

27/03/2012



PROJET
BIOMETRIE

ACQUISITION ET RECONNAISSANCE DE
SIGNATURES

Dossier de conception



Réseaux Informatiques et Communication Multimédia (5^{ème} année)

Renaud Collin - Corentin Foucault - Flavien Garcia - Christophe
Havard - Thibaud Michel - Aurélie Moiroux

Table des révisions

Version	Auteur	Date	Informations
0.1	Renaud Collin	20/02/12	Création du document, établissement du plan
0.2	Renaud Collin	07/03/12	Rédaction : "Architecture du projet"
0.3	Renaud Collin Aurélie Moiroux	14/03/12	Rédaction : "Tablette Bamboo"

Sommaire

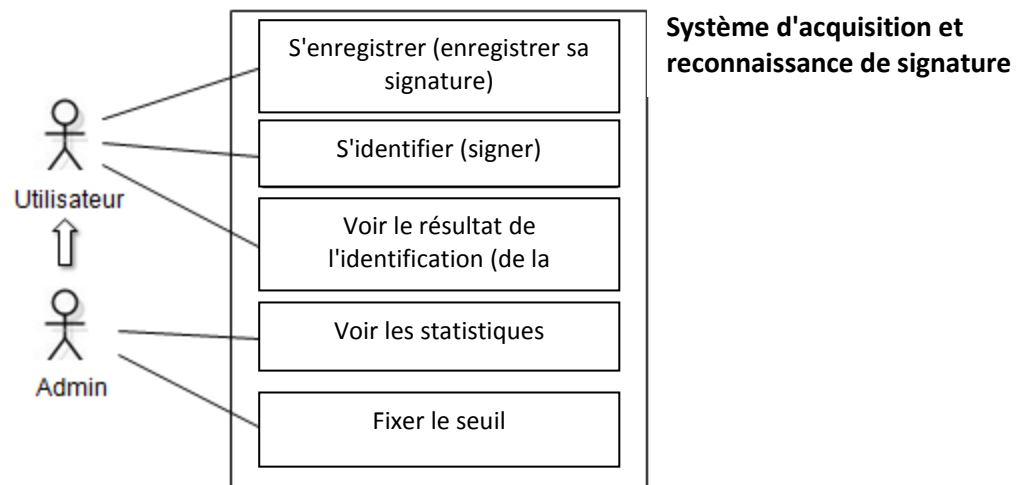
Architecture du projet.....	4
Diagramme de contexte.....	4
Diagramme de cas d'utilisation.....	4
Vue logique.....	5
Vue physique.....	5
Mise en place du matériel.....	6
Montage Arduino-DS.....	6
Tablette Bamboo.....	6
IHM.....	7
Algorithmes.....	12
Prétraitement des acquisitions.....	12
Ré-échantillonnage sur le temps.....	13
Interpolation.....	14
Analyse en composantes principales.....	14
Centrage et Rotation.....	15
Normalisation (zoom).....	17
Ré-échantillonnage sur le nombre de points.....	18
Algorithmes de comparaison.....	18
Comparaisons simplistes.....	18
Coefficient de corrélation temporel.....	18
Dynamic Time Warping.....	18
Matrice de radiation.....	19
Somme des différences d'angles entre vecteurs.....	19
Génération d'un modèle classification.....	19
Construction du fichier de comparaison (apprenstissage.arff).....	19
Construction du modèle.....	20
Classification par arbre de décision.....	21

Architecture du projet

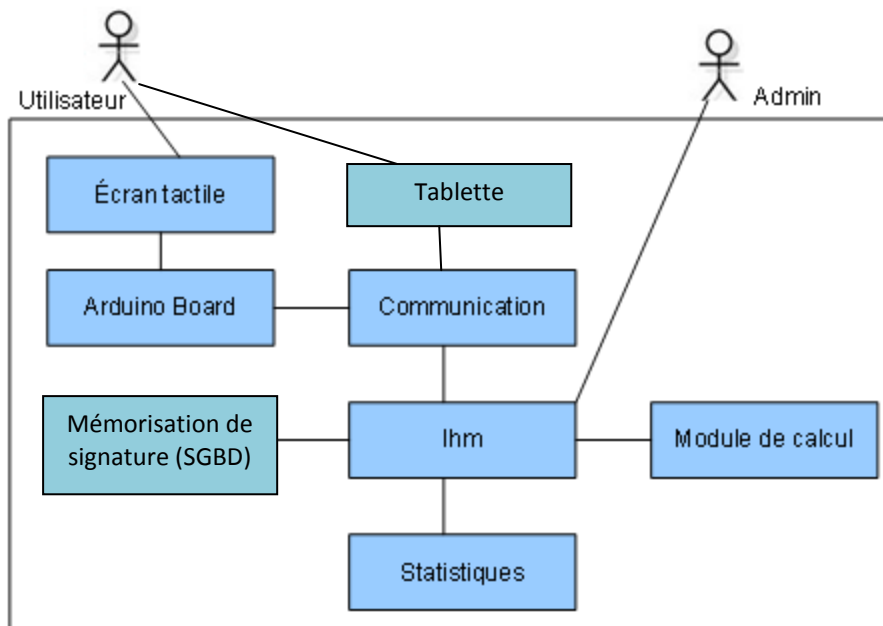
Diagramme de contexte



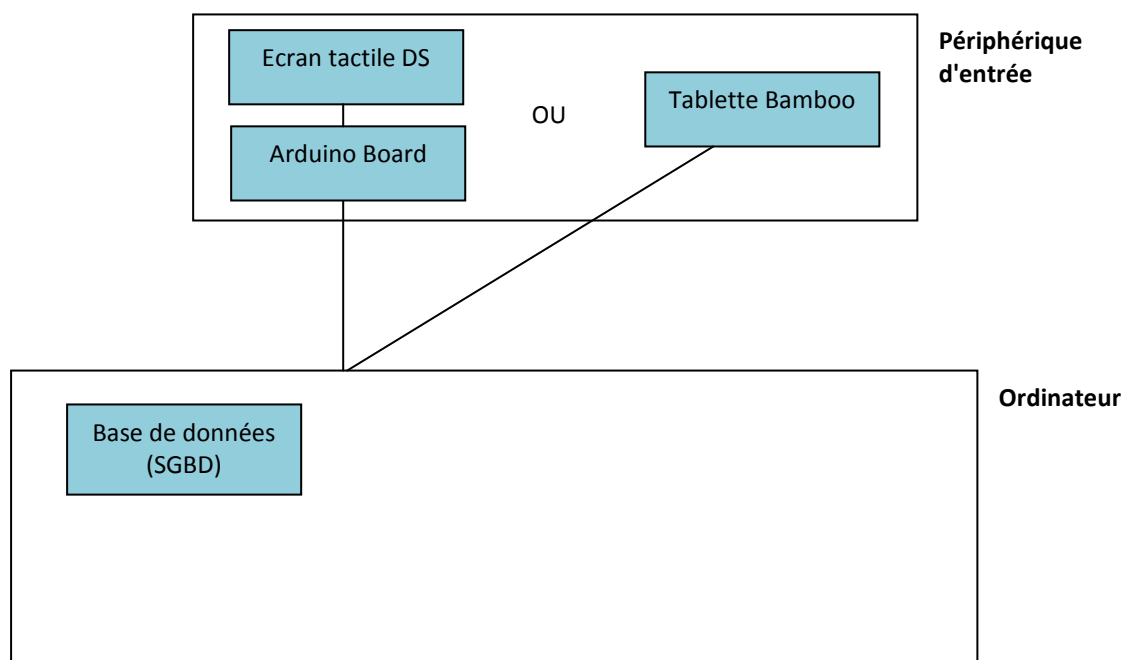
Diagramme de cas d'utilisation



Vue logique

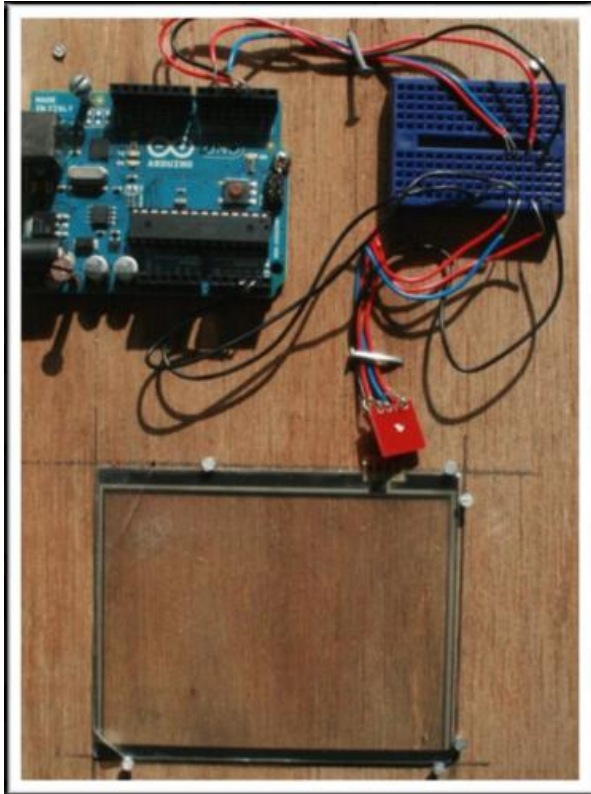


Vue physique



Mise en place du matériel

Montage Arduino-DS



Tablette Bamboo

Cette tablette est un ajout de matériel par rapport à l'année précédente. Ainsi, nous avons pour notre projet, deux interfaces pour l'acquisition des signatures. Cela nous a donc amené à étudier la tablette graphique et les bibliothèques Java permettant de nous en servir. L'IHM a également été modifiée pour permettre à l'utilisateur de choisir quelle interface d'acquisition il souhaite utiliser.

La bibliothèque permettant d'utiliser cette tablette est : Jpen

(http://sourceforge.net/apps/mediawiki/jpen/index.php?title=Main_Page)

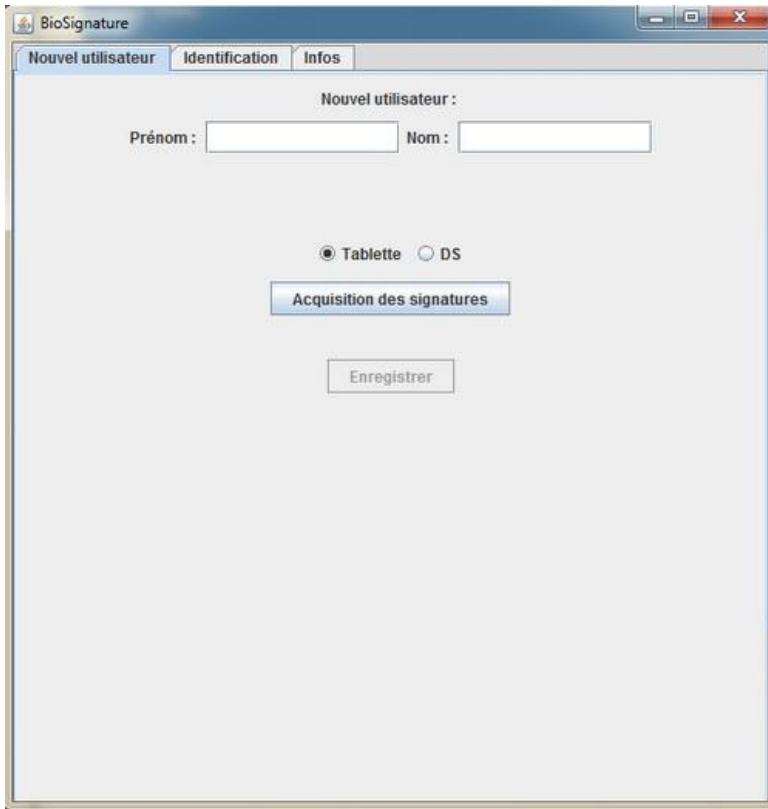
Le driver de la tablette CTH-460 est disponible ici :

http://www.wacom-asia.com/download/download_index.html



IHM

L'interface est composée de trois onglets. Lorsque l'utilisateur démarre l'application, celle-ci s'ouvre sur l'onglet "Nouvel utilisateur". Si il n'a pas encore de compte, il doit alors rentrer son nom, son prénom, et procéder à l'acquisition de cinq signatures (la même) afin de pouvoir s'enregistrer.

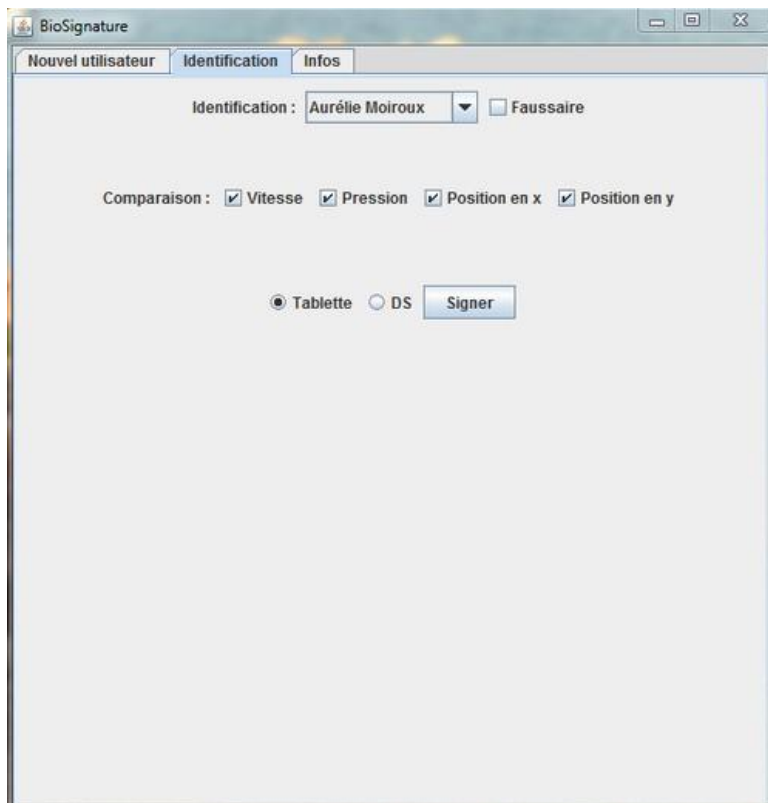


The screenshot shows a window titled "BioSignature" with three tabs: "Nouvel utilisateur", "Identification", and "Infos". The "Nouvel utilisateur" tab is active. The interface contains the following elements:

- Label: "Nouvel utilisateur :"
- Form fields: "Prénom : []" and "Nom : []"
- Radio buttons: "Tablette" and "DS"
- Buttons: "Acquisition des signatures" and "Enregistrer"

Une fois que l'utilisateur possède un compte, il peut se rendre sur l'onglet "Identification". Afin de s'authentifier l'utilisateur doit sélectionner son nom, puis signer de la même signature que lors de l'acquisition.

Nous avons ajouté une option "faussaire" sur cette onglet. Lorsque l'option est cochée, la signature du compte sur lequel l'utilisateur essaie de s'identifier est affiché sur l'écran de signature. Cette option a été mise en place pour tester d'avantage notre système. Ainsi nous pouvons voir si, avec le dessin de la signature, un imposteur peut réussir à s'identifier.

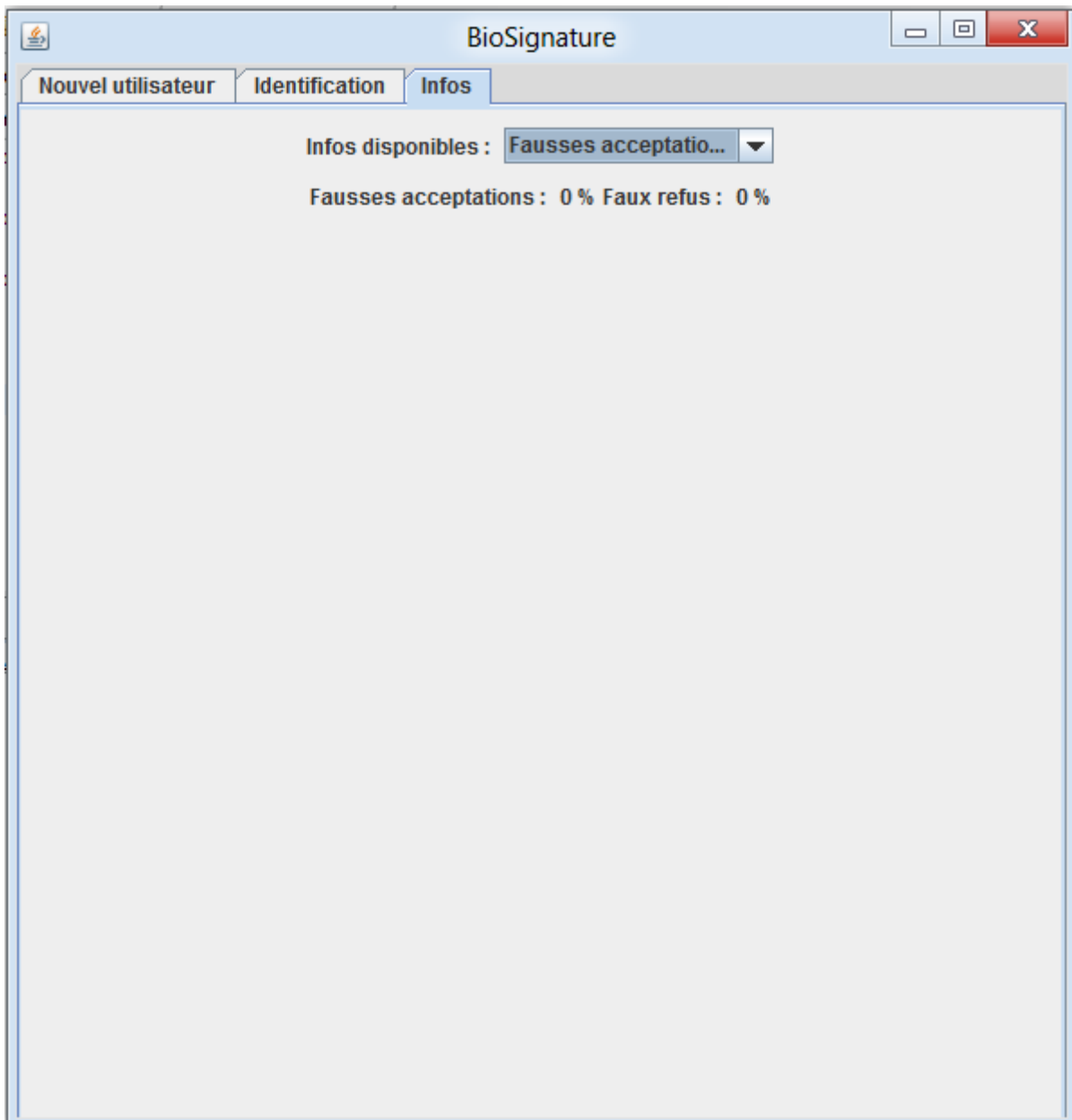


Le système affiche alors la réponse de l'authentification : la réussite ou l'échec. Une question apparaît également afin d'établir des statistiques de performance du système.

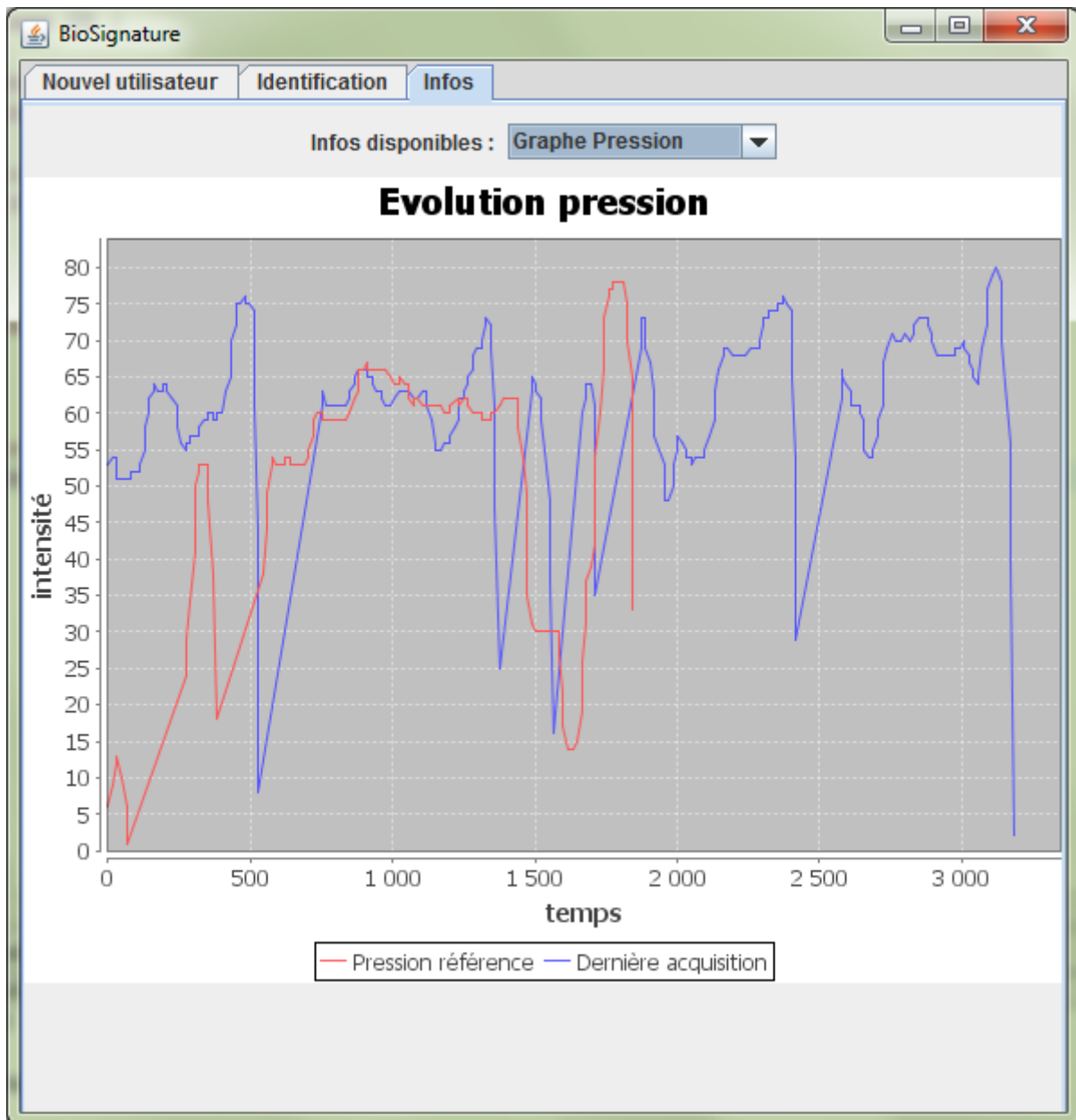


Onglet "Infos"

Lorsque l'utilisateur a tenté de s'identifier, il peut alors consulter différentes informations liées à sa tentative :



Le pourcentage de fausses acceptations et faux refus



Un graphe qui affiche l'évolution de la pression sur la signature référence de l'utilisateur et sur sa dernière acquisition

BioSignature

Nouvel utilisateur Identification Infos

Infos disponibles : **Arbre de Décision**

J48 pruned tree

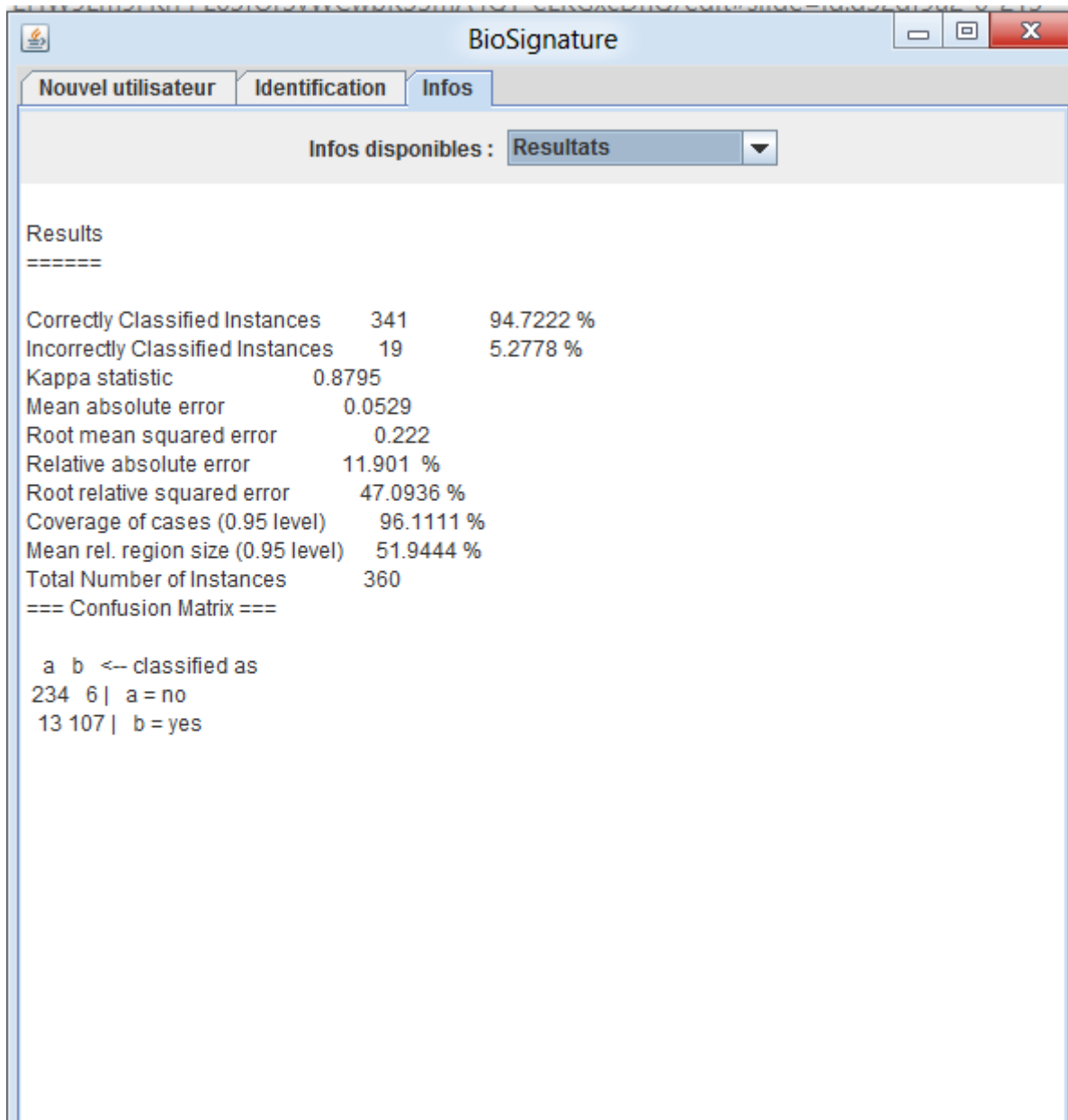
```

DTWY <= 28431.056674: yes (96.0)
DTWY > 28431.056674
| CORTX <= 0.958615
| | numberPenUp <= 4
| | | CORTY <= 0.921416
| | | | totalTime <= 662.399902: yes (6.0)
| | | | totalTime > 662.399902: no (3.0)
| | | | CORTY > 0.921416
| | | | DTWX <= 78822.720651
| | | | | CORTPressure <= 0.980004
| | | | | | CORTY <= 0.950637: yes (2.0)
| | | | | | CORTY > 0.950637: no (25.0)
| | | | | | CORTPressure > 0.980004: yes (3.0)
| | | | | DTWX > 78822.720651: no (65.0)
| | | numberPenUp > 4: no (136.0)
| | CORTX > 0.958615
| | | totalTime <= 145.857178: yes (9.0)
| | | totalTime > 145.857178
| | | | totalTime <= 1406: no (9.0)
| | | | totalTime > 1406
| | | | | numberPenUp <= 5: yes (4.0)
| | | | | numberPenUp > 5: no (2.0)

```

Number of Leaves : 12

Size of the tree : 23



Les informations du traitement algorithmique comme l'arbre de décision ou les résultats des différentes étapes de l'algorithme.

Algorithmes

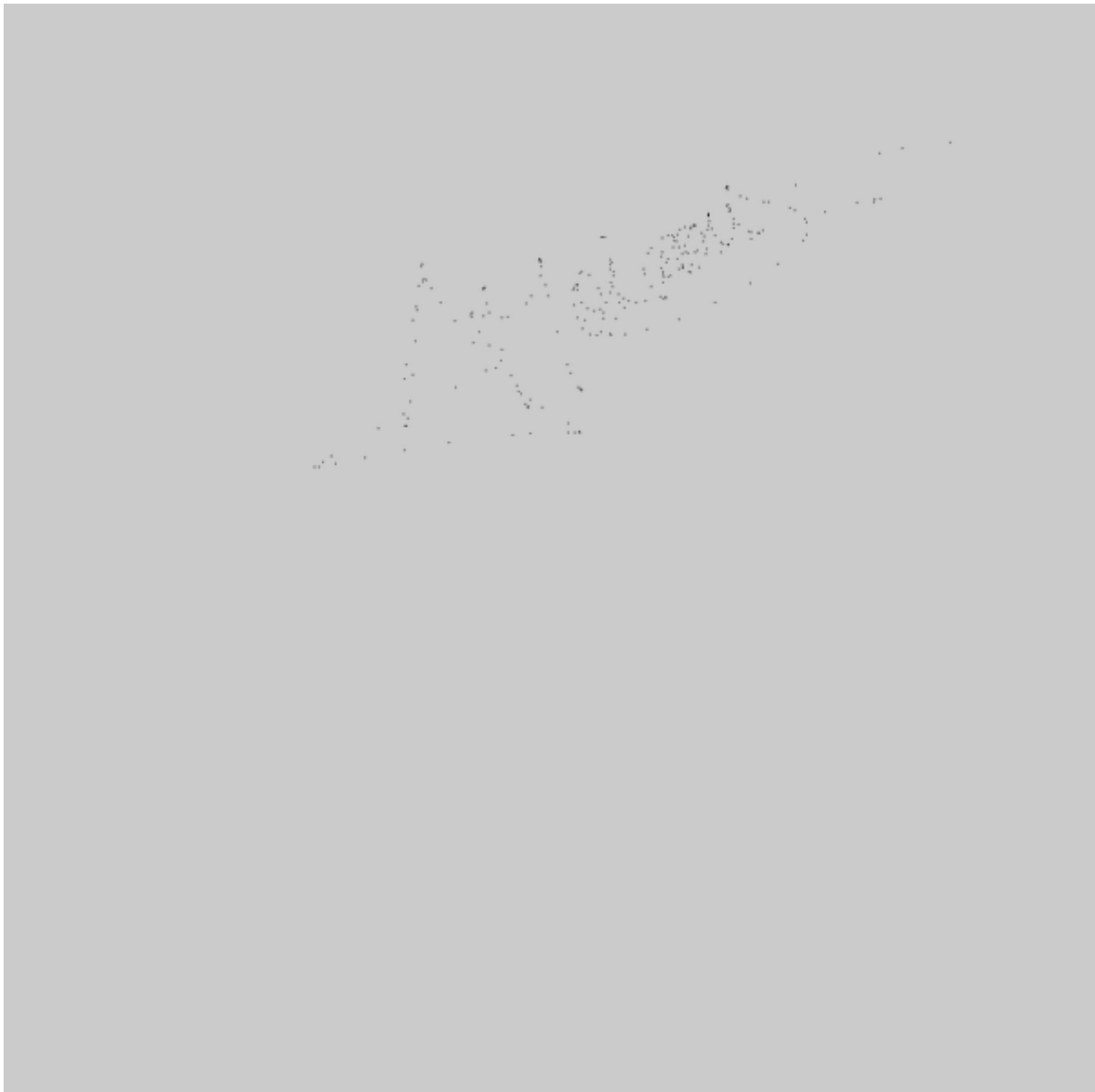
Un système de reconnaissance par signature nécessite l'application d'algorithmes, que ce soit dans un premier temps pour obtenir des acquisitions de qualité, qu'ensuite pour pouvoir les comparer de manière cohérente et enfin construire un modèle efficace de reconnaissance.

Prétraitement des acquisitions

Les acquisitions obtenues par l'intermédiaire de l'interface, pour représenter la signature d'un individu, sont des listes de points, auxquels sont rattachés une abscisse, une ordonnée, une valeur de pression et un temps d'acquisition. Cependant, il est nécessaire d'appliquer un prétraitement à ces acquisitions afin d'en augmenter le nombre de points, corriger les différences entre des signatures

successives (décalage, orientation et taille) et également obtenir un nombre de points fixe pour toutes les acquisitions pour pouvoir les comparer.

Voilà une image d'une acquisition originale :



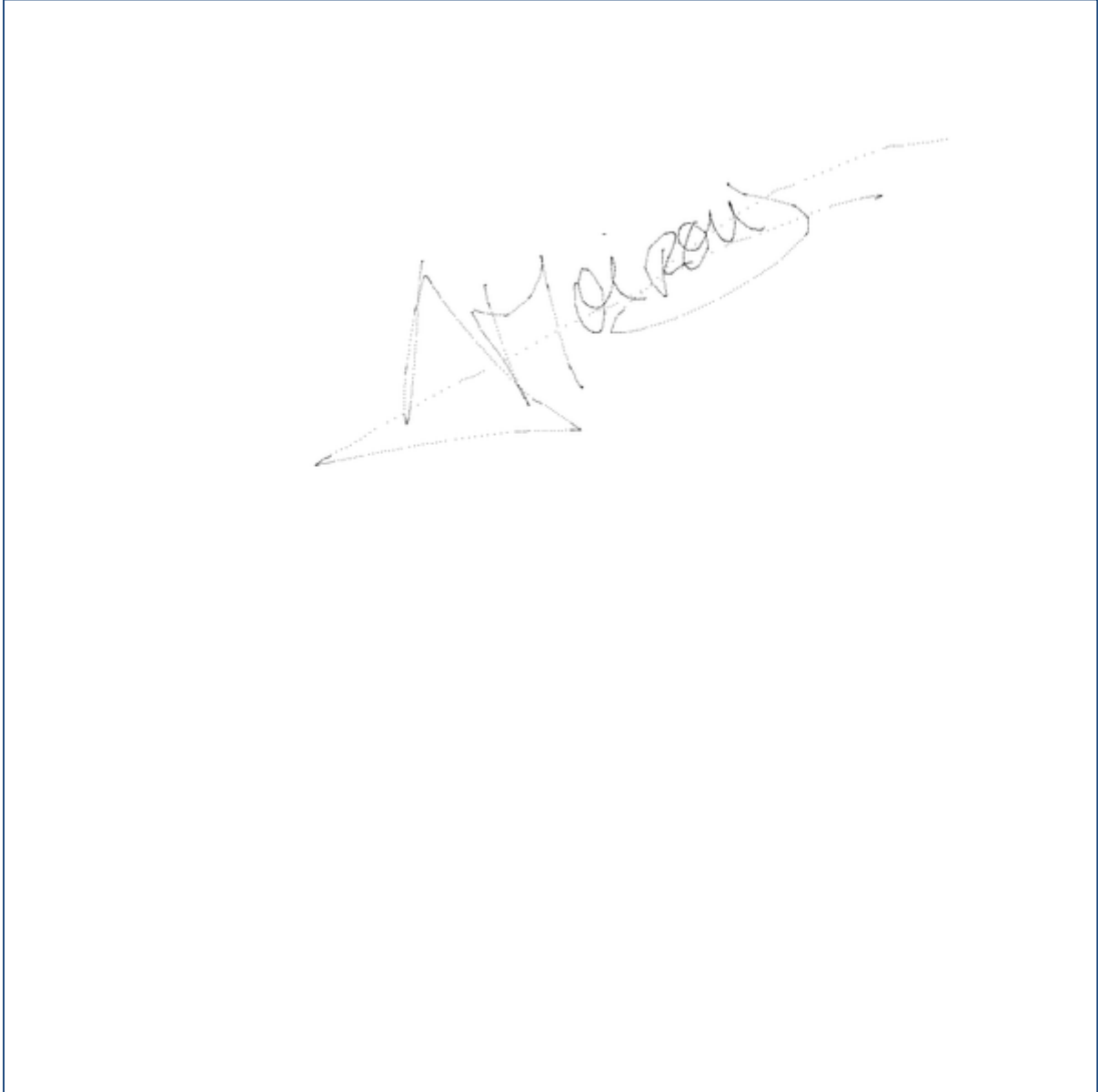
Ré-échantillonnage sur le temps

La première étape est propre à l'utilisation de la tablette Bamboo et consiste à ne conserver qu'un seul point par temps d'acquisition. En effet, le système d'acquisition de la tablette renvoie la liste de points acquis durant chaque période de 16ms et attache à chaque point cette période d'acquisition plutôt que le temps réel. Par exemple les points acquis en réalité aux temps 11ms et 14,5ms seront enregistrés avec un temps d'acquisition de 16ms alors qu'un point acquis au temps 16,8ms sera enregistré avec un temps d'acquisition de 32ms. Nous avons considéré que ce temps de 16ms étant relativement court, on ne conserverait qu'un seul point pour un même temps d'acquisition (par défaut le premier de la liste).

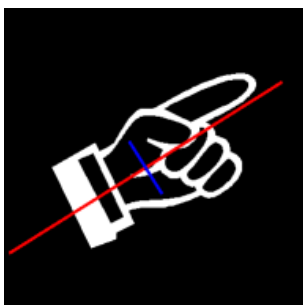
De plus, on a également considéré que deux points espacés d'un temps d'acquisition supérieur à 32ms correspondait à un levé de stylo.

Interpolation

On applique une interpolation cubique sur l'acquisition dans le but d'obtenir un rendu de signature plus continu et lisse, et mieux comprendre les courbures de la signature d'un individu. De plus, il est nécessaire de travailler sur des données assez fournies notamment pour des algorithmes de recouvrement par exemple.

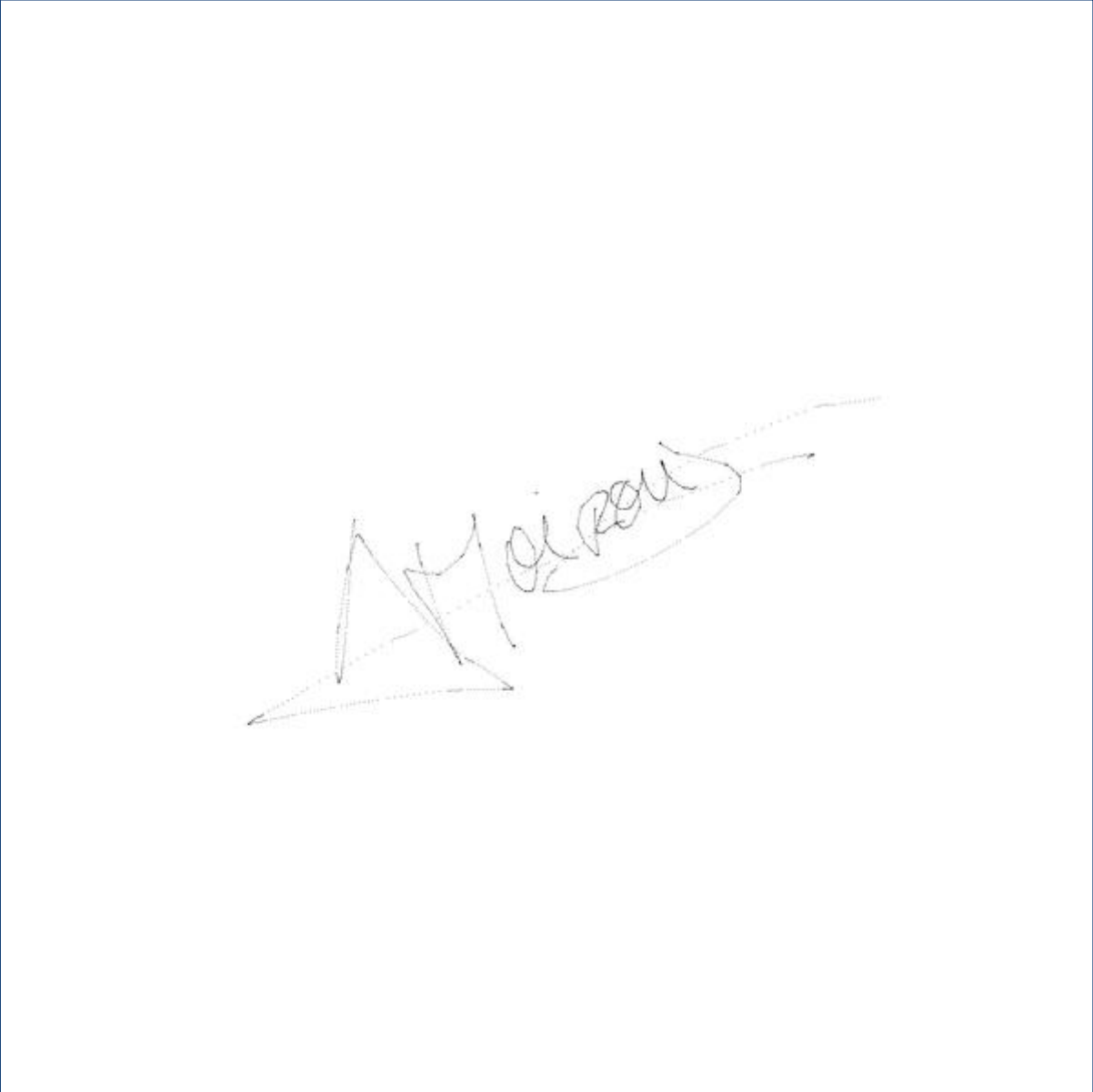


Analyse en composantes principales



L'ACP est utilisé pour trouver les informations mathématiques qui décrivent la position des points : l'axe rouge et l'axe bleu et le centre de gravité d'une forme binaire.

Centrage et Rotation





Pour réduire les problèmes de décalage, entre l'image à tester et le modèle, on utilise les informations fournies par l'ACP.

Le centre de gravité permet de centrer l'image comme le modèle.

Les deux axes qui décrivent l'alignement des points permettent de faire la rotation. Cette rotation est calculée à partir de l'angle entre l'axe principal et l'axe horizontal.

Normalisation (zoom)



Le zoom est calculé pour obtenir la taille la plus grande possible pour l'image.

La distance entre le centre et le point le plus proche du rebord de l'image évaluée = (distance max).

Chaque point subit alors une translation proportionnelle au rapport :

$$(distance\ du\ point/distance\ max) * (demi-largeur\ de\ l'image)$$

Ré-échantillonnage sur le nombre de points

Suivant la durée totale d'acquisition, le nombre de points acquis peut être plus ou moins important et les précédentes étapes de prétraitement peuvent en supprimer quelques-uns également. Il est donc essentiel d'effectuer un ré-échantillonnage sur les acquisitions après normalisation pour obtenir un nombre de points fixe (2000 points choisis actuellement) pour que les différents algorithmes de comparaison soient cohérents. On a choisi d'interpoler des points de manière uniforme si le nombre de points était inférieur à 2000 et d'en supprimer, également de manière uniforme, si ce nombre était supérieur.

Algorithmes de comparaison

Pour la création du modèle (voir plus loin), il est nécessaire d'obtenir un ensemble de données de comparaisons. On a ainsi sélectionné les comparaisons suivantes :

Comparaisons simplistes

- Différence de temps total d'acquisition
- Différence de nombre de lever de stylos
- Différence de pression moyenne

Coefficient de corrélation temporel

Cette méthode permet de calculer la ressemblance existante entre deux séries de données de même longueur, à savoir l'évolution des coordonnées en fonction du temps.

$$r_p = \frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}$$

Le Résultat varie entre 1 et -1.

Plus le résultat est proche de 0, moins la relation est forte entre les deux listes de données.

Dynamic Time Warping

Les données des signatures sont converties en graphe temporel. Par exemple : x/temps ; y/temps ; etc.

La méthode du DTW permet de calculer les dissimilarités entre les informations.

Ici, les deux listes permettent de calculer une matrice dont le dernier élément contient la distance minimale entre les deux listes d'informations.

Matrice de radiation

La matrice de radiation est une méthode de reconnaissance de forme.

Il existe deux possibilités pour calculer la matrice modèle :

1- Les points de la signature sont placés avec une valeur de 100. Autour de ces points, les cellules subissent une radiation qui modifie leur valeur suivant la distance entre la cellule courante et les points originaux de la signature. Chaque cellule de la matrice ne peut que subir le rayonnement du point original le plus proche.

2- De même que précédemment sauf que chaque cellule de la matrice subie la somme des rayonnements de tous les points originaux.

Après l'évaluation de cette matrice modèle, le calcul se fait de deux manières différentes :

1- Le résultat est la somme des valeurs enregistrées, dans la matrice modèle, qui sont aux coordonnées des points de la signature à tester.

2- Le résultat est le rapport du nombre de cellules non nulles communes entre la matrice modèle et la signature à tester et le nombre de cellules non nul total.

Somme des différences d'angles entre vecteurs

Cette méthode permet une reconnaissance de forme simple.

Elle permet d'obtenir l'angle d'écart entre la signature à tester et le modèle.

Le résultat est la somme des angles entre chaque vecteur et le vecteur horizontal. Chaque angle est compris entre $-\pi$ et π .

Génération d'un modèle classification

Afin de réaliser la tâche de vérification pour la reconnaissance de signature, nous avons opté pour le choix de WEKA (<http://www.cs.waikato.ac.nz/ml/weka>) pour la création du modèle par apprentissage et la décision finale.

Construction du fichier de comparaison (apprentissage.arff)

WEka utilise des fichiers ARFF pour construire et tester son modèle. Dans ce fichier on retrouve dans l'ordre:

Le nom de la relation (utile pour nommer le fichier d'apprentissage aux fichiers d'identification et de décision)

@relation signature

La liste des attributs de comparaison utilisés et leur type

@attribute totalTime numeric

@attribute numberPenUp numeric

...

@attribute CORTX numeric

@attribute CORTY numeric

@attribute pourcentageRadiance numeric

L'attribut contenant les différentes classes (2 dans notre cas : signatures équivalentes ou non)

@attribute accepte {no,yes}

Les données de comparaisons entre différentes signatures

@data

1019.466796875,2.0,0.9912756661011811,0.9970564473307176,17.837034225463867,yes

663.466796875,0.0,0.9837515744201278,0.996214642908734,15.106257438659668,yes

...

1278.466796875,0.0,0.9729839482086351,0.9951841501724467,9.820034980773926,no

1044.06640625,0.0,0.9859247460800329,0.9970413832907982,9.922861099243164,no

Pour construire ces données, on doit donc comparer des signatures équivalentes (d'une même personne) et des signatures différentes (de personnes différentes). Etant donné qu'un utilisateur enregistre 5 acquisitions pour former sa "signature", on peut donc réaliser 10 comparaisons entre signatures équivalentes par utilisateur. Ainsi pour n utilisateurs, on aura $5*n$ comparaisons de signatures équivalentes. Ensuite pour construire le modèle par apprentissage, il faut également comparer des signatures différentes. Pour cela, on effectue des comparaisons de manière aléatoire entre les signatures d'utilisateurs différents sans jamais refaire la même comparaison. On se permet de réaliser deux fois plus de comparaisons inter-utilisateur qu'intra-utilisateur (soit $10*n$ comparaisons).

Construction du modèle

Ensuite, on fournit ce jeu de données d'apprentissage à Weka qui construit son modèle selon le classifieur choisi. Pour cela, il construit un modèle à partir des 9/10 des données et teste sur le 1/10 restant. Il répète l'opération 10 fois pour parcourir toutes les données et améliorer son modèle au fur et à mesure.

Nous avons opté pour un arbre de décision (plus précisément le J48) pour classer nos données parmi les nombreuses méthodes possibles car il nous permet à posteriori de voir quelles sont les paramètres discriminants et pour quelles raisons il a décidé que 2 signatures se ressemblaient ou non.

Classification par arbre de décision

Une fois le modèle construit, on peut via l'interface tenter de s'identifier en sélectionnant la personne que l'on prétend être et en faisant une acquisition. Le programme va alors récupérer les cinq acquisitions représentant la signature de la personne supposée, effectuer le prétraitement de ces acquisitions, ainsi que de la nouvelle et les comparer comme pour la création du modèle.

On va ainsi créer un fichier ARFF "identification.arff" qui contiendra les données de comparaisons comme pour celui de l'apprentissage à la différence que le dernier attribut, celui de la classe, sera rempli d'un "?" pour signifier que c'est au modèle de décider à quelle classe associer cette comparaison (si les deux signatures se ressemblent ou pas).

Une fois que le modèle a rendu sa décision concernant ces cinq comparaisons, on a décidé que la personne avait réussie à s'identifier si la nouvelle acquisition était désignée comme ressemblante à au moins trois des cinq acquisitions enregistrées, pour cet utilisateur, dans la base de données.