

# TIPE : Reconnaissance de visages par Analyse en Composantes Principales (ACP)

2020 - 2021

## 1 Histoire et enjeux de la reconnaissance faciale

### 1.1 Différentes méthodes d'authentification

Il existe différentes méthodes de reconnaissance, d'authentification ou d'identification. La première est la méthode dite de Knowledge Based Authentication et nécessite un mot de passe ou un code PIN pour s'authentifier. La deuxième est dite Token Based Authentication et demande à l'utilisateur la possession d'un objet (comme une clé) ou utilisation d'un jeton unique (comme code unique envoyé par SMS par la banque lors d'une transaction internet). La troisième méthode, qui est le sujet de ce TIPE, est la catégorie des méthodes d'authentification biométriques qui sont plus difficiles à falsifier ou à perdre et "uniques et permanentes" car les informations biométriques d'un individu sont invariables dans le temps et sont propres à chacun [4].

### 1.2 Biométrie et intérêts

La biométrie se divise en 3 parties :

- la biométrie à grande fiabilité (iris) ;
- la biométrie à traces (empreintes digitales, la plus répandue des méthodes) ;
- la biométrie non coopérative (visage).

La biométrie à trace est très répandue mais nécessite un contact physique, ce qui par ailleurs n'est pas idéal en temps de pandémie à cause des problèmes d'hygiène que cela provoque (déverrouillage par empreintes digitales ou code PIN à composer).

La biométrie de reconnaissance de l'iris ou du visage se fait sans contact. La reconnaissance faciale comporte de nombreuses applications comme le déverrouillage des téléphones, des ordinateurs mais aussi l'accès à un site sécurisé, les transactions bancaires ou encore le passage aux frontières.

### 1.3 Histoire de la reconnaissance faciale

L'intérêt porté à la reconnaissance faciale remonte aux années 60 avec l'américain Woody Blesdoo et son équipe [2], qui doivent rentrer manuellement une vingtaine de mesures dans un ordinateur qui classe alors les visages. Puis avec le japonais Takeo Kanade [8], l'un des pères de la reconnaissance faciale, qui effectue sa thèse sur la reconnaissance faciale en 1974 en automatisant le processus semi-manuel de Woody Blesdoo. Ensuite, l'évolution des technologies et des recherches a mené à l'idée d'une méthode utilisant l'algèbre à des fins algorithmiques, l'Analyse en Composantes Principales (ACP). Cette idée a été formulée initialement par Sirovich et Kirby en 1987 puis utilisée pour la classification des visages par deux élèves du Massachusetts Institute of Technology, Matthew Turk et Alex Pentland, en 1991 [9]. Plus récemment, avec les progrès du deep-learning et de la technologie en général, des chercheurs de Facebook annoncent en 2014 pouvoir identifier un visage avec une certitude de 97% grâce à leur application DeepFace. Enfin, depuis 2008 en Chine (à Pékin avec les Jeux Olympiques) ou en 2019 en France (à Nice avec le carnaval), des logiciels de reconnaissance faciale équipent le système de vidéo-surveillance.

## 1.4 Objectifs du TIPE

Ce TIPE a pour objectif l'implémentation d'une méthode simple et efficace de reconnaissance faciale en utilisant la technique d'Analyse en Composantes Principales. La programmation de l'algorithme d'ACP se fera en Python. Grâce à une base de données de visages, on construit un espace engendré par les visages propres, qui sont les vecteurs propres des matrices de visages. L'identification d'un nouveau visage se fera en projetant ce visage sur un hyper-espace de visages propres bien choisi, de sorte à maximiser la variabilité. La distance du visage à l'espace permettra de définir un facteur de sécurité au-dessus duquel un nouveau visage pourra être considéré comme reconnu et accepté.

On parlera ensuite de quelques améliorations possibles de l'ACP puis de ses limites auxquelles on pourra trouver quelques solutions, dont une consiste en un algorithme d'Analyse Discriminante Linéaire de Fisher.

## 1.5 Principe de l'ACP

L'ACP permet de déterminer les « directions principales » dans un jeu d'observation, ici des visages, c'est-à-dire les directions dans lesquelles la variabilité des données est la plus grande. Ces directions sont déterminées par les vecteurs propres de l'opérateur linéaire associé à la matrice de variance-covariance des observations. Pour un nuage de données, il est difficile de déceler un lien entre elle. L'analyse en composantes principales est la méthode cherchant à expliquer la variabilité des données au moyen d'un petit nombre de combinaisons linéaires des  $n$  composantes originales. Alors, la découverte de tels liens permettra une réduction de la dimension des observations. Souvent,  $k \ll n$  combinaisons linéaires expliqueront la plus grande partie de la variabilité. Il sera alors possible, au prix d'une perte d'information minimale, de réduire le jeu de données en ne considérant que les  $k$  premières composantes du jeu de données dans la base « principale » [13].

# 2 Construction de l'algorithme

## 2.1 Pré-requis

On notera en *bleu* les noms des fonctions associées aux étapes de l'algorithme.

On parle de  $n$  qui est le nombre de visages, de l'ordre de 50, et de  $N$  qui est le nombre de pixels par image, de 225000 pour nos images.

On se servira dans ce TIPE de différents programmes dont on ne cherchera pas à creuser le fonctionnement. Notamment, on se servira d'un module Python nommé *haarcascade* pour détecter les visages capturés par la webcam. De bons réglages permettront de capturer seulement le bon visage (*detecteVisage* pour les captures d'image et *visageRealTime* pour la détection en direct). Le module *openCV* servira à capturer l'image de la webcam.

En revanche, on programmera quelques fonctions pour recadrer et redimensionner les visages (*processImage*) ainsi que pour transformer les images en matrices puis en vecteurs (*transformeImage* et *matriceToVect*). On supposera aussi que les visages sont tous pris avec une bonne luminosité, une bonne définition et une bonne prise de vue (de face, droit). Pour corriger ce problème, on pourrait faire plusieurs prises de vue avec plusieurs luminosités. Pour ces conditions, on verra en partie 3 quelques solutions.

## 2.2 Étapes de construction de l'ACP

On se donne une liste de visages  $(V_1, \dots, V_n) \in (\mathbb{R}_N)^n$  (vecteurs colonnes de taille  $N$ ). Ceci suppose qu'en se donnant  $n$  matrices de taille  $m \times p$  on peut les transformer en  $n$  vecteurs colonne de taille  $N = mp$ . On calcule le visage moyen (*moyenne*) :

$$\bar{V} = \frac{1}{n} \sum_{i=1}^n V_i$$

Puis, on pose :

$$\forall i \in \llbracket 1, n \rrbracket, W_i = V_i - \bar{V}$$

Ce qui permet d'obtenir :

$$A = (W_1, \dots, W_n)$$

Toutes ces dernières étapes sont effectuées par la fonction [construitA](#).

On rappelle que la matrice de la covariance  $C$  du visage  $V_i$  est  $C = \sum_{i=1}^n (V_i - \bar{V})^t (V_i - \bar{V})$ . Donc la matrice des covariances est proportionnelle à  $A^t A$ .

Il faut maintenant trouver les valeurs propres de  $A^t A \in \mathcal{M}_N(\mathbb{R})$ . Mais comme  $N \gg n$ , il est plus judicieux (au niveau du temps de calcul surtout) de chercher les éléments propres de  ${}^t A A \in \mathcal{M}_n(\mathbb{R})$ . En effet, si  $X_i \in \mathbb{R}^n$  est un vecteur propre de  ${}^t A A$  associé à la valeur propre  $\alpha$  (i.e.  ${}^t A A X_i = \alpha X_i$ ), alors  $A X_i = e_i$  est un vecteur propre de  $A^t A$  associé à la même valeur propre  $\alpha$  (i.e.  $A^t A (A X_i) = \alpha (A X_i)$ ).

**Démonstration :** On pose  $L = {}^t A A$  et  $Sp(L) = \lambda_1, \dots, \lambda_n$ . Alors,  $\forall \lambda_i \in Sp(L), \exists X_i \in \mathbb{R}^n \setminus \{0\} / L X_i = \lambda_i X_i \Rightarrow {}^t A A X_i = \lambda_i X_i \Rightarrow A^t A (A X_i) = \lambda_i (A X_i) \Rightarrow \forall \lambda_i \in Sp(L), \exists e_i \in \mathbb{R}_N \setminus \{0\} / A^t A e_i = \lambda_i e_i$  donc  $\lambda_i \in Sp({}^t A A) \Rightarrow \lambda_i \in Sp(A^t A)$ .

On a donc que  $card(Sp(A^t A)) = card(Sp({}^t A A)) = n$ . Ainsi, pour chaque vecteur propre  $X_i$  de  ${}^t A A$  on tire un vecteur propre  $e_i = A X_i$ . On trouve  $n$  valeurs propres au système. D'où notre liste  $(e_1, \dots, e_n)$  de vecteurs propres (ce sont les **visages propres**) de  $A^t A$  ([construitEi](#)).

Notons que l'on peut dès maintenant voir les "fantômes" c'est-à-dire les images engendrées par les vecteurs propres. On peut afficher  $n$  fantômes.

$(e_1, \dots, e_n)$  engendre un espace de dimension  $n$  appelé l'**espace des visages**.

## 2.3 Reconnaissance d'un nouveau visage

Tout nouveau visage  $\tilde{V} \in \mathbb{R}_N$  peut être projeté sur cet espace. Posons  $\tilde{V} \in \mathbb{R}_N$  le nouveau visage. On s'intéresse à,

$$\forall i \in \llbracket 1; n \rrbracket, \omega_i = {}^t e_i (\tilde{V} - \bar{V}) \in \mathbb{R}$$

D'où la liste  $(\omega_1(V), \dots, \omega_n(V)) \in \mathbb{R}^n$  [projectionVisage](#).

En pratique, on ne garde qu'un nombre  $k < n$  de visages propres (cf 3.2).

Ainsi, tout nouveau visage peut être décrit, sans trop de perte, par  $\Omega_k(V) = (\omega_1(V), \dots, \omega_k(V)) \in \mathbb{R}^k$  où  $k < n$  ([construitOmega](#)).

Enfin, pour tout nouveau visage  $\tilde{V}$ , on calcule sa distance à l'image  $i$  avec la formule :

$$\forall i \in \llbracket 1; n \rrbracket, d_i = \left\| \Omega(\tilde{V}) - \Omega(V_i) \right\|$$

On obtient une liste de distances  $(d_1, \dots, d_n) \in \mathbb{R}^n$  ([distance](#)). On associe le nouveau visage  $\tilde{V}$  à l'individu  $j$  tel que  $j = \min_{i \in \llbracket 1; n \rrbracket} d_i^2$ .

Il ne reste plus qu'à chercher le visage associé à la distance la plus petite de cette liste, c'est-à-dire le visage  $j$  dans notre cas, ce qui est effectué par la fonction [trouveVisageCorrespondant](#).

## 2.4 Facteur de reconnaissance

La liste des distances donnée précédemment ne peut rien signifier sans un peu d'expérience. On définit donc un facteur de reconnaissance  $\eta = 100 * \left( 1 - \frac{\min_{i \in \llbracket 1; n \rrbracket} d_i}{\max_{i \in \llbracket 1; n \rrbracket} d_i} \right)$  qui est un nombre entre 0 et 100, de manière à obtenir un pourcentage. A 100% le visage en entrée appartient à la base de données. Plus le pourcentage est bas, moins le visage ressemble à un visage de la base de données. Expérimentalement, on peut définir à 75% le seuil de reconnaissance d'un visage. Au-dessus de ce seuil, le visage est reconnu et accepté. En dessous, le visage est refusé. Mais de nombreux facteurs peuvent causer des faux positifs comme nous allons le voir en partie 4.

### 3 Discussion et améliorations possibles

#### 3.1 Importance de la taille de la base de données

Il est important de noter que plus le nombre de visages dans la base de données est grand, plus l'ACP sera précise. En effet, avec plus de visages la variabilité est plus grande et donc on peut choisir une direction de sous-espace plus précise.

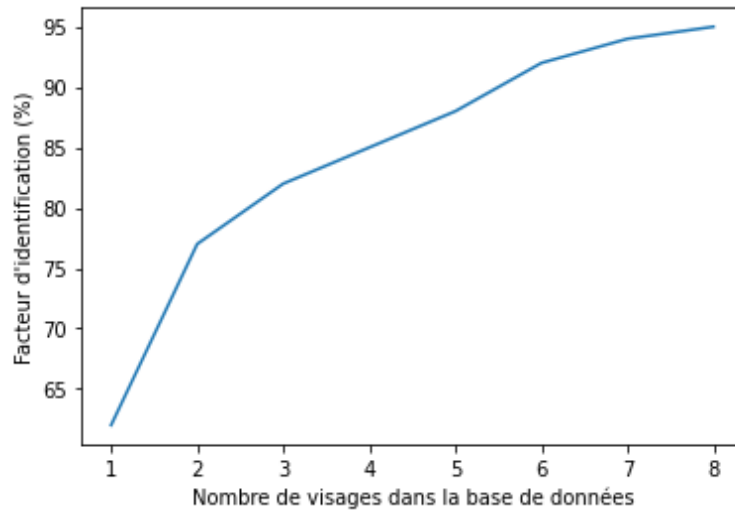


FIGURE 1 – Intérêt d'un grand nombre de visages

Dans une moindre mesure, c'est ce qui est effectué par les forces de l'ordre après arrestation d'un individu. Celui-ci est photographié sous plusieurs angles pour augmenter le nombre de prises de vues différentes et donc permettre une identification plus aisée pour un besoin futur.

#### 3.2 Choix de la dimension du sous-espace

Plusieurs critères sont possibles pour choisir au mieux la dimension du sous-espace des visages propres. Dans tous les cas, on veut minimiser la dimension de cet espace en maximisant l'information connue.

Le critère dit du "coude" ou de "l'effondrement des valeurs propres" part de l'observation d'un effondrement des valeurs propres. L'abscisse du "coude" donne la dimension de l'espace (environ 6 sur la figure) sur lequel projeter (c'est  $k$  dans le programme) [6]. Ceci nous donne  $k$  afin d'avoir le plus d'information en ayant le moins d'image. Au passage, cela réduit le temps de calcul.

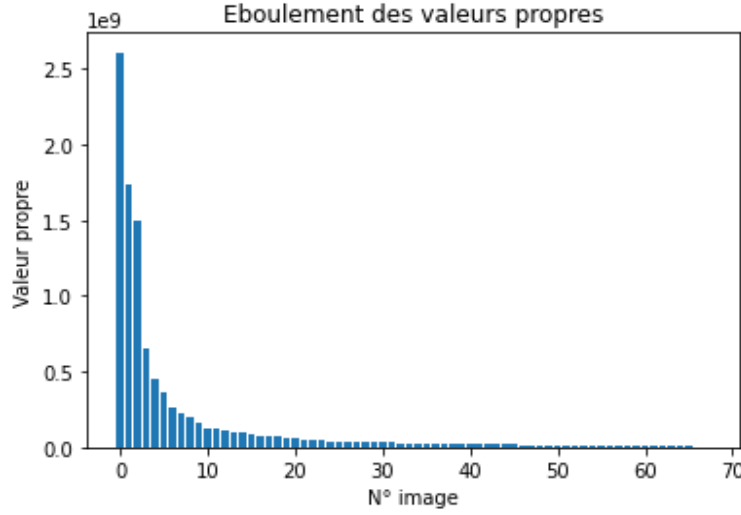


FIGURE 2 – Éboulement des valeurs propres

Le critère de Kaiser dit que seuls les vecteurs propres associés à une valeur propre supérieure à 1 sont conservés, ce qui est valable pour une ACP normée c'est-à-dire où la matrice des corrélations est symétrique avec une diagonale de 1 (mais ce n'est pas le cas dans notre programme) [3].

### 3.3 Choix de la norme pour la distance

Après expérimentations, on aboutit à la norme 1 pour l'ACP ( $\|X\|_1 = \sum_{i=1}^n |x_i|$ ) et la distance cosinus pour la LDA

$$(\cos(\theta) = \frac{\langle X, Y \rangle}{\|X\| \|Y\|}).$$

### 3.4 Méthode (PC)<sup>2</sup>A

On a vu en partie 3.1 que l'on ne peut généralement pas s'attendre à obtenir de bonnes performances avec la technique standard de l'ACP si on ne dispose que d'un seul échantillon par personne. Il faut donc améliorer la robustesse du programme. Une solution proposée par Wu Zhou en 2006 [12] est la méthode de projection du champ de la détection des visages. Il définit deux projections verticales et horizontales de l'intensité lumineuse du visage pour générer une carte de projection qu'il combine ensuite avec le visage. Ces deux projections reflètent la distribution des traits saillants du visage qui sont utiles pour la reconnaissance des visages. Par conséquent, les caractéristiques non importantes pour la reconnaissance des visages sont effacées et les caractéristiques importantes deviennent plus saillantes après le pré-traitement. Après cela, une ACP est effectuée sur ces visages traités. Leur méthode a été testée sur un sous-ensemble de la base de données FERET [11] avec une image d'entraînement pour chaque personne et ils ont rapporté qu'ils avaient atteint une précision de 3 à 5 % supérieure à l'ACP "traditionnelle" en utilisant 10 à 15 % de visages en moins [7].

Les étapes imposées aux images sont les suivantes :

Soit  $I(x, y)$  l'intensité du pixel  $(x, y)$  de l'image  $(a)$  de dimension  $m \times p$  représentée par  $M$ . Les projections horizontale  $HI(x)$  et verticale  $VI(y)$  de cette image sont respectivement définies par :

$$HI(x) = \sum_{y=1}^p I(x, y)$$

$$VI(y) = \sum_{x=1}^m I(x, y)$$

La matrice de projection  $M_p$  a donc ses coefficients égaux à  $M_p(x, y) = \frac{HI(x) VI(y)}{J}$  où  $J$  est l'intensité moyenne de l'image et  $(x, y) \in \llbracket 0, m \rrbracket \times \llbracket 0, p \rrbracket$ . Ceci nous donne (b).

Enfin, l'image (c) de matrice  $M' = (M'(x, y))_{(x, y) \in \llbracket 0, m \rrbracket \times \llbracket 0, p \rrbracket}$  est donnée par la formule :

$$\forall (x, y) \in \llbracket 0, m \rrbracket \times \llbracket 0, p \rrbracket, M'(x, y) = M(x, y)M_p(x, y)$$

Ces étapes sont effectuées par [uniformiseLumiere](#)

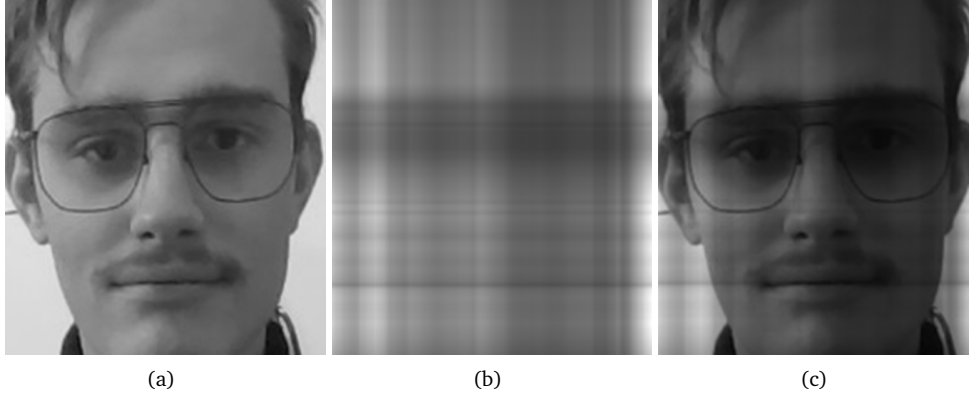


FIGURE 3 – (a) Visage original (b) Carte de projection (c) Images combinées

## 4 Limites de l'ACP

La technique de l'Analyse en Composantes Principales peut-être simplement mise en échec par des configurations de visages. En effet, l'ACP se révèle particulièrement inefficace face aux variations de pose des sujets, d'illumination de l'image ou encore d'expression des individus. En effet, en choisissant la projection qui maximise la dispersion totale, l'ACP conserve les variations indésirables dues à l'éclairage et à l'expression faciale. Ainsi, alors que les projections de l'ACP sont optimales pour la reconstruction à partir d'une base de faible dimension, elles peuvent ne pas être optimales du point de vue de la discrimination.

Comme le disait déjà Woody Bledsoe en 1966 : "Ce problème de reconnaissance est rendu difficile par **la grande variabilité de la rotation et de l'inclinaison de la tête, de l'intensité et de l'angle d'éclairage, de l'expression faciale, du vieillissement, etc.** D'autres tentatives de reconnaissance faciale par machine n'ont permis que peu ou pas de variabilité dans ces quantités. Pourtant, la méthode de corrélation (ou de comparaison de motifs) de données optiques non traitées, souvent utilisée par certains chercheurs, est vouée à l'échec dans les cas où la variabilité est grande. En particulier, **la corrélation est très faible entre deux images de la même personne avec deux rotations de tête différentes.**" [2]. En effet, la reconnaissance faciale est non coopérative. Ces limites font l'objet de nombreux travaux depuis les années 2000. D'où l'importance de la normalisation des visages, avec une inclinaison nulle, un éclairage constant et une expression neutre [3].

## 5 Solutions possibles

On a vu que l'ACP peut être simplement mise en échec par des configurations de visages ou d'éclairage. Il faut donc demander à la personne de prendre la photo de face, avec une bonne luminosité et expression neutre, ce qui n'est pas toujours possible, par exemple dans le cas de la vidéo surveillance. C'est en revanche souhaitable pour l'identification faciale d'un individu aux frontières grâce à son passeport qui contient une photo prise de face, avec un bon éclairage, une expression neutre, sans accessoire susceptible de brouiller la reconnaissance.

### 5.1 Localiser l'Analyse en Composantes Principales

Une première solution est de prendre quelques échantillons du visage et les comparer avec les autres échantillons des visages de la base : on localise l'ACP. Il convient de découper le visage en imagerie sur lesquelles on effectue l'ACP.

Pour les limites temporelles c'est-à-dire pour une rapidité convenable du programme (moins d'une seconde), quand on a beaucoup de visages et/ou de grande définition, on peut effectuer l'ACP sur des régions du visage (bouche, nez, yeux), ce qui donne des résultats aussi satisfaisants qu'avec la méthode traditionnelle [7]. Aussi, en ces temps de pandémie, avec les masques, on peut adapter la méthode de reconnaissance, par exemple aux yeux.

## 5.2 Augmenter le nombre de clichés par personne

La prise de plusieurs clichés du même visage augmente la reconnaissance des visages. On passe de 65% pour un visage à 95% pour 9 visages [7]. L'ACP est d'autant plus efficace que le nombre de visages de la base de données est grand.

## 5.3 L'Analyse Discriminante Linéaire

Une autre solution est l'Analyse Discriminante Linéaire. Nous en parlerons juste après, en partie 6.

# 6 L'Analyse Discriminante Linéaire (ADL)

## 6.1 Sur l'intérêt de l'Analyse Discriminante Linéaire

Pour résoudre les problèmes de l'ACP que sont la luminosité, les expressions et l'inclinaison, on dispose d'une autre méthode nommée **Analyse Discriminante Linéaire** (ADL), méthode proposée par Peter N. Belhumeur, Joao P. Hespanha, et David J. Kriegman en 1997 [1], s'appuyant eux-même sur l'analyse discriminante de Fisher datant de 1938. La méthode discriminante linéaire de Ronald Fisher essaie de "façonner" la dispersion afin de réduire la taille de l'échantillon.

Contrairement à l'ACP, l'algorithme d'ADL effectue une véritable séparation de classes. Ceci sous-entend qu'il faut effectuer un traitement sur la base de données. Donc, pour pouvoir l'utiliser, il faut avoir au préalable au moins deux visages par personne et les avoir organisés dans un dossier de quelques images par personne. L'ADL analyse les vecteurs propres de la matrice de dispersion des données, avec pour objectif de maximiser les variations inter-classes tout en minimisant les variations intra-classes [10].

Avec l'Analyse Discriminante Linéaire, une marge de performance supérieure de 10 % sur la méthode des visages propres traditionnelle a été obtenue [7].

## 6.2 Programmation de l'ADL

On utilise directement la méthode donnée [1].

On se donne une liste de visages  $(V_1, \dots, V_n) \in (\mathbb{R}_N)^n$  (vecteurs colonnes de taille  $N$ ). Ceci suppose qu'en se donnant  $n$  matrices de taille  $m \times p$  on peut les transformer en  $n$  vecteurs colonne de taille  $N = mp$ . On se donne aussi  $(C_1, \dots, C_c)$  des classes, c'est-à-dire un nombre  $c$  de personnes différentes. Pour chaque personne, donc pour chaque classe, on a  $q_i \geq 2$  prises de vues différentes.

De la même manière que pour l'ACP, on calcule le visage moyen de tous les visages (*moyenneADL*) :

$$\bar{V} = \frac{1}{n} \sum_{i=1}^n V_i \in \mathbb{R}_N$$

Pour chaque classe  $C_i$ , on calcule le visage moyen de cette classe (*calculePsi*) :

$$\bar{V}_{C_i} = \frac{1}{q_i} \sum_{i=1}^{q_i} V_i \in \mathbb{R}_N$$

Chaque image  $V_i$  de chaque classe  $C_i$  est recentrée par rapport à la moyenne  $\bar{V}_{C_i}$  (*calculePhi*) :

$$W_i = V_i - \bar{V}_{C_i} \in \mathbb{R}_N$$

Il faut maintenant calculer les 3 matrices de dispersions différentes :

- la matrice de **dispersion intra-classe** (*calculeDispersionIntraClasse*) :

$$S_w = \sum_{i=1}^c \sum_{V_k \in C_i} (V_k - \bar{V}_{C_i})^t (V_k - \bar{V}_{C_i}) \in \mathcal{M}_N(\mathbb{R})$$

- la matrice de **dispersion inter-classe** (*calculeDispersionInterClasse*) :

$$S_b = \sum_{i=1}^c q_i (\bar{V}_{C_i} - \bar{V})^t (\bar{V}_{C_i} - \bar{V}) \in \mathcal{M}_N(\mathbb{R})$$

- la matrice de **dispersion totale** (*calculeDispersion*) :

$$S_T = \sum_{i=1}^n (V_i - \bar{V})^t (V_i - \bar{V}) \in \mathcal{M}_N(\mathbb{R})$$

Note : comme on obtient des matrices  $N \times N$ , il faut veiller à avoir  $N$  petit, pour la rapidité de l'algorithme et la taille en mémoire.

Ensuite, il faut trouver une projection optimale  $W$  pour maximiser la dispersion intra-classe, relative à  $S_w$ , tout en minimisant la dispersion inter-classe, relative à  $S_b$ . Il faut trouver  $W$  qui maximise le *critère d'optimisation de Fisher*  $J(T)$  :

$$W = \max_T \left( \frac{{}^t T S_b T}{{}^t T S_w T} \right) \in \mathcal{M}_N(\mathbb{R})$$

$W$  peut être trouvée en résolvant le problème généralisé aux valeurs propres [5] :

$$S_b W = \lambda_w S_w W$$

On se ramène alors au même schéma que l'Analyse en Composantes Principales. La projection du visage  $V_i$  par rapport à la moyenne est définie par :

$$p(W_i) = {}^t W W_i$$

Ainsi, tout nouveau visage  $\tilde{V} \in \mathbb{R}_N$  peut être projeté :

$$p(\tilde{V}) = {}^t W \tilde{V}$$

Toujours de la même manière que pour l'ACP, on mesure la distance entre le nouveau visage  $\tilde{V}$  et sa projection dans l'espace engendré par  ${}^t W$ . On calcule sa distance à l'image  $i$  avec la formule :

$$\forall i \in \llbracket 1; n \rrbracket, d_i = \left\| p(\tilde{V}) - p(V_i) \right\|$$

Finalement, une image test est dans la classe dont la distance est minimale par rapport à toutes les autres distances de classe, c'est-à-dire la classe  $i$  telle que  $i = \min_{i \in \llbracket 1; c \rrbracket} d_i$ .

## 6.3 Conclusion

L'ACP ou l'ADL forment donc une bonne introduction aux méthodes de reconnaissance faciale. Cependant, leur inconvénient majeur réside dans le fait qu'elles utilisent uniquement des photos 2D d'apparence faciale. Or, on sait qu'une telle représentation est forcément sensible aux changements d'expression, d'illumination et de poses [10]. Pour aller plus loin, ce qui est le cas des téléphones par exemple mais pas encore des caméras de surveillance, il faudrait pouvoir capturer la profondeur du visage, avec des capteurs adaptés, ce qui résoudrait la problématique d'éclairage.

## Références

- [1] P.N. BELHUMEUR, J.P. HESPANHA et D.J. KRIEGMAN. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (1997), p. 711-720. DOI : 10.1109/34.598228. URL : <https://cseweb.ucsd.edu/classes/wi14/cse152-a/fisherface-pami97.pdf>.
- [2] W. BLEDSOE. "Man-machine facial recognition". In : *Panoramic Research Inc. California* (1966).
- [3] B. CHU. "Neutralisation des expressions faciales pour améliorer la reconnaissance du visage". Thèse de doct. 2015, p. 17. URL : [https://bibli.ec-lyon.fr/exl-doc/TH\\_T2445\\_bchu.pdf](https://bibli.ec-lyon.fr/exl-doc/TH_T2445_bchu.pdf).



- [4] CNIL. "Reconnaissance faciale". In : (). URL : <https://www.cnil.fr/fr/definition/reconnaissance-faciale>.
- [5] Benjamin GHOJOGH, Fakhri KARRAY et Mark CROWLEY. "Eigenvalue and Generalized Eigenvalue Problems: Tutorial". In : (2019). URL : <https://arxiv.org/pdf/1903.11240.pdf>.
- [6] P-L. GONZALEZ. "L'ANALYSE EN COMPOSANTES PRINCIPALES (A.C.P.)" In : (). URL : <https://maths.cnam.fr/IMG/pdf/A-C-P-.pdf>.
- [7] Souhila GUERFI. "Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D". Thèse de doct. Université d'Evry-Val d'Essonne, 2008. URL : <https://tel.archives-ouvertes.fr/tel-00623243>.
- [8] Takeo KANADE. "Picture Processing System by Computer Complex and Recognition of Human Faces". In : (1974). URL : [https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/162079/2/D\\_Kanade\\_Takeo.pdf](https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/162079/2/D_Kanade_Takeo.pdf).
- [9] A. Pentland M. TURK. "Eigenfaces for recognition". In : *Journal of Cognitive Neuroscience* 3.1 (1991), p. 71-86.
- [10] N. MORIZET et al. "Revue des algorithmes PCA, LDA et EBGM utilisés en reconnaissance 2D du visage pour la biométrie". In : (). URL : <http://www.dreamlabs.eu/nmorizet/wp-content/uploads/2013/03/Revue-des-Algorithmes-PCA-LDA-et-EBGM-utilise%5C%CC%5C%81s-en-reconnaissance-2D-du-visage-pour-la-biome%5C%CC%5C%81trie-MajecSCTIC-2006.pdf>.
- [11] P.Jonathon PHILLIPS et al. "The FERET database and evaluation procedure for face-recognition algorithms". In : *Image and Vision Computing* 16.5 (1998), p. 295-306. ISSN : 0262-8856. DOI : [https://doi.org/10.1016/S0262-8856\(97\)00070-X](https://doi.org/10.1016/S0262-8856(97)00070-X). URL : <https://www.sciencedirect.com/science/article/pii/S026288569700070X>.
- [12] Xiaoyang TAN et al. "Face Recognition from a Single Image Per Person: A Survey". In : *Pattern Recognition* 39 (2006), p. 1730. DOI : 10.1016/j.patcog.2006.03.013. URL : [https://www.researchgate.net/publication/222672070\\_Face\\_Recognition\\_from\\_a\\_Single\\_Image\\_Per\\_Person\\_A\\_Survey](https://www.researchgate.net/publication/222672070_Face_Recognition_from_a_Single_Image_Per_Person_A_Survey).
- [13] G. VAN-BEVER. "De la reconnaissance faciale via l'analyse en composantes principales". In : (). URL : <https://bssm.ulb.ac.be/data/books/Bssm-2011-Germain-Van-Bever.pdf>.