

# Modélisation et reconnaissance des formes

## TP Reconnaissance de visage et “Eigenfaces”

Ce TP est à rendre impérativement pour le **24 novembre 2021** sur Arche. Le compte rendu doit contenir un fichier pdf ou word avec les réponses aux questions posées ainsi que des illustrations bien choisies de vos expériences. Vous joindrez également l'ensemble des codes demandés et les scripts permettant leur exécution.

### 1 Introduction

Ce TP concerne la méthode de représentation et de reconnaissance de visages par “Eigenfaces” [Turk & Pentland 91], qui est disponible sur arche. Vous allez l'implanter et l'étudier plus en détails.

Le principe de base est l'analyse en composantes principales qui permet de représenter une forme comme une combinaison linéaire d'une base de vecteurs bien choisie. Elle ne repose sur aucune identification particulière de traits du visages. Nous travaillerons sur des données de taille modeste disponibles sur arche et extraites de la base de yale. Plusieurs images par visage sont disponibles, prises dans des conditions légèrement différentes.

### 2 Préparation des données

Récupérer le zip sur arche qui contient la base de Yale et un utilitaire de lecture des images. La base contient des visages éclairés différemment, pris sous des angles légèrement différents. Les images correspondant à un visage  $i$  sont du type yaleBi\*. Les visages dit *de référence* sont ceux dont le suffixe est “P00A+000E+00.pgm”.

- Exécutez la fonction `lectureYale.m`. Cette fonction lit les 34 premières images de référence de la base (à noter que l'image 14 est absente de la base, attention donc aux indices des images si vous devez les utiliser). Les autres sont laissées pour des tests. Chaque image est stockée comme un vecteur colonne de taille  $\text{dimx} \times \text{dimy}$  (cf fonction `reshape` qui permet de passer d'une matrice 2d à une représentation en ligne et inversement). La matrice  $X$  des données ainsi constituée contient l'ensemble des visages stockés comme vecteurs colonnes.
- On note  $m$  le visage moyen correspondant à ces données. En utilisant la fonction `mean`, calculer le visage moyen et affichez le en utilisant la fonction `imshow`. Vous n'oublierez pas de faire un reshape pour passer d'un vecteur colonne  $V$  à un tableau 2d affichable par `imshow` : `reshape(V,dimx,dimy)`.

L'analyse en composantes principales vise à déterminer une base de vecteurs  $E_{i|1 \leq i \leq q}$  (ici appelés eigenfaces) permettant d'approximer avec une faible erreur un visage  $V$  comme une combinaison linéaire de ces eigenfaces :

$$V \approx m + \sum_{i=1}^q \alpha_i E_i$$

$V$  est ainsi représenté par les  $\alpha_i$  qui sont les coordonnées du visage dans la base. L'intérêt de cette représentation est sa compacité car on utilise en général une valeur de  $q$  très inférieure à la taille des images  $\text{dimx} \times \text{dimy}$ .

### 3 ACP

D'après la théorie vue en cours, les eigenfaces sont les vecteurs propres de la matrice  $XX^t$ , de taille  $n \times n$ , avec  $n = \text{dimx} \times \text{dimy}$ , où  $X$  représente l'ensemble des visages centrés. Matlab intègre

directement une fonction *pca* permettant de calculer la base  $E$  de la pca, les valeurs propres  $\lambda_i$  ainsi que les coordonnées de chaque visage projeté sur la base  $E_i$ .

- Regardez la documentation de la fonction *pca* et appliquez-la sur les données (attention, il faut la lancer sur  $X'$ ).
- La projection d'une image  $I$  de test dans le sous-espace des eigenfaces se fait en soustrayant le visage moyen et en effectuant le produit scalaire de l'image obtenue avec chaque eigenface. Ceci donne les coordonnées de l'image test dans le sous-espace des visages, qui est de dimension  $q$ . Les coordonnées de l'image projetée dans la base des eigenfaces sont donc

$$\alpha_k = \langle E_k, V - m \rangle = E_k^t \cdot (V - m)$$

Le visage projeté s'écrit donc

$$V_p = m + \sum_{i=1}^q \alpha_i E_i$$

Cette image est aussi appelée **image reconstruite**, car c'est l'image reconstruite à partir de la base des eigenfaces.

Ecrire une fonction *projectionACP* prenant en entrée une image de visage  $V$ , la matrice de base des eigenfaces, le vecteur moyenne  $m$ , ainsi que  $q$  et fournissant en sortie le vecteur des  $\alpha_i \in \mathbb{R}^q$  des coordonnées du visage sur la base des eigenfaces. Ecrire une fonction *reconstructionACP* permettant de reconstruire le visage à partir de la base, de la moyenne, de  $q$  et du vecteur  $\alpha$  (attention au fait que *pca* travaille sur des données en double. L'affichage doit se faire par *imshow(uint8(I\_r))*).

- Vérifier que votre fonction est correcte en utilisant l'image 1. Les coordonnées de la projection de cette image sur la base vous sont données par le deuxième argument de sortie de la fonction *pca*. Vous pouvez donc vérifier l'exactitude de votre fonction.
- Pour le visage 1 des données utilisées pour construire l'ACP ( $X(:,1)$ ), afficher les images reconstruites successivement avec la première eigenface, 2 eigenfaces, puis avec 5, 10, 15, 20 et 25 eigenfaces.
- Faites le même travail avec les images de référence 3- à 39 qui n'ont pas servi pour le calcul de la base. Que constatez vous ?
- Choix de la valeur de  $q$ . Tracez la somme cumulée normalisée des valeurs propres (c-a-d le graphique fournissant en  $i$  la somme  $\sum_{1 \leq k \leq i} \lambda_k / \sum \lambda_k$ ). Comment choisir une valeur appropriée pour  $q$  ?

## 4 Reconnaissance de visages

Dans cette partie, nous allons nous intéresser à identifier une personne. Dans notre base de données, chaque individu est représenté par un numéro  $i$  et par une image de référence. Etant donnée une image test, nous voulons retrouver l'identité de la personne sur l'image ou identifier la personne comme n'étant pas présente dans la base de référence.

La méthode utilisée consiste à travailler dans l'espace des eigenfaces et à comparer l'image  $J_p$ , projetée de  $J$  dans l'espace des eigenfaces avec la projection de chacune des images de référence. On note  $R^1, \dots, R^K$ , les images de référence et  $R_p^1, \dots, R_p^K$ , leurs projetés. La méthode proposée pour identifier un visage  $J$  est de calculer la distance entre la projection de cette image et la projection des images de référence :

$$d_{ident}(J, i) = \|J_p - R_p^i\|^2$$

La norme utilisée ici est la norme euclidienne  $\|X\|^2 = \sum_{i=1}^q x_i^2$ . Le visage de référence  $i$  fournissant la plus petite valeur de  $d_{ident}(J, i)$  sera alors considéré comme l'identité de la personne.

- Justifier la méthode d'identification proposée
- Écrire une fonction *identification* prenant en paramètres le visage  $J$ , la base,  $m$ , la matrice de score des visages de référence et  $q$  et fournissant en sortie le numéro du visage de référence le plus proche de  $J$  et la valeur de la distance associée.
- Vous disposez pour chaque individu de la base de données de plusieurs images prises dans des conditions différentes. Considérons d'abord les images suffixées par "P00A-010E+00.pgm" Calculer pour chaque image de cette série le visage de référence qui est le plus proche. Calculer le ratio de bonne identification, c'est à dire le nombre de fois où la personne identifiée est correcte. Le ratio varie-t-il en fonction de  $q$ ? Montrez un exemple d'échec de l'identification et analysez le résultat. Même question avec la série d'image suffixée par "P00A+010E-20.pgm". Montrez un ou des exemples d'échec et analysez ces résultats. Comment expliquez vous la différence de ratio entre les deux séries de test? Que pourrait-on faire pour améliorer les résultats?

## 5 Visage ou non visage ?

La projection sur l'espace des eigenfaces peut s'appliquer à n'importe quelle image, même à des non visages. Proposer une méthode permettant de tester si une image s'apparente à un visage ou n'en est pas un.

Vous développerez d'abord une méthode utilisant la base de l'ACP. Testez votre méthode et analysez les facteurs de difficultés.

Vous pouvez aussi suggérer d'autres méthodes.