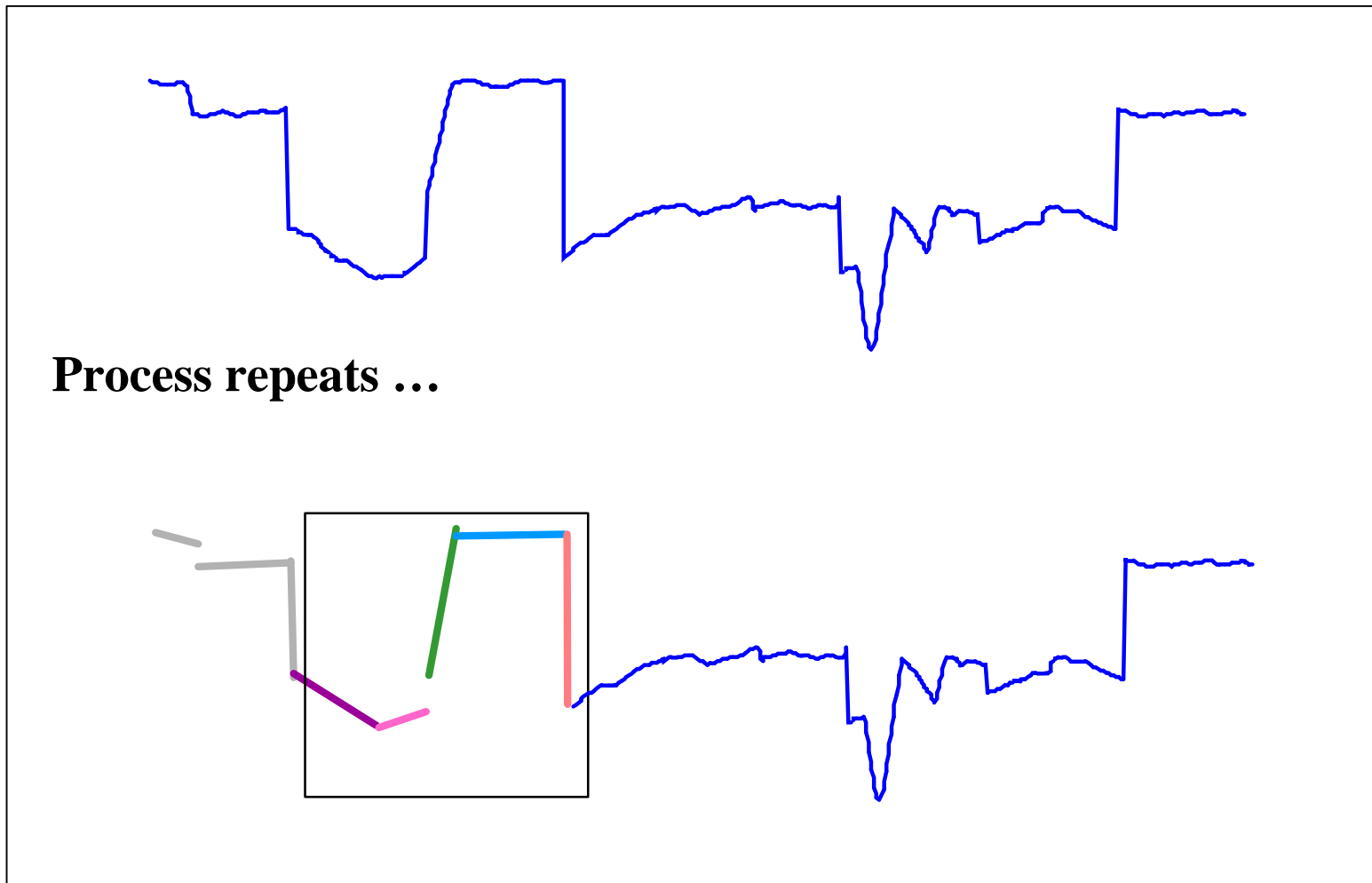
- Report leftmost segment and read in more data points into buffer.
- Again, apply Bottom-Up in the buffer.



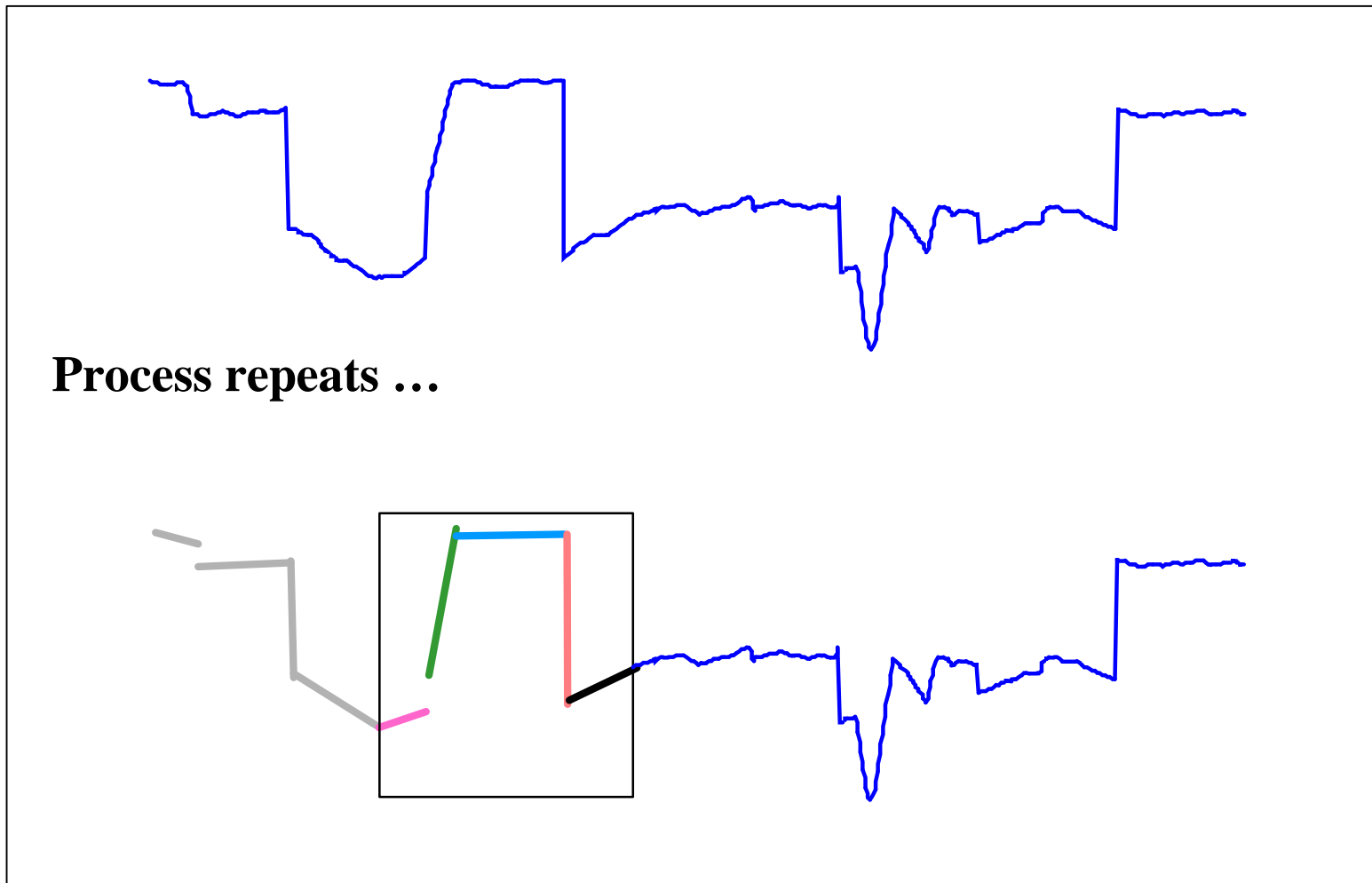
Sliding Window Bottom-Up



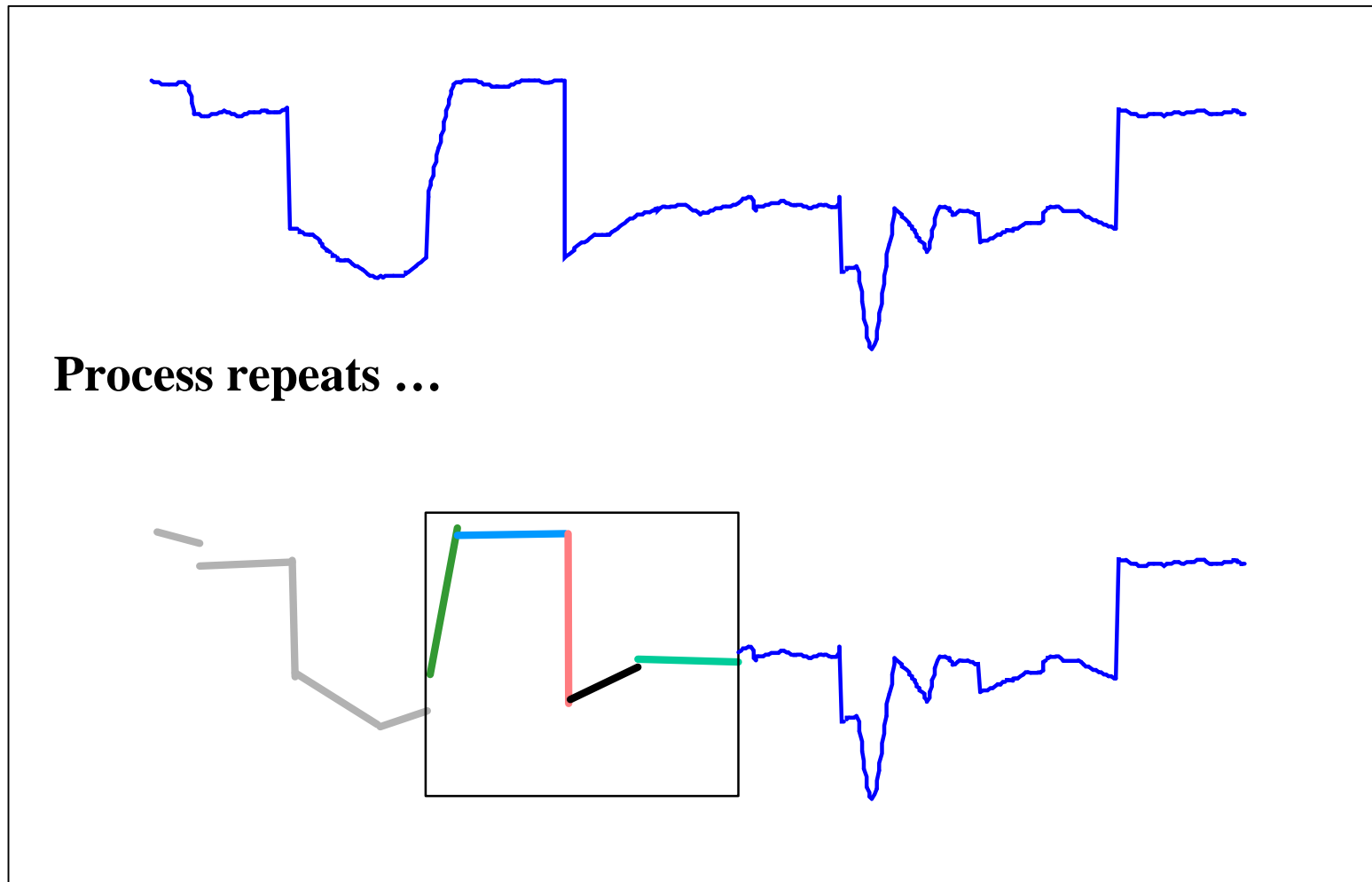
Sliding Window Bottom-Up



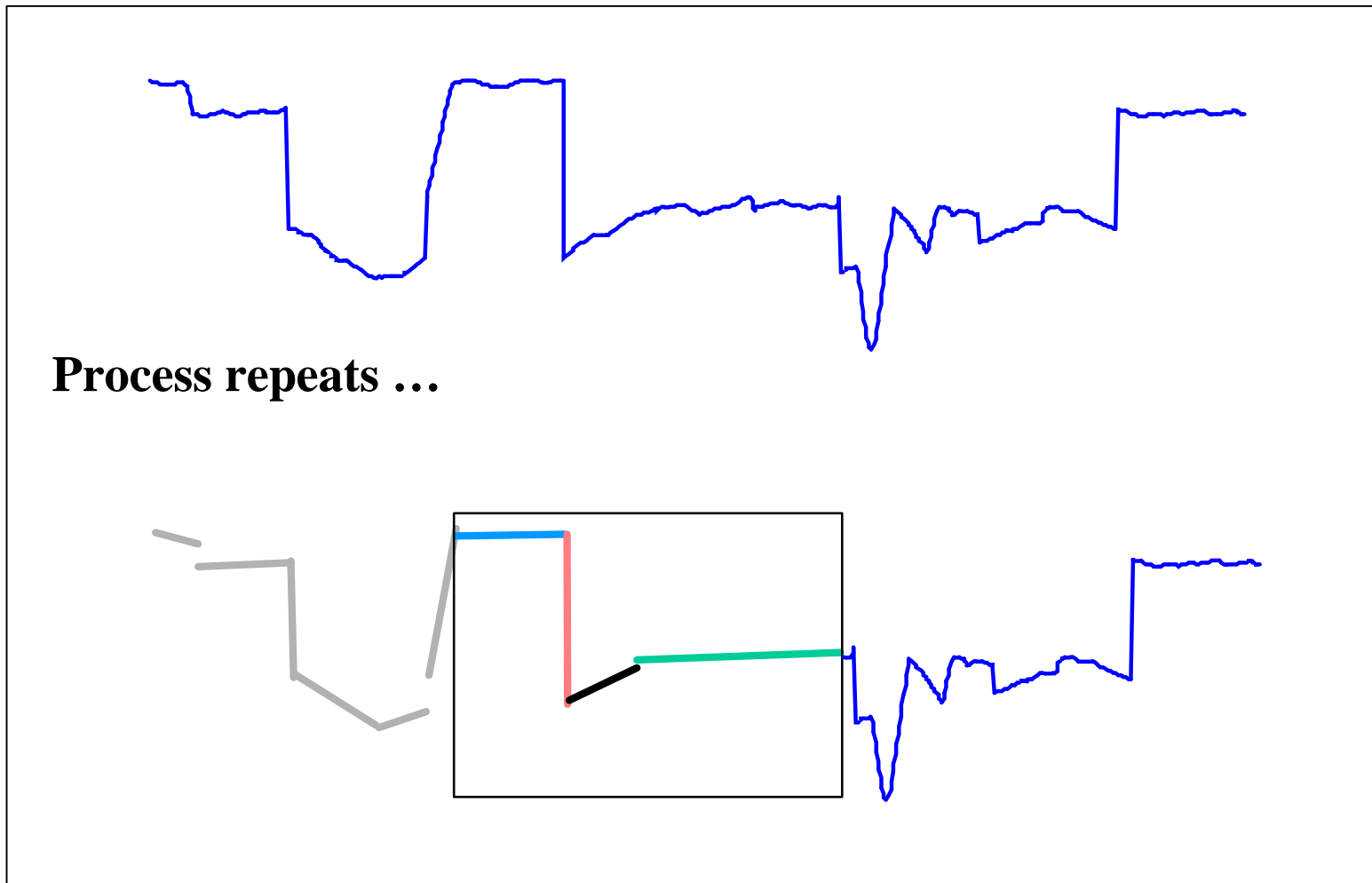
Sliding Window Bottom-Up



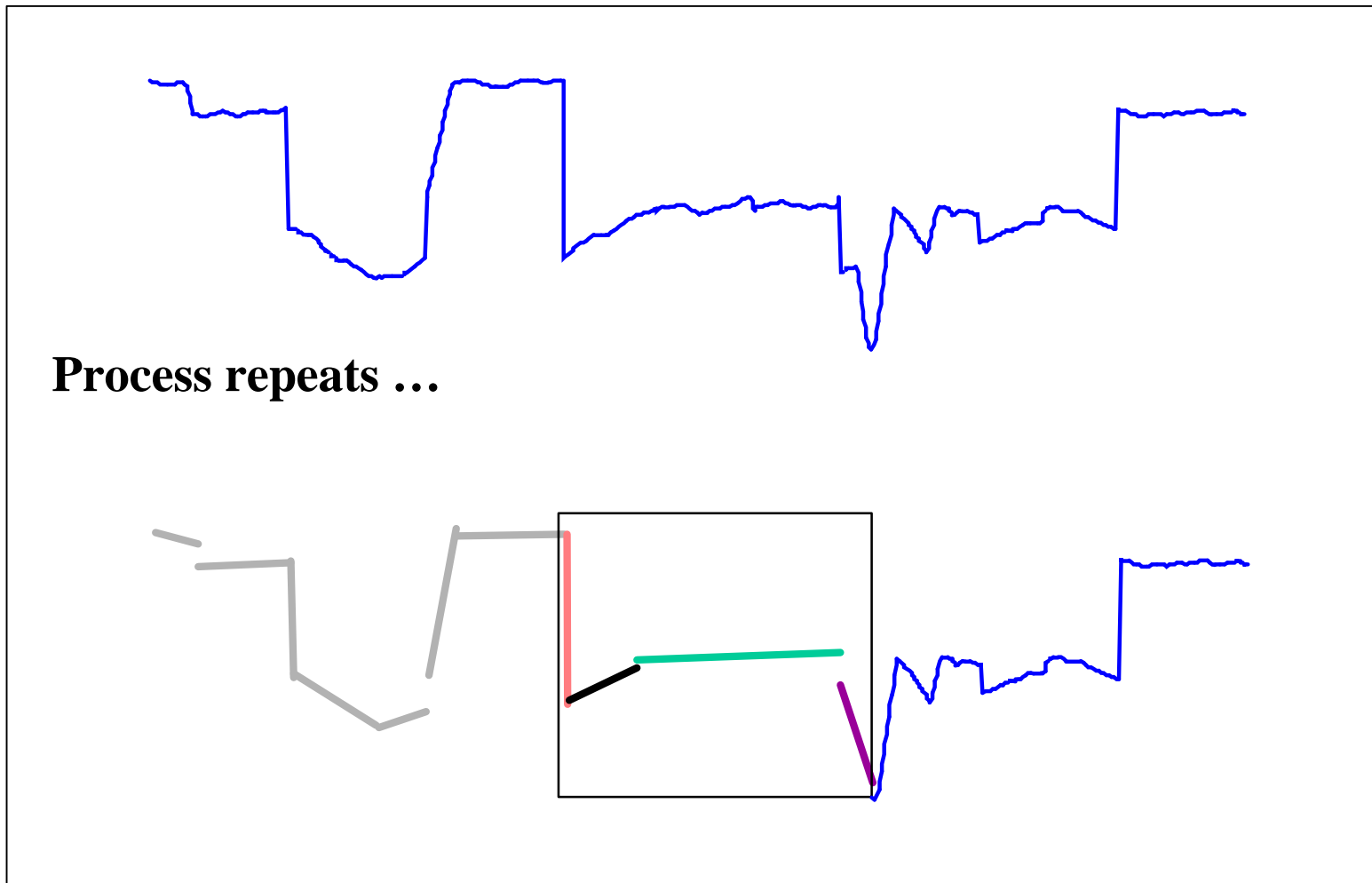
Sliding Window Bottom-Up



Sliding Window Bottom-Up



Sliding Window Bottom-Up



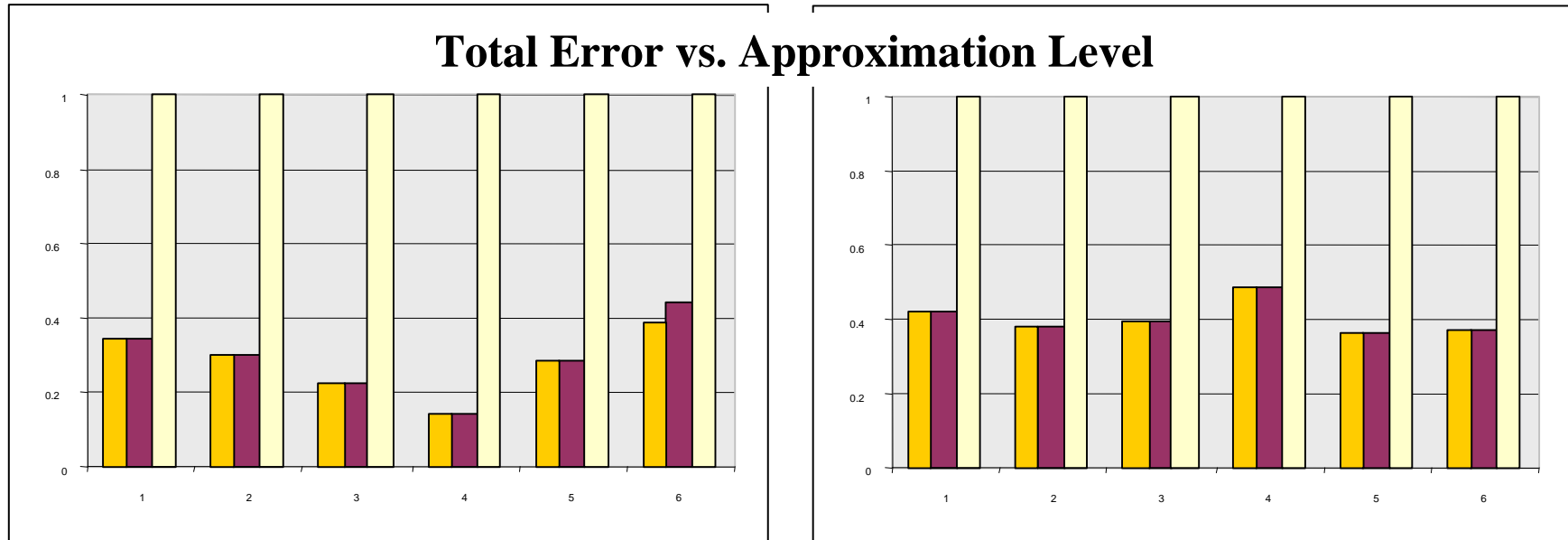
Intuition behind SWAB

- We can view **SWAB** as acting on a continuum between **Sliding Window** and **Bottom-up**.
- As buffer size $\rightarrow 1$, emulates **Sliding Window**
 $\rightarrow \infty$, emulates **Bottom-Up**

The surprising result is...

Once the buffer is initialized to a small reasonable size length, (about 5 segments), the *quality of approximation* returned by **SWAB** is essentially the *same* as **Bottom-up**.

Experimental Results



Finer ← → Coarser
Approximation Level

Finer ← → Coarser
Approximation Level



SWAB has essentially the same performance as batch **Bottom-Up**.

Conclusions

We conducted the first large scale empirical comparison of the 3 major time series segmentation techniques.

We introduce **SWAB**, a new **time and space efficient** segmentation algorithm, which is **online** and produces **high quality** results.

Future Work

- Extensions to 2D and 3D time series.
- Remove some redundant calculations from SWAB.