
Les «Eigenfaces» : Méthode de la puissance



Auteurs :

Saïd AIT FASKA
Othman AIT OUAKRIM
Nouhaila ACHEHBOUNE

Encadrant :

M. Justin LOYE

ANNÉE UNIVERSITAIRE
2020-2021

Table des matières

1	Introduction	3
2	Les « Eigenfaces »	4
2.1	analyse en composantes principales	4
2.2	projection des images sur les eigenfaces	5
3	L'ACP et la méthode de la puissance itérée	7
3.1	Réponse aux questions	7
4	Conclusion	9

1 Introduction

Avec la pandémie de la Covid-19, nous avons appris à porter des masques quotidiennement. Désormais la moitié de notre visage est cachée nous laissant le regard pour seul moyen d'expression visible. Cette opération de cacher le visage s'apparente à un domaine en traitement d'images et de vidéos appelé "inpainting". En effet, les pixels correspondant à la zone du masque (modélisé par un rectangle) sont mis à 0. Et les eigenfaces permettent, entre autres, de restaurer la zone dégradée.

Eigenfaces

(Sirovich-Kirby, 1987; Turk-Pentland, 1991)



(25-dimensional representation)

FIGURE 1 – Eigenfaces 25-representation

2 Les « Eigenfaces »

Cette première partie consiste à partir d'une base des individus constitué de 16 hommes et 16 femmes, d'en tirer un individu qu'on qualifiera individu moyenne

La figure 2 suivante représente les 4 individus et leurs différentes postures

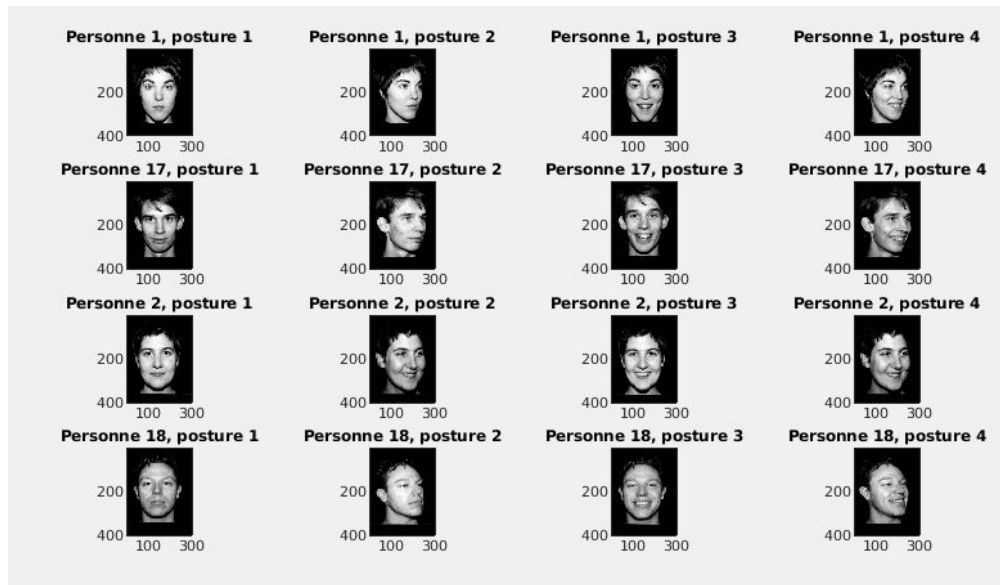


FIGURE 2 – Individus et postures

2.1 analyse en composantes principales

Le premier exercice consiste à compléter le script "eigenfaces.m" qui vise à calculer les axes principaux des images d'apprentissage à partir des vecteurs propres associés aux $n - 1$ valeurs propres non nulles de la matrice de variance/covariance Σ des données

En calculant l'individu moyen qui est tiré parmi les 4 individus initiaux, grâce au TP1 en analyse de données on obtient donc le résultat de la figure 3 suivante qui affiche l'individu moyenne et les différents eigenfaces associés.

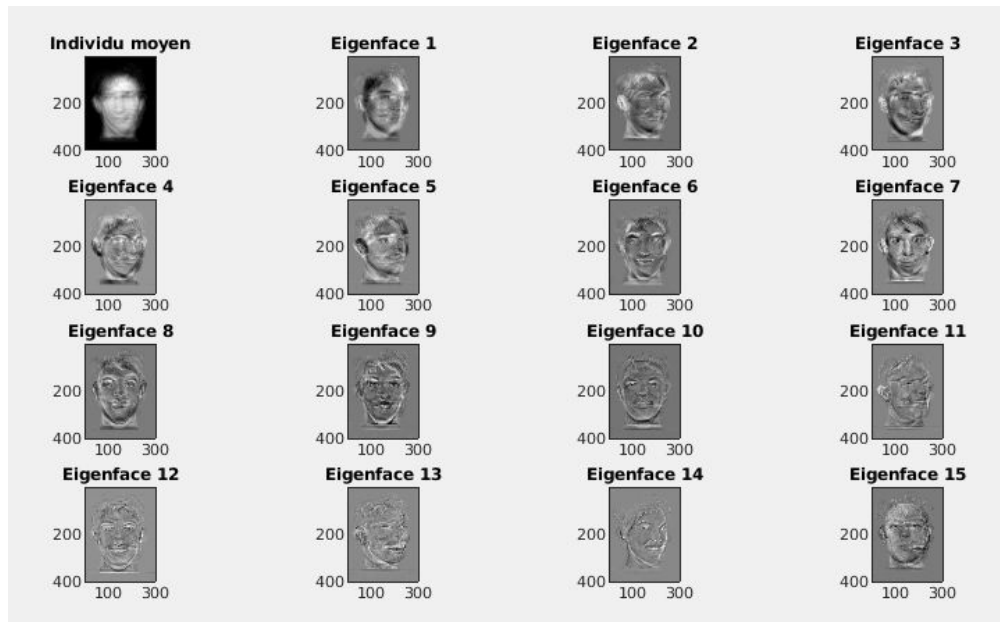


FIGURE 3 – Individus et eigenfaces associés

2.2 projection des images sur les eigenfaces

Cette sous-partie permet à partir des $n - 1$ eigenfaces obtenus grâce à l'analyse en composante principale, d'afficher les images d'apprentissages reconstruites à l'aide des q premières eigenfaces et des q premières composantes principales.

Dans la figure 3 suivante, on obtient les 4 individus précédents (c.f figure 1) ainsi que l'évolution, en fonction de q , de la racine carrée de l'erreur quadratique moyenne (Root Mean Square Error, ou RMSE) entre les images originales et les images ainsi reconstruites.

Dans cette première simulation on peut remarquer que dans l'état initial qui correspond aux images des individus "floues" l'erreur quadratique q est importante.

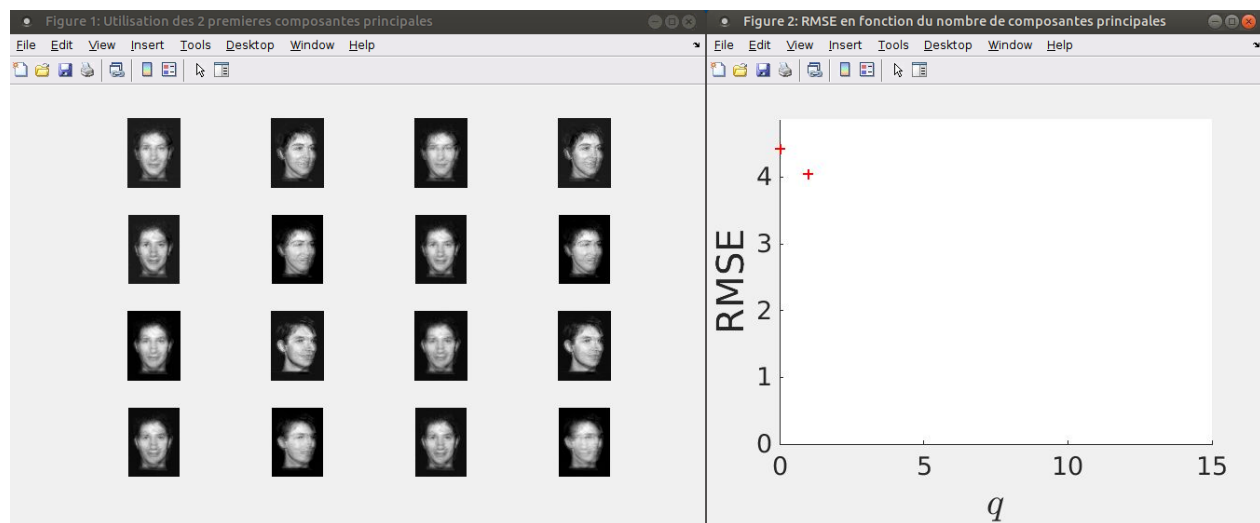


FIGURE 4 – <eigenfaces> et erreur quadratique.

A mesure que les images deviennent "nettes" *i.e* l'erreur quadratique q entre les images restituées et les images originales tend vers 0.

Ci-dessous la simulation à l'état final Figure 4 lorsque les images (originales et restituées) coïncident à un erreur *epsilon* près.

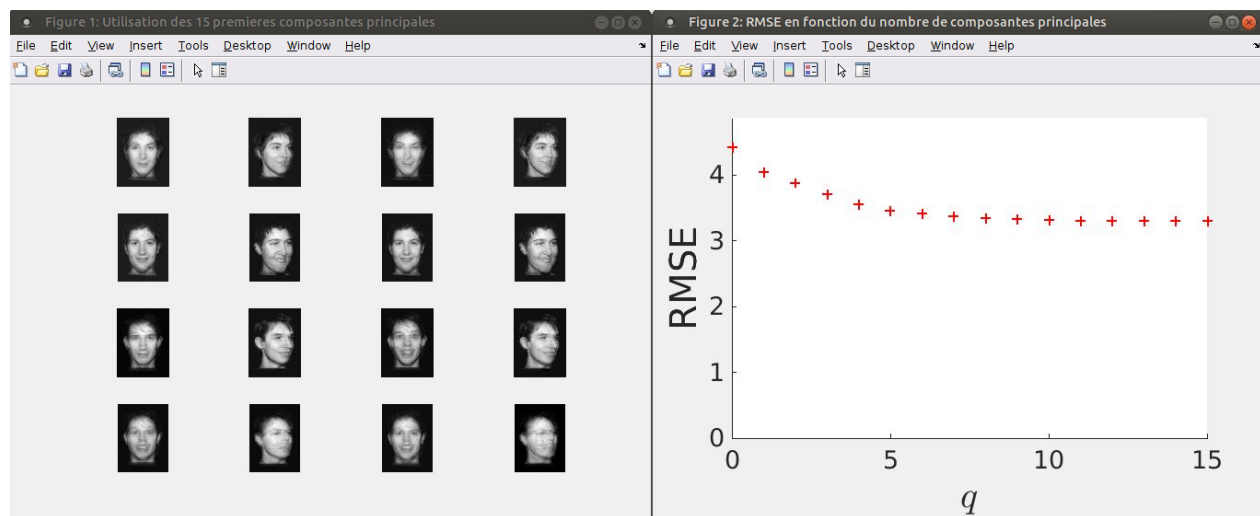


FIGURE 5 – <eigenfaces> et erreur quadratique optimal.

A partir des individus et leurs Eigenfaces obtenus en Figure 1, on peut appliquer les masques aux individus et voir comment les eigenfaces sont obtenus.

Ci-dessous le résultat du traitement de ce procès

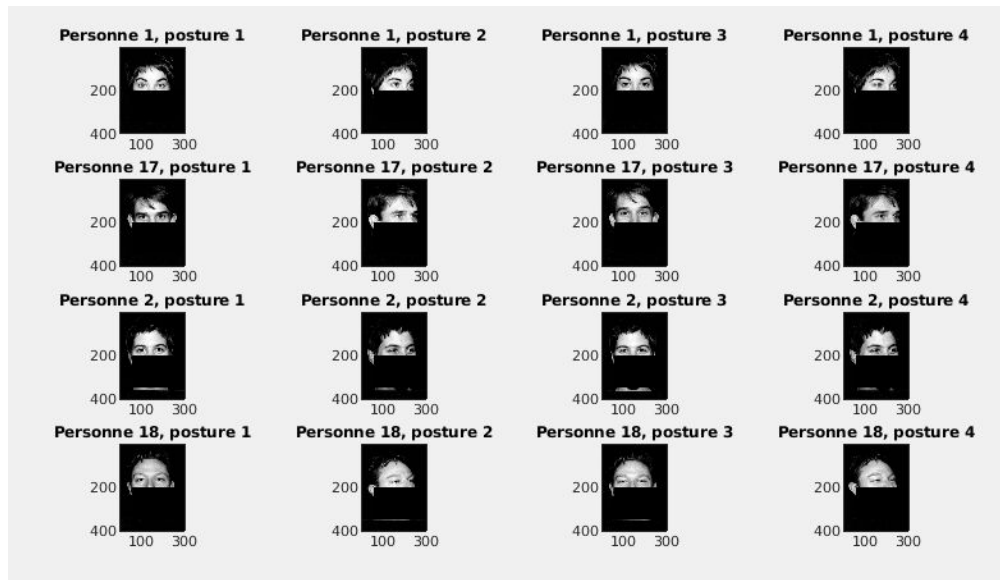


FIGURE 6 – eigenfaces sur les images masquées.

3 L'ACP et la méthode de la puissance itérée

Dans cette partie on utilise la très classique méthode de la puissance itérée avec déflation, qui consiste à renvoyer les couples propres directement dans l'ordre voulu. Ci-dessous l'algorithme de la puissance itérée pour trouver le couple propre dominant

MÉTHODE DE LA PUISSANCE ITÉRÉE

Données : une matrice $\mathbf{M} \in \mathbb{R}^{p \times p}$, un vecteur normé $\mathbf{x} \in \mathbb{R}^p$, une tolérance $\epsilon > 0$, it_{max} nombre max d'itérations.

Sortie : $(\lambda, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^p$ couple propre dominant de \mathbf{M}

Initialisation : $cv \leftarrow FALSE$, $i \leftarrow 0$, $\lambda \leftarrow \mathbf{x}^T \mathbf{M} \mathbf{x}$

1. Tant que $NOT.cv$:

```

 $\mu \leftarrow \lambda$ 
 $\mathbf{x} \leftarrow \mathbf{M} \mathbf{x}$ 
 $\mathbf{x} \leftarrow \mathbf{x} / \|\mathbf{x}\|$ 
 $\lambda \leftarrow \mathbf{x}^T \mathbf{M} \mathbf{x}$ 
 $i \leftarrow i + 1$ 
 $cv \leftarrow \left( \frac{|\lambda - \mu|}{|\mu|} \leq \epsilon \right).OR.(i \geq it_{max})$ 

```

2. Retourner (λ, \mathbf{x})

FIGURE 7 – Algorithme de puissance itérée

3.1 Réponse aux questions

★ Question 4 :

Soit $H \in \mathbb{R}^{n \times p}$ en connaissant les valeurs propres et les vecteur propres de $H^t H$ on peut connaître ceux de HH^t car :

Soit H une matrice rectangulaire de $\mathbb{R}^{n \times p}$, $H^t H$ est symétrique réelle semi-définie positive.

Par le théorème spectrale, $H^t H$ est orthogonalement diagonalisable.

Soit r le rang de H qui est aussi le rang de $H^t H$, 0 est valeur propre de multiplicité $n-r$. Soit λ (indice allant de 1 à r , strictement croissants, et à partir de l'indice $r+1$ à n sont nuls).

les n valeurs propres de $H^t H$ et (u indice i) une BON formée de vecteurs propres associés.

On pose pour tout i allant de 1 à r : u indice $i = \frac{1}{\sqrt{\lambda(\text{indice } i) H(\text{indice } i)}}$

Il est facile à montrer que la famille ainsi construite est orthonormée.

Pour tout i , u (indice i) appartient à $\text{Im}(H)$, et comme $\text{rg}(u \text{ (indice } i)) = \text{rg}(H)$, donc cette famille est une BON de $\text{Im}(H)$.

On la complète par u (indice allant de $r+1$ à p) de $\text{Im}(H)^\perp$ en une base de \mathbb{R}^t

★ **Question 5 :** Le script Matlab concernant la puissance itérée permet d'avoir les valeurs propres avec les différentes matrices (*voir code Matlab*) :

```
Valeur propre dominante (methode avec la grande matrice) = 9.305e+04
Valeur propre dominante (methode avec la petite matrice) = 9.305e+04
Valeur propre dominante (fonction eig) = 9.305e+04
```

FIGURE 8 – Calcul des valeurs propres

D'après ces résultats obtenus, on remarque bien que l'algorithme de la puissance itérée utilisé est plus performant on obtient le même résultat avec les deux matrices et ainsi avec la fonction *eig* de Matlab qui à partir d'une matrice M , retourne sa valeur singulière dominante.

Le calcul des valeurs propres sur les deux matrices engendre un erreur, ci-dessous l'erreur de calcul ainsi que l'écart relative entre les deux valeurs propres trouvées.

```
Erreur relative pour la methode avec la grande matrice = 9.894e-09
Erreur relative pour la methode avec la petite matrice = 9.927e-09
Ecart relatif entre les deux valeurs propres trouvees = 6.84e-08
```

FIGURE 9 – Temps d'exécution des algorithmes

Pour pouvoir pousser le raisonnement, on peut comparer le temps d'exécution de deux matrices. Ci-dessous le résultat obtenu :

```
Temps pour une ite avec la grande matrice = 7.391e-03
Temps pour une ite avec la petite matrice = 9.259e-04
```

FIGURE 10 – Temps d'exécution des algorithmes

→ **Discussion :** L'algorithme de utilisé pour calculer la valeur propre dominante avec la matrice petite est bien plus optimal que celui utilisé pour la grande matrice . En effet on choisira le calcul avec la petite matrice A^tA .

★ **Question 6 :** La fonction *eig* de Matlab permet bien évidemment de retourner les valeurs propres de la matrice, cependant ces valeurs propres ne sont pas triées et c'est à nous donc de les trier (*voir script exercice 1*).

D'autre part l'algorithme de la puissance itérée utilisé ici permet de faire les mêmes fonctions d'un seul coup ,c'est à dire qu'à la fin de l'algorithme on obtient les valeurs propres triées par ordre décroissante .

On préférera donc l'algorithme de la puissance itérée par sa grande précision et son temps d'exécution.

★ **Question 7 :**

D'après le résultats obtenus précédemment on choisira d'effectuer les calculs sur une matrice de type A^tA , car on minimise l'erreur de calcul ainsi que le temps d'exécution du programme , ce qui rend le calcul des valeurs propres plus optimal.

4 Conclusion

Cette partie nous a permis de découvrir les eigenfaces ainsi que le processus des calculs des valeurs propres en utilisant deux types de matrices (matrice grande et matrice petite d'après l'énoncé) ainsi découvrir les différents erreurs de calcul et l'écart type entre les deux méthodes , tout cela grâce à la méthode de la puissance itérée .

Enfin la partie de la projection sur les eigenfaces nous montre que pour avoir une reconnaissance optimale (excellente) il faudra avoir un erreur quadratique q faible , à ce stade l'image reconstruite et l'image réelle sont quasiment identiques.