TP : Dynamic Time Warping

Objectifs

— se familiariser avec la technique Dynamic Time Warping.

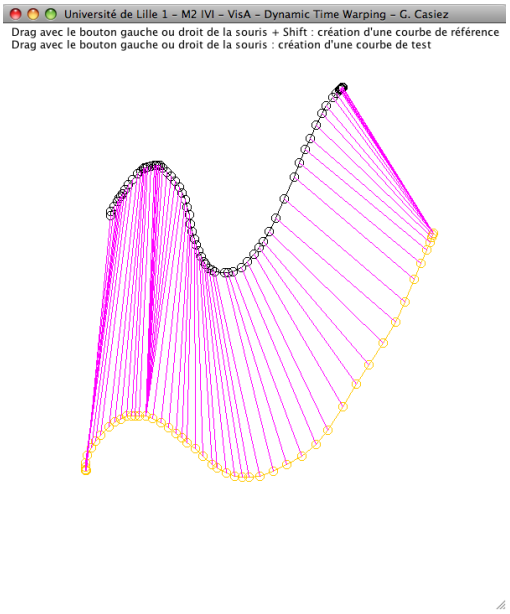

FIGURE 1 – L'application finale.

1 Matériel à disposition

Une interface écrite en Java Swing est mise à votre disposition pour saisir des courbes en utilisant la souris. Cette interface comprend les fichiers TestGUI.java [1] pour définir la fenêtre de l'application et Canvas2D.java [2], qui hérite de Canvas [3], pour définir la zone de dessin. L'interface vous permet de saisir et afficher (en noir) une courbe de référence en cliquant sur un des boutons de la souris tout en appuyant sur la touche Shift. La courbe de test qui va servir à l'appareillement (en orange) s'obtient de la même façon sans appuyer sur la touche shift.

Une classe Matrix [4] est également mise à votre disposition pour faciliter la manipulation de matrices.

2 Mise en correspondance de tous les points

Question 1. Créez une classe `DTW` qui calcule la matrice D vue en cours à partir de deux gestes tracés.

Question 2. Ajoutez lors du calcul de la matrice, le prédécesseur de D(i,j) (utilisez `couple` de `Matrix`)

Question 3. En utilisant ces calculs, définissez la liste des points à appareiller.

Question 4. Affichez le résultat de la mise en correspondance (couleur mauve sur la figure 1)

1. TestGUI.java
2. Canvas2D.java
3. http://download.oracle.com/javase/1.4.2/docs/api/java/awt/Canvas.html
4. Matrix.java

3 Recherche de motifs

Question 5. Adaptez votre code pour rechercher des motifs, comme vu en cours (exemple figure 2).

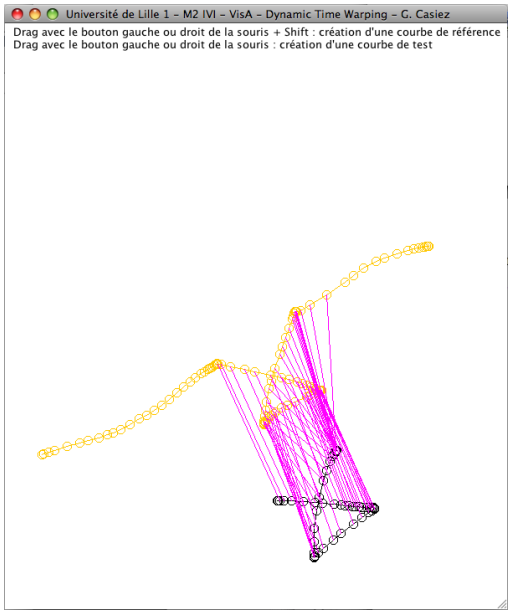

FIGURE 2 – Utilisation de DTW pour la recherche de motif.