Projet FBI (Face Basic Identification)

Gilles Simon et Stéfane Paris

{Gilles.Simon,Stefane.Paris}@loria.fr

Introduction

L'objectif de ce projet est d'implémenter en Matlab un algorithme de reconnaissance de visages utilisant l'analyse spectrale. Nous nous appuierons sur une version simplifiée de la méthode de Zhong et Defée [1]. Dans ses grandes lignes l'algorithme effectue avant tout un apprentissage pour identifer des visages. Puis une fois l'apprentissage fait, l'algorithme est capable (plus ou moins bien) d'identifier de nouveaux profils de visages appris.

Nous vous fournirons une base d'images de visages observés sous des angles et des expressions différents (à chaque visage correspond 10 images présentant ce visage sous différents profils). Ces images seront réparties (certaines dupliquées) dans deux bases ayant des fonctions différentes :

- Une **base d'apprentissage** contenant des visages *identifiés*. Cette base sera utilisée à la fois lors de l'apprentissage et de la reconnaissance.
- Une **base de test** contenant des images de visages inconnus que l'algorithme devra reconnaître parmi les images stockées de la base d'apprentissage.

Algorithme

Extraction des patterns AC/DC d'une image

L'algorithme est basé sur la transformée en cosinus discrète (DCT) de blocs de taille 4x4 extraits des images (avec un recouvrement de 2 pixels horizontalement et verticalement). Chaque bloc correspond à un *pattern AC* et une valeurDC, obtenus de la manière suivante (cf. figure 1) :

1. On calcule la DCT de la matrice des niveaux de gris du bloc 4x4,
2. On quantifie la DCT en calculant la partie entière de la DCT divisée par un paramètre de l'algorithme appelé QP.

La valeur de DC est le coefficient (1,1) de la matrice ainsi obtenue, le *pattern AC* est le vecteur formé des 15 coefficients restant (peu importe l'ordre d'extraction de ces coefficients, du moment qu'on utilise toujours le même ordre dans la suite de l'algorithme).

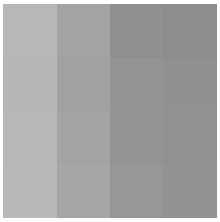
Le *pattern AC* correspond à la description fréquentielle du bloc, nous l'utiliserons pour la reconnaissance des visages. Un rapide calcul montre que la valeur de DC est la moyenne des niveaux de gris du bloc. Nous allons utiliser cette valeur pour normaliser le *pattern AC* vis-à-vis des conditions d'éclairage.

Normalisation des *patterns AC* vis-à-vis de la luminance

Pour éviter que les résultats de l'algorithme de reconnaissance dépendent de la luminosité des photos, on ramène la luminance moyenne de chaque image à la luminance moyenne de toutes les images de la base d'apprentissage. La luminance moyenne d'une image j est donnée par :

$$DC_{mean}(j) = \frac{1}{N} \sum_{i=1}^{N} DC_i(j),$$

Image originale



Bloc 4x4

183	163	143	141
182	161	148	144
183	164	147	145
184	166	151	146

Niveaux de gris

637.75	59.75	16.25	-0.12
-6.09	2.29	1.63	-0.71
0.75	1.98	-0.75	-1.09
-0.99	1.78	0.67	-2.29

DCT

106	10	3	0
-1	0	0	0
0	0	0	0
0	0	0	0

$Arrondi(DCT * DC_{all}/DC_{mean}/QP)$ avec $QP = 6$

106

Valeur de DC

-1	0	0	10	0	0	0	3	0	0	0	0	0	0	0

pattern AC

Figure 1: Extraction du *pattern AC* et du DC d'un bloc 4x4 extrait au milieu de l'image en haut à gauche.

où l'indice i parcourt l'ensemble des blocs de l'image j et $DC_i(j)$ est la valeur DC du bloc i de l'image j.
Soit

$$DC_{all} = \frac{1}{M} \sum_{j=1}^{M} DC_{mean}(j)$$

la moyenne des $DC_{mean}(j)$ obtenus sur l'ensemble des M images de la base d'apprentissage. Toute image j utilisée par l'algorithme (image de la base d'apprentissage ou de test) doit être normalisée à l'aide du ratio :

$$R(j) = \frac{DC_{all}}{DC_{mean}(j)}$$

Plus précisément, les *patterns AC* de l'image j ($1 \leq j \leq M$) manipulée par l'algorithme doivent être multipliés par $R(j)$ *avant* d'être quantifiés.

Remarque : La valeur de DC_{all} dépendant uniquement de la base d'apprentissage, nous vous conseillons de ne la calculer qu'une seule fois et de l'enregistrer dans un fichier.

Extraction des *patterns AC* les plus représentés dans la base d'apprentissage

Une manière simple de comparer deux images de visages (par exemple, une image identifiée avec une image non identifiée) serait d'extraire tous les *patterns AC* de l'image apprise et tous les *patterns AC* de l'image de test puis de compter combien de *patterns* elles partagent. L'image de la base de référence la plus proche d'une image test serait alors celle qui partagerait le plus grand nombre de *patterns* avec l'image test. À noter que les *patterns AC* étant quantifiés, leur comparaison est aisée et leur nombre limité (plus la valeur de QP est grande, plus le nombre de *patterns* différents est petit).

Cette procédure serait toutefois peu performante car certains *patterns* sont fréquents dans les images, y compris dans des images de visages ne correspondant pas à la même personne. Inversement, certains *patterns* sont rares voire uniques, et n'ont pas plus de chance d'être trouvés dans une image du visage recherché que dans l'image d'un visage différent.

Afin d'obtenir des résultats plus pertinents, la méthode de Zhong et Defée commence par identifier les *patterns AC* les plus présents dans la base d'apprentissage, puis à décrire les images de référence et les images de test en considérant ces *patterns* uniquement. Ce type d'approche est en fait relativement classique et a été transposé à de nombreux autres problèmes.

L'algorithme commence donc par construire un histogramme des *pattern globaux* à toute la base d'apprentissage. Autrement dit, il compte le nombre de fois que chaque *pattern AC* (normalisé et quantifié) apparaît *dans toute la base d'apprentissage*. Les *patterns globaux* sont ensuite triés par ordre décroissant de leur nombre d'apparitions et on ne conserve que les $N\_AC\_PATTERN$ premiers *patterns AC* où $N\_AC\_PATTERN$ est un second paramètre

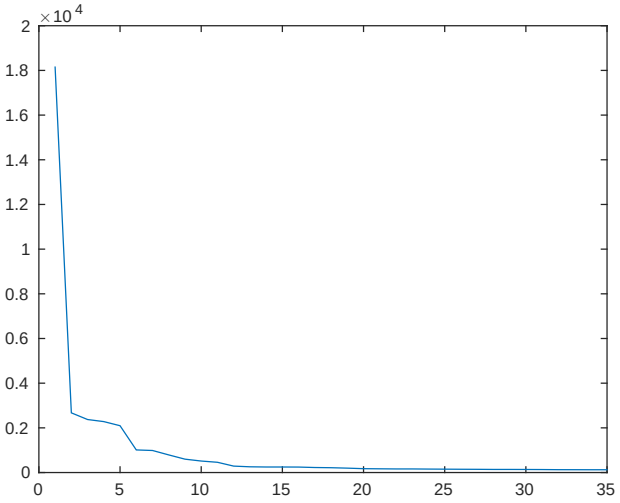


Figure 2: Nombre de fois qu'apparaissent les *patterns AC* N°1 à 35 d'une base d'apprentissage de 25 images.

de l'algorithme. La figure 2 montre les valeurs obtenues pour les $N\_AC\_PATTERN = 35$ premiers *patterns* d'une base d'apprentissage constituée de 25 images (5 visages × 5 images par visage), avec QP = 22.

Remarque : cette partie de l'algorithme est en $O(NM^2)$, où N est le nombre de blocs d'une image et M le nombre d'images dans la base d'apprentissage : elle est donc très coûteuse et peut prendre beaucoup de temps à l'exécution suivant le nombre d'images dans la base d'apprentissage. Tout comme pour la valeur de DC_{all}, nous vous recommandons donc de conserver les *patterns AC* de la base d'apprentissage dans une fichier que vous rechargerez à chaque fois.

Comparaison de deux images

Une fois l'histogramme des *patterns globaux* calculé sur la base d'apprentissage, on peut comparer une image i (issue de la base d'apprentissage) avec une image j (issue de la base de test) en utilisant la procédure suivante :

1. calcul des *patterns AC* (normalisés et quantifiés) de l'image i,
2. calcul de l'histogramme H_i de l'image i. Cet histogramme est obtenu en comptant le nombre de fois que les *patterns globaux* N°1 à $N\_AC\_PATTERN$ *de la base d'apprentissage* (obtenus de la manière décrite ci-dessus) apparaissent dans l'image i. H_i est donc un vecteur de taille $N\_AC\_PATTERN$ dont l'ordre des valeurs doit correspondre à celui des *patterns globaux* de la base d'apprentissage ;
3. calcul des *patterns AC* (normalisés et quantifiés) de l'image j,
4. calcul de l'histogramme H_j de l'image j (même procédure qu'en 2).
5. comparaison des deux histogrammes à l'aide de la distance *city-block* :

$$B_{i,j} = \sum_{k=1}^{N\_AC\_PATTERN} |H_i(k) - H_j(k)|$$

Les visages représentés dans les deux images sont supposés être d'autant plus proches que cette distance est petite. La figure 3 montre trois exemples d'histogrammes obtenus de cette manière : l'histogramme H_1 a été calculé sur une image de test, l'histogramme H_2 sur une image d'apprentissage présentant le même visage que dans l'image de test et l'histogramme H_3 sur une image présentant un visage différent de celui de l'image de test. Nous obtenons $B_{1,2} = 236 < B_{1,3} = 371$, ce qui est un résultat conforme à ce que l'on attend.

Rechercher les TOP_RANK images de référence les plus proches d'une image de test donnée consiste simplement à calculer la distance *city-block* entre l'image de test et toutes les images de référence, et à retourner les TOP_RANK premières images de la liste des images de référence, ordonnée par ordre croissant de la distance (quatre exemples de résultats sont présentés en figure 4)

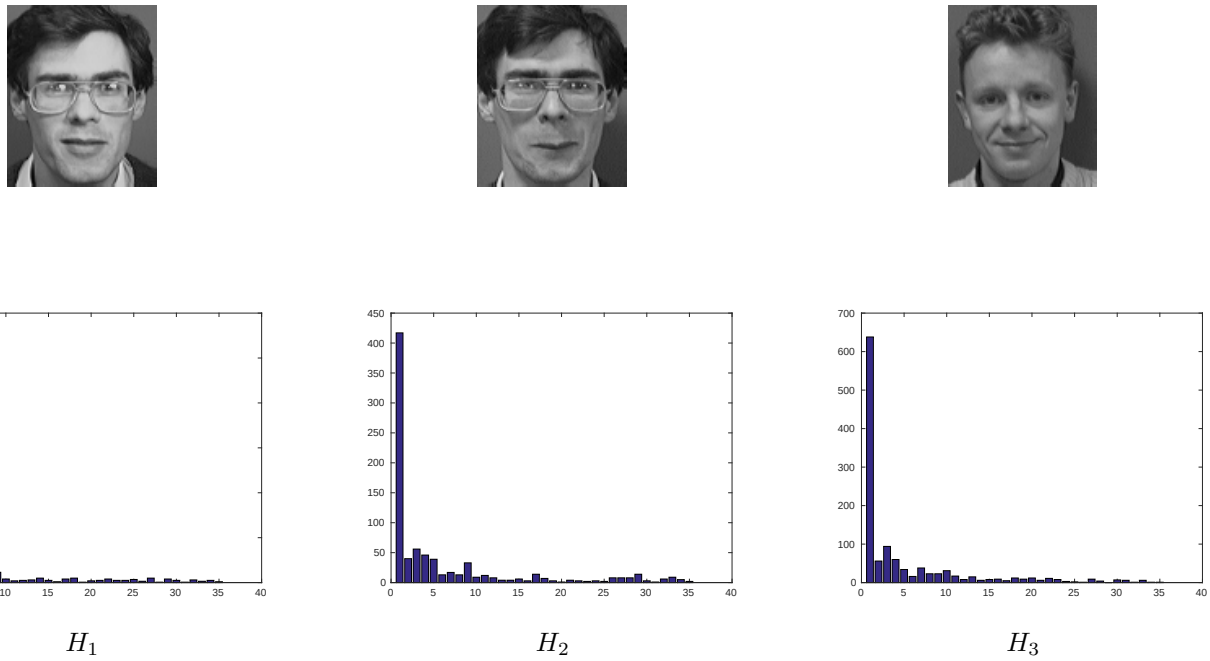


Figure 3: Trois histogrammes de *patterns* AC correspondant à deux visages identiques et un visage différent.

Remarque : on ne vise pas la perfection qui est impossible dans ce domaine ! Votre programme ne pourra pas trouver à chaque fois un visage identique à celui de l'image de test en première position du top rank. Ce qui compte c'est que statistiquement les visages identiques se retrouvent plus souvent dans les premières places du top rank que les visages différents. À vous de réfléchir à des critères qui vous permettront d'évaluer la pertinence des résultats et de régler les paramètres de l'algorithme de manière à mieux satisfaire ces critères. Nous vous demandons de décrire le(s) critère(s) utilisé(s) à cette fin dans votre rapport.

*References

[1] D. Zhong and I. Defée. Dct histogram optimization for image database retrieval. *Pattern Recogn. Lett.*, 26(14):2272–2281, Oct. 2005.

Figure 4: Quelques exemples de résultats : la colonne de gauche montre l'image de test, les colonnes suivantes montrent les 10 images de référence les plus proches de l'image de test, par ordre croissant de la distance city_block.