

# DYNAMIC TIME WARPING: DEFORMATION TEMPORELLE DYNAMIQUE

Géry Casiez      <http://www.lifl.fr/~casiez>  
VisA - Master 2 spécialité IVI – Université de Lille 1

# Références

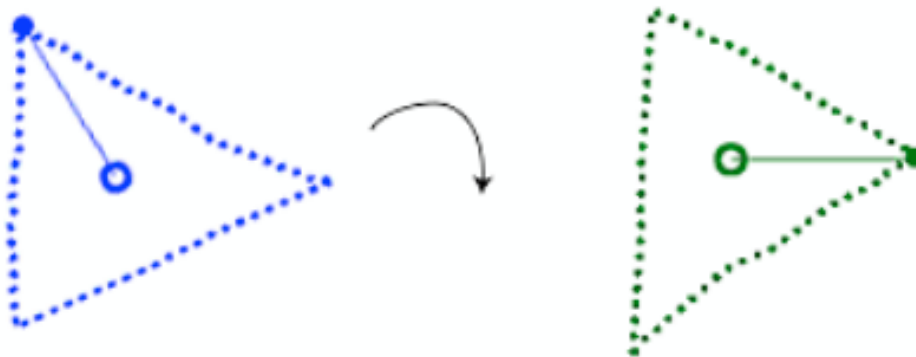
2

- C. S. Myers and L. R. Rabiner.  
A comparative study of several dynamic time-warping algorithms for connected word recognition.  
The Bell System Technical Journal, 60(7):1389-1409,  
September 1981.
- Darrell T.J., and Pentland, A. P., “Recognition of Space- Time Gestures using a Distributed Representation”, (1992) MIT Media Laboratory Perceptual Computing

# Introduction

3

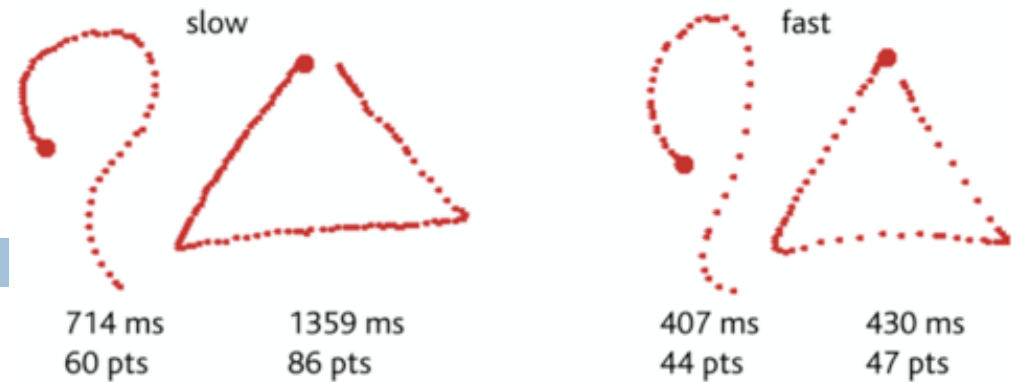
- Exemple des gestes
- Invariance en position
  - ▣ Calcul de centroïde
  - ▣ changement de repère
- Invariance en rotation



Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (UIST '07). ACM, New York, NY, USA, 159-168. DOI=10.1145/1294211.1294238 <http://doi.acm.org/10.1145/1294211.1294238>

# Introduction

4

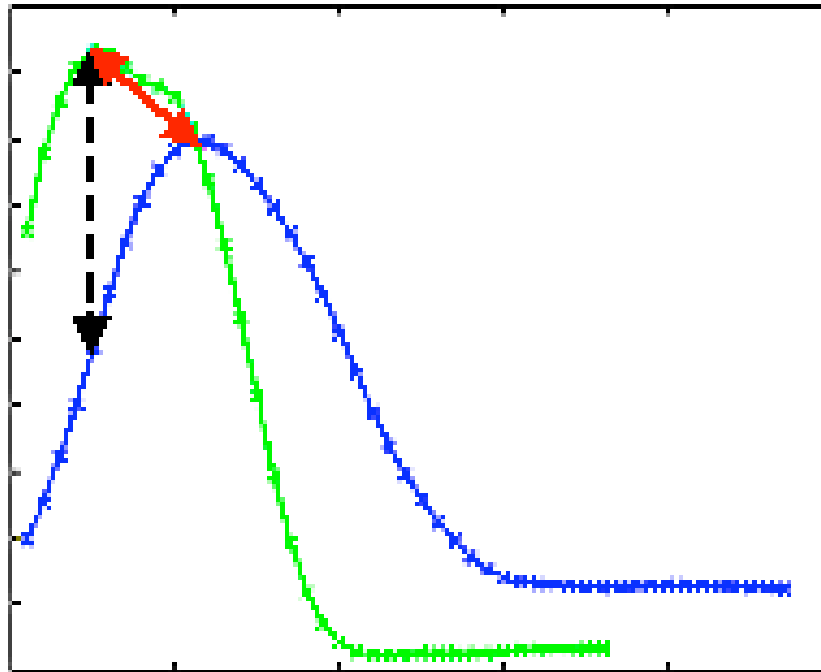


- Invariance à l'échelle
  - Calcul de boîte englobante
- Invariance à la fréquence d'échantillonnage, vitesse d'exécution, variations de vitesse en cours d'exécution
  - Ré-échantillonnage en N points équidistants (en distance euclidienne) par interpolation linéaire
  - L'interpolation linéaire n'est pas réalisable pour n'importe quel type de signal (ex: son)
- Comment reconnaître des gestes sans explicitement spécifier le début et la fin?

# Introduction

5

- Comment mettre en correspondance deux signaux en étant insensible aux variations de rythme?
- Comment gérer les différences en nombre d'échantillons de deux signaux?



# Dynamic Time Warping

6

- DTW
- Déformation temporelle dynamique
- Déterminer pour chaque élément d'une séquence, le meilleur élément correspondant dans l'autre séquence relativement à un certain voisinage et à une certaine métrique
- Complexité polynomiale

# Applications

7

- Vidéo, audio, graphique...
- Toutes données qui peuvent être transformées en représentation linéaire en fonction du temps (séries temporelles)
- Echantillons ordonnés par une étiquette de temps
- Reconnaissance vocale
- Reconnaissance de gestes off-line et on-line
- alignement de protéines...

# Principe de base

8

- Séquence de référence  $R = [r_1, r_2, \dots, r_n]$
- Séquence de test  $T = [t_1, t_2, \dots, t_m]$
- Si  $m=n$  alors on peut calculer la distance entre les deux signaux de la façon suivante (pas forcément idéal):

$$D = \sum_{i=1}^n \text{distance}(r_i, t_i)$$

- Possibilité de calculer la distance euclidienne ou d'utiliser une autre métrique (fonction qui donne un réel)
- Plus possible d'utiliser cette méthode dès que  $n \neq m$
- Attention: les échantillons des séquences doivent être équidistants en temps

# Principe de base

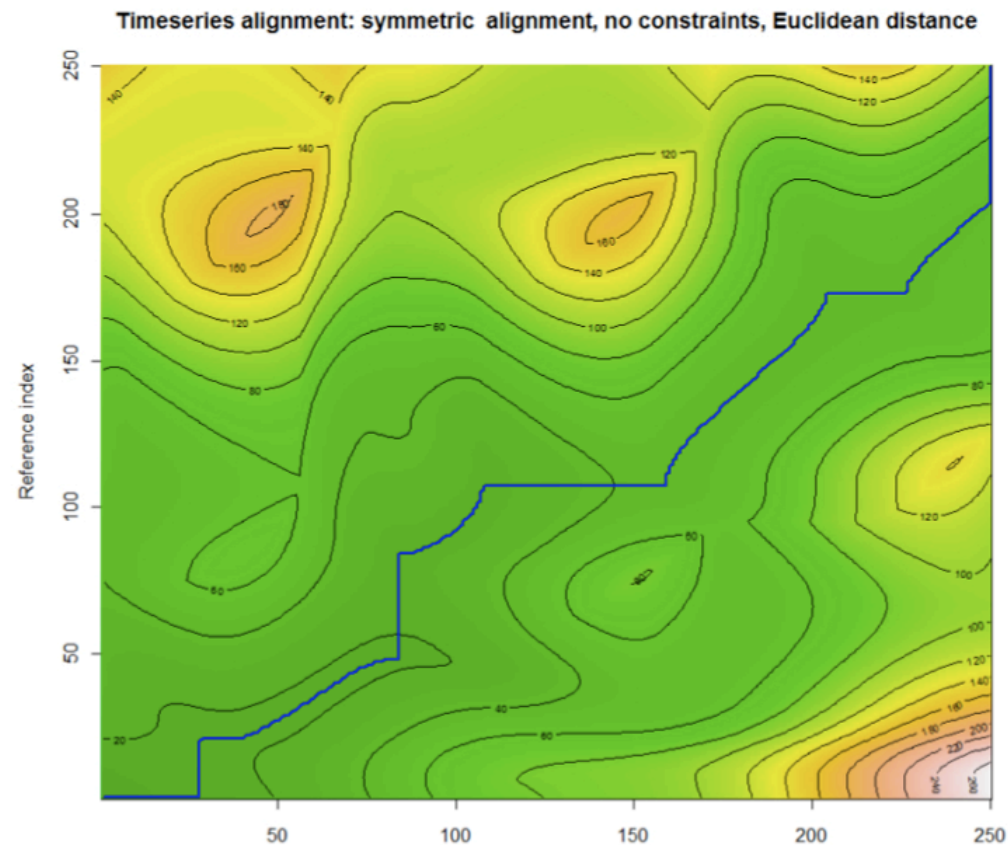
9

- DTW réalise d'abord un alignement non linéaire en recherchant parmi tous les alignements possibles, celui qui minimise une fonction de coût cumulé
- «Time Warping»: Dilation ou compression des séquence pour obtenir le meilleur alignement possible

# Principe de base

10

- Détermination du chemin  $W = [w_1, w_2, \dots, w_k]$  de distance minimale  $\sum_{i=1}^k distance(w_i)$



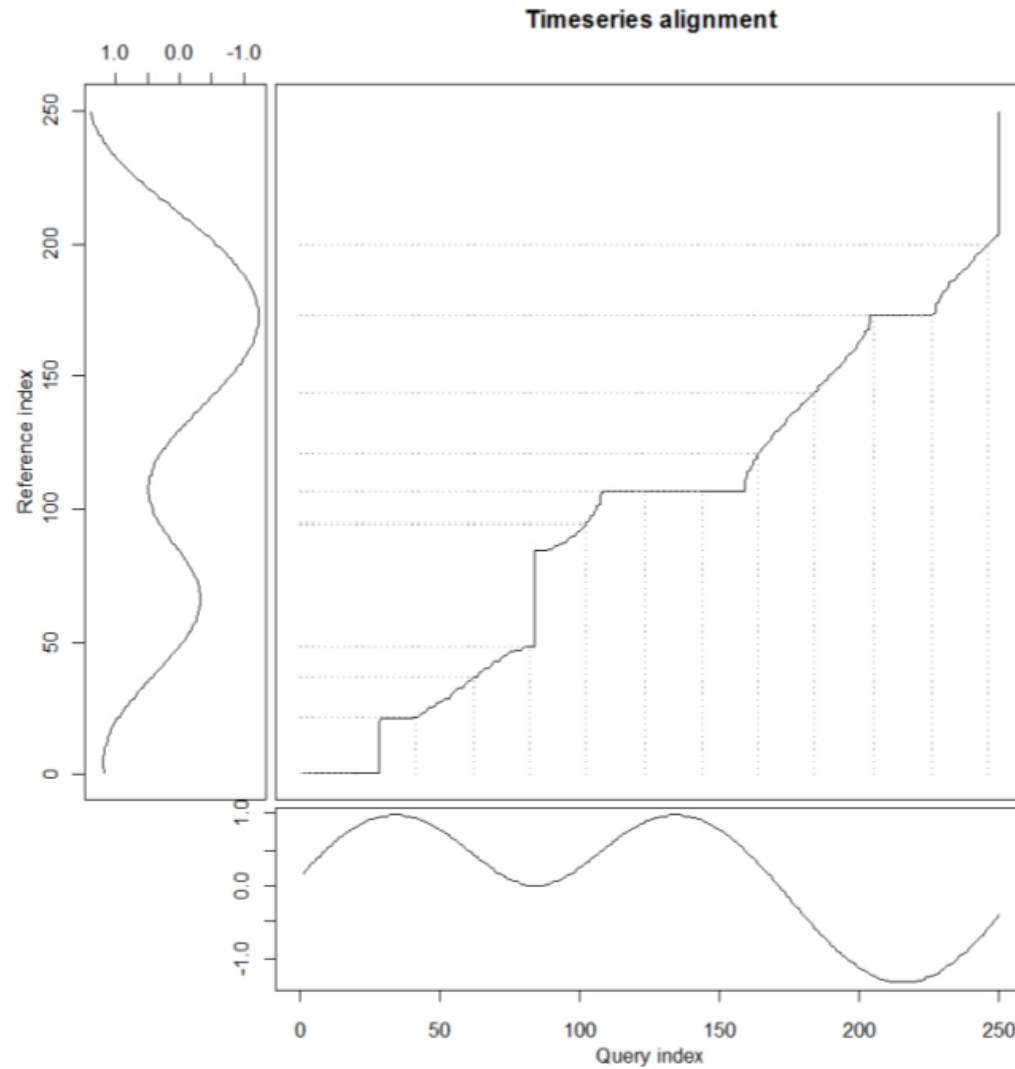
# Principe de base

11

- Conditions aux frontières
  - $w_1 = (r_1, t_1)$
  - $w_k = (r_n, t_m)$
- Contraintes locales
  - Monotonie pour respecter le séquençage des points
  - Éviter les sauts dans le temps
  - Pour tout couple  $(r_i, t_j)$ , le choix des prédécesseurs est limité à
    - $(r_{i-1}, t_j), (r_i, t_{j-1}), (r_{i-1}, t_{j-1})$
- Exhaustivité
  - Chaque élément de R doit être mis en relation avec au moins un élément de T et vice-versa
  - $\max(m, n) \leq k \leq m + n - 1$

# Principe de base

12



# Principe de base

13

- Programmation dynamique
- Fonction d'optimisation:
  - ▣ Soit  $D(i,j)$  la longueur du chemin entre  $(r_1, t_1)$  et  $(r_i, t_j)$
- Récursion:
  - ▣  $D(i,j) = \text{dist}(r_i, t_j) + \min(D(i-1, j), D(i-1, j-1), D(i, j-1))$
  - ▣ condition initiale:  $D(1,1) = \text{dist}(r_0, t_0)$
- Distance minimale entre les deux séquences
  - ▣  $D(n,m)$

# Mise en application

14

- Construction d'une matrice  $D$  de dimensions  $n \times m$  dans laquelle chaque élément  $(i,j)$  contient  $D(i,j)$
- Remplissage de  $D(1,1)$  avec la condition initiale
- Utilisation de la formule récursive pour remplir la matrice ligne par ligne ou colonne par colonne
- Cas particuliers première ligne et première colonne
- Distance minimale donnée par l'élément  $D(n,m)$
- La distance minimale peut être normalisée par la longueur du chemin

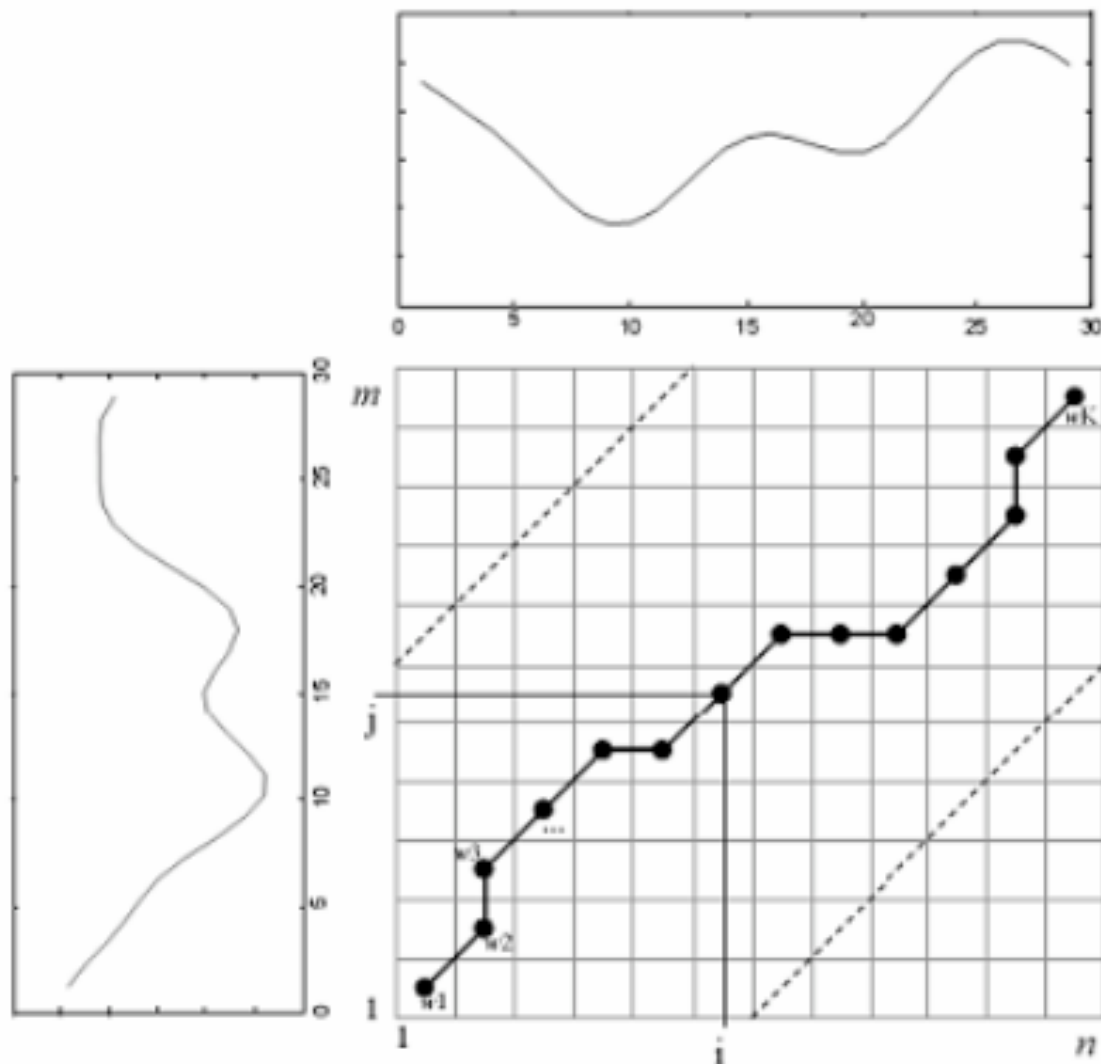
# Algorithme

15

```
n ← |X|
m ← |Y|
dtw[] ← new [n × m]
dtw(0, 0) ← 0
for i = 1; i ≤ n; i ++ do
    dtw(i, 1) ← dtw(i - 1, 1) + c(i, 1)
end for
for j = 1; j ≤ m; j ++ do
    dtw(1, j) ← dtw(1, j - 1) + c(1, j)
end for
for i = 1; i ≤ n; i ++ do
    for j = 1; j ≤ m; j ++ do
        dtw(i, j) ← c(i, j) + min {dtw(i - 1, j); dtw(i, j - 1); dtw(i - 1, j - 1)}
    end for
end for
return dtw
```

# Chemin de déformation

16



# Chemin de déformation

17

- A chaque calcul de  $D(i,j)$ , sauvegarde du prédécesseur qui minimise la distance
- Parcours des prédécesseurs en partant de  $D(n,m)$

# Complexité

18

- Complexité:  $O(m*n)$
- Optimisation en limitant la région de recherche

# Optimisation

19

- La zone supérieure gauche et la zone inférieure droite ne sont pas calculées
- Les distances locales associées sont mises à une valeur très élevée afin que le chemin n'y passe pas



# Optimisation

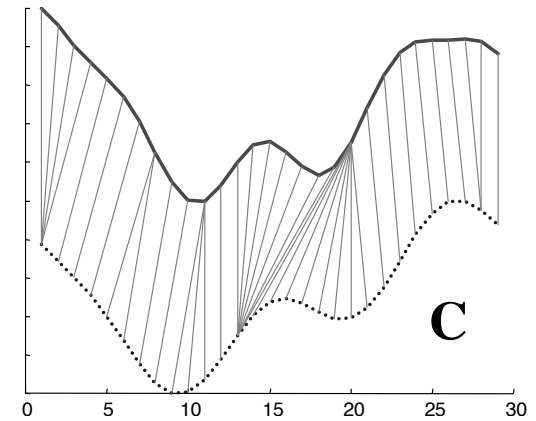
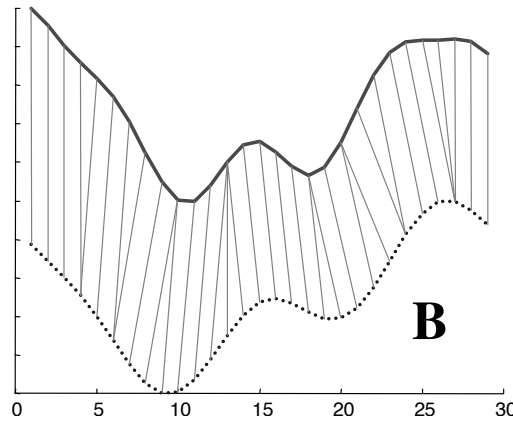
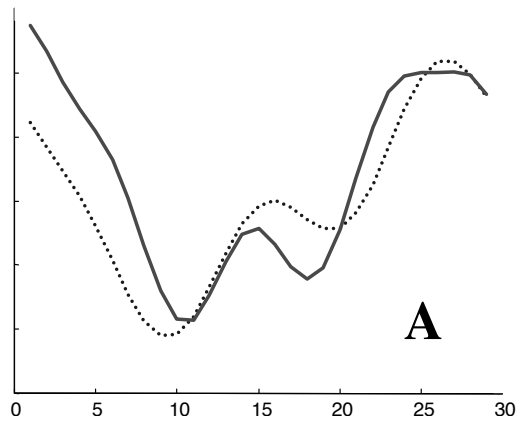
21

- Mise à l'échelle
  - Réduire les tailles de R et T

# Singularités

22

- Une singularité apparaît quand un point d'une séquence est associé à de nombreux points de l'autre séquence sans raison valable



# Singularités: technique de fenêtrage

23

- Ajout de contraintes dans les alignements possibles
- Technique de fenêtrage
  - ▣ Donne une borne supérieure à la singularité

```
for  $i = 1; i \leq n; i++$  do      for  $j = \max(1, i-w); j \leq \min(m, i+w); j++$  do
  for  $j = 1; j \leq m; j++$  do
     $dtw(i, j) \leftarrow c(i, j) + \min \{ dtw(i-1, j); dtw(i, j-1); dtw(i-1, j-1) \}$ 
  end for
end for
```

# Singularités: slope weighting

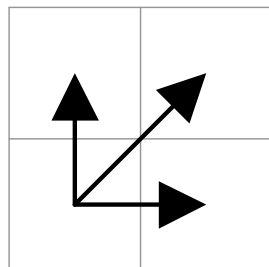
24

- Pondérer les déplacements suivant les directions:
  - $D(i,j) = \text{dist}(r_i, t_j) + \min(X * D(i-1, j), D(i-1, j-1), X * D(i, j-1))$
  - $X > 1$
- Force l'alignement suivant la diagonale

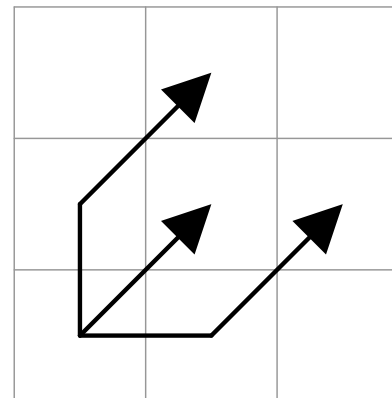
# Singularités: Step patterns

25

- $D(i,j) = \text{dist}(r_i, t_j) + \min(D(i-1, j-2), D(i-1, j-1), D(i-2, j-1))$
- Force le déplacement suivant la diagonale



**A**



**B**

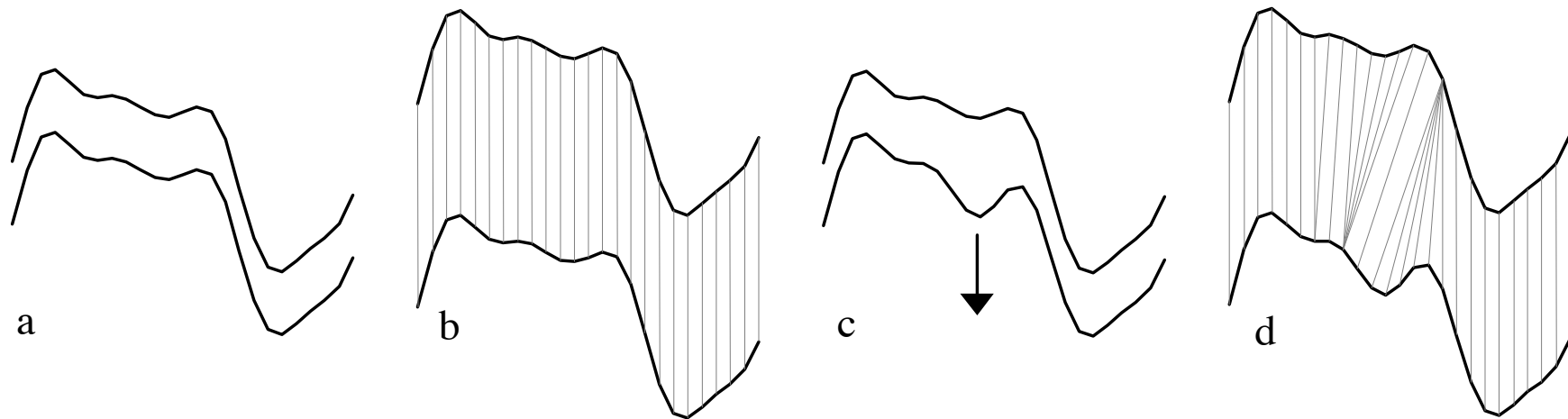
# Singularités

26

- Problème: ces contraintes peuvent empêcher d'obtenir l'alignement correct
- Comment définir les différents paramètres des techniques de levée des singularités (taille de fenêtre, coefficients de pondération, motifs)?

# DDTW: Derivative Dynamic Time Warping

27



```
@INPROCEEDINGS{Keogh01derivativedynamic,  
  author = {Eamonn J. Keogh and Michael J. Pazzani},  
  title = {Derivative Dynamic Time Warping},  
  booktitle = {In First SIAM International Conference on Data Mining (SDM'2001)},  
  year = {2001}  
}
```

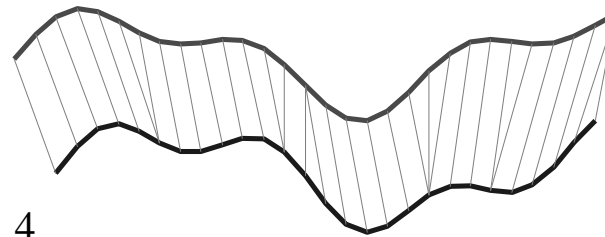
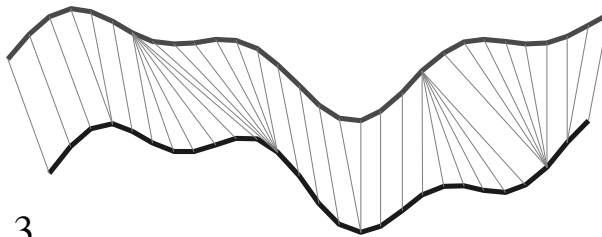
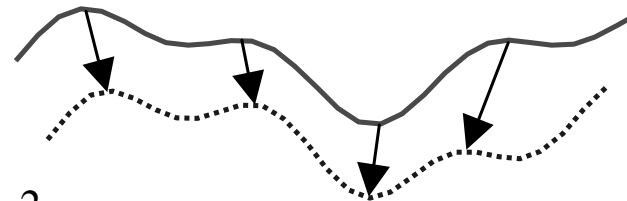
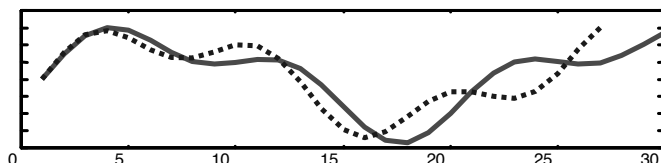
# DDTW: Derivative Dynamic Time Warping

28

- Utilisation de la dérivée des données des séquences

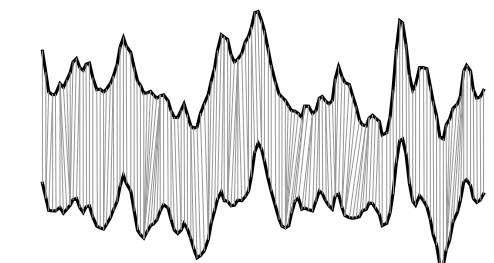
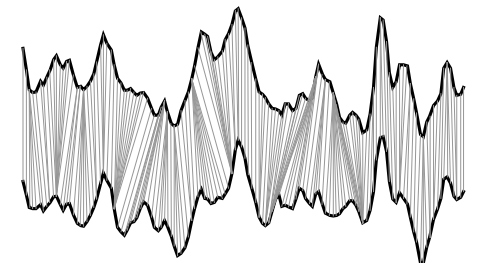
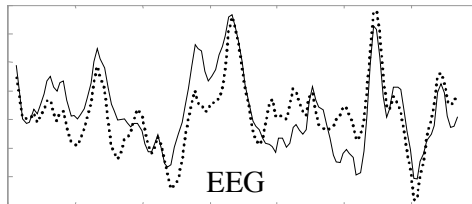
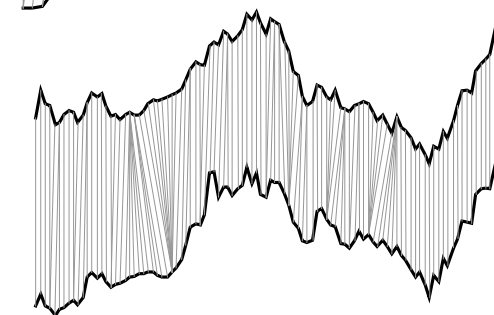
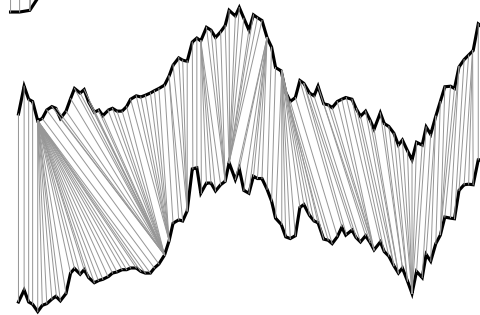
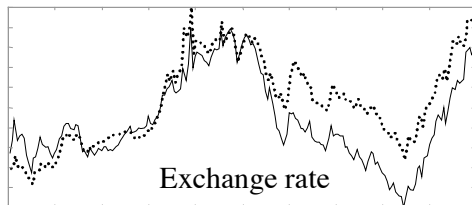
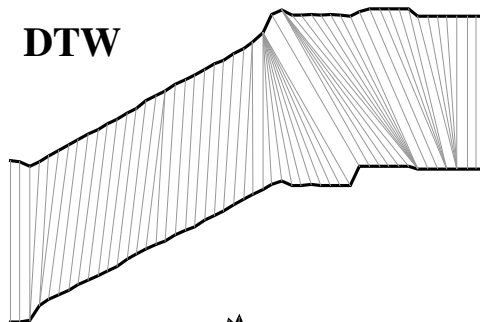
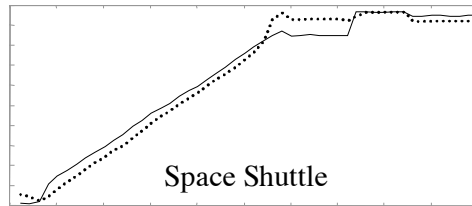
$$D_x [q] = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2}$$

- Calcul du carré de la différence des dérivées



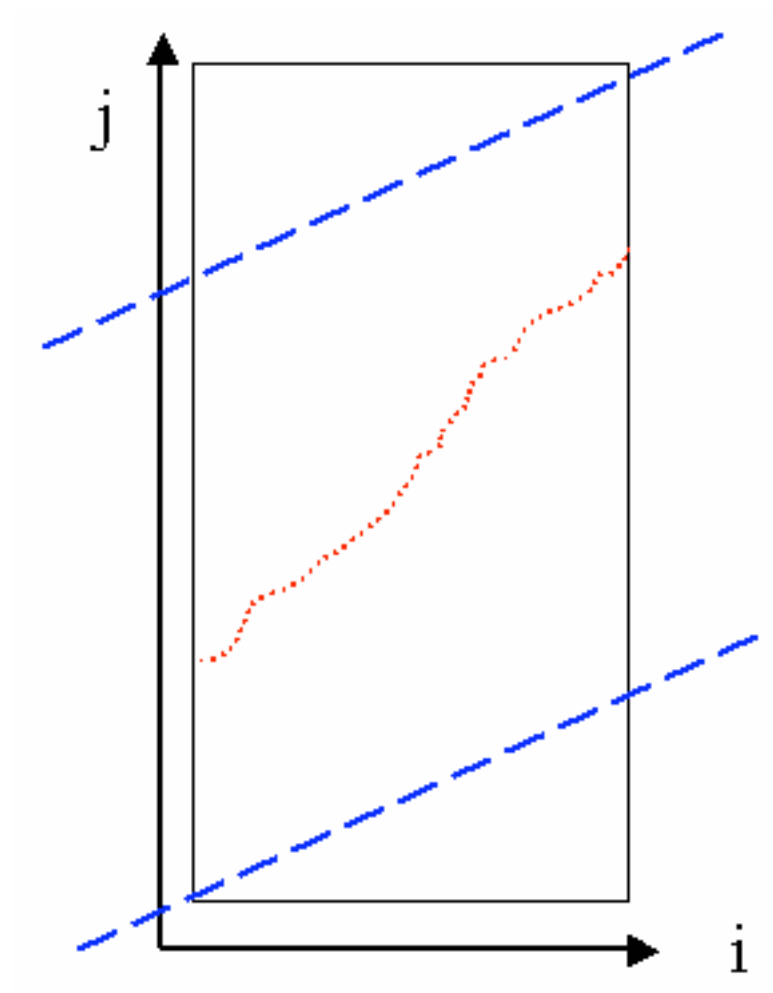
# DDTW: Derivative Dynamic Time Warping

29



# Recherche de motifs

30



# Reconnaissance de gestes

31

- Le geste fait généralement l'objet de variance dans sa reproduction
- Un seul geste exemple n'est souvent pas suffisant
- Définition d'un modèle statistique prenant en compte ces variations
- Utilisation de plusieurs gestes exemples
  - ▣ Calcul de la moyenne et de la variance en fonction du temps

# Reconnaissance de gestes

32

- Chaque exemple est mis en correspondance, en utilisant DTW, avec le geste exemple le plus long avant de calculer les caractéristiques statistiques
- Calcul d'un modèle de geste  $g$  pour chaque classe d'exemples
  - ▣ Calcul de la moyenne et de la variance à chaque pas de temps:  $\tilde{g}_m[t]$  et  $\sigma^2(g_m[t])$

# Reconnaissance de gestes

33

- Un geste candidat est aligné avec chaque modèle en utilisant DTW
- A chaque nouvel événement, la séquence de test  $T$  est comparée à chaque modèle en utilisant la métrique suivante:

$$D_{i,j} = \sum \frac{1}{\sigma^2(g_m[t])} (\tilde{g}_m[t] - T[j])^2$$

# Reconnaissance de gestes

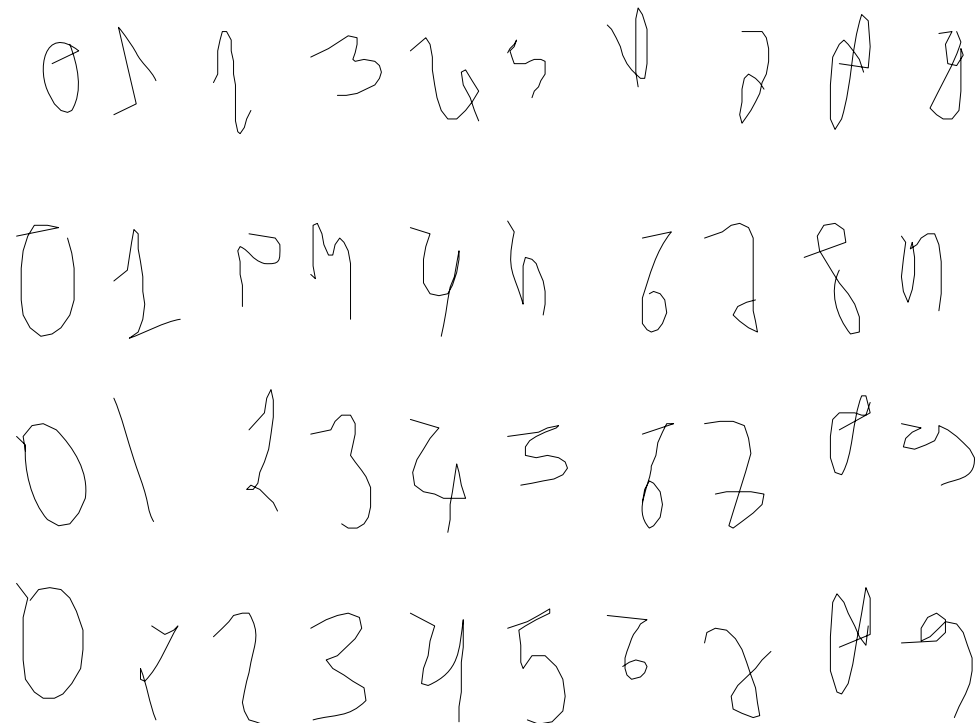
34

- L'alignement de la séquence T avec les modèles se fait par retour dans le temps en considérant que le temps présent est le temps 0
- Relaxation des contraintes en considérant une longueur minimale pour la séquence T
- Le score du geste est l'inverse de la distance calculée par DTW
- Le geste avec le score le plus élevé est considéré comme le geste reconnu
- Utilisation de seuils

# Affine invariant DTW

35

- Problème: identifier dans une séquence un objet qui subit une transformation affine
- Choix de la fonction de coût?



# Affine invariant DTW

36

## Optimization Algorithm of AI-DTW:

Initialize

The warping path  $w^{(1)} = DTW\_PATH(T, R)$ .

Iteration number  $k=1$

While not convergence

$k = k+1$ ;

Update the transformation matrix by:

$$A^{(k)} = \arg \min_A \left\{ \sum_{i=1}^n \left\| t_i - r_{w^{(k-1)}(i)} A \right\|^2 \right\} .$$

Update the warping path by:

$$w^{(k)} = DTW\_PATH(T, RA^{(k)}).$$

End While

# Multi-Dimensional Dynamic Time Warping

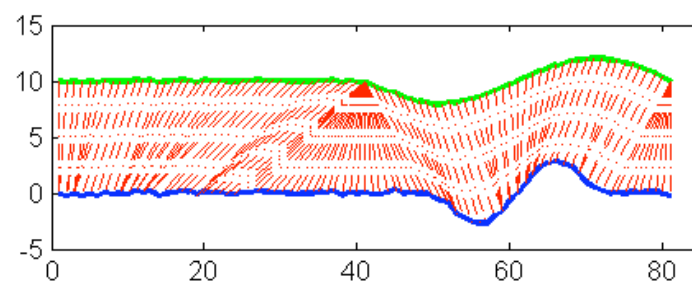
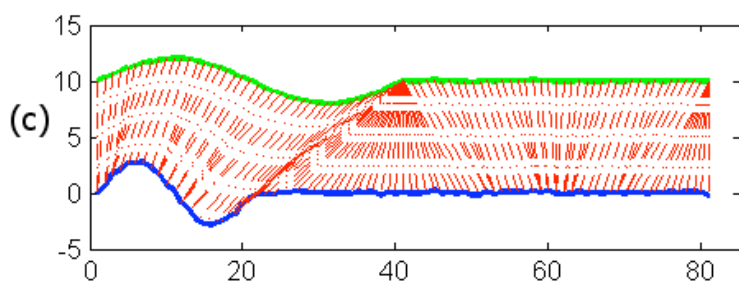
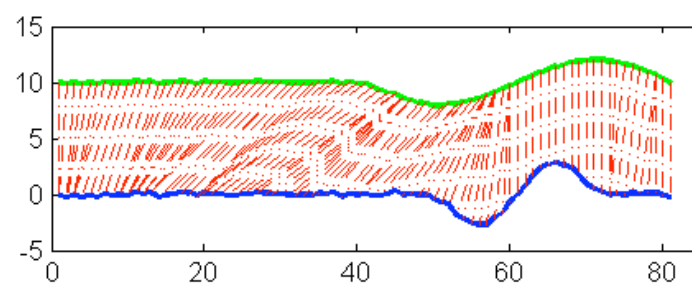
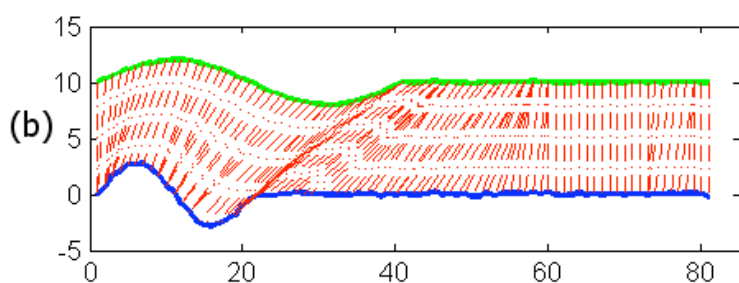
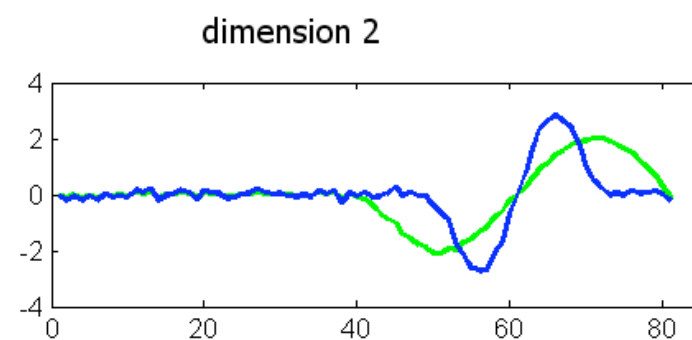
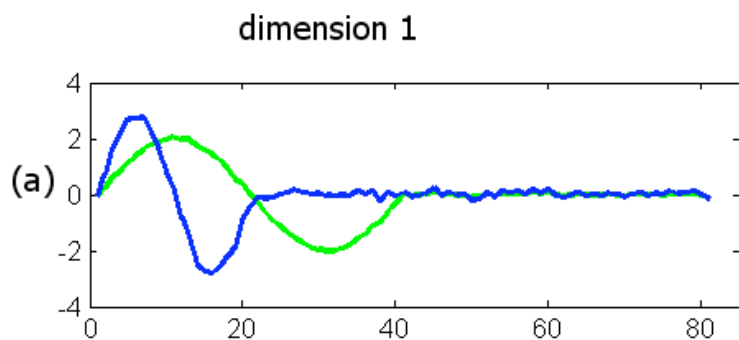
37

- Comment appliquer DTW quand plusieurs grandeurs sont mesurées à chaque instant?
- Modification de la fonction de distance:
  - ▣ Chaque dimension est normalisée avec une moyenne de 0 et une variance de 1
  - ▣ Calcul de la somme des valeurs absolues des différences suivant chaque dimension

G. A. ten Holt, M. J. T. Reinders and E. A. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In Proceedings of the Thirteenth annual conference of the Advanced School for Computing and Imaging, 2007.

# Multi-Dimensional Dynamic Time Warping

38



# Mutli-Dimensional Dynamic Time Warping

39

- Possibilité d'utiliser les signaux et leur dérivée

# TP

Université de Lille 1 - M2 IVI - VisA - Dynamic Time Warping - G. Casiez

Drag avec le bouton gauche ou droit de la souris + Shift : création d'une courbe de référence  
Drag avec le bouton gauche ou droit de la souris : création d'une courbe de test

40

