# Efficient Rotation Invariant Retrieval of Shapes with Applications in Medical Databases

Selina Chu, Shrikanth Narayanan, C.-C. Jay Kuo

Department of Computer Science
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

Email: selinach@sipi.usc.edu

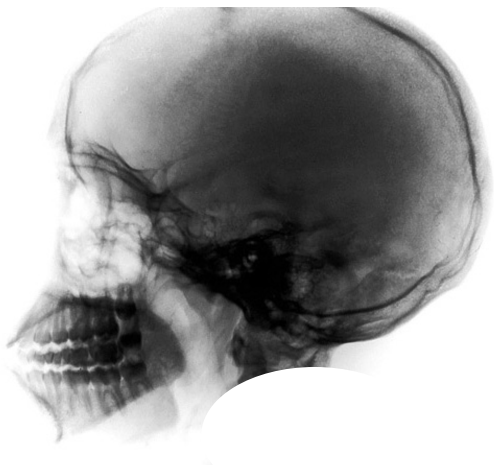
# Outline

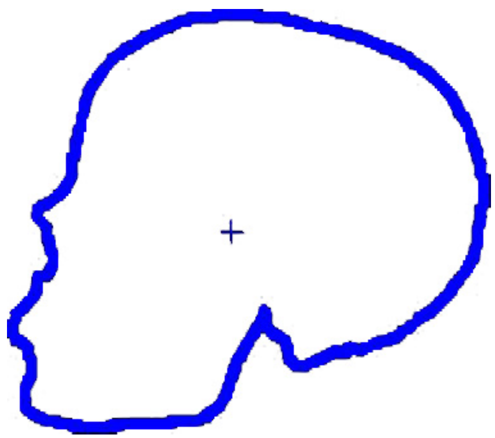
- Shape recognition and representation
- Checking for rotation invariance of shapes
- Background
- Current method and problem
- Our framework
- Experimental setup and results
- Conclusion and future work

# Shape Recognition

- Efficient content-based image retrieval
  - *Recognition of contours provide a mean to accelerate search*
- Direct comparison of images is intractable for large databases
  - *High dimensionality in raw data*
  - *Idiosyncrasies in details hinder the matching process*
- Conditions for effective shape matching
  - Similarity measure must be robust to various transformations and modest occlusions
  - Transformations, such as scaling and translation, can be dealt with by data representations or the similarity measure
  - Rotation invariance: More difficult compared with translation and scaling *[22]*

# Representing Shapes



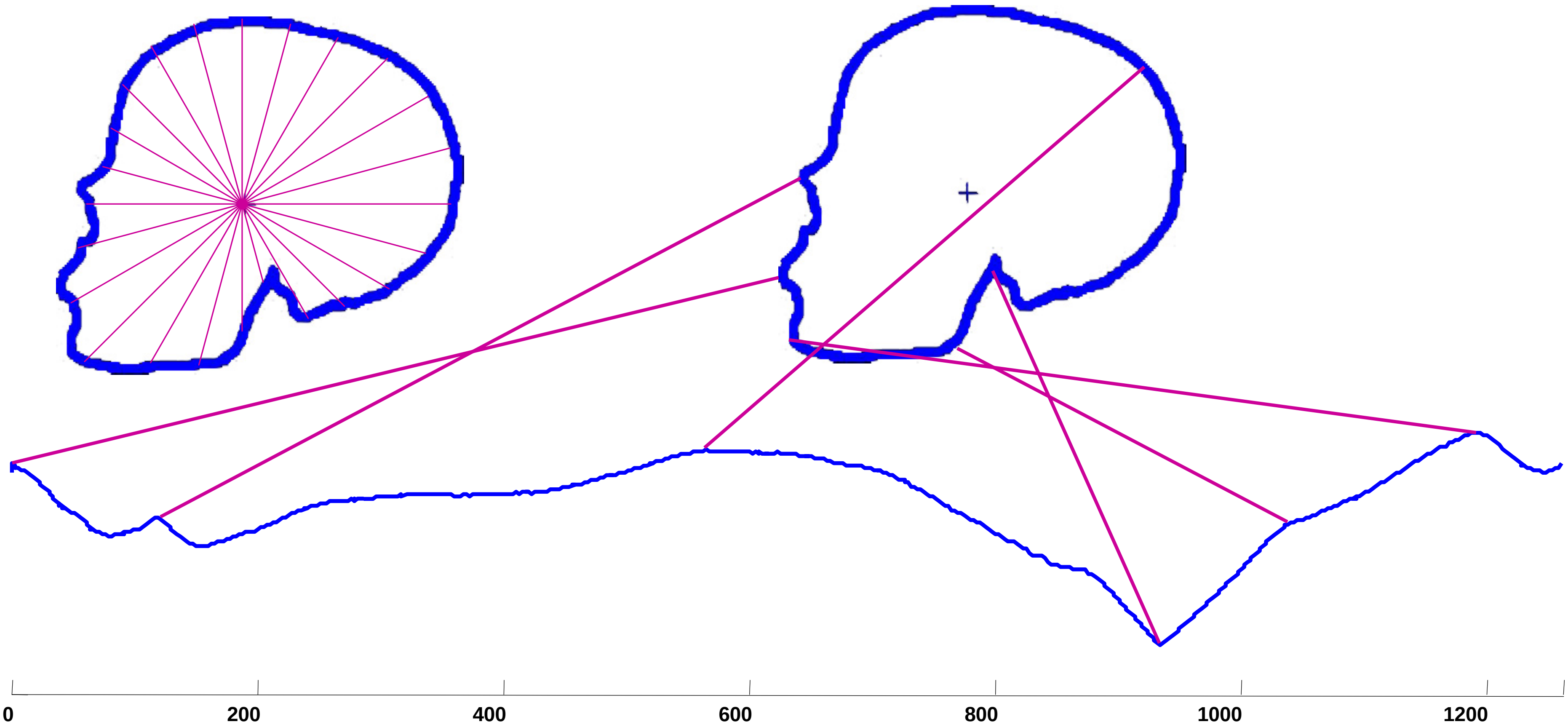


Represent shapes as one-dimensional data or time series:

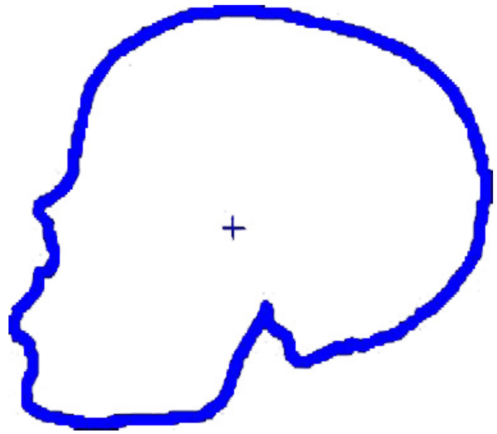
Numerous techniques for converting shapes to time series.

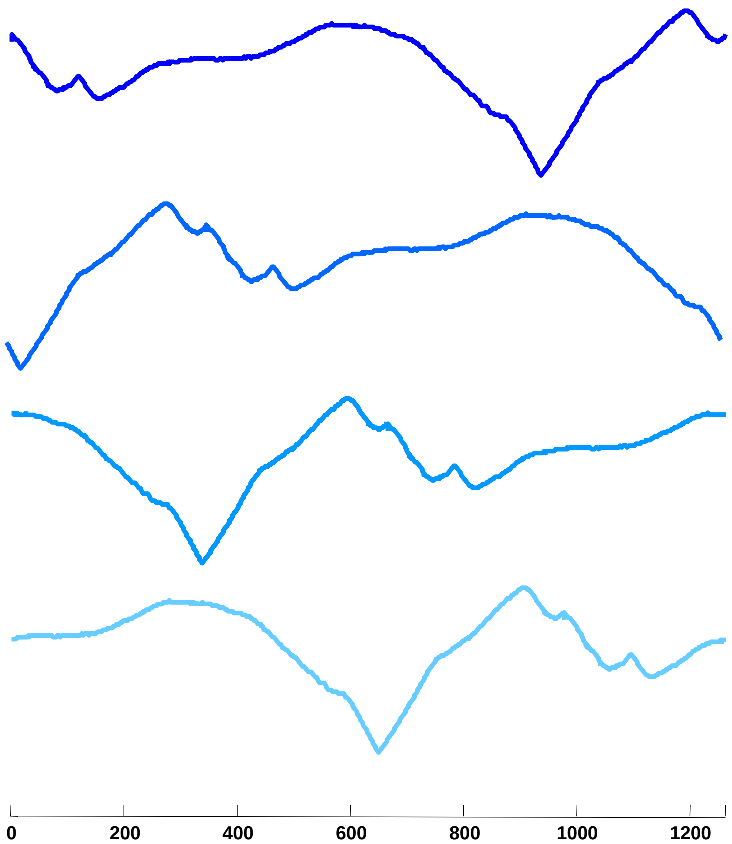
We'll see an example of a simple method…

# Representing Shapes as Time Series



# Checking Rotation Invariance






Rotation invariance can be achieved by checking all possible circular shifts (for every sampled point)…

# Dealing with Rotation Invariance

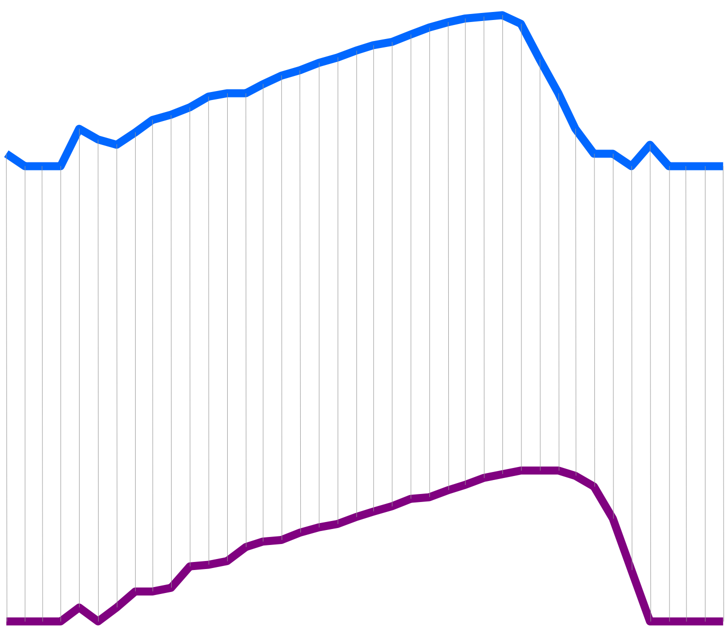
3 Major Techniques:

- Landmarking
  - To find the one “true” rotation and use that particular alignment as input into the distance measure
- Rotation Invariant Features
  - Extracting rotation invariant features, such as ratio of perimeter to area, fractal measure, elongatedness, curvature, entropy, perimeter of convex hull, etc.
- Brute force rotation alignment
  - Hold one shape fixed
  - While rotating the other shape (candidate object), find the similarity measure between the two, for all different rotations
  - Take minimum distance among all rotations
  - Repeat for every candidate in the database

# Rotation Invariant Problem

- Was demonstrated that brute force rotation alignment produces the best precision/recall and accuracy [1]
- Powerful contour matching algorithms uses dynamic programming, i.e. dynamic time warping (DTW)
  - *Robust to occlusions*
  - *Many distance measures for shapes, DTW and LCS are used in majority of literatures*
- Comparison between two shapes alone is $O(n^3)$
- Complexity grows exponentially as we increase the size of database

# Fixed Time Measures

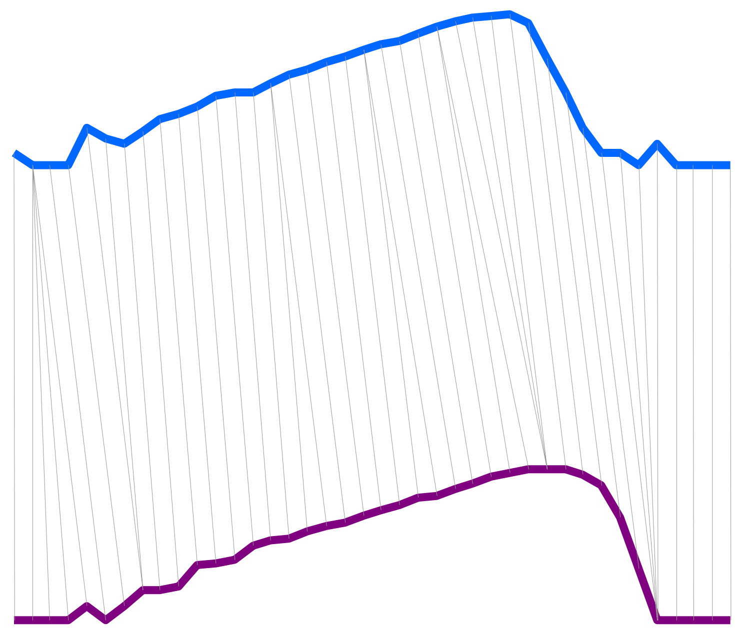


Fixed Time Axis

*Sequences are aligned “one to one”.*

**Euclidean Distance**

# Time Warped Measures



“Warped” Time Axis

*Nonlinear alignments are possible.*

**Dynamic Time Warping**

# Computing the DTW distance

$$DTW(Q,C) = \min\left\{ \sqrt{\sum_{k=1}^{K} w_k} \right.$$

$$w_k = (i,j)_k$$

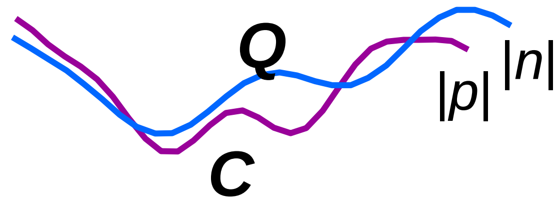
$w_k$ is a matrix element that defines a mapping between Q and C.

$$\gamma(i,j) = \mathrm{d}(qi,cj) + \min\{\, \gamma(i\text{-}1,j\text{-}1)\,,\ \gamma(i\text{-}1,j\,)\,,\ \gamma(i,j\text{-}1)\,\}$$

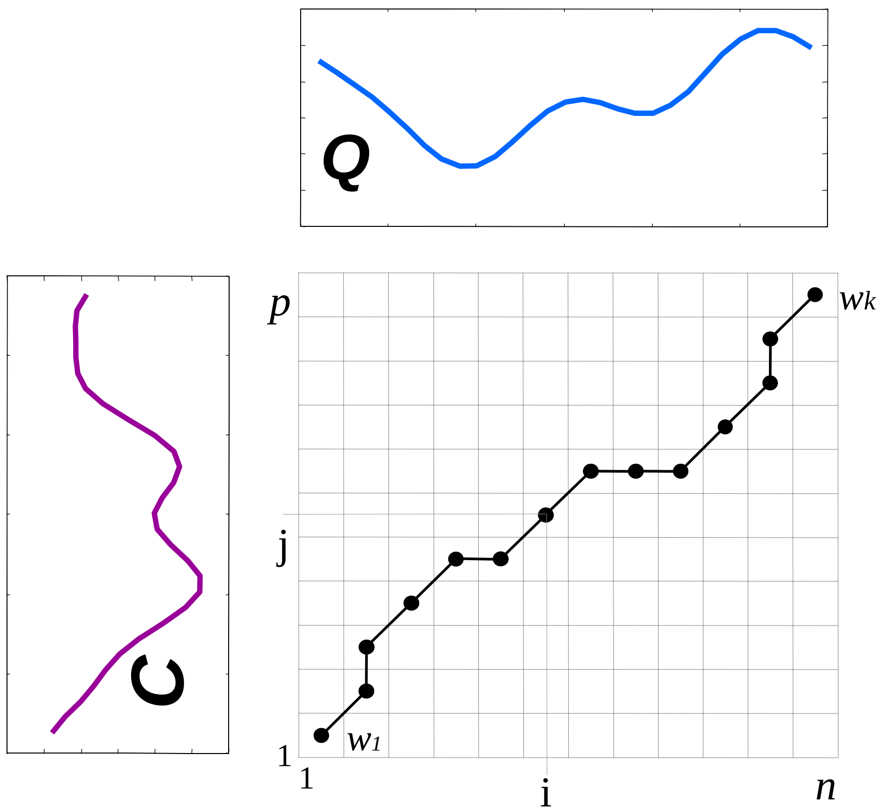
- $\gamma(i,j)$ is the cumulative distance of $d(i,j)$,
- d($i,j$) is the distance found in the current cell
- Find the minimum of the cumulative distances of the three adjacent elements

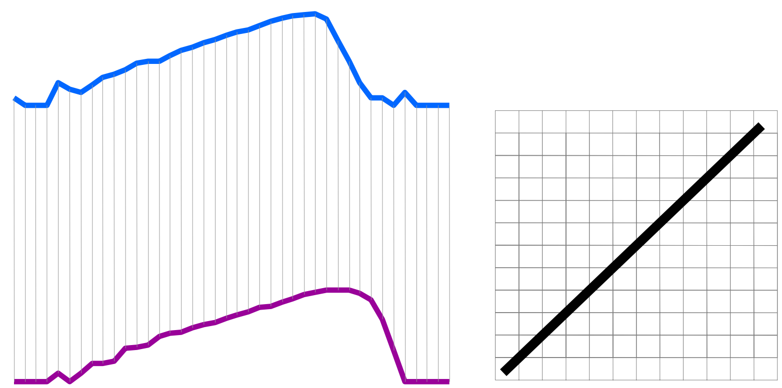
We'll see a visualization of that …

# Computing the Dynamic Time Warp Distance

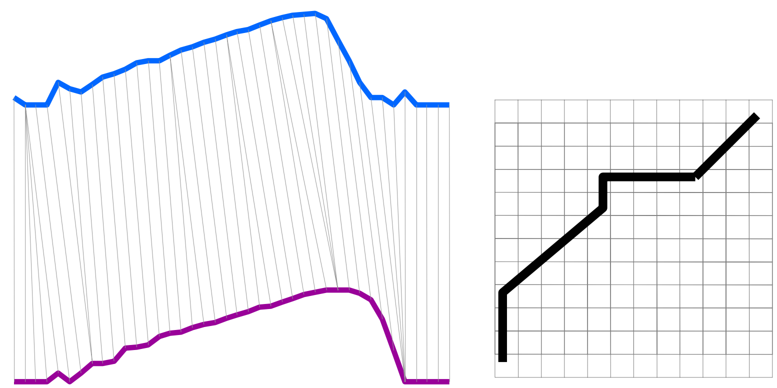



Note that the input sequences can be of different lengths



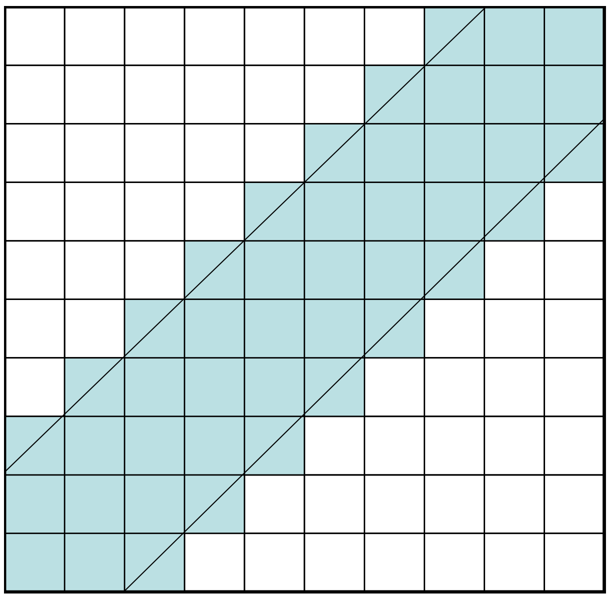



Euclidean distance is a special case of DTW

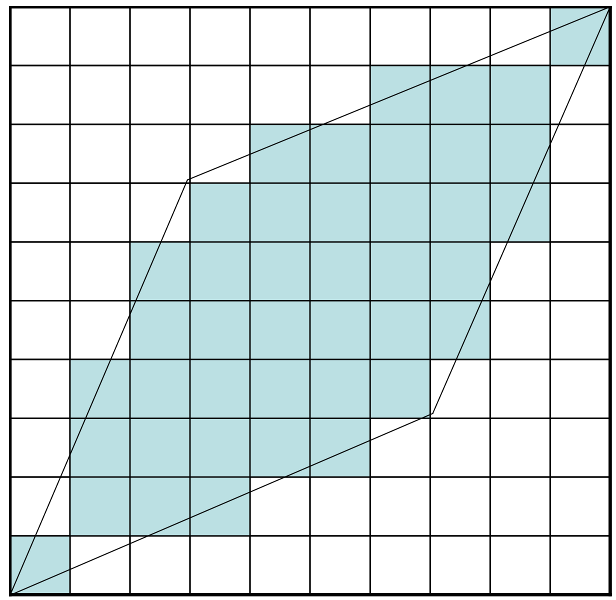


# Global Constraints

Constraining the warping path:



Sakoe-Chiba Band

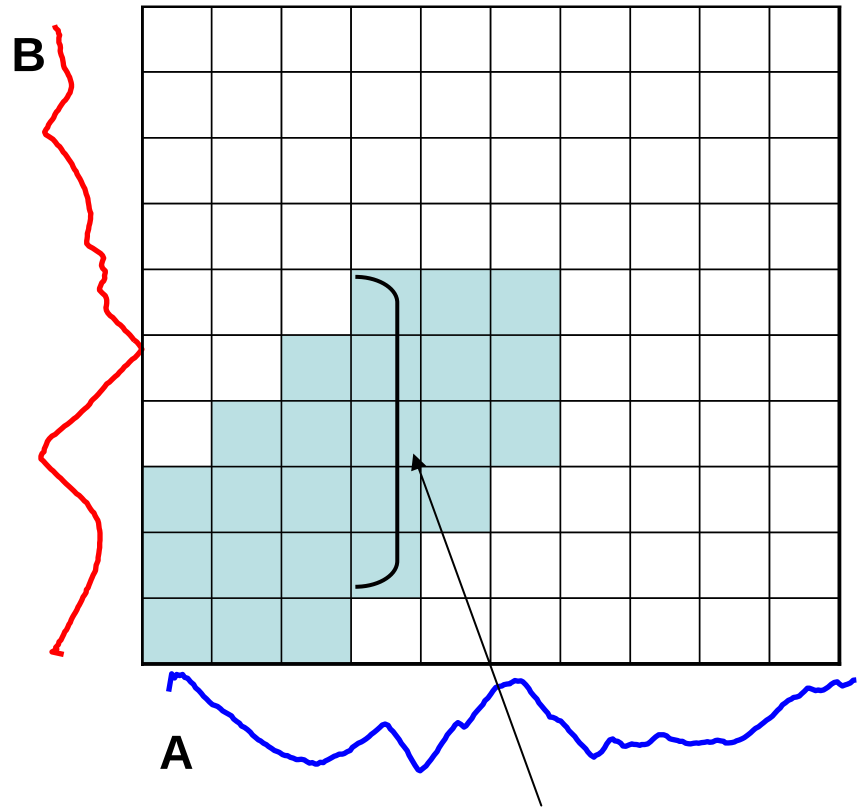


Itakura Parallelogram

# More on speeding up DTW

Early abandonment…

Discard the candidate from further consideration when the minimum of a column in the distance matrix is greater than "*best so far*"




Check if column is greater than "*best so far*"

# Optimizing search when using Euclidean Distance

• Trivially: Discard the candidate from further consideration as soon as the accumulated distance for that candidate exceeds the "*best so far*"

$$D(X,Y) = \sum_{i=1}^{n} \left(x_i - y_i\right)^2$$

Ignore the square root for Euclidean distance

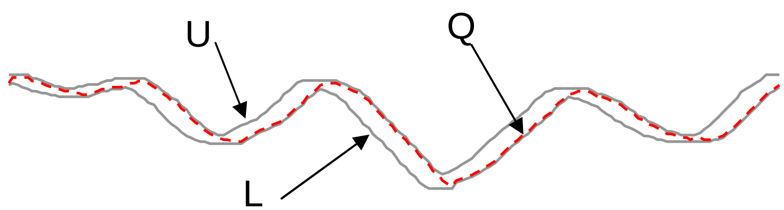
• Using best_so_far works as a lower bound for Euclidean distance because it obeys triangular inequality rule

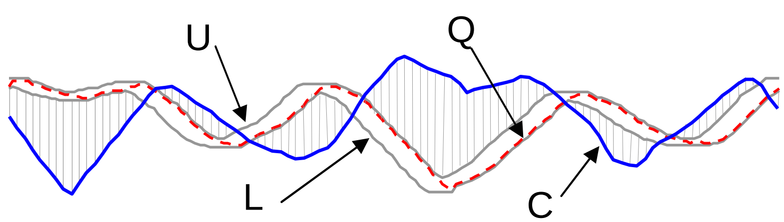
• DTW: does not obey triangular inequality

• Lower bounding is not as trivial, but possible…

# Lower Bounding for DTW

- Warping window with global constraints creates a bounding envelope around the time series, *Q*
- **Lower bound function:** squared sum of the distances of *C* that is outside of the envelope
- Obeys triangular inequality - to guarantee no false dismissals




- *r,* the allowed range of warping
  - \- Sakoe Chiba: *r* is constant
  - \- Itakura: *r* is function of *i*

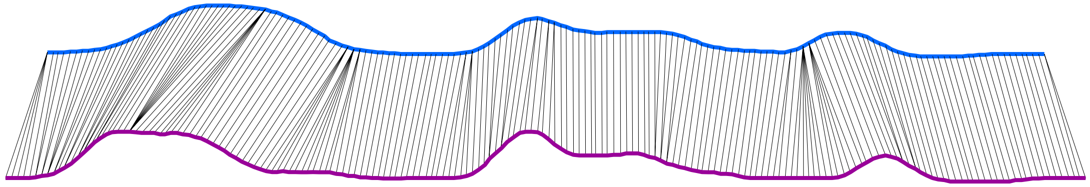
- U and L, upper and lower of envelope,
  - $U_i = max(q_{i-r}, \ldots, q_{i+r})$
  - $L_i = min(q_{i-r}, \ldots, q_{i+r})$ $\quad \forall i \quad U_i \geq q_i \geq L_i$

$$LB\_DTW(Q,C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i - U_i)^2 & \text{if} \quad c_i > U_i \\ (c_i - L_i)^2 & \text{if} \quad c_i < L_i \\ 0 & \text{otherwise} \end{cases}}$$

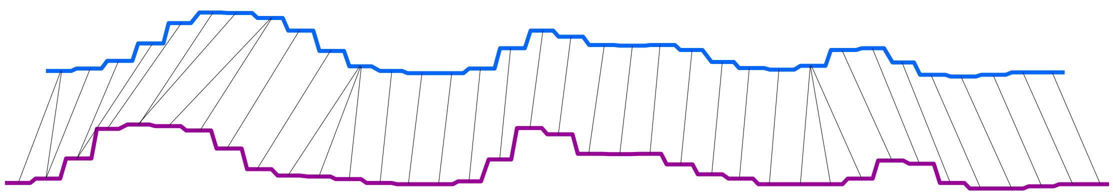
Keogh, *Exact Indexing of Dynamic Time Warping*, VLDB 2002: 406-417.

# Representing the Data – Dimensionality Reduction

**Piecewise Constant Approximation:** approximates a time series by dividing into equal-length segments



22.7 sec



1.3 sec

$$\bar{c}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} c_j$$

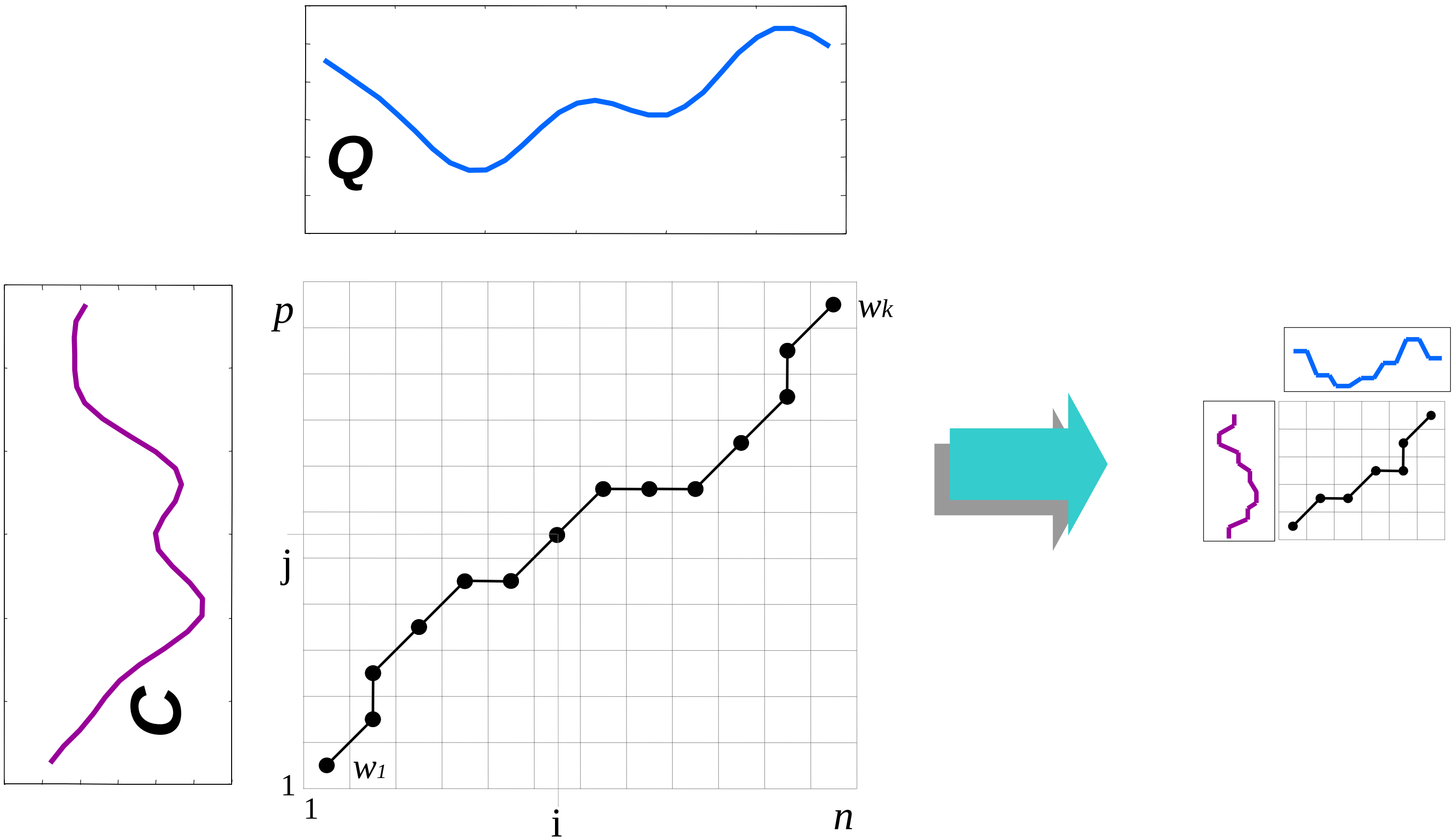
*C*: time series

*n*: original dimension

*N*: reduced dimension


- Yi and Faloutsos, *Fast Time Sequence Indexing for Arbitrary Lp Norms*, VLDB 2000.
- Keogh et al. *Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases*, Knowl. Inf. Syst. 2000.

# An Approximation to Dynamic Time Warp Distance




Simple Idea: Approximate the time series with some compressed or downsampled representation and perform DTW on the new representation, resulting in a smaller distance matrix

# Now What?

So how do we apply all these different techniques to the rotation invariance shape matching problem..

Lower bounding,

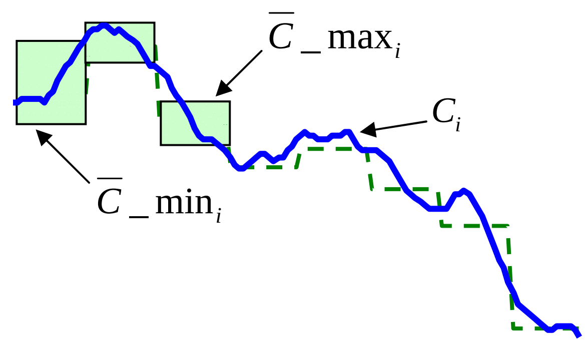
Dimensionality reduction,

And an iterative process

# Lower Bounding for Rotation Invariant using Euclidean Distance

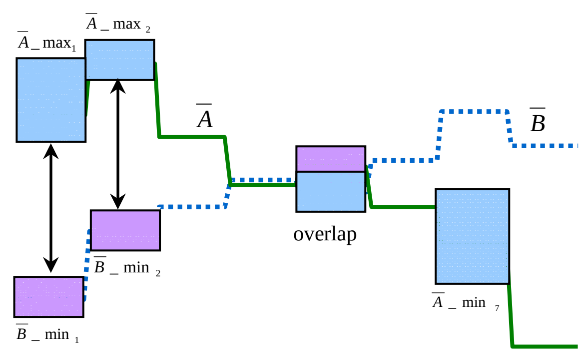
Similar to finding PAA using the mean; we can use the min and max instead:

$$\overline{C}\_\max_i = \max\left(C_{\frac{n}{N}(i-1)+1},\cdots,C_{\frac{n}{N}(i)}\right) \qquad \overline{C}\_\min_i = \min\left(C_{\frac{n}{N}(i-1)+1},\cdots,C_{\frac{n}{N}(i)}\right)$$




**Lower bound function:** The length of the arrows, squared and summed, times the root of the compression ratio now lower bounds circular shifts of D(A,B).

$$LB\_ED\_rotation(\overline{A},\overline{B}) = \sqrt{\sum_{i=1}^{n}\frac{n}{N}\begin{cases}(\overline{A}\_\min_i - \overline{B}\_\max_i)^2, & if \quad \overline{A}\_\min_i > \overline{B}\_\max_i \\ (\overline{B}\_\min_i - \overline{A}\_\max_i)^2, & if \quad \overline{B}\_\min_i > \overline{A}\_\max_i \\ 0 & otherwise \quad (overlap)\end{cases}}$$



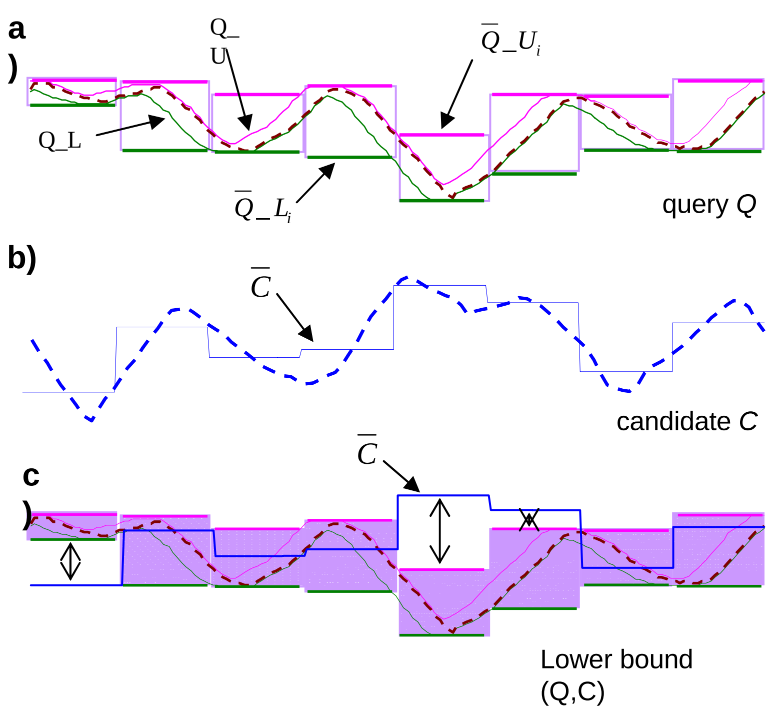
# Lower Bounding for Rotation Invariant using DTW

Similarly, we can use the min and max instead on the bounding envelopes:

$$\overline{C}\_U_i = \max(\overline{C}_{i-r}, \cdots, \overline{C}_{i+r}) \qquad \overline{C}\_L_i = \min(\overline{C}_{i-r}, \cdots, \overline{C}_{i+r})$$

**Lower bound function:** The length of the arrows, squared and summed, times the root of the compression ratio now lower bounds circular shifts of D(A,B).

$$LB\_DTW\_rotation(\overline{Q},\overline{C}) = \sqrt{\sum_{i=1}^{N} \frac{n}{N} \begin{cases} (\overline{C}-\overline{Q}\_U_i)^2, & if \quad \overline{C} > \overline{Q}\_U_i \\ (\overline{C}-\overline{Q}\_L_i)^2, & if \quad \overline{C} < \overline{Q}\_L_i \\ 0 \quad otherwise & (between\_U\_\&\_L) \end{cases}}$$



# Iterative Deepening Rotation Invariant Shape Matching

Best_so_far = inf;

Find approximations of the target shape using different compression ratio, from coarse to fine, until it's the true sequence

**For** every candidate in the database

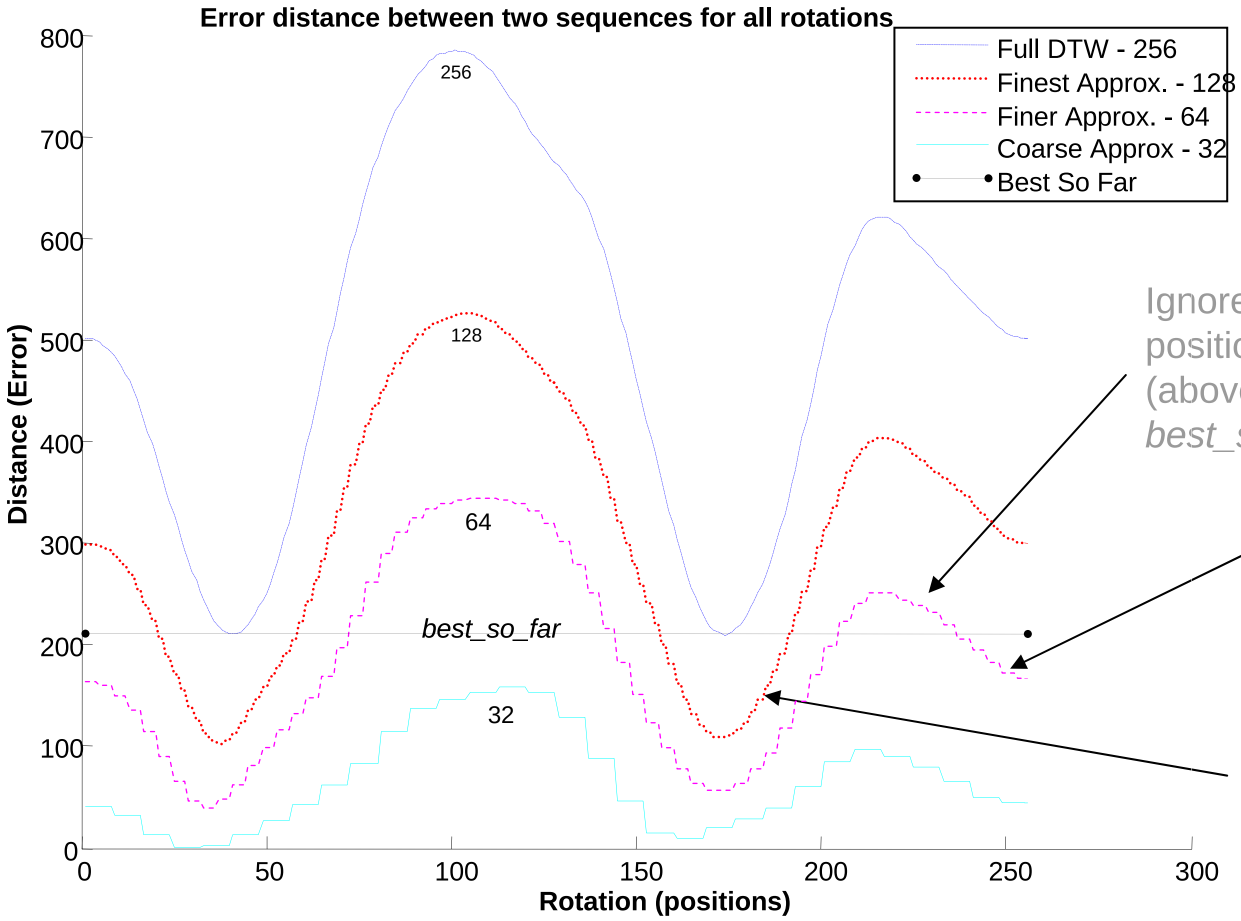
1. Coarse approximation candidate shape
2. Find minimum lowerbounding distance between the approximated target and candidate, for all rotations
3. If result > best_so_far (result not promising)

   -- prune the candidate from further consideration

4. Otherwise, we continue to a finer approximation of the data, go to step 2

**End**

# Optimizing the Iterative Approach




Once we know that at a certain position, the lowerbound, at a coarse level, is greater than the best_so_far, we don't need to consider these corresponding positions anymore, as we continue to a finer level

Ignore these positions (above *best_so_far*)

Search these positions that is below *best_so_far*

Need to check their DTW distance for the red points (positions) that is below *best_so_far*

# Experimental Setup and Results

- Perform experiments on two publicly available datasets
  - 1024 samples per sequence
  - Leaves: 15 classes, 1125 instances
  - Chicken: 5 classes, 446 instances
- Levels of compression: $2^5, 2^6, \ldots, 2^{10}$
- Leave-one-out cross validation, 1-nearest neighbor

| Dataset | Size of database | Accuracy | | Average time per query (minutes) | | | |
|---|---|---|---|---|---|---|---|
| | | Accuracy: Euclidean Distance | Accuracy: DTW | Euclidean Distance (brute-force) | Iterative Deepening Euclidean Distance | DTW (brute-force) | Iterative Deepening DTW |
| **15-class** | 1125 | 86.7% | 89.2% | 0.329 (20s) | 0.068 (4s) | 192.18* | 0.293 (18s) |
| **5-class** | 446 | 80.0% | 80.0% | 0.092 (6s) | 0.031 (2s) | 72.78 | 0.267 (16s) |

*Randomly sampled targets for retrieval due to time constraints (40%)

# Observations

- 70% of the computational time (for iterative approach) is spent on the initial minimum distance calculation
  - Once we have a best_so_far value, it acts as a catalyze for quickly pruning out other irrelevant choices
- The lowerbound for filtering out poor matches becomes even tighter as more sequences are observed.
- As the size of the database increases, our iterative method will achieve a greater speedup

# Conclusion

- Introduce novel framework for speeding up rotation invariant shape matching
- Took the slow but accurate approach and speeded up by 3-4 orders of magnitude, depending on the size of database
- And it guarantee no false dismissals
- Future works include a more detailed analysis this work, with datasets, and extending to other similarity search problems

# References

[1] T. Adamek, N. O'Connor, A multiscale representation method for nonrigid shapes with a single closed contour, IEEE Trans. Circuits Syst Video Techn 14(5): 742-753, 2004.

[2] S Antani, D. Lee, L. Long, G. Thoma, Evaluation of shape similarity measurement methods for spine x-ray images, J. of Vis Comm and Image Rep, 15(3): 285-302, 2004.

[3] W. Zhang, S. Dickinson, J. Feldman, S. Dunn and S. Sclaroff, Shape-based indexing in a medical image database, IEEE Workshop on Biomedical Image Anal., pp 221-230, 1998.

[4] The ADIAC project, http://rbg-web2.rbge.org.uk/ADIAC/

[5] A. Cardone, S. K. Gupta, M. Karnik, A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications, J. Comput. Inf. Sci. Eng. 3(2): 109-118, 2003.

[6] S. Loncaric, A survey of shape analysis techniques, Pattern Recognition, 31(8):983-1001, 1998.

[7] S. Chu, E. Keogh, D. Hart, M. Pazzani, Iterative Deepening Dynamic Time Warping for Time Series, SDM 2002.

[8] Y. Sakurai, M. Yoshikawa, C. Faloutsos: FTW: Fast Similarity Search under the Time Warping Distance. PODS 2005

[9] S. Salvador, P. Chan, FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space, KDD Workshop on Mining Temporal and Sequential Data, pp. 70-80, 2004.

[10] N. Adams, D. Marquez, G. Wakefield, Iterative deepening for melody alignment and retrieval, ISMIR 2005.

[11] M. Agarwal, S. Sarangi, Speech Recognition using Iterative Deepening Dynamic Time Warping, PRASTUTI 2004

[12] L. Rabiner, B.-H. Juang. Fundamentals of Speech Recognition, Prentice-Hall, 1993.

[13] J. B. Kruskall, M. Liberman, The symmetric time warping algorithm: From continuous to discrete, In Time Warps. Addison-Wesley, 1983.

[14] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. ASSP-26(1):43- 49, Feb. 1978.

[15] F. Itakura, Minimum prediction residual principle applied to speech recognition, IEEE Trans. ASSP-23, 67-72, 1975.

[16] E. Keogh, Exact Indexing of Dynamic Time Warping, VLDB 2002: 406-417.

[17] X. Ding, W. Kong, C. Hu, S. Ma, Image Retrieval Using Schwarz Representation of One-Dimensional Feature, VISUAL 1999: 443-450

[18] N. Arica, F. Yarman-Vural, Shape Similarity Measurement for Boundary Based Features, ICIAR 2005: 431-438

[19] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases, Knowl. Inf. Syst. 3(3): 263-286, 2000.

[20] B-K Yi, C. Faloutsos, Fast Time Sequence Indexing for Arbitrary Lp Norms, VLDB 2000: 385-394

[21] E. Keogh, T. Folias, The UCR Time Series Data Mining Archive, http://www.cs.ucr.edu/~eamonn/TSDMA/index.html

[22] D. Li, S. Simske, Shape Retrieval based on Distance Ratio Distribution. HP Tech Report. HPL-2002-251, 2002.

# Thank you

selinach@sipi.usc.edu

http://sipi.usc.edu/~selinachu