



Ensemble-Index: A New Approach to Indexing Large Databases

Eamonn Keogh *University of California, Riverside*

Selina Chu *University of California, Irvine*

Michael Pazzani *University of California, Irvine*

{eamonn,selina,pazzani}@ics.uci.edu

Some copies of these slides are available at the back of the room

Outline of Talk

- The utility of similarity search for data mining.
- The GEMINI Framework
 - *Which dimensionality reduction technique is best?*
- The E-Index
- A worked example.
- Experiment results
- Conclusions/future directions

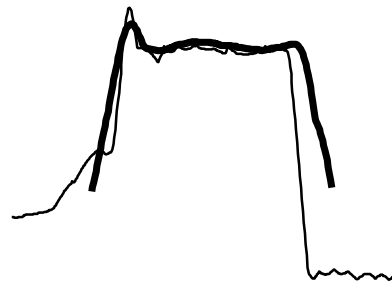
Although the indexing technique introduced in this paper is very general, allowing indexing of histograms, images etc. For concreteness and brevity we only consider time series in this talk.

What is Similarity Search?

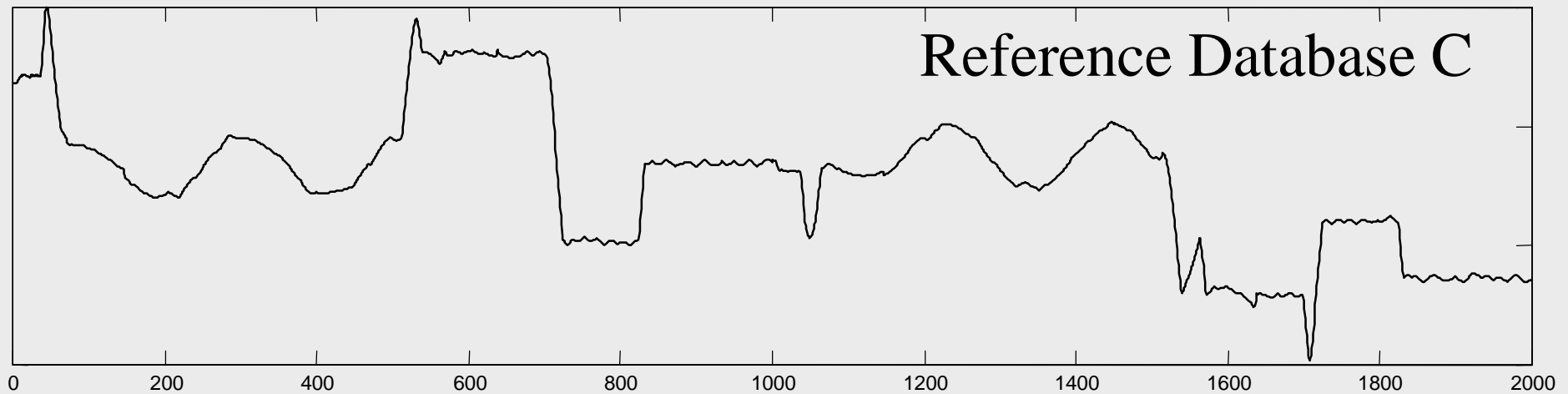
(In the Context of Time Series)



Query Q
(template)



Given a Query Q , a reference database C and a distance measure, find the subsection in C that best matches Q and return its position.



The Utility of Similarity Search?

(In the context of Time Series)

- **Classification:** *Do other genes express themselves like this gene?*

Aach, J and Church, GM (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics* 17:495-508

- **Clustering:** *Grouping robot experiences.*

Oates, Tim; Schmill, Matthew D. and Cohen, Paul R. A Method for Clustering the Experiences of a Mobile Robot that Accords with Human Judgments. In *AAAI 2000*.

- **Association Rules:** *Peak followed plateau implies a downward trend with a confidence of 0.4 and a support of 0.2.*

Das, et al. (1998). Rule discovery from time series.

- **Exploratory Data Analysis:** *Understanding the data by interacting with it.*

Wijk, J.J. van, E. van Selow.(1999). Cluster and Calendar-based Visualization of Time Series Data. *IEEE InfoVis'99*.

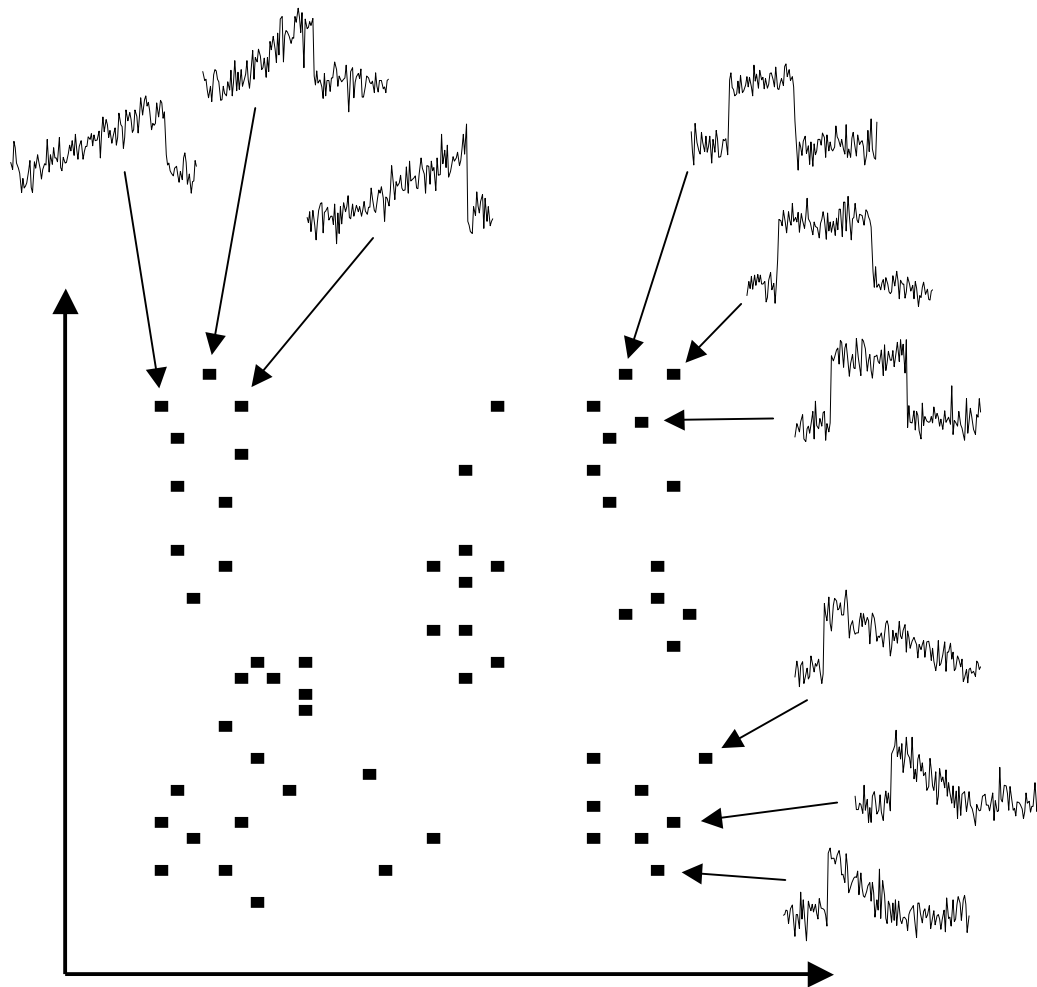
Why is data mining time series so difficult?

How do we search very large databases?

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

Since most of the data lives on disk (or tape), we need to avoid reading all the data (sequential scanning).

A general solution to similarly search is to project the data in to n -dimensional space...

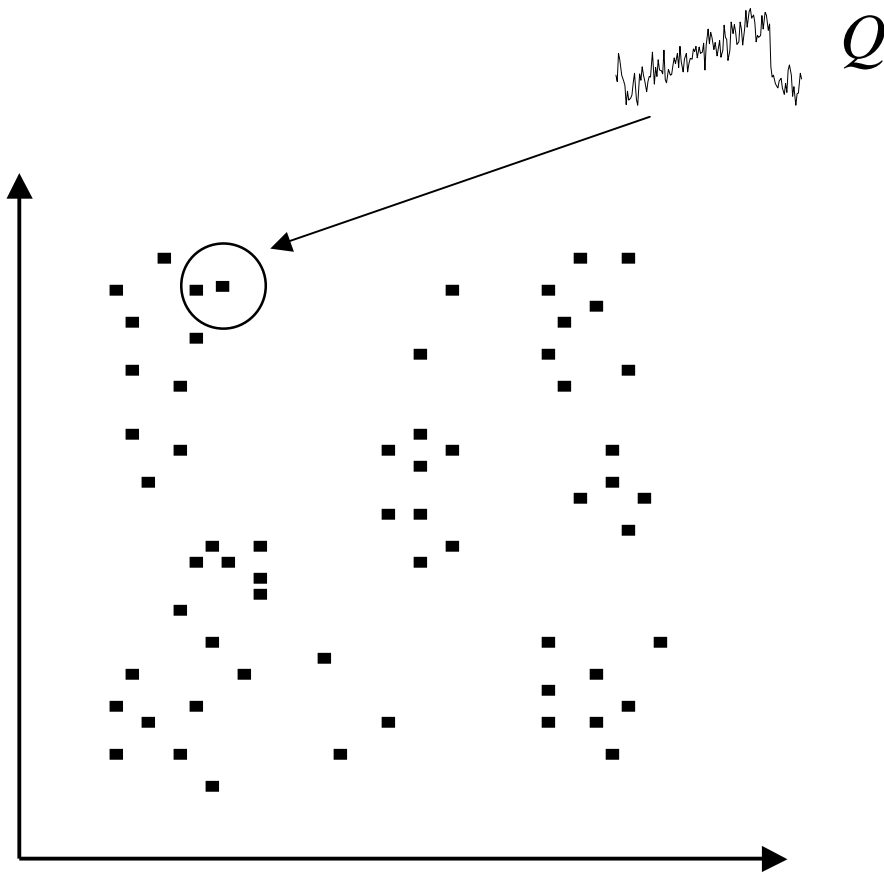


We can project time series of length n into n -dimensional space.

The first value in C is the X-axis, the second value in C is the Y-axis etc.

One advantage of doing this is that we have abstracted away the details of “time series”, now all query processing can be imagined as finding points in space...

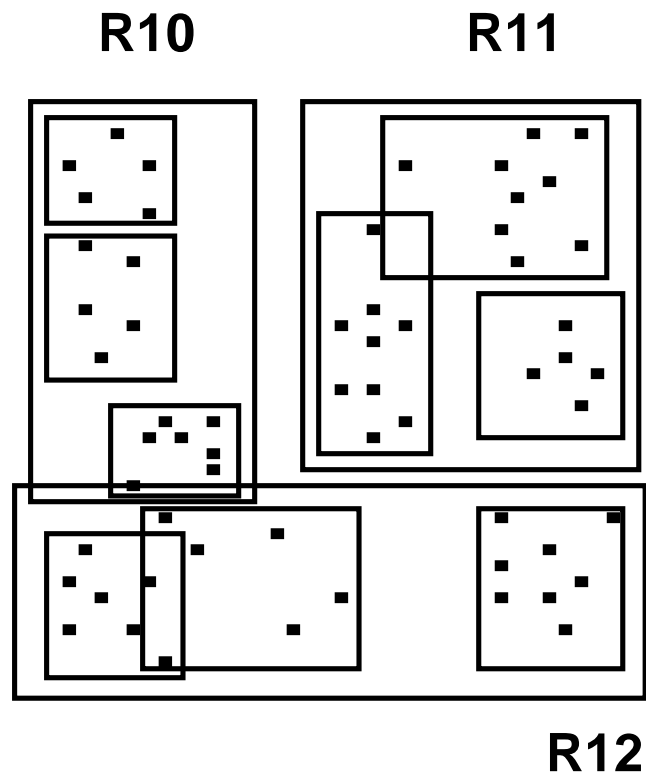
...we can project the query time series Q into the same n -dimension space and simply look for the nearest points.



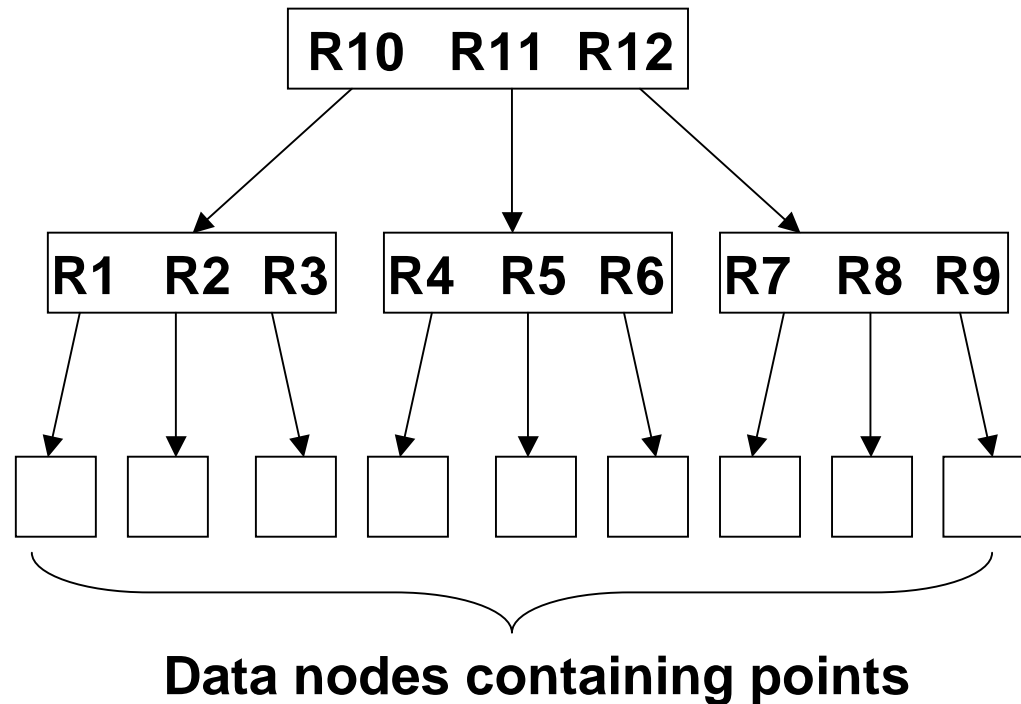
The utility of this approach leverages off the fact that we can organize the points in n -dimensional space by using a multidimensional structure such as an R-Tree...

Spatial Access Methods I

True Euclidean space is divided into regions...



...those regions are stored in a tree structure..



Spatial Access Methods II

We can use Spatial Access Methods like the R-Tree to index our data, but...

The performance of R-Trees degrades exponentially with the number of dimensions. Somewhere above 6-20 dimensions the R-Tree degrades to linear scanning.

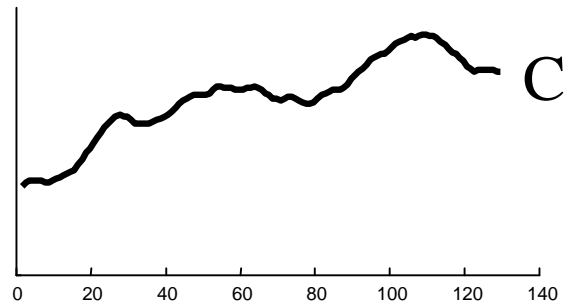
Often we want to index time series with hundreds, perhaps even thousands of features....

GEMINI GEneric Multimedia INdexIng

{Christos Faloutsos}

- Establish a distance metric from a domain expert.
- Produce a dimensionality reduction technique that reduces the dimensionality of the data from n to N , where N can be efficiently handled by your favorite SAM.
- Produce a distance measure defined on the N dimensional representation of the data, and prove that it obeys $D_{\text{indexspace}}(A,B) \leq D_{\text{true}}(A,B)$.
i.e. The lower bounding lemma.
- Plug into an off-the-shelf SAM.

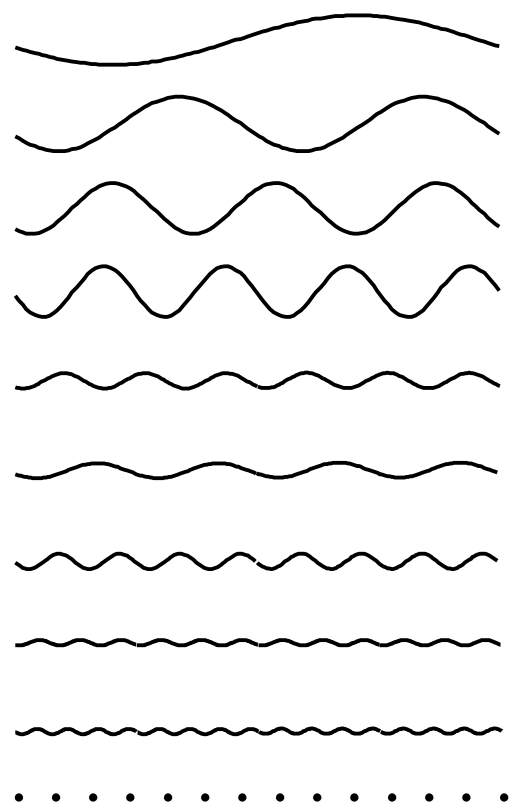
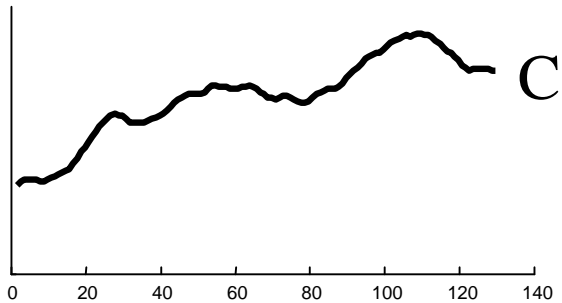
An Example of a Dimensionality Reduction Technique I



Raw Data

0.4995
0.5264
0.5523
0.5761
0.5973
0.6153
0.6301
0.6420
0.6515
0.6596
0.6672
0.6751
0.6843
0.6954
0.7086
0.7240
0.7412
0.7595
0.7780
0.7956
0.8115
0.8247
0.8345
0.8407
0.8431
0.8423
0.8387
...

An Example of a Dimensionality Reduction Technique II



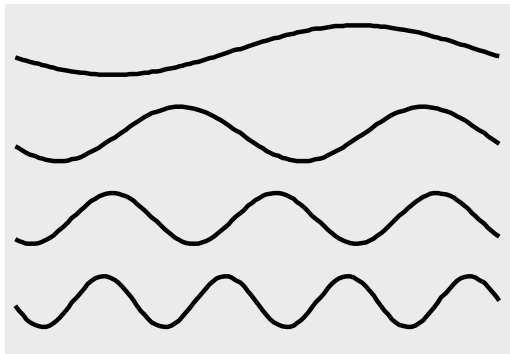
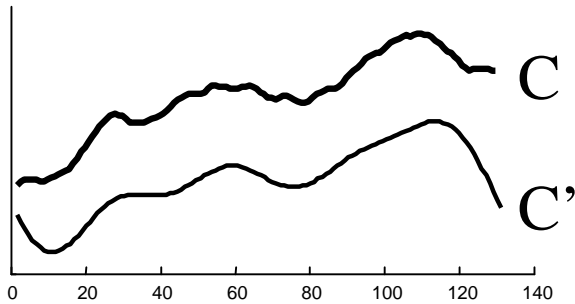
**Raw
Data**

0.4995
0.5264
0.5523
0.5761
0.5973
0.6153
0.6301
0.6420
0.6515
0.6596
0.6672
0.6751
0.6843
0.6954
0.7086
0.7240
0.7412
0.7595
0.7780
0.7956
0.8115
0.8247
0.8345
0.8407
0.8431
0.8423
0.8387
...

**Fourier
Coefficients**

1.5698
1.0485
0.7160
0.8406
0.3709
0.4670
0.2667
0.1928
0.1635
0.1602
0.0992
0.1282
0.1438
0.1416
0.1400
0.1412
0.1530
0.0795
0.1013
0.1150
0.1801
0.1082
0.0812
0.0347
0.0052
0.0017
0.0002
...

An Example of a Dimensionality Reduction Technique III



**Raw
Data**

0.4995
0.5264
0.5523
0.5761
0.5973
0.6153
0.6301
0.6420
0.6515
0.6596
0.6672
0.6751
0.6843
0.6954
0.7086
0.7240
0.7412
0.7595
0.7780
0.7956
0.8115
0.8247
0.8345
0.8407
0.8431
0.8423
0.8387
...

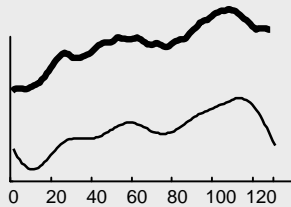
**Fourier
Coefficients**

1.0698
0.5485
0.4160
0.3406
0.2709
0.1670
0.1667
0.1928
0.1635
0.1602
0.0992
0.1282
0.1438
0.1416
0.1400
0.1410
0.1530
0.0795
0.1013
0.1150
0.0801
0.1282
0.0812
0.1347
0.0652
0.0977
0.0998
...

**Truncated
Fourier
Coefficients**

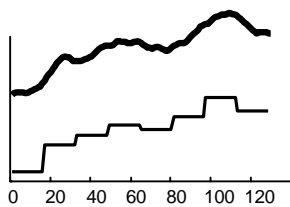
1.0698
0.5485
0.4160
0.3406
0.2709
0.1670
0.1667
0.1928

We have
discarded $\frac{15}{16}$
of the data



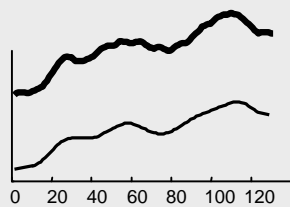
DFT

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



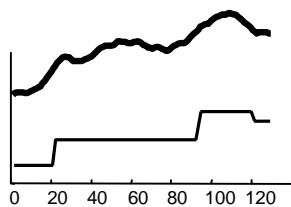
DWT

Chan & Fu. ICDE 1999



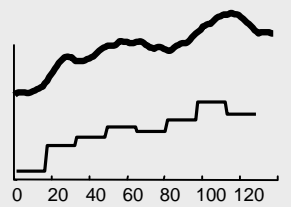
SVD

Korn, Jagadish &
Faloutsos. SIGMOD 1997



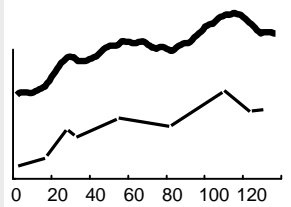
APCA

Keogh, Chakrabarti, Pazzani &
Mehrotra SIGMOD 2001



PAA

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



PLA

Morinaka,
Yoshikawa, Amagasa, &
Uemura. PAKDD 2001

But which is the best dimensionality reduction technique?

- Discrete Fourier Transform **DFT** {1}, [6, 10, 13, 18, 19, 21, 24].
- Discrete Wavelet Transform **DWT** {5}, [13, 24].
- Piecewise Constant Approximation **PCA** {15}, [26].
- Piecewise Linear Approximation **PLA**{17}, [22].
- Inner Product Approximation {9}.
- Adaptive Piecewise Constant Approximation {6}.

Key { *Introducing paper* }, [*follow up papers*]

Empirical studies [24], suggest that at least DFT and DWT have the same performance. Our insight, this is true only on average!

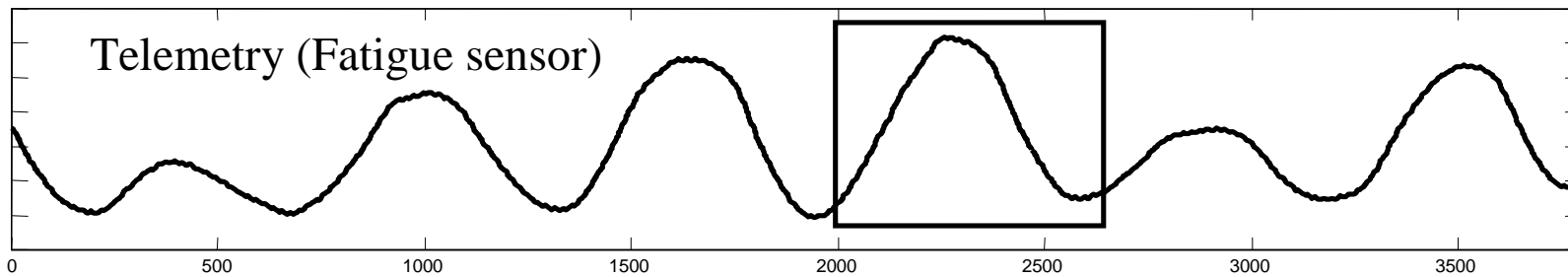
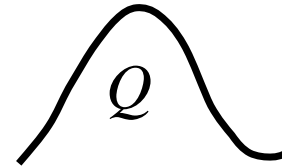
Three independent studies in the last few years suggest that DWT and DFT have about the same pruning power

(*Pruning power is measures in disk accesses, the fewer the better*)

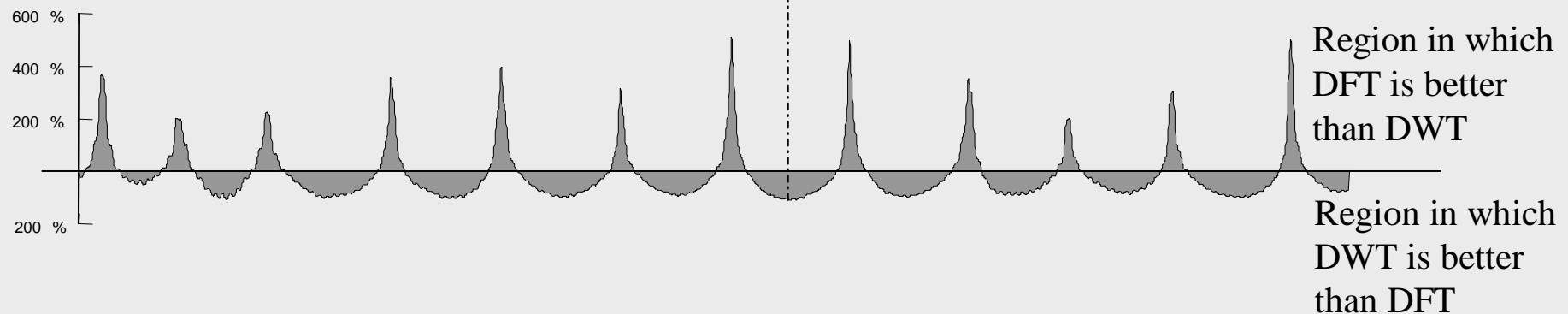
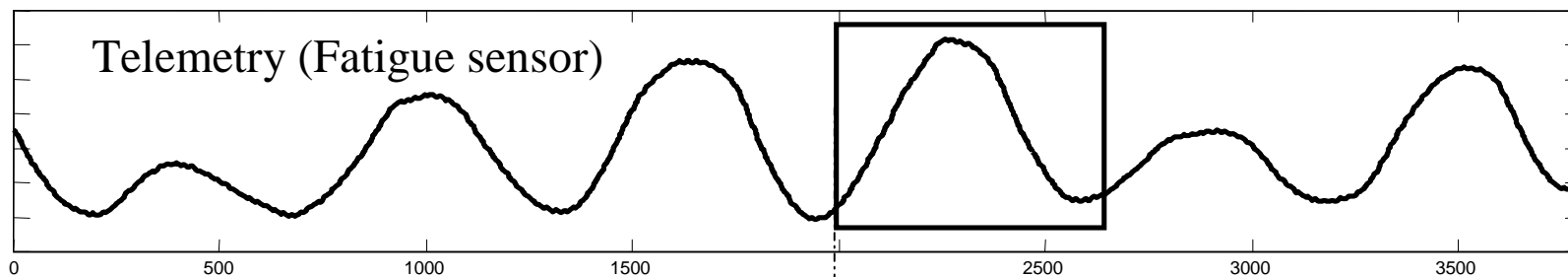
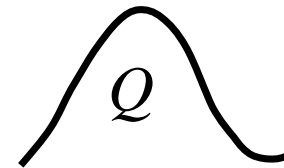
	DWT	DFT
Query1	55	100
Query2	122	22
Query3	34	89
<i>Mean</i>	<i>70.3</i>	<i>70.6</i>

However, only the mean performance was studied, this masks the fact that on any single query, the two approaches can have very different performance.

If two people index the dataset below, one using DFT and one using DWT, the average performance will be about the same...



...however, on any individual query one approach may be orders of magnitude better than the other.



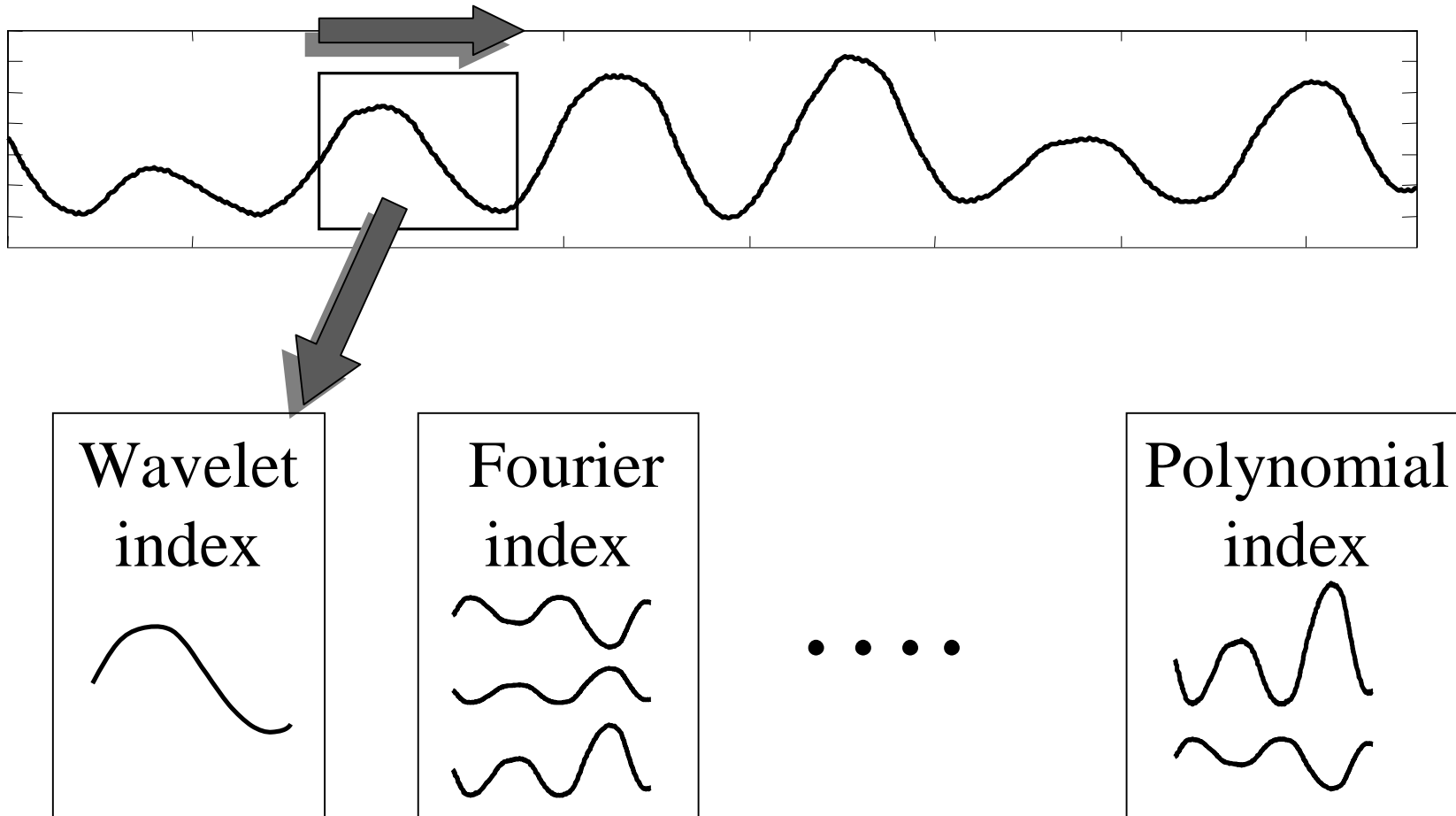
This observation motives our approach

If some sections of the data would better off being indexed with one representation and some sections of the data would better off being indexed with a different representation, then let us index each data object into the index to which it is best suited.

Instead of one big homogeneous index with some objects poorly approximated, we have several heterogeneous indices, each one containing tightly approximated objects.

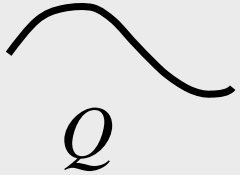
We call this idea E-Index (Ensemble-Index).

Building the index

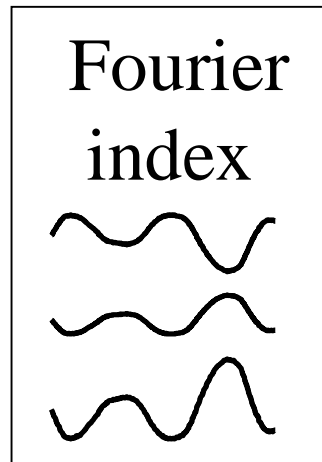
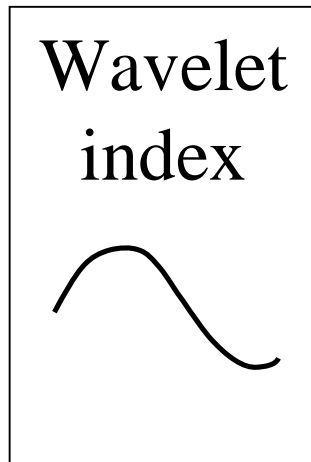


Instead of one big homogeneous index, we have several heterogeneous indices.

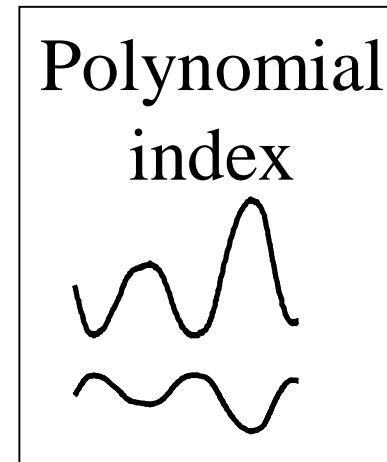
Searching the Indices I



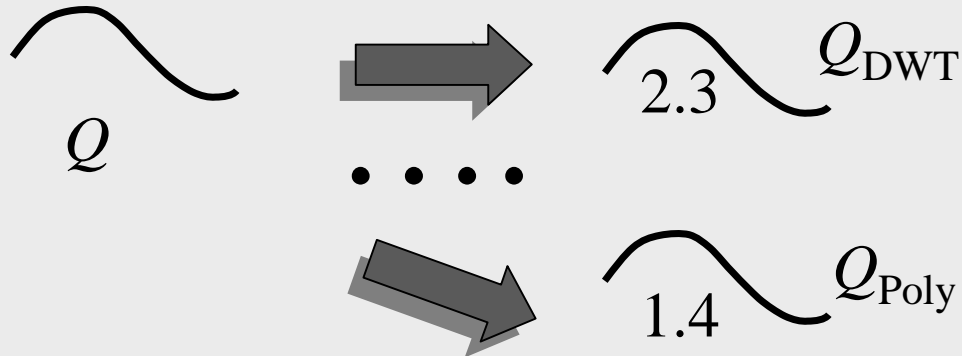
We have a Query Q , and a set of indices \mathbf{R} . In what order should we search the indices?



...



Searching the Indices II

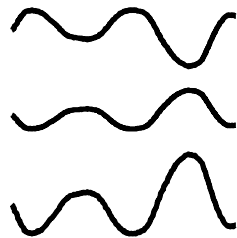


The relative fidelity of the query Q in each representation is measured.

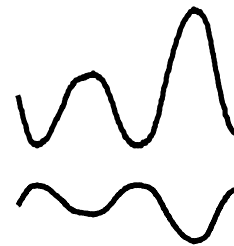
This information is used to sort the indices in the order in which they should be searched.

Indices are searched in this order

Fourier index



Polynomial index



Wavelet index



Surprising Property of E-Index

It appears that E-Index will perform as well as the best individual representation on any individual query...

	DWT	DFT	E-Index $\{\mathbf{R} = \text{DWT}, \text{DFT}\}$
Query1	55	100	55
Query2	122	22	22
Query3	34	89	34
Mean	70.3	70.6	37.0

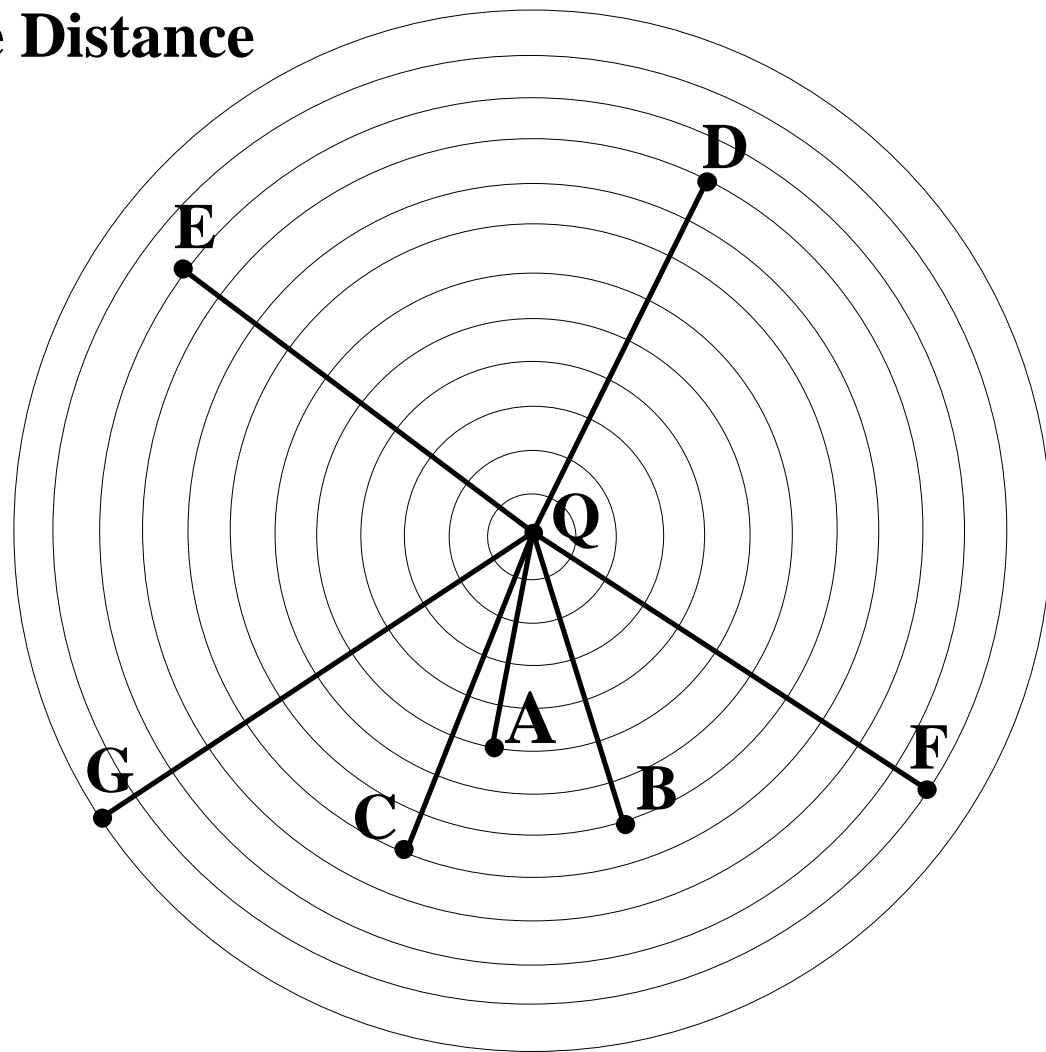
In fact, it is possible that E-Index can perform **better** than the best individual representation on any individual query!

	DWT	DFT	E-Index $\{\mathbf{R} = \text{DWT}, \text{DFT}\}$	
Query1	55	100	21	“superlinear” speedup
...	

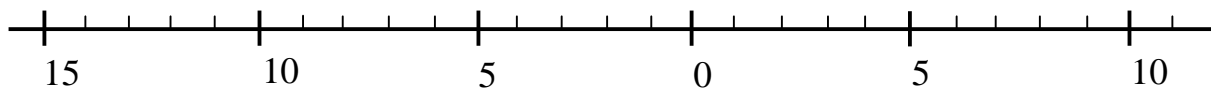
The “superlinear” property of E-Index is so unintuitive it is worth seeing a worked example.

	$D_{\text{true}}(Q,O)$	$D_{\text{DFT}}(\bar{Q},\bar{O})$	$D_{\text{DWT}}(\bar{Q},\bar{O})$
A	5	1	4
B	7	2	6
C	8	3	7
D	9	4	8
E	10	8	1
F	11	9	2
G	12	10	3

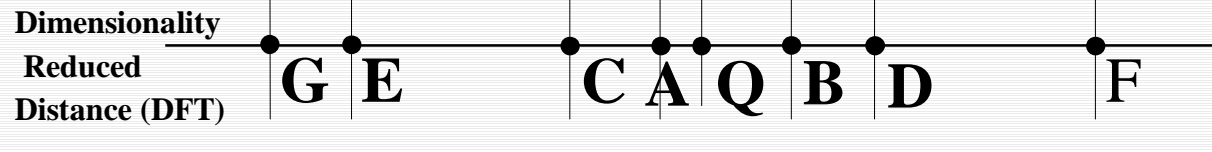
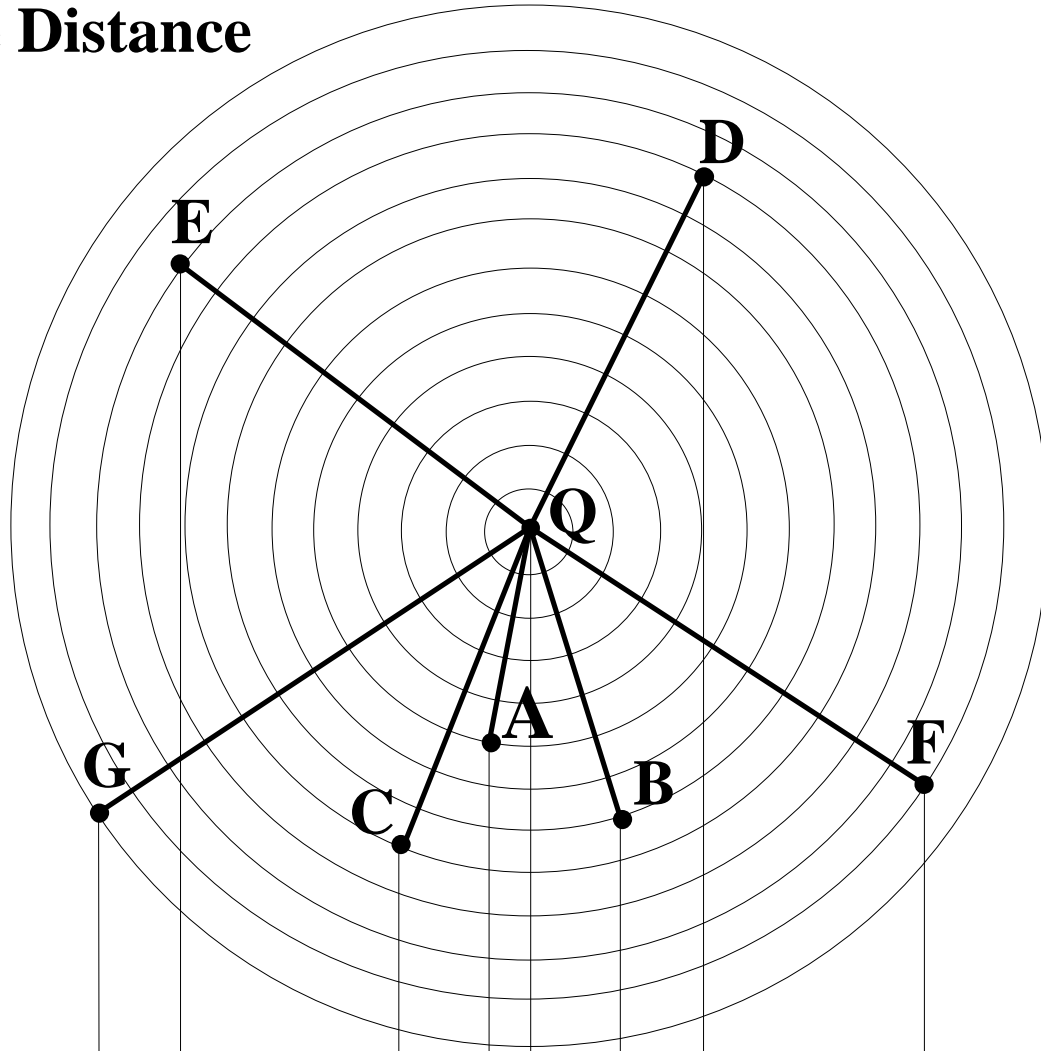
True Distance



	$D_{\text{true}}(Q, O)$
A	5
B	7
C	8
D	9
E	10
F	11
G	12

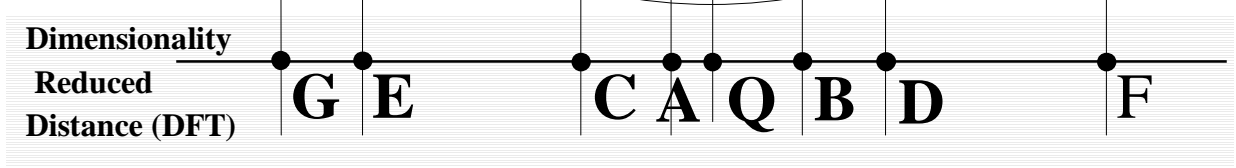
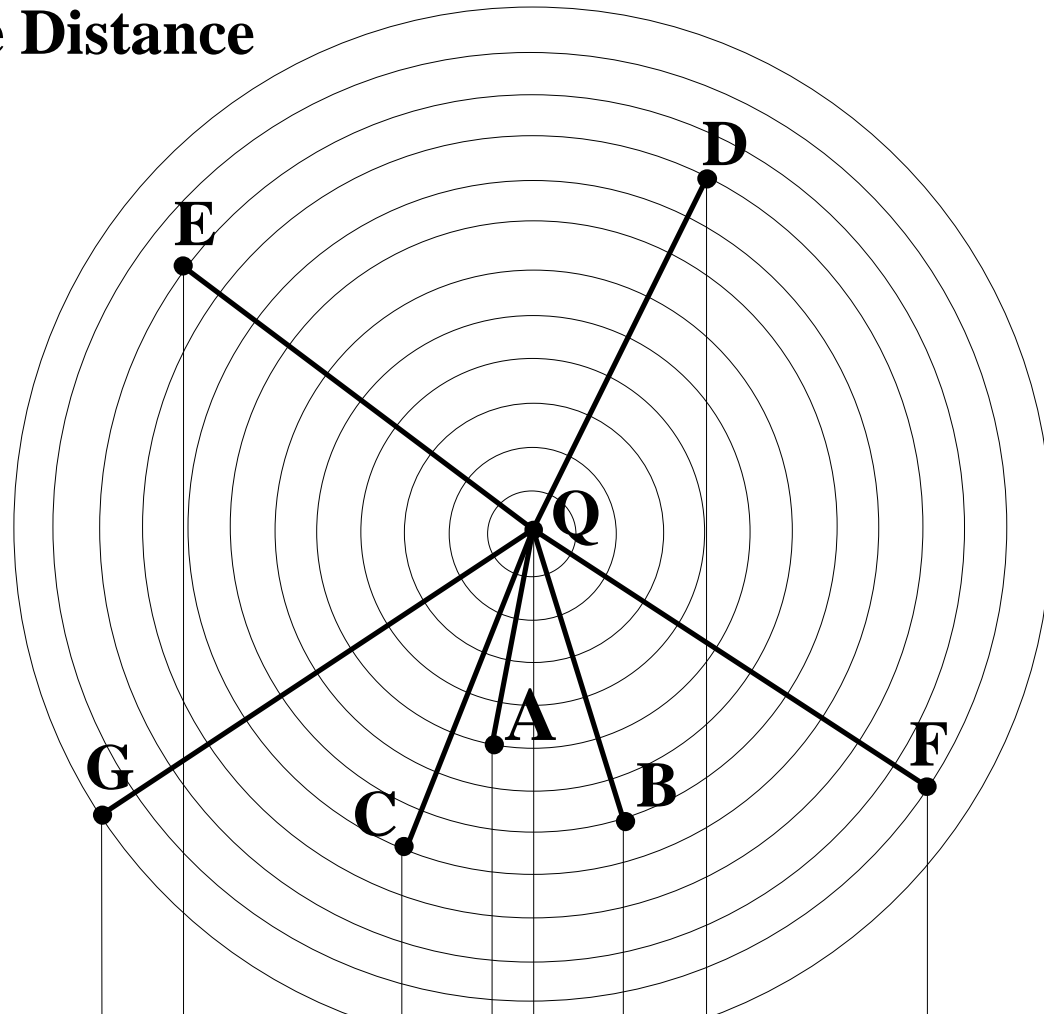


True Distance



	$D_{\text{true}}(Q, O)$	$D_{\text{DFT}}(\bar{q}, \bar{o})$
A	5	1
B	7	2
C	8	3
D	9	4
E	10	8
F	11	9
G	12	10

True Distance



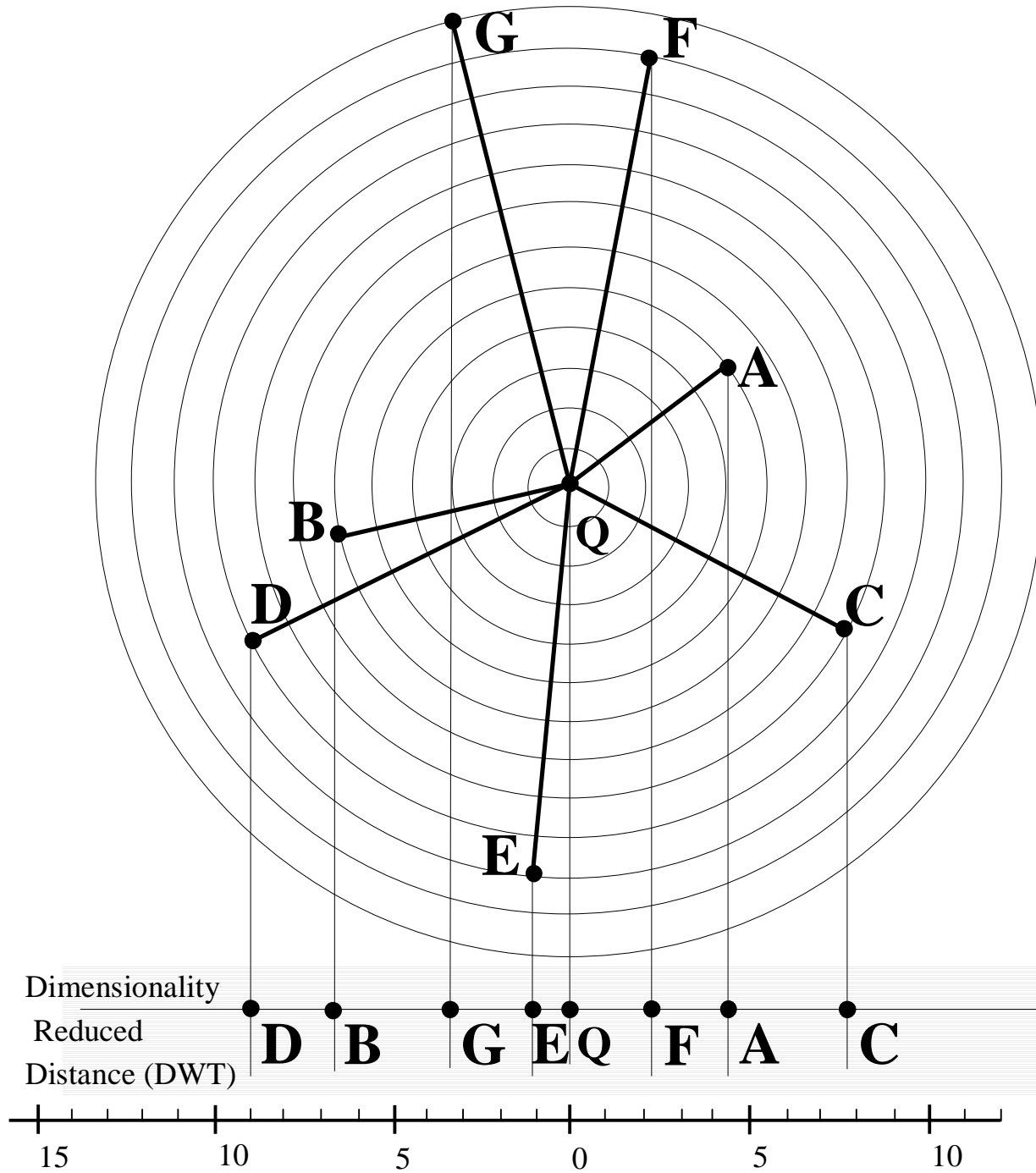
Pointers to objects
A, B, C, D, E, F, G
1, 2, 3, 4, 8, 9, 10
are place into priority queue

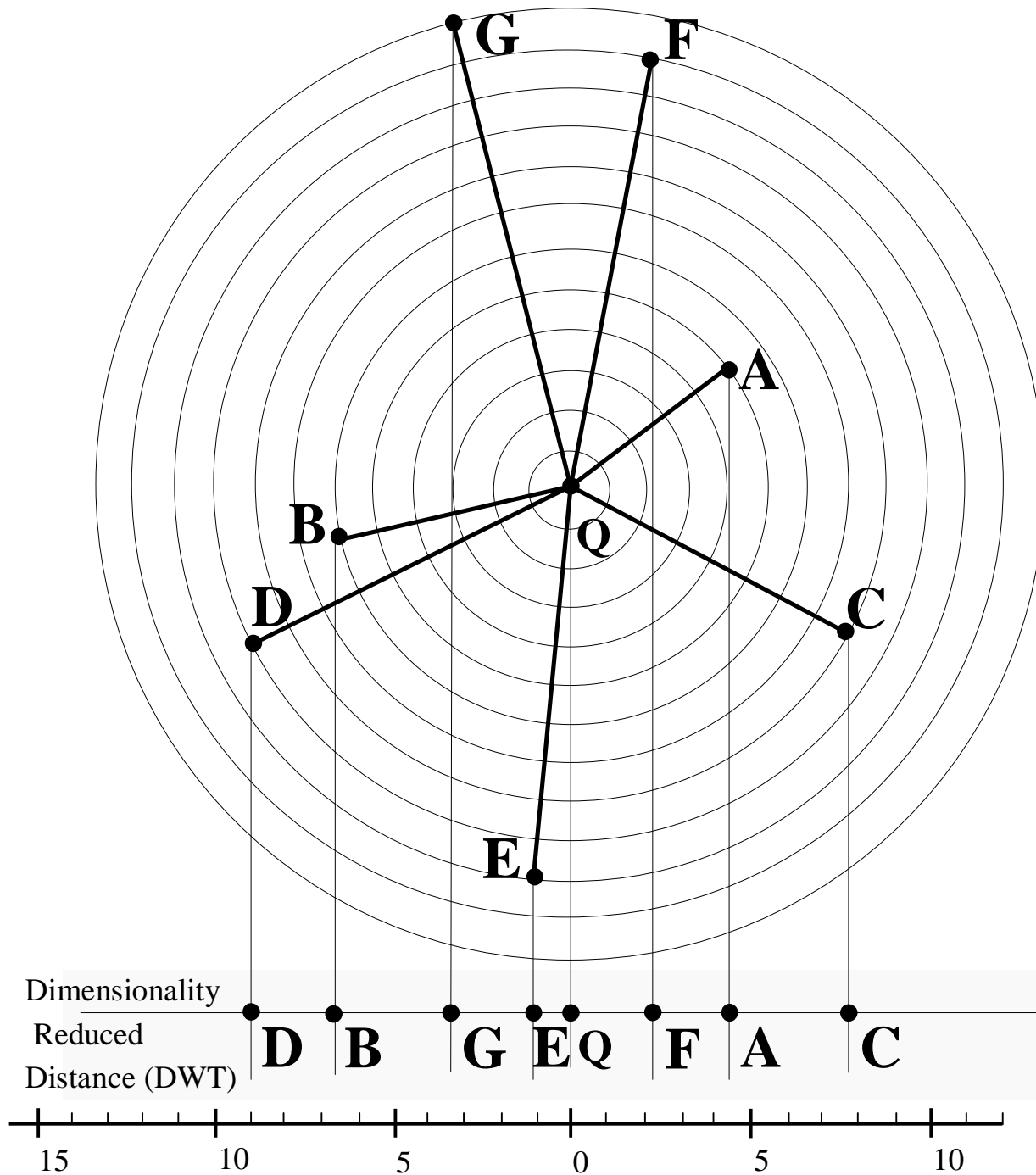
A is retrieved and the true distance $D(Q, A)$ is found to be 5.

E, F, G
8, 9, 10
can be pruned...

.... the DFT-Index must make 4 disk accesses, to objects **A, B, C and D,**

Now let us consider the wavelet dimensionality reduction...





Pointers to objects
E, F, G, A, B, C, D
 1, 2, 3, 4, 6, 7, 8
 are place into priority queue

E is retrieved and the true distance $D(Q, E)$ is found to be 10.

Nothing can be pruned yet,
F is retrieved next...

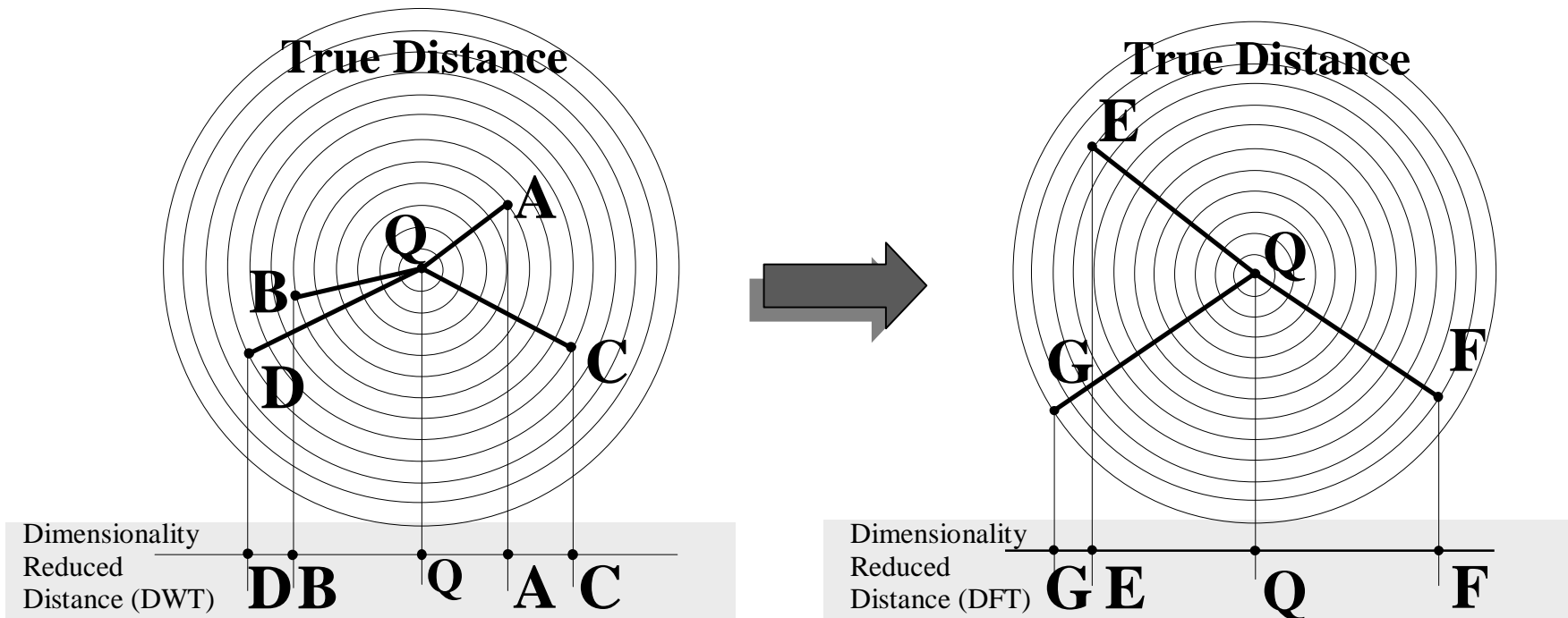
.... the DWT-Index must make 4 disk accesses (to objects **E, F, G** and **A**).

E-Index places objects **A**, **B**, **C**, and **D** into a DWT index and objects **G**, **E** and **F** into a DFT index.

The query **Q** is transformed into both representations. Because it is better represented in DWT, the DWT index will be searched first.

Objects **A**, **B**, **C**, **D** are placed into the priority queue
4, 6, 7, 8

A is retrieved because $D(Q, A)$ is 5, we can prune **B**, **C**, and **D**. As we are building the DFT priority queue, we discover its smallest element is greater than 5, so we are done!



Summary of Worked Example

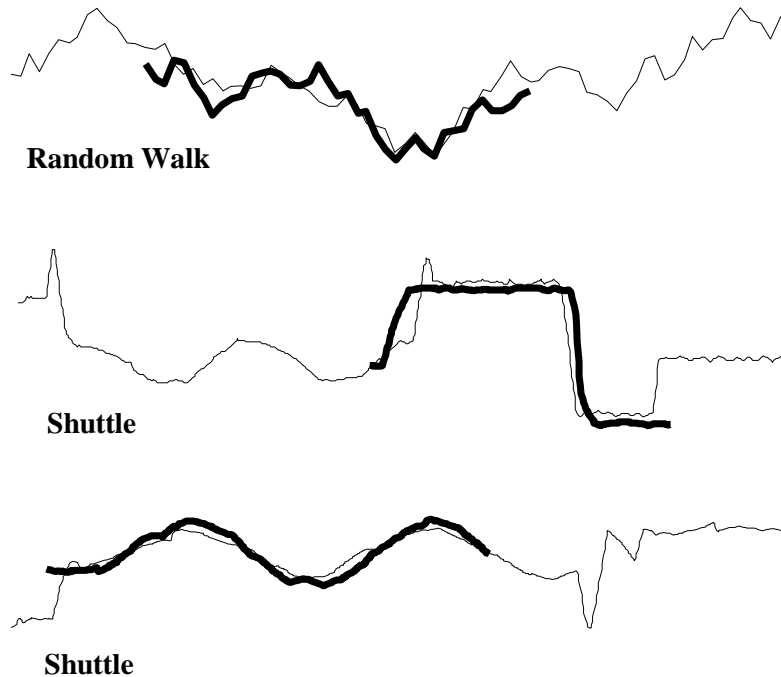
So DWT-index requires 4 disk accesses

DFT-index requires 4 disk accesses

But E-index only requires 1 disk access

So E-Index can actually be better than the best of its component parts!

Experimental results



We measure P , the fraction of the database that must be examined before we can guarantee that we have found the nearest match to a 1-NN query.

For simplicity and clarity we limit ensembles to size 2 (E-Index-2, $\mathbf{R} = \{\text{DWT}, \text{DFT}\}$) and size 3 (E-Index-3, $\mathbf{R} = \{\text{DWT}, \text{DFT}, \text{APCA}\}$)

and we compared them to standalone versions of DWT, DFT and APCA.

Data / Query Length	DWT	DFT	APCA	E-Index-2	E-Index-3
Random Walk 512	0.31	0.24	0.32	0.19	0.19
Random Walk 1024	0.47	0.43	0.51	0.38	0.37
Space Shuttle 512	0.023	0.021	0.016	0.011	0.006
Space Shuttle 1024	0.041	0.039	0.022	0.027	0.010

The ensemble technique outperforms all its component representations.

The more representations you add to the ensemble the better it gets.

Conclusions

We have introduced a novel framework for indexing data with ensembles of representations.

We have empirically shown that our approach outperforms existing techniques.

Future Work

We are currently using a heuristic to order the indices at query time. If we could make all representations share the same priority queue, performance would improve and we could prove some properties of our approach...

- [1] Agrawal, R., Faloutsos, C., & Swami, A. (1993). Efficient similarity search in sequence databases. *Proceedings of the 4th Conference on Foundations of Data Organization and Algorithms*.
- [2] Agrawal, R., Psaila, G., Wimmers, E. L., & Zait, M. (1995). Querying shapes of histories. *Proceedings of the 21st International Conference on Very Large Databases*.
- [3] Agrawal, R., Lin, K. I., Sawhney, H. S., & Shim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in times-series databases. *Proceedings of 21th International Conference on Very Large Data Bases*. Zurich. pp 490-50.
- [4] Bay, S. D. (2000). The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science.
- [5] Bennett, K., Fayyad, U. & Geiger, D. (1999). Density-based indexing for approximate nearest-neighbor queries. *Proceedings 5th International Conference on Knowledge Discovery and Data Mining*. pp. 233-243, ACM Press, New York.
- [6] Chakrabarti, K & Mehrotra, S. (1999). The Hybrid Tree: An index structure for high dimensional feature spaces. *Proceedings of the 15th IEEE International Conference on Data Engineering*.
- [7] Chakrabarti, K & Mehrotra, S (2000). Local dimensionality reduction: A new approach to indexing high dimensional spaces. *Proceedings of the 26th Conference on Very Large Databases, Cairo, Egypt*.
- [8] Chakrabarti, K., Ortega-Binderberger, M., Porkaew, K & Mehrotra, S. (2000) Similar shape retrieval in MARS. *Proceeding of IEEE International Conference on Multimedia and Expo*.
- [9] Chan, K. & Fu, W. (1999). Efficient time series matching by wavelets. *Proceedings of the 15th IEEE International Conference on Data Engineering*.
- [10] Chandrasekaran, S., Manjunath, B.S., Wang, Y. F. Winkeler, J. & Zhang, H. (1997). An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, Vol. 59, No. 5, pp. 321-332.
- [11] Chu, K & Wong, M. (1999). Fast time-series searching with scaling and shifting. *Proceedings of the 18th ACM Symposium on Principles of Database Systems*, Philadelphia.
- [12] Das, G., Lin, K. Mannila, H., Renganathan, G., & Smyth, P. (1998). Rule discovery from time series. *Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining*. pp 16-22.
- [13] Debregeas, A. & Hebrail, G. (1998). Interactive interpretation of Kohonen maps applied to curves. *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*. pp 179-183.
- [14] Evangelidis, G., Lomet, D. & Salzberg B (1997). The hB-Pi-Tree: A multi-attribute index supporting concurrency, recovery and node consolidation. *VLDB Journal* 6(1): 1-25.
- [15] Faloutsos, C., Jagadish, H., Mendelzon, A. & Milo, T. (1997). A signature technique for similarity-based queries. *SEQUENCES 97*, Positano-Salerno, Italy.
- [16] Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*. Minneapolis.
- [17] Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. *Proceedings ACM SIGMOD Conference*. pp 47-57.
- [18] Hellerstein, J. M., Papadimitriou, C. H., & Koutsoupias, E. (1997). Towards an analysis of indexing schemes. *Sixteenth ACM Symposium on Principles of Database Systems*.
- [19] Hjaltason, G., Samet, H (1995). Ranking in spatial databases. *Symposium on Large Spatial Databases*. pp 83-95.
- [20] Huang, Y. W., Yu, P. (1999). Adaptive Query processing for time-series data. *Proceedings of the 5th International Conference of Knowledge Discovery and Data Mining*. pp 282-286.
- [21] Jonsson, H., & Badal, D. (1997). Using signature files for querying time-series data. *First European Symposium on Principles of Data Mining and Knowledge Discovery*.
- [22] Kahveci, T. & Singh, A (2001). Variable length queries for time series data. *Proceedings 17th International Conference on Data Engineering*. Heidelberg, Germany.
- [23] Kanth, K.V., Agrawal, D., & Singh, A. (1998). Dimensionality reduction for similarity searching in dynamic databases. *Proceedings ACM SIGMOD Conf.*, pp. 166-176.
- [24] Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra (2000) Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*.
- [25] Keogh, E. & Pazzani, M. (1999). Relevance feedback retrieval of time series data. *Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*.
- [26] Keogh, E., & Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*. pp 239-241, AAAI Press.
- [27] Keogh, E., & Smyth, P. (1997). A probabilistic approach to fast pattern matching in time series databases. *Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining*. pp 24-20.

- [28] Korn, F., Jagadish, H & Faloutsos, C. (1997). Efficiently supporting ad hoc queries in large datasets of time sequences. Proceedings of *SIGMOD '97*, Tucson, AZ, pp 289-300.
- [29] Lam, S., & Wong, M (1998) A fast projection algorithm for sequence data searching. *Data & Knowledge Engineering* 28(3): 321-339.
- [30] Li, C., Yu, P. & Castelli V.(1998). MALM: A framework for mining sequence database at multiple abstraction levels. *CIKM*. pp 267-272.
- [31] Loh, W., Kim, S & Whang, K. (2000). Index interpolation: an approach to subsequence matching supporting normalization transform in time-series databases. *Proceedings 9th International Conference on Information and Knowledge Management*.
- [32] Moody, G. (2000). MIT-BIH Database Distribution [<http://ecg.mit.edu/index.html>]. Cambridge, MA.
- [33] Ng, M. K., Huang, Z., & Hegland, M. (1998). Data-mining massive time series astronomical data sets - a case study. *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*. pp 401-402
- [34] Park, S., Lee, D., & Chu, W. (1999). Fast retrieval of similar subsequences in long sequence databases. In *3rd IEEE Knowledge and Data Engineering Exchange Workshop*.
- [35] Pavlidis, T. (1976). Waveform segmentation through functional approximation. *IEEE Transactions on Computers*, Vol C-22, NO. 7 July.
- [36] Perng, C., Wang, H., Zhang, S., & Parker, S. (2000). Landmarks: a new model for similarity-based pattern querying in time series databases. *Proceedings 16th International Conference on Data Engineering*. San Diego, USA.
- [37] Porkaew, K., Chakrabarti, K. & Mehrotra, S. (1999). Query refinement for multimedia similarity retrieval in MARS. Proceedings of the ACM International Multimedia Conference, Orlando, Florida, pp 235-238
- [38] Qu, Y., Wang, C. & Wang, S. (1998). Supporting fast search in time series for movement patterns in multiples scales. *Proceedings 7th International Conference on Information and Knowledge Management*. Washington, DC.
- [39] Refiei, D. (1999). On similarity-based queries for time series data. *Proc of the 15th IEEE International Conference on Data Engineering*. Sydney, Australia.
- [40] Roussopoulos, N., Kelley, S. & Vincent, F. (1995). Nearest neighbor queries. *SIGMOD Conference 1995*: 71-79.
- [41] Seidl, T. & Kriegel, H. (1998). Optimal multi-step k-nearest neighbor search. *SIGMOD Conference*: pp 154-165.
- [42] Shatkay, H., & Zdonik, S. (1996). Approximate queries and representations for large data sequences. *Proceedings 12th IEEE International Conference on Data Engineering*. pp 546-553.
- [43] Shevchenko, M. (2000). [<http://www.iki.rssi.ru/>] Space Research Institute. Moscow, Russia.
- [44] Stollnitz, E., DeRose, T., & Salesin, D. (1994). Wavelets for computer graphics A primer: *IEEE Computer Graphics and Applications*.
- [45] Struzik, Z. & Siebes, A. (1999). The Haar wavelet transform in the time series similarity paradigm. *Proceedings 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases*. pp 12-22.
- [46] Wang, C. & Wang, S. (2000). Supporting content-based searches on time Series via approximation. *International Conference on Scientific and Statistical Database Management*.
- [47] Weigend, A. (1994). The Santa Fe Time Series Competition Data [<http://www.stern.nyu.edu/~aweigend/Time-Series/SantaFe.html>]
- [48] Welch, D. & Quinn, P (1999). <http://www.macho.mcmaster.ca/Project/Overview/status.html>
- [49] Wu, Y., Agrawal, D. & Abbadi, A.(2000). A Comparison of DFT and DWT based Similarity Search in Time-Series Databases. *Proceedings of the 9th International Conference on Information and Knowledge Management*.
- [50] Wu, D., Agrawal, D., El Abbadi, A. Singh, A. & Smith, T. R. (1996). Efficient retrieval for browsing large image databases. *Proc of the 5th International Conference on Knowledge Information*. pp 11-18, Rockville, MD.
- [51] Yi, B.K., Jagadish, H., & Faloutsos, C. (1998). Efficient retrieval of similar time sequences under time warping. *IEEE International Conference on Data Engineering*. pp 201-208.
- [52] Yi, B.K., & Faloutsos, C.(2000). Fast time sequence indexing for arbitrary Lp norms. *Proceedings of the 26th International Conference on Very Large Databases*, Cairo, Egypt.