HAL
open science

Sélection de courbes de la surface nasale pour l'authentification de personnes en utilisant Adaboost

Lahoucine Ballihi, Boulbaba Ben Amor, Mohamed Daoudi, Anuj Srivastava, Driss Aboutajdine

To cite this version:

Lahoucine Ballihi, Boulbaba Ben Amor, Mohamed Daoudi, Anuj Srivastava, Driss Aboutajdine. Sélection de courbes de la surface nasale pour l'authentification de personnes en utilisant Adaboost. RFIA, Jan 2010, Caen, France. pp.00-00. hal-00666262

HAL Id: hal-00666262

https://hal.science/hal-00666262v1

Submitted on 3 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sélection de courbes de la surface nasale pour l'authentification de personnes en utilisant Adaboost

L. Ballihi[1,2] B. Ben Amor[1,3] M. Daoudi[1,3] A. Srivastava[4] D. Aboutajdine[2]

[1] Laboratoire d'Informatique Fondamentale de Lille (UMR 8022), Villeneuve d'Ascq Cedex France.
[2] GSCM/LRIT, Faculté des Sciences, Université Mohammed V - Agdal, Rabat, Maroc.
[3] Institut TELECOM ; TELECOM Lille 1, Villeneuve d'Ascq Cedex France.
[4] Departement of Statistics, Florida State University, Tallahassee, FL 32306, USA.

Résumé

Nous proposons dans cet article d'étudier l'apport de chaque courbe de la région nasale, dans le cadre de la biométrie faciale 3D, en utilisant l'algorithme Adaboost. En effet, nous représentons les surfaces nasales par des collections de courbes fermées appelées courbes nasales puis de comparer celles-ci dans l'espace des courbes fermées en s'appuyant sur une analyse riemannienne de cet espace. En considérant qu'à chacune des courbes on peut associer un classifieur faible, nous proposons de construire un classifieur final basé sur l'algorithme Adaboost. Le boosting permet d'optimiser les performances individuelles de chacune des courbes. Les expérimentations sur un ensemble de données de la base FRGC v2 (Face Recognition Grand Challenge) montre l'amélioration nette des résultats d'authentification.

Mots Clef

Surfaces nasales, géométrie riemannienne, chemin géodésique, Adaboost, authentification de visages.

Abstract

The main contribution of this paper is a learning algorithm, based on AdaBoost, which build a strong classifier from a set of weak classifiers associated to level curves in the nasal region used for 3D face authentication. The basic idea is to represent nasal surfaces using indexed collections of iso-curves, and to analyze shapes of noses by comparing their corresponding curves. Adaboost considers each curve as a weak classifier and selects iteratively relevant curves to increase authentication accuracy. We demonstrate these ideas on a subset taken from FRGC v2 (Face Recognition Grand Challenge) database. The proposed approach increase authentication performances using Adaboost classifier compared to a simple fusion of scores from all curves.

Keywords

Nasal surfaces, Riemannian geometry, geodesic path, Adaboost, face authentication.

1 Introduction

Afin de répondre aux besoins liés à la sécurité, devenue aujourd'hui une préoccupation internationale, la biométrie se présente comme une technologie potentiellement puissante. En effet, les différents moyens biométriques visent à utiliser des caractéristiques comportementales et/ou physiologiques spécifiques à chaque personne. Ces caractéristiques présentent l'avantage d'être universelles, uniques, permanentes et qu'elles ne peuvent être falsifiées à la différence des moyens classiques tels que les mots de passe ou les badges. Les empreintes digitales et l'iris ont montré de bonnes performances. Cependant, ils exigent une coopération des utilisateurs qui les trouvent intrusifs. Par conséquent, la tendance actuelle se dirige vers les biométries qu'on peut collecter à distance, comme le visage. Bien que les approches 2D aient donné de bonnes performances, elles restent très sensibles aux problèmes de changements de conditions d'éclairage, de pose et d'expressions faciales.

L'introduction de la forme 3D du visage dans la reconnaissance de personnes est considérée comme une voie très prometteuse pour remédier à ces problèmes. Cependant, l'un des défis ouverts est celui de la robustesse de ces approches 3D aux expressions faciales. Afin de faire face à ces variabilités de la forme du visage, Faltmier et al. [1] proposent de recaler, en utilisant l'algorithme ICP, différentes parties du visage puis de les fusionner pour améliorer les performances biométriques. Bronstein et al. [2] proposent une nouvelle représentation basée sur la nature isométrique de la surface faciale (c'est à dire elle préserve les longueurs des géodésiques entre les points en présence de différentes expressions). Pour conserver la consistance des géodésiques en cas de bouche ouverte, les auteurs proposent de couper la partie de la surface correspondante à la bouche sur toutes les surfaces. Mpiperis et al. [3] proposent une paramétrisation polaire. Au sein de cette nouvelle paramétrisation intrinsèque, les attributs du visages sont invariants aux déformations isométriques. Quand la bouche est ouverte, cette paramétrisation n'est plus valable, les auteurs proposent donc de déconnecter les lèvres sur toutes

les surfaces faciales et changent cette paramétrisation en se basant sur trois points de références. Ter Haar et Veltkamp [4] proposent d'approximer la surface faciale par une collection de courbes radiales. Ils ne traitent pas le problème de déformations dues aux expressions faciales. Une autre catégorie d'approches proposent de restreindre l'étude à une partie du visage qui reste stable lors des expressions faciales, comme l'oreille [5, 6] ou le nez [7].

Dans le présent travail, nous nous intéressons à l'analyse géométrique de la forme de la région nasale. Elle consiste à comparer deux surfaces nasales à travers les formes des courbes nasales. L'idée de base est d'approximer grossièrement la surface nasale par un ensemble fini de courbes de niveau, appelées courbes nasales. En utilisant la géométrie riemannienne nous définissons la notion de chemin géodésique entre deux surfaces, et la distance entre elles. Des résultats expérimentaux sur la base FRGC v.2 [8] ont démontré tout l'intérêt de cette approche pour analyser la forme de la région nasale. Néanmoins, nous avons pu constater que toutes les courbes de la région nasale n'ont pas le même pouvoir discriminant. En effet, l'analyse du taux de reconnaissance des courbes nasales prises individuellement montre clairement que certaines courbes ont un très bon taux de reconnaissance. Dans ce papier nous proposons la sélection des courbes les plus pertinentes en utilisant l'algorithme Adaboost.

Le reste de cet article est organisé de la façon suivante : dans la section 2, nous présentons notre solution automatique de pré-traitement des données 3D. La section 3 détaille l'approche géométrique pour analyser la forme de la région nasale. Dans la section 4, nous proposons l'utilisation de l'algorithme Adaboost pour sélectionner les courbes les plus pertinentes. Enfin, nous présentons les résultats expérimentaux sur des données de la base FRGC v2 dans la section 5 et nous terminons par une conclusion et des perspectives.

2 Pré-traitement des scans 3D

La sortie des scanners 3D (basés sur un balayage de laser ou sur la projection d'une lumière structurée) nécessite une étape de pré-traitement. En effet, ces images de profondeur présentent plusieurs imperfections tels que des pics, des trous ou encore la présence de parties indésirables comme les cheveux ou les habits.

La figure 1 illustre notre pipeline d'étapes afin de corriger ces imperfections par l'élimination des pics, le bouchage des trous et la segmentation de la partie utile du visage. D'abord, un filtre médian 2D est passé sur l'image de profondeur en entrée afin d'éliminer le bruit. Ce bruit se présente sous forme de pixels incohérents avec leurs voisinages. En suite, nous appliquons une interpolation bidimensionnelle pour boucher les trous dues à l'absorption du laser ou de la lumière structurée par des régions sombres du visage comme l'iris, les sourcils ou tout simplement quand la bouche est ouverte. A partir de l'image

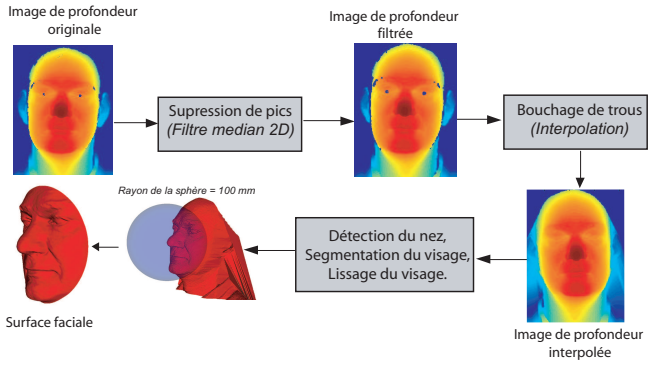

FIGURE 1 – Pré-traitement des images de visages 3D.

obtenue, nous générons le maillage 3D sur lequel nous détectons le bout du nez. Enfin, nous segmentons la partie utile du visage en récupérant la partie à l'intérieur de la sphère ayant comme centre le bout du nez et comme rayon $100mm$. Le pré-traitement est suivie d'une étape d'extraction de courbes 3D fermées que nous appelons des courbes faciales. Dans la suite nous détaillerons l'étape de localisation du bout du nez puis l'étape d'extraction de ces courbes de niveau.

– **Détection du bout du nez** : Comme l'illustre la figure 2, la détection du bout du nez nécessite une première coupe transverse de la surface faciale au niveau de son centre de masse. La sortie de cette coupe est une courbe (ou profil) horizontale, nous localisons le point de cette courbe pour lequel la profondeur Z est minimale. Nous utilisons ce point pour effectuer une seconde coupe sur la surface faciale, cette fois sagittale, qui donnera naissance à une courbe verticale. Le minimum de profondeur sur un intervalle de cette courbe est interprété comme le bout du nez. Cette procédure simple permet de détecter correctement à hauteur de $99,1\%$ le bout du nez sur l'ensemble des images de la base FRGC v2 (4007 images 3D).

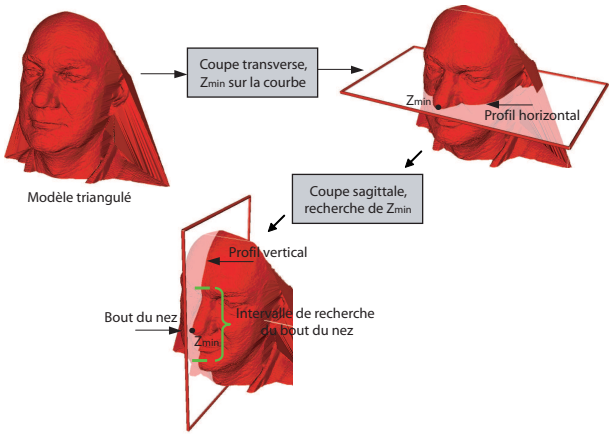

FIGURE 2 – Procédure de localisation du bout du nez.

– **Extraction des courbes de niveau** : Nous commençons par considérer les surfaces faciales comme des surfaces continues bien qu'elles soient en réalité formées de données discrètes. Soit S une surface faciale et $D : S \longrightarrow \mathbb{R}^+$ une fonction continue définie par $D(p) = DistGeo(r, p)$, où $DistGeo$ représente la distance géodésique sur S entre r et p. Le point r représente le point de référence, c'est à dire le pic du nez dans notre cas. Notons par c_λ la courbe de niveau $\lambda \in D(S)$ de la fonction D, c'est-à-dire. $c_\lambda = \{p \in S | D(p) = \lambda\} \subset S$. Ces courbes seront utilisées pour caractériser la forme de S, en effet la surface S peut être théoriquement totalement reconstruite à partir de ces courbes. Ce choix de fonction basée sur la distance géodésique est motivé par la robustesse de cette distance aux déformations faciales. Cette idée a été introduite pour la première fois dans [2] dans un contexte de reconnaissance faciale 3D. La figure 3 illustre queleques exemples de surfaces nasales extraites à partir de maillage triangulé avec les courbes de niveau de la fonction $DistGeo$.

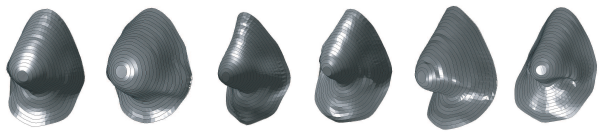

FIGURE 3 – Exemples de surfaces nasales extraites à partir de maillage triangulé avec les courbes de niveau de la fonction $DistGeo$.

A ce stade nous représentons donc une surface nasale quelconque par une collection indexée de courbes de niveau qui capturent localement la forme de la surface. L'union de ces courbes forme la surface. Nous allons donc nous intéresser, dans un premier temps, à l'analyse de la forme de ces courbes puis présenter une extension pour les surfaces nasales.

3 Analyse riemannienne des surfaces nasales

Ces dernières années ont vu le développement de plusieurs approches pour décrire les formes des courbes 2D. Nous pouvons citer, les approches basées sur les descripteurs de Fourier, celles basées sur les moments ou celles basées sur les axes médians. Cependant, il semble que aucune de ces méthodes ne s'est imposée ni du point de vu conceptuelle ni de point de vue calculatoire, pour décrire la forme des surfaces. Une approche très récente et prometteuse, consiste à considérer l'espace des formes à comparer comme un objet géométrique à part entière qui s'apparente à une variété riemannienne de dimension infinie sur lequel nous pourrons avoir recours aux outils "classiques" de la géométrie riemannienne. Les résultats récents de Michor et Mumford [9] et Klassen et al. [10] dans le cas des courbes planes montrent toute la puissance de cette approche. Joshi et al. [11] ont proposé récemment une généralisation de ces travaux au cas des courbes définies dans $\mathbb{R}^n$. Nous allons adopter ces derniers travaux à notre problème.

3.1 Analyse des courbes fermées dans $\mathbb{R}^3$

Nous commençons par considérer que les courbes nasales β sont des courbes fermées dans $\mathbb{R}^3$. Elles sont paramétrées par $\beta : \mathbb{S}^1 \to \mathbb{R}^3$, en supposant que la courbe est non singulière, en d'autres termes $\|\dot{\beta}(t)\| \neq 0$ pour tout t, la norme utilisée ici est la norme euclidienne de $\mathbb{R}^3$. Il est à noter que nous supposons que la courbe n'est pas paramétrisée par l'abscisse curviligne. Nous permettons une classe plus large des paramétrisations afin d'améliorer l'analyse de la forme des courbes fermées dans $\mathbb{R}^3$. Pour analyser la forme de la courbe β, nous utilisons la paramétrisation par la fonction *SRVF (Square Root Velocity Function)* suivante :

$$q(t) \doteq \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}} \,. \tag{1}$$

$q(t)$ est alors une fonction spéciale qui capture la forme de β. La métrique classique élastique pour comparer formes des courbes devient une métrique dans L_2 en vertu de cette représentation [11]. Des idées similaires ont été présentées par Younes [12]. Nous définissons l'ensemble des courbes fermées dans $\mathbb{R}^3$ par :

$$\mathcal{C} = \{q : \mathbb{S}^1 \to \mathbb{R}^3 | \int_{\mathbb{S}^1} q(t)\|q(t)\|dt = 0\} \subset \mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^3) \,. \tag{2}$$

Où $\mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^3)$ désigne l'ensemble de toutes les fonctions intégrables de $\mathbb{S}^1$ à $\mathbb{R}^3$. La quantité $\int_{\mathbb{S}^1} q(t)\|q(t)\|dt$ désigne le déplacement total dans $\mathbb{R}^3$ tout en se déplaçant depuis l'origine de la courbe jusqu'à la fin. Lorsqu'elle est nulle, la courbe est fermée. Ainsi, $\mathcal{C}$ représente l'ensemble de toutes les courbes fermées de $\mathbb{R}^3$. Cet ensemble est une variété non linéaire vue la propriété non linéaire de fermeture. Elle sera munie alors d'une structure riemannienne en utilisant le produit scalaire suivant : pour deux vecteurs tangents $u, v \in T_q(\mathcal{C})$, nous définissons :

$$\langle u, v \rangle = \int_{\mathbb{S}^1} \langle u(t), v(t) \rangle \, dt \,. \tag{3}$$

Pour tout $q \in \mathcal{C}$, l'espace tangent est définit par :

$$T_q(\mathcal{C}) = \{v : \mathbb{S}^1 \to \mathbb{R}^3 | \, \langle v, w \rangle = 0, \; w \in N_q(\mathcal{C})\} \,,$$

Où $N_q(\mathcal{C})$ désigne l'espace des vecteurs normaux à q. Les courbes sont ainsi représentées dans l'espace de Hilbert ce qui permet de profiter des propriétés de cet espace, voir [11]. En outre, pour définir une forme, la représentation doit être indépendante des différentes rotations et reparamétrisations. Ce qui se traduit mathématiquement par un quotionnement par rapport au groupe des rotations $SO(3)$ et le groupe des reparamétrisations Γ.

Nous définissons alors les orbites du groupe de rotation $SO(3)$ et le groupe de re-paramétrisation Γ comme classes d'équivalence dans $\mathcal{C}$. Les éléments de l'ensemble :

$$[q] = \{\sqrt{\dot{\gamma}(t)}Oq(\gamma(t))|O \in SO(3), \ \ \gamma \in \Gamma\} \ ,$$

sont alors équivalents à la même forme q, et l'ensemble de ces classes d'équivalence hérite la structure riemannienne de $\mathcal{C}$ et représente alors notre espace d'étude. Il est noté : $\mathcal{S} \doteq \mathcal{C}/(SO(3) \times \Gamma)$.

L'objectif de notre approche est la construction d'un chemin géodésique entre deux éléments de $\mathcal{S}$, sous la métrique riemannienne donnée par l'équation 3. Étant données deux courbes β_1 et β_2 représentées par leurs SRVF respectives q_1 et q_2, nous cherchons à trouver un chemin géodésique entre les orbites $[q_1]$ et $[q_2]$ dans l'espace $\mathcal{S}$. Nous utilisons, à ce propos, une méthode numérique, appelée méthode de *dressage* qui consiste à relier les deux points $[q_1]$ et $[q_2]$ par un chemin arbitraire α puis mettre à jour ce chemin itérativement dans le sens négatif du gradient de son énergie donnée par : $E[\alpha] = \frac{1}{2}\int_s \langle \dot{\alpha}(s), \dot{\alpha}(s)\rangle \, ds$. Il a été démontré dans [13] que les points critiques de E sont des chemins géodésiques dans $\mathcal{S}$. Nous noterons $d([q_1],[q_2])$ la distance géodésique ou la longueur de la géodésique dans $\mathcal{S}$, entre deux représentations β_1 and β_2.

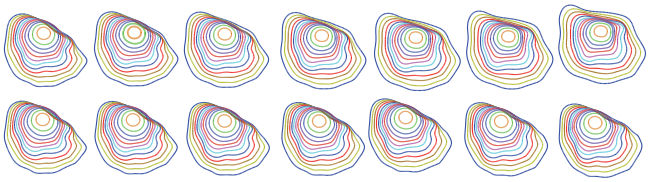

FIGURE 4 – Quelques exemples de chemins géodésiques entre courbes nasales

La figure 4 illustre deux exemples de chemins géodésiques entre des courbes nasales. La première ligne illustre un chemin géodésique entre des courbes nasales de deux personnes différentes (chemin inter-classe), alors que la deuxième ligne présente le chemin entre deux ensembles de courbes de deux sessions différentes d'une même personne (chemin intra-classe).

3.2 Analyse des surfaces

Dans cette section, nous étendons notre étude des formes des courbes nasales aux formes des surfaces formées par la région nasale. Nous représentons la surface de la région nasale N par une collection de courbes nasales :

$$N \leftrightarrow \{c_\lambda, \lambda \in [0, \lambda_0]\} \ ,$$

où c_λ représente la courbe de niveau associée à $DisGeo = \lambda$. Chaque région nasale est représentée alors comme un élément de $\mathcal{C}^{[0,\lambda_0]}$. Dans notre approche, les formes de deux régions nasales sont comparées par leurs courbes nasales correspondantes. Pour deux régions nasales N^1 et N^2, et leurs courbes nasales $\{c^1_\lambda, \lambda \in [0, \lambda_0]\}$ et $\{c^2_\lambda, \lambda \in [0, \lambda_0]\}$ respectivement.

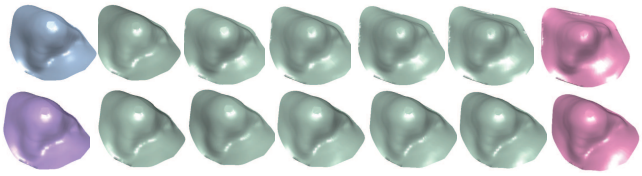

FIGURE 5 – Quelques exemples de chemins géodésiques entre surfaces nasales.

La figure 5 illustre deux exemples de chemins géodésiques entre deux surfaces nasales. La première ligne illustre un chemin géodésique entre des surfaces nasales de deux personnes différentes, alors que la seconde ligne représente un chemin géodésique entre des surfaces nasales de deux sessions différentes d'une même personne.

4 Classification pour l'authentification de visages 3D

Nous pouvons à ce stade combiner les résultats issus de toutes les courbes de niveau en les considérant comme des classifieurs séparés. Nous proposons d'utiliser l'algorithme Adaboost, introduit par Freund et Schapire dans [14], pour apprendre un **classifieur final** *(strong classifier)* basé sur une sélection pondérée de **classifieurs faibles** *(weak classifier)* qui sont les courbes de niveau. Le boosting permet alors d'optimiser leurs performances. Adaboost repose sur la sélection itérative de classifieurs faibles en fonction d'une distribution des exemples d'apprentissage. A chaque itération, le classifieur fourni est pondéré par la qualité de sa classification : plus il classe bien, plus il sera important.

Adaboost a été utilisé avec succès sur des applications aussi diverses que la détection de visages [15] où il fournit un classifieur final binaire (par exemple visage/non-visage). Pour cela, nous commençons d'abord par formuler le problème d'authentification de personnes comme un problème de classification binaire.

4.1 Formulation du problème

L'authentification ou la vérification de personnes permet de confirmer une identité proclamée ou de l'infirmer. Un algorithme d'authentification doit alors répondre à la question suivante : «Suis-je réellement la personne que je suis en train de proclamer ?». Un *imposteur* est une personne qui se fait passer pour une autre personne en présentant son identité. En revanche, un *client* est une personne qui présente sa vraie identité. Un algorithme d'authentification fiable doit être capable de détecter les imposteurs et de laisser passer les clients. Nous proposons de formuler le problème d'authentification comme un problème de classification binaire. Nous considérons les deux classes : Accès imposteur et Accès client.

– **Accès imposteur** : il s'agit de comparer des biométries appartenant à la même classe de personne.

– **Accès client** : il s'agit de comparer des biométries de deux personnes différentes.

Disposant des scores de similarités entre les courbes de même niveau, notre objectif est de construire un classifieur final qui minimise le taux de fausses acceptations *(False Accept Rate (FAR))* c'est-à-dire le taux d'imposteurs et minimise le taux de faux rejets *(False Reject Rate (FRR))*, c'est-à-dire le taux de clients rejetés.

4.2 Adaboost pour la classification

L'algorithme Adaboost nécessite deux phases : la phase d'apprentissage et la phase de test. La phase d'apprentissage requiert un ensemble des exemples d'apprentissage contenant des observations x_n appartenant à un ensemble χ avec $\chi = \{x_n\}$, de type "Accès imposteur" et "Accès client". Bien sûr, les observations utilisées pour ces deux phases sont complètement disjointes.

Nous utilisons pour l'apprentissage et pour le test un sous-ensemble de la base FRGC v2 [16] constitué de 2000 scans de 209 personnes différentes. Chaque personne est présente avec plusieurs sessions avec des expressions différentes. Nous décomposons cet ensemble en deux sous-ensembles dont un pour la phase d'apprentissage (1052 sessions) et le second pour la phase de test (948 sessions). Nous calculons ensuite les matrices de similarité pour ces deux ensembles pour chaque courbe de niveau. Une matrice de similarité est constituée de scores de similarité entre deux sessions d'une même personne (accès client) et de scores entre deux sessions de personnes différentes (accès imposteur). Quelques sessions de la base FRGC v2 sont illustrées par la figure 6.

FIGURE 6 – Quelques exemples prétraités de sessions d'une même personne de la base FRGC v2.

Nous disposons ainsi d'un ensemble de couples $(x_n^k, y_n)_{1 \leq n \leq N}$ où $(x_n^k)_{1 \leq k \leq \lambda}$ est un score de similarité entre deux courbes de même niveau k et y_n peut prendre deux valeurs 0 s'il s'agit d'un accès imposteur et 1 dans le cas d'un accès client. Pour chaque caractéristique (courbe c_j), le classifieur faible permet de déterminer le seuil optimal pour la classification, de sorte que le nombre d'exemples mal classés soit minimisé. Ce classifieur désigné par $h_j(x)$ est associé à la courbe c_j avec un seuil θ.

$$h_j(x_n^k) = \begin{cases} 1 & \text{si } x_n^k > \theta \text{ (Client)} \\ 0 & \text{sinon. (Imposteur)} \end{cases}$$

On note $W_t = \{w_{t,n}\}_{1 \leq n \leq N}$ l'ensemble des poids à l'itération t et h_t désigne les hypothèses faibles, donné par :

$$h_t : \chi \rightarrow \{0, 1\}$$

Le pseudo-code de l'algorithme AdaBoost est défini par l'algorithme 1 :

Algorithm 1 L'algorithm AdaBoost

– Entrées : L'ensemble des exemples $(x_1^k, y_1), .., (x_N^k, y_N)$ avec x_i^k est le score de similarité selon la courbe de niveau k ($1 \leq k \leq \lambda$) et $y_n = \{0, 1\}$.
– Soit m le nombre d'accès imposteurs et l le nombre d'accès clients. L'initialisation des poids $w_{1,n} = \frac{1}{2m}, \frac{1}{2l}$ dépend de la valeur de y_n avec $n \in 1..N$.
– Pour $t = 1, ..., T$:

1- Normaliser les poids $w_{t,n}$ sachant que $\sum_{n=1}^{N} w_{t,n} = 1$.
2- Pour toute courbe c_j (feature), apprendre le classifieur h_j qui utilise une seule courbe. L'erreur ϵ_j de classifieur h_j est determinée avec le poids correspondant $w_{t,1}, ..., w_{t,N}$:

$$\epsilon_j = \sum_{n=1}^{N} w_{t,n} |h_j(x_n^j) - y_n|$$

3- Choisir le classifieur h_t avec l'erreur la plus faible.
4- Mis à jour des poids $w_{t+1,n} = w_{t,n} \beta_t^{1-e_n}$, avec $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ et $e_n = 0$, si l'exemple x_n est correctement classifié par h_t et 1 sinon.

– Le classifieur final est donné par :

$$H(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^{T} log \frac{1}{\beta_t} h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} log(\frac{1}{\beta_t}) ; \\ 0 & \text{sinon.} \end{cases}$$

La sortie de l'algorithme Adaboost est un classifieur H qui est une somme pondérée des classifieurs faibles h_t sélectionnés à l'itération t.

5 Résultats expérimentaux

Comme décrit dans la section 4, nous utilisons deux sous-ensembles de la base FRGC v2 dont un ensemble pour apprendre l'algorithme Adaboost et un second pour évaluer les résultats de classification. Nous nous servirons des taux d'évaluation classiques utilisés en biométrie et particulièrement dans le scénario d'authentification à savoir le *VR (Verification Rate)* ou le taux de vrais positifs - le *FAR (False Accept Rate)* ou le taux de faux positifs - *FRR (False Reject Rate)* ou le taux de faux négatifs - *TRR (True Reject Rate)* ou le taux de vrais négatifs. Les scores de similarités utilisés sont extraits aléatoirement de la matrice d'apprentissage (ou de test) pour la phase d'apprentissage (ou pour la phase de test).

Une comparaison des résultats avec la classification donnée par la moyenne arithmétique d_{MA}, c'est-à-dire le cumul des distances entre les courbes de même niveau divisé par le nombre de courbes utilisées. d_{MA} est définie par :

$$d_{MA} : \mathcal{C}^{[0,\lambda_0]} \times \mathcal{C}^{[0,\lambda_0]} \rightarrow \mathbb{R}_{\geq 0},$$
$$d_{MA}(N^1, N^2) = \frac{1}{\lambda_0} \sum_{\lambda=1}^{\lambda_0} d(c_\lambda^1, c_\lambda^2).$$

Cette comparaison nous permet de quantifier l'amélioration des performances par l'utilisation de l'algorithme

Adaboost. Dans son critère de classification final, Adaboost fournit des pondérations des classifieurs associés aux courbes en fonction de leurs qualités de classification. A la différence du critère de la fonction d_{MA} où la même pondération est donnée à chacune des courbes de niveau. Autrement dit, la classification fournie par la fonction d_{MA} servira comme une approche de base ou de référence *(Baseline)*.

5.1 Résultats sur la base d'apprentissage

Le tableau 1 présente quelques résultats obtenus sur l'ensemble d'apprentissage. Les valeurs m et l représentent le nombre d'observations en accès client et en accès imposteurs respectivement. Chaque ligne du tableau présente les résultats sous forme de taux (*VR*, *FAR*, *FRR* et *TRR*) pour une expérimentation donnée. Nous avons effectué différentes expériences, en tirant d'une manière aléatoire un échantillon d'accès clients et d'accès imposteurs. Pour chaque expérience on calcule les valeurs *VR*, *FRR*, *FAR* et *TRR* ainsi que leurs moyennes. Nous pouvons clairement constater l'amélioration des taux *VR* et *TRR* pour les différentes expérimentations en comparaison avec les résultats donnés sur les mêmes échantillons produits par la moyenne arithmétique. Par exemple, Adaboost (Ad) donne une taux de vérification moyen de 97.4% devant 68.06% donné par la moyenne arithmétique (d_{MA}) pour une valeur de $m = l = 5000$ d'observations.
Afin d'accélérer la phase d'apprentissage nous avons pris des échantillons aléatoirement choisis, de la base d'apprentissage, de tailles variant de $m = l = 2000$ à $m = l = 5000$ scores pour les accès client et les accès imposteur. Comme le montre le tableau 1, les performances du classifieur sur ces échantillons ne changent quasiment pas (*VR* au tour de 97%). Cela veut dire qu'un échantillon de taille 2000 accès client et 2000 accès imposteurs suffirait pour achever la phase d'apprentissage et produire le classifieur final.

Même si ces résultats ne sont pas significatifs, puisqu'on travaille sur la base d'apprentissage, ils montrent que le classifieur, basé sur Adaboost, améliore significativement les performances par rapport à la moyenne arithmétique. Une analyse plus fine de ce classifieur, montre que les classifieurs faibles h_t associés aux courbes 8, 3 et 7 présentent les poids les plus forts dans le classifieur final. Ce qui montre que ces courbes sont les plus pertinentes sur le visage pour faire de l'authentification. La figure 7 montre la localisation de ces courbes sur le visage. Celles-ci ne passent pas par les narines et donc leurs formes ne sont affectées par les cavités présentes souvent sur les narines. Ces cavités engendrent un changement des formes des courbes passant par celles-ci ce qui affecte le calcul des distances. En particulier, les distances intra-classes où on compare deux surfaces nasales appartenant à la même personne.
Nous avons calculé les distances entre toutes les courbes deux-à-deux, pour différentes configurations de courbes calculés pour 2000 accès clients et 2000 accès imposteurs.

Adaboost (Ad) et Moyenne arithmétique (d_{MA})								
m, l	*VR (%)*		*FAR (%)*		*FRR (%)*		*TRR (%)*	
	Ad	d_{MA}	Ad	d_{MA}	Ad	d_{MA}	Ad	d_{MA}
2000	98.65	66.85	4.45		1.35	33.15	95.55	
2000	98.45	67.55	6.75		1.55	32.45	93.25	
2000	95.15	59.05	0.40		4.85	40.95	99.60	
2000	96.35	61.05	1.70		3.65	38.95	98.30	
2000	97.35	62.30	2.25		2.65	37.70	97.75	
Moy	**97.19**	**63.36**	3.11		**2.81**	**36.64**	96.89	
3000	96.57	65.67	2.37		3.43	34.33	97.63	
3000	96.67	65.97	2.53		3.33	34.03	97.47	
3000	98.60	72.67	6.53		1.40	27.33	93.47	
3000	98.70	71.23	5.87		1.30	28.77	94.13	
3000	98.10	69.73	4.10		1.90	30.27	95.90	
Moy	**97.73**	**69.05**	4.28		**2.27**	**30.95**	95.72	
4000	97.20	69.50	3.58		2.80	30.50	96.43	
4000	97.65	68.53	3.93		2.35	31.48	96.08	
4000	95.53	66.50	2.83		4.48	33.50	97.18	
4000	98.38	72.00	4.83		1.63	28.00	95.18	
4000	97.83	67.28	3.20		2.18	32.73	96.80	
Moy	**97.32**	**68.76**	3.67		**2.69**	**31.24**	96.33	
5000	98.30	68.92	4.06		1.70	31.08	95.94	
5000	97.58	69.44	3.58		2.42	30.56	96.42	
5000	97.72	68.70	2.84		2.28	31.30	97.16	
5000	97.58	67.64	2.38		2.42	32.36	97.62	
5000	95.82	65.60	1.94		4.18	34.40	98.06	
Moy	**97.40**	**68.06**	2.96		**2.60**	**31.94**	97.04	

TABLE 1 – Résultats de la vérification sur l'ensemble d'apprentissage Ad : résultats donnés en utilisant Adaboost, d_{MA} : résultats donnés par la moyenne arithmétique

La figure 8 montre quatre combinaisons : (a) courbes 3, 7 et 8, (b) courbes 1,6 et 9 - (c) courbes 2, 4 et 5 - (d) 3, 4 et 10. Dans cette figure, un point représente le triplet $(d(c_3, c_3), d(c_7, c_7), d(c_8, c_8))$. On peut observer que les ensembles d'accès imposteurs et d'accès clients sont plus séparés dans le cas de mesures de similarités données par les courbes 8, 3 et 7 par rapport à d'autres combinaisons de courbes. On peut dire donc que l'algorithme Adaboost a bien réussi à sélectionner les meilleurs courbes en attribuant aux classifieurs associés à ces courbes les poids les plus forts dans le classifieur final.

5.2 Résultats sur la base test

Nous avons présenté au classifieur final fourni par Adaboost des exemples de test. Pour chaque exemple le classifieur produit une décision binaire (accès imposteur/accès client). Les résultats du tableau 2 montrent les performances de l'algorithme proposé et montrent clairement l'amélioration des résultats par rapport à la moyenne arithmétique d_{MA}. Par exemple, pour des échantillons de taille $m = l = 5000$ clients et imposteurs, la moyenne arithmétique donne un *VR* moyen de 68.18% alors que Adaboost

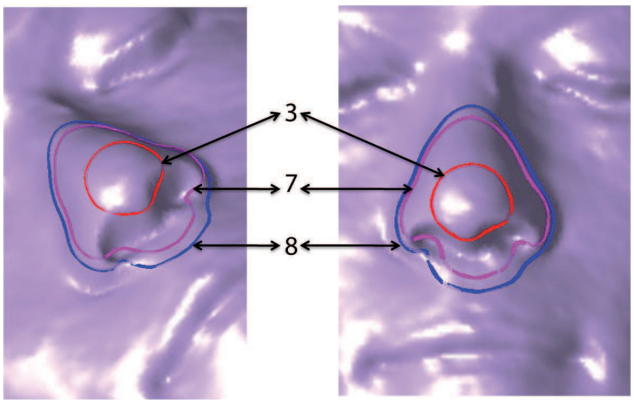

FIGURE 7 – L'emplacement des courbes 8, 7 et 3 sélectionnées par Adaboost

donne un *VR* de 81.5%.

6 Conclusion et perspectives

Dans ce papier nous avons présenté un nouveau classifieur basé sur les courbes nasales pour l'authentification faciale. D'abord, nous avons formulé le problème d'authentification comme un problème de classification à décision binaire : accès imposteur/accès client. Ensuite, nous avons proposé d'utiliser l'algorithme Adaboost pour optimiser les performance du classifieur. En se basant sur un ensemble d'apprentissage, Adaboost choisi les courbes les plus pertinentes sur le nez en attribuant les poids les plus important à leurs classifieurs faibles associés. Enfin, nous avons présenté des experimentations sur des ensembles d'échantillons de tailles variables. Ces expérimentations montrent l'intêret d'introduire le boosting pour améliorer les résultats d'authentification par rapport à une moyenne arithmétique.
Nous allons poursuivre nos travaux afin de trouver l'ensemble des courbes pertinentes sur le visage. Nous travaillerons aussi sur l'extension de l'algorithme Adaboost version binaire à une version multi-classes pour pouvoir traiter le scénario d'identification.

Remerciements

Ce travail a reçu le soutien du projet ANR-07-SESU-004 et du Contrat Plan Etat- Région (CPER) Région Nord-Pas de Calais Intelligence Ambiante. Nous remercions Hassen Drira pour l'aide qu'il nous a apportée et toutes les discussions que nous avons eues.

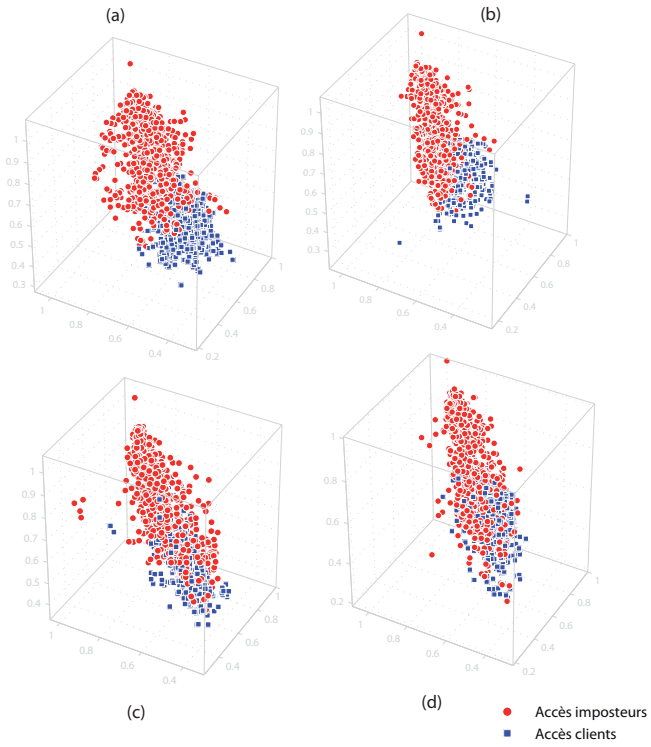

FIGURE 8 – Différentes distributions d'accès clients et d'accès imposteurs pour différentes combinaisons de courbes : (a) courbes 3, 7 et 8 - (b) courbes 1,6 et 9 - (c) courbes 2, 4 et 5 - (d) 3, 4 et 10.

Références

[1] T.C. Faltemier, K.W. Bowyer, and P.J. Flynn. A region ensemble for 3-d face recognition. *IEEE Transactions on Information Forensics and Security*, 3(1) :62–73, 2008.

[2] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *IEEE Transactions on Image Processing*, 16(1) :188–197, 2007.

[3] I. Mpiperis, S. Malassiotis, and M. G. Strintzis. 3-D face recognition with the geodesic polar representation. *IEEE Transactions on Information Forensics and Security*, 2(3) :537 – 547, 2007.

[4] F. B. ter Haar and R. C. Veltkamp. A 3d face matching framework for facial curves. volume 71, pages 77–91, 2009.

[5] P. Yan and Kevin W. Bowyer. Biometric recognition using 3d ear shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8) :1297–1308, 2007.

[6] H. Chen and B. Bhanu. Human ear recognition in 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4) :718–737, 2007.

[7] K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10) :1695–1700, 2006.

[8] H. Drira, B. Ben Amor, A. Srivastava, and M. Daoudi. A riemannian analysis of 3d nose shapes for partial

Adaboost (Ad) et Moyenne arithmétique (d_{MA})								
m, l	*VR (%)*		*FAR (%)*		*FRR (%)*		*TRR (%)*	
	Ad	d_{MA}	Ad	d_{MA}	Ad	d_{MA}	Ad	d_{MA}
2000	81.80	67.35	4.50		18.20	32.65	95.50	
2000	80.15	67.50	6.55		19.85	32.50	93.45	
2000	73.60	61.10	1.00		26.40	38.90	99.00	
2000	79.80	61.40	1.90		20.20	38.60	98.10	
2000	81.70	64.50	3.35		18.30	35.50	96.65	
Moy	**79.41**	**64.37**	3.46		**20.59**	**35.63**	96.54	
3000	77.50	67.03	3.07		22.50	32.97	96.93	
3000	74.83	68.07	3.13		25.17	31.93	96.87	
3000	83.43	72.20	6.03		16.57	27.80	93.97	
3000	80.80	71.33	6.13		19.20	28.67	93.87	
3000	82.77	70.33	4.70		17.23	29.67	95.30	
Moy	**79.87**	**69.79**	4.61		**20.13**	**30.21**	95.39	
4000	81.90	69.68	3.75		18.10	30.33	96.25	
4000	80.15	67.75	3.68		19.85	32.25	96.33	
4000	79.73	66.80	3.08		20.28	33.20	96.93	
4000	85.78	72.88	5.43		14.23	27.13	94.58	
4000	76.68	67.35	3.45		23.33	32.65	96.55	
Moy	**80.85**	**68.89**	3.88		**19.16**	**31.11**	96.13	
5000	83.90	69.76	4.48		16.10	30.24	95.52	
5000	81.94	69.32	3.46		18.06	30.68	96.54	
5000	81.18	69.34	3.06		18.82	30.66	96.94	
5000	81.22	66.86	2.28		18.78	33.14	97.72	
5000	79.24	65.60	1.90		20.76	34.40	98.10	
Moy	**81.50**	**68.18**	3.04		**18.50**	**31.82**	96.96	

TABLE 2 – Résultats de la vérification sur l'ensemble de test Ad : résultats donnés en utilisant Adaboost, MA : résultats donnés par la moyenne arithmétique

human biometrics. In *IEEE International Conference on Computer Vision*, pages 2050–2057, 2009.

[9] P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *Journal of the European Mathematical Society*, 8 :1–48, 2006.

[10] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machine Intelligence*, 26(3) :372–383, 2004.

[11] S. H. Joshi, E. Klassen, A. Srivastava, and I. H. Jermyn. A novel representation for efficient computation of geodesics between n-dimensional curves. In *IEEE CVPR*, pages 1–7, 2007.

[12] L. Younes. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58 :565–586, 1998.

[13] E. Klassen and A. Srivastava. Geodesics between 3D closed curves using path-straightening. In *Proceedings of ECCV, Lecture Notes in Computer Science*, pages I : 95–106, 2006.

[14] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95 : Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[15] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2) :137–154, 2004.

[16] P. J. Phillips, P. J. Flynn, W. T. Scruggs, K. W. Bowyer, J. Chang, K.Hoffman, J. Marques, J. Min, and W. J. Worek. Overview of the face recognition grand challenge. In *IEEE Computer Vision and Pattern Recognition*, pages 947–954, 2005.