Morphing appliqué à la reconnaissance de visages

Nicolas SCAPEL - Maxime SERMESANT
École Centrale Paris

En collaboration avec Damien FENEYROU
MATRA Systèmes et Information

17 février 1999

Résumé

La reconnaissance de visages est une opération complexe qu'il serait très utile de faire réaliser par des ordinateurs, dans des domaines tels que l'authentification, la visiophonie, les interfaces homme-machine et l'intelligence artificielle. Certains produits sont déjà opérationnels, et font appel à une grande variété de techniques : modèles déformables, filtrages par ondelettes, réseaux de neurones à couches cachées, analyse statistique. La plupart se révèlent délicats à mettre en oeuvre et ne donnent de résultats satisfaisants que sur des bases de visages spécifiques.

L'objectif du séminaire est d'étudier et de comparer ces différentes méthodes, puis de mettre en oeuvre un algorithme inspiré des travaux concernant les modèles déformables. Ainsi, nous avons décomposé la tâche de reconnaissance en deux étapes. Une première étape consiste en un recalage non-rigide d'une ellipse sur le gradient de l'image. Elle permet de détecter le visage dans l'image et d'en extraire une représentation normalisée. Ce recalage fait appel à une technique d'optimisation de type descente de gradient à pas adaptatif. La seconde étape consiste à corréler un visage normalisé avec les différents visages d'une base, pour chercher à identifier un individu. Le critère de décision correspond à un écart quadratique calculé après recalage rigide entre chaque visage de la base et le visage à identifier. Ces différentes étapes ont été implémentées sous Matlab et nous présentons les résultats obtenus sur deux bases de visages.

Table des matières

Chapitre 1

INTRODUCTION

1.1 Les enjeux

La reconnaissance des visages est pour nous une faculté naturelle, familière, bien plus que les signaux que nous utilisons aujourd'hui pour communiquer avec une machine. Le développement de tels algorithmes s'inscrit dans une démarche visant à rendre l'ordinateur plus "humain", et va de pair avec la reconnaissance vocale et la reconnaissance gestuelle.

La reconnaissance de visages trouve également de nombreuses applications dans le domaine de la sécurité. Citons la restriction d'accès à un lieu ou une machine, et la criminologie (retrouver un visage dans une base de données à partir d'un portrait robot), pour lesquels des systèmes sont déjà opérationnels.

La plupart des algorithmes de reconnaissance de visages cherchent à extraire un code descriptif réduit du visage (entre 10 et 500 paramètres), puis à rattacher ce code à une classe statistique correspondant à un individu. Pour une application de vidéoconférence ou de visiophonie, on peut se servir de ce code pour ne transmettre que les "variables intrinsèques" du visage sur le réseau, et effectuer une synthèse du visage à la réception (opération inverse du codage). Ainsi pour transmettre une image de visage animée, on pourrait diviser la bande passante d'un facteur 1000 (e.g. de 10mB.s à 10kB.s). Le standard international de vidéo numérique MPEG 4 intègre déjà de telles fonctions.

Enfin, un tel algorithme permettrait de concevoir un moteur de recherche audiovisuel intelligent auquel on pourrait demander "Trouver une séquence vidéo dans la base de données Hollywood.com dans laquelle figurent Marilyn Monroe".

1.2 L'authentification par reconnaissance de visages: limitations

L'identification d'une personne par ce procédé ne sera jamais assez fiable pour des applications de haute sécurité:

- Un humain moyen atteint un taux de reconnaissance de 99%. Peut-on le dépasser?
- Toute population comporte environ 1% de vrais jumeaux... Les humains peuvent parvenir à les reconnaître avec de l'expérience, qu'en est-il des ordinateurs?
- Comment l'ordinateur pourra-t-il comprendre qu'il n'a affaire qu'à une photo d'une photo, ou pire, d'un masque?

Une méthode plus fiable d'authentification automatique, sans-contact (contrairement aux empreintes digitales), a été découverte par John Daugman en 1993: la reconnaissance de l'iris. Bien que de faible surface (environ 1 cm^2), on peut extraire 267 paramètres intrinsèques indépendants de ses variations locales de phase, propres à chaque individu et invariantes au cours de son existence. Par comparaison, une analyse globale du visage ne conduit le plus souvent qu'à une dizaine de distances caractéristiques. Cependant, cette méthode introduit des contraintes fortes sur le dispositif et les modalités d'acquisition.

Il reste néanmoins de nombreuses applications pour ce procédé:

- Authentification dans un système moyenne-sécurité (ordinateur personnel, bâtiment public)
- Authentification dans un système haute sécurité, en conjonction avec un autre procédé (empreintes digitales en particulier).

1.3 Démarche générale, principales difficultés

La performance d'un algorithme de Reconnaissance de Visages dépend principalement du nombre de paramètres statistiquement indépendants extraits, et de leur variance.

- La variance doit être faible au sein d'une même classe
- La variance entre les classes doit être grande

Nota: La notion de classe ne désigne pas seulement un individu particulier. Suivant les paramètres extraits, on peut s'intéresser à la classe des hommes / femmes, des personnes qui sourient / ne sourient pas, etc.

On se heurte alors à trois principales source de variations au sein d'une même classe: l'éclairage, l'angle de vue ou la pose, et l'expression du visage. Généralement, ces trois paramètres engendrent des variations plus grandes que les variations entre classes distinctes, à paramètres constants ! Une grande partie des recherches actuelles (Eigenfaces, Fischerfaces) consistent à analyser l'influence de ses paramètres, pour s'en affranchir et réellement extraire les paramètres intrinsèques du visage.

1.4 Comment extraire les bons paramètres ?

Un visage consiste en la surface d'un solide partiellement déformable. Pour une même personne l'image projetée dépend principalement des paramètres suivants:

- Paramètres liés a la personne
  - Instantanés (peuvent être modifiés pour la prise de vue)
    - Expression
    - Accessoires (lunettes, chapeau)
  - Temporaires (ne peuvent être modifiés pour la prise de vue)
    - Âge
    - Capillarité
- Paramètres liés au dispositif d'acquisition
  - Instantanés
    - Éclairage
    - Pose
  - Temporaires
    - Paramètres de la caméra

Dans une première approche, on peut chercher à fixer les paramètres instantanés, en standardisant les modalités d'acquisition: expression neutre, sans lunettes, éclairage de face, etc. Une approche plus générale consiste à prendre tous ces paramètres en compte. Il en résultera un algorithme plus complexe, avec une phase d'apprentissage beaucoup plus importante (photos du même visage sous plusieurs éclairages, avec différentes expressions).

On doit ensuite choisir une représentation basée sur l'approche en 2D ou sur un modèle 3D. Il faut alors extraire les degrés de liberté optimaux, et distinguer:

- Les paramètres qui identifient un visage par rapport à un autre objet (problème de détection du visage dans l'image)

- Les paramètres qui identifient un visage particulier

De nombreuses possibilités existent enfin quant à la règle de décision à adopter qui discriminera les classes entres elles. Le plus souvent on aura recours à une méthode de Maximum a Posteriori (MAP estimate).

1.5 Un peu de philosophie

Comment expliquer qu'une opération aussi complexe que la Reconnaissance de Visages puisse être réalisée par la plupart des primates?

Des études ont montré que le cerveau humain comporte une zone spécialisée dans la Reconnaissance de Visages (le traumatisme appelé Prosopagnosie Faciale rend la personne incapable de reconnaître un visage, voire de distinguer un visage d'un objet, alors que les facultés de vision ne sont pas affectées). Chez les autres primates, cette faculté fait appels à plus de 30 zones distinctes du cerveau. Cette spécialisation du cerveau sur ce qui nous semble être un détail peut cependant s'expliquer; l'expression corporelle (corps, mains, visages) a toujours été au coeur des interactions sociales, et le cerveau semble s'être précisément développé pour pouvoir effectuer ces interactions. Ainsi, doter nos ordinateurs de telles facultés, c'est en quelque sorte les faire "évoluer", et peut-être les rendre plus proches des humains.

Source: John DAUGMAN, Face and Gesture Recognition: Overview, IEEE PAMI, vol.19, no.7, July 1997

Chapitre 2

ÉTAT DE L'ART

2.1 Reconnaissance à base de déformations de type "morphing"

L'approche consiste, étant données deux photos de visages, à définir une déformation de l'espace permettant de superposer entre elles ces deux images, puis a calculer l'énergie de cette déformation pour décider si il s'agit de la même personne. Ceci nécessite une extraction automatique de paramètres permettant de calculer la déformation, puis un calcul d'énergie de déformation. On utilise ici le modèle de plaque mince introduit par Bookstein en 1989. Pour traiter séparément les variations systématiques dues aux conditions d'acquisition des données (éclairage, orientation du visage, expressions faciales, coiffure) et les variations biométriques qui permettent d'identifier et de différencier les visages (distance entre les yeux, le nez et la bouche, etc..), on relaxe les conditions d'interpolation: l'énergie est alors la somme d'une énergie de déformation et d'une énergie représentant l'écart entre les points tranformés et les points de référence. On attribue à chacun des huit points caractéristiques un poids dépendant de sa variance spatiale moyenne au sein d'une même classe.

2.2 Reconnaissance à base d'ondelettes de Gabor

Cette méthode utilise les ondelettes de Gabor en deux dimensions :

$$\psi_{\overrightarrow{k}}\left(\overrightarrow{x}\right) = \frac{\overrightarrow{k}^2}{\sigma^2} \exp\left(-\frac{\overrightarrow{k}^2.\overrightarrow{x}^2}{2\sigma^2}\right)\left[\exp\left(-i\overrightarrow{k}.\overrightarrow{x}\right) - \exp\left(\sigma^{-2}\right)\right]$$

Les modèles sont constitués des visages des personnes connues, dont on a découpé la partie supérieure (les cheveux). On applique les ondelettes de Gabor au modèle et à la photo à reconnaître. Les basses fréquences servent à caler le visage du modèle sur celui de la photo car elles sont moins perturbées par les déformations locales. On obtient un vecteur de translation, qu'on applique à tous les points du modèle par une correspondance calculée sur les amplitudes des transformées.

La similarité entre l'image et le modèle est caractérisée par deux valeurs calculées sur un ensemble P de correspondances pour N couples ('mapping') :

$$P\left(M,I\right)=\left\{\left(\overrightarrow{x_i}^M,\overrightarrow{x_i}^I\right)|i=1,\ldots,N\right\}$$

- Le déplacement moyen

$$\overrightarrow{A}\left(P\right)=\frac{1}{N}\sum_{i=1}^{N}\left(\overrightarrow{x_i}^I-\overrightarrow{x_i}^M\right)$$

- La distorsion

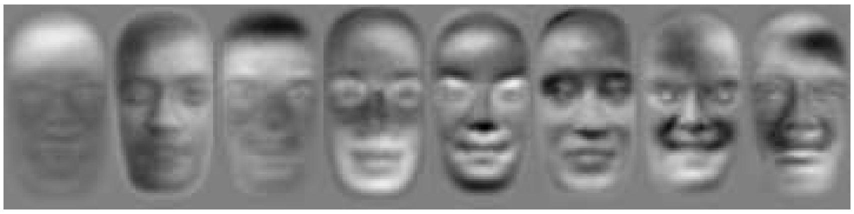
$$D_m\left(P\right)=\frac{1}{N-1}\sqrt{\sum_{i=1}^{N}\left(x_{mi}^I-x_{mi}^M-A_m\left(P\right)\right)^2},m\in\{1,2\}$$

On ajoute les fréquences une à une pour faire correspondre les points ajoutés, en découpant les matrices du modèle et de l'objet en sous matrices que l'on déplace. Si on arrive pas à recaler une sous matrice, on suppose qu'elle reste proche des sous matrices voisines. Ensuite, on vérifie que le changement de phase entre le modèle et l'objet est compatible avec la déformation trouvée, ce qui permet d'éliminer les correspondances trop faibles. Cette méthode est robuste aux translations, déformations, et aux changements du décor.

2.3 Reconnaissance à base de "eigenfaces"

Il s'agit d'un système développé au MEDIALAB du MIT (lauréat du concours FERET '96 organisé par l'US-ARMY). Le système comporte une opération de détection et cadrage en deux étapes, une étape de normalisation du contraste , et une étape d'extraction de caractéristiques basée sur une expansion de Karhunen-Loeve (eigenspace), dont la sortie est utilisée pour l'identification et le codage. Ceci mène à une représentation compacte du visage qui peut être utilisé pour des l'identification aussi bien que pour la compression d'images. Des images de visages de bonne qualité sont automatiquement produites en utilisant approximativement 100-bytes des données encodées. Le système a été avec succès testé sur une base de données de presque 2000 photographies faciales de la base de données d'ARPA FERET avec un taux de détection de 97%. Des taux de 99% ont pu êtres obtenus sur un sous-ensemble de la base comportant 2 vues frontales de 155 individus.

Voici un exemple d'eigenfaces :



2.4 Reconnaissance à base de modèles déformables

Cet article introduit une méthode unifiée d'analyse de visages robuste à l'éclairage, la pose, et l'expression faciale. Elle comporte deux étapes :

- Modélisation : on génère deux modèles déformables d'apparence faciale (forme et radiométrie). Cette opération n'est réalisée qu'une fois pour une base de visages.
- Interprétation : on utilise ces modèles pour coder et interpréter des images de visages. Cette opération est lancée chaque fois que l'on soumet un visage à l'algorithme.

2.4.1 Modélisation

A partir d'une base de données de visages, on établit un modèle de contours déformables (flexible shape model) permettant de localiser les principaux traits du visage. Ce contour est défini par un nuage de 152 points, sa position "au repos" correspond à la moyenne quadratique des visages de la base (les points caractéristiques de chaque visage de la base ont été saisis manuellement pendant la phase d'apprentissage). Une analyse en composantes principales permet de trouver les principaux modes de variations de ce contour. Dans cette étude, 16 paramètres suffisent pour déformer le contour moyen vers n'importe quel visage de la base (i.e. 16 paramètres suffisent pour reconstituer 95% des variations). Sur ces 16 paramètres seuls 6 sont discriminants (i.e. caractéristiques d'un individu), par exemple les trois premiers modes correspondent aux trois rotations spatiales du visage. L'extraction de ces variables discriminantes se fait grâce à l'introduction d'une distance de Mahalanobis dans l'espace des paramètres (analyse de discriminant).

Notons X le contour d'une image de la base et Xm le contour moyen. L'analyse en composantes principales consiste à chercher les valeurs propres de la matrice de covariance C donnée par:

$$C_{ij} = \frac{E\left[(X - X_m)_i (X - X_m)_j\right]}{\sigma\left[(X - X_m)_i\right] \sigma\left[(X - X_m)_j\right]}$$

Chaque vecteur propre est "responsable" d'une variance égale à sa valeur propre associée, soit une contribution:

$$\alpha_i = \frac{\lambda_i}{\sum \lambda_j}$$

On a ainsi décomposé les variations de X en variations indépendantes d'importance décroissante. On construit alors une matrice orthogonale P en prenant pour colonnes les premiers vecteurs propres de la décomposition précédente (permettant, par exemple, de reconstituer 95% de la variance). On peut alors écrire pour tout X :

$$X \simeq X_m + P.b$$

b étant le vecteur des paramètres du modèle. P étant orthogonale, on obtient donc les paramètres par la formule suivante :

$$b = P^t(X - X_m) \ \ avec\ P^{-1} = P^t$$

On cherche ensuite a établir un modèle radiométrique (= niveaux de gris) déformable (i.e. qui aura certains degrés de liberté). Pour cela, on cherche a normaliser la géométrie de chaque visage, pour en extraire le profil radiométrique, et faire ensuite une analyse en composante principales sur ces profils de manière similaire à l'opération faite sur les contours. Ainsi, on commence par ajuster le modèle de contours déformables sur chaque visage (opération détaillée dans la partie Interprétation), sur lequel on prélève quatorze points. On effectue ensuite un morphing pour superposer ces points sur ceux du visage moyen (en utilisant l'approche de type plaque mince introduite par Bookstein). On extrait alors le vecteur de niveau de gris normalisés (pour éliminer le paramètre de l'intensité globale de l'image, d'influence prépondérante) pour une zone correspondant au centre du visage. L'analyse en composantes principales montre que 76 paramètres sont nécessaires pour expliquer 95% des variations. Ici encore, on peut effectuer une analyse de discriminant pour séparer les modes de variations inter et intra-classes. Une autre approche, basée sur des profils radiométriques locaux, est également présentée dans l'article. Elle permet de rendre l'algorithme robuste à des occultations partielles du visage, et peut être utilisée en conjonction avec la méthode précédente.

2.4.2 Identification

Lorsqu'une nouvelle image est présentée au système, on commence par localiser les caractéristiques du visage grâce à une recherche multi-résolution de modèle de contour actif. On part du modèle de contours déformables initial, puis on minimise itérativement l'énergie du contour en libérant certains degrés de liberté. On commence par une transformation rigide, puis on fait varier les paramètres caractéristiques issus de l'étape de modélisation. On extrait ainsi les paramètres de forme. Ce type d'algorithme peut également être utilisé pour suivre les mouvements du visage en temps réel, les paramètres d'une image étant utilisés comme initialisation de la suivante. Ensuite on utilise ce contour pour déformer le visage vers la forme "canonique" et calculer le profil radiométrique. On extrait ainsi les paramètres radiométriques.

L'algorithme de classification utilise le vecteur constitué des paramètres de forme et de radiométrie pour associer ce visage à une classe de la base. On utilise une approche statistique de décision Bayésienne, en cherchant à maximiser la probabilité a posteriori, pour retrouver à qui appartient le visage. La classification peut également porter sur d'autres critères, comme la pose, l'expression du visage ou le sexe.

2.5 Sites Internet

- MIT Media Laboratory Vision and Modeling Group Face Recognition Project Page A.Pentland
  http://vismod.www.media.mit.edu/vismod/demos/facerec/index.html

- MIT AI Lab Center for Computational and Biological Learning Example-based Image Analysis and Synthesis for Faces Tony Ezzat and Tomaso Poggio
  http://cuneus.ai.mit.edu:8000/research/ebias/ebias.html

- Et la page de référence : Face Recognition Home page
  http://www.cs.rug.nl/~peterkr/FACE/face.html

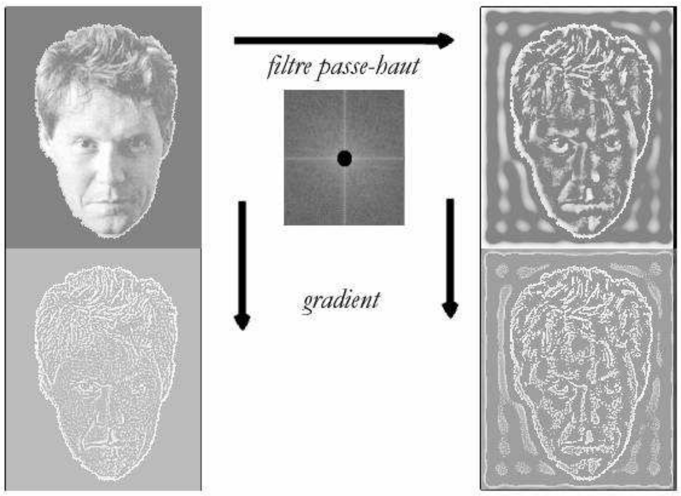
Chapitre 3

Méthode utilisée

3.1 Première approche : recalage rigide

Nous avons décidé de partir d'un modèle de visage auquel on applique une transformation rigide (ici une translation 2D) pour qu'il s'adapte au visage considéré.

Cette méthode est robuste aux rotations et aux déformations, on va donc chercher à comparer différents visages en se limitant à la représentation de la zone recalée. Mais les contours d'un visage varient beaucoup suivant l'éclairage. Différentes méthodes ont été envisagées pour éviter ce problème : imposer une certaine symétrie au modèle du visage, ce qui intervient dans la paramétrisation des contours, une décomposition en composantes principales des différents éclairages possibles ("eigenfaces") pour un visage, mais la base d'apprentissage nécessaire est plus importante. Un prétraitement intéressant est un filtrage passe-haut qui permet de minimiser l'effet de grosses différences d'éclairage :

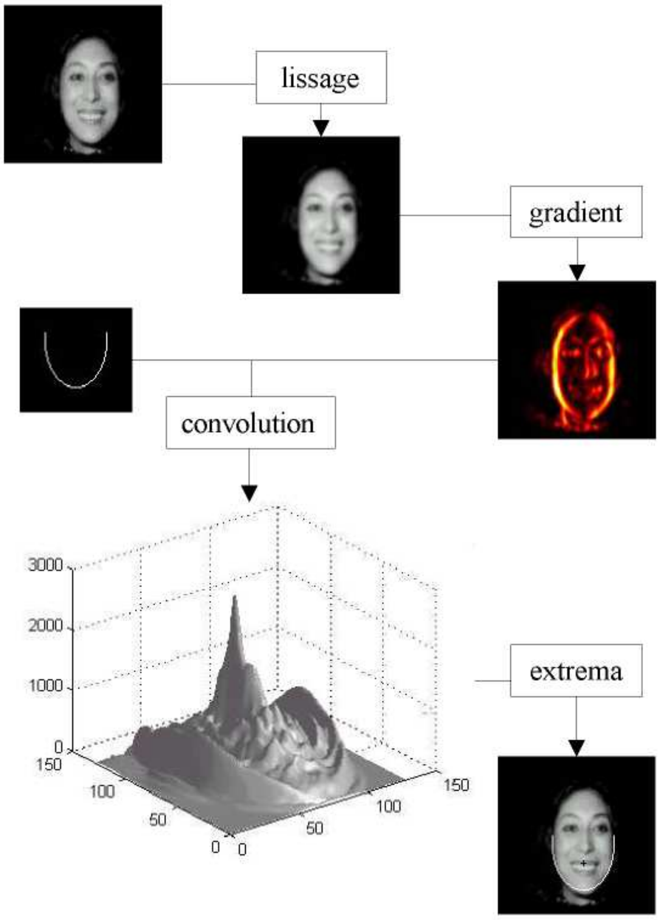


3.1.1 Mise en place technique

Pour ne pas passer trop de temps à coder des fonctions de base, nous travaillons sous Matlab, qui comporte toutes les fonctions mathématiques et graphiques nécessaires. Pour tester notre programme, de nombreuses bases de données de visages sont disponibles sur Internet. Nous en avons choisis deux. La première (Yale database) comporte des visages de même taille et bien éclairés sur fond sombre. La deuxième (Ferret database) comprend des visages de tailles différentes sur fond blanc.

La première étape consiste à localiser le visage dans l'image. En première approche pour cette étape, nous n'avons recalé que la partie du modèle constituée du contour du visage. Pour créer ce contour, nous avons recalé sur une des images une ellipse, que nous avons tronquée pour éviter les problèmes causés par des coiffures différentes, des chapeaux...

Ensuite, nous avons calculé la corrélation entre cette ellipse et les images de la première base de données. Ce calcul présente un maximum très net lorsque l'ellipse est sur le contour du visage.

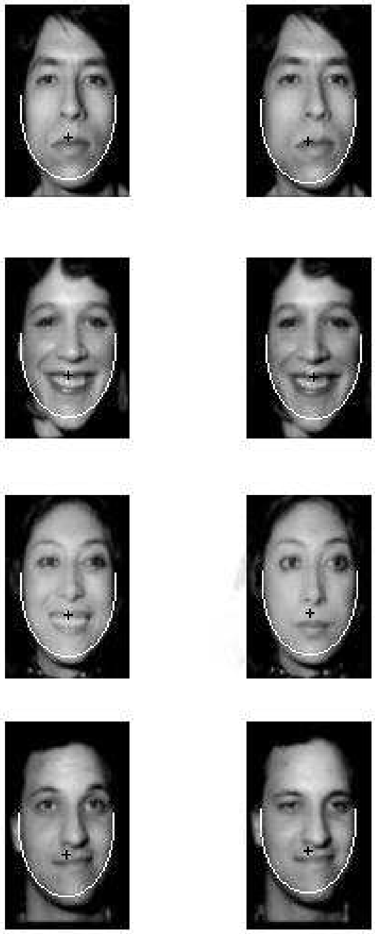


Donc pour une taille donnée de visages, il est facile de localiser le visage dans l'image. Mais sur la deuxième base, les résultats sont beaucoup moins bons. Car l'échelle ne correspond plus.

Il faut donc rajouter des paramètres. L'ellipse doit comporter un facteur d'échelle. Et la méthode envisagée est de faire une descente de gradient sur les paramètres de position et de taille. Ceci permet d'éviter de calculer la corrélation avec tous les points de l'image, et économise donc du temps de calcul. La position de départ peut être une suite de positions aléatoires ou le centre.

3.1.2 Résultats

Le temps de calcul, sur un Pentium 120, est d'environ 5 secondes par visage. Le modèle d'ellipse tronquée choisi s'adapte indiféremment sur des visages masculins ou féminins, de formes différentes. Voici les recalages obtenus :



3.2 Deuxième approche : recalage non rigide

3.2.1 Choix techniques

Vouloir identifier les visages directement grâce à un masque déformable conduit à des algorithmes complexes, coûteux en calcul et dont la robustesse repose sur une étape d'apprentissage contraignante . Nous n'avons donc pas cherché à implémenter cette méthode, étant donné nos contraintes de temps et de moyens. Par contre, on peut se servir de cette approche pour localiser les principaux traits du visage afin de les normaliser avant l'étape de reconnaissance. Dans cette optique, il est inutile d'utiliser un masque détaillé.

Pour notre étude, nous avons utilisé une ellipse, pour les raisons suivantes:

- Les résultats obtenus pour le racalage rigide d'une ellipse tronquée sur la base Yale montrent une bonne robustesse du recalage vis à vis de la forme et des expressions du visage, et des variations de pose.

- En testant ce premier algorithme sur la base Ferret, nous avons constaté qu'il fallait autoriser une variation d'échelle, et qu'une ellipse complète donnait une meilleure robustesse. Nous avons donc retenu une ellipse possédant deux paramètres de position et un paramètre d'échellle. Ce recalage désormais non-rigide ne peut plus se faire par recherche exhaustive car cela serait trop coûteux. Il a donc fallu mettre en place un algorithme d'optimisation que nous allons présenter dans le paragraphe suivant.

Enfin, on se servira du recalage pour extraire et normaliser la zone centrale du visage, sur laquelle va travailler l'algorithme de classification. L'étape de reconnaissance peut faire appel à de nombreuses techniques : analyse en composantes principales, perceptrons, ondelettes. Dans le cadre du séminaire, nous avons mis en place un algorithme simple de classification basé sur un écart quadratique entre matrices.

3.2.2 Le problème d'optimisation

Le problème discret

I est l'image de départ, c'est une matrice $m \times n$ à valeurs dans [0 255].

L'image lissée A est obtenue par convolution discrète de I avec un noyau gaussien N.

$$a_{ij} = \sum_{k,l \in [-p,p]^2} n_{kl} I_{i-k,j-l}$$

G est la matrice donnant la norme du gradient de l'image lissée.

$$g_{ij} = \sqrt{(g_{ij}^x)^2 + (g_{ij}^y)^2}$$

$$g_{ij}^x = a_{i+1,j} - a_{ij}$$

On note $u = [\ u_1 \quad u_2 \quad u_3\ ]$ le vecteur des paramètres de recalage:

- u_1, u_2 donnent la position du centre de l'ellipse
- u_3 est la facteur d'homothétie par rapport à une ellipse de référence.

Le recalage optimal est celui qui maximise la moyenne -J du gradient G sur le contour de l'ellipse, c'est à dire qui minimise le critète $J(u)$. On réalise cette intégration curviligne discrète par le produit terme à terme d'une sous-matrice de G centrée en (u_1, u_2) et du masque de taille u_3. Ce masque est du type:

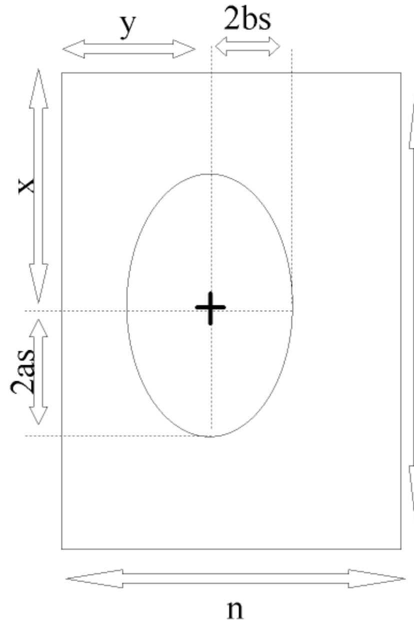
$$\frac{1}{8}\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Dans notre implémentation le masque M initial mesure 71*55 pixels. Le masque $M(u_3)$ est obtenu en redimensionnant M par la méthode des plus proche voisins, et en renormalisant le résultat pour que la somme de ses éléments soit 1.

Il s'agit d'un problème d'optimisation sous contraintes, car l'ellipse doit restée cadrée dans l'image initiale I.

Pour simplifier les expressions, on ne notera pas le fait qu'il faut prendre la partie entière de certaines grandeurs pour conserver des indices entiers.

Notons 2a, 2b la taille du masque initial. Pour un u donné, l'ellipse est placée sur l'image suivant ce schéma :



Le problème d'optimisation à résoudre peut alors s'écrire sous la forme suivante:

$$\begin{cases} \min(J(u)) = -\sum_{i,j\in W} g_{ij} m_{i-\frac{u_1}{2},j-\frac{u_2}{2}}(u_3) \\ u \in U \begin{cases} au_3 < u_1 < m - au_3 \\ bu_3 < u_2 < n - bu_3 \\ 2au_3 < m \\ 2bu_3 < n \end{cases} \end{cases}$$

Le problème continu: parallèle avec les "snakes"

On peut chercher à formuler ce problème sous forme continue, en considérant que l'image est une surface $I(x, y)$ définie sur un compact [0..m, 0..n].

L'image lissée est obtenue par convolution de I avec une gaussienne d'écart type σ.

Le gradient de l'image est calculé par une méthode de dérivation classique.

L'ellipse déformable est un contour fermé défini par l'équation quadratique:

$$\Gamma(u) : \quad \frac{(x-u_1)^2}{a^2} + \frac{(y-u_2)^2}{b^2} = u_3^2$$

Les paramètres du recalage sont définis par le vecteur $u = [\ u_1 \quad u_2 \quad u_3\ ]$.

Alors, la fonctionnelle à minimiser peut s'écrire sous la forme d'une intégrale curviligne:

$$J(u) = -\oint_{\Gamma(u)} G(x, y)ds$$

Les contraintes peuvent être écrites sous une forme similaire au cas discret.

On constate la similarité de ce problème d'optimisation avec l'approche de type "snakes". En effet le terme intégral peut être interpreté comme l'énergie externe du contour dans le champ de potentiel défini par G. Par contre, le terme d'énergie interne semble absent. La régularité de la courbe ne peut pas varier ici, puisqu'elle ne subit que des transformations affines. En fait, les contraintes imposées à u pourraient être assimilées à une énergie interne si, au lieu de fixer des frontières planes, on associe une énergie de type élastique qui augmente lorsque l'on rétrécit ou agrandit l'ellipse, et lorsqu'on l'éloigne du centre de l'image. Le problème se ramènerait au problème de minimisation de l'énergie totale de la courbe Γ:

$$E_{tot} = E_{interne} + E_{externe}$$

La stratégie adoptée: une exploration active de l'espace des paramètres

On utilise une méthode d'optimisation itérative sur 3 paramètres de type descente de gradient. Le problème vient de la grande irrégularité du champ de potentiel dans lequel on se déplace. En effet, pour un facteur d'échelle fixé, la surface de potentiel obtenue en faisant varier le vecteur translation est très irrégulière. L'allure de cette surface met à jour les difficultés qui vont se poser lorsque l'on implémentera une descente de gradient classique. La surface possède bien un maxima global unique mais :

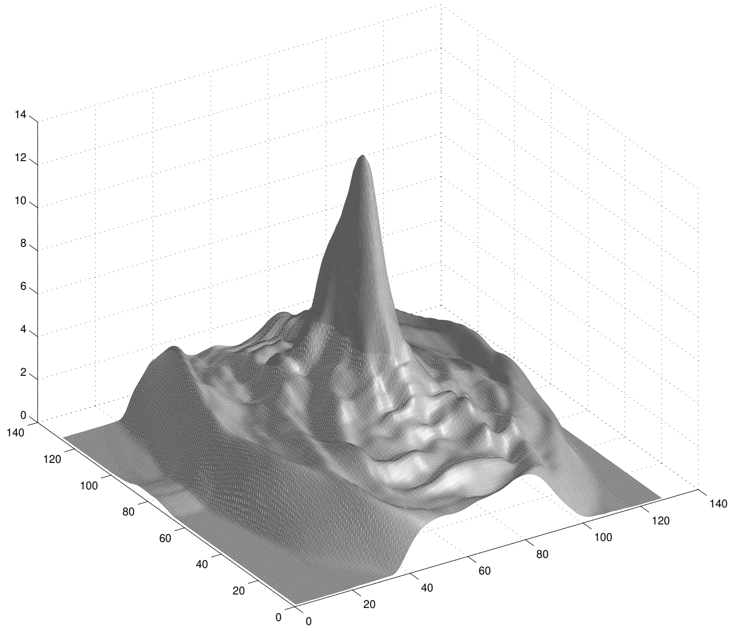
- Il existe de nombreux extrêmas locaux.

– La surface n'est pas convexe.

– Le pic correspondant à l'extrêmum global est très étroit, ce qui rend difficile la convergence de l'algorithme.

– Ce pic n'est pas régulier, la surface ne semble pas dérivable localement, ce qui menace l'objectif d'un algorithme d'optimisation classique, qui consiste à chercher le point où le gradient est nul.

Variation du potentiel par translation de l'ellipse sur le gradient de l'image :



Nous avons testé plusieurs méthodes de type descente de gradient, par ordre de complexité croissante. Il s'agit de méthodes itératives:

$$\begin{cases} \textit{initialisation } u_0 \\ u_{n+1} = u_n - \rho_n w_n \end{cases}$$

ρ_n est le pas de l'optimisation, w_n est la direction de descente. Le tableau suivant présente quelques algorithmes classiques d'optimisation sans contraintes :

Méthode	ρ_n	w_n	*Remarques*
Pas constants	$\rho_n = constante$	$w_n = grad(J(u_n))$	sous les hypothèses sur J, convergence pour ρ "suffisamment" petit
Pas variables	$\rho_n = argmin(J(u_{n-1} - \rho_n w_n))$	$w_n = grad(J(u_n))$	sous les hypothèses sur J, convergence
Newton	$\rho_n = 1$	$w_n = (H(u_n))^{-1} \star grad(J(u_n))$	$H(u_n)$ est le Hessien de J en u_n
Gradient conjugué	$\rho_n = argmin(J(u_{n-1} - \rho_n w_n))$	$w_n = grad(J(u_n)) - \lambda_{n-1} w_{n-1}$	$\lambda_{n-1} = \frac{\Vert grad(J(u_n))\Vert^2}{\Vert grad(J(u_{n-1}))\Vert^2}$ (Fletcher-Reeves)

Pour tenir compte des contraintes sur u, nous avons utilisé une méthode de projection : lorsque u_{n+1} ne vérifie pas les contraintes de domaine, on le remplace par sa projection dans U. Par exemple, si le centre de l'ellipse est situé trop à droite, faisant dépasser le masque du cadre, on ajuste sa position de manière à ce que l'ellipse soit tangente au bord droit de l'image. D'autre part, nous avons du rajouter une contrainte sur la taille minimale de l'ellipse, apres avoir constaté que dans certains cas de figures, l'ellipse diminuait jusqu'à devenir un point situé sur le pic du gradient, ce qui minimise effectivement J…

Pour les raisons citées ci-dessus, rien ne permet d'assurer que ces algorithmes vont converger dans le cas de notre problème d'optimisation. En effet, nous avons implémenté la méthode à pas constant et la méthode à pas variables, mais les résultats n'étaient pas satisfaisants.

Pour la méthode à pas constants, le choix du pas rend l'algorithme peu robuste. Pour un pas petit, l'algorithme converge mais l'ellipse ne fait que des déplacements limités et reste dans un minimum local. Pour un pas trop grand, l'algorithme devient instable.

Pour la méthode à pas variables, l'algorithme s'est révélé instable. Le gradient de J, calculé localement ne représente pas bien la forme globale du potentiel, qui n'est pas convexe. Cette méthode serait bien adaptée à un critère quadratique.

Pour ces deux méthodes, il s'est révélé très difficile de fixer un critère d'arrêt à cause de leur manque de stabilité. Nous avons donc cherché une méthode facilement paramétrable et robuste permettant de trouver l'optimum en un temps déterminé. L'idée a été de fixer le nombre d'itérations, et d'adopter une stratégie "d'exploration active" de l'espace des paramètres. L'objectif est de "visiter", le plus rapidement possible, les extremas de la fonction à minimiser que nous appellerons "candidats", pour retenir à la fin le meilleur des candidats. Le schéma itératif utilisé est le suivant:

$$\begin{cases} \rho_n = \frac{K}{\varepsilon + ||grad(J(u_n))||} \\ w_n = \frac{grad(J(u_n))}{\left|\left|grad(J(u_{n))}\right|\right|} \end{cases}$$

Le déplacement se fait en suivant la direction du gradient. Lorsque la norme du gradient est faible, le pas est grand, on va donc rapidement quitter une zone uniforme. Lorsque l'on approche d'un extrema, la norme du gradient augmente, donc le pas diminue : on "ralentit" à l'approche d'une solution. Une fois sur le sommet, le gradient va être faible, donc le pas important: on part "explorer" une autre zone. Au cours de cette exploration, une variable stocke la meilleure solution visitée. Au bout de n itérations, cette variable doit nous donner le minimum global. Avec cette démarche, l'algorithme s'est montré capable :

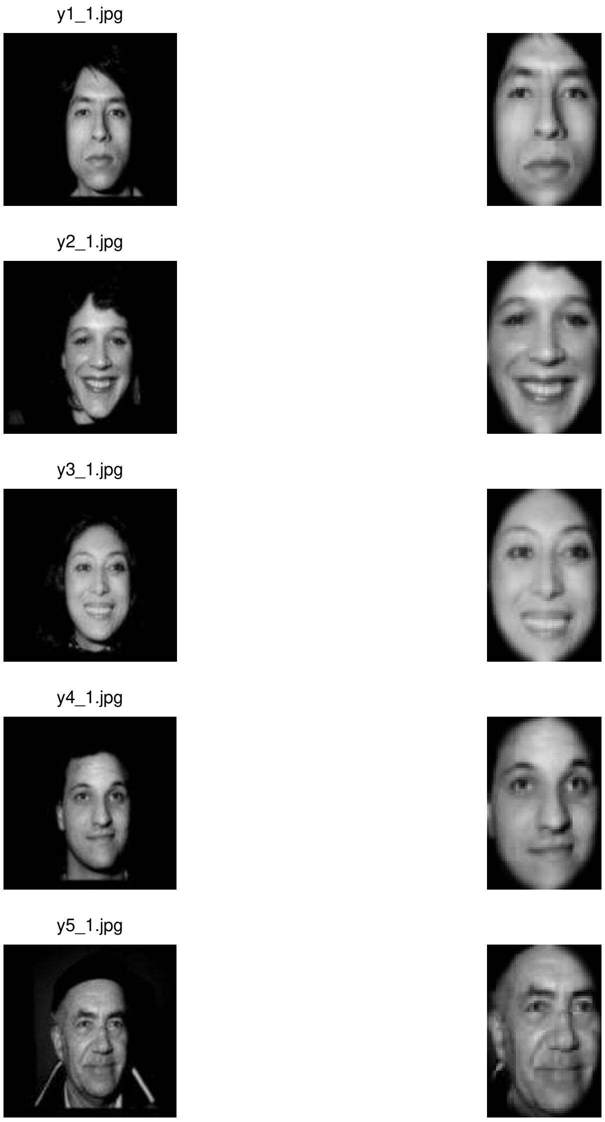
1. de sortir la plupart des extremas locaux
2. de "visiter" l'optimum global avec une bonne probabilité
3. d'assurer une bonne localisation de cet optimum grâce au "ralentissement" à l'approche d'une solution

Pour augmenter la robustesse de l'algorithme, nous avons utilisé trois initialisations différentes pour le recalage de l'ellipse, à partir desquelles on a effectué 20 itérations de l'algorithme d'exploration. On retient la meilleure des trois solutions.

3.2.3 Résultats

Yale Database

La localisation du visage est réalisée par l'algorithme décrit précedemment. Puis le visage est déformé pour que l'image soit de 77*51 pixels. Voici le résultat des recalages puis déformations sur la base de Yale :



Sur cette base, les visages se détachant bien du fond noir, le recalage est donc bon.

Ferret Database

f1_1.jpg

f2_1.jpg

f3_1.jpg

f4_1.jpg

f5_1.jpg

Les visages se détachant moins, l'influence des cols et des coiffures est plus importante. Les resultats sont donc moins précis. De plus la taille des visages varie de façon importante. Le problème de cette base vient du fait que les plus fortes variations de gradient des photos ne sont pas sur les contours du visage. Pour renforcer l'algorithme et eviter un mauvais recalage, on peut aussi comparer localement l'angle du gradient à l'angle de l'ellipse (par rapport à un axe horizontal par exemple) et pénaliser les endroits ou l'angle est diférent.

3.3 Utilisation des paramètres de recalage pour la reconnaissance

Une fois que tous les visages ont été normalisés en taille, c'est sous cette forme que la discrimination va être faite. Pour minimiser l'effet de fortes différences de luminosité, les visages sont normalisés en intensité en leur retranchant leur valeur moyenne et en les divisant par leur écart type. Puis on extrait de ce visage une partie correspondant au centre du visage (décentrée vers le bas pour eviter la coiffure). Enfin on balaye tous les visages de la base avec cette sous-matrice en calculant un écart quadratique entre cette sous-matrice et la partie correspondante du visage considéré. Le visage correspondant à l'écart quadratique le plus faible est retenu comme étant le visage reconnu. a et b étant les deux sous-matrices, le critère est défini par :

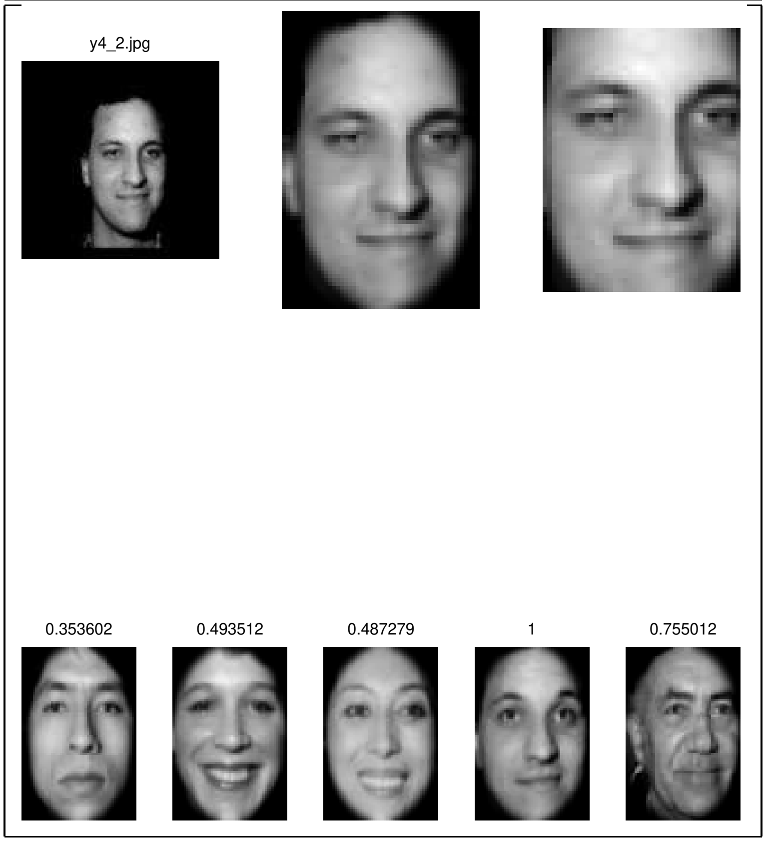
$$\theta = \frac{1}{\varepsilon + \sum (a_{ij} - b_{ij})^2} \quad avec \;\; \varepsilon = 10^{-6}$$

La constante évite de diviser par zéro si la photo d'entrée est dans la base d'apprentissage. Une marge d'erreur doit être fixée si deux visages ont un résultat proche.

Yale Database

Voici les résultats obtenus en comparant des visages de la base de Yale. Il y a 5 personnes (A,B,C,D,E) et deux visages par personne. L'apprentissage se fait en mémorisant le recalage effectué sur le premier visage de chaque personne. La reconnaissance se fait en entrant un visage, qui sera comparé aux 5 mémorisés. On divise tous les résultats par le meilleur, qui obtient 1. Voici deux exemples de reconnaissance et un tableau récapitulatif.


y2_2.jpg
0.771296
1
0.958722
0.956005
0.51682

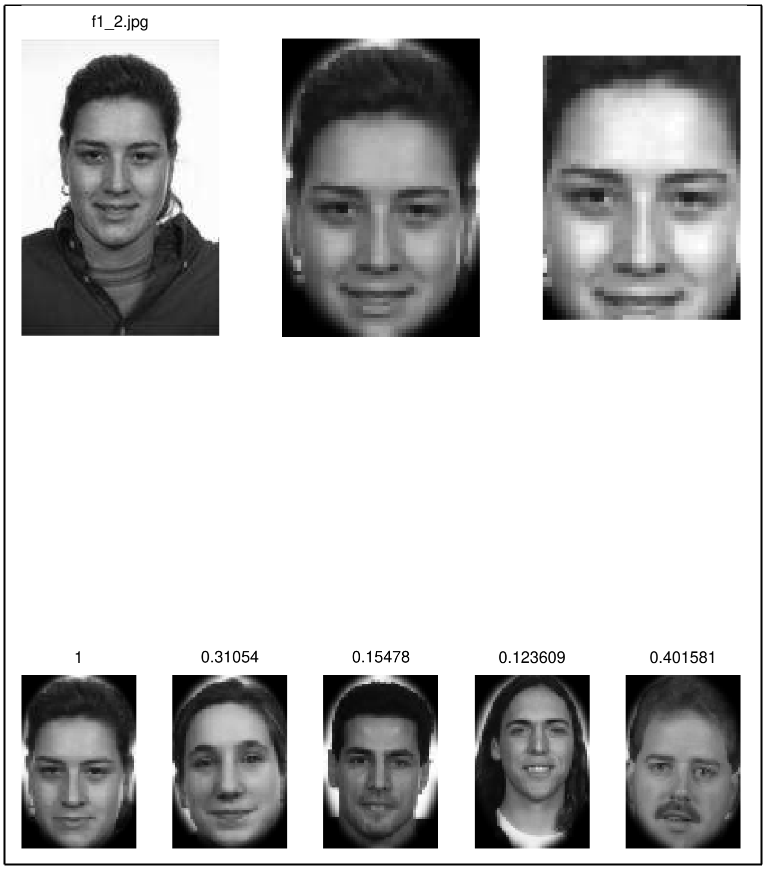

y4_2.jpg
0.353602
0.493512
0.487279
1
0.755012

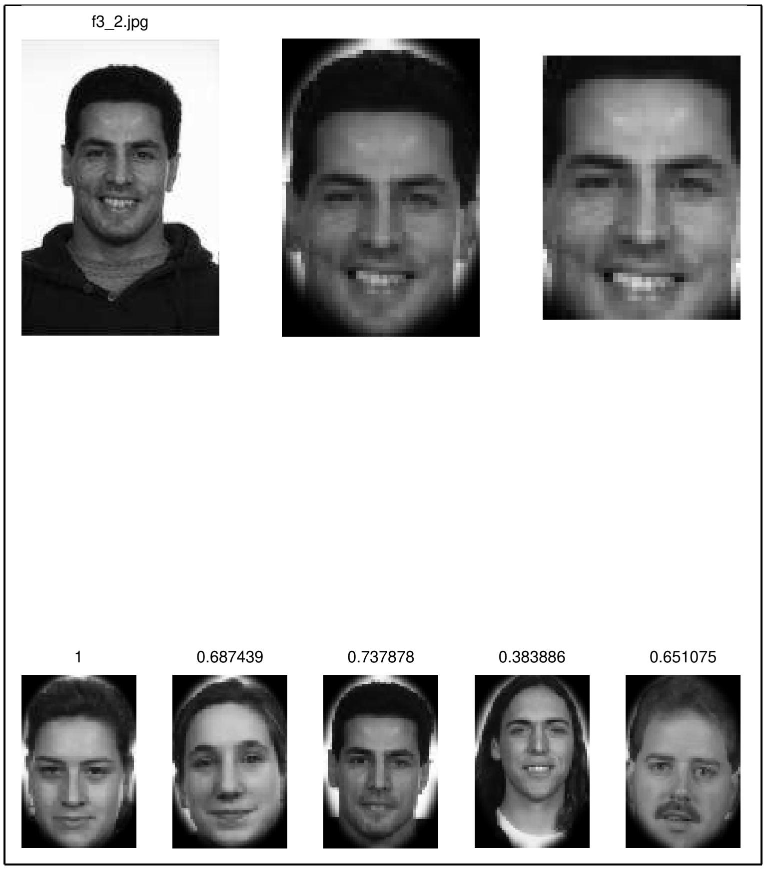
	Critère de corrélation θ				
Visage d'entrée	A 1	B 1	C 1	D 1	E 1
A 1	1	1.5e-9	2.4e-9	1.6e-9	9.5e-10
B 1	1.5e-9	1	2.7e-9	1.5e-9	9.6e-10
A 2	1	0.14	0.23	0.15	0.096
B 2	0.77	1	0.96	0.95	0.51
C 2	0.55	0.40	1	0.45	0.37
D 2	0.35	0.49	0.49	1	0.75
E 2	0.53	0.65	0.75	0.97	1

Lorsque le visage d'entrée appartient à la base d'apprentissage, la différence est très nette. Sinon, la marge est parfois très faible.

Ferret Database

Voici les résultats sur la base de Ferret :





	Critère de corrélation θ				
Visage d'entrée	F 1	G 1	H 1	I 1	J 1
F 1	1	1.5e-9	7.9e-10	6.2e-10	1.5e-9
G 1	1.2e-9	1	6.7e-10	5.0e-10	9.9e-10
F 2	1	0.31	0.15	0.12	0.40
G 2	0.21	1	0.15	0.10	0.17
H 2	1	0.69	0.74	0.38	0.65
I 2	0.19	0.19	0.16	1	0.18
J 2	0.78	0.62	0.42	0.30	1

Comme les recalages étaient beaucoup plus variés car moins précis, la reconnaissance donne des marges extrêmes. Souvent très importantes mais aussi très faibles (il y a même un erreur de reconnaissance).

Apparemment une explication pourrait être l'apparition des dents. C'est une zone blanche donc qui comporte de valeurs importantes (proches de 255). Elle provoque donc des frtes variations de l'ecart quadratique. Dans les deux bases, ce sont ces photos qui posent des problemes. Une solution serait de rajouter dans la base d'apprentissage des photos où les dents apparaissent.

Chapitre 4

Programmes réalisés

4.1 Fonctions programmées sous Matlab

4.1.1 Extraction du visage

fdetect (face,affiche)

Fonction principale de l'extraction de visage. On réalise le recalage par une descente de gradient sur une fonctionnelle F qui à un emplacement et une échelle donnés de l'ellipse associe l'energie de l'ellipse considérée. Après le nombre d'itérations choisi, on prend les paramètres ayant donné un résultat optimal pour extraire le visage de la photo. Puis on le met à une taille déterminée (77*51 pixels).

face : visage entré (format JPEG)

affiche : décide de l'affichage des étapes intermédiaires (0 ou 1)

potentiel (rec)

Calcule l'energie de l'ellipse à partir d'un emplacement et d'une échelle.

rec : paramètres du recalage ([x y échelle])

fgrad (rec)

Calcule le gradient de F pour un recalage (rec) donné.

optim (recinit,nbiter,affiche)

Effectue la descente de gradient, avec un pas adaptatif (inverse de la norme du gradient). Test la validité du resultat (l'ellipse doit rester dans l'image).

recinit : recalage de départ

nbiter : nombre d'itérations de l'algorithme de recalage

4.1.2 Reconnaissance du visage

apprent_y

Applique la fonction *fdetect.m* à tous les premiers visages de chaque personne de la base de Yale et enregistre le résultat dans une matrice de dimension 3.

correl_y (face)

Calcule la fonction de corrélation choisie entre le visage entré (face) et la base de Yale. On ne considére que la partie centrale du visage entré pour pouvoir essayer plusieurs recalages.

4.1.3 Paramètres à fixer

- le nombre la taille et l'emplacement des initialisations. Une initialisation en bas de l'image et en haut de l'image ont été choisies pour éviter d'être bloqué par les extrema des cheveux et du col.
- le nombre d'itérations pour chaque initialisation. Comme il y a plusieurs initialisations et que le pas est adaptatif, le nombre d'itérations n'est pas très élevé (entre 20 et 40)
- le pas de la descente de gradient
- l'importance du flou appliqué à l'image (écart-type du noyau gaussien avec lequel on va convoluer l'image).

4.2 Code Matlab de ces fonctions

fdetect.m

```
%Cette fonction localise un visage sur une photo d'identité, grace
%à une ellipse (masque) déformable.
%Elle renvoie une matrice image normalisée (moyenne 128, écart-type
%20, de dimensions (XX,XX) contenant le milieu du visage.
%
%Syntaxe:
%J=fdetect('nom_image_format_jpg' ou Matrice_image,option_affichage)
```

```
%
%Option_affichage=0: affichage du résultat seul
%Option_affichage=1: affichage des résultats intermédiaires
%
%Bases de test:
%Yale database   'y1_1', 'y1_2', 'y2_1', ...., 'y5_2'
%Ferret database 'f1_1', 'f1_2', 'f2_1', ...., 'f5_2'

function out=fdetect(face,affiche);
global G; %matrice de gradient
global M; %ellipse initiale (masque)
global N; %ellipse déformée
global A; %image initiale
global echbornes %bornes du facteur d'échelle

%Paramètres internes
nbiter=20; %Nombre d'itérations pour la descente de gradient

%Lecture d'une image jpg
if ischar(face)
  A=imread(face,'jpg');
  A=rgb2gray(A);
else
  A=face;
end
[Al Ac]=size(A);

if affiche
  figure; imshow(A);
end

%Lissage
H=fspecial('gaussian',10,10);
As=conv2(A,H,'same');

if affiche
  figure; imshow(As, [0 256]);
```

```
end

%Calcul du gradient de l'image lissée
[Gx Gy]=gradient(As);
G=sqrt(Gx.^2+Gy.^2);

%Affichage du potentiel en 3D
%figure; colormap(gray); surfl(-G), shading interp, rotate3d on;

%Chargement du masque initial
load masque2.mat;
load fenetre.mat;

%Initialisation du masque déformable
N=M;
[Nl, Nc]=size(N);

%Calcul des bornes du paramètres d'echelle
echbornes=[(Al/Nl)*0.5 (Al/Nl)*0.9];

xmid=floor(Al/2);
ymid=floor(Ac/2);
xbottom=Al-round(mean(echbornes)*Nl/2)-5;
xup=round(mean(echbornes)*Nl/2)+5;

paraminit=[[xmid ymid echbornes(2)];[xup ymid mean(echbornes)];[xbottom ...
    ymid mean(echbornes)]];
%Initialisation des paramètres de recalage
rec=zeros(1,3); %Position du centre: rec(1),rec(2), Facteur d'échelle: rec(3)
potmax=0;
for k=1:3
  %Recherche des paramètres de recalage: fonction optim
  [rectemp,pot]=optim(paraminit(k,:),nbiter,affiche);
  if pot>potmax
    potmax=pot;
    rec=rectemp;
  end
```

```
end


%Affichage du recalage obtenu
N=imresize(M,rec(3));
N=N/sum(sum(N));
[Nl, Nc]=size(N);
Am=A;
Am((rec(1)-2):(rec(1)+2),rec(2))=0;
Am(rec(1), (rec(2)-2):(rec(2)+2))=0;
Am=double(Am);
xcorner=rec(1)-floor(Nl/2);
ycorner=rec(2)-floor(Nc/2);
Am((xcorner):(xcorner+Nl-1),(ycorner):(ycorner+Nc-1))=Am((xcorner): ...
  (xcorner+Nl-1),(ycorner):(ycorner+Nc-1))+256*(N~=0);
if affiche
  figure; imshow(Am, [0 256]);
end

%Découpage de la partie centrale du visage
out=double(A(xcorner:(xcorner+Nl-1),ycorner:(ycorner+Nc-1)));
out=imresize(out,[77 51]);
out=out.*F;
if affiche
  figure; imshow(out, [0 256]);
end

%imwrite(uint8(out),gray,[face 'rec.bmp'],'bmp');
```

potentiel.m

```
function x=potentiel(rec);
global G;
global M;
global N;

O=M;
```

```
if rec(3)~=1
   O=imresize(M,rec(3));
   O=O/sum(sum(O));
end
[Ol, Oc]=size(O);
xcorner=rec(1)-floor(Ol/2);
ycorner=rec(2)-floor(Oc/2);
x=sum(sum(O.*G((xcorner):(xcorner+Ol-1),(ycorner):(ycorner+Oc-1))));
```

fgrad.m

```
%Compute the gradient of the potentiel
function grad=fgrad(rec);

grad=zeros(1,3);
grad(1)=(potentiel([(rec(1)+1) rec(2) rec(3)])- ...
 potentiel([(rec(1)-1) rec(2) rec(3)]))/2;
grad(2)=(potentiel([rec(1) (rec(2)+1) rec(3)])- ...
 potentiel([rec(1) (rec(2)-1) rec(3)]))/2;
grad(3)=(potentiel([rec(1) rec(2) (rec(3)+.1)])- ...
 potentiel([rec(1) rec(2) (rec(3)-.1)]))*5;
```

optim.m

```
function [recmin,potmax]=optim(recinit,nbiter,affiche);

global N;
global M;
global A;
global G;
global echbornes;

%Paramètres
pas=[1 .3];


[Al Ac]=size(A);
rec=recinit;
```

```
graphpot=zeros(1,nbiter);
potmax=0;
%pasprec=[0 0 0];
%recprec=recinit;

if affiche
  figure
end

for i=1:nbiter
   grad=fgrad(rec);
   rec(1:2)=rec(1:2)+round(pas(1)*grad(1:2)/(norm(grad(1:2))+.000001)^2);
   rec(3)=max([min([rec(3)+pas(2)/(grad(3)+.000001) echbornes(2)]) echbornes(1)]);

   %  pasprec=round((rec-recprec)/2); %à utiliser pour lisser la
                                      %descente de potentiel
   %  recprec=rec;

   N=imresize(M,rec(3));
   N=N/sum(sum(N));
   [Nl, Nc]=size(N);
   xcorner=rec(1)-floor(Nl/2);
   ycorner=rec(2)-floor(Nc/2);

   if xcorner<10
     rec(1)=floor(Nl/2)+8;
     xcorner=rec(1)-floor(Nl/2);
   elseif xcorner>(Al-Nl-10)
     rec(1)=Al-floor(Nl/2)-8;
     xcorner=rec(1)-floor(Nl/2);
   end

   if ycorner<10
     rec(2)=floor(Nc/2)+8;
     ycorner=rec(2)-floor(Nc/2);
   elseif ycorner>(Ac-Nc-10)
     rec(2)=Ac-floor(Nc/2)-8;
```

```
    ycorner=rec(2)-floor(Nc/2);
  end

  graphpot(i)=potentiel(rec);

  if graphpot(i)>potmax
    recmin=rec;
    potmax=graphpot(i);
  end

  if affiche
    gradmax=max(max(G));
    Am=G;
    Am((rec(1)-2):(rec(1)+2),rec(2))=0;
    Am(rec(1), (rec(2)-2):(rec(2)+2))=0;
    Am((xcorner):(xcorner+Nl-1),(ycorner):(ycorner+Nc-1))=Am((xcorner): ...
(xcorner+Nl-1),(ycorner):(ycorner+Nc-1))+gradmax*(N~=0);
    imshow(Am,[0 gradmax]); title(i);drawnow;

  end
end

if affiche
  figure; plot(graphpot);
end
```

apprent_y.m

```
%Crée une base contenant 5 images de la base Yale (5 personnes)
basey=zeros(77,51,5);
for i=1:5,
  basey(:,:,i)=fdetect(['y' char(48+i) '_' char(49)],0);
end
save basey basey;
```

correl_y.m

```
%Effectue la corrélation entre une photo d'entrée et les 5 personnes
```

```
%de la base Yale

function correl_y(face);drawnow;
correl=zeros(1,5);
load basey;
figure;
subplot(2,3,1);imshow([face '.jpg']);
out=fdetect(face,0);
subplot(2,3,2);imshow(out,[0 255]);
out=(out-mean2(out))/std2(out);
a=out(15:70,5:46);
subplot(2,3,3);imshow(a,[min(min(a)) max(max(a))]);

for i=1:5
  b=(basey(:,:,i)-mean2(basey(:,:,i)))/std2(basey(:,:,i));
  for j=1:22
    for k=1:10
      c=b(j:j+55,k:k+41);
      m=1/(0.000001+sum(sum((c-a).^2)));
      correl(i)=max(correl(i),m);
    end
  end
end

for i=1:5
  subplot(2,5,i+5);imshow(basey(:,:,i),[0 255]);
  title(correl(i)/max(correl));drawnow;
end
```

Conclusion

Cette étude nous a permis d'aborder des domaines variés qui dépassent le cadre du traitement d'images ainsi la detection d'un visage dans une image s'est rammené à un problème d'optimisation dans un espace de dimension 3. La fonctionnelle à minimiser n'ayant aucune propriété de régularité a priori, cette optimisation a du se faire se faire sous forme d'une exploration active de l'espace des paramètres. Cet algorithme de detection s'est révélé suffisant pour extraire tous les visages des deux bases dont nous disposions. La précision de cette extraction a permis de reconnaître 90% des visages à l'aide d'un critère d'écart quadratique. Nous nous étions imposé la contrainte de n'avoir qu'un seul visage par personne dans la base d'apprentissage. Pour améliorer cet algorithme, c'est au niveau de la discrimination qu'il faut implémenter une méthode plus robuste, mettant en jeu une véritable étape d'apprentissage.

Bibliographie

[Ler96] B. Leroy, *Modèles déformables et modèles de déformation pour la reconnaissance de visage*, Thèse de l'Université Paris IX - Dauphine, 1996.

[Coh96] B. Leroy, I. L. Herlin, L. Cohen, "Face identification by deformation measure", *International Conference on Pattern Recognition (ICPR)*, Vienne, 1996.

[Wur97] R.P. Wurtz, "Object recognition robust under translations, deformations, and changes in background", *IEEE Transactions on pattern analysis and machine intelligence*, Vol 19, No 7, Juillet 1997.

[Lan97] A. Lanitis, C.J. Taylor et T.F. Cootes, "Automatic Interpretation and Coding of Face Images", *IEEE PAMI*, vol. 19, no. 7, July 1997.

[Lan95] A. Lanitis, C.J. Taylor et T.F. Cootes, "Active shape models-their training and application", *Computer Vision & Image Understanding*, vol.61 no.1, 1995.