
RECONNAISSANCE DE VISAGES : LA MÉTHODE DES EIGENFACES

Rapport de mini-projet

Année universitaire 2015-2016
ENSSAT, IMR2

NANTEL Maëlig

Introduction

Dans le cadre de la formation Informatique, Multimédia et Réseaux dispensé à l'ENSSAT, nous avons étudié des méthodes de traitement d'images. Il s'agissait essentiellement de comprendre la représentation informatique ainsi que les principaux modèles mathématiques permettant de traiter des images ou photos existantes. Durant la première partie de ce cours nous nous sommes intéressés à la problématique de détection des contours dans image. Dans cette seconde partie de ce cours, nous allons étudier la problématique de reconnaissance faciale. *Comment détecter un visage ? Comment comparer deux visages ? Comment avoir les meilleurs résultats de reconnaissances ?* Pour répondre à ces questions, nous avons implémenté une reconnaissance faciale par la méthode des eigenfaces dans un mini-projet encadré.

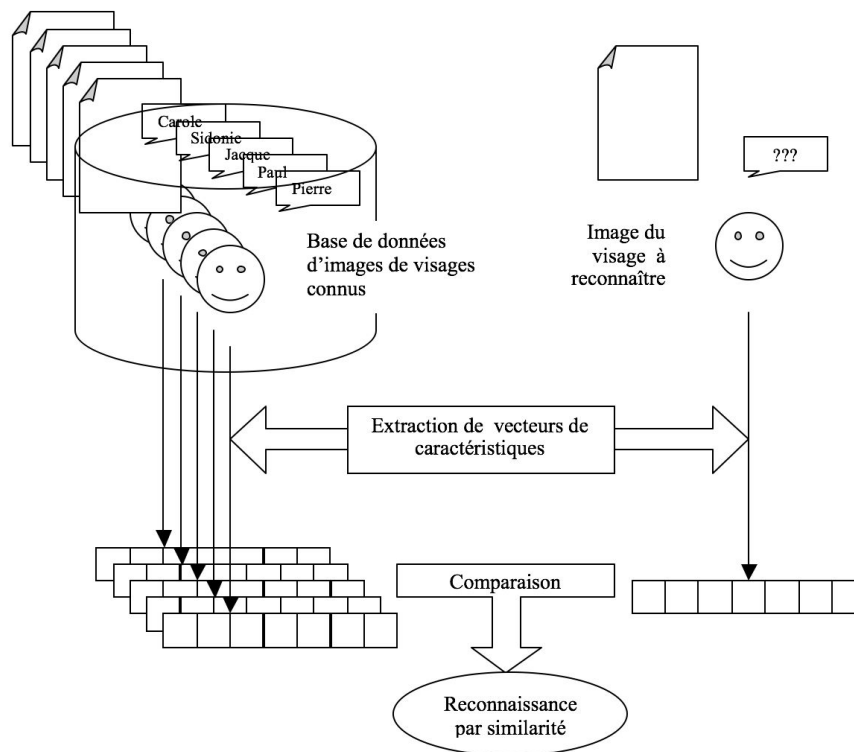
Le principe général de la reconnaissance faciale par eigenfaces consiste à extraire des caractéristiques d'une photographie représentant un visage afin d'en calculer ce que l'on appelle des descripteurs (ou vecteurs caractéristiques). On construit alors une base d'apprentissage à partir de plusieurs photographies de plusieurs individus. À partir de cette base d'apprentissage, il sera alors possible de reconnaître automatiquement un individu (présent dans la base) à partir d'une nouvelle photographie (non présente dans la base).

Ce rapport détaille plus précisément la méthode des eigenfaces et présente son implémentation sur le logiciel Scilab ainsi que les résultats obtenus.

1) Présentation de la reconnaissance faciale par la méthode des eigenfaces

a) Généralités sur la méthode

La reconnaissance faciale par la méthode des eigenfaces fonctionne en deux étapes. Tout d'abord, à partir d'une base d'images existantes, il faut construire une base d'apprentissage. Cette étape consiste à traiter chaque image de chaque individu et à en calculer un descripteur (un descripteur par image). Pour cela, on extrait des vecteurs caractéristiques (voir 1.b)). Une fois la base d'apprentissage construite, on va pouvoir l'interroger pour effectuer des tests de reconnaissance. Pour reconnaître une image, on prend une nouvelle image d'un individu connu. De cette image, on calculera son descripteur (comme pour la phase d'apprentissage) et on pourra ensuite le comparer à tous les descripteurs précédemment calculés. Le descripteur présentant le plus de similarités sera alors identifié comme l'individu. À partir du descripteur retenu, il sera alors possible de remonter à l'individu.



Fonctionnement général de la reconnaissance faciale par eigenfaces
(Source : mandrin-facial-recognition.googlecode.com)

Comme son nom l'indique, la reconnaissance par eigenfaces se base sur le calcul de vecteur propre (eigenvectors en anglais) et de valeurs propres (eigenvalues) d'une image. Nous avons vu dans la première partie du cours de traitement d'image qu'une image se modélisait comme un ensemble de pixels dans un support à deux dimensions. En

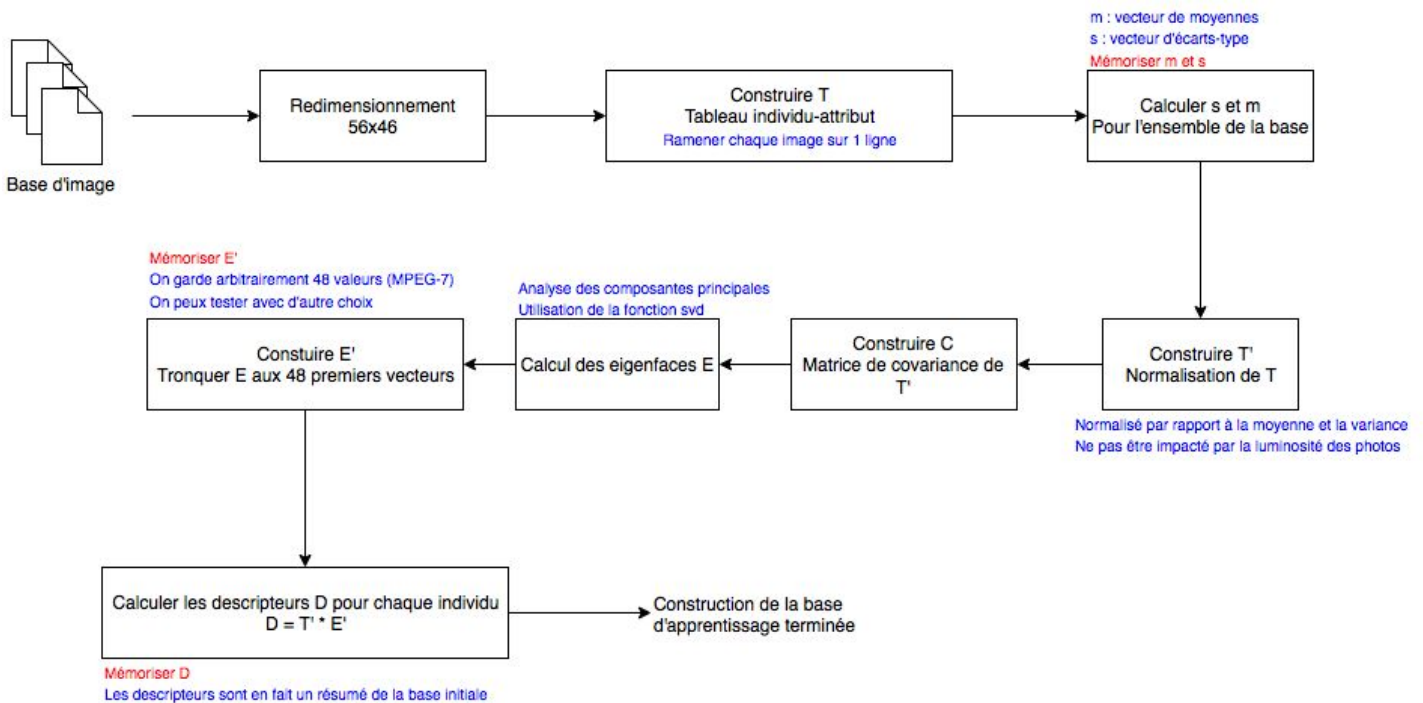
mathématique et en information, on peut donc représenter une image sous forme de matrice.

La méthode de reconnaissance faciale par eigenfaces se base sur le calcul d'un visage moyen d'une base d'image, puis sur les variations par rapport à ce visage moyen pour identifier chaque individu.

Pour des images de visage, il y a une très forte corrélation entre les différents points, ce qui fait que l'on va pouvoir réduire fortement l'information. On va alors perdre en qualité pour l'affichage, mais ce n'est pas un souci (l'objectif étant ici d'identifier un individu).

Nous allons maintenant voir plus en détail le fonctionnement des deux phases de cette méthode des eigenfaces.

b) Construction de la base d'apprentissage

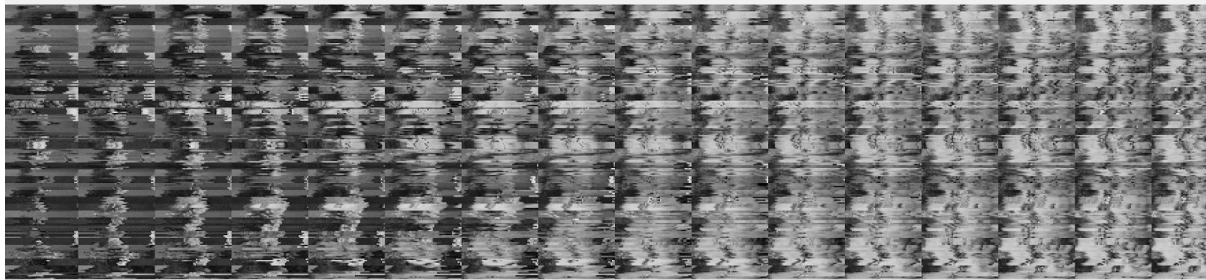


Fonctionnement de la phase d'apprentissage

À partir d'un certain nombre d'images de différents individus, on souhaite construire une base d'apprentissage permettant d'identifier un de ces individus à partir d'une nouvelle photo.

On commence par redimensionner nos images (si besoin). En effet, trop d'informations va être préjudiciable au traitement (« trop d'information tue l'information »). Ici nous avons fait le choix de travailler uniquement avec des images 56*46 pixels.

À partir des images chargées en mémoire et redimensionnées, nous pouvons construire un tableau individu attribut. L'individu correspond à la personne sur la photo et l'attribut est une photo le représentant. Lors de la construction de ce tableau, nous souhaitons linéariser chaque image. Comme nous l'avons vu précédemment, toutes les images sont modélisées par des matrices. Linéariser une image revient donc à l'écrire sous la forme d'un seul vecteur ligne. Chaque vecteur ligne ainsi créé représente une image et aura donc une taille de 2576 éléments ($56 \times 46 = 2576$). Le nombre d'éléments dans le tableau individu attribut dépend du nombre d'images utilisées pour la base d'apprentissage.



Affichage du tableau individu attribut

Nous n'avons pas connaissance des conditions dans lesquelles ont été prises les photos de la base d'images qui nous a été fournie. De ce fait, pour avoir de bons résultats, il est indispensable de normaliser les images pour qu'elles aient toutes la même moyenne et le même écart-type (sur leur histogramme). Cela permet de rendre la détection indépendante à l'éclaircissement de la scène. Toutes les images auront la même moyenne et le même écart type dans les niveaux de gris pour simplifier la reconnaissance. Cette moyenne et cet écart-type représentent donc les caractéristiques du visage moyen de la base d'apprentissage et sont donc constants pour toute cette dernière. Ces deux valeurs sont calculées directement sur le tableau individu attribut et il est nécessaire de les mémoriser, car nous verrons dans 1.c) que nous en aurons besoin dans la phase de reconnaissance.

Comme nous l'avons vu, l'approche de reconnaissance faciale par eigenfaces se base sur l'étude des vecteurs propres. Pour les calculer, on effectue ce que l'on appelle une analyse en composantes principales. Sur Scilab, cette opération s'effectue à l'aide de la fonction « `svd` ». On l'applique sur la matrice de covariance du tableau individu attribut normalisé. À partir de cette fonction, on peut donc récupérer les eigenfaces (ensemble de vecteurs propres).



Affichage des eigenfaces calculées

Nous allons maintenant déterminer tronquer ces eigenfaces à un certain nombre de colonnes (vecteurs). Cette troncature permettra de déterminer la taille des descripteurs que nous allons calculer pour chaque image. Après des études sur des bases de données

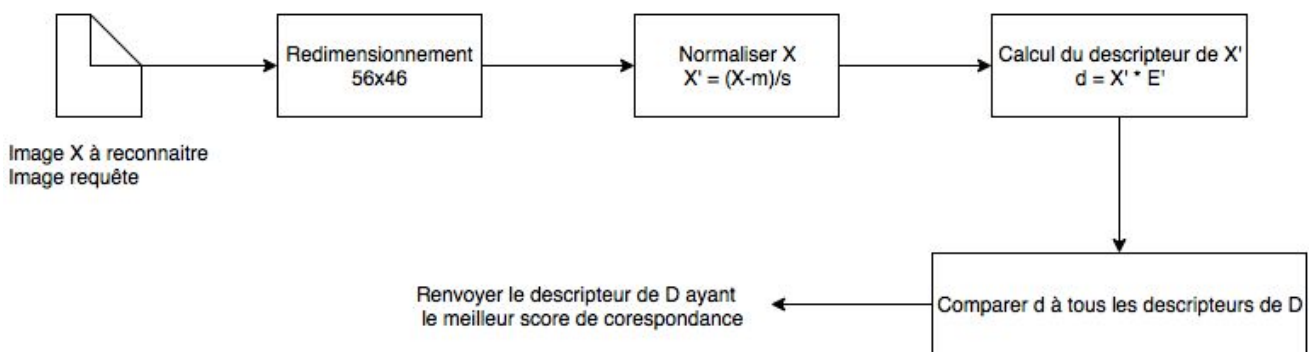
conséquentes, les auteurs de la norme MPEG-7 ont conclu que 48 vecteurs étaient un bon compromis. C'est donc ce nombre que nous allons retenir durant notre projet.

On obtient ensuite les descripteurs en appliquant une projection de l'image sur les eigenfaces. Ces descripteurs comportent évidemment beaucoup moins de données que les images initiales, mais cela sera suffisant pour effectuer notre reconnaissance. Prendre des descripteurs trop grands risquerait de noyer la reconnaissance dans trop d'informations, mais à l'inverse, avec une taille de descripteurs trop petite, nous n'aurons pas suffisamment d'informations descriptives sur chaque visage pour procéder à une reconnaissance fiable. On peut résumer les descripteurs comme étant un résumé de la base d'image initiale.

Après cette étape, la phase d'apprentissage est terminée. Nous avons calculé un descripteur pour chaque image chargée. Notons que certains éléments sont à mémoriser afin de pouvoir les utiliser lors de la prochaine phase de reconnaissance :

- La moyenne des images ;
- L'écart-type des images ;
- Les eigenfaces ;
- Les descripteurs ;
- Le vecteur d'identifiants.

c) Reconnaissance d'une nouvelle image



Fonctionnement de la phase de reconnaissance

Le schéma ci-dessus présente le fonctionnement de la phase de reconnaissance. À partir d'une nouvelle image d'un individu, nous souhaitons déterminer automatiquement de quel individu il s'agit. Nous rappelons que la nouvelle image (image requête) doit être celle d'un individu ayant déjà été analysé dans la base d'apprentissage pour pouvoir faire une reconnaissance correcte.

Nous l'avons vu dans 1.a), l'objectif est de calculer le descripteur de l'image requête afin de pouvoir le comparer aux descripteurs obtenus dans la phase d'apprentissage. Pour cela, il est donc nécessaire d'appliquer les mêmes transformations à l'image. Nous devons donc :

1. Redimensionner l'image requête en 56*46 pixels ;
2. Normaliser par rapport la moyenne et à la variance. Pour cette étape, il est donc nécessaire de récupérer la valeur de la moyenne et de la variance de la phase d'apprentissage.

À partir de là, nous pouvons calculer le descripteur de l'image requête. Comme nous l'avons vu lors de l'apprentissage, le descripteur correspond en fait à la projection de l'image sur les eigenfaces (calculés lors de l'apprentissage).

Il faut maintenant comparer le descripteur de l'image requête à l'ensemble des descripteurs calculés lors de l'apprentissage (il faut donc également les récupérer). On comprends bien qu'en utilisant une image requête n'ayant pas été utilisée lors de la phase d'apprentissage, il sera impossible d'obtenir une correspondance parfaite. De ce fait, il existe plusieurs méthodes pour choisir le descripteur ayant la meilleure correspondance ; nous avons décidé de retourner uniquement le plus proche voisin (donc le descripteur ayant le moins de différence avec le descripteur requête). Pour le calcul de la différence entre les deux vecteurs descripteurs, on compare les normes du delta (différence) et on choisit le descripteur ayant la norme minimale (donc la distance la plus faible avec le descripteur requête).

À partir de l'index du descripteur ayant la meilleure correspondance, il est directement possible de remonter à l'individu identifié. Pour cela, il faut accéder au vecteur d'identification qui a été construit lors de la phase d'apprentissage. L'individu reconnu sera donc la valeur de ce vecteur d'identification à l'index du descripteur retenu.

2) Détails sur l'implémentation

Nous avons maintenant une bonne compréhension de la reconnaissance de visages par la méthode des eigenfaces. Nous allons maintenant implémenter cette méthode sur le logiciel Scilab. Cette partie ne présente pas tous les détails de la mise en œuvre de la méthode, mais présente cependant les spécificités du code produit, ainsi que les choix d'implémentation.

a) Tests automatiques

Tout au long du développement, j'ai souhaité mettre en place des tests automatiques. Scilab n'étant pas particulièrement prévu pour ce genre de tests, les vérifications que j'ai mises en place sont donc des assertions à vérifier lors de chaque appel aux fonctions. La majorité de ces tests ont été réalisés en suivant le principe de la méthode « Test Driven Development » (développement piloté par les tests). En effet, à l'issue de la phase d'étude de la méthode, nous connaissions déjà la taille de chaque vecteur, matrice ou hypermatrice à produire. Ces tests permettent donc d'avoir une première validation sur le résultat des fonctions écrites.

b) Choix personnels

Plusieurs choix personnels ont été faits durant le développement de la phase de développement, ayant un impact sur les résultats observés, mais restants cependant conformes à la méthode des eigenfaces :

i) Chargement des images pour l'apprentissage

La base d'image qui nous a été proposée pour ce mini-projet était composée de 400 éléments : 40 individus différents ayant chacun 10 photos. On souhaite allouer une partie de cette base pour la phase d'apprentissage, l'autre partie étant réservée pour les tests. Pour construire la base d'apprentissage, nous avons vu qu'il était préférable de traiter l'ensemble des individus. Nous allons donc utiliser un nombre entre 1 et 9 images pour l'apprentissage. Dans la partie 3) nous adapterons cette valeur afin d'observer son impact sur les résultats.

Le choix que j'ai retenu pour le chargement des images et de le faire par ordre croissant. Si on souhaite charger 5 images, le programme prendra dans l'ordre :

1. 1.pgm
2. 2.pgm
3. ...

Ce choix est discutable. En effet, on peut imaginer que sur les premières photos de la base, des consignes précises aient été données quant à l'expression de visage à prendre. On pourrait envisager de prendre 5 images aléatoirement dans chaque répertoire et observer l'impact sur les résultats (*non implémenté par manque de temps*).

ii) Construction du tableau individu attribut

Comme nous l'avons vu précédemment, nous avons construit un tableau individu attribut. Nous avons présenté un tableau à deux dimensions, la première correspondant à un identifiant d'individu et la seconde étant un vecteur image. Pour des raisons pratiques lors de l'implémentation, j'ai choisi de séparer ces informations en deux tableaux. Ainsi, lors du chargement des images, j'ai construit un vecteur d'identifiants indiquant pour chaque image chargée, l'individu associé. Le tableau contenant les vecteurs images a ensuite été construit.

iii) Choix du plus proche descripteur

Lors de la reconnaissance de l'individu, nous avons vu qu'il s'agissait en réalité de la comparaison de descripteurs. Il existe plusieurs méthodes pour retourner le meilleur descripteur. Dans la solution implémentée, on se contente de toujours retourner le plus proche voisin (même si la distance reste importante). Une autre solution serait de retourner le descripteur si sa distance est inférieure à un certain seuil (à fixer... voir module précédent dans le cours de traitement d'image). Si la distance est supérieure, on annonce alors que l'individu n'est pas connu par la base. On pourrait également envisager d'ajouter cette photo à la base d'apprentissage pour de prochaines requêtes.

3) Tests de la solution

a) Test « unitaire »

Une fois l'implémentation terminée, on peut tester de manière « unitaire » la reconnaissance d'un individu. Pour cela, on choisit une des photos n'ayant pas été utilisées dans l'apprentissage et on compare l'individu détecté à celui que l'on a introduit dans le système pour la requête.



Résultat d'une reconnaissance réussie (à droite la photo requête, à gauche la photo « 1.pgm » de l'individu reconnu)

Ces tests permettent d'avoir un premier aperçu des résultats du programme. On observe ci-dessus la capacité à reconnaître correctement un individu à partir d'une nouvelle photo. Cependant, ce n'est pas suffisant pour déterminer le taux de réussite.

b) Calcul de taux de réussite

Pour déterminer le taux de réussite de la méthode, il est nécessaire de tester, pour chaque individu, toutes les images n'ayant pas été utilisées lors de l'apprentissage. On compte alors le nombre d'erreurs de reconnaissance pour permettre de déterminer un taux de réussite.

Nous avons donc un aperçu des résultats de la méthode que nous avons implémenté. Pour une base d'image d'un total de 400 images, nous pouvons être satisfaits de nos résultats. Il faut cependant garder à l'esprit que les résultats vont être totalement dépendant de la taille et de la qualité de la base d'image fournie à notre système.

Nombre d'images pour l'apprentissage	Taux de réussite
1	69,44 %
3	81,01 %
5	88,50 %
7	96,67 %
9	95 %

Comme on pouvait s'y attendre, plus le nombre d'images est important dans la base d'apprentissage, plus les résultats sont bons lors de la reconnaissance. Cependant, on remarque que lorsqu'on utilise 9 images pour l'apprentissage (sur les 10 disponibles), on descend à 95 %. On en conclut donc que prendre trop d'images peut avoir un impact négatif sur la reconnaissance (trop d'informations). L'erreur peut s'expliquer par deux images ressemblantes de deux individus différents. Il serait intéressant de comparer ces taux de réussite avec un chargement des images dans un ordre différent (voir 2.b)).

Conclusion

Ce mini projet nous a permis de bien comprendre la problématique de reconnaissance de visages. En mettant en œuvre la méthode des eigenfaces, nous avons pu apprendre à manipuler et à interpréter des modélisations mathématiques appliquées à une image. Nous pouvons conclure que la méthode des eigenfaces impose de travailler avec des images représentant uniquement des visages et est également très sensible à toutes les petites variations dans la position du visage. De ce fait, c'est une méthode particulièrement efficace sur des images type « photo d'identité ».