

TP de vision par ordinateur :

Détection et suivi de visages par pseudo- ondelettes de Haar

1. Introduction

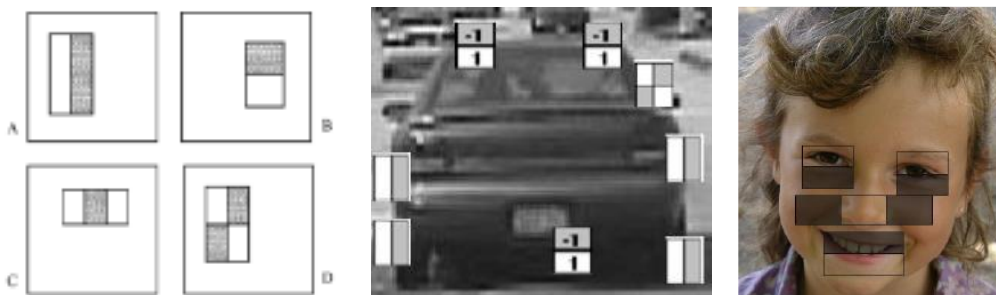
La méthode Viola et Jones a été publiée par Paul Viola et Michael Jones dans le journal scientifique IJCV en 2001 (Viola & Jones, 2001). Elle permet la détection en temps réel de classes d'objets dans les images numériques de manière robuste (taux de détection très élevé et taux de faux positifs très faible). Cette méthode est basée sur la description d'une classe par des caractéristiques pseudo-Haar très peu coûteuses en calcul. Les caractéristiques les plus représentatives d'une classe donnée sont sélectionnées dans une phase d'entraînement. La détection est basée sur une cascade de plusieurs couches de classifieurs faibles formant un classifieur fort.

L'objectif de ce TP est d'implémenter l'algorithme de Viola-Jones avec un classifieur pré-entraîné pour détecter des visages dans des images puis de l'adapter au suivi de visage dans des séquences vidéo.

2. Principe de la méthode :

Un objet est codé par un ensemble de filtres binaires (primitives) de différents types, à différents endroits de l'image et à différentes échelles. Les filtres retenus pour chaque classe (type, position et taille) sont sélectionnés par un entraînement sur une grande base de d'images. Cet entraînement consiste à trouver les filtres dont la réponse pour une classe donnée est la plus stable. C'est-à-dire que l'objet est détecté dans l'image si la somme des réponses à tous les filtres est supérieure à un seuil.

Chaque masque est appelé « classifieur faible » car il ne permet de reconnaître que très peu d'exemples. Mais la combinaison de plusieurs classifieurs faibles va constituer un classifieur fort.



Représenter une forme avec tous les classifieurs faibles possibles (type, position, taille) est impossible car cela représenterait des centaines de milliers de valeurs (bien plus que les pixels de l'image). On va donc se limiter à un nombre restreint de classifieurs faibles positionnés aux endroits adéquats dans l'image.

Evidemment, chaque classe d'objet nécessite un jeu de classifieurs adaptés avec des tailles adaptées et à des endroits spécifiques. La difficulté est alors d'apprendre à partir d'une base d'exemples à générer les meilleurs classifieurs faibles pour chaque classe. Cette tâche est réalisée grâce à l'algorithme Adaboost. Cette étape d'apprentissage ne sera pas abordée ici. Les classifieurs sélectionnés sont donc fournis. Ils ont été entraînés sur une base de plusieurs milliers d'images « visage » et « non-visage ».

3. Travail à réaliser :

Dans ce TP, nous allons mettre en œuvre l'algorithme de détection de visages par la méthode de Viola et Jones. Pour cela, un ensemble de filtres qu'il faudra organiser en couches (cascade) est fourni.

Structure des données fournies :

- Une base d'images test comportant des visages (humains).
- Un ensemble de classifieurs faibles préalablement entraînés dans le fichier «trainedClassifiers.mat». Chaque ligne de la matrice « selectedClassifiers » représente un classifieur faible avec les propriétés suivantes :
 - haar = classifier(1); type du classifieur
 - pixelX = classifier(2); pixelY = classifier(3); position X, Y dans l'imagette
 - haarX = classifier(4); haarY = classifier(5); taille
 - classifier(9); classifier(10) ; seuil min et seuil max.
 - classifier(12); poids du classifieur

Ces classifieurs seront rangés en couches elles-mêmes organisées en « cascade de Haar ». Les couches sont constituées comme suit :

Couche	Calssifieurs
1	1 à 2
2	3 à 12
3	13 à 20
4	21 à 40
5	41 à 70
6	71 à 150
7	151 à 200

Algorithme Cascade :

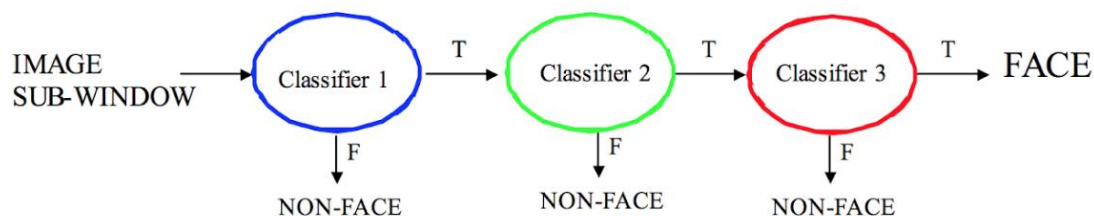
Un classifieur est validé si la valeur de sa réponse pour une imagerie est située dans l'intervalle [seuil min, seuil max].

Une couche est validée (sortie à 1) si la somme des poids de ses classifieurs validés est supérieure à un seuil. La sortie d'une couche $h(x)$ pour une image d'entrée x est ainsi obtenue par seuillage de la somme pondérée des classifieurs faibles sélectionnés :

$$h(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{sinon} \end{cases}$$

Les α_t sont des coefficients calculés à partir de l'erreur statistique de chaque classifieur faible.

Le but, est de mettre en œuvre la classification pour chaque couche en partant de la couche d'entrée. Dès qu'une couche de classifieurs donne un résultat nul, la recherche s'arrête et le résultat est nul (visage non détecté). Ceci permet de rendre la détection plus rapide car la recherche devant se faire dans toute l'image, la majorité des régions ne contiendra pas de visage. Si toutes les couches donnent une sortie à 1, alors le visage est détecté.



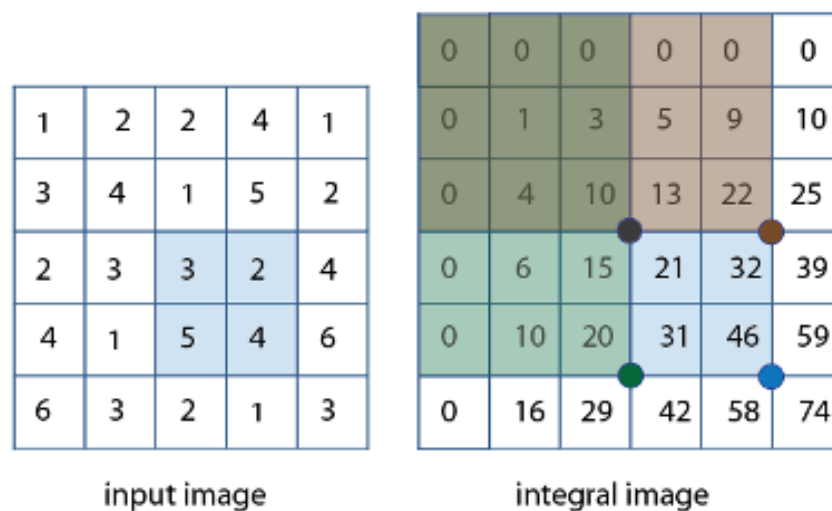
Mise en œuvre de la détection

L'algorithme de détection de visages consiste donc à effectuer une recherche par fenêtre glissante en calculant le score de la cascade à chaque position dans l'image. La taille du visage n'étant pas connue par avance, il faut donc répéter le balayage pour différentes tailles de la fenêtre. Afin d'accélérer le traitement, il est plus pratique de redimensionner l'image plutôt que de redimensionner chacun des filtres.

On utilisera une fenêtre glissante de taille 19x19 (c'est la taille utilisée lors de l'entraînement). Commencer par l'image à sa taille normale. Effectuer une recherche. Si aucun visage n'est trouvé, redimensionner l'image en appliquant un facteur d'échelle de 0,8. Recommencer jusqu'à ce qu'un visage soit détecté ou que la taille de l'image soit inférieure à deux fois la taille de la fenêtre glissante.

Calcul de la sortie d'un classifieur faible :

La sortie d'un classifieur faible est égale à la somme des régions « noires » du masque moins la somme des régions « blanches » du masque. Afin d'accélérer le traitement, Viola et Jones ont introduit la notion d'image intégrale. Cette image sera calculée une seule fois pour une échelle donnée. Elle permet de calculer le score avec quelques additions et quelques accès mémoire comme nous le montre la figure ci-dessous. En effet, la somme des valeurs dans le rectangle bleu se déduit des valeurs de ses coins dans l'image intégrale ($46 - 22 - 20 + 10 = 14$).



Quelques indications :

Ecrire les fonctions ci-dessous qui calculent respectivement le score d'un classifieur faible et le score d'une couche de classifieurs en fonction d'un seuil donné.

output = calcHaarVal(image, haar, pixelX, pixelY, haarX, haarY)

output = cascade(classifiers, image, thresh) qui calcule

4. Suivi de visages dans une séquence vidéo

Une fois le détecteur validé, il peut être utilisé pour détecter puis suivre le mouvement d'un visage dans une séquence vidéo. Il s'agit de rechercher un visage dans la première image dans sa totalité, puis d'utiliser le résultat afin de limiter la recherche dans les images suivantes.

La zone de recherche sera calculée grâce à un modèle d'évolution (vitesse nulle, vitesse constante, accélération constante) qui prédira la position du visage suivi dans

la prochaine image. Les paramètres de ce modèle sont évalués et mis à jour au cours du suivi sur la base des positions $[X_i, Y_i]$ du visage dans les n images précédentes. Une fenêtre de recherche sera sélectionnée autour de la prédiction du modèle avec une marge à définir $[dX, dY]$.

Implémenter ce cet algorithme et testez-le sur les vidéos fournies.

Quelques indications :

Code Matlab pour charger une vidéo image par image

```
readerobj = VideoReader('videos\003.mp4', 'tag', 'myreader1');  
vidFrames = read(readerobj);  
numFrames = get(readerobj, 'NumberOfFrames');
```