DISCA

Pattern Matching Techniques
(*Dynamic Time Warping*)

Deterministic Approach

Dr.Ing. David Cuesta Frau
Polytechnic University of Valencia (Spain)
dcuesta@disca.upv.es

Index

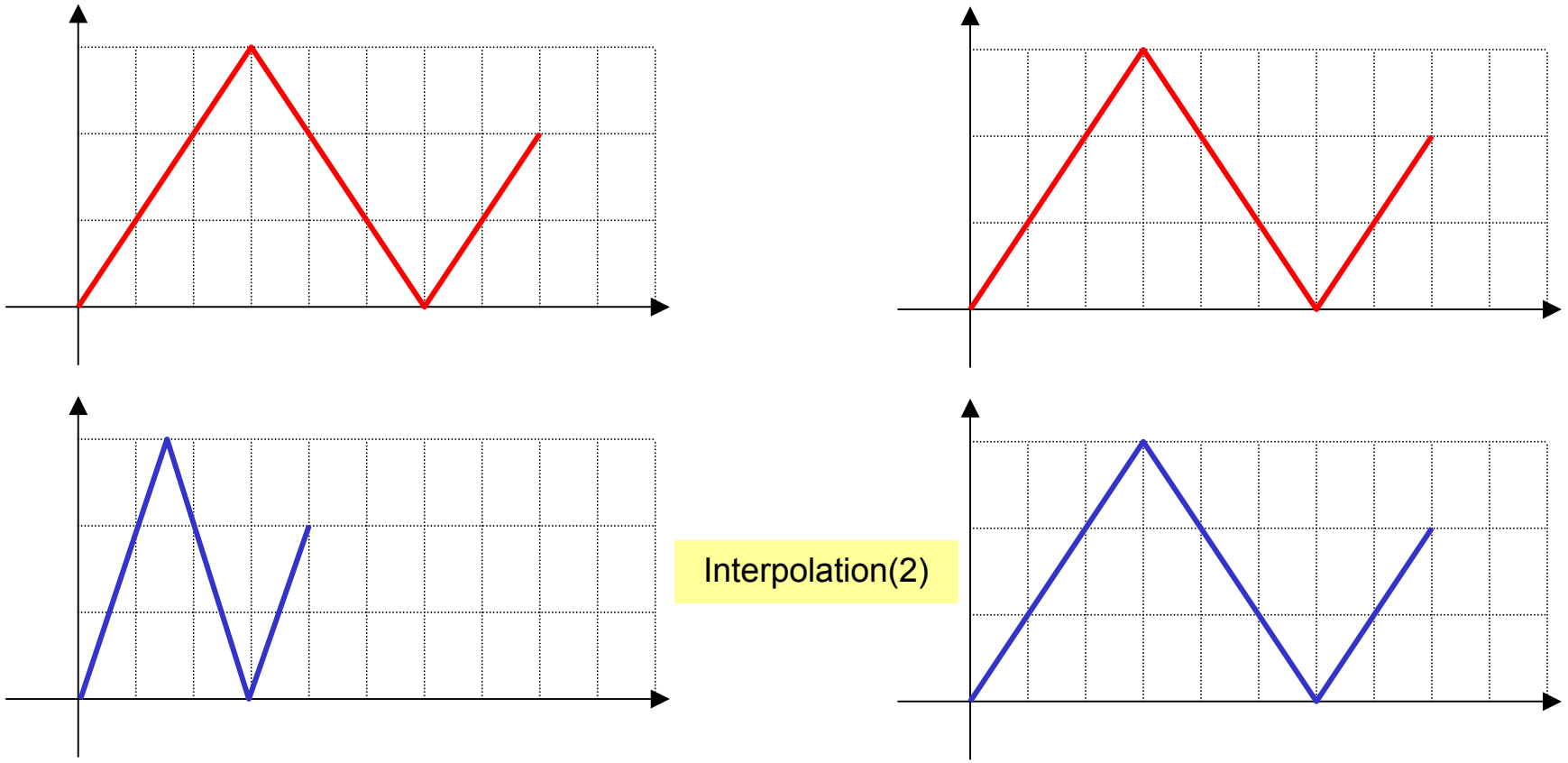
Index

1.- Introduction.

- Common Patterns with Timing Variations:
  - Speech Signals.
  - Growth of Animals.
  - Economics.
  - Biological Signals.
  - Etc.
- Temporal Variability:
  - Uniform: Upsampling, Downsampling.
  - Not Uniform: Phonemes (Vowels ⇔ Consonants).
    - Non-Linear Temporal Alignment.

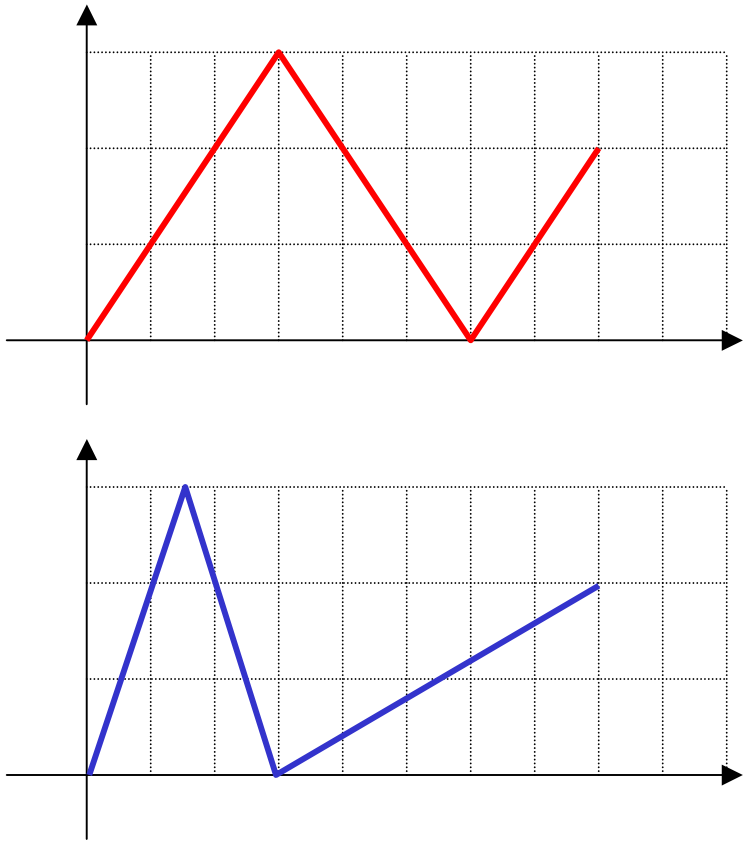
1.- Introduction.

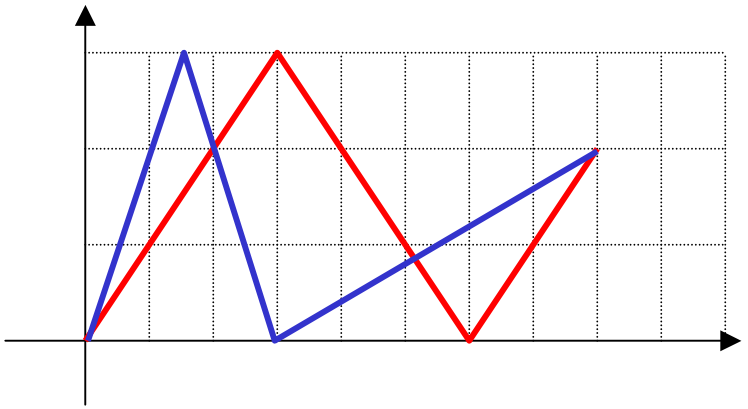
- Example: Linear Temporal Alignment.



1.- Introduction.

- Example: Non-Linear Temporal Alignment.





1.- Introduction.

- Elastic Template-Matching Technique.
- Aimed at finding the “optimal” alignment application between two sequences.
- Local Metric between pairs.
- Dissimilarity measure.
  - Nomenclature:
    - x, y : Sequences.
    - [ ] Discrete Time Vector or Matrix Index.
    - ( ) Continuous Time Vector or Matrix Index.

1.- Introduction.

Definitions:

Objects: x, y

n_x :Length of Object x

n_y :Length of Object y

n_F :Length of Alignment Path

I_x :Definition Interval of Object x

I_y :Definition Interval of Object y

I_F :Definition Interval of Alignment Path

F :Alignment Application

1.- Introduction.

Definitions:

Alignment Axis: i, j

Independent Variable: $k \in I_F$

Path: $F(k) = (i(k), j(k)), i(k) \in I_x\ ,\ \ j(k) \in I_y\ , k \in I_F$

Local Metric: $d(x, y)$

Dissimilarity Measure:

$$\Delta_F = \frac{\int_{I_F} d(x(i(k)), y(j(k)))dk}{\int_{I_F} dk}$$

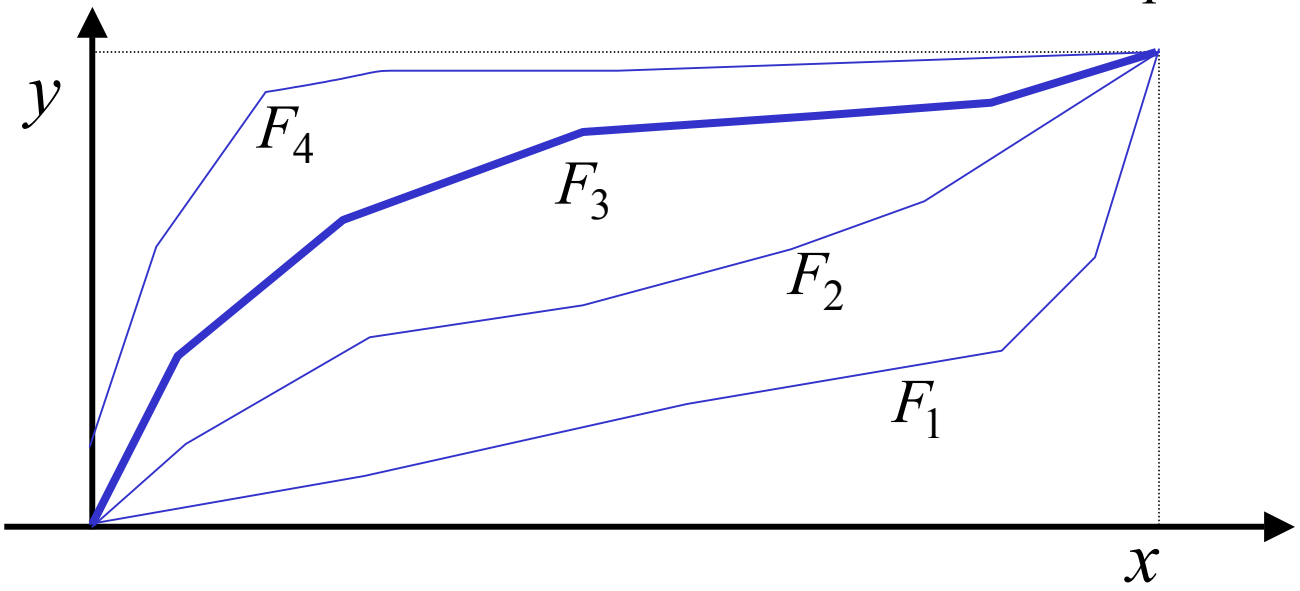
Normalization Factor

1.- Introduction.

Definitions:

Optimal Path: $F_o = \underset{F}{\arg\min}\left(D_F\left(x,y\right)\right)$

Minimal Dissimilarity: $\Delta\left(x,y\right) = \Delta_{F_o}\left(x,y\right) = \underset{F}{\min}\left(\Delta_F\left(x,y\right)\right)$



Index

2.- Discrete Time Formulation.

Input feature sequences x and y.

$$x[n] = \{x[0], x[1], \ldots, x[n_x - 1]\}$$

$$y[n] = \{y[0], y[1], \ldots, y[n_y - 1]\}$$

Path: $F[k] = (i[k], j[k]), 0 \le i[k] < n_x\ ,\ 0 \le j[k] < n_y\ , k \in I_F$

$$\Delta_F = \frac{\sum_{I_F} d(x[i[k]], y[j[k]])}{L(F)}$$

2.- Discrete Time Formulation.

Alignment Plane ⟹ Dynamic Programming Matrix G.

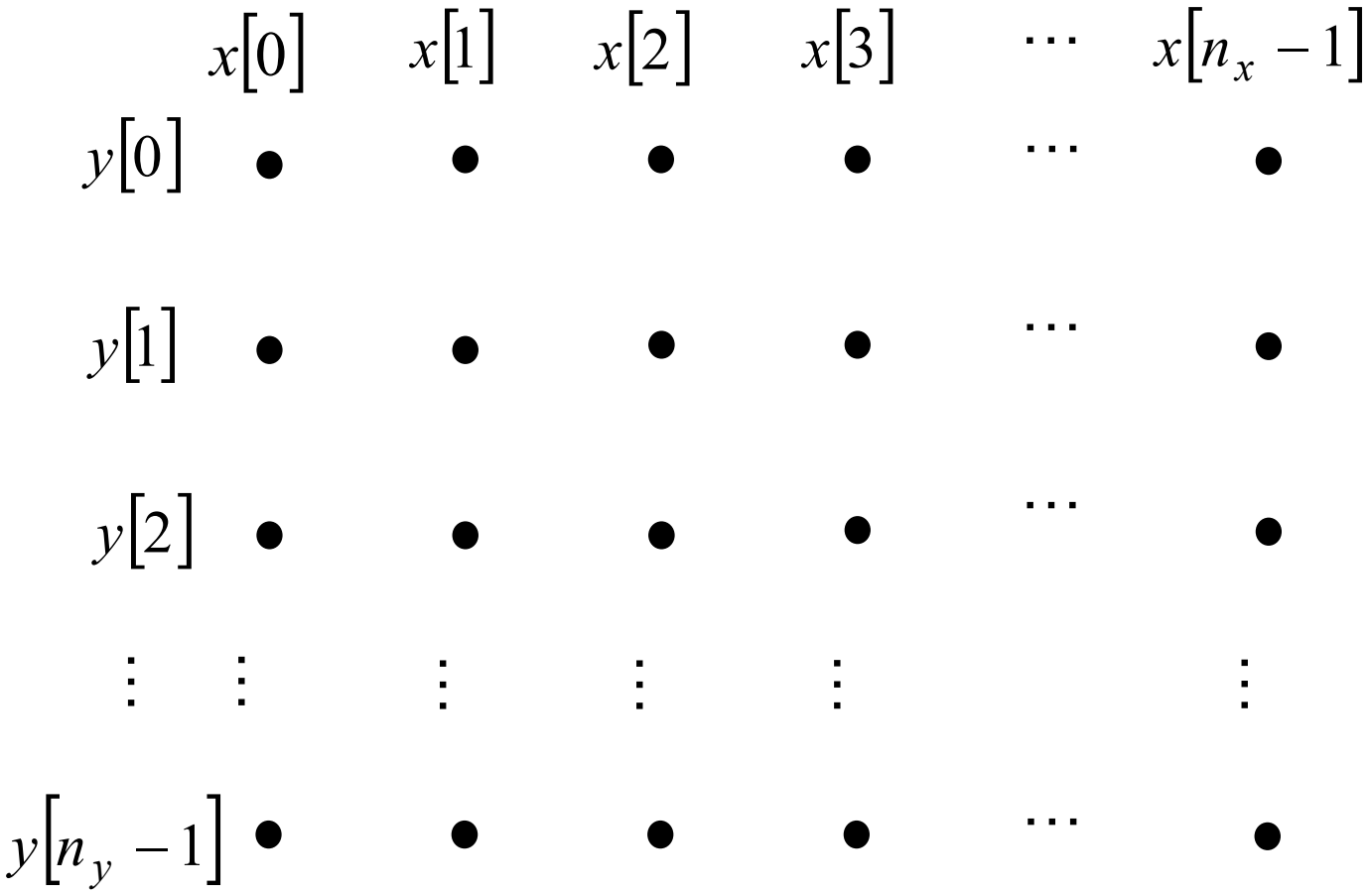
$$G = \{g_{ij}\} = \begin{pmatrix} G[0,0] & G[1,0] & \dots & G[n_x - 1,0] \\ G[0,1] & & \dots & \dots \\ \dots & & \dots & \dots \\ G[0,n_y - 1] & G[1,n_y - 1] & \dots & G[n_x - 1,n_y - 1] \end{pmatrix}$$

Each element of G: Node.

Alignment Path results from joining nodes.

Path Length is ponderated according to some weights .

2.- Discrete Time Formulation.



2.- Discrete Time Formulation.

$$G[i,j] = \min_{(a,b)\in P} \left(G[i-a, j-b] + d(x[i], y[j]) w(a,b) \right)$$

Node Value

Productions

Accumulated Cost

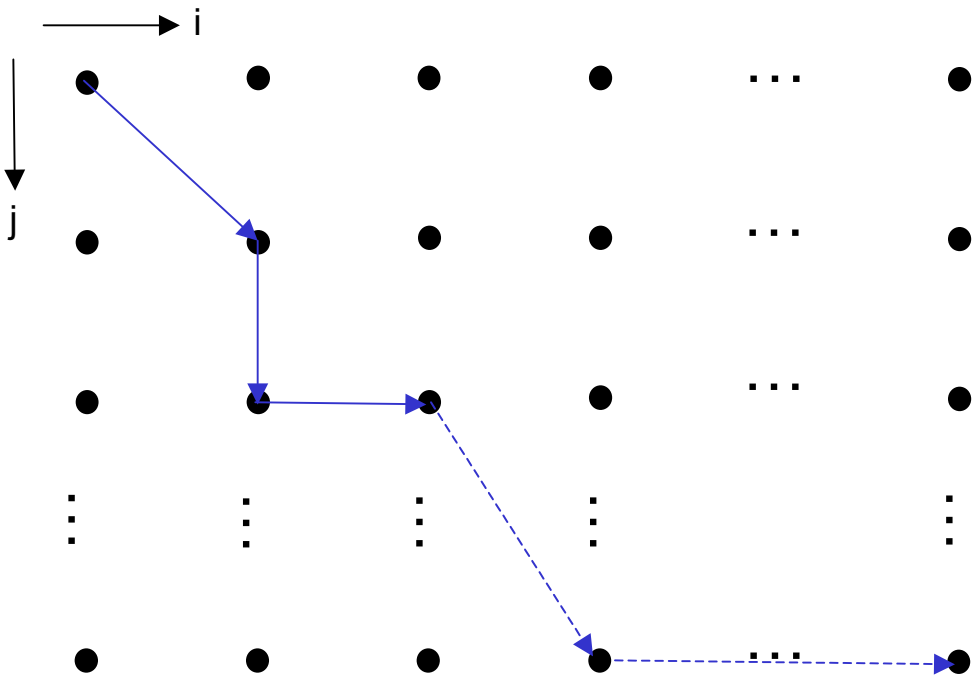
Node cost

Weights

Index

3.- The Alignment Path.

$$F = \{(i[0], j[0]), (i[1], j[1]), \ldots, (i[k], j[k]), \ldots, (i[K-1], j[K-1])\}$$

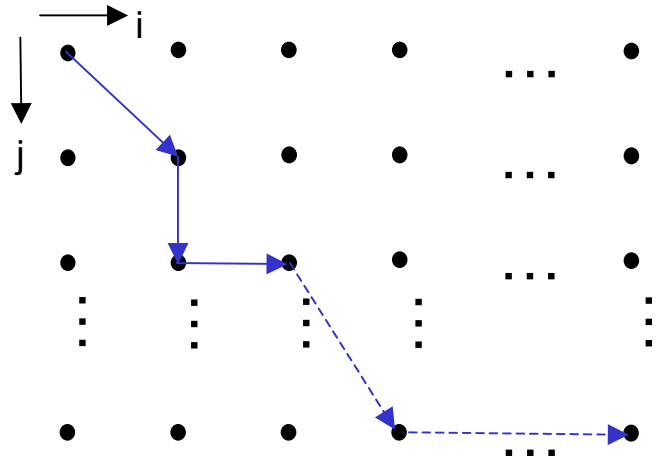


Cost:

$$O(n_x n_y)$$

3.- The Alignment Path.

Weighted Directed Path: Arcs



Insertion : $\{G[i,j], G[i,j+1]\} \rightarrow w(0,1)$

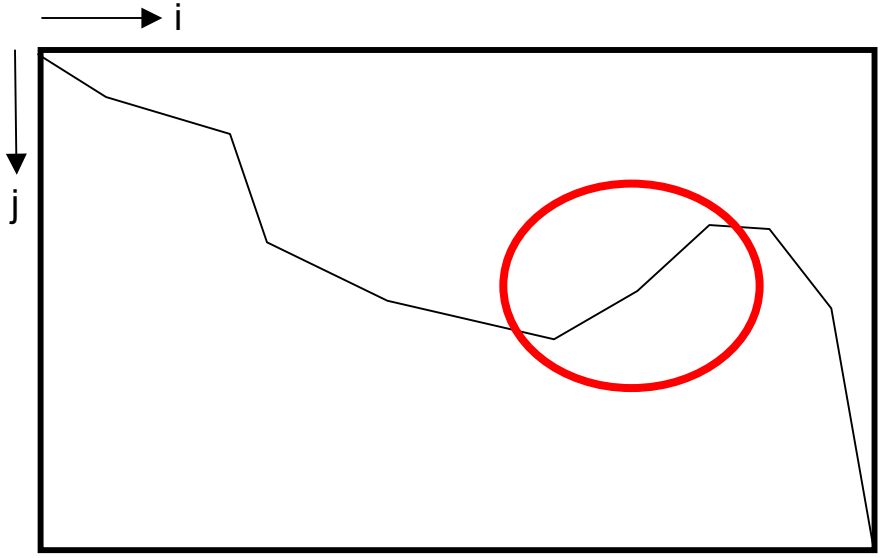
Deletion : $\{G[i,j], G[i+1,j]\} \rightarrow w(1,0)$

Change or Substitution : $\{G[i,j], G[i+1,j+1]\} \rightarrow w(1,1)$

3.1.- Properties.

1.- Monotonic : $i[k] \leq i[k+1], j[k] \leq j[k+1]$

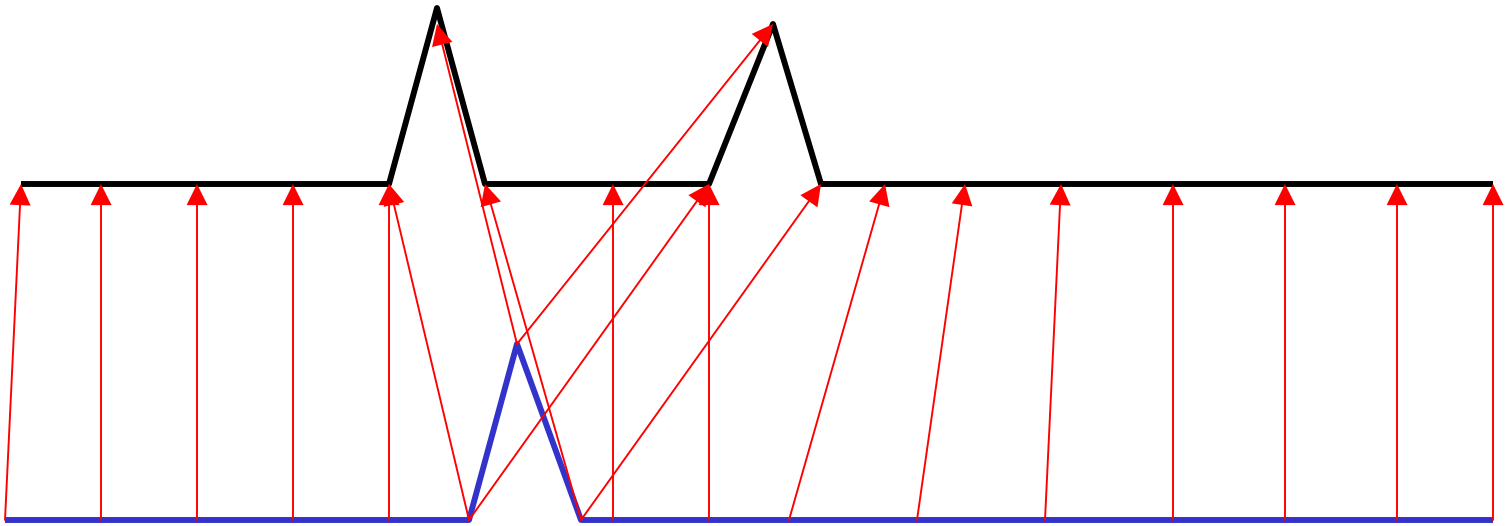
The alignment path should not go back in *"time"* index:



otherwise...

3.1.- Properties.

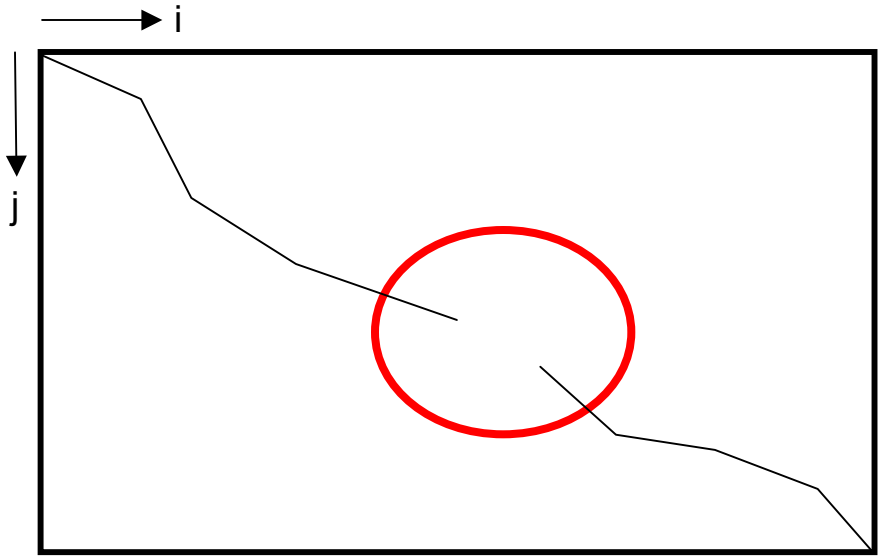
- Some features could be repeated, and therefore, the dissimilarity would not be meaningful.



3.1.- Properties.

2.- Continuity : $i[k+1]-i[k]\leq 1, j[k+1]-j[k]\leq 1$

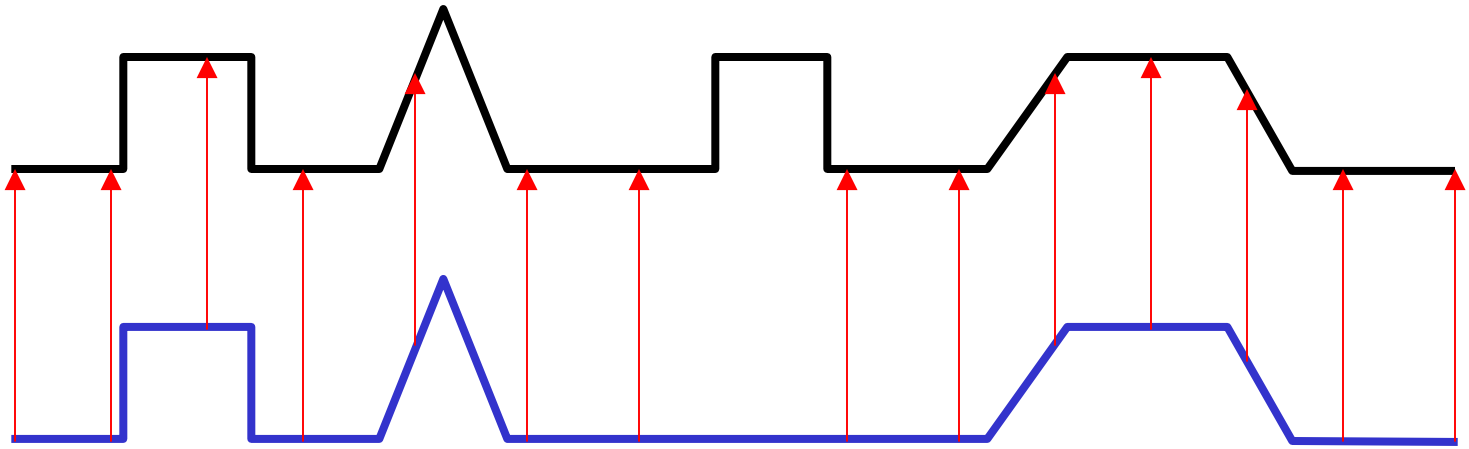
The alignment path should not jump:



otherwise...

3.1.- Properties.

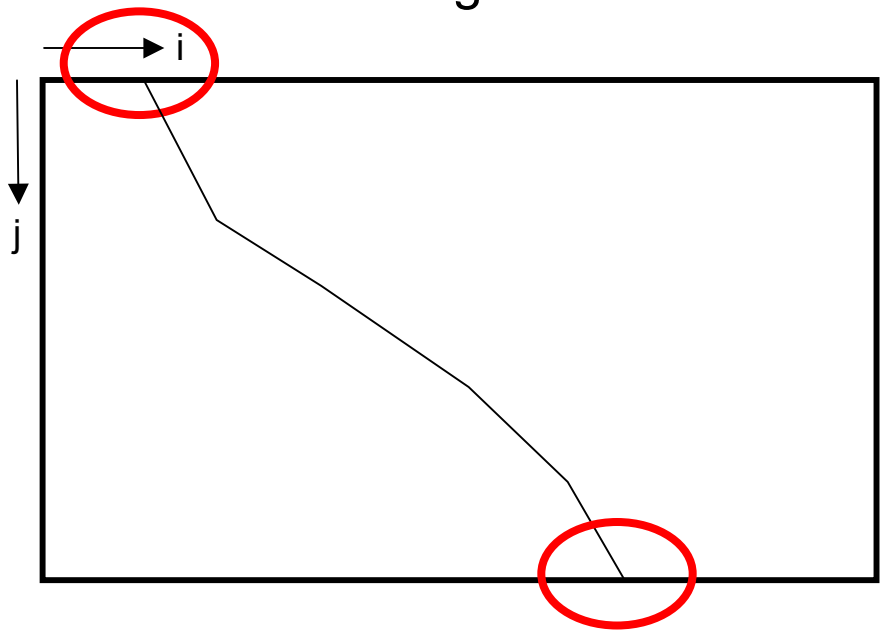
- The alignment could omit important features, and therefore, the dissimilarity would not be meaningful.



3.1.- Properties.

3.-Boundary: $i[0] = j[0] = 0, i[K-1] = I-1, j[K-1] = J-1$

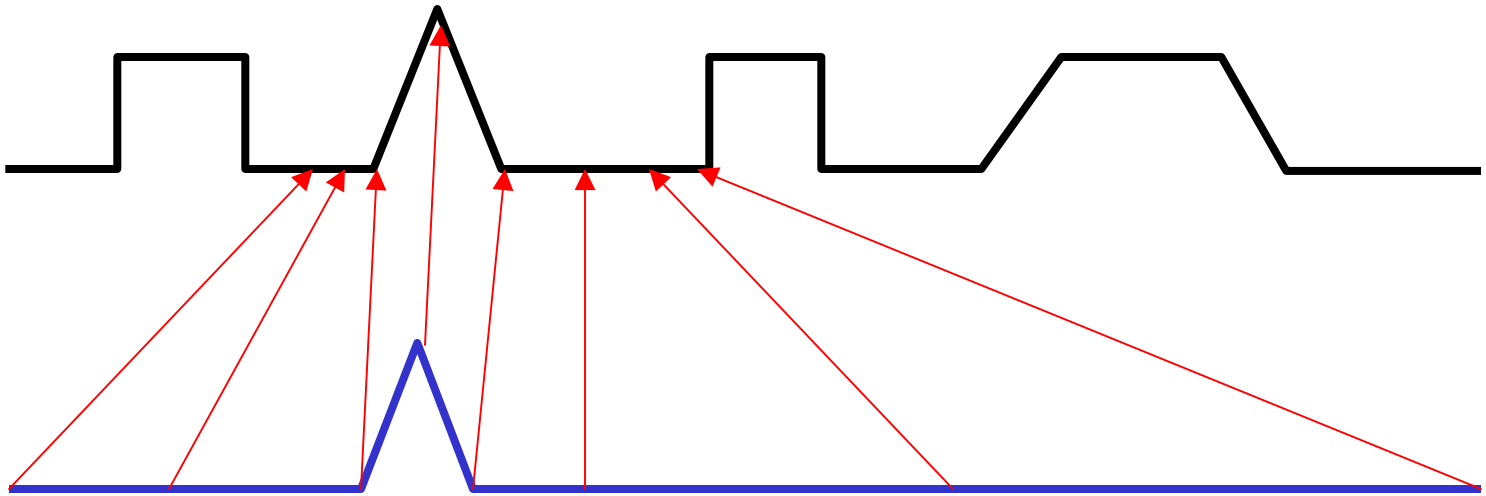
The alignment path must start at the upper left corner,and finish at the lower right corner.



otherwise...

3.1.- Properties.

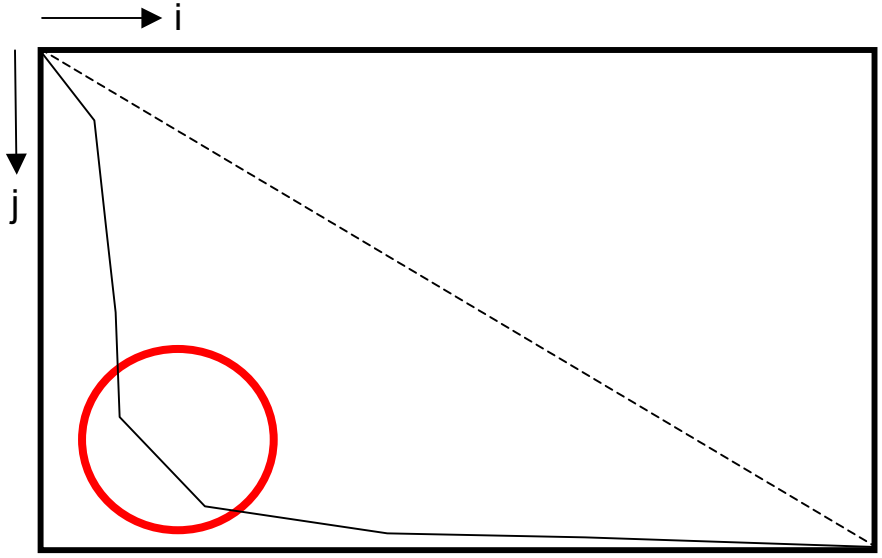
- The alignment could consider partially one of the sequences, and therefore, the dissimilarity would not be meaningful.



3.1.- Properties.

4.- Window : $|i[k] - j[k]| \leq \delta \, , \delta \in \Re^{+}$

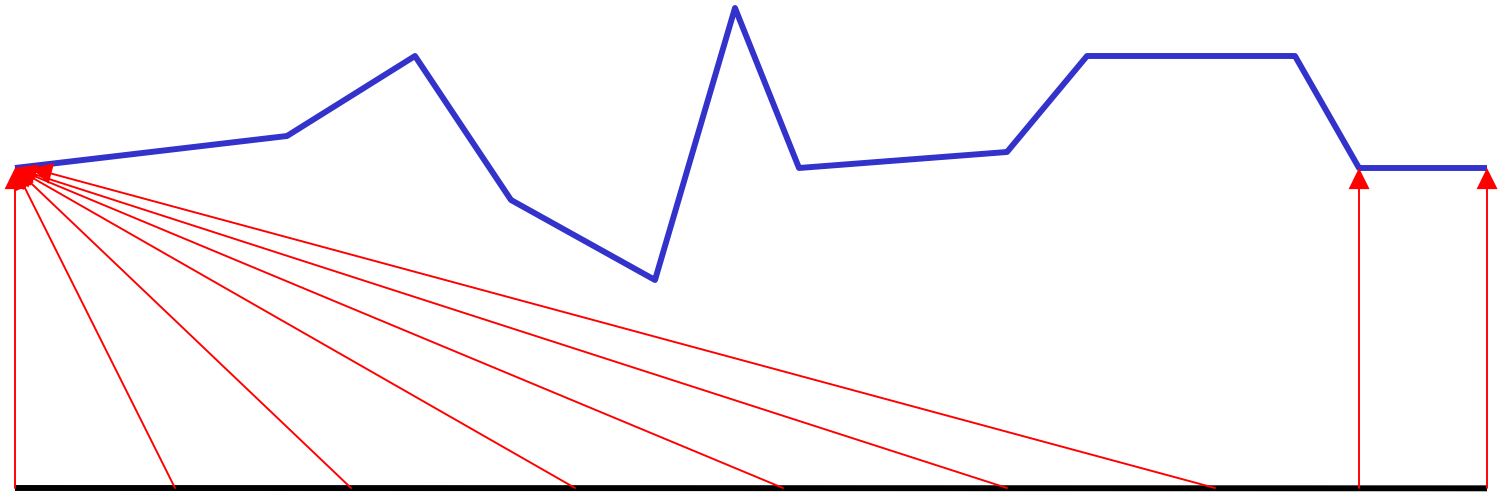
The alignment path must be *"relatively close"* to the identity path.



otherwise...

3.1.- Properties.

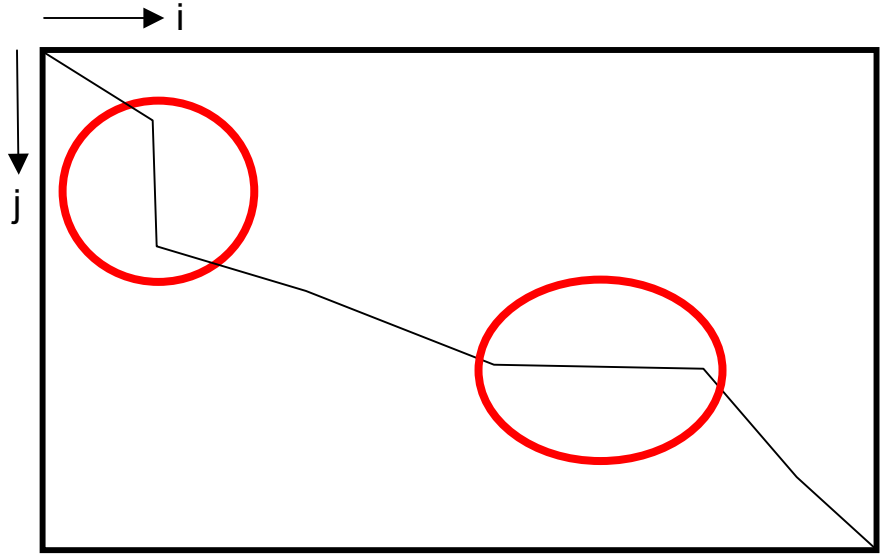
- The alignment tries to skip different features, and gets stuck at similar features, and therefore, the dissimilarity would not be meaningful.



3.1.- Properties.

$$5.\text{- Slope}: \frac{j[k_1]-j[k_0]}{i[k_1]-i[k_0]} \le \delta\ ,\ \delta \ge 0, \frac{i[k_1]-i[k_0]}{j[k_1]-j[k_0]} \le \gamma\ ,\ \gamma \ge 0$$

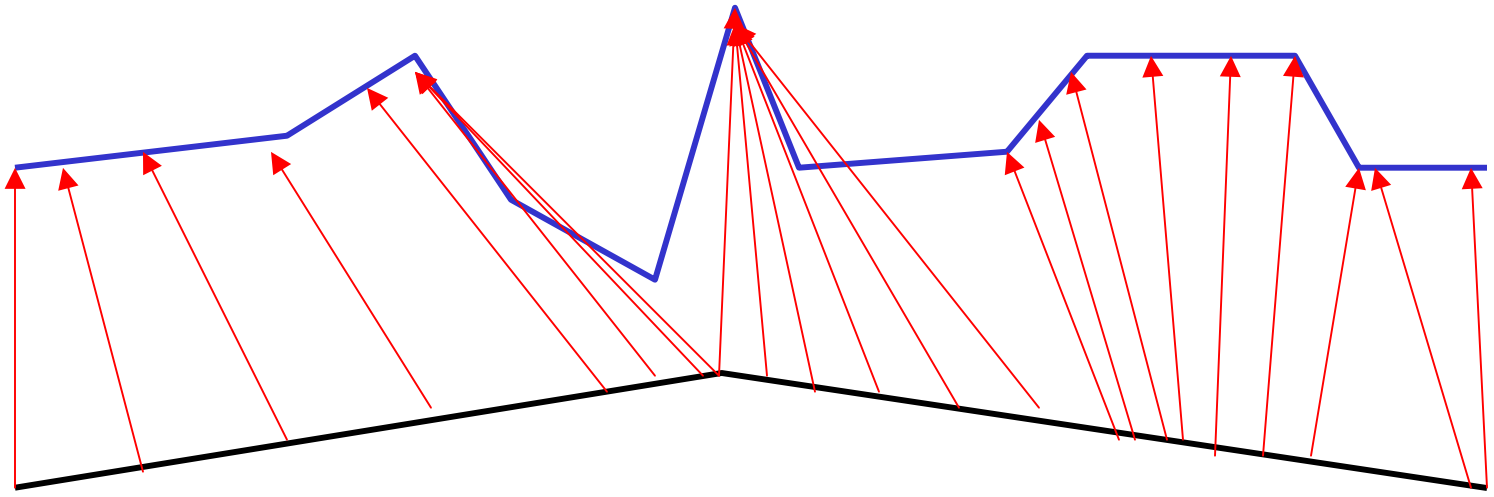
The alignment path must be neither too steep nor too gentle.



otherwise...

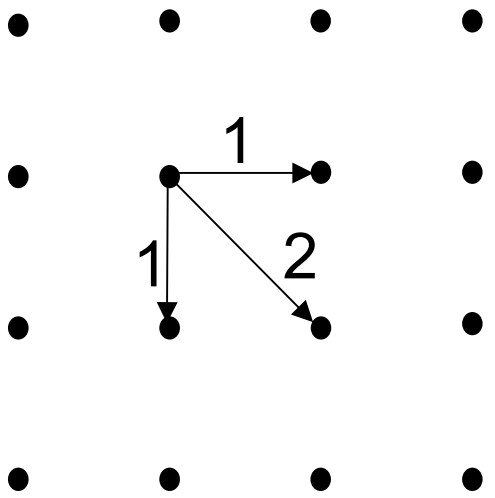
3.1.- Properties.

- Similar to window property, but locally. Diagonal path corresponds to linear alignment. The further we are from the diagonal, the more different sequences are. Very long parts are matched with very short parts.



3.2.- Constraints.

- Local Constraints:



Constraint DP1

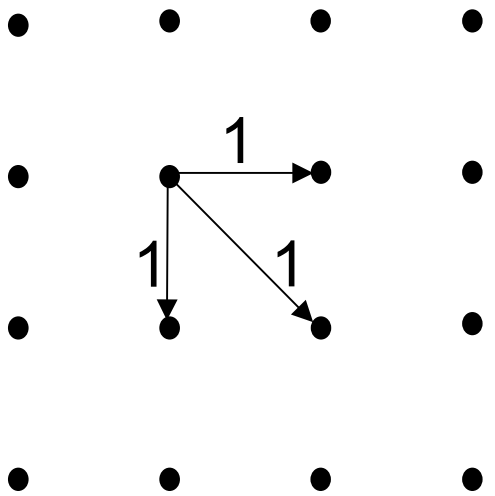
$$G[0,0] = 2d[0,0]$$

$$G[i,j] = \min\begin{cases} G[i,j-1] + d[i,j] \\ G[i-1,j-1] + 2d[i,j] \\ G[i-1,j] + d[i,j] \end{cases}$$

$$\Delta(x,y) = \frac{G[n_x - 1, n_y - 1]}{n_x + n_y}$$

3.2.- Constraints.

- Local Constraints:



Constraint DP2

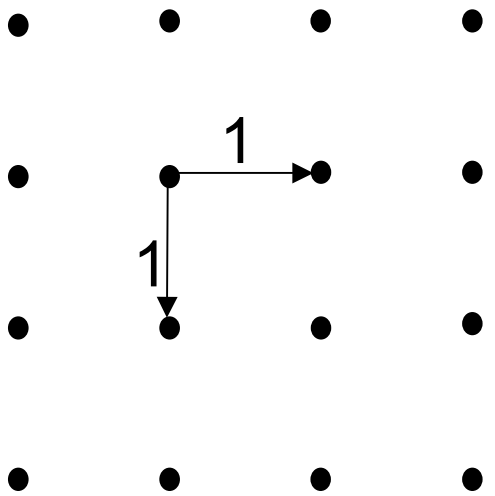
$$G[0,0] = d[0,0]$$

$$G[i,j] = \min \begin{cases} G[i, j-1] + d[i,j] \\ G[i-1, j-1] + d[i,j] \\ G[i, j-1] + d[i,j] \end{cases}$$

$$\Delta(x,y) = \frac{G[n_x - 1, n_y - 1]}{n_x + n_y}$$

3.2.- Constraints.

- Local Constraints:



Constraint DP3

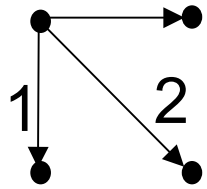
$$G[0,0] = 2d[0,0]$$

$$G[i,j] = \min\begin{cases} G[i,j-1] + d[i,j] \\ G[i-1,j] + d[i,j] \end{cases}$$

$$\Delta(x,y) = \frac{G[n_x - 1, n_y - 1]}{n_x + n_y}$$

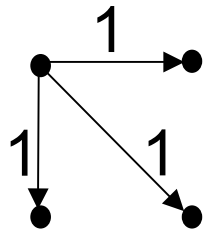
3.2.- Constraints.

- Properties Derived from Local Constraints:



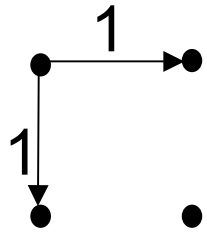
Commutative

Compatibility with Linear Temporal Alignment



Commutative

Temporal Reversibility



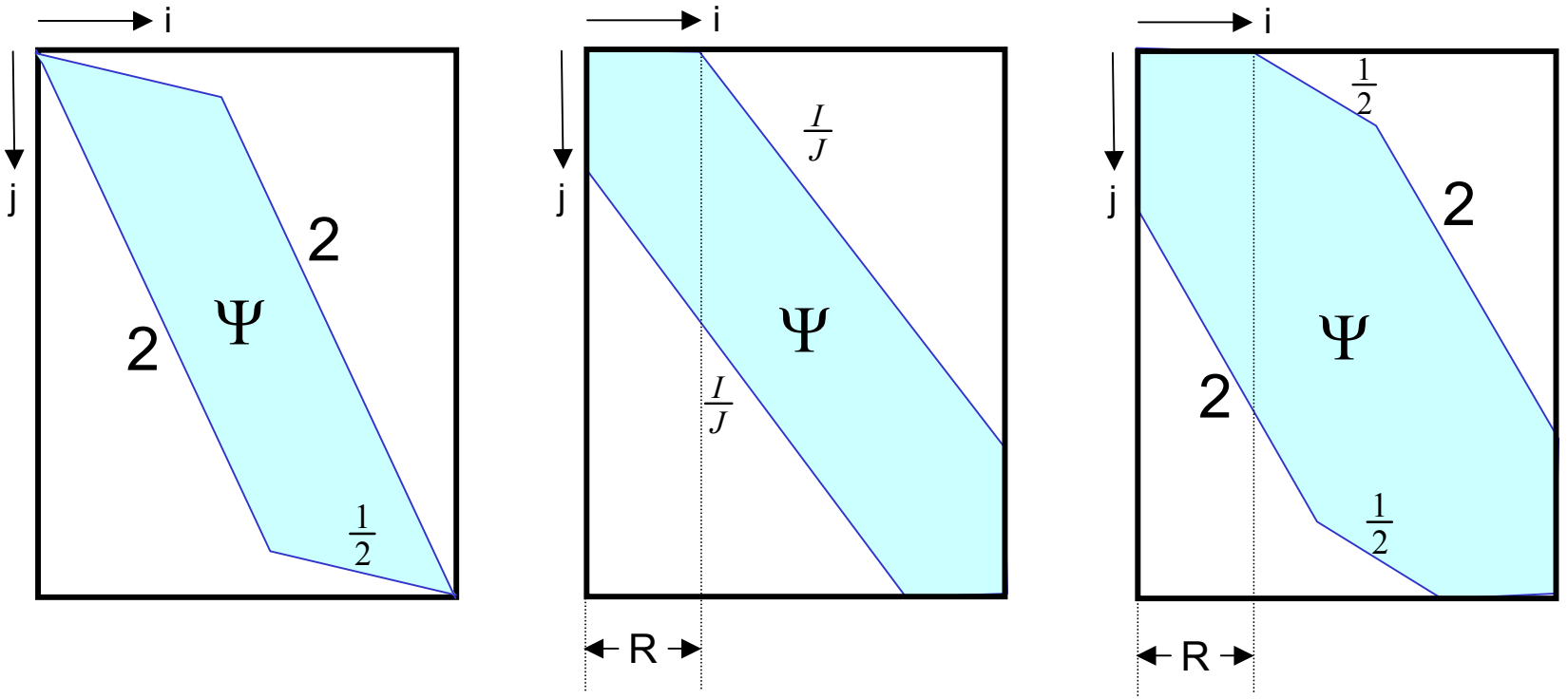
Commutative

Temporal Reversibility

3.2.- Constraints.

- Global Constraints:

$$G[i,j] = \infty \; \forall (i,j) \notin \Psi$$



Index

4.- Local Cost Functions.

$$d(i,j) = \left| x[i] - y[j] \right|$$

$$d(i,j) = x[i]y[j] - \beta P(F)$$

$$d(i,j) = \begin{cases} 0, \text{ if } x[i] = y[j] \\ 1, \text{ Otherwise} \end{cases}$$

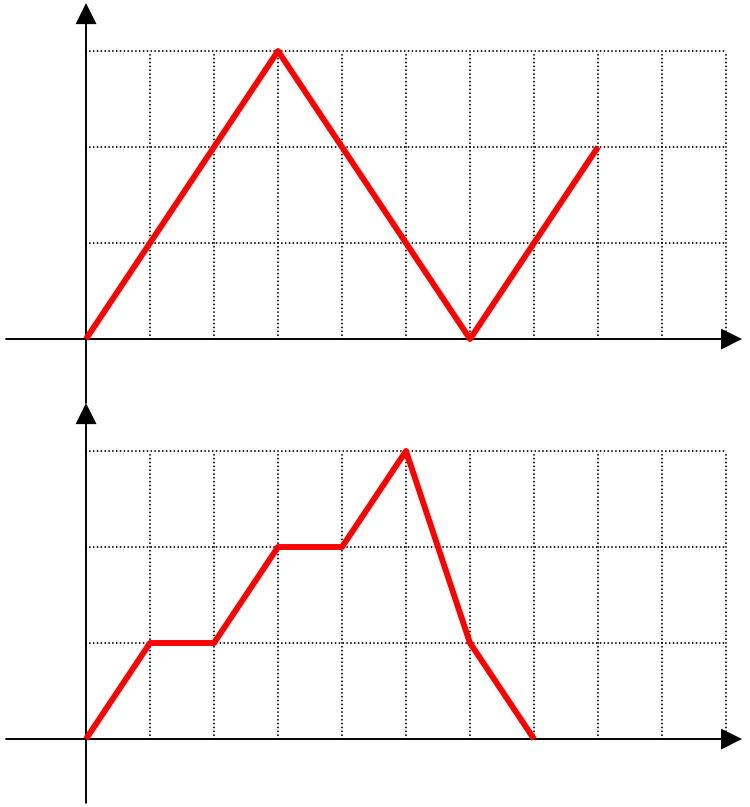
$$d(i,j) = \left| \frac{x[i+1] - x[i-1]}{2} \_ \frac{y[j+1] - y[j-1]}{2} \right|$$

4.- Local Cost Functions.

- Example: Find the alignment path and dissimilarity of the following two sequences, according to the issues explained so far.

$$x[n] = \{0,1,2,3,2,1,0,1,2\}$$

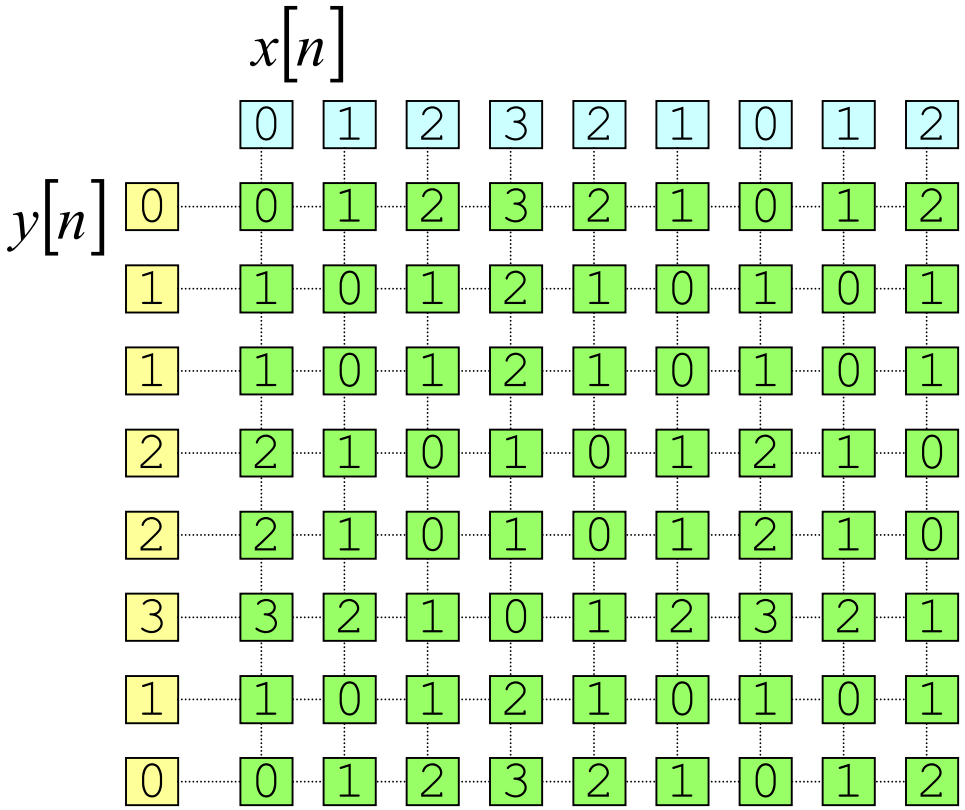
$$y[n] = \{0,1,1,2,2,3,1,0\}$$



4.- Local Cost Functions.

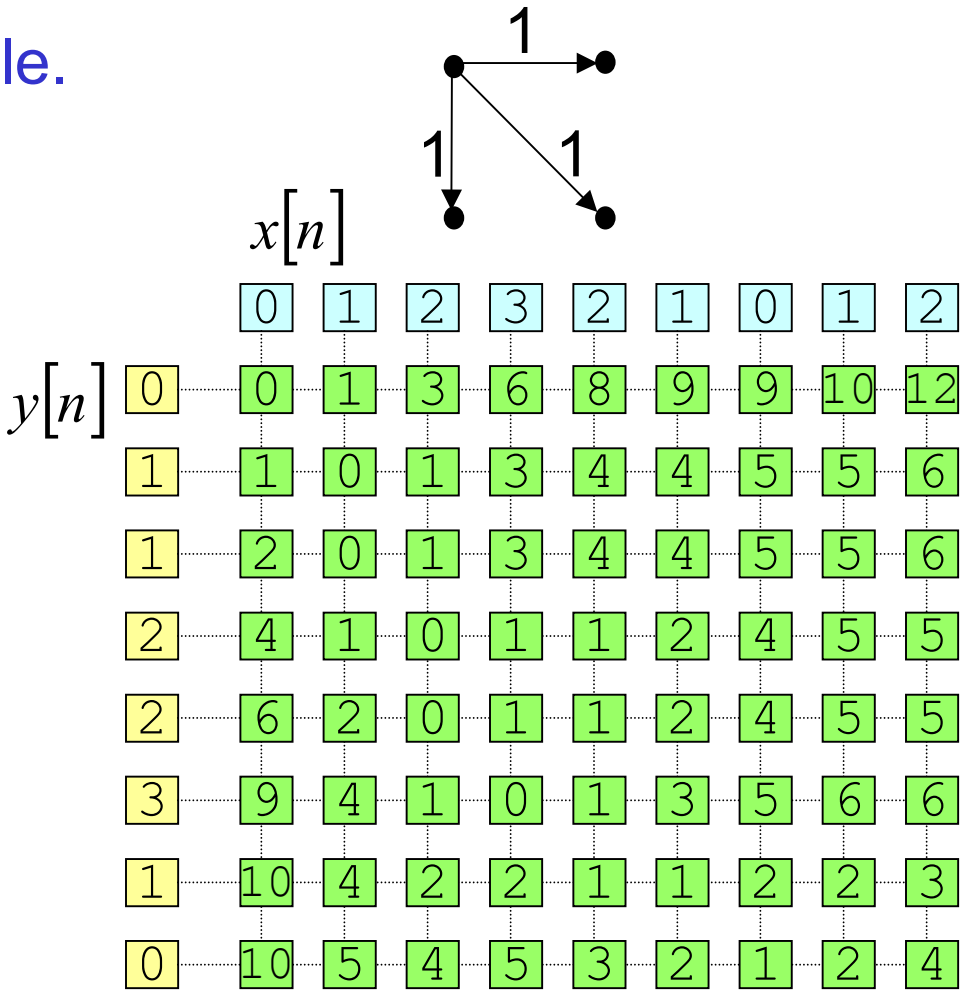
- Example.

$$d(i,j) = |x[i] - y[j]|$$



4.- Local Cost Functions.

- Example.

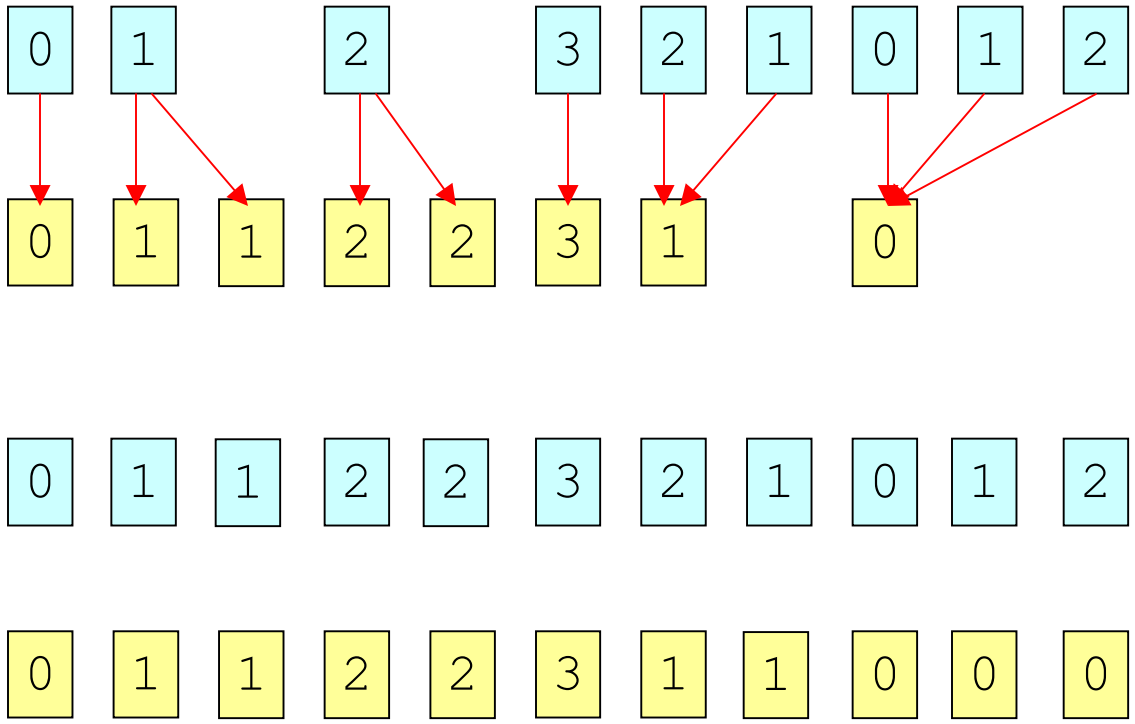


4.- Local Cost Functions.

- Example. Boundary Constraints.

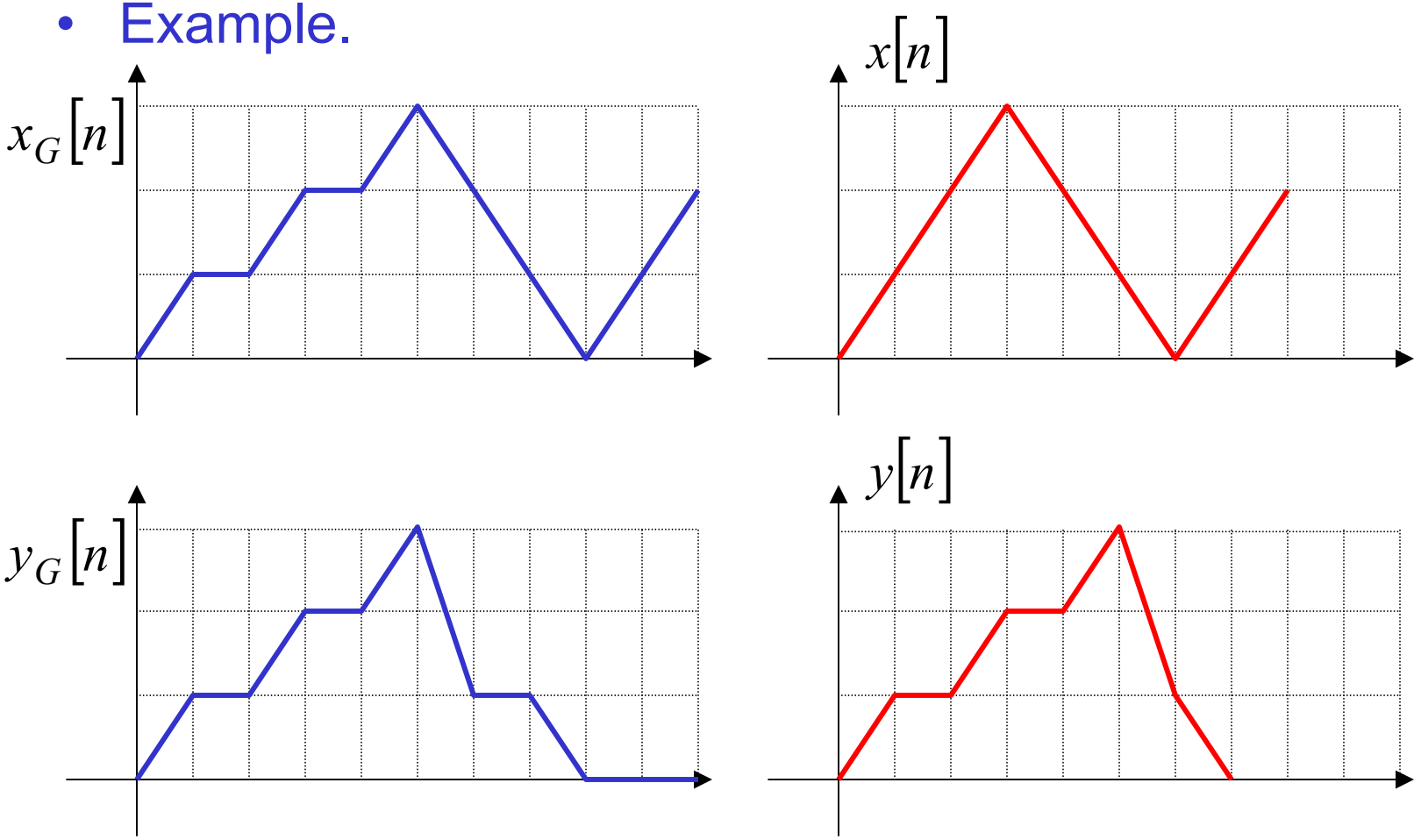
4.- Local Cost Functions.

- Example.



4.- Local Cost Functions.

- Example.



Index

5.- Algorithms.

```
for i=1 to i=n_x-1 do G[i][n_y-1] ← INFINITY  endfor
for j=1 to j=n_y-1 do G[n_x-1][j] ← INFINITY  endfor
```

Global Constraints

```
for i=1 to i=n_x-1 do
    for j=1 to j=n_y-1 do
        NodeCost ← CalculateNodeCost(x,i,y,j)
        Diagonal ← G[i-1][j-1]+w[1][1]*NodeCost
        Horizontal ← G[i-1][j]+w[1][0]*NodeCost
        Vertical ← G[i][j-1]+w[0][1]*NodeCost
        if((Diagonal<=Horizontal)AND(Diagonal<=Vertical))
                G[i][j] ← Diagonal
                elseif ((Horizontal<=Diagonal)AND(Horizontal<=Vertical))
                        G[i][j] ← Horizontal
                else G[i][j] ← Vertical
        endif
    endfor
endfor
```

Basic Algorithm

5.- Algorithms.

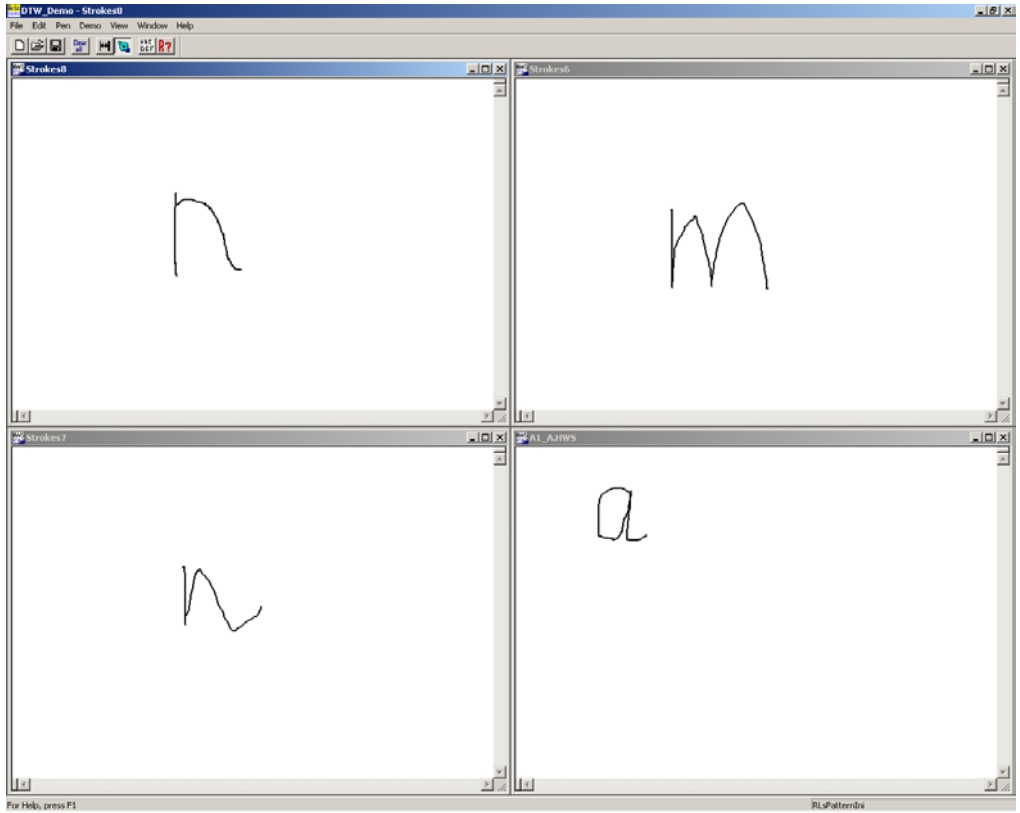
```
for i=1 to i=n_x-1 do
    for j=1 to j=n_y-1 do
        if (G[i][j]!=Infinity)
                NodeCost ← CalculateNodeCost(x,i,y,j)
                Diagonal ← G[i-1][j-1]+w[1][1]*NodeCost
                Horizontal ← G[i-1][j]+w[1][0]*NodeCost
                Vertical ← G[i][j-1]+w[0][1]*NodeCost
                if((Diagonal<=Horizontal)AND(Diagonal<=Vertical))
                        G[i][j] ← Diagonal
                elseif ((Horizontal<=Diagonal)AND(Horizontal<=Vertical))
                        G[i][j] ← Horizontal
                else G[i][j] ← Vertical
                endif
        endif
    endfor
endfor
```

Basic Algorithm with Global Constraints

Index

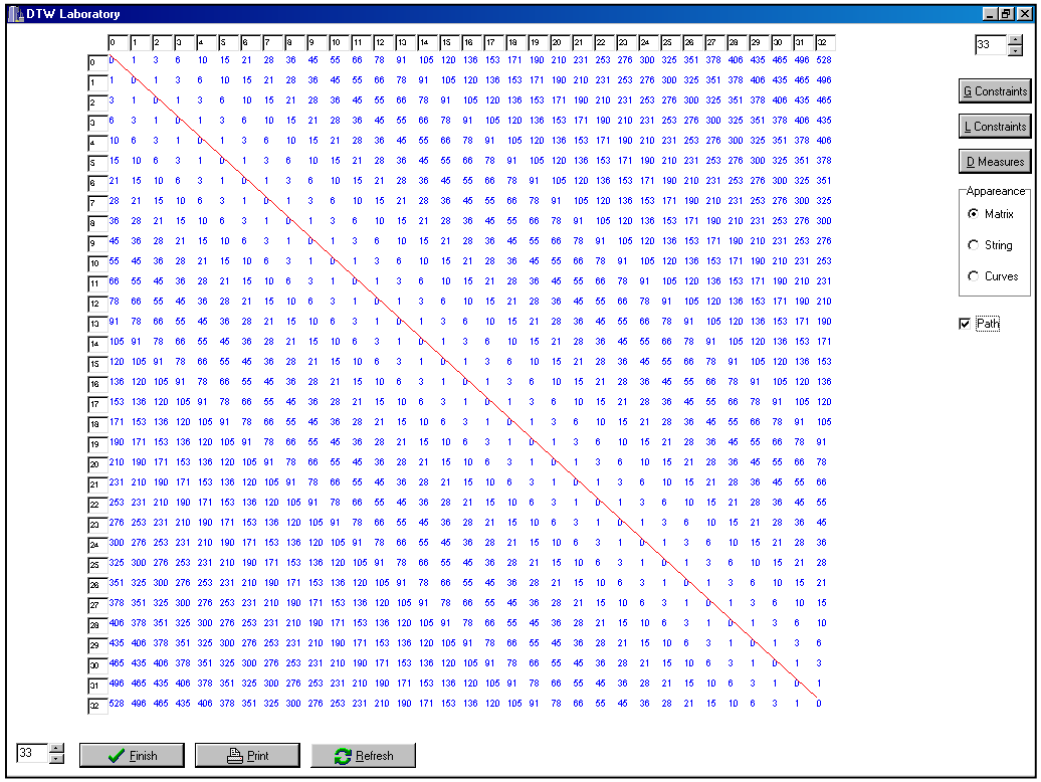
6.- Implementation.

- © Intel Recognition Primitives Library.



6.- Implementation.

- DTW Laboratory.



Index

7.- Decision Rules.

How we use the distance among objects in a set, for classification?

Set of Objects: P

Feature Space: R

Object: x

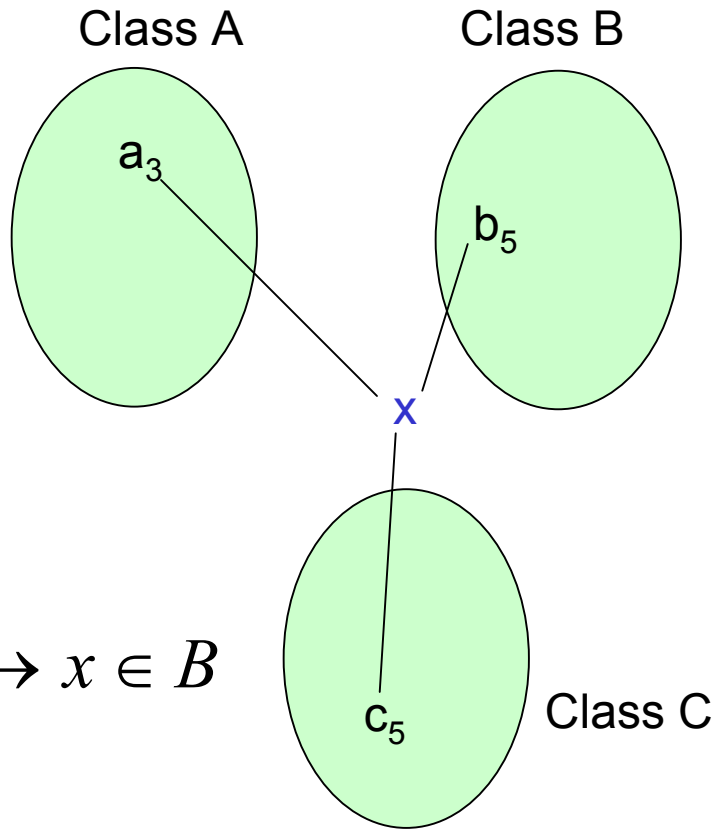
Dissimilarity Measure: $\Delta : R \times R \rightarrow \Re^{\geq 0}$

Space Partition: $\Omega = \{c_1, c_2, ..., c_m\}$

7.- Decision Rules.

- Minimum Distance(MD)

- One prototype per class.



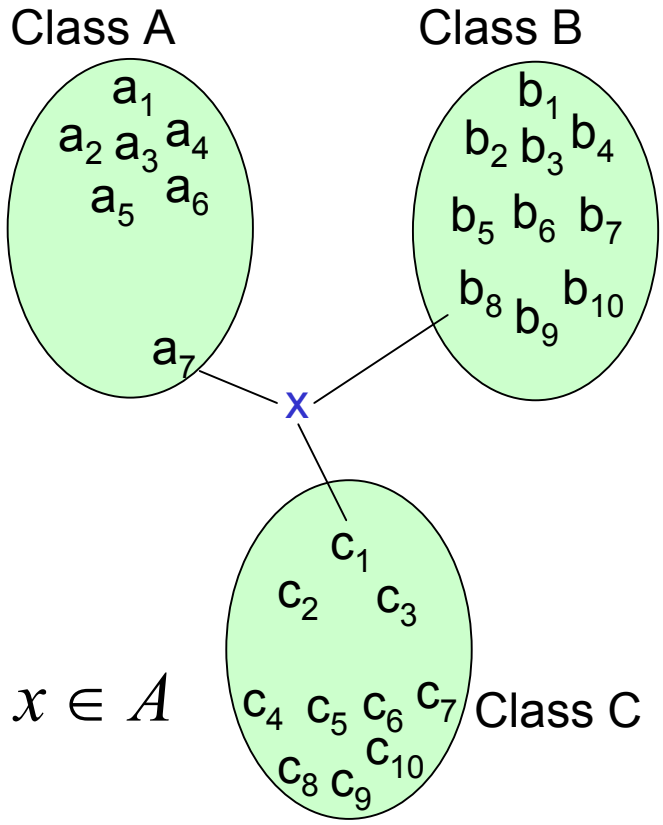
$$\Omega = \{A, B, C\}$$

$$\min\left(\Delta_{DM}(x, \Omega)\right) = \Delta_{DM}(x, b_5) \rightarrow x \in B$$

7.- Decision Rules.

- Nearest Neighbour(NN)

- Some prototypes per class.
- One distance considered.



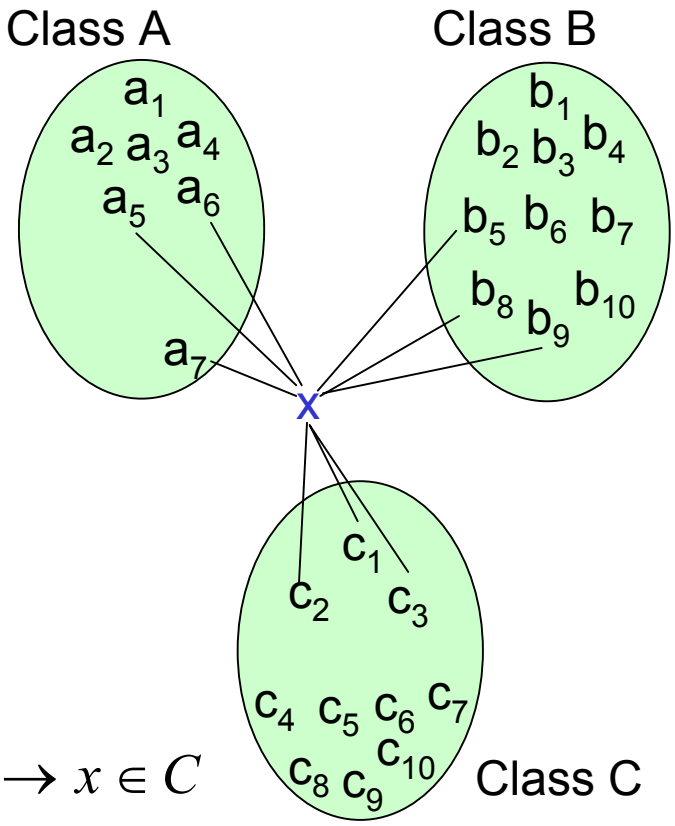
$$\Omega = \{A, B, C\}$$

$$\min\left(\Delta_{NN}(x, \Omega)\right) = \Delta_{NN}(x, a_7) \rightarrow x \in A$$

7.- Decision Rules.

- K-Nearest Neighbour(K-NN)

- Some prototypes per class.
- K Distances considered.



$$\Omega = \{A, B, C\}$$

$$K = 3$$

$$\min\left(\Delta_{K-NN}(x, \Omega)\right) = \Delta_{K-NN}\left(x, \{c_1, c_2, c_3\}\right) \rightarrow x \in C$$

Index

8.- Applications.

- Speech Recognition.
- String Matching.
- Handwritten Character Recognition.
- Object Recognition.
- Prototype Formation.
- Morphing.
- Polygon Recognition.
- Curve Alignment.

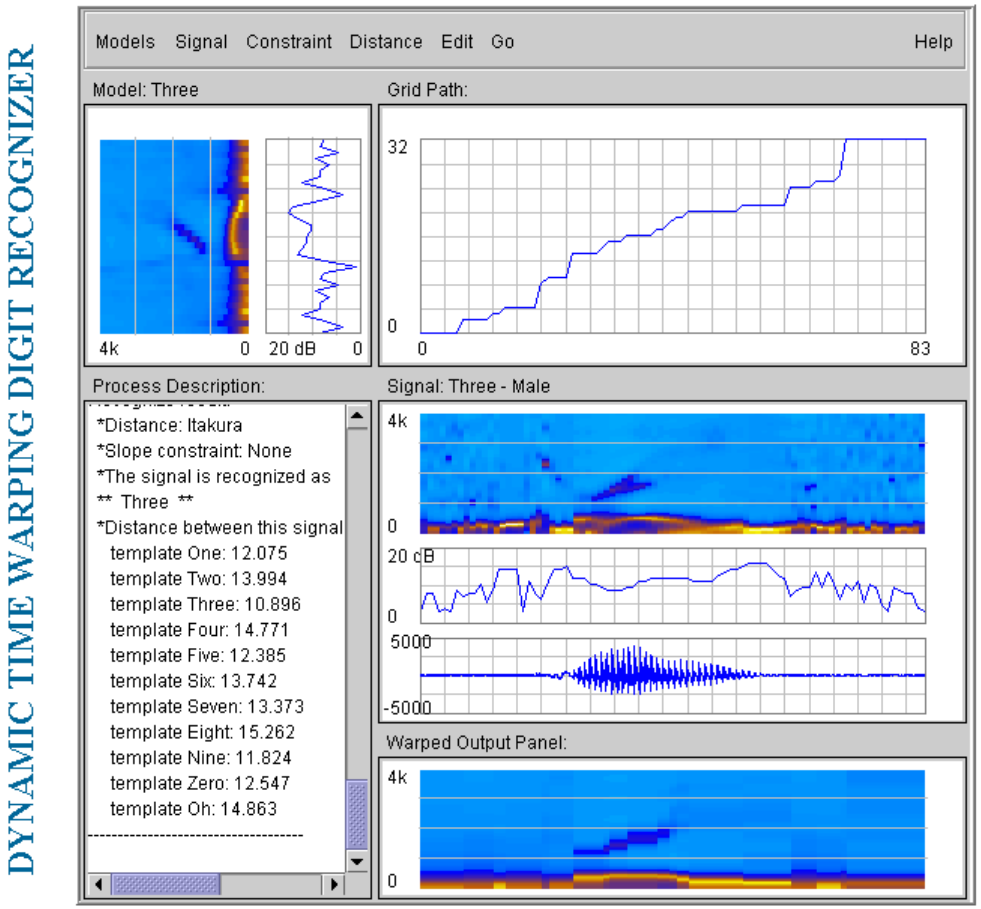
8.- Applications.

- Speech Recognition.
  - For each word in the vocabulary, store at least one reference pattern.
  - Record a spoken word.
  - Apply the feature extraction method.
  - Compute the dissimilarity measure with all the reference patterns.
  - Recognize the word with the lowest dissimilarity.

  - *For small vocabularies and speaker-dependent recognition.*

8.- Applications.

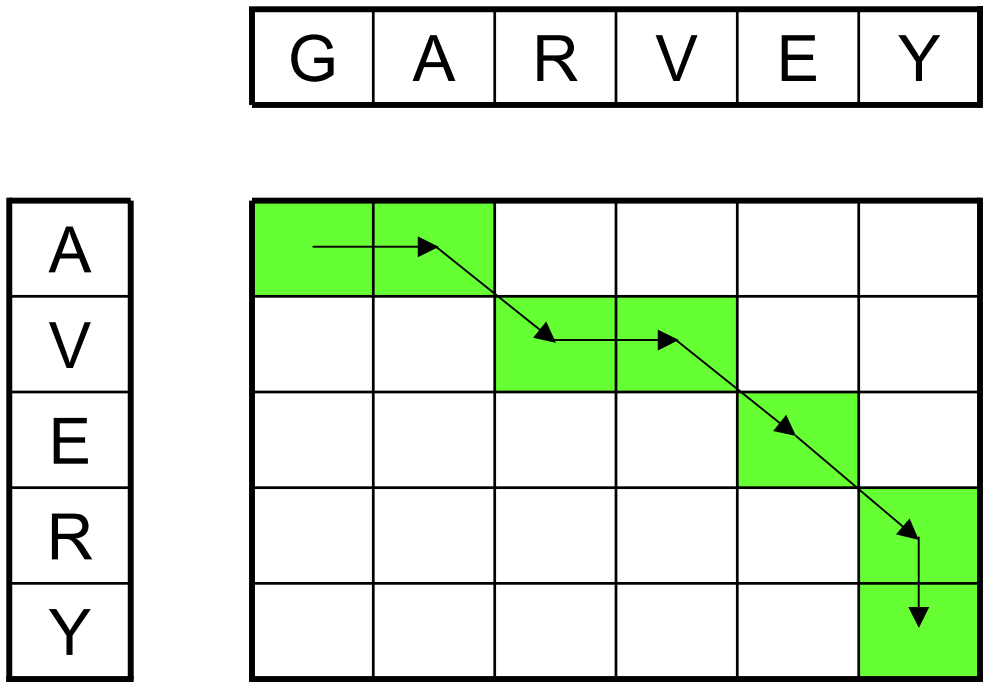
- Speech Recognition (Using LPC Coefficients).



http://www.isip.msstate.edu/projects/speech/software/demonstrations/applets/

8.- Applications.

- String Matching.
  - Basic Operations: Insertion, Deletion, Substitution.



8.- Applications.

- String Matching.

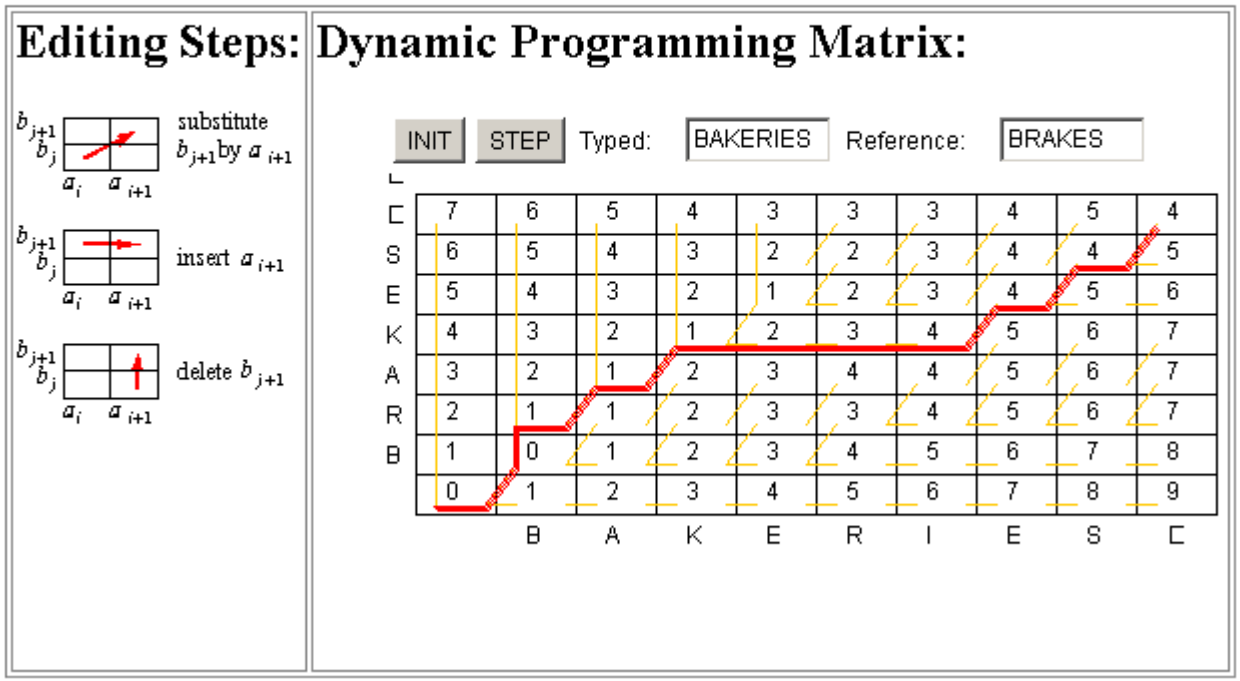
G	A	R	V	E	Y	Y
A	A	V	V	E	R	Y

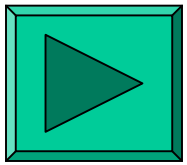
Dissimilarity:

$$\frac{3}{7} = 0.43$$

8.- Applications.

Computing the Minimal Editing Distance

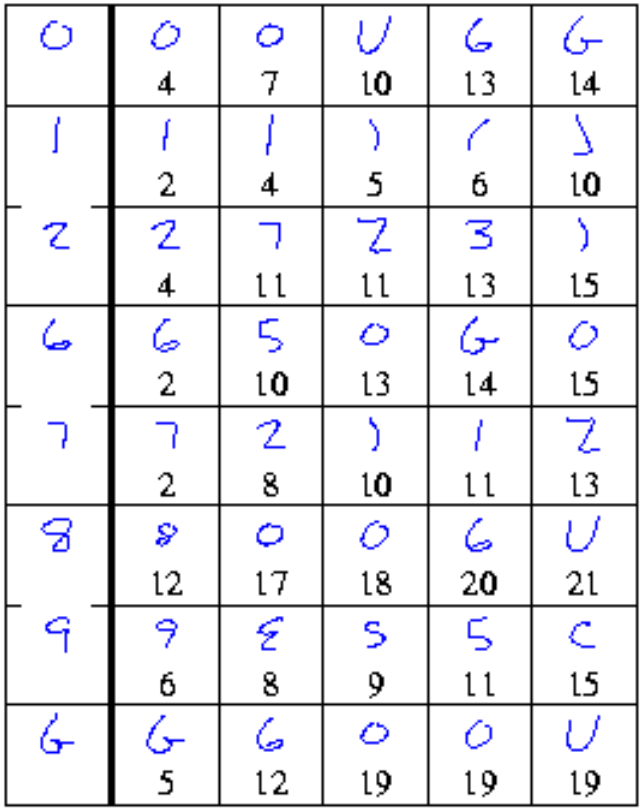




http://isl.ira.uka.de/speechCourse/slides/dtw/editdist/applet/applet.html

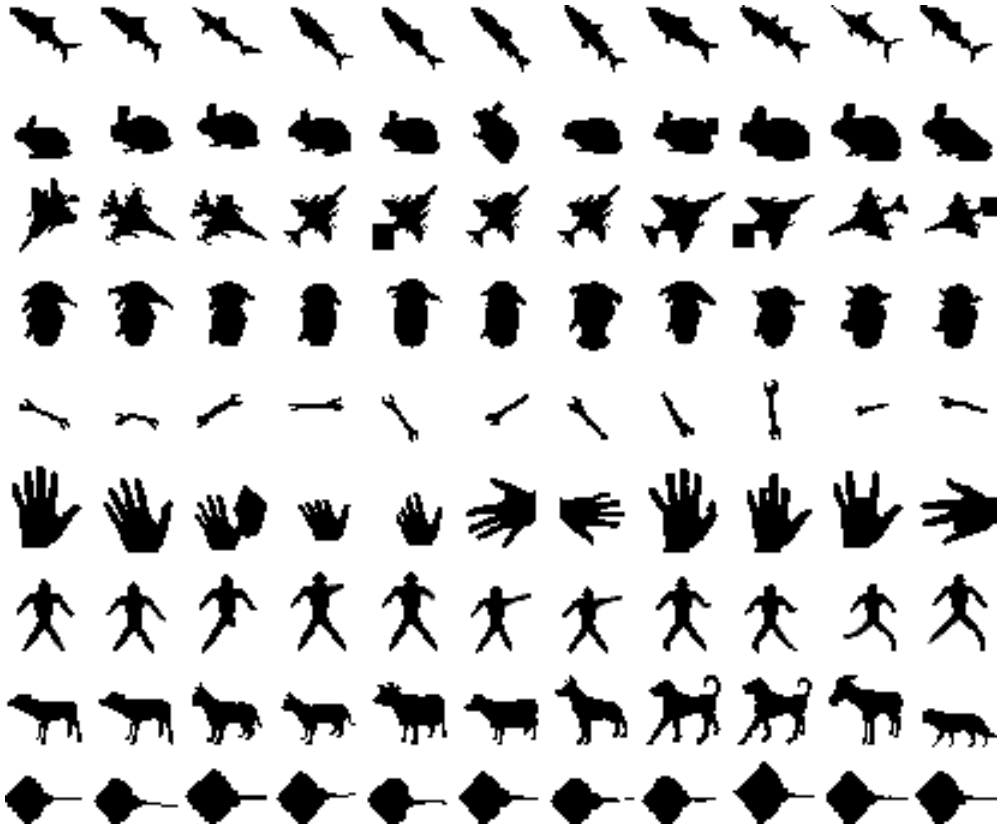
8.- Applications.

- Handwritten Character Recognition.



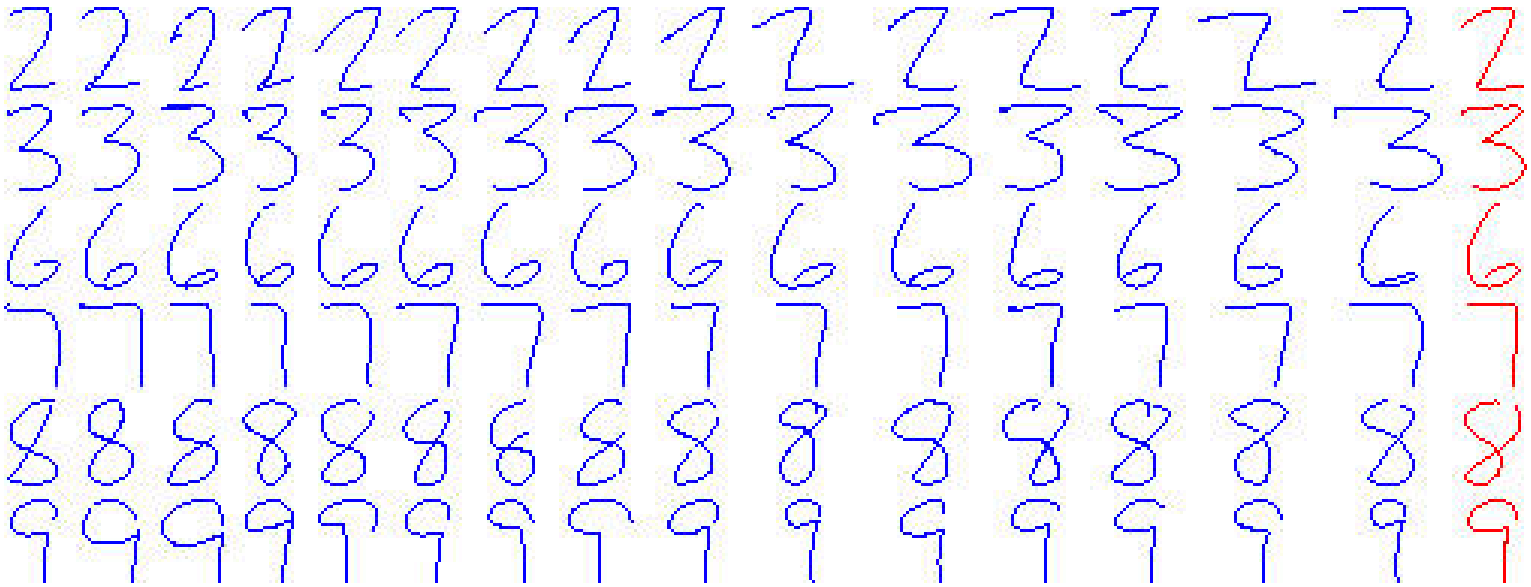
8.- Applications.

- Object Recognition.



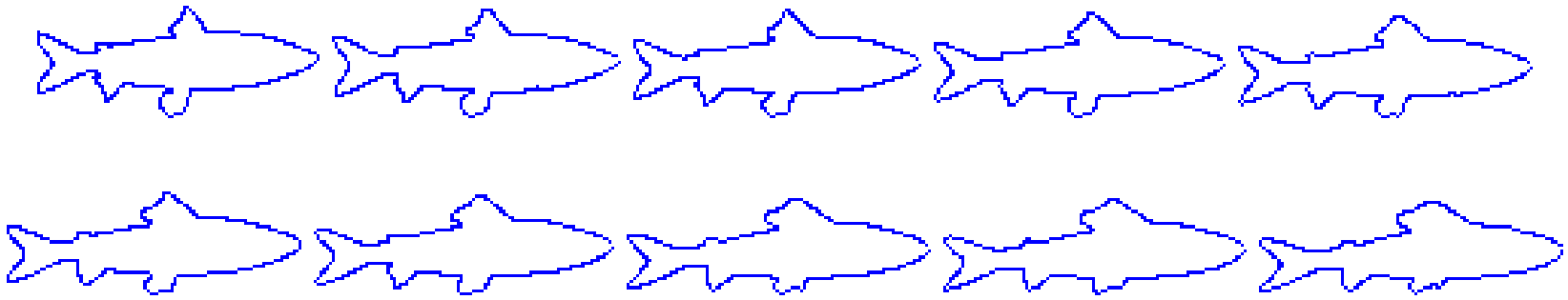
8.- Applications.

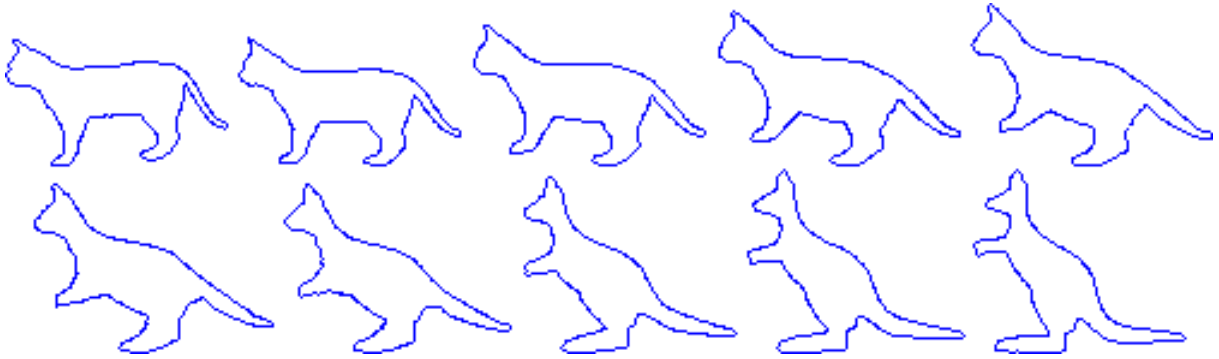
- Prototype Formation.
  - Non-Linear Average using DTW



8.- Applications.

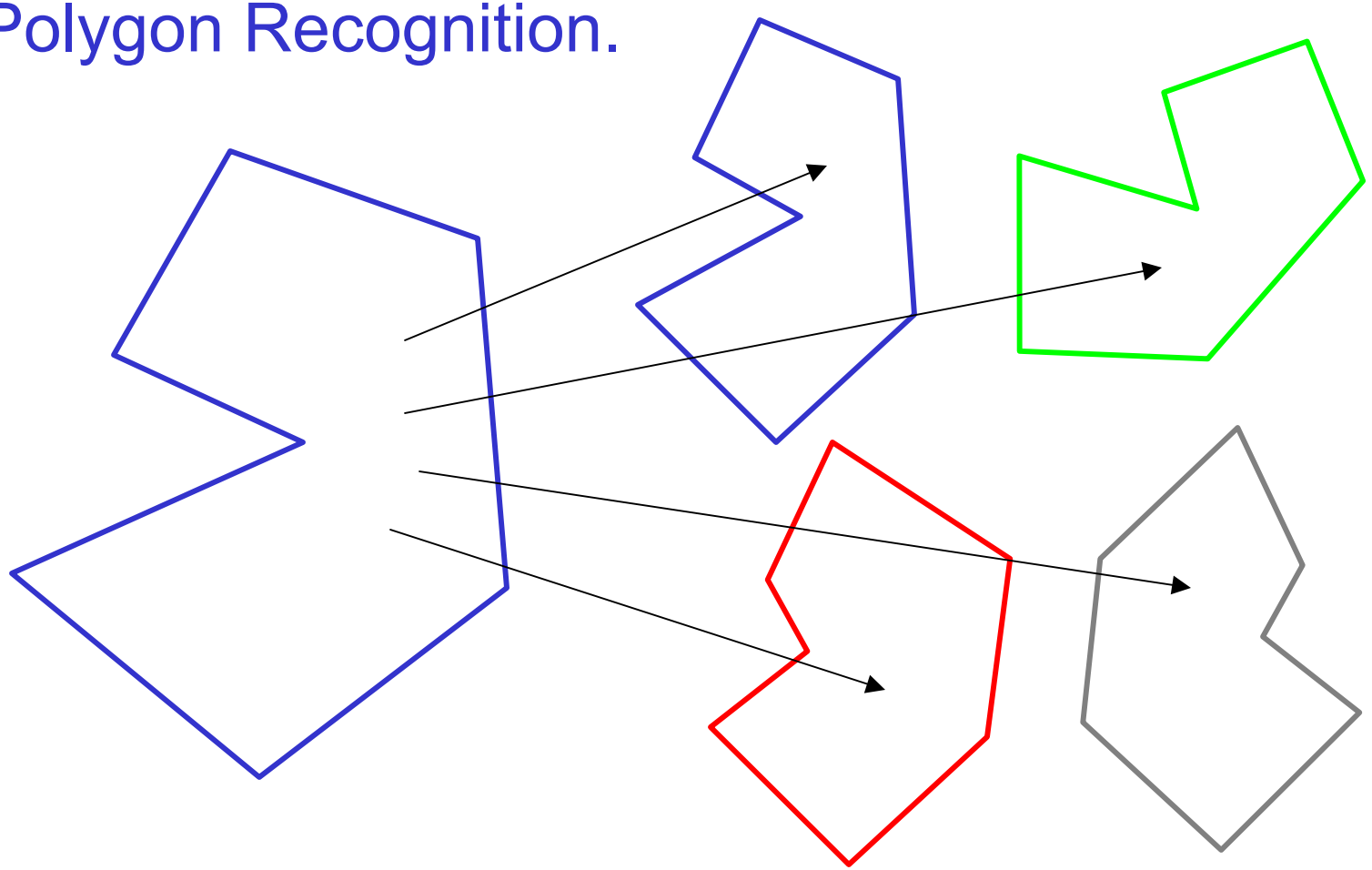
- Morphing.





8.- Applications.

- Polygon Recognition.



8.- Applications.

- Polygon Recognition.
- Symbols with attributes to deal with scaling and rotation.

$$P = \{p_0, p_1, \ldots, p_{n-1}\}$$

$$\text{symbols} \begin{cases} \alpha_i, \beta_i \in [-180^\circ, +180^\circ] \\ \lambda_i, \mu_i \in [0,1], \left(\lambda_i = \dfrac{l_i}{l(P)}\right) \end{cases}$$

8.- Applications.

- Polygon Recognition.

$$[x] = \{\alpha_0, \lambda_0, \ldots, \alpha_{n_x-1}, \lambda_{n_x-1}\}$$
$$[y] = \{\beta_0, \mu_0, \ldots, \beta_{n_y-1}, \mu_{n_y-1}\}$$

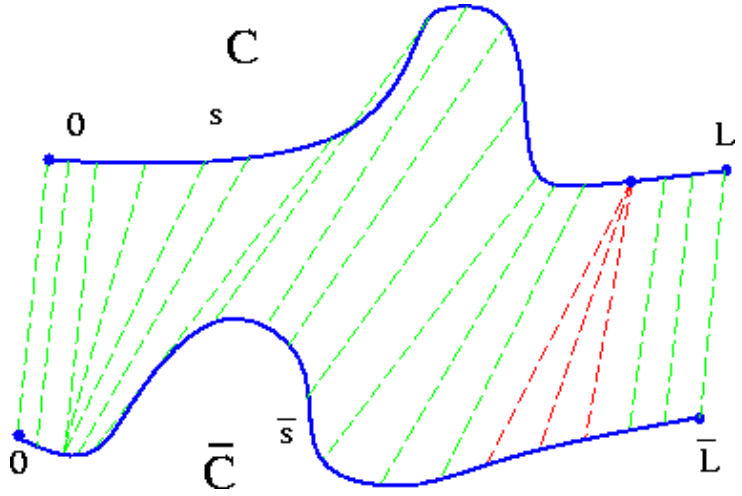
$$\Delta = \min(\delta([x],[y]))$$

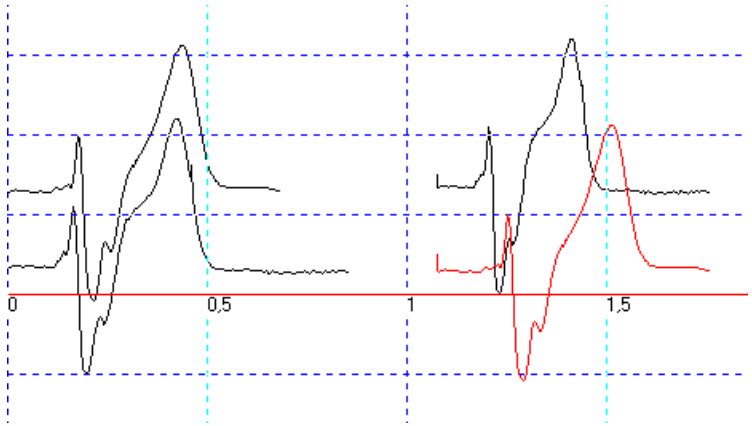
$$d(\alpha_i, \beta_j) = |\alpha_i - \beta_j|$$

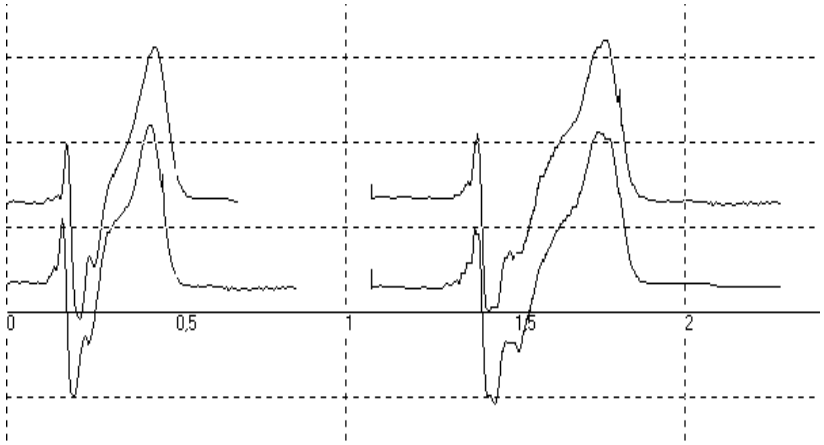
$$d(\lambda_i, \mu_j) = w|\lambda_i - \mu_j|$$

8.- Applications.

- Curve Alignment.







Index

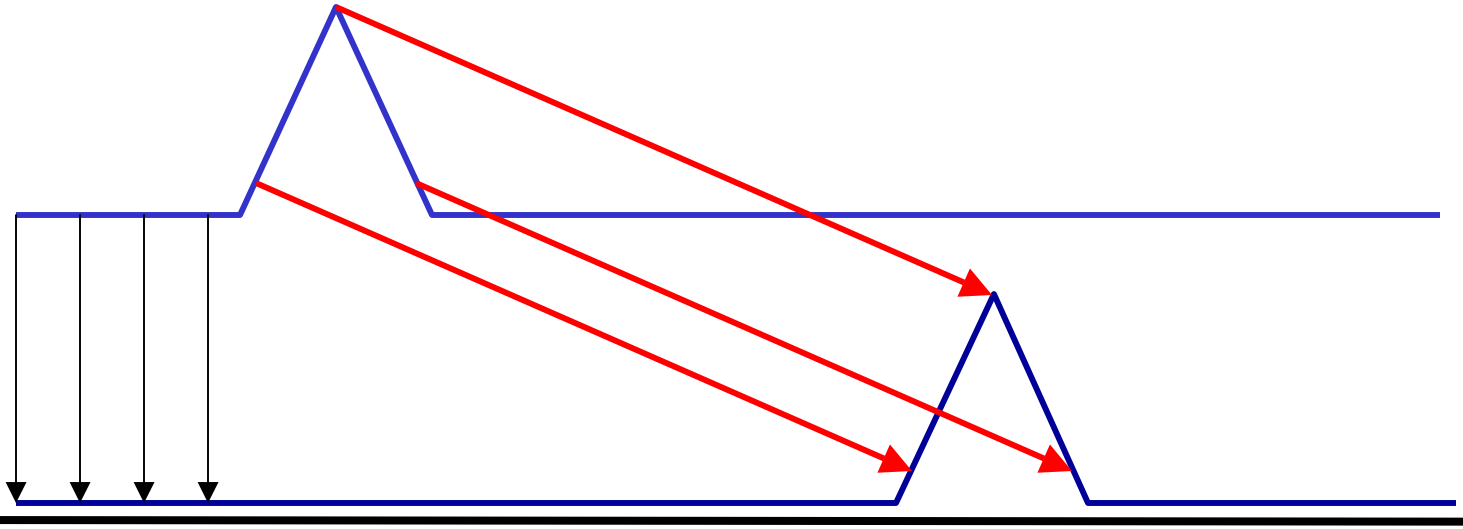
9.- Other Issues.

- DTW Applied to Cyclic Patterns.
- Beam Search.
- Approximate String Matching Techniques.
- Metric Spaces.
- High Level DTW.

9.- Other Issues.

DTW Applied to Cyclic Patterns.

- Linear strings: difficult to find the proper starting symbol $\rightarrow$ **CYCLIC STRINGS**.
- Increasing slightly the computational cost, but the dissimilarity measure does not depend on the starting symbol.



9.- Other Issues.

DTW Applied to Cyclic Patterns.

$$\sigma(a_1 a_2 a_3 \ldots a_n) = a_2 a_3 \ldots a_n a_1$$

$$\forall k \in \aleph, \sigma^k$$

$$[x] = \sigma^k(x), 0 \le k \le n_x - 1$$

$$\Delta([x],[y]) = \min\left\{ \delta\left(\sigma^k(x), \sigma^l(y)\right), k,l \in \aleph \right\}$$

9.- Other Issues.

DTW Applied to Cyclic Patterns.

Brute Force Method:

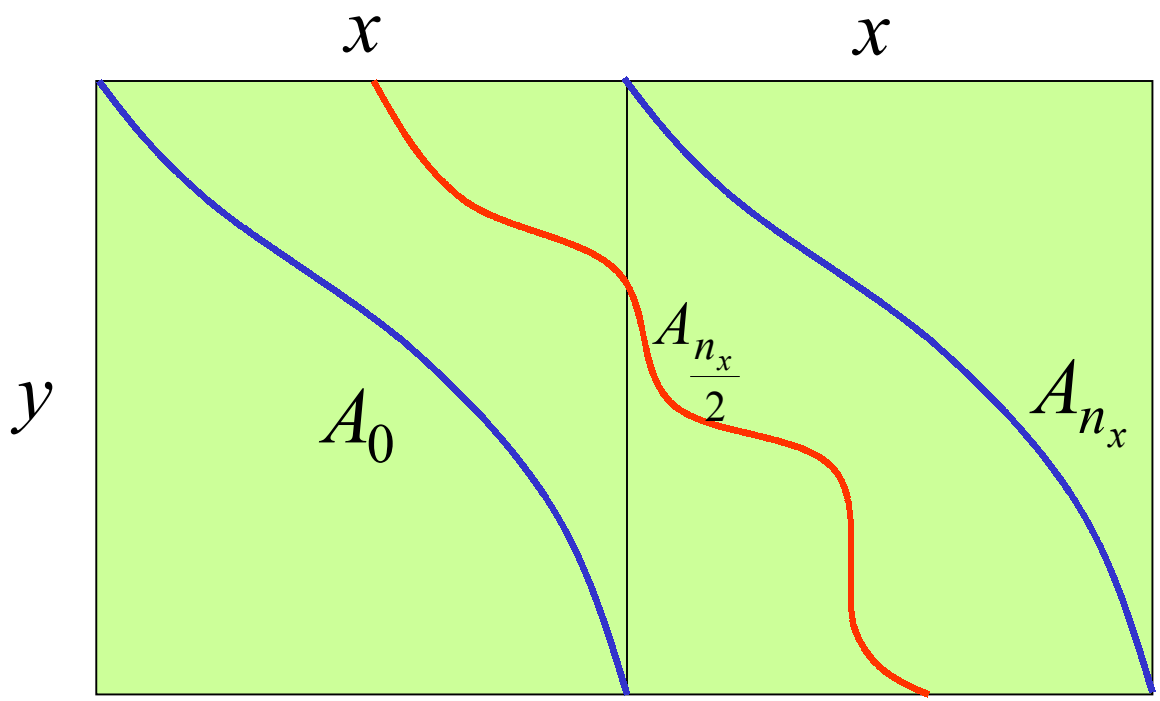
$$\Delta\left(x,\sigma^{l}(y)\right),\forall l\in\{0,1,2\ldots,n_y-1\}\qquad\left(O\left(n_x n_y^2\right)\right)$$

Indirect Method (Paths do not intersect):

$$\Delta\left(x,\sigma^{l}(y)\right),\ \text{from } G[0,l]\ \text{to } G\left[n_x-1,n_y+l-1\right]\qquad\left(O\left(n_x n_y \log n_y\right)\right)$$

9.- Other Issues.

DTW Applied to Cyclic Patterns.



9.- Other Issues.

Beam Search.

- It is not necessary to calculate all the nodes to get the alignment path. Pruning method.
- A limited window is considered at each node.
  - Fixed size k:(O(nk)).
  - Variable size according to a predefined threshold. Data dependent.
- Not optimal.
- Works well in practice.

9.- Other Issues.

Approximate String Matching Techniques.

- Associate strings of symbols with another one according to some similarity criteria:
  - Positional Similarity: matched symbols are in the same position.
  - Ordinal Similarity: matched symbols are in the same order.
  - Material Similarity: measure of the strings symbols coincidence degree.
- Node costs 0 or 1.
- $O(nm) \Rightarrow O(sn)$ or $O(n)$.

9.- Other Issues.

DTW Dissimilarity and Metric Properties.

– DTW dissimilarity does not satisfy the properties of a metric (triangle inequality).

$$(i)\ \Delta(x,y) \geq 0\,,\ \Delta(x,y) = 0\ iff\ x = y\ (\text{Always})$$
$$(ii)\ \Delta(x,y) = \Delta(y,x)\quad (\text{Depends on Production})$$
$$(iii)\ \Delta(x,y) + \Delta(y,z) \geq \Delta(x,z)\ (\text{NO!})$$

– Therefore, DTW dissimilarity can not be considered a metric.

– In practice, there is “some” satisfaction of such properties.

– A computation reduction can be applied when calculating “distances” between a sequence, and each class prototype.

9.- Other Issues.

High Level DTW.

- DTW is computationally expensive. O(nm).
- Sequence length reduction using higher data abstraction.
  - Polynomial approximation of the data (linear).
  - Symbol approximation.
- Significant computational reduction, depending on abstraction grade.
- Detail loss – Exactitude Degradation.

Index

10.- Summary.

- Dynamic Time Warping is a technique to align two sequences in order to obtain a dissimilarity measure using non-linear temporal alignment.
- Computational cost O(nm). Dynamic matrix calculation G and path according to dynamic programming principle.
- Local distance measure can be L_1, L_2, derivative based, etc.
- Global and local constraints applied to the search.
- Cyclic string matching techniques also available.
- Many applications: Speech processing, string matching, polygonal shapes recognition, etc.

Index

11.- Bibliography.

- H. Sakoe and S.Chiba. “Dynamic Programming Algorithm Optimization for Spoken Word Recognition”. IEEE Trans. Acoust. Speech Signal Process. 26, pp. 43-49 (1978).
- Maurice Maes. “Polygonal Shape Recognition Using String-Matching Techniques”. Pattern Recognition, vol.24, pp. 433-440 (1991).
- K.Wang and T.Gasser. “Alignment of Curves by Dynamic Time Warping”. The Annals of Statistics, vol. 25, no.3, pp. 1251-1276 (1997).
- E.Vidal et al. “Is the DTW Distance Really a Metric? An Algorithm Reducing the Number of DTW Comparisons in Isolated Word Recognition”. Speech Communication, vol.4, pp. 333-344 (1995).
- M.Maes. “On a Cyclic String-to-string Correction Problem”. Information Processing Letters, vol.35, pp.73-78 (1990).
- J.Gregor and M.Thomason, “Dynamic Programming Alignment of Sequences Representing Cyclic Patterns”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.15, no.2 (1993).
- J.Deller et al. “Discrete-Time Processing of Speech Signals”. McMillan Publishing Company (1993).
- E.Vullings et al.”Waveform Detection in Holter ECG Using Dynamic Time Warping”, CASEIB97, pp. 313-316 (1997).

Pattern Matching Techniques
(*Dynamic Time Warping*)

- Deterministic Approach
  - Laboratory Exercises
  - Assignments