Méthodes variationnelles pour la segmentation avec application à la réalité augmentée

Pauline Julian

L'UNIVERSITÉ DE THÈSE

THÈSE

pour obtenir le grade de

DOCTEUR de l'Université de Toulouse

Spécialité : **Spécialité Informatique**

préparée au laboratoire **IRIT**

dans le cadre de l'École Doctorale **EDMITT**

Titre:

Méthodes variationnelles pour la segmentation avec application à la Réalité Augmentée.

présentée et soutenue publiquement
par

Pauline JULIAN

le 12/10/12

Directeur de thèse: **Vincent Charvillat**

Jury

M. Philippe Joly,	Professeur - Président
M. Xavier Descombes,	Directeur de recherches - Rapporteur
M. Laurent Wendling,	Professeur - Rapporteur
M. François Lauze,	Professeur - Examinateur
M. Christophe Dehais,	Chercheur - Examinateur
M. Vincent Charvillat,	Professeur - Directeur de thèse
M. Ariel Choukroun,	Chercheur - Invité

Remerciements

Les travaux de thèse présentés dans ce manuscrit ont été effectués au sein de l'entreprise FittingBox et du laboratoire IRIT. Je suis donc reconnaissante envers Monsieur Ariel Choukroun, responsable de l'équipe de recherche de FittingBox, pour m'avoir accueilli. Je suis tout aussi reconnaissant envers Monsieur Vincent Charvillat pour m'avoir accueilli et pour avoir dirigé ma thèse. Tous les deux ont fait preuve de disponibilité malgré des emplois du temps chargés.

Je tiens aussi à remercier les membres du jury d'avoir été présents à Toulouse ou disponibles à distance lors de la soutenance. Je remercie Monsieur Xavier Descombes de l'INRIA et Monsieur Laurent Wendling de l'Université Paris 5, qui m'ont fait l'honneur d'accepter de relire ce document et d'apporter « texture et couleur » à ce manuscrit grâce à leurs observations et remarques pertinentes. Je remercie Christophe Dehais et François Lauze pour l'intérêt qu'ils ont chacun porté à ce manuscrit en l'examinant. Je remercie François Lauze pour m'avoir accueillie pendant quelques mois dans son équipe de recherche au DIKU, dans sa ville, son pays. J'ai beaucoup bénéficié de son co-encadrement. Je remercie beaucoup Christophe Dehais qui m'a énormément appris chez FittingBox et avec qui il est très agréable de travailler. Mes remerciements vont aussi à Philippe Joly de l'Université Paul Sabatier pour avoir présidé la soutenance d'un travail portant sur un sujet qu'il connait bien.

Je remercie grandement Pierre Gurdjos et Jean-Denis Durou chercheurs à l'IRIT qui ont su me donner de bons conseils pour la rédaction de ce manuscrit : merci pour les relectures et la belle expérience à Figeac durant l'école de printemps. Merci également à Adrien Bartoli pour notre publication commune. Je souhaite remercier aussi tous les membres de l'IRIT, permanents et thésards qui ont rythmé mes années de recherche sur les sites de l'ENSEEIHT. Que ce soit scientifiquement ou humainement, j'ai eu plaisir à partager ces nombreux moments d'échanges.

Je remercie l'ensemble de mes collègues de FittingBox pour tous les bons moments (les séminaires, les pauses café, les cinés, les parties de badminton, ...), pour les belles rencontres que j'ai pu faire durant toutes mes années : Vincent, Pierre, Diane, Alex, Nico, Delphine, Yannick, Khaled, Sylvain, ... Un merci à l'équipe des doctorants : Jérome, Olivier, Lilian, Benoit, Viorica, Benjamin, Axel pour ces bons jours partagés.

Je remercie enfin les membres de ma famille et plus particulièrement mes parents qui m'ont toujours soutenu et qui ont su me pousser pour aller plus loin.

Plus généralement, c'est avec plaisir que j'exprime ici mes remerciements à toutes les personnes qui ont contribué, de près ou de loin à ce document durant ces dernières années.

Enfin, je te remercie toi, lecteur, qui a pris la peine de lire ces remerciements jusqu'au bout. Je n'ai plus qu'à te souhaiter une bonne lecture de ce manuscrit.

Résumé

Dans cette thèse, nous nous intéressons au problème de la segmentation de portraits numériques. Nous appelons *portrait numérique* la photographie d'une personne avec un cadre allant grossièrement du gros plan au plan poitrine. Le problème abordé dans ce travail est un cas spécifique de la segmentation d'images où il s'agit notamment de définir précisément la frontière de la région « cheveux ». Ce problème est par essence très délicat car les attributs de la région « cheveux » (géométrie, couleur, texture) présentent une grande variabilité à la fois entre les personnes et au sein de la région. Notre cadre applicatif est un système d'« essayage virtuel » de lunettes à destination du grand public, il n'est pas possible de contrôler les conditions de prise de vue comme l'éclairage de la scène ou la résolution des images, ce qui accroît encore la difficulté du problème.

L'approche proposée pour la segmentation de portraits numériques est une approche du plus grossier au plus fin procédant par étapes successives. Nous formulons le problème comme celui d'une segmentation multi-régions, en introduisant comme « régions secondaires », les régions adjacentes à la région « cheveux » , c.-à-d. les régions « peau » et « fond ». La méthode est fondée sur l'apparence (*appearance-based method*) et a comme spécificité le fait de déterminer les descripteurs de régions les plus adaptés à partir d'une base d'images d'apprentissage et d'outils statistiques.

À la première étape de la méthode, nous utilisons l'information contextuelle d'un portrait numérique — connaissances *a priori* sur les relations spatiales entre régions— pour obtenir des échantillons des régions « cheveux », « peau » et « fond ». L'intérêt d'une approche fondée sur l'apparence est de pouvoir s'adapter à la fois aux conditions de prises de vue ainsi qu'aux attributs de chaque région. Au cours de cette étape, nous privilégions les modèles de forme polygonaux couplés aux contours actifs pour assurer la robustesse du modèle.

Lors de la seconde étape, à partir des échantillons détectés à l'étape précédente, nous introduisons un descripteur prenant en compte l'information de couleur et de texture. Nous proposons une segmentation grossière par classification en nous appuyant à nouveau sur l'information contextuelle : locale d'une part grâce aux champs de Markov, globale d'autre part grâce à un modèle *a priori* de segmentation obtenu par apprentissage qui permet de rendre les résultats plus robustes.

La troisième étape affine les résultats en définissant la frontière des « cheveux » comme une région de transition. Cette dernière contient les pixels dont l'apparence provient du mélange de contributions de deux régions (« cheveux » et « peau » ou «fond »). Ces deux régions de transition sont post-traitées par un algorithme de «démélange » (*digital matting*) pour estimer les coefficients de transparence entre « cheveux » et « peau », et entre « cheveux » et « fond ».

À l'issue de ces trois étapes, nous obtenons une segmentation précise d'un portrait numérique en trois « calques », contenant en chaque pixel l'information de transparence entre les régions « cheveux », « peau » et « fond ». Les résultats obtenus sur une base d'images de portraits numériques ont mis en évidence les bonnes performances de notre méthode.

Table des matières

Notations

- Les scalaires sont représentés par des lettres minuscules : a.
- Les vecteurs sont représentés par des lettres minuscules en caractère gras : $\boldsymbol{a}$.
- Les matrices sont représentées par des lettres majuscules : A.
- Les tenseurs d'ordre 3 sont représentés par des lettres majuscules en caractères gras : $\boldsymbol{A}$.
- L'opérateur transposée est noté : $.^t$
- Les images sont représentées par des fonctions

$$I : \Omega \longrightarrow \mathbb{R}^m,$$

  avec l'intensité lumineuse associée à chaque pixel de l'image, avec Ω un sous-ensemble de $\mathbb{R}^2$ représentant le support de l'image.
- Une région est un résultat de segmentation, elle est notée R.
- Une zone est un sous ensemble de la région, elle est notée Z.
- Un échantillon est un sous ensemble d'une zone.
- L'espace des courbes au moins deux fois différentiables : C^2

Termes anglais

Nous utiliserons ces termes dans les cas où la traduction d'expression vers le français peut être lourde ou complètement inappropriée.

- *Feature* : désigne les primitives géométriques d'intérêt, les caractéristiques d'un élément.
- *Snake* : (littéralement serpent) désigne le premier contour actifs introduit par Kass *et al.* [39].
- *Digital matting* : désigne un processus permettant d'extraire un objet à partir d'une image en estimant l'opacité couverte par l'objet en chaque pixel de l'image.
- *Chroma keying* : désigne le processus de segmenter des objets à partir d'images utilisant des arrières plans de couleur bleu.
- *Boosting* : est un principe qui regroupe de nombreux algorithmes qui s'appuient sur des ensembles de classifieurs binaires en optimisant leurs performances.
- *Eigenfaces* : (littéralement visage propre) désigne les vecteurs propres d'un ensemble de données visage.

Chapitre 1

Contexte de nos travaux au sein de la société FittingBox

1.1 Cadre de l'étude et contexte industriel

Dans ce chapitre, nous allons définir le contexte scientifique et industriel dans lequel ont été effectués nos travaux.

1.1.1 La réalité augmentée

La Réalité Augmentée (RA) consiste à combiner visuellement des éléments virtuels générés par un ordinateur, avec des éléments réels. Plus précisément, dans la littérature nous recensons deux définitions pour caractériser la RA. La première est introduite par Ronald Azuma [4]. Il définit la RA comme étant capable de :

- combiner les éléments réels et virtuels,
- gérer l'interaction des éléments en temps réel,
- inscrire des éléments virtuels dans l'espace tridimensionnel.

Cette définition exclut les cas d'ajout d'objet 2D, comme cela a pu être fait pour des interfaces homme-machine ou bien pour des traitements de post-production pour des films qui ne sont pas en temps réel.

Milgram et Kishino dans [55], proposent une définition plus large, pour eux la RA est définie au travers d'un *continuum* de réalité-virtuelle (cf Figure 1.1). La réalité virtuelle est définie en se déplaçant d'une scène réelle vers une scène virtuelle en passant progressivement par la réalité augmentée et la virtualité augmentée (VA). Selon les auteurs, la RA et la VA se distinguent par le fait que l'une est attachée à l'environnement réel tandis que l'autre est attachée à l'environnement virtuel. Plus précisément, la réalité augmentée correspond à l'immersion d'un élément virtuel dans une scène réelle alors que la virtualité augmentée est l'immersion d'un élément réel dans une scène virtuelle.

Avec cette nouvelle définition, la RA peut être temps réel ou pas ; l'objet virtuel peut être 2D ou 3D, réaliste ou pas.

Dans la suite, nous donnons un aperçu des applications qui utilisent la RA dans des domaines très variés : les applications médicales et militaires, le divertissement, le marketing.

FIGURE 1.1 – Le *continuum* de réalité-virtuelle, selon Milgram et Kishino.

Applications médicales : Dans le domaine médical, la RA aide le chirurgien à réaliser une intervention en superposant en temps réel un objet virtuel sur le patient. L'objet virtuel est bien souvent le résultat d'un jeu de données IRM ou radiographiques du patient lui-même. La composition des données virtuelles avec les données réelles permet au chirurgien d'avoir une vue d'ensemble sur les organes sans avoir recours à des techniques plus intrusives. Par exemple, Nicolau *et al.* [57] présentent un système d'aide à l'ablation de foie. Ils utilisent la RA, pour superposer le foie virtuel du patient dans le flux vidéo de l'intervention du patient (*cf.* Figure 1.2).

FIGURE 1.2 – Visualisation d'un foie virtuel pour aider à l'intervention d'ablation du foie - Source [57]

Applications militaires : Dans l'armée, la RA a vu le jour dans les années 80, en développant des applications permettant d'afficher des informations en temps réel sur les tableaux de bord ou bien sur les visières des pilotes de chasses (*cf.* Figure 1.3). Le pilote avait ainsi toutes les informations dans son champs de vision. Les besoins dans le domaine n'ont cessé d'augmenter. Aujourd hui, des produits sont en phase de recherche pour aider l'interaction entre les soldats sur le terrain. L'objectif est d'utiliser la RA pour informer et être informé clairement par ses coéquipiers des éventuels risques, ou bien d'avoir une vue d'ensemble de l'environnement en localisant la position de ses coéquipiers. L'idée est alors de porter une visière permettant d'augmenter en temps réel la scène réelle par des éléments visuels (*cf.* Figure 1.4).

FIGURE 1.3 – Affichage tête-haute pour un pilote de chasse

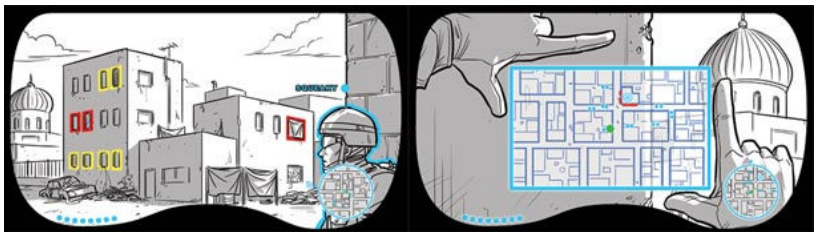


FIGURE 1.4 – Maquette d'affichage de l'information sur une visière d'un soldat.

Applications divertissantes : Avec l'apparition de smart-phone les petites applications de jeu utilisant la RA sont nombreuses. La majorité de ces jeux utilisent une mire de calibrage de la scène réelle. Elle permet de faciliter les interactions entre l'environnement réel et les objets virtuels. Par exemple, dans le cas illustré sur la Figure 1.5, le but du jeu est de protéger la tour virtuelle des attaques virtuelles, le tout dans un environnement réel. La protection se fait en bougeant le smart-phone pour que l'ennemi soit situé au milieu de l'écran et, en appuyant sur une touche du téléphone, l'ennemi est abattu. Ici, l'interaction entre le monde réel et le monde virtuel reste plutôt faible.

FIGURE 1.5 – Image issue du jeu *AR Defender*

Une autre application de jeu qui prend en compte plus d'interaction, est illustrée sur la figure 1.6. Ici le joueur fait un élevage d'animaux virtuels, il fait partie de la scène réelle et il doit interagir avec l'animal virtuel en le caressant virtuellement. Ce jeu utilise des interactions plus riches.

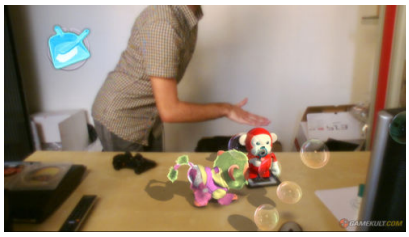


FIGURE 1.6 – Image issue du jeu *Eye Pet* développé par Sony.

Application marketing : Avec la vulgarisation des outils informatiques, la RA s'est naturellement implantée dans le secteur du marketing. A l'origine c'est un outil de communication véhiculant une image innovante. Avec l'apparition de sites e-commerce ce mode de communication tend à évoluer et il est devenu un outil permettant d'essayer virtuellement des objets. C'est le début de l'essayage virtuel.

1.1.2 Les solutions d'essayage virtuel

L'essayage virtuel est une solution technologique qui permet à un internaute d'essayer virtuellement et à distance des vêtements, des chaussures, des accessoires de mode... Cette fonctionnalité contribue à l'acte d'achat en projetant l'utilisateur dans un monde de réalité virtuelle. Scientifiquement, cette application n'est pas triviale. Elle doit présenter un objet virtuel fidèle à l'objet réel, et dans la majorité des cas, la scène réelle est augmentée. Il faut donc l'analyser pour superposer l'objet virtuel de la façon la plus réaliste.

FIGURE 1.7 – Essayage virtuel vêtements avec construction d'un personnage de synthèse.

Pour éviter la tâche difficile d'analyse de la scène, certaines applications, comme celle proposée par la firme H&M (*cf.* Figure 1.7), demande à l'utilisateur de construire un personnage de synthèse qui lui ressemble. Ceci permet d'acquérir une représentation tridimensionnelle de notre personnage, notre avatar. Ainsi, la superposition entre l'objet virtuel et le mannequin est plus aisée. En revanche, le résultat n'est pas assez proche de la réalité pour que l'utilisateur se sente acteur de l'essayage.

(a) Essayage virtuel de T shirt proposé par l'IFHH

(b) Essayage virtuel de lunettes proposé par FittingBox

FIGURE 1.8 – Essayage virtuel de vêtements utilisant la RA.

Pour qu'il ait l'illusion d'essayer un produit, l'utilisation de la scène réelle est primordiale et l'interaction avec l'utilisateur est un plus. Suite à ces remarques, nous voyons apparaître des solutions mettant en place un dispositif équipé d'une caméra pour filmer l'utilisateur et d'un écran qui se transforme en un miroir virtuel. Ainsi, l'utilisateur est filmé, puis on superpose en temps réel l'objet virtuel sur le flux vidéo , et instantanément l'utilisateur se projette avec l'objet essayé. La figure 1.8 illustre deux types de miroirs virtuels. L'image de gauche montre un produit développé par l'Institut Fraunhofer des techniques de communication Heinrich-Hertz. Il permet d'essayer virtuellement des T shirts

à condition de porter leur T-shirt de référence. L'image de droite est un produit développé par la société FittingBox. Il permet d'essayer virtuellement des lunettes sans avoir besoin de marqueurs, ou de connaissances *a priori* (contrairement au T-shirt de référence).

FIGURE 1.9 – Solution envisagée pour que l'utilisateur ait un indicateur du confort lors du port des lunettes.

Les logiciels d'essayage virtuel actuels permettent à l'utilisateur de faire un choix esthétique. Mais pour le moment, aucune solution ne prend en compte la notion de confort. Cette problématique n'est pas évidente et nécessite une analyse fine de la scène. Nous pourrions envisager dans le cas des lunettes de faire apparaître une carte de chaleur pour qualifier les zones de pression (*cf.* Figure 1.9). Il est très important de connaître la physique des matériaux, la géométrie des lunettes et du visage. Ici nous voyons donc qu'une analyse précise de certaines parties de la scène comme le nez, les oreilles est primordiale pour répondre à la question de confort pour l'essayage virtuel de lunettes.

1.1.3 Les applications existantes chez FittingBox

FittingBox est une entreprise spécialisée dans l'édition de logiciels pour l'essayage virtuel de lunettes. L'idée est née lorsque les fondateurs de l'entreprise réalisent que leur myopie les rendait incapables de se voir dans le miroir en essayant des lunettes chez l'opticien. Ils décident alors de mettre en place une solution qui permet au client d'essayer des lunettes depuis leur domicile à partir d'une photo. L'intérêt devient double, il est alors plus facile de demander conseil à son entourage sur le choix des montures car les personnes n'ont pas besoin de se déplacer chez l'opticien pour voir le résultat. Le résultat est en fait une photo augmentée d'un modèle virtuel de lunettes. Le deuxième avantage est que l'utilisateur muni de sa correction peut alors se voir en photo avec ses nouvelles montures sans avoir la vue troublée.

De cette analyse sont nés deux produits **Fit Photo** et **Fit Live**, qui sont respectivement des solutions d'essayage virtuel utilisant un support photo ou une webcam. Une démonstration de ces deux logiciels est disponible sur le site web de l'entreprise [1]. Tous deux se décomposent en deux grandes phases, qui sont la détection du visage et la composition entre le monde réel et l'objet virtuel (*cf.* Figure 1.10). Que le support soit une image ou bien une séquence vidéo, la première étape de l'algorithme du logiciel d'essayage virtuel de lunettes consiste à détecter la pose du visage par rapport à la caméra, c'est-à-dire connaître la position, l'orientation et l'échelle du visage dans la scène réelle. Dans le cas

1. `http://demo.fittingbox.com/`

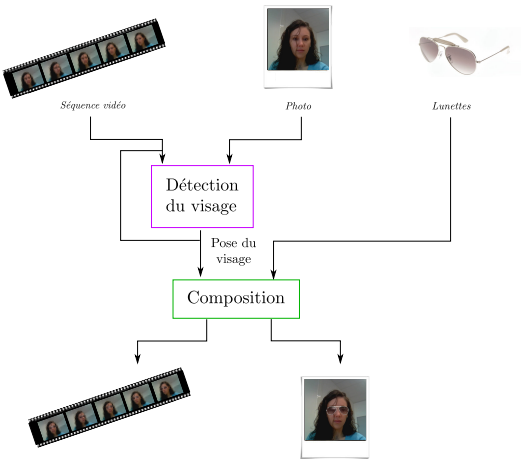


FIGURE 1.10 – Schéma de principe de l'essayage virtuel de lunettes.

de la séquence vidéo, une boucle sur l'étape de détection du visage permet de le suivre image après image. Après avoir détecté le visage, l'étape de composition ajoute, de manière réaliste, les lunettes virtuelles dans la scène réelle. Le résultat final est donc une séquence vidéo ou une photo augmentée de lunettes virtuelles.

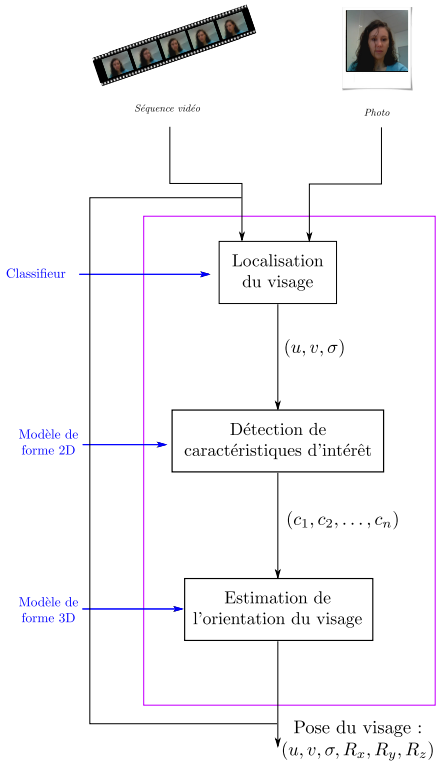


FIGURE 1.11 – Détail de l'étape de détection du visage de la Figure 1.10.

Plus précisément, la détection du visage, schématisée sur la figure 1.11, a pour objectif de définir pour une image de type portrait, les six paramètres suivants :

- la position du visage dans l'image (u, v),
- la taille du visage σ,
- l'orientation du visage par rapport à la caméra, dans l'espace tridimensionnel (R_x, R_y, R_z).

L'algorithme de détection se décompose en trois temps. Conjointement, nous localisons la position (u, v) du visage dans le repère associé à l'image et définissons la taille (σ) du visage dans l'image. Cette étape est rendue possible, par exemple, en utilisant l'algorithme développé par Viola et Jones [79] fondé sur l'utilisation de classifieurs. En réduisant l'image à la zone contenant le visage, nous recherchons dans ce sous ensemble les caractéristiques d'intérêt du visage. La recherche peut se faire sur des points d'intérêts du visage en utilisant un modèle actif de forme de forme 2D. La combinaison de l'information des caractéristiques d'intérêt du visage et d'un modèle 3D de visage permettent d'estimer son orientation (R_x, R_y, R_z) par rapport à la caméra.

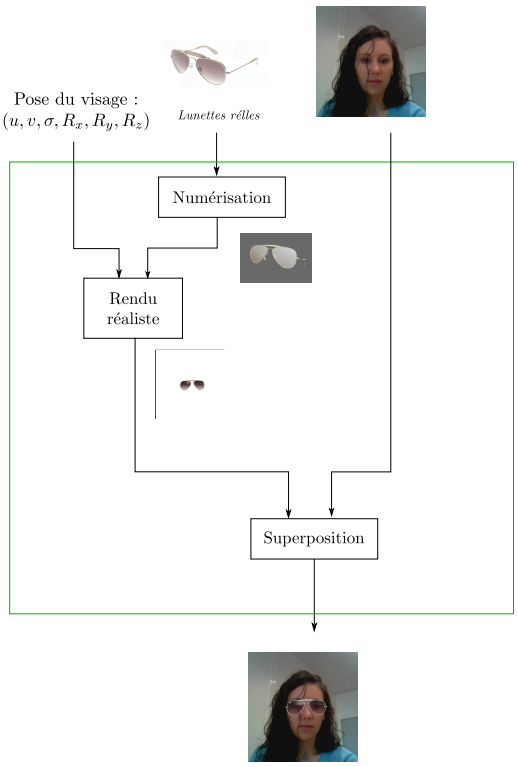


FIGURE 1.12 – Détail de l'étape de composition de la Figure 1.10.

L'étape de composition, schématisée à la figure 1.12, superpose les lunettes virtuelles dans l'image de type portrait. A partir des lunettes réelles, la numérisation permet de construire un modèle numérique de l'objet, comprenant généralement des informations géométriques (maillage 3D) et des informations d'apparence (texture, paramètres des matériaux ...). Après numérisation et connaissant les paramètres d'orientation du visage, nous nous intéressons à la phase de rendu réaliste. A partir des paramètres $(u, v, \sigma, R_x, R_y, R_z)$, nous positionnons dans la scène réelle un modèle générique 3D de visage. Ce dernier permet d'effacer les parties des lunettes occultées par le modèle 3D de visage (ex : la fin des branches occultées par les oreilles). Le résultat est un "calque" contenant les lunettes virtuelles à l'emplacement souhaité pour l'essayage. Par superposition sur l'image de type portrait, nous obtenons le résultat de l'algorithme d'essayage virtuel de lunettes.

Pour que l'utilisateur soit satisfait des résultats de l'essayage virtuel de lunettes, il est primordial d'avoir un rendu photo-réaliste de la scène. Dans la section suivante, nous listons un ensemble de fonctionnalités qui contribuent à l'amélioration de la phase de composition.

1.1.4 Pour aller plus loin dans le réalisme

FittingBox s'applique à proposer une solution d'essayage virtuel le plus réaliste possible. De ce fait, l'étape de composition, qui superpose l'objet virtuel dans la scène réelle, est primordiale. Les fonctionnalités avancées pour que l'entreprise puisse atteindre un degré élevé de photo-réalisme sont :

– L'adaptation du rendu des lunettes à la qualité de la photographie. Une image bruitée augmentée avec un modèle de lunettes parfait ne fourni pas un rendu très réaliste. Il faudrait pouvoir débruiter l'image afin de mettre en valeur l'essayage (*cf.* Figure 1.13).

(a) Image source. (b) Essayage sur l'image bruitée. (c) Essayage sur l'image débruitée.

FIGURE 1.13 – Débruitage de l'image source pour améliorer l'intégration photo-réaliste.

– L'estimation de l'éclairage de la scène réelle pour pouvoir appliquer sur le modèle virtuel le même éclairage. Ce ré-éclairage provoque un changement d'apparence de la texture des lunettes (*cf.* Figure 1.14) et un changement de position et d'intensité de l'ombre projetée virtuelle des lunettes.

(a) Image source. (b) Essayage sans analyse de la lumière. (c) Essayage avec analyse de la lumière.

FIGURE 1.14 – Ré-éclairage de l'objet virtuel pour améliorer l'intégration photo-réaliste.

– L'interaction physique entre les lunettes et les objets de la scènes, qui sont : les oreilles, le nez, les cheveux. Nous pouvons constater que sur la Figure 1.15 une mèche de cheveux occulte une partie des lunettes.

Comme nous venons de le signaler, les produits de FittingBox visent à obtenir une augmentation réaliste de la scène. De plus, nous pouvons d'autant plus facilement juger si un produit nous convient que nous pouvons l'essayer en contexte. Une idée pour ajouter

(a) Image source. (b) Essayage virtuel avec une mèche de cheveux qui passe sous les lunettes. (c) Essayage virtuel avec une mèche de cheveux qui passe sur les lunettes.

FIGURE 1.15 – Gestion des interactions entre les lunettes virtuelles et les cheveux réels.

(a) Image source chargée par l'utilisateur. (b) Essayage virtuel sans modification de l'arrière plan. (c) Essayage virtuel avec modification de l'arrière plan.

FIGURE 1.16 – Immersion de l'utilisateur dans un nouvel environnement.

du contexte lors de l'essayage est de modifier l'arrière plan de l'image source (*cf.* Figure 1.16).

Après présentation des fonctionnalités avancées qui intéressent l'entreprise, nous voyons apparaître un grand nombre de problèmes scientifiques. Nous avons décidé de concentrer nos efforts sur deux des améliorations citées précédemment, et cela dans le cas du produit utilisant la technologie photo. Le premier apport doit permettre de gérer l'interaction physique entre les cheveux et les lunettes. Pour cela, nous proposons de détecter la frontière entre la peau et les cheveux, pour segmenter les lunettes en fonction de la position des cheveux. Le deuxième apport est celui lié à l'essayage en contexte, nous proposons alors de segmenter l'arrière plan. Ces deux besoins stratégiques pour FittingBox se rejoignent scientifiquement autour du thème de la segmentation, et viennent enrichir l'étape de composition (*cf.* Figure 1.17). Par conséquent, en partenariat avec l'Institut de Recherche en Informatique de Toulouse (IRIT) et le laboratoire DIKU de Copenhague, nous travaillons sur la segmentation d'image de type portrait, en trois régions. Les trois régions sont la peau, les cheveux et le fond (ou arrière-plan).

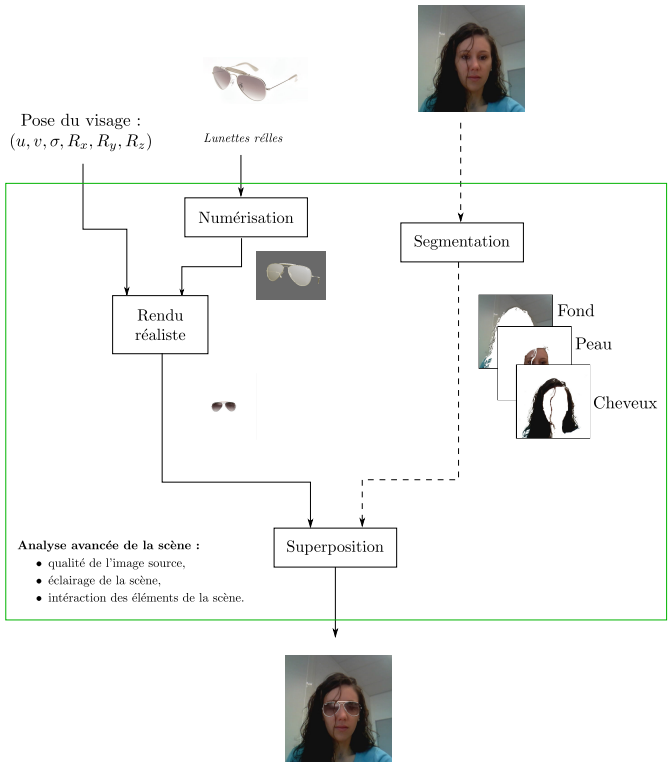


FIGURE 1.17 – Amélioration de l'étape de composition par la segmentation.

1.2 Le problème de segmentation posé

1.2.1 Sujet et méthode

L'objectif de nos travaux est de segmenter une image de type Portrait en trois régions ou segments : la peau, les cheveux, et le fond. Dans la suite, nous employons parfois le terme de "catégorie" pour désigner les classes Peau, Cheveux et Fond sans préciser si nous parlons des étiquettes associées aux pixels, aux superpixels, aux régions, aux échantillons de l'image, *etc*. Lorsque nous parlons de catégorie peau, nous entendons les propriétés visuelles de la (classe) peau indépendamment de l'entité visuelle à étiqueter. Nous proposons de définir la segmentation d'image comme la séparation d'un ensemble de segments qui composent l'image. Ici, les segments correspondent aux régions de la peau, des cheveux et du fond. La segmentation d'image est une tâche difficile puisque c'est un problème mal posé. Pour y remédier, il faut définir des contraintes sur chaque segment et ainsi rendre le problème mieux posé. Par exemple, nous pouvons imposer des contraintes sur les propriétés colorimétriques ou encore géométriques des segments. La difficulté est de définir les contraintes qui fourniront le résultat attendu. Pour cela, il est important d'analyser au préalable les différents segments.

Nous constatons que la peau est située au centre du visage ; sa teinte est contrainte à varier entre les peaux claires et les peaux mates, il est clair que nous ne pouvons rencontrer des peaux de teinte bleue. Enfin, le visage est composé d'éléments caractéristiques comme la bouche, les yeux, le nez. Ces élements présentent des propriétés d'invariance qui les rendent détectables.

Ensuite, nous remarquons pour les cheveux qu'ils sont situés autour de la peau et ont des formes variables mais contraintes. Tout comme la peau, la teinte des cheveux

ne couvre pas la totalité de l'espace de couleur. Ils sont bruns, châtains, roux, blonds ou blancs. Enfin, un dernier élément qui permet de décrire la chevelure est sa texture correspondant aux cheveux frisés, lisse, crépus.

Si la peau et les cheveux comportent des propriété intra-classe, c'est plus délicat dans le cas du fond. La seule remarque que nous pouvons faire à son sujet, est qu'il se situe autour du personnage.

Outre le problème de définir pour chaque catégorie, les contraintes qui permettent d'obtenir la segmentation souhaitée (la segmentation de la peau, des cheveux et du fond), la segmentation soulève une autre difficulté. La localisation précise des frontières qui séparent la peau des cheveux et les cheveux du fond n'est pas triviale. Ces contours sont difficiles voire impossibles à définir, car il existe une zone de transition entre les cheveux et la peau (ou le fond). Cette zone de transition, que nous appelons par la suite zone de mélange (en gris sur la Figure 1.18), correspond aux pixels dont les couleurs entre les cheveux et la peau (ou le fond) se mélangent.

(a) Image source (b) En gris la zone de mélange

FIGURE 1.18 – Visualisation d'une zone de mélange entre le fond et les cheveux.

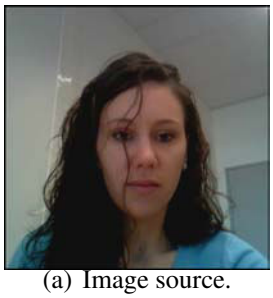
Nous devons être capable de fournir pour chaque catégorie, trois masques complémentaires à valeur dans $[0, 1]$ (*cf.* Figure 1.19). La valeur 0 signifie que le pixel n'appartient pas à la catégorie, la valeur 1 signifie que le pixel appartient complètement à la catégorie, et les valeurs intermédiaires signifient que le pixel est composé d'un mélange de catégories. L'ensemble de ces pixels est appelé zone de mélange.

La segmentation d'image de type portrait est un problème complexe. Une approche grossière qui s'affine étape par étape en prenant en compte la notion de contexte, nous semble être une bonne solution. Cet algorithme se décompose en quatre étapes :

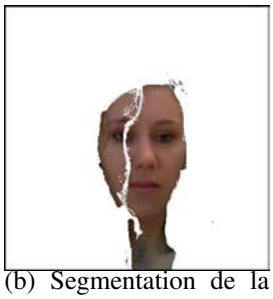
- la détection,
- la segmentation grossière par classification,
- la segmentation précise en utilisant les méthodes variationnelles,
- le *matting* (défini dans la suite de ce paragraphe).

Avant de définir comment le contexte intervient dans chaque étape, nous le définissons. Galleguillos et Belongie [30] scindent l'information contextuelle en trois parties.

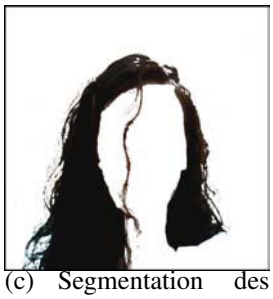
- Le contexte sémantique : il exploite la vraisemblance qu'un objet puisse appartenir à une scène. Par exemple, nous détectons un iris dans une scène de visage et non une bille à jouer. Nous remarquons que du fait que nous travaillons uniquement avec des images de type portrait, nous fixons le contexte sémantique et donc il ne pas y avoir de doute sur la vraisemblance qu'un objet appartienne ou non à la scène.
- Le contexte spatial : il exprime la vraisemblance de la position d'un objet dans la scène. Par exemple, dans le visage le nez doit se retrouver au centre des yeux et de la bouche.



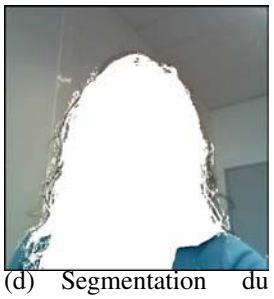
(a) Image source.



(b) Segmentation de la peau.



(c) Segmentation des cheveux.



(d) Segmentation du fond.

FIGURE 1.19 – Segmentation de l'image source en trois catégories : la peau, les cheveux et le fond.

– Le contexte d'échelle : il exploite la vraisemblance de la taille d'un objet par rapport à la taille de ses objets voisins. Toujours dans le cas du visage, le contexte d'échelle est illustré par le fait que la taille des yeux contraint la taille de la bouche. Cette contrainte est issue des analyses sur l'anthropométrie du visage.

La première étape de détection est une étape à deux niveaux qui doit permettre dans un premier temps de détecter le visage et donc définir dans l'image la plus petite boite englobante contenant le visage. La deuxième étape de détection doit permettre de définir trois zones de l'image comme étant *a priori* des échantillons des segments de peau, de cheveux et de fond. La détection de visage n'est pas l'objet de notre analyse, nous utilisons des algorithmes existants qui ont fait leurs preuves, comme par exemple, celui de Viola et Jones [80]. Pour la détection des échantillons des régions de peau, de cheveux, de fond nous utilisons le contexte spatial et d'échelle dans lequel nous nous trouvons. Nous mettons en place une détection qui vise à maximiser la dissemblance entre les 3 échantillons. Le choix des échantillons est fondé sur un modèle géométrique, dynamique et contraint.

L'étape de segmentation grossière par classification segmente l'image à partir de ses données colorimétriques et fréquentielles. Le contexte d'échelle permet dans ce cas de définir une plage de fréquence intéressante pour analyser l'image. Le contexte spatial quant à lui, permet de réduire les erreurs de classification des pixels en mettant en place des cartes de segmentation *a priori*. En utilisant les informations de l'étape de détection, nous mettons en place un apprentissage adaptatif aux caractéristiques colorimétriques et fréquentielles de chacune des classes Peau, Cheveux et Fond.

L'étape de segmentation précise, utilisant les méthodes variationnelles, affine les résultats précédents. Pour cela, nous mettons en équation une segmentation prenant en compte les zones de mélange (*cf.* Figure 1.18) entre les classes Peau et Cheveux et les classes

Cheveux et Fond. Nous obtiendrons alors une segmentation en 5 régions. Les méthodes variationnelles permettent de traiter conjointement les segmentations des différentes régions et ainsi mettre en compétition le choix d'étiquettage des pixels ; car nous aurions pu traiter uniquement la segmentation des cheveux pour intégrer de manière réaliste des lunettes virtuelles dans la scène. Enfin, la segmentation par méthodes variationnelles permet surtout de garder le contrôle sur la topologie des courbes produisant la segmentation.

Enfin, l'étape de *matting* calcule le coefficient de transparence entre la peau, les cheveux et le fond et permet ainsi de retrouver les trois masques de segmentation de la peau, des cheveux et du fond (*cf.* Figure 1.19).

En résumé, les quatre étapes de notre algorithme de segmentation sont fondée sur :
- la mise en place d'un modèle géométrique *a priori* de la scène,
- l'apprentissage adaptatif des caractéristiques de l'image pour les classes Peau, Cheveux et Fond,
- la segmentation conjointe des régions,
- la contrôle de la topologie des courbes de segmentation.

1.2.2 Images traitées

Le logiciel que nous souhaitons améliorer est une application grand public, cela implique que nous ne maîtrisons pas un ensemble de paramètres qui sont :
- la résolution de l'image,
- les conditions d'illumination,
- l'environnement de prise de vue.

Nous ne nous intéressons qu'aux images en couleur et faisons l'hypothèse de toujours être en présence d'un visage, de cheveux et d'un arrière plan. Nous appellerons le type des images traitées : Portrait. Nous constituons une base d'images Portrait (*cf.* Figure 1.20). Elle est représentative de ce que nos algorithmes sont susceptibles de rencontrer. Elle contient des images avec une grande variabilité en résolution, en apparence en termes de forme, de texture et de couleur des cheveux. Enfin, la variabilité des images est également due aux conditions de prise de vue et d'illumination.

Face à la grande variabilité des images à traiter voyons les conséquences sur les différentes catégories. Pour le Fond, nous ne disposons pas d'*a priori* colorimétrique comme nous pourrions en avoir en utilisant une technique de *chroma keying*. Pour les Cheveux, la difficulté de segmentation vient du fait que les frontières ne sont pas précises, la résolution et la qualité de la photo sont très importantes pour capturer la texture des cheveux. Une image de faible résolution ou bien floue ne permettra pas de capturer finement cette caractéristique. Pour la Peau, nous devons être capable de traiter des personnes ayant la peau mate comme des personnes ayant la peau claire.

1.3 Organisation du manuscrit

Dans les chapitres suivants, nous développons une méthode de segmentation des images Portrait dans laquelle s'inscrit notre contribution.

Le chapitre 2 présente un état de l'art sur les travaux traitant du visage en vision par ordinateur. Nous nous intéressons également aux différentes techniques de segmentation.

Dans le chapitre 3, nous présentons notre segmentation grossière. Elle est le résultat d'une classification qui se fonde sur un des trois concepts de notre contribution, qui est

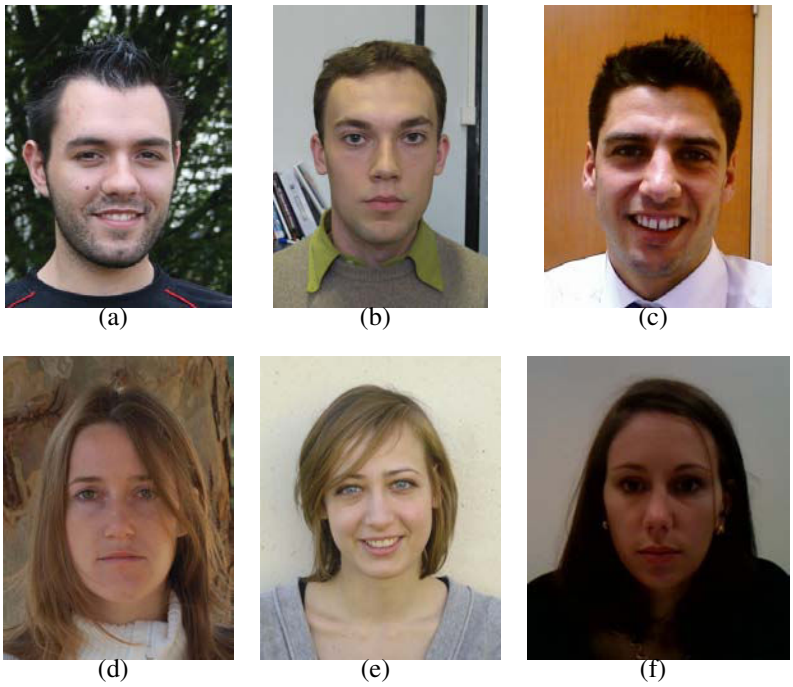


FIGURE 1.20 – Échantillon de la base d'images Portrait.

l'apprentissage supervisé des caractéristiques de la peau, des cheveux et du fond.

Dans le chapitre 4, nous proposons une segmentation plus fine, initialisée proche de la solution finale et employant les méthodes variationnelles. A cette étape, nous exploitons l'idée de segmentation conjointe.

Dans le chapitre 5, ce n'est qu'une fois les résultats de segmentation validés que nous proposons de détecter les échantillons de Peau, Cheveux et Fond, pour supprimer la partie supervisée du chapitre 3.

Finalement, dans le chapitre 6, nous intégrons l'ensemble des solutions technologiques présentées précédemment.

Chapitre 2

Portrait numérique : de la détection à la segmentation

L'objet de cette thèse est de définir une frontière précise entre les régions Peau, Cheveux et Fond pour une image portrait aussi appelée *portrait numérique*.

Nous appellerons *portrait numérique* une photographie d'une personne, qu'elle soit prise de près, révélant le visage uniquement, ou cadrée plus large jusqu'au buste. Nous faisons l'hypothèse que la prise photo est proche de la position de face.

La segmentation de portraits numériques est un cas spécifique de la segmentation d'images puisque le contexte "portrait" réduit fortement l'espace des solutions. Toute technique de segmentation du portrait numérique nécessite une étape préliminaire de « détection du visage » dans l'image. Dans la littérature [87], on généralise souvent cette tache à la « détection de visages » *au pluriel* : étant donnée une image arbitraire, il s'agit de déterminer la présence ou non de visages dans l'image en l'associant, dans le cas d'une décision positive, à une région (connexe ou pas) de l'image.

La détection de visage(s) est donc dans sa formulation la plus simple un problème de classification binaire, pour lequel toute région de l'image est classée comme Visage ou ¬Visage. Bien que, dans ce cas, il s'agit d'un problème que nous supposerons ici résolu, nous donnerons au sein de ce chapitre un rapide état de l'art. Celui-ci nous semble en effet être une introduction naturelle aux descripteurs de régions regroupant les caractéristiques visuelles d'un visage, qu'elles soient colorimétriques, géométriques, de texture etc, mais aussi une première approche pour l'exploitation de ces descripteurs par des techniques de classification par apprentissage. Nous compléterons cet état de l'art en complexifiant le problème de détection de visage(s) par l'introduction de classes supplémentaires, pour devenir un problème de classification n-aire dès lors qu'il s'agit de déterminer en plus la pose du visage (c.-à-d. l'orientation de la personne qui pose relativement à l'appareil : profil droit ou gauche, vue de trois quarts, de face etc.). À noter que, dans ce cas, il y a une difficulté supplémentaire qui est celle d'induire une information 3D à partir d'indices visuels 2D. Nous terminerons l'état de l'art en déclinant le problème « du plus grossier au plus fin » pour aboutir au problème de la segmentation du portrait numérique. Nous énumérerons les difficultés que l'on peut rencontrer pour segmenter un portrait numérique liées à l'introduction de classes Peau, Cheveux, Vêtements et Fond.

Au cours de ce chapitre, nous nous intéresserons plus particulièrement :

– aux méthodes de détection de visage(s) fondées sur l'apparence (*appearance-based methods*) dont l'objet est de déterminer les « meilleurs » descripteurs à partir d'une base d'images, dite d'apprentissage, et d'outils statistiques. La base d'apprentissage

collecte des exemples positifs permettant de classer une région de l'image comme Visage ou ¬Visage. À ce titre, nous définirons ce qu'est la classification et quel est son rôle pour l'apprentissage.
– aux méthodes variationnelles par courbes de niveaux et nous argumenterons pourquoi une de leurs propriétés, qui est de permettre les changements de topologie des courbes (une frontière pouvant être représentée par un ou plusieurs contours fermés), est très adaptée à la segmentation de la région Cheveux.

2.1 Détection du visage

La détection du visage est une tâche de la vision par ordinateur qui suscite un très grand intérêt et qui a fait l'objet de très nombreuses publications. Récemment, elle est une composante de certaines applications grand public, utilisée :
– par les appareils photos numériques,
– par les webcams motorisées,
– pour taguer des personnes identifiées (*Picasa*, *Facebook*, ...),
– pour « anonymiser » des personnes, en utilisant le flou ou tout autre artifice, par exemple dans les images *Street View* de *Google Maps*...

Ainsi, la détection du visage est souvent utilisée pour la reconnaissance d'identité qui cherche à identifier le visage détecté (et localisé) en vérifiant sa présence dans une base de données. Lorsqu'elle sont automatisées, les techniques de reconnaissance de visage peuvent être utilisés pour la vidéo-surveillance par exemple.

Dans ce qui suit, nous présentons les différentes problématiques de détection du visage et les difficultés qu'elles engendrent.

2.1.1 Différents problèmes

En fonction des besoins de l'application, différents niveaux de précision peuvent être attendus des résultats de détection du visage. Nous pouvons vouloir simplement détecter la présence ou l'absence de visage dans l'image, ou plus précisément déterminer l'orientation du visage dans l'image. Nous définissons trois niveaux de tâches pour ce sujet. Le premier niveau, le plus simple est de définir la détection de visage comme une tâche qui localise le visage en donnant une position qui le caractérise (centre de gravité, etc.). Dans ce cas là, nous sommes en mesure de dire si oui ou non l'image traitée comporte un visage et/ou de compter le nombre de visages présents dans l'image. Le deuxième niveau de difficulté est atteint lorsque nous souhaitons aussi déterminer l'« échelle » du visage, par exemple par le biais du boîte englobante (c.-à-d. un rectangle entourant le visage). Cette échelle permet d'aligner les visages entre eux et ainsi appliquer des algorithmes de reconnaissance (sur l'expression du visage, le genre, l'âge, l'identité de la personne). Enfin, le dernier niveau de difficulté ajoute aux problématiques précédentes l'estimation de la pose (de face, vers la droite, vers la gauche) voire de l'orientation de la tête de la personne relativement à la caméra. Ceci permet d'obtenir des données « comparables » pour les algorithmes de reconnaissance.

Ces différents niveaux de complexité font apparaître différentes exigences de robustesse aux variations des paramètres des portraits numériques. La position, l'échelle et la pose du visage ne doivent pas influer sur la qualité de la détection. À cela, nous ajoutons

le fait que la détection doit être robuste aux variations de condition d'illumination de la scène, de la morphologie et des origines ethniques (apparence des yeux, de la bouche, de la couleur de peau etc.), de l'expression du visage, des occultations, de la résolution du visage dans l'image.

2.1.2 Extraction des descripteurs du visage

Il s'agit ici de décrire les caractéristiques visuelles d'un visage qui vont permettre de le détecter dans une image. Un visage est composé de peau ; la couleur joue donc un rôle primordial. Mais un visage est aussi une organisation fortement contrainte de ses éléments (comme la bouche, le nez...). On voit que les caractéristiques du visage regroupent à la fois des *caractéristiques d'apparence* et des caractéristiques structurelles que nous désignons par *caractéristiques de forme*. Les caractéristiques les plus compactes et les plus pertinentes —regroupées sous le terme *descripteur du visage*— vis-à-vis du niveau de difficulté de détection doivent être considérées. La compacité implique que les descripteurs sont représentés par des points dans un espace de « dimension réduite ». La pertinence doit garantir les deux propriétés suivantes :

- si deux visages ont la même apparence/forme alors leurs représentations dans l'espace des descripteurs doivent être identiques ;
- si deux visages ont des apparences/formes différentes alors leurs représentations doivent être différentes.

2.1.2.1 Descripteur colorimétrique de la peau

Le problème sous-jacent ici est la détection de la « Peau » dans un portrait numérique via une classification binaire qui consiste à classer toute région comme Peau ou ¬Peau. Dans la littérature, la construction de descripteurs de couleur adaptés à la peau est directement liée à la définition d'un espace de couleur dans lequel un pixel est représenté par un système de coordonnées. Nous invitons le lecteur à se reporter à [27], pour une introduction complète aux systèmes auxquels nous faisons référence dans ce qui suit : les systèmes de primaires *RGB* et *RGB* normalisé, les systèmes luminance-chrominance *YUV*, *YES* et *L*a*b* et enfin le système perceptuel *HSV* .

Nous présentons maintenant comment certains auteurs font le choix d'un espace de couleur en fonction des résultats de classification. Bien sûr, le résultat de la classification ne dépend pas uniquement de la qualité des descripteurs mais également du modèle décisionnel.

Choix d'un système de représentation de la couleur. Zarit *et al.* [88] ont essayé de déterminer l'espace de couleur le plus adapté à la détection de la peau, c'est à dire celui qui obtient les meilleurs résultats de classification. Ils utilisent deux méthodes de classification (par seuillage et à l'aide du théorème de Bayes) et quatre scores d'évaluation (pourcentage de pixels Peau et ¬Peau bien détectés, faux positifs, faux négatifs, pourcentage de pixels Peau bien détectés). La conclusion de cet article est que l'espace qui donne les meilleurs résultats pour la classification de pixels peau est l'espace *HSV*.

Combinaison de plusieurs systèmes de représentation. Gomez *et al.* [32] cherchent à construire un nouvel espace de couleur pour lequel chaque composante est extraite d'un

espace existant. Pour cela, ils évaluent indépendamment chaque composante des espaces *RGB*, *TSV*, *L*a*b**. Les composantes sélectionnées répondent aux critères suivants :

- la plage de variation des intensités des pixels de la peau est compacte ;
- le recouvrement entre la plage de variation des pixels de la peau et celle des pixels n'étant pas de la peau est minimal ;
- les composantes sont le plus complémentaires possible.

Les auteurs obtiennent l'espace de couleur en sélectionnant la composante E issue de l'espace *YES*, la composante R/V associée à l'espace *RVB* et enfin H de *HSV*. Cette étude montre que la teinte (composante H) possède des propriétés intéressantes pour la description de la peau. Comme on peut le prévoir, une grande partie de l'information est contenue « près » de la couleur rouge, mais l'étude révèle que le vert joue un rôle important. Nous pensons que ceci est peu être dû au fait que le vert soit « l'opposé » du rouge pour le cerveau, et donc encode des caractéristiques complémentaires.

Utilisation de la redondance des systèmes de représentation. Singh *et al.* [67] proposent de combiner les résultats de détection de la peau obtenus en employant les espaces de couleur RGB, *YUV* et *HSI* (qui est très proche des caractéristiques de *HSV*, déterminé comme étant l'espace le plus adapté pour la détection de peau par Zarit *et al.*). Indépendamment, les résultats de détection de la peau utilisant *YUV* ou *HSI* sont largement plus performants que ceux utilisant *RGB*, ce qui corrobore l'étude de Zarit *et al.* [88]. Mais leur complémentarité rend le résultat final utilisant les trois espaces beaucoup plus performant. Dans cet article, les auteurs ont eu l'idée de multiplier les résultats de détection robuste (sans faux positif). La redondance d'information aide à obtenir une détection finale à la fois précise et robuste.

La conclusion naturelle à la lumière des études effectuées ces dernières années est que caractériser la peau uniquement par la couleur n'est pas suffisant. D'où les nombreux travaux visant à intégrer la texture [1, 23, 35].

2.1.2.2 Descripteur géométrique du visage

Au début des années 90, les modèles actifs de forme ont été introduits par Cootes et *al.* [17], pour décrire la « géométrie » d'un objet. L'idée d'un descripteur d'apparence géométrique est de construire un modèle des contours qui doit être capable de se déformer en fonction des propriétés de la classe d'objets qu'il représente. Dans le cas des visages, ce type de descripteur regroupe un ensemble de caractéristiques d'apparence communes à tous les individus : contours du visage, des yeux, de la bouche, etc. La difficulté est ici de modéliser les variations d'apparence entre individus car les déformations des contours ont une très grande variabilité d'un visage à un autre : les yeux peuvent être plus ou moins grands, le visage plus ou moins large ...

Une approche très courante concernant la plupart des modèles d'apparence géométrique consiste à modéliser ces déformations par apprentissage à partir d'une base d'images (instances du modèle) annotées manuellement. L'apprentissage au sens large est l'acquisition de savoir-faire. En vision par ordinateur, il s'agit d'élaborer des méthodes statistiques, à partir d'une base de données initiale, permettant de remplir des tâches qu'il est impossible ou trop complexe de résoudre par une mise en équation et des mises en œuvre algorithmiques plus classiques.

L'annotation des données d'apprentissage :

Les techniques d'apprentissage incluent en général une phase d'annotation de la base de données. Cette phase d'annotation est très importante pour que le modèle soit adapté à l'objet que l'on souhaite décrire. Une approche très répandue consiste à modéliser la forme de l'objet par un nombre fixe de n points de contrôle. Elle nécessite une expertise humaine « au cas par cas » afin d'associer une sémantique précise de l'objet à chacun de ces points. Si le choix des points à étiqueter n'étaient pas correctement établi alors nous pourrions obtenir un modèle incomplet qui ne serait pas fidèle à la base d'images. Également, si l'étiquetage des points n'était pas fait précisément cela introduirait du biais et donc un modèle inadapté.

L'analyse des données :

Afin de capturer la statistique des données apprises, une normalisation géométrique des données est nécessaire et va permettre de comparer les différents individus de la base d'apprentissage. Cette normalisation géométrique consiste à « aligner » les ensembles de points de contrôle de chaque individu en leur appliquant individuellement une certaine similitude plane, c.-à-d. la composition d'une homothétie, d'une rotation et d'une translation. En ce qui concerne le cas des visages, les points sont normalisés par rapport à la « géométrie » des yeux (*cf.* Chapitre 5). Ainsi l'ensemble des données annotées sont exprimées dans un même repère affine et sont donc comparables.

Cootes *et al.* proposent de construire un modèle à distribution de points en analysant N données, issues de la base d'apprentissage, caractérisées par les coordonnées des n points. Une analyse en composantes principales, sur les données alignées, permet de mettre en évidence la corrélation entre données apprises et ainsi d'obtenir les modes de variation / déformation de la classe de l'objet.

Notons $\mathbf{x_i}$ le vecteur du descripteur correspondant à la i-ème donnée apprise normalisée :

$$\mathbf{x_i} = (x_{i_0}, y_{i_0}, x_{i_1}, y_{i_1}, \ldots, x_{i_{n-1}}, y_{i_{n-1}})^t.$$

À partir des N formes de la base d'apprentissage, nous calculons la forme moyenne notée :

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i}. \tag{2.1}$$

La variabilité inter-données (c.-à-d. des différents points de contrôle) est décrite par la matrice de covariance empirique, Σ, calculée sur les données normalisées et centrées :

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^t. \tag{2.2}$$

Les modes de variation des formes de la classe et leurs amplitudes sont respectivement définis par les vecteurs propres $\mathbf{v_k}$ et les valeurs propres λ_k de $\mathbf{\Sigma}$, solutions de l'équation :

$$\mathbf{\Sigma v_k} = \lambda \mathbf{v_k},$$

avec λ_k la k-ième valeur propre de $\mathbf{\Sigma}$, $\lambda_k \geq \lambda_{k+1}$ et $\mathbf{v_k}^t \mathbf{v_k} = 1$.

L'analyse en composantes principales génère un nouvel espace associé à une base orthogonale (modes ou vecteurs propres) c.-à-d. où les composantes des vecteurs représentent des variables indépendantes. L'espace peut être reconstruit « mode par mode » en

minimisant la perte d'information au sens de l'erreur d'approximation. En supposant que les l premiers modes issus de l'analyse en composantes principales représentent l'essentiel de l'information, l'espace des nouvelles données est un sous-espace de $\mathbb{R}^{2n}$: $\mathbb{R}^l$ où $l << 2n$.

Il est alors possible de générer une nouvelle forme appartenant à la classe d'objet appris à partir de la formule suivante :

$$\mathbf{X} = \bar{\mathbf{X}} + \sum_{k=1}^{l} \beta_k \mathbf{v_k},$$

où β_k peut être vue comme la quantité d'information à prendre en compte pour le mode k.

En faisant l'hypothèse que les données sont réparties suivant une loi normale, nous pouvons dire, d'après la loi normale, que si les paramètres du modèle de forme β_k sont compris entre $-3\sqrt{\lambda_i}$ et $3\sqrt{\lambda_i}$, alors 99% des formes peuvent être reproduites.

A titre d'exemple, la Figure 2.1 montre les résultats obtenus par Cootes [18] sur un modèle de forme entraîné sur une base d'images. Chaque ligne de la figure représente les effets de la variation des paramètres des trois premiers modes. La colonne de visages du centre représente la forme moyenne, les colonnes de gauche et de droite représentent les formes extrêmes obtenue en mettant les paramètres $\beta_1, \beta_2, \beta_3$ successivement égaux à $\pm -3\sqrt{\lambda_1}, \pm -3\sqrt{\lambda_2}, \pm -3\sqrt{\lambda_3}$. Les deux premiers modes semblent encoder les expressions et la morphologie du visage tandis que le dernier encode l'orientation du visage.

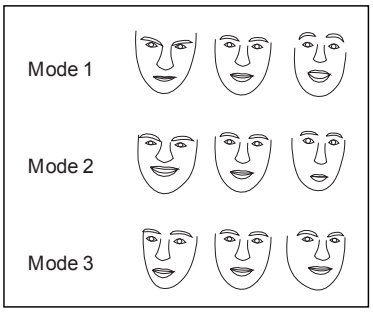


FIGURE 2.1 – Les 3 premiers modes de variation du visage - Source [18]

Bien que les descripteurs par modèles actifs de forme n'utilisent aucune information de l'image, ils restent précis, rapides, simples et s'étendent assez facilement aux formes 3D. Par la suite, pour utiliser plus d'information dans l'image, Cootes *et al.* ont proposé un modèle actif d'apparence (AAM pour *Active Appearance Model* en anglais) [15]. Cette modélisation prend en compte comme précédemment la forme de l'objet mais également son apparence en niveaux de gris dans l'image. L'algorithme est présenté en détail dans [18] et [69]. Nous avons fait le choix de développer la partie sur les modèles actifs de forme (ASM pour *Active Shape Model*) plutôt que celle sur les AAM car dans la suite nous utiliserons cette modélisation.

2.1.2.3 Descripteur de la structure du visage

L'analyse de la structure du visage revient à définir l'organisation des éléments du visage et leur variabilité. Pour cela, nous présentons deux approches. Les *eigenfaces*, une approche globale qui décrit le visage en utilisant une combinaison linéaire de vecteurs propres. La deuxième approche utilise les caractéristiques semblables aux ondelettes de Haar pour décrire par parties les éléments du visage.

***Eigenfaces* :**
La technique de description du visage par *eigenfaces* a été introduite en 1991 par Turk et Pentland [73] dans un contexte de reconnaissance de visages. Comme les modèles actifs de forme, les *eigenfaces* sont construits via une analyse en composantes principales sur les données. L'espace des variables décrivant ces données (comme des vecteurs) est de très grande dimension car tous les pixels d'une image correspondent à des variables. En effet, chaque image I_i de taille $n \times m$ est représentée par un vecteur $\mathbf{x}_i \in \mathbb{R}^{nm}$. Ce vecteur est issu de la concaténation des colonnes de l'image (on parlera de « vectorisation » de l'image). La même mise en équation que pour les modèles actifs de forme (*cf.* Figure 2.1 et 2.2) permet d'obtenir les l vecteurs / visages propres, qui sont les supports des différentes caractéristiques des visages.

pseudo-Harr Les caractéristiques pseudo-Harr doivent leur nom à leur similarité avec les ondelettes de Haar. Elles sont introduites par Viola et Jones [79] pour coder la structure originale et répétitive de la distribution des intensités des objets et en particulier des visages (*cf.* Figure 2.2) dans le but de les détecter dans une image. Les auteurs s'inspirent des travaux de Papageorgiou [60] qui emploie la base de Haar pour la détection d'objet.

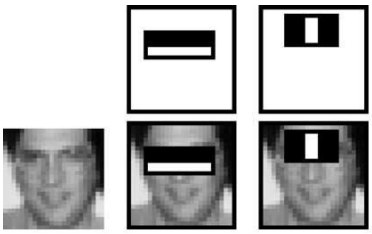


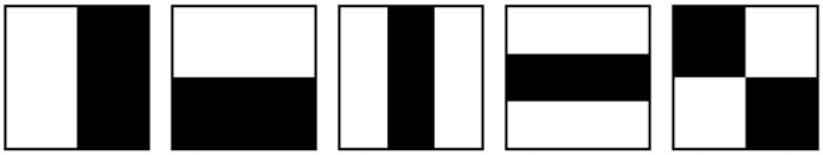
FIGURE 2.2 – Deux caractéristiques révélatrices pour la détection de visage. La première caractéristique mesure la différence d'intensités entre la région des yeux et une région à travers les pommettes. La deuxième caractéristique compare les intensités de la région des yeux à celle sur le nez - Source Viola et Jones [79]

L'avantage primordial d'utiliser des caractéristiques pseudo-Haar (*cf.* Figure 2.3) est la simplicité et la rapidité de calcul. Le descripteur D lié à la caractéristique est calculé en utilisant la différence des sommes des intensités en niveaux de gris, de régions voisines de l'image R^{noir}, R^{blanc}(correspondant respectivement à l'ensemble des pixels noirs ou blancs des caractéristiques pseudo-Haar). Formellement, on écrit :

$$D = \sum_{(x,y)\in R^{\text{noir}}} i(x,y) - \sum_{(x,y)\in R^{\text{blanc}}} i(x,y)$$

Les caractéristiques illustrées sur les Figures 2.3(a) et 2.3(b)) capturent la présence de contours, celles des Figures 2.3(c), 2.3(d) et 2.3(e)) la présence de lignes verticale, horizontale ou diagonale.

Après avoir proposé une version simplifiée des ondelettes de Haar pour améliorer les temps de calcul, Viola et Jones ont utilisé une représentation connue de la communauté de compression d'image, l'image intégrale. Cette image est de même taille que l'image source ; en chaque pixel est assignée la valeur de la somme des intensités des pixels situés au dessus et à gauche. Plus formellement, ils définissent l'image intégrale, notée S, en



(a) (b) (c) (d) (e)

FIGURE 2.3 – Exemples de caractéristiques pseudo-Haar utilisées par Viola et Jones

fonction de l'image source, notée I, de la manière suivante :

$$S(x,y) = \sum_{x' \leq x,\ y' \leq y} I(x',y')$$

Cette méthode permet de réduire le temps de calcul nécessaire associé à chaque caractéristique indépendamment de leur taille. La somme des intensités des pixels situés dans un rectangle $ABCD$ de l'image source peut être calculée à partir de quatre entrées en utilisant l'image intégrale. Plus formellement, cela s'écrit :

$$\sum_{\substack{x_A < x' \leq x_C \\ y_A < y' \leq y_C}} I(x',y') = S(A) + S(C) - S(B) - S(D).$$

Les descripteurs de l'image sont alors obtenus en balayant l'image de manière exhaustive avec le jeu de caractéristiques.

Pour être capable de décrire « correctement » une image, nous aurions besoin d'une grande quantité de descripteurs. Dans la section 2.1.3.2, nous donnons un classement de ces descripteurs, l'idée étant d'en extraire le sous-ensemble le plus petit possible tout en ayant les meilleurs résultats possible de détection de visage.

2.1.3 Post-traitement des descripteurs d'apparence par apprentissage automatique

2.1.3.1 Rôle de la classification

Nous définissons la classification par une fonction h liant un espace d'entrée noté $\mathcal{X}$ à un espace discret de sortie à k classes $\mathcal{Y}$.

$$x \in \mathcal{X} \longrightarrow \boxed{\ h\ } \longrightarrow y = h(x) \in \mathcal{Y}.$$

Dans le cas de la détection du visage, la fonction h est obtenue par une phase d'apprentissage sur les descripteurs définis précédemment. En sortie de la classification nous disposons de deux classes complémentaires : présence ou absence du visage. Comme nous venons de le voir, le problème de classification est sous-jacent à la détection. La détection doit permettre d'analyser des données classifiées dans le but d'étiqueter une nouvelle donnée avec l'étiquette : « présence d'objet » ou « absence d'objet ». Nous pouvons définir des données de classification à différentes échelles, de l'image au pixel. Dans le cas d'un visage, nous pouvons vouloir classer une image comme contenant un visage ou pas. Plus précisément, nous pouvons vouloir classer les régions de l'image et ainsi détecter laquelle contient un visage. Ou bien, nous pouvons souhaiter classer dans l'image, les

pixels appartenant à la classe Peau. Pour répondre à ces questions, les algorithmes se répartissent en trois catégories. Dans le premier cas, nous parlons de classification ou apprentissage supervisé lorsque nous disposons d'un ensemble d'exemples connus étiquetés par un expert. Ici, le rôle du classifieur est de trouver la fonction qui permet d'affecter une bonne étiquette à ces exemples. Dans le deuxième cas, celui de la classification automatique ou non supervisée, le système dispose uniquement d'un ensemble d'exemples mais sans étiquettes associées. L'algorithme de classification doit découvrir par lui-même la structure des données pour éventuellement définir une métrique regroupant les exemples ayant des descripteurs homogènes. Enfin, nous parlons de classification semi-supervisée lorsque nous disposons de données partiellement étiquetées. L'objectif de l'algorithme est trouver la fonction qui permet de classifier au mieux la totalité des données en utilisant les contraintes de classification fournit par les données étiquetées.

2.1.3.2 Principaux algorithmes d'apprentissage

Dans la littérature, il existe une grande quantité d'algorithmes d'apprentissage pour la détection de visage, mais le plus cité est sans doute *AdaBoost*, utilisé par Viola et Jones [78]. Il est à noter que cet algorithme est celui utilisé dans les produits de Fitting-Box ; pour cette raison nous détaillerons les caractéristiques pseudo-Haar et la technique d'apprentissage par boosting. Nous présentons également les séparateurs à vaste marge (SVM pour *Support Vector Machine* en anglais), le classifieur bayésien, et enfin un algorithme non supervisé, l'algorithme des K-moyennes.

***K-moyennes* :**
L'algorithme des K-moyennes, ou *K-means* en anglais, est une méthode de classification non supervisée très simple. Cet algorithme est introduit en 1967 par MacQueen [51], l'idée principale est de classer un ensemble d'éléments en K classes en s'assurant d'une cohérence intra-classes et d'un incohérence inter-classes. Pour cela, l'auteur cherche à minimiser une énergie qui somme sur chaque classe ω_i, contenant N_i éléments, les distances entre chaque élément x_j appartenant à la classe ω_i et le centre de la classe μ_i. Formellement, cela s'écrit :

$$E = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \|x_j - \mu_i\|_A^2$$

Il est courant de choisir une métrique euclidienne, et dans ce cas A correspond à la matrice identité.

La fonctionnelle à minimiser est non linéaire et le problème n'a pas de solution directe. En revanche, face à cet algorithme NP-complet, il est possible d'utiliser une heuristique pour atteindre le minimum de cette fonctionnelle.

Cette heuristique s'articule en quatre étapes :

1. Choisir K éléments représentant les K classes.
2. Assigner à chaque élément x_j la classe ω_i qui minimise la distance $\|x_j - \mu_i\|^2$.
3. Mettre à jour la valeur du centre μ_i de chaque classe ω_i.
4. Boucler en 2) jusqu'à avoir atteint une stabilité.

Ceci ne garantit pas d'atteindre le minimum global et le résultat final dépend de l'initialisation. Comme cette heuristique est en général peu coûteuse en temps de calcul, il est d'usage de minimiser l'énergie à partir de plusieurs initialisations.

K-moyennes et la reconnaissance de visage : [10]

Une faille de cette méthode est que l'on ne peut pas créer des classes non convexes. Pour ce type de problème, nous proposons de regarder la classification par les séparateurs à vaste marge.

Les séparateurs à vaste marge :

Les séparateurs à vaste marge (SVM pour *Support Vector Machine* en anglais), ont été introduites par Cortes et Vapnik [19]. Dans un cadre d'apprentissage supervisé, cette méthode permet de résoudre des problèmes de discrimination non-linéaire à deux classes. Les SVM sont des classifieurs qui reposent sur deux idées clés. La première est l'introduction de la notion de marge maximale qui permet de définir l'hyperplan optimal pour la séparation linéaire des classes. La deuxième idée clé est une généralisation des classifieurs linéaires pour les données non séparables linéairement. L'idée est d'augmenter la dimension des espaces initiaux des variables dans le but de rendre les données linéairement séparables.

Nous disposons d'un jeu de données $(x_1, y_1) \dots (x_n, y_n)$ où $x_i \in \mathbb{R}^p$ est le vecteur de donnée et $y_i = \{-1, 1\}$ son étiquette associée.

La marge est la distance entre l'hyperplan séparateur et les données les plus proches. Nous notons $h(x) = w^t x + w_0 = 0$ l'équation de l'hyperplan séparateur. Nous cherchons $w \in \mathbb{R}^p$ et $w_0 \in \mathbb{R}^p$, les paramètres de l'hyperplan, qui maximisent la marge. Formellement, en supposant le problème séparable linéairement, on cherche :

$$\arg\max_{w,w_0} \min_{k} \{\|x - x_k\| : x \in \mathbb{R}^p,\ w^t x + w_0 = 0\}. \tag{2.3}$$

Retrouver les paramètres de l'hyperplan revient à retrouver quels sont les vecteurs de données qui jouent le rôle de vecteur support.

La solution de l'équation 2.3 s'écrit :

$$h(x) = \sum_{k=1}^{n} \alpha_k y_k (x \cdot x_k) + w_0, \tag{2.4}$$

où α_k permet de sélectionner et pondérer les vecteurs de données qui jouent le rôle de vecteurs de support.

Du fait que l'hyperplan séparateur soit fonction du produit scalaire entre le vecteur de donnée et le vecteur support, nous pouvons envisager de reproduire les calculs pour des données non séparables linéairement. Pour cela, il suffit de trouver une fonction ϕ qui augmente la dimension des vecteurs de données et les plongent dans un nouvel espace, où ils deviennent séparables linéairement. L'équation 2.4 s'écrit alors de la manière suivante :

$$h(x) = \sum_{k=1}^{n} \alpha_k y_k (\phi(x)^t \cdot \phi(x_k)) + w_0. \tag{2.5}$$

Le théorème de Mercer permet d'écrire $K(x, y) = \phi(x)^t \phi(y)$ avec $K(x, y)$ un noyau dont la matrice est continue, symétrique et semi-définie positive. D'où l'hyperplan séparateur en fonction de la fonction noyau :

$$h(x) = \sum_{k=1}^{n} \alpha_k y_k K(x, x_k) + w_0. \tag{2.6}$$

Dans la pratique, nous pourrons prendre un noyau :

– linéaire : $K(x, y) = x \cdot y$,
– polynomial : $K(x, y) = (c + xy)^d$,
– gaussien : $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$.

En raison de leur performance de classification pour différents problèmes, les SVM se sont également introduit dans le domaine de la détection de visage en 1997 par Osuna *et al.* [59].

Boosting :

AdaBoost est un algorithme fondé sur le principe du *boosting* introduit par Freund et Schapir [29] en 1996. Le principe du *boosting* est de combiner un ensemble de petites contributions issues de classifieurs dit faibles pour construire un classifieur fort qui donne des résultats très satisfaisants. Nous appelons classifieur faible un classifieur donnant des résultats au moins aussi satisfaisants que le hasard. La combinaison des classifieurs faibles est pondérée en fonction de leurs résultats de classification : plus le classifieur faible est performant, plus il intervient dans le classifeur fort. La construction du classifieur fort est un processus itératif qui, à chaque étape, ajoute un classifieur faible. Le dernier point du principe de *boosting* est de mettre un poids sur les éléments mal classés pour forcer l'algorithme à les classer correctement par le prochain apprenant faible. Ce processus de mise en avant des éléments mal classés donne son nom à la méthode.

AdaBoost, qui signifie *boosting* adaptatif, apprend un classifieur fort $h(x)$ à partir du jeu de données étiquetées $(x_1, y_1), \ldots, (x_n, y_n)$ où x_i est la donnée (une image pour le cas de la détection du visage) et $y_i = \{-1, 1\}$ les étiquettes associées. Lorsque $y_i = -1$ l'objet est absent dans l'image x_i et lorsque $y_i = 1$ l'objet est présent.

Le classifieur fort $h(x)$ est défini par une combinaison linéaire de T classifieurs faibles $h_t(x)$ de la manière suivante :

$$h(x) = \operatorname{sgn}\left(\sum_{j=1}^{M} C_j h_j(x)\right) \tag{2.7}$$

C_t est une constante permettant de pondérer l'importance des classifieurs faibles par rapport à leur performance (plus le $h_t(x)$ est performant plus C_t sera important). La fonction $\operatorname{sgn}(a)$ extrait le signe du nombre réel a. Un classifieur est performant si l'erreur de classification ϵ_t, définie à l'équation 2.8, tend vers 0. Le choix des classifieurs faibles est obtenu de manière itérative en minimisant l'erreur de classification

$$\min_{h_t}(\epsilon_t)$$

où

$$\epsilon_t = \sum_{i=1}^{n} \omega_i |h_t(x_i) - y_i|. \tag{2.8}$$

h_t est fonction d'une image (x) et est paramétré par une caractéristique (f), un seuil (θ) et une parité (p) qui permet de fixer le sens de l'inégalité de l'équation 2.9. ω_i est le poids associé au couple (x_i, y_i) permettant ou pas de les classer en priorité.

$$h_t(x; f, \theta, p) = \begin{cases} 1 & \text{si} \quad pf(x) > p\theta \\ -1 & \text{sinon} \end{cases} \tag{2.9}$$

Pour simplifier les notations, nous écrivons $h_t(x)$.

Algorithme d'AdaBoost

Données $(x_1, y_1) \dots (x_n, y_n)$

Initialisation $\omega_i = \frac{1}{n}$ pour $i = 1 \dots n$

Pour $t = 1 \dots T$

1. Recherche du classifieur faible le plus performant :

$h_t = \min_{h_k} \left(\sum_{i=1}^{n} \omega_i \, \mathbf{1}_{(h_k(x_i) \neq y_i)} \right)$

Qui s'écrit également :

$h_t = \min_{f,p,\theta} \left(\sum_{i=1}^{n} \omega_i \, \mathbf{1}_{(h_k(x_i,f,p,\theta) \neq y_i)} \right)$

C'est l'algorithme d'entraînement (expliqué par la suite) qui permet de trouver les paramètres optimaux du classifieur faible.

2. Mise à jour de ω_i pour les éléments bien classés et normalisation

$\omega_i \leftarrow \omega_i \times \frac{\epsilon_t}{1-\epsilon_t}$ et $\omega_i \leftarrow \frac{\omega_i}{\sum_j \omega_j}$

3. Calcul du poids C_t

$C_t = \min_{C} \left(\sum_{i=1}^{n} \exp\left(-y_i \left(C h_t(x_i) + g_{t-1}(x_i)\right)\right)\right)$

avec $g_{t-1}(x_i) = \sum_{r=1}^{t-1} C_r h_r(x_i)$

$C_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

Résultat $h(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} C_t h_t(x)\right)$

L'algorithme d'entraînement est un algorithme permettant pour chaque caractéristique f de connaître la valeur du seuil θ et de la parité p. Pour cela on étudie en parallèle les deux cas de parité possibles $p^1 = 1$ et $p^2 = -1$ auxquels on associe respectivement θ^1 et θ^2. Nous rappelons que l'objectif est de maximiser les bonnes détections et minimiser les mauvaises.

On note $F_{b,f}(\theta)$, la fonction qui compte la quantité d'information bien détectée pour la caractéristique f en fonction de la valeur de θ. Formellement, on écrit :

$$F_{b,f}(\theta) = \sum_{i=1}^{n} \omega_i \, I(x_i, y_i, \theta) \tag{2.10}$$

$$\text{Avec } I(x_i, y_i, \theta) = \begin{cases} 1 & \text{si} \quad f(x_i) < \theta \text{ et } y_i = 1 \\ 0 & \text{sinon} \end{cases} \tag{2.11}$$

De la même façon, on note $F_{m,f}(\theta)$, la fonction qui compte la quantité d'information mal détectée pour la caractéristique f en fonction de la valeur de θ. Formellement, on écrit :

$$F_{m,f}(\theta) = \sum_{i=1}^{n} \omega_i \, I(x_i, y_i, \theta) \tag{2.12}$$

$$\text{Avec } I(x_i, y_i, \theta) = \begin{cases} 1 & \text{si} \quad f(x_i) < \theta \text{ et } y_i = -1 \\ 0 & \text{sinon} \end{cases} \tag{2.13}$$

θ^1 est obtenu en minimisant la fonction $c^1 = 1 - F_{b,f}(\theta^1) + F_{m,f}(\theta^1)$ et θ^2 est obtenu en minimisant la fonction $c^2 = 1 - F_{m,f}(\theta^2) + F_{b,f}(\theta^2)$. Enfin, on calcule l'erreur de classification pour les deux valeurs de θ. Le couple (θ^i, p^i) qui fournit la plus petite erreur donne les paramètres manquants du classifieur faible.

À ce stade, nous sommes en mesure de construire le classifieur $h(x)$ défini à l'équation 2.7, à partir de l'algorithme d'AdaBoost. Viola et Jones améliorent la détection en calculant plusieurs classifieurs forts et en les organisant en cascade. Cette amélioration permet d'éliminer rapidement dans la cascade les régions de l'image qui ne contiennent pas de visage. Ainsi les taux de détection restent élevés, la détection est plus rapide, puisqu'elle est exécutée en temps réel. En revanche, dès qu'une image est rejetée à l'issue d'une étape de la cascade alors cette région ne pourra pas être ré-injectée dans l'algorithme.

Classifieur Bayesien :

L'apprentissage statistique fait partie des apprentissages supervisés. À partir d'un jeu de caractéristiques, nous cherchons à estimer la probabilité d'apparition d'un élément. Ici, nous nous intéressons plus particulièrement à apprendre la statistique des données pour la réinjecter dans un classifieur bayésien. Ce dernier chercher à maximiser la probabilité *a posteriori* d'appartenance à une classe. Formellement, nous écrivons :

$$\omega_i = \underset{\omega_i}{argmax}(P(\omega_i|X)) \quad \text{avec} \quad P(\omega_i|X) = \frac{p(X|\omega_i)P(\omega_i)}{P(X)}. \tag{2.14}$$

$P(\omega_i)$ est la probabilité *a priori* de la classe ω_i. $P(X)$ est l'évidence, nous remarquons que ce terme est une constante de normalisation qui n'influe pas la décision du classifieur. $p(X|\omega_i)$ est la vraisemblance, c'est à ce niveau que les données d'apprentissage interviennent, ce terme encode la statistique des données au sein de la classe ω_i.

Liu [49] présente un cas de détection du visage par classification bayésienne, en discriminant la classe Visage, notée ω_1, et la classe ¬Visage, notée ω_0. Pour cela, Liu propose d'exprimer la vraisemblance des deux classes par une fonction de densité de probabilité. S'il est envisageable de modéliser la distribution de la classe Visage par une loi normale multi-dimensionnelle, cela l'est beaucoup moins pour la classe ¬visage. Cette classe est trop variée et complexe pour qu'une loi normale soit adaptée. Lui décide de réduire le support de la classe en modélisant par une loi normale la distribution d'un sous-ensemble des éléments de la classe. Il choisit ces éléments parmi ceux qui sont les plus proche possibles de la classe Visage.

Formellement, la vraisemblance des classes Visage et ¬Visage s'écrivent :

$$p(X|\omega_1) = \frac{1}{(2\pi)^{N/2}|\Sigma_1|^{1/2}} \exp\left(-0.5(X - M_1)^t \Sigma_1^{-1}(X - M_1)\right) \tag{2.15}$$

et

$$p(X|\omega_0) = \frac{1}{(2\pi)^{N/2}|\Sigma_0|^{1/2}} \exp\left(-0.5(X - M_0)^t \Sigma_0^{-1}(X - M_0)\right) \tag{2.16}$$

où $M_1 \in \mathbb{R}^N$ et $\Sigma_1 \in \mathbb{R}^{N\times N}$ sont la moyenne et la matrice de covariance des données étiquetés Visage et M_0 et Σ_0 celles des données étiquetés ¬Visage.

Le théorème de Bayes nous permet de relier la vraisemblance à la probabilité *a posteriori* :

$$P(\omega_1|X) = \frac{p(X|\omega_1)P(\omega_1)}{P(X)}, \text{ et } P(\omega_0|X) = \frac{p(X|\omega_0)P(\omega_0)}{P(X)} \tag{2.17}$$

Finalement, l'étiquette du vecteur de donnée X est attribué de la manière suivante :

$$X \in \begin{cases} \omega_1 & \text{si } P(\omega_1|X) > P(\omega_0|X) \text{ et } P(\omega_1|X) > \theta \\ \omega_0 & \text{sinon} \end{cases} \tag{2.18}$$

L'introduction du paramètre θ et de l'inégalité $P(\omega_1|X) > \theta$ permet de rendre robuste les résultats en supprimant les cas de fausses détections où la probabilité d'avoir un visage est faible mais moins faible que celle de ne pas avoir de visage, avec on le rappelle une classe "non visage" modélisée sur des données tronquées. Outre le fait que cette technique de détection donne de très bons résultats, elle nous a semblé très intéressante pour l'utilisation du classifieur bayésien que nous aussi nous utiliserons par la suite à des fins de segmentation.

2.1.4 Conclusion

Dans cet état de l'art, nous avons développé les contributions qui nous semblaient à la fois les plus remarquables et les plus complémentaires pour la détection de visage. Il est marquant de voir les techniques de détection des visages de plus en plus utilisées dans les applications industrielles. Cependant, la détection de visage reste une très tâche difficile, notamment en raison de l'importante variabilité de l'information de pose et d'éclairage. Par conséquent, nous croyons qu'il y a encore beaucoup de travaux qui peuvent être fait pour améliorer encore les performances. La direction la plus simple est de continuer d'améliorer l'algorithme d'apprentissage et les descripteurs. Les caractéristiques de Haar utilisées dans les travaux de Viola et Jones [79] sont très simples et efficaces pour la détection de visage « frontal », mais ils sont moins adaptés pour les visages dont la pose est arbitraire. Nous pouvons penser que des caractéristiques riches améliorent les performances du détecteur. Une autre idée très intéressante est de considérer l'information contextuelle. Les visages sont liés à d'autres parties du corps, la prise en compte des autres parties du corps dans la caractérisation peut rendre le problème plus robuste et améliorer la détection de visages. Comme nous l'avons dit, la détection de visage est un domaine très actif, nous n'avons alors pas pu développer toutes les approches mais si vous souhaitez plus d'information à ce sujet les articles de Yang et *al.* et surtout de Zhang et *al* [87, 89] proposent une revue sur le sujet.

2.2 De la détection du visage à la segmentation d'image Portrait

2.2.1 Différents problèmes

Segmenter un portrait numérique peut consister à segmenter la peau, segmenter les cheveux et/ou segmenter le fond d'une image. Ces trois segmentations révèlent un ensemble de problématiques. Par exemple, une des difficultés à laquelle se confronte la segmentation de la peau est que le résultat doit être indépendant des origines ethniques de la personne. Une autre difficulté repose sur la définition de la peau, nous pouvons nous demander si les lèvres ou bien les oreilles en font partie. En ce qui concerne la segmentation des cheveux, on recense un autre type de problème. Il est en effet difficile de tracer la frontière entre la région Cheveux et la région ¬Cheveux. En fait, cela est dû au fait

que bien souvent, il n'existe pas de frontière franche entre la région Cheveux et la région ¬Cheveux, mais une zone de transition dans laquelle nous pouvons voir le mélange des deux régions. Enfin, la difficulté de segmenter le fond d'un portrait numérique est qu'il peut prendre tellement d'apparences différentes que la modélisation en devient très complexe. Tous ces exemples montrent que la segmentation de portrait numérique n'est pas une tâche facile à réaliser. D'ailleurs aujourd'hui, nous trouvons peu d'articles dans la littérature qui traitent ce sujet. Nous avons choisi de vous les présenter dans la section suivante.

2.2.2 Descripteurs et techniques de segmentation pour les portraits numériques

2.2.2.1 La référence pour la segmentation des cheveux : une analyse de la couleur

L'article de référence, précurseur dans le domaine de la segmentation des cheveux est très récent puisqu'il ne date que de 2006. L'existence de quelques études antérieures [38, 50] montre l'intérêt de la communauté pour ce problème mais met aussi en évidence le fait que la tâche à réaliser n'est pas triviale. Yacoob et Davis, dans [85], présentent un algorithme de segmentation des cheveux dans le but d'extraire certaines propriétés et ainsi de proposer une nouvelle approche pour la reconnaissance de personnes. Pour leur étude, les auteurs se limitent aux portraits numériques où la personne est de face.

Pour la segmentation, les auteurs utilisent le contexte de l'image pour détecter des zones rectangulaires contenant uniquement des cheveux et ainsi pour apprendre leurs caractéristiques colorimétriques.

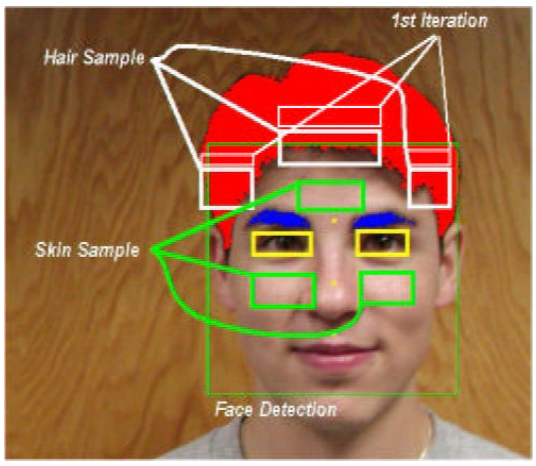


FIGURE 2.4 – Localisation des zones échantillonnées sur la peau et les cheveux pour les caractériser - Source [85]

Pour cela, ils développent un algorithme en quatre étapes :

- **La détection du visage :** afin de réduire la zone de recherche des cheveux dans l'image, la première étape est d'utiliser des algorithmes existants de détection du visage. Ils emploient l'algorithme de Viola et Jones [79] que nous venons de décrire dans la section 2.1.3.2.
- **La détection des yeux :** par la même méthode de détection que les visages, ils détectent la position des yeux à l'intérieur du visage. Cette étape leur permet de

normaliser les images et ainsi de pouvoir comparer entre elles, les propriétés des cheveux.

– **La segmentation de la peau :** la détection des yeux permet de sélectionner trois zones rectangulaires de peau (une sur le front et deux sous les yeux *cf.* Figure 2.4) sur le visage. Ces trois zones permettent de définir les paramètres du modèle de la distribution de la peau. Le modèle choisi est celui présenté par Horprasert *et al.* [33], il est exprimé dans l'espace RVB et il permet de restreindre la distorsion de la couleur tout en modélisant les variations de lumière.
– **La segmentation des cheveux :** la position des yeux, le masque de segmentation de la peau sont d'autant de connaissances *a priori* utiles pour la segmentation des cheveux. Les auteurs font l'hypothèse réaliste qu'au dessus, à droite ou à gauche de la région Peau se trouvent une (ou des) région(s) Cheveux ; ils définissent ainsi trois nouvelles zones rectangulaires (les rectangles blancs sur la Figure 2.4). Pour chaque rectangle, les modèles colorimétriques sont calculés, en retirant les pixels trop proche du modèle colorimétrique de la peau. Si la distance entre les trois modèles est faible alors un modèle global est recalculé, sinon le modèle correspondant à la zone au dessus de la peau est pris comme référence. À chaque itération les zones rectangulaires sont étendues et si le modèle courant est proche du modèle de référence alors il devient à son tour modèle de référence et on réitère. Sinon la segmentation des cheveux est calculée en utilisant le modèle colorimétrique de référence.

(a) (b) (c) (d)

FIGURE 2.5 – Quatre résultats de segmentation des cheveux : deux corrects et deux erronés - Source [85]

L'algorithme de segmentation des cheveux de Yacoob et Davis fournit les résultats de la Figure 2.5. Les images 2.5(a) et 2.5(b) montrent que l'algorithme est capable de segmenter aussi bien des cheveux courts que des cheveux longs. En revanche, l'image 2.5(c) montre que modèle colorimétrique choisi pour les cheveux ne peut pas prendre en compte les chevelures ayant plusieurs teintes (ici le noir et le rouge). L'image 2.5(d) montre une autre faiblesse, les vêtements peuvent être vus comme des cheveux si leur teinte est trop proche. Face à ces résultats, nous pouvons envisager que la texture apporte l'information qu'il nous manque.

Dans l'ensemble, cette publication présente des résultats très encourageants. Il est tout de même dommage que l'évaluation des résultats soit faite visuellement par un opérateur.

2.2.2.2 La segmentation des cheveux : une analyse de la couleur et de la fréquence

En 2008, Rousset et Coulon dans [65] proposent de combiner l'information fréquentielle et colorimétrique de l'image pour segmenter les cheveux. Tout comme Yacoob et Davis, ils initialisent leur système en définissant la partie de l'image qui contient le visage par le détecteur de Viola et Jones [79]. Ensuite, trois étapes sont nécessaires avant de produire le résultat de leur segmentation

– **Analyse fréquentielle :** les cheveux ayant une texture particulière, les auteurs proposent un filtrage fréquentiel de l'image qui permet de révéler la texture des cheveux. Ils définissent un filtre isotrope passe-bande. L'isotropie du filtre permet de s'affranchit de l'information d'orientation, et la fonction passe-bande ne sélectionne qu'une plage de fréquence précise. Les composantes continues (les basses fréquences) et le bruit (les hautes fréquences) sont ainsi éliminés. Le filtre est défini selon l'équation suivante :

$$G_{(f_0,\sigma)} = \int_{\theta=0}^{360} \exp\left(-\frac{(f_\theta - f_0)^2}{2\sigma^2}\right). \tag{2.19}$$

où σ à le rôle de bande passante et f_0 de fréquence centrale. Après avoir appliqué le filtre à l'image, un simple seuillage donne le résultat de segmentation de l'analyse fréquentielle :

$$\text{Seg\_frequence}(i,j) = \begin{cases} 1 & \text{si } \left(I * G_{(f_0,\sigma)}\right)(i,j) \leq \mu - \sigma \\ 0 & \text{sinon} \end{cases} \tag{2.20}$$

où μ et σ sont respectivement la moyenne et l'écart type de la l'image filtrée.

– **Analyse colorimétrique :** l'information apportée par la couleur semble être adaptée à la segmentation de cheveux. La difficulté est d'être capable de la modéliser précisément. Pour cela, Rousset et Coulon font l'hypothèse que la personne dans un portrait numérique a toujours des cheveux sur le dessus du crâne et y définissent une zone rectangulaire. À partir de la localisation de cette zone rectangulaire et en utilisant les résultats de segmentation obtenus par l'analyse fréquentielle, les auteurs calculent l'information colorimétrique des cheveux, exprimée dans l'espace YCbCr. À partir de ces données, ils calculent la moyenne, $\mu = (\mu_y, \mu_{Cb}, \mu_{Cr})$, et l'écart type, $\gamma = (\gamma_y, \gamma_{Cb}, \gamma_{Cr})$. La segmentation issue de l'analyse colorimétrique est obtenue en appliquant le seuillage suivant :

$$\text{Seg\_couleur}(i,j) = \begin{cases} 1 & \text{si } \begin{cases} I_{(Y,Cb,Cr)}(i,j) \geq \mu_{(Y,Cb,Cr)} - \gamma_{(Y,Cb,Cr)} \\ I_{(Y,Cb,Cr)}(i,j) \leq \mu_{(Y,Cb,Cr)} + \gamma_{(Y,Cb,Cr)} \end{cases} \\ 0 & \text{sinon} \end{cases} \tag{2.21}$$

– **Combinaison de la segmentation par la couleur et par la fréquence :** après avoir obtenu deux cartes de segmentation, l'une par une analyse sur la fréquence et l'autre par une analyse sur la couleur, ils décident de combiner les informations de la manière donnée dans Table 2.1, où 1 signifie que nous sommes en présence de cheveux et 0 dans le cas contraire. Pour les résultats $\in$ [0,1], cela signifie que la segmentation est indécise. Les auteurs proposent d'utiliser la technique de *matting* développée par Levin *et al.* [47] pour obtenir une segmentation « lisse » des cheveux.

(a) (b) (c) (d)

FIGURE 2.6 – Quatre résultats de segmentation des cheveux : deux corrects et deux erronés - Source [65]

L'algorithme de segmentation des cheveux de Rousset et Coulon fournit les résultats de la Figure 2.6. Les images 2.5(a) et 2.5(b) montrent que l'algorithme est capable de segmenter des cheveux longs, des coupes de cheveux ayant des composantes non connexes comme dans la figure 2.6(b). Par contre, les images 2.6(c) et 2.6(d) montrent des erreurs des segmentation, mais au vu des résultats l'analyse fréquentielle à semble efficace notamment pour les partie de l'image ayant des textures remarquables.

2.2.2.3 La segmentation de la peau, des cheveux et du fond : une analyse de la couleur et un *a priori* de localisation

En 2008, Lee *et al.* [45] proposent un algorithme de reconnaissance de coupe de cheveux. Pour cela, ils développent une méthode de segmentation de portrait numérique en trois classes : Peau, Cheveux et Fond. Leur stratégie de segmentation est fondée sur l'utilisation de l'information colorimétrique, de l'utilisation d'un *a priori* de localisation et d'un lissage de la carte de segmentation. L'algorithme de segmentation se scinde en une phase hors ligne et une phase en ligne. La phase hors ligne est une phase d'apprentissage ayant pour but de définir certains *a priori* pour les classes à partir d'un jeu de données d'apprentissage.

- **L'*a priori* de localisation :** Les coupes de cheveux ayant une grande variabilité de forme, Lee *et al.* décident de construire six modèles de coupe de cheveux (courtes, longues, volumineuse, etc.). Pour chacun d'eux, à partir d'un ensemble de segmentations effectuées manuellement, ils apprennent les probabilités *a priori* de localisation $P(\omega|C^i)$ avec $i = \{p, f, c\}$ pour les classes Peau, Cheveux et Fond.
- **L'*a priori* colorimétrique de la classe Cheveux $P(I(\omega)|C^c)$:** en plus d'avoir une grande variabilité de forme, les coupes de cheveux possèdent une grande variabilité de teinte. Les auteurs proposent de séparer la classe Cheveux en 5 catégories qui correspondent aux cheveux blonds, châtains, bruns, roux et gris. Pour chaque catégorie, ils modélisent la distribution par un modèle de mélange de gaussiennes.
- **L'*a priori* colorimétrique de la classe peau $P(I(\omega)|C^p)$:** la couleur de la peau

Seg_frequence	Seg_couleur	Seg_finale
1	1	1
0	0	0
0	1	$\in$ [0,1]
1	0	$\in$ [0,1]

TABLE 2.1 – Résultat de segmentation finale en fonction des segmentations issues des analyse fréquentielle et colorimétrique

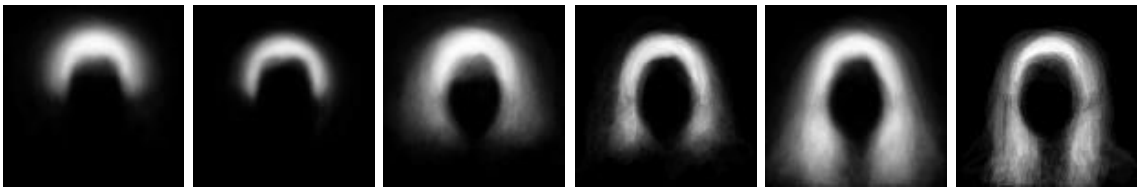


FIGURE 2.7 – Probabilité *a priori* de localisation des cheveux - Source [45]

est elle aussi modélisée par un modèle de mélange de gaussiennes. Le nombre de gaussiennes est fixé à cinq.

La deuxième partie de l'analyse de la distribution couleur dans les portraits numériques se fait donc en ligne, elle permet d'affiner les modèles définis précédemment.

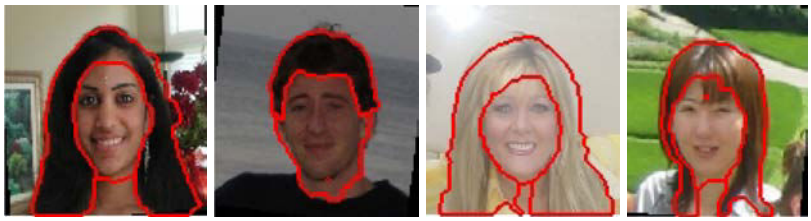
Les cartes de probabilité *a priori* sur la position $P(\omega|C^i)$ permettent de définir dans l'image trois zones, chacune d'elle ayant une forte probabilité d'être de la peau, des cheveux et du fond.

- **Initialisation :** À partir des cartes de probabilité, $P(\omega|C^i)$ avec $i = \{p, f, c\}$, de localisation des différentes classes, on récupère pour chacune d'elles un échantillon. Les échantillons de peau et de fond sont un à un mis en relation avec leurs modèles colorimétriques. Pour la peau, tout échantillon n'appartenant pas au modèle n'est plus pris en compte. Pour les cheveux, un calcul par maximum de vraisemblance permet de fixer la catégorie de cheveux qui entre en jeu. Ensuite tout comme pour la peau, on ne prendra en compte que les échantillons de cheveux appartenant aux modèles de mélange de gaussiennes appris dans la phase hors ligne.
- **Mise à jour des modèles colorimétriques :** À partir des échantillons de peau, de fond et de cheveux définis précédemment, les auteurs apprennent en ligne les paramètres des trois modèles de mélange de gaussiennes.
- **Segmentation :** Les algorithmes de segmentation sont *Graph-cut* et *Loopy Belief Propagation*, dans les deux cas, la fonctionnelle sur laquelle il faut agir prend en compte l'*a priori* sur la couleur et sur la position, mais aussi un terme de lissage pour l'étiquetage des classes. Le terme de lissage vise à pénaliser la fonctionnelle lorsque deux pixels voisins sont étiquetés différemment. Le poids de pénalisation est fonction de la distance dans l'espace de couleur entre les deux pixels. Il en résulte que les deux algorithmes produisent des résultats très intéressants avec une petite préférence pour celui qui utilise *Graph-Cut* car le taux d'erreur de pixels mal classée est la plus faible. Cet article reste flou sur le choix *a priori* de $P(\omega|C^i)$.
  Dans le cas de l'algorithme *Graph-Cut*, la fonctionnelle à minimiser s'écrit :

$$\begin{aligned} E(C) &= \left(\sum_{\omega} -\log P\left(I(\omega)|C(\omega)\right) - \beta \log P\left(\omega|C(\omega)\right) \right) \\ &+ \alpha \sum_{\substack{\omega_i\omega_j \\ \omega_j \in \mathcal{N}(\omega_i)}} \delta\left(C(\omega_i) \neq C(\omega_j)\right) \exp\left(-\gamma \|I(\omega_i) - I(\omega_j)\|^2\right) \end{aligned} \tag{2.22}$$

- **Itération :** Les auteurs itèrent les étapes de mise à jour des modèles colorimétriques et de segmentation jusqu'à convergence du système.

L'algorithme de segmentation des cheveux de Lee *et al.* fournit les résultats de la Figure 2.6. Les images 2.8(a) et 2.8(b) montrent que l'algorithme est capable de segmenter des portraits avec cheveux longs, des portraits dont la teinte des cheveux est proche



(a) (b) (c) (d)

FIGURE 2.8 – Quatre résultats de segmentation des cheveux : deux corrects et deux erronés - Source [45]

de la teinte des vêtements lorsque les deux régions ne se rencontrent pas. En revanche, l'image 2.8(c) montre une mauvaise segmentation car des pixels représentant les vêtements viennent s'insérer dans la région correspondant à la segmentation des cheveux. Colorimétriquement les cheveux blonds et le vêtement jaune ne sont pas assez discriminant et le terme de localisation n'a aucune influence car les éléments sont voisins. Le deuxième cas de fausse segmentation 2.8(d) apparaît au niveau de la frange. Certains pixels étant des cheveux appartiennent à la région peau. Colorimétriquement, les deux classes peau et cheveux ne sont pas semblables. L'erreur est sûrement due au fait que l'*a priori* de localisation ne doit pas assez prendre en compte de coupe de cheveux avec la frange et donc le modèle force les pixels du front à appartenir à la classe peau.

L'*a priori* de localisation semble produire des résultats de segmentation en limitant les cas aberrants de segmentation et le terme de régularisation permet de lisser le résultat de segmentation.

Wang *et al.* [81], proposent un algorithme très proche de celui énoncé précédemment. L'objectif est de segmenter les cheveux d'un portrait numérique. Ils se fondent sur la construction d'un *a priori* de localisation, sur l'utilisation d'un modèle colorimétrique pour décrire les cheveux et enfin sur une régularisation des résultats de segmentation, effectuée en employant l'algorithme *Mean-Shift*.

2.2.2.4 La segmentation de la peau, des cheveux, des vêtement et du fond :

Dans [82], Wang *et al.* proposent une nouvelle approche pour la segmentation des portraits numériques en quatre classes : Peau, Cheveux, Vêtement et Fond. Cette approche est fondée sur la construction d'un modèle par combinaison d'exemples (en anglais *Compositional Exemplar-based Model* CEM). Elle se scinde en trois étapes regroupées dans deux phases :

- **le mode hors ligne** permet de construire le CEM ;
- **le mode en ligne** se décompose en deux parties :
  - l'exploitation du CEM pour obtenir pour chaque classe un masque de probabilité,
  - l'exploitation des masques de probabilité et l'utilisation de plusieurs résolutions pour la segmentation.

La construction du CEM : Il est fondé sur l'utilisation d'une base d'apprentissage comprenant n couples (I, S) avec I une image Portrait et S sa segmentation définie manuellement par un expert. Pour chaque couple (I, S) est défini q échantillons $(e_1, e_2, ..., e_q)$. Chaque échantillon e_i contient une zone de recouvrement avec ses échantillons voisins e_j avec $j \in \mathcal{N}(i)$. La construction du CEM s'appuie sur l'algorithme de *RankBoost* [28]. Cet

algorithme est très proche de celui *AdaBoost* dont nous venons de parler dans la section 2.1.3.2. Il permet de construire un classement fort, noté H, à partir de la combinaison linéaire de classements faibles. Ce classement est capable de trier par les échantillons en fonction de leur ressemblance.

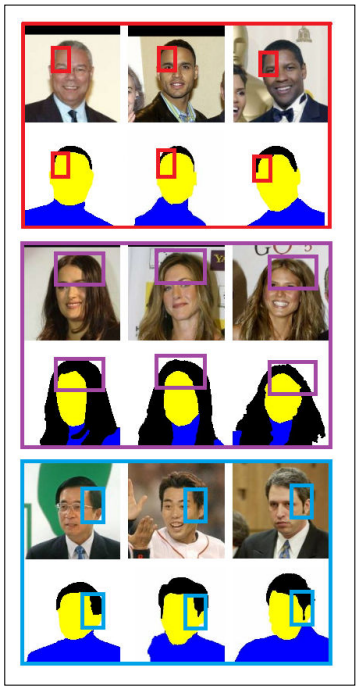


FIGURE 2.9 – Exemple d'apprentissage de trois types d'échantillons.

Les masques de probabilité : L'objectif est de trouver pour une image Portrait les masques de probabilité de chaque classe en utilisant le modèle par combinaison d'exemples. On cherche les q échantillons $(e_1^{k_1}, e_2^{k_2}, ..., e_q^{k_q})$, avec $k_i \in [1...n]$, contenus dans la base d'apprentissage qui s'approche le plus des q échantillons de l'image traitée $(f_1, f_2, ..., f_q)$. Les auteurs proposent de minimiser l'énergie suivante :

$$E(e_1^{k_1}, e_2^{k_2}, ..., e_q^{k_q}) = \sum_{i=1}^{q} \left(\phi_i(k_i) + \sum_{j \in \mathcal{N}(i)} \phi_{i,j}(k_i, k_j) \right) \tag{2.23}$$

$$\phi_i(k_i) = -\log(H(f_i, e_i^{k_i}))$$
$$\phi_{i,j}(k_i, k_j) = -\log(C_A(e_i^{k_i}, e_j^{k_j}))$$

Le terme $\phi_i(k_i)$ recherche dans la base d'apprentissage le i-ème échantillon le plus semblable. Le terme $\phi_{i,j}(k_i, k_j)$ est un terme de régularisation qui fait intervenir la notion de voisinage. Il pénalise les échantillons ayant des étiquettes différentes sur les zones de recouvrement.

La minimisation de cette énergie nous fournit pour les q échantillons de l'image traitée, q échantillons de la base d'apprentissage pour lesquels on connait les étiquettes de chaque pixel. Le masque de probabilité de chaque classe est défini pixel à pixel en calculant le rapport entre le nombre d'étiquettes de la classe sur le nombre total d'étiquette.

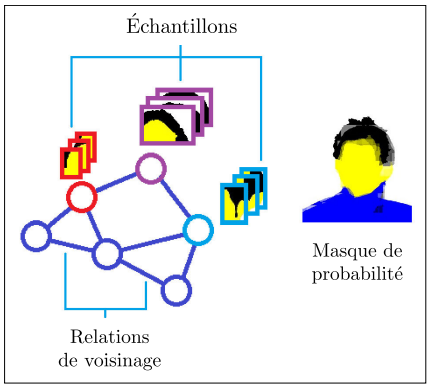


FIGURE 2.10 – Construction du masque de probabilité à partir du CEM.

La segmentation : Elle est obtenue en exploitant deux niveaux de résolutions (*cf.* Figure 2.11) et la notion de voisinage.

$$E(S) = \sum_i \left(\Phi_1(i, S(i)) + \sum_{j \in \mathcal{N}(i)} \Phi_2(i, j, S(i), S(j)) + \widetilde{\Phi}_3(i, S(i)) + \sum_{j \in \mathcal{R}(i)} \widetilde{\Phi}_4(i, j, S(i), S(j)) \right) \tag{2.24}$$

Les deux premiers termes de l'énergie 2.24 travaillent sur la grille de pixels tandis que les deux derniers travaillent au niveau des super-pixels (regroupement de pixels en petite régions ayant dans ce cas les même caractéristiques colorimétriques). Φ_1 et $\widetilde{\Phi}_3$ sont des termes d'attache aux données, quant à Φ_2 et $\widetilde{\Phi}_4$, ils régularisent la solution. $\Phi_1(i, S(i))$ cherche pour chaque pixel i sa classe $S(i) \in \{$ *peau, cheveux, vêtement ou fond* $\}$ en prenant en compte le masque de probabilité (calculé à l'étape précédente) et la probabilité liée à la couleur. $\Phi_2(i, j, S(i), S(j))$ est un terme qui intervient uniquement lorsque deux pixels voisins i et j sont étiquetés différemment $S(i) \neq S(j)$. Dans ce cas là, plus la distance dans l'espace de couleur RVB entre les deux pixels est grande moins ce terme pénalise l'énergie. $\widetilde{\Phi}_3$ est semblable à Φ_1 à l'exception prêt que l'on rend uniforme les valeurs du masque de probabilité et de la probabilité de la couleur la moyenne sur le super-pixel. $\widetilde{\Phi}_4$ intervient uniquement lorsque deux pixels d'un même super-pixel ont des étiquettes différentes. La minimisation de cette énergie est effectué par l'algorithme d'α-expansion.

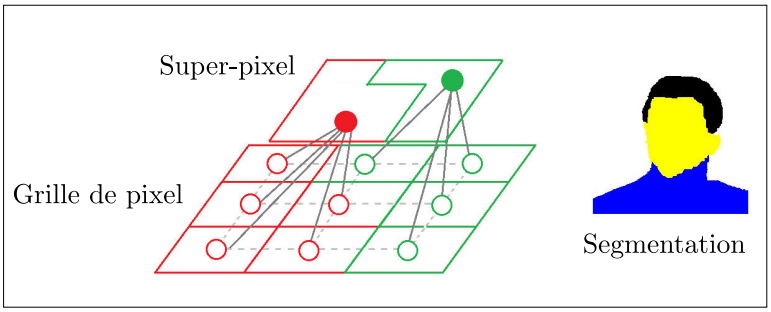
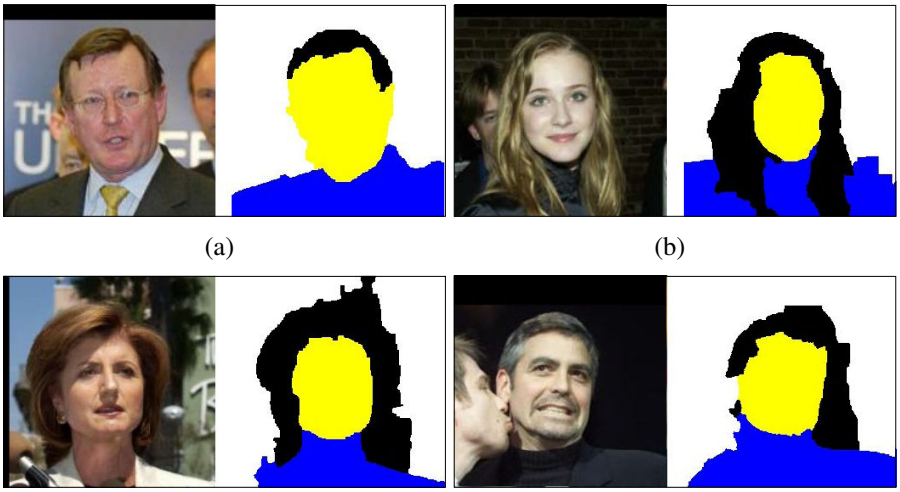


FIGURE 2.11 – Résultat de segmentation en utilisant deux niveaux de résolution de l'image.

La figure 2.12 montre les résultats de la segmentation de l'algorithme de Wang *et al.*. Les deux premières images, 2.12(a) et 2.12(b), montrent des cas où la segmentation donne de bons résultats. Nous remarquons que les images à traiter sont difficiles car l'arrière plan est très variable, il peut contenir d'autres personnes. Dans le cas des mauvaises segmen-

tation illustrées sur les images 2.12(c) et 2.12(d) c'est la frontière entre la classe cheveux et la classe fond qui donne le plus de difficultés. Dans les résultats de l'article, nous remarquons que la zone correspondant à la peau semble s'arrêter au niveau du menton, les pixels inférieurs sont apparemment forcé à appartenir à la classe vêtement.



(a) (b)

(c) (d)

FIGURE 2.12 – Quatre résultats de segmentation des cheveux : deux corrects et deux erronés - Source [82]

Les résultats de segmentation de ce papier sont très prometteurs mais le discours n'est pas très clair, les auteurs passent trop rapidement sur les étapes clés de leur algorithme.

2.2.2.5 Conclusion

Le sujet d'analyse des cheveux, de segmentation des cheveux [48, 50, 54, 68, 70, 90], est un thème de recherche très actif actuellement et qui cache encore beaucoup de secrets. Les publications détaillées précédemment commencent à présenter des résultats satisfaisants. Nous notons que pour segmenter les cheveux, les auteurs prennent en compte les éléments de l'environnement comme la peau, le fond. De plus la notion d'*a priori* apporte une robustesse aux résultats tout comme le fait de prendre en compte le voisinage lors de la segmentation. Nous allons donc exploiter ses remarques et utiliser les méthodes variationnelles couramment utiliser pour la segmentation d'image médicale. Une mise en équation avec une représentation implicite de la courbe permet d'expliquer des contours complexes comme ceux des cheveux.

2.2.3 Notre technique de segmentation : les méthodes variationnelles

Les contours actifs apparaissent comme un outil pertinent pour mettre en équation une énergie à minimiser dans le but d'obtenir la segmentation d'un objet. Le minimum de la fonctionnelle est déduit de l'expression de l'équation aux dérivées partielles (EDP) qui permettra de faire évoluer le contour.

Les contours actifs sont devenus très populaire grâce au papier de Kass *et al* [39] publié en 1988 sur les *snakes*. Cet article compte aujourd'hui plus de 3000 citations. Les *snakes* sont énormément utilisés en vision par ordinateur et en imagerie médicale pour

détecter et segmenter les objets dans une image. Dans le papier [39], les auteurs proposent de mettre en équation l'évolution d'une courbe qui prend en compte des propriétés de régularité et qui a tendance à se coller aux contours des objets. Nous supposons que les contours d'un objet sont définis par les lieux où la norme du gradient de l'image atteint un extremum. Kass *et al.* représentent la courbe de segmentation par une courbe paramétrique (ou explicite) $\Gamma \in C^2$.

$$\begin{aligned} \Gamma : [0,1] \times [0,T] &\rightarrow \mathbb{R}^2 \\ (s,t) &\mapsto \Gamma(s,t) = (x(s,t), y(s,t)) \end{aligned}$$

Le paramètre $s \in [0,1]$ permet de déduire la géométrie de la courbe et $t \in [0,T]$ est le paramètre d'évolution de la courbe. Γ évolue en minimisant localement la fonctionnelle :

$$E(\Gamma) = \int_0^1 \left(\alpha|\Gamma_s(s)|^2 + \beta|\Gamma_{ss}(s)|^2\right) ds - \int_0^1 |\nabla I\left(\Gamma(s)\right)| ds \tag{2.25}$$

Où α et β sont des constantes positives ; Γ_s et Γ_{ss} correspondent aux dérivées premières et secondes de la courbe $\Gamma(s)$ paramétrée par s à t fixée ; $|\nabla I|$ est la norme du gradient de l'image.

Le premier terme de l'équation 2.25 est un terme de régularisation, aussi appelé énergie interne. Il contrôle la longueur de la courbe ainsi que sa courbure. Le deuxième terme est l'énergie externe. Il prend en compte les données image, dans le sens où la courbe qui fournit l'énergie externe minimale est celle située sur les zones de l'image correspondant à un fort gradient.

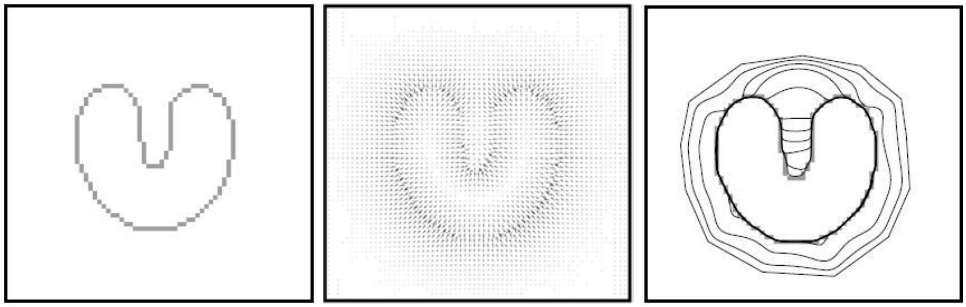
Malgré le fait que les *snakes* soient faciles à implémenter, ils comportent des points faibles non négligeables : la re-paramétrisation de la courbe à chaque itération est délicate ; l'énergie globale dépend de cette paramétrisation et non de la géométrie de l'objet ; la représentation explicite de la courbe ne permet pas les changements de topologie ; le résultat de la segmentation par minimisation dépend de la position initiale de la courbe ; et enfin, l'évolution de l'énergie externe vers d'autres critères comme la couleur, la texture n'est pas évidente comme nous pouvons le voir dans le papier de Jehan-Besson *et al.* [34].

Pour apporter quelques éléments de réponse à l'ensemble de ces problèmes, nous allons par la suite discuter des différentes solutions proposées par la communauté :

- la force Ballon,
- les contours actifs géodésiques,
- la représentation implicite des courbes (en particulier le cas des courbes de niveaux),
- des fonctionnelles qui prennent en compte des information sur les régions de l'image.

La méthode d'optimisation locale proposée pour les *snakes* a été énormément discutée car le résultat de segmentation dépend fortement de l'initialisation de la courbe. L'algorithme a tendance à attirer la solution finale vers un minimum local de la fonctionnelle. Cela se produit plus particulièrement lorsque l'on travaille avec des images bruitées, puisque l'énergie externe contient beaucoup de minima locaux, ce qui se répercute sur l'énergie globale (équation 2.25). Pour palier ce problème et diriger les courbes de segmentation vers le minimum global de la fonctionnelle, Cohen et Cohen, dans [13], la modifient en ajoutant un terme de force ballon $F = k\mathbf{N}$ qui vise à pousser la courbe vers l'intérieur ou l'extérieur de la courbe initiale, suivant le signe de la constante k et l'orientation de la normale. Malheureusement, cette technique fait l'hypothèse que l'on sait si les contours de l'objet à segmenter sont à l'intérieur ou l'extérieur de la courbe initiale.

Dans le cas des *snakes*, si l'objet à traiter possède des frontières non convexes (cf Figure 2.13(a)) alors le contour actif n'aura pas assez d'énergie pour coller aux frontières. Pour palier ce problème, Xu et Prince [84] utilisent le champ de vecteurs gradients (cf Figure 2.13(b)) pour diriger l'évolution du contour actif et pour ainsi atteindre les frontières des objets dans les parties convexes. Les auteurs cherchent à diffuser l'information du gradient de l'image dans le but d'attirer les contours actifs vers les frontières de l'objet (cf Figure 2.13(c)).



(a) (b) (c)

FIGURE 2.13 – Segmentation d'un objet ayant des frontières non convexes par le champ de vecteur gradient - Source [84]

Les contours actifs géodésiques proposés indépendamment par Caselles *et al.* [6] et Kichenassamy *et al.* [40] proposent une nouvelle paramétrisation de la fonctionnelle 2.25. On apporte quelques modifications sur la fonctionnelle, on fixe $\beta = 0$ pour supprimer les termes de dérivée trop élevée mais nous conservons le terme en α qui nous garantit que la courbe soit lisse. Pour l'énergie extérieur, on ajoute une fonction g strictement décroissante vers 0 à l'infini. La nouvelle fonctionnelle s'écrit :

$$E(\Gamma) = \alpha \int_0^1 |\Gamma_s(s)|^2 ds + \lambda \int_0^1 g\left(|\nabla I(\Gamma(s))|\right)^2 ds \tag{2.26}$$

Avec l'aide des principes de Maupertuis et Fermat, on montre que le minimum de l'équation 2.26 est donné par une courbe géodésique pour une métrique riemannienne sur le domaine de l'image. Minimiser l'équation 2.26 revient à minimiser l'équation suivante :

$$\begin{aligned} Eg &= \int_0^1 g\left(|\nabla I\left(\Gamma(s)\right)|\right) |\Gamma_s(s)| ds \\ &= \int_0^L g\left(|\nabla I\left(\Gamma(s)\right)|\right) ds \end{aligned} \tag{2.27}$$

Avec L la longueur euclidienne de la courbe et ds l'élément infinitésimal le long de la courbe.

La solution du problème est donc la courbe de longueur minimale où la longueur prend en compte les caractéristiques de l'image.

Caselles *et al.* [6] montrent que l'évolution de la courbe Γ vers un minimum local suit l'équation suivante :

$$\frac{\partial \Gamma}{\partial t} = g(I)\kappa \mathbf{N} - (\nabla g \cdot \mathbf{N})\,\mathbf{N} \tag{2.28}$$

Avec κ la courbure de la courbe et $\mathbf{N}$ le vecteur normal à la courbe. Le premier terme de l'équation 2.28 permet de ralentir la diminution de la longueur de la courbe et de stopper l'évolution de la courbe sur les contours de l'image. Le deuxième terme de cette équation fait évoluer la courbe dans sa direction orthogonale vers les contours de l'image.

En calcul variationnel, nous avons posé la segmentation d'une image comme une minimisation d'une fonctionnelle adaptée au problème. L'idée directrice est de faire évoluer la courbe Γ par une méthode de descente du gradient de l'énergie totale. Si l'on représente la courbe de manière explicite, son évolution se traduit par l'équation $\frac{\partial \Gamma}{\partial t} = -\frac{\partial E(\Gamma)}{\partial \Gamma} = F \cdot \mathbf{N}$. (Nous remarquons que l'évolution du contour est modélisée par une force portée uniquement par la normale, car les composantes tangentielles n'interviennent pas sur l'évolution de la courbe Γ mais sur sa paramétrisation.)

Si l'on représente la courbe de manière implicite comme étant le niveau zéro d'une fonction de dimension supérieure $\Phi : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ on a :

$$\omega \in \Gamma(s,t) \Leftrightarrow \Phi(\omega,t) = 0$$

que l'on écrit aussi $\Phi(\Gamma(s,t),t) = 0 \qquad \forall s \in [0,1] \text{et} \forall t \geq 0.$

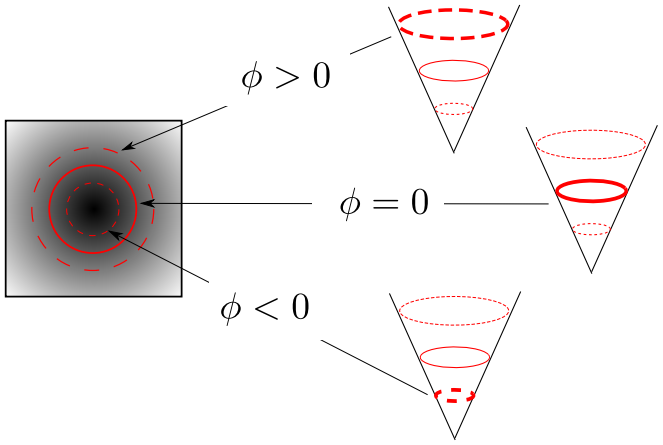


FIGURE 2.14 – Représentation implicite d'un contour actif par une fonction distance signée.

Il est courant de choisir Φ comme étant une fonction distance signée au contour $\Gamma(s,t)$ négative à l'intérieur et positive à l'extérieure (*cf.* Figure 2.14) qui est solution de l'équation Eikonale $|\nabla_\omega \Phi| = 1$. Le contour cessera d'évoluer lorsque $\frac{\partial \Phi(\Gamma(s,t),t)}{\partial t} = 0$.

$$\frac{\partial \Phi(\Gamma(s,t),t)}{\partial t} = \nabla\Phi \frac{\partial \Gamma}{\partial t} + \frac{\partial \Phi}{\partial t} = 0 \tag{2.29}$$

Nous avons vu précédemment que $\frac{\partial \Phi}{\partial t} = F \cdot \mathbf{N}$. Φ étant une fonction distance signée, nous pouvons écrire $\mathbf{N} = \frac{\nabla \mathbf{\Phi}}{|\nabla \mathbf{\Phi}|}$.

L'équation 2.29 devient alors :

$$\frac{\partial \Phi}{\partial t} + F|\nabla\Phi| = 0 \tag{2.30}$$

Nous pouvons alors réécrire l'équation 2.28 avec la représentation implicite de la courbe :

$$\frac{\partial \Phi}{\partial t} = g(I)|\nabla\Phi| \text{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) + \nabla g(I) \cdot \nabla\Phi \tag{2.31}$$

En plus de permettre des changements topologiques, les représentations implicites des courbes permettent de pouvoir exploiter assez aisément les informations sur les régions intérieures et extérieures à la courbe.

L'article [56] de Mumford and Shah apporte un nouvel élément primordial pour la segmentation d'image utilisant les méthodes variationnelles. Ils proposent une fonctionnelle qui allie les données contour et région de l'image.

$$E_{MS}(u,\Gamma) = \frac{1}{2}\int_{\Omega}(u-I)^2 d\omega + \frac{\alpha}{2}\int_{\Omega\setminus\Gamma}|\nabla u|^2 d\omega + \beta\ L(\Gamma) \tag{2.32}$$

Où u est une version lissée qui conserve les contours de l'image I (c'est l'image *cartoon*)et Γ la traditionnelle courbe de segmentation.

Cette fonctionnelle sera reprise par Cohen *et al.* dans [12] pour de la reconstruction 3D par contours actifs. Chan et Vese [8] proposent une version simplifiée de la fonctionnelle de Mumford Shah. Les auteurs décomposent le domaine image Ω en deux régions Ω_i et $\Omega_e = \Omega\setminus\Omega_i$ qui sont respectivement des domaines correspondant aux parties intérieures et extérieures à la courbe, et cherchent à maximiser l'homogénéité des régions en minimisant la longueur de la courbe et en maximisant la région correspondant à l'objet à segmenter. La fonctionnelle s'écrit :

$$\begin{aligned} E_{CV}(\Gamma, c_i, c_e) &= \lambda_i \int_{\Omega_i} |I(\omega) - c_i|^2 d\omega \\ &+ \lambda_e \int_{\Omega_e} |I(\omega) - c_e|^2 d\omega \\ &+ \alpha\ L(\Gamma) \\ &+ \mu \int_{\Omega_i} 1\ d\omega \end{aligned} \tag{2.33}$$

Où c_i et c_e sont des réels à déterminer ; λ_i, λ_e, α et μ sont des constantes à valeurs réelles ;et enfin, $L(\Gamma)$ est la longueur euclidienne de la courbe. Nous pouvons facilement monter que les valeurs optimales de c_i et c_e correspondent respectivement aux moyennes des valeurs de l'image I sur les régions Ω_i et Ω_e.

$$c_i = \frac{\int_{\Omega_i} I(\omega)d\omega}{\int_{\Omega_i} 1 d\omega} \quad \text{et} \quad c_e = \frac{\int_{\Omega_e} I(\omega)d\omega}{\int_{\Omega_e} 1 d\omega} \tag{2.34}$$

Lorsque l'on représente la courbe de manière implicite, par le niveau zéro de la fonction distance signée Φ, l'équation 2.33 s'écrit :

$$\begin{aligned} E(\Phi, c_i, c_e) &= \lambda_i \int_{\Omega} H(\Phi(\omega))|I(\omega) - c_i|^2 d\omega \\ &+ \lambda_e \int_{\Omega} (1 - H(\Phi(\omega)))\,|I(\omega) - c_e|^2 d\omega \\ &+ \alpha \int_{\Omega} \delta(\Phi(\omega))|\nabla\Phi(\omega)| d\omega \\ &+ \mu \int_{\Omega} H(\Phi(\omega)) d\omega \end{aligned} \tag{2.35}$$

Où H est la fonction échelon ou Heaviside et δ sa fonction dérivée, soit la fonction de Dirac.

Nous rappellons que :

$$\begin{aligned} H(\Phi(\omega)) &= \left(\begin{array}{lll} 1 & si & \Phi(\omega) \geq 0 \\ 0 & si & \Phi(\omega) < 0 \end{array} \right) \text{Sélection des pixels } \in \Omega_i \\ 1 - H(\Phi(\omega)) &= \left(\begin{array}{lll} 0 & si & \Phi(\omega) \geq 0 \\ 1 & si & \Phi(\omega) < 0 \end{array} \right) \text{Sélection des pixels } \in \Omega_e \end{aligned}$$

Des descripteurs statistiques plus complexes que la moyenne ont été étudiés. Aubert *et al.* [3] mettent en place un critère qui s'appuie sur les histogrammes des régions. Dans [44], Leclerc obtient une segmentation en utilisant un critère de longueur de description minimale (*minimum description length* en anglais). Un dernier exemple de papier qui utilise un critère statistique dans la fonctionnelle, Paragios et Deriche [61] introduisent une probabilité *a posteriori* d'une décomposition optimale des régions connaissant l'image.

Cet état de l'art s'appui sur l'article de Lauze et Nielsen [42], l'article de Cremers *et al.* [20] et le livre de Osher et Paragios [58].

Chapitre 3

Segmentation par classification supervisée

Précédemment, nous avons énoncé la problématique, effectué un état de l'art sur l'analyse des images de type Portrait en vision par ordinateur et décrit notre stratégie qui consiste à utiliser des connaissances *a priori* pour aider à résoudre le problème de segmentation du visage posé.

Dans ce chapitre, nous utilisons des hypothèses très fortes et des connaissances fiables pour simplifier le problème. Un utilisateur supervise la segmentation en désignant manuellement trois zones échantillonnant les trois classes considérées. Il annote donc trois régions caractéristiques des classes Peau, Cheveux et Fond. Grâce à cette supervision, nous proposons des descripteurs colorimétriques et texturaux suffisamment discriminants pour classer correctement chaque pixel. Nous introduisons, en particulier, un descripteur original de texture que nous comparons à des propositions récentes issues de la littérature. Ce descripteur utilise une analyse en composantes principales pour comparer les trois hypothèses de classification. Un modèle de mélange simple, permet alors de modéliser les distributions des descripteurs proposés avant d'utiliser des classifieurs standards avec ou sans régularisation spatiale.

Nous obtenons finalement des résultats encourageants qui forment une segmentation grossière initiale à préciser avec les méthodes variationnelles proposées dans le chapitre 4.

Ces résultats, obtenus grâce à une supervision manuelle, devront être confirmés dans le cadre d'une approche plus automatique que nous envisageons au chapitre 5.

3.1 Problématique

Dans le chapitre 2, nous avons recensé plusieurs techniques de segmentation. Ici, nous présentons une technique de segmentation "grossière" par classification. L'objectif du chapitre est d'associer à chaque pixel de l'image une des trois étiquettes : Peau, Cheveux ou Fond.

Pour cela, nous étudions une méthode supervisée nous permettant d'acquérir des connaissances sur l'image que nous devons segmenter, et ainsi adapter le modèle de classification à l'image traitée. L'idée générale est d'étudier les performances d'une supervision manuelle pour, ensuite, effectuer un apprentissage automatique en ligne sur les données à classifier. En pratique, la phase de supervision manuelle considérée ici, localise trois zones notées Z^p, Z^c et Z^f, correspondant respectivement à un échantillon des régions Peau, Cheveux et Fond (exemple sur la Figure 3.1).

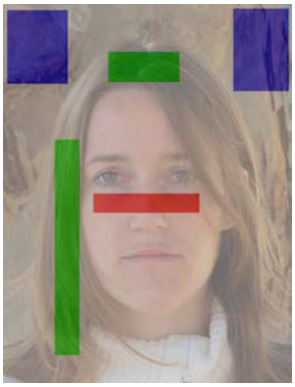


FIGURE 3.1 – Affichage des zones Z^p, Z^c et Z^f en rouge, vert et bleu.

Une fois que nous aurons validé les résultats de notre segmentation utilisant une étape de supervisation manuelle, nous pourrons (dans le chapitre 5) proposer une technique pour détecter automatiquement les trois zones. Dans ce chapitre, nous nous intéressons uniquement au système (*cf.* Figure 3.2) permettant de segmenter les images Portrait à partir des zones Peau, Cheveux et Fond.

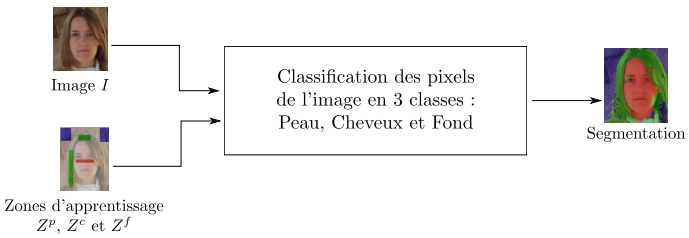


FIGURE 3.2 – Schéma d'entrée-sortie de notre segmentation supervisée.

Ce système peut être décomposé en trois grandes étapes (*cf.* Figure 3.3) :

– **La construction de descripteurs adaptés à l'image** doit, à partir des zones d'apprentissage, construire un descripteur de texture capable de révéler les caractéristiques texturales discriminantes de chaque classe. Notre méthode s'appuie sur une Analyse en Composantes Principales (ACP). Nous utilisons également la couleur pour distinguer les trois classes en présence.

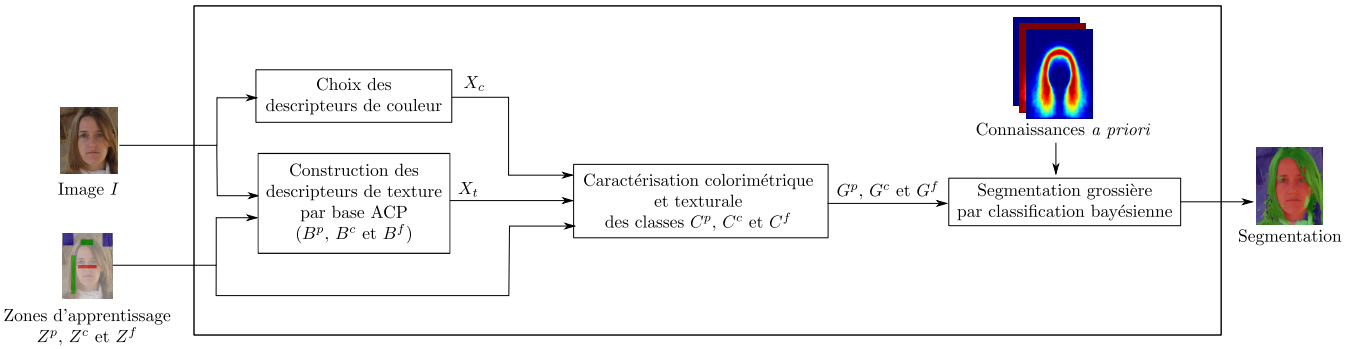


FIGURE 3.3 – De l'image à la segmentation en passant par les descripteurs

– **La modélisation des caractéristiques de chacune des classes** doit, à partir des descripteurs de couleur et de texture de l'étape précédente et des zones d'apprentissage, donner un modèle caractéristique de la distribution des descripteurs de chaque classe.
– **Le choix et l'apprentissage du modèle décisionnel** doit permettre de classifier l'ensemble des pixels de l'image en leur attribuant une des trois étiquettes Peau, Cheveux ou Fond.

Dans la section suivante, nous mettons en place les notations utilisées dans ce chapitre. Ensuite, nous construisons notre descripteur de texture que nous comparons à des travaux récents proposés par Mairal *et al.* [52]. Dans la section 3.3, nous proposons une façon de caractériser chacune des classes Peau, Cheveux et Fond. Enfin, nous mettons en place un modèle décisionnel prenant en compte les caractéristiques des classes mais également le contexte global des images Portrait, c'est-à-dire des relations spatiales inter-classes.

3.2 Les Descripteurs de texture

De nombreux travaux ont montré l'intérêt d'exploiter la texture pour la segmentation d'images [20, 37, 66]. Classiquement, une première étape est la construction de descripteurs caractéristiques de textures associées aux différents segments. Il est courant d'utiliser des modèles paramétriques comme les *curvelets*, *bandelets* ou d'autres formes encore d'ondelettes pour révéler les propriétés de la texture et ainsi construire des descripteurs. Récemment, dans [25, 63], l'idée a évolué et les auteurs remplacent les modèles paramétriques, par un dictionnaire composé de k atomes redondants appris sur le signal à traiter. Les travaux de Mairal *et al.*, dans [52], sont parmi les plus aboutis dans ce domaine. L'intérêt principal de ce travail est d'optimiser explicitement le caractère « discriminant » des atomes formant le dictionnaire. Nous proposons de comparer notre modélisation de la texture avec la leur. Dans un premier temps, nous présentatons en détail les travaux de Mairal *et al.*. Ensuite, nous introduisons et développons notre approche. Enfin, nous comparons les qualités de ces descripteurs de texture pour les images Portait en s'appuyant sur une segmentation utilisant uniquement l'information de texture (comme proposé dans les travaux de Mairal *et al.*).

3.2.1 L'apprentissage de dictionnaires discriminants

Mairal *et al.* [52] s'intéressent à la segmentation d'image en utilisant uniquement les informations de texture. L'algorithme proposé est fondé sur la construction de dictionnaires dont les atomes sont redondants. Au lieu de former une base (au sens Hilbertien), les atomes constituant le dictionnaire forment une famille redondante permettant de reconstruire le signal à analyser. Grâce à la richesse du dictionnaire considéré, la texture à analyser peut être reconstruite et caractérisée avec seulement quelques atomes. La sélection des atomes au sein du dictionnaire exploite donc le concept de parcimonie. Les représentations parcimonieuses sont populaires pour le traitement de l'image et de la vidéo. Dans le cas de la segmentation, pour que le dictionnaire associé à une classe soit adapté à elle et inadapté aux autres, les auteurs proposent d'optimiser le caractère discriminant des dictionnaires.

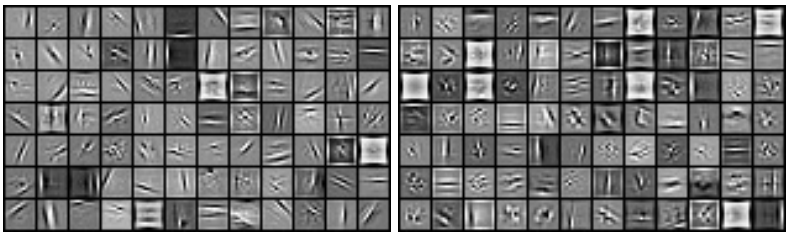
Dans la suite, nous décrivons précisément les travaux de Mairal *et al.* et nous les adaptons à notre problématique. Tout d'abord, nous présentons leur technique utilisée pour construire un dictionnaire. Dans un deuxième temps, nous adaptons leur modélisation concernant la construction de dictionnaires discriminants. Nous allons construire trois dictionnaires discriminants adaptés à nos trois classes Peau, Cheveux et Fond.

Il est important de s'attarder sur la mise en équation de Mairal *et al.* pour s'assurer que notre descripteur de texture, défini par ACP et présenté dans la suite, est comparable.

3.2.1.1 Apprentissage d'un dictionnaire pour la reconstruction d'une image

Dans cette partie, nous présentons comment Mairal *et al.* construisent un dictionnaire dont les atomes possèdent de bonnes propriétés : ils sont redondants et adaptés au signal. Nous disons que le dictionnaire est adapté au signal, si le signal initial peut être reconstruit grâce à quelques atomes du dictionnaire, c'est une approche parcimonieuse.

Plus formellement, considérons un signal $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$, on dit que $\mathbf{x}$ admet une approximation parcimonieuse issue d'un dictionnaire $D \in \mathbb{R}^{\mathbf{n}\times\mathbf{k}}$ composé de k atomes de dimension $\mathbf{n}$, si on peut construire D et rechercher L atomes de D (avec $L << k$) qui permettent d'approcher au mieux le signal $\mathbf{x}$.



(a) Dictionnaire "Vélo" appris sur des zones de l'image contenant le vélo.

(b) Dictionnaire "Fond" appris sur des zones de l'image contenant le fond.

FIGURE 3.4 – Echantillons des atomes des dictionnaires appris pour la classe "Vélo" de la basse de données *Pascal VOC6*

Pour les signaux 2D, comme les images en niveau de gris, le signal $\mathbf{x} \in \mathbb{R}^{\mathbf{n}}$ correspond à un échantillon de l'image taille $n \times n$. Chaque échantillon de l'image est noté $\mathbf{x}_l$ avec $l \in [1 \dots M]$. Intuitivement, nous découpons l'image en M blocs de taille $n \times n$, chaque bloc étant un échantillon. Nous apprenons sur cet ensemble d'échantillons, le dictionnaire

(a) Image de test (b) Résultat de classification

FIGURE 3.5 – Illustration des résultats de classification utilisant les dictionnaires.

composé de k atomes de dimension n^2. La contrainte de parcimonie est définie par le fait que chaque échantillon peut être reconstruit uniquement par la combinaison linéaire de $L \leq n^2$ atomes du dictionnaire.

Le dictionnaire est alors appris en résolvant le problème suivant :

$$\min_{\alpha_1, D} \sum_{l=1}^{M} \|\mathbf{x}_l - D\alpha_l\| \qquad \text{Avec} \quad |\alpha_l|_0 \leq L \tag{3.1}$$

Dans cette équation, $\mathbf{x}_l \in \mathbb{R}^n$ est un échantillon de l'image ; $D \in \mathbb{R}^{n \times k}$ est le dictionnaire que l'on souhaite apprendre, dont chaque atome est normalisé (les k colonnes de D sont unitaires) ; $\alpha_l \in \mathbb{R}^k$ est un vecteur creux qui recense les coefficients de la combinaison linéaire des L vecteurs qui minimisent l'erreur de reconstruction de l'échantillon $\mathbf{x}_l$; $|.|_0$ représente la « norme L^0 » qui permet de limiter le nombre d'atomes utilisés dans la construction de $D\alpha_l$. Formellement, nous pouvons écrire $|\alpha_l|_0 = \text{Card}\{k, \text{tel que } \alpha_{l,k} \neq 0\}$. Du fait de la présence de la contrainte de parcimonie, le système à résoudre n'est pas linéaire. Pour minimiser l'équation 3.1, Mairal *et al.* proposent une résolution itérative sur α et D. Le paramètre α est estimé par l'algorithme dit d'*Orthogonal Matching Pursuit* [53] ; quant à la minimisation sur D, elle est opérée par les algorithmes MOD [26] ou K-SVD.

Orthogonal Matching Pursuit (OMP pour *Orthogonal Matching Pursuit* en anglais)

Mallat et Zhang [53] proposent un algorithme pour résoudre l'expression 3.1 non-linéaire en α avec D fixé, en prenant en compte la contrainte de parcimonie. OMP est un algorithme glouton qui décompose un signal en une combinaison linéaire d'atomes sélectionnés dans un dictionnaire redondant adapté au signal. A chaque itération, nous choisissons l'atome le plus adapté pour approcher tout ou partie du signal. A la première itération, nous pouvons écrire :

$$\mathbf{x}_l = \langle \mathbf{x}_l, d_{j_1} \rangle d_{j_1} + R_{l/d_{j_1}}.$$

Où $R_{l/d_{j_1}}$ représente l'erreur résiduelle obtenue en retirant du signal l'information portée par l'atome noté d_{j_1} (l'atome est en pratique une colonne du dictionnaire D). Par construction, d_{j_1} est orthogonal à $R_{l/d_{j_1}}$, ainsi nous pouvons écrire :

$$\|\mathbf{x}_l\|^2 = |\langle \mathbf{x}_l, d_{j_1} \rangle|^2 + \|R_{l/d_{j_1}}\|^2.$$

Minimiser l'erreur résiduelle R_{l/d_j} revient à maximiser $|\langle \mathbf{x}_l, d_j \rangle|^2$. On cherche donc la valeur de j qui maximise le terme précédent. On écrit alors :

$$j_1 = \arg \max_{j \in [1..k]} |\langle \mathbf{x}_l, d_j \rangle|^2$$

Le résultat de la maximisation permet de définir l'atome du dictionnaire D le plus adapté au signal $\mathbf{x}_l$ et d'en déduire la valeur de la projection du signal sur cet atome d_{j_1}. Cette projection est alors notée α_{l,j_1}.

Aux itérations suivantes, le résidu joue le rôle du signal et ce pendant L itérations. Formellement, nous écrivons

$$j_q = \arg \max_{j \in [1..k]} |\langle R_{l/\{d_{j_1},\ldots,d_{j_{q-1}}\}}, d_j \rangle|^2.$$

En itérant sur chacun des signaux $\mathbf{x}_l$, nous obtenons une matrice creuse $\alpha \in \mathbb{R}^{k \times M}$ qui comporte au maximum L éléments non-nuls par colonne.

Méthode de direction optimale (MOD pour *Method of Optimal Direction* en anglais)

Engan *et al.* [26] présentent un algorithme itératif qui permet d'optimiser les k atomes du dictionnaire D afin de minimiser l'équation 3.1, avec α fixé. Nous notons δ_j le vecteur d'ajustement de l'atome d_j avec $j \in [1 \ldots k]$. Les mises à jour de chaque atome d_j et de l'erreur résiduelle sont respectivement notées $\tilde{d}_j$ et $\tilde{R}_j$. Elles s'écrivent :

$$\tilde{d}_j = d_j + \delta_j$$

$$\begin{aligned} R_l &= \mathbf{x}_l - \sum_{j=1}^{k} d_j \alpha_{l,j} \\ \tilde{R}_l &= R_l - \sum_{j=1}^{k} \delta_j \, \alpha_{l,j} \end{aligned} \tag{3.2}$$

Après chaque mise à jour, on veut que l'erreur résiduelle soit plus faible que l'erreur résiduelle avant mise à jour. On a donc l'inégalité suivante :

$$\sum_l ||\tilde{R}_l||^2 \leq \sum_l ||R_l||^2 \tag{3.3}$$

Le terme de gauche de l'inégalité peut s'écrire :

$$\sum_l ||\tilde{R}_l||^2 = \sum_l ||R_l - \sum_{j=1}^{k} \delta_j \alpha_{l,j}||^2.$$

En développant, nous obtenons un terme de la forme :

$$\sum_l ||\tilde{R}_l||^2 = \sum_l ||R_l||^2 + D,$$

avec

$$D = -2 \sum_l \sum_{j=1}^{k} \alpha_{l,j} \delta_j^t R_l + \sum_l \sum_{j=1}^{k} \sum_{i=1}^{k} \alpha_{l,j} \alpha_{l,i} \delta_j^t \delta_i.$$

On pose :

$$a_{ij} = \sum_l^M \alpha_{l,i} \alpha_{l,j} \quad \text{et} \quad b_j = \sum_l^M \alpha_{l,j} R_l$$

L'inégalité définie à l'équation 3.3 permet de déterminer que D est de signe négatif. Nous pouvons donc en déduire que minimiser $\tilde{R}_l$ revient à minimiser D.

Pour cela, nous cherchons à satisfaire la condition nécessaire suivante :

$$\frac{\partial}{\partial \delta_q(p)} \left(\sum_{j=1}^{k} \sum_{i=1}^{k} \sum_{m=1}^{M} a_{ij} \delta_j(m) \delta_i(m) - 2 \sum_{j=1}^{k} b_j(m) \delta_j(m) \right) = 0,$$

avec $q \in [1...k]$ et $p \in [1...M]$.

$$\Longleftrightarrow$$

$$\frac{\partial}{\partial \delta_q(p)} \left(\sum_{j=1}^{k} \sum_{i=1}^{k} a_{ij} \delta_j(p) \delta_i(p) - 2 \sum_{j=1}^{k} b_j(p) \delta_j(p) \right) = 0$$

$$\Longleftrightarrow$$

$$\frac{\partial}{\partial \delta_q(p)} \left(2 \sum_{i=1}^{k} a_{iq} \delta_q(p) \delta_i(p) - a_{qq} \delta_q^2(p) - 2\, b_j(p) \delta_q(p) \right) = 0$$

$$\Longleftrightarrow$$

$$2 \sum_{\substack{i=1 \\ i \neq q}}^{k} a_{iq} \delta_i(p) + 4 a_{qq} \delta_q(p) - 2 a_{qq} \delta_q(p) - 2\, b_j(p) = 0$$

$$\Longleftrightarrow$$

$$\sum_{i=1}^{k} a_{iq} \delta_i(p) - b_j(p) = 0,$$

On peut en déduire l'écriture matricielle suivante :

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \dots \\ \delta_k \end{bmatrix} = \begin{bmatrix} \sum_l \alpha_{l,1}^2 & \sum_l \alpha_{l,1}\alpha_{l,2} & \dots & \sum_l \alpha_{l,1}\alpha_{l,k} \\ \sum_l \alpha_{l,2}\alpha_{l,1} & \sum_l \alpha_{l,2}^2 & \dots & \sum_l \alpha_{l,2}\alpha_{l,k} \\ \dots & \dots & \dots & \dots \\ \sum_l \alpha_{l,k}\alpha_{l,1} & \sum_l \alpha_{l,k}\alpha_{l,2} & \dots & \sum_l \alpha_{l,k}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_l \alpha_{l,1} R_l \\ \sum_l \alpha_{l,2} R_l \\ \dots \\ \sum_l \alpha_{l,k} R_l \end{bmatrix}. \tag{3.4}$$

On conclut alors que la mise à jour de chaque atome $d_j \forall j \in [1 \dots k]$ du dictionnaire après normalisation s'écrit :

$$\tilde{d_j} = \frac{d_j + \delta_j}{|d_j + \delta_j|}.$$

K-SVD Une généralisation de l'algorithme K-means pour les dictionnaires appelé l'algorithme K-SVD, a été proposé par Aharon *et al.* [2]. K-SVD est une méthode itérative qui alterne entre la définition parcimonieuse du signal sur les atomes du dictionnaire (OMP) et la mise à jour des atomes pour mieux reconstruire le signal, en utilisant une décomposition en valeurs singulières dans le but de minimiser l'erreur d'approximation. En pratique, il permet de mettre à jour à la fois le dictionnaire D mais aussi la valeur des éléments non nuls de α, $\tilde{\alpha}$ note la valeur mise à jour. On fait remarquer que cet algorithme met à jour le dictionnaire atome par atome. Nous notons d_j l'atome à mettre à jour et $\tilde{d_j}$ sa mise à jour. On cherche, pour l'ensemble des échantillons $\mathbf{x}_l$ avec $l \in [1...M]$, la valeur de d_j qui minimise l'erreur résiduelle R_{l/d_j} obtenue en utilisant le dictionnaire D privé de l'atome d_j :

$$R_{l/d_j} = \mathbf{x}_l - D\alpha_l + d_j \alpha_{l,j}$$

On cherche alors à minimiser le critère suivant :

$$\min_{\|\tilde{d}_j\|=1} \sum_{l} \|R_{l/d_j} - \tilde{d}_j \tilde{\alpha}_{l,j}\|_2^2 \tag{3.5}$$

On veut garantir que la mise à jour des données permette toujours une décomposition parcimonieuse du signal. Pour assurer cette propriété, on définit :

$$w_j = \{l | 1 \leq l \leq M, \alpha_{l,j} \neq 0\}.$$

Le vecteur w_j représente l'ensemble des indices (pour les échantillons $\mathbf{x}_l$) qui utilisent l'atome d_j (c'est à dire ceux pour lesquels $\alpha_{l,j} \neq 0$). La méthode pour optimiser d_j est alors la suivante :

– On restreint la minimisation en ne considérant dans la minimisation que les erreurs résiduelles définit en fonction de l'atome d_j. L'équation 3.5 devient :

$$\min_{\|\tilde{d}_j\|=1} \sum_{w_j} \|R_{l/d_j} - \tilde{d}_j \tilde{\alpha}_{l,j}\|_2^2 \tag{3.6}$$

– La solution est le couple $(\tilde{d}_j, \tilde{\alpha}_{l,j})$, il est assuré que les supports de $\alpha_{l,j}$ (c'est à dire w_j, ceux non nuls) restent inchangés. Pour déterminer le minimum, on va utiliser une décomposition en valeurs singulières (SVD pour *Singular Value Decomposition* en anglais) de la matrice $E = [R_{w_j(1)/d_j} | R_{w_j(2)/d_j} | ... | R_{w_j(|w_j|)/d_j}] = U \Delta V^t$. La solution pour $\tilde{d}_j$ et $\tilde{\alpha}_{l,j}$ est alors la suivante :
  – $\tilde{d}_j$ est la première colonne de U,
  – $\tilde{\alpha}_{l,j}$ est la première colonne de V multipliée par $\Delta(1,1)$

L'étape de mise à jour est donc une généralisation de l'algorithme K-means. Chaque échantillon peut être représenté par plusieurs atomes de poids différents. En théorie, il n'est pas garanti que cet algorithme converge, mais en pratique les dictionnaires appris via cette méthode montrent des résultats très convaincants et notamment pour supprimer le bruit dans les images.

3.2.1.2 Apprentissage de plusieurs dictionnaires discriminants pour la segmentation d'une image

Dans cette partie, nous adaptons les travaux de Mairal *et al.* à notre problème en construisant trois dictionnaires D^p, D^c et D^f à partir des trois zones *a priori* Z^p, Z^c et Z^f (*cf.* Figure 3.1) caractéristiques du problème de segmentation. Tout comme dans le paragraphe précédent, nous cherchons à minimiser l'erreur résiduelle de chaque zone avec son dictionnaire associé. Le dictionnaire D^p correspond à celui dont les atomes sont appris avec les données de la zone Z^p. Par construction ce dictionnaire est plus adapté à la reconstruction de signal du type Peau qu'à la reconstruction du signal Cheveux ou Fond. Autrement dit, plus l'erreur de reconstruction produite par l'utilisation du dictionnaire Peau est élevée, moins le signal a de chance d'être de la Peau. Ce caractère discriminant des dictionnaires est assuré par l'introduction de la fonction *Softmax* .

La fonction *Softmax* Elle permet de construire des dictionnaires qui à la fois produisent une "bonne" reconstruction pour leur classe et une "mauvaise" reconstruction pour les

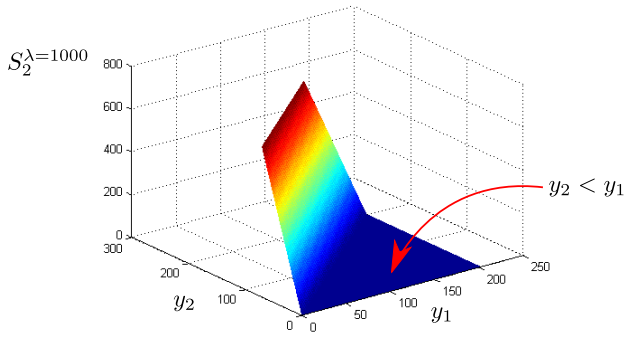


FIGURE 3.6 – Représentation de la fonction *softmax* avec $\lambda = 1000$ et $N = 2$.

autres. C'est pour mettre en équation cette idée que nous utilisons la fonction discriminante *Softmax*. Cette dernière est une fonction de $\mathbb{R}^N$ dans $\mathbb{R}^N$ qui associe à chaque vecteur $(y_1, y_2, \ldots, y_N)$ le vecteur $S^\lambda(y_1, y_2, \ldots, y_N)$ dont la i-ème composante s'écrit :

$$S_i^\lambda(y_1, y_2, \ldots, y_N) = \log\left(\sum_{j=1}^{N} \exp\left(-\lambda(y_j - y_i)\right)\right) \tag{3.7}$$

S_i^λ est d'autant plus proche de 0 que $y_i < y_j \qquad \forall j \neq i.$

La mise en équation pour la construction de dictionnaires discriminants Dans cette partie, nous modélisons la construction de dictionnaires à la fois discriminants et aussi adaptés à chacune des classes de l'image.

Mairal *et al.* proposent pour notre problème à trois classes de minimiser la fonctionnelle suivante :

$$\min_{\{D^j\}_{j=[p,c,f]}} \sum_{i=[p,c,f]} \sum_{l \in Z^i} S_i^\lambda\left(\{R(x_l, D^j)\}_{j=[p,c,f]}\right) + \lambda\gamma R(x_l, D^i), \tag{3.8}$$

$$\text{avec} \qquad R(x_l, D^i) = \|\mathbf{x_l} - D^i\alpha_l\|.$$

Dans cette équation , le paramètre γ est une constante positive qui permet de pondérer le score obtenu par le calcul de l'erreur résiduelle (terme en R) et le score de discrimination entre les dictionnaires (terme en S). Nous remarquons de plus que dans le cas où $\gamma = 0$, nous nous ramenons à une fonctionnelle qui prend en compte uniquement le caractère discriminant des dictionnaires. Cette fonctionnelle admet une infinité de solution. De plus sa résolution est hautement instable d'après les auteurs. Le paramètre λ contenu dans la deuxième partie de la fonctionnelle permet d'assurer la comparaison des deux termes. Il joue également un rôle important pour la stabilité et l'efficacité du schéma proposé.

Nous cherchons le minimum d'une fonctionnelle composée de deux termes. L'un encode l'erreur de reconstruction et l'autre la discrimination des dictionnaires. Par rapport à la fonctionnelle 3.1, l'ajout du terme assurant la discrimination des dictionnaires rend plus complexe la minimisation de la fonctionnelle. En effet, ce terme est hautement instable. Les auteurs décident de faire évoluer la vitesse de discrimination en remplaçant le paramètre λ par une série croissante. L'effet de ce changement entraîne une rapidité pour atteindre le minimum global de la fonctionnelle.

Pour que la variation du paramètre λ ait une influence uniquement sur la vitesse de discrimination et pas sur le poids entre les deux termes énergétiques nous devons exprimer le terme énergétique lié à l'erreur de reconstruction en fonction de λ. Asymptotiquement, la fonction de coût associée au caractère discriminant se comporte comme la fonction $\lambda(y_i - \min_j y_j)$. Nous pouvons donc en déduire que pour que les termes énergétiques soient comparables au cours du temps, l'énergie associée à l'erreur de reconstruction s'écrit $\lambda R(x_l, D_i)$.

Nous réécrivons la fonctionnelle de l'équation 3.9 de la manière suivante :

$$f(y_1, y_2, \ldots, y_N) = \sum_{i=1}^{N} \sum_{l \in Z^i} S_i^{\lambda}(y_1, y_2, \cdots, y_N) + \lambda\gamma y_i$$

et nous introduisons la notation :

$$g(y) = f(y_1, \cdots, y_{p-1}, y, y_{p+1}, \cdots, y_N).$$

Le développement de Taylor de la fonction g à l'ordre 1 s'écrit :

$$g(y_0 + y) = g(y_0) + \frac{dg}{dy}(y_0)y \tag{3.9}$$

Avec ces outils, résoudre l'équation 3.9 revient à résoudre :

$$\min_{\{D^j\}_{j=[p,c,f]}} \sum_{i=[p,c,f]} \sum_{l \in Z^i} \omega_l R(x_l, D^j) \qquad \text{Avec } \omega_l = \frac{\partial S_i^{\lambda}}{\partial y_p}\left(\{R(x_l, D^j)\}_{j=[p,c,f]}\right) + \lambda\gamma\delta_{pi} \tag{3.10}$$

Où δ_{pi} est le symbole de Kronecker, c'est à dire que $\delta_{pi} = 1$ si $i = p$ et $\delta_{pi} = 0$ sinon.

En utilisant l'algorithme de K-SVD, la mise à jour de chacun des dictionnaires s'effectue de manière itérative. Pour chaque dictionnaire, pour chaque atome d_j on résout l'équation :

$$\min_{\|d_j^{(t+1)}\|=1, \alpha_l^{(t+1)}(j)} \sum_{k=[p,c,f]} \sum_{l \in Z^k \cap w} \omega_l \|R_{/d_j} - d_j^{(t+1)} \alpha_l^{(t+1)}(j)\|_2^2 \tag{3.11}$$

La solution de cette minimisation correspond au couple (vecteur propre - valeur propre) associé à la plus grande valeur propre de la matrice $B = \sum_{k=[p,c,f]} \sum_{l \in Z^k \cap w} \omega_l (R_{/d_j})(R_{/d_j})^t$.

3.2.2 L'apprentissage de dictionnaires adaptées aux zones

Nous définissons la texture associée à un pixel ω par un vecteur de taille n correspondant à la concaténation des colonnes d'un échantillon de taille $\sqrt{n} \times \sqrt{n}$, centrée en ω (*cf.* Figure 3.7).

L'idée qui guide ce chapitre est qu'un échantillon de Peau (respectivement Fond, Cheveux) est naturellement plus vite expliqué (recomposé/reconstruit) en utilisant un dictionnaire (ou une base) appris sur la zone Peau (respectivement Fond, Cheveux). Nous cherchons alors à définir pour chaque classe une représentation adaptée de l'espace de texture en utilisant les connaissances *a priori* des zones Z^p, Z^c et Z^f (*cf.* Figure 3.1).

Chaque zone Z^i, nous permet de former une matrice de texture, notée $T^i \in \mathbb{R}^{n \times M^i}$, qui correspond aux M^i vecteurs de texture de taille $n = \sqrt{n} \times \sqrt{n}$, le j-ième vecteur de texture

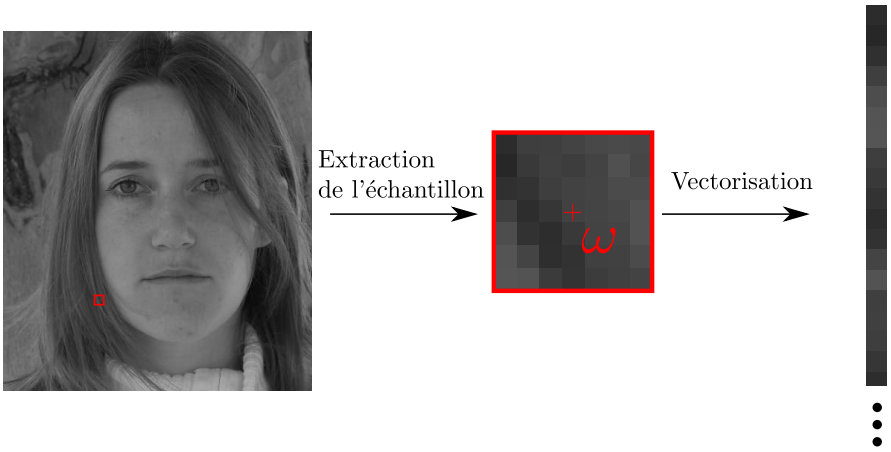


FIGURE 3.7 – Représentation d'un échantillon de texture de cheveux associé à un pixel ω de l'image en niveau de gris I_g.

de la zone Z^i s'écrit : t^i_j . Formellement, la matrice d'apprentissage de la texture de la zone Z^i s'écrit :

$$T^i = \left[t^i_1 | \cdots | t^i_{M^i}\right].$$

Une analyse en composantes principales sur les données T^i, nous fournit une représentation de l'espace de texture adaptée à la zone Z^i. Cette représentation est une base, B^i, dont l'orientation et l'importance de chacun des vecteurs dépend de la nature des données apprises. En pratique, le calcul de cette base est direct. Nous introduisons $\overline{t^i}$:

$$\overline{t^i} = \frac{1}{M^i} \sum_{j=1}^{M^i} t^i_j$$

le vecteur correspondant au centre de gravité du nuage de points de la matrice T^i. Une décomposition en valeurs singulières [24] (SVD pour *Singular Value Decomposition* en anglais) sur les vecteurs centrés $(t^i_j - \overline{t^i})_{j\in[1..M^i]}$ construit la base B^i. Dans cette base, les vecteurs ϕ^i_k sont ordonnés par ordre décroissant des valeurs propres associées.

$$B^i = \{\phi^i_1, \phi^i_2, \ldots, \phi^i_n\}$$

Chaque élément de texture $t^i_j \in T^i$ s'écrit comme une combinaison linéaire des vecteurs de la base B^i :

$$\forall i \in \{p, c, f\} \qquad \forall j = 1 \cdots M^i \qquad t^i_j = \overline{t^i} + \sum_{k=1}^{n} \alpha^i_{k,j} \phi^i_k$$

Où $\alpha^i_{k,j}\phi^i_k$ est la projection de l'échantillon $t^i_j - \overline{t^i}$ sur ϕ^i_k, le k-ième vecteur de la base B^i :

$$\alpha^i_{k,j} = \left\langle (t^i_j - \overline{t^i}) | \phi^i_k \right\rangle$$

Pour chaque base B^i, pour chaque échantillon de l'image $e(\omega)$ de taille n centré en ω, l'erreur de reconstruction (aussi appelée erreur résiduelle) s'écrit :

$$E^i(\omega) = \left\| e(\omega) - \left(\overline{t^i} + \sum_{j=1}^{k\leq n} \langle e(\omega) - \overline{t^i} \cdot \phi^i_j \rangle \phi^i_j \right) \right\|^2_2 . \tag{3.12}$$

Avec k le nombre de vecteurs des bases B^i utilisé pour décrire la texture des zones Z^i. Dans le cas où $k = n$, nous employons la totalité des vecteurs de la base et l'erreur de reconstruction est alors nulle pour tous les pixels de l'image, quelque soit sa classe. Ceci est sans intérêt. Par contre, pour exploiter la différence entre les différentes bases, nous recherchons le nombre k de vecteurs à utiliser pour qu'un échantillon de Peau (respectivement Cheveux et Fond) ait la plus petite erreur résiduelle en utilisant les k premiers vecteurs de la base B^p (respectivement B^c et B^f). Pour l'ensemble des classes Peau, Cheveux et Fond, il en résulte donc que les k premiers vecteurs de la base B^i seront plus adaptés à la description des éléments de la zone Z^i qu'à la description des éléments de la zone Z^j avec $i \neq j$.

Ainsi la base B^i sera plus adaptée à la classe C^i qu'à la classe C^j avec $j \neq i$.

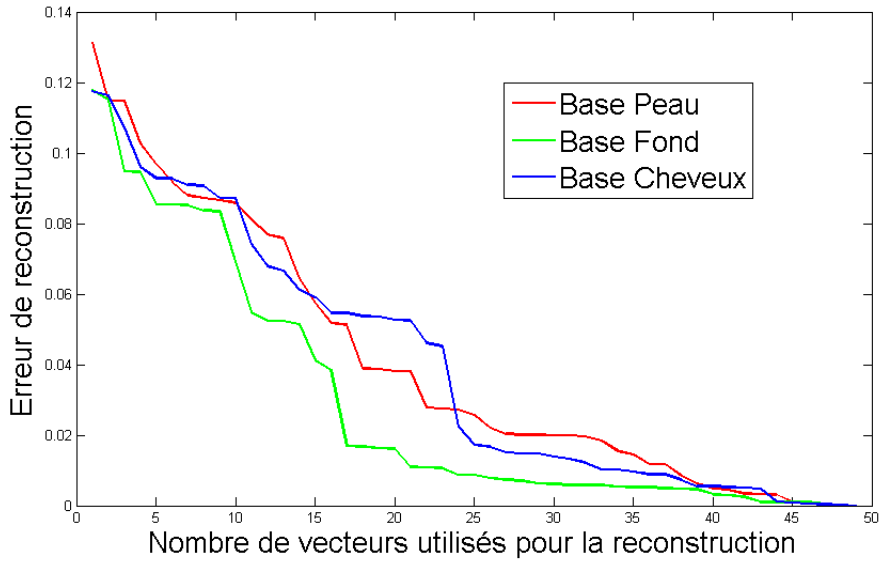


FIGURE 3.8 – Trois Erreurs de reconstruction d'un échantillon de taille 7x7 de Fond en utilisant la famille de vecteurs liée aux données Peau, Cheveux et Fond en fonction du nombre de vecteurs utilisés.

Sur la Figure 3.8, nous représentons l'évolution des erreurs résiduelles (équation 3.12) d'un échantillon de fond en fonction du nombre k de vecteurs utilisés pour les bases B^p, B^c et B^f. Nous constatons que pour un échantillon de fond, les premiers vecteurs de la base B^f minimisent plus vite l'erreur résiduelle que ceux des deux autres bases. Dans cet exemple plusieurs valeurs de k ($k \in [2 \cdots 40]$) vérifient les conditions suivantes : $E^f < E^p$ et $E^f < E^c$. Le descripteur de texture composé des trois erreurs de reconstruction E^p, E^c et E^f peut être vu comme l'entrée d'un classifieur élémentaire. En effet l'étiquette $C(\omega)$ d'un pixel peut être directement déduite de l'erreur minimale de reconstruction. Formellement, nous pouvons écrire :

$$\hat{C}(\omega) = \arg \min_{i=\{p,c,f\}} \left(E^i(\omega) \right).$$

La valeur des descripteurs E^i dépend du choix de k. L'objectif est de choisir k pour optimiser la capacité de généralisation de la classification (et non la minimisation de l'erreur de reconstruction).

Pour cela, nous allons, pour la deuxième fois, faire appel aux zones Z^i pré-définies par l'utilisateur. Nous distinguons deux possibilités pour définir la valeur de k. La pre-

mière possibilité consiste à utiliser l'ensemble des données fournies par les zones Z^i à la fois comme base de données d'apprentissage mais également comme base de données de test, on risque alors un sur-apprentissage. Le principal reproche que l'on peut adresser à cette estimation est qu'elle est fortement biaisée car les données sont à la fois données d'apprentissage et données de test.

La deuxième possibilité , appelée validation croisée, vise à s'affranchir du biais cité précédemment, en séparant les zones Z^i en deux types de données : les données d'apprentissage et les données de test. Le choix des données d'apprentissage et de test s'inscrit dans un processus itératif qui, étape par étape, modifie le choix des données d'apprentissage et de test (*cf.* Figure 3.9). Les données sont séparées en l-groupes, et tour à tour chaque groupe de données forme des données de test et le restant des données est utilisé pour l'apprentissage.

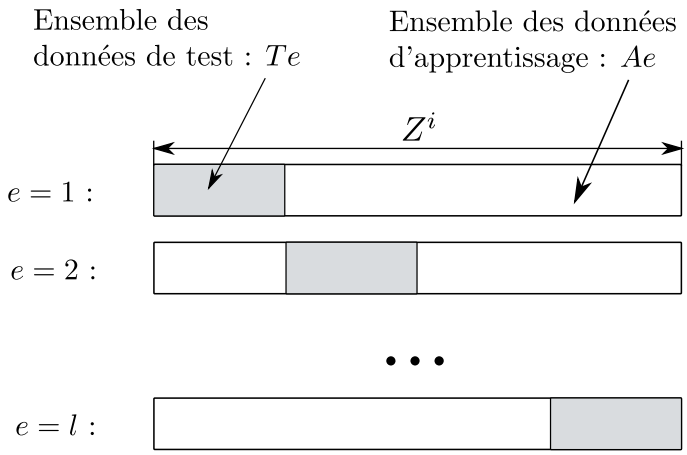


FIGURE 3.9 – Exemple de séparation des données pour l-groupes

En pratique, nous travaillons avec une séparation en 10-groupes (*cf.* Figure 3.10)

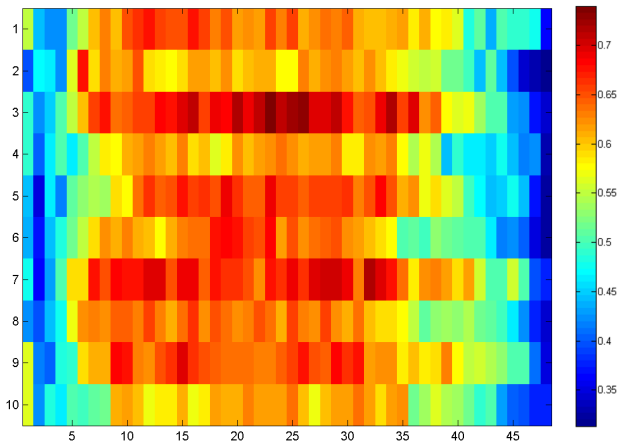


FIGURE 3.10 – Taux de bonnes classification avec en abscisse le nombre de vecteurs utilisés (pour un échantillon de taille 7x7) et en ordonnée $l = 10$ jeux de données d'apprentissage et de test.

La Figure 3.10 illustre la qualité de classification, en bleue les données de tests ont plutôt mal répondues aux conditions de classification tandis qu'en rouge, nous pouvons

voir les données qui ont le mieux répondu. Nous constatons très rapidement, que la valeur optimale de k pour la généralisation de la classification n'est ni trop petite ni trop grande. Cela semble raisonnable car pour des valeurs trop petites, le signal reconstruit manque cruellement d'informations. Au contraire pour des valeurs trop élevées de k seul le bruit ne serait pas reconstruit et il semble déraisonnable de classifier des signaux en travaillant sur le bruit. La figure 3.10 permet donc d'estimer la valeur de k correspondant au nombre optimal de vecteur à prendre en compte pour l'ensemble des bases B^i. Ici nous travaillons sur un échantillon de taille 7×7. Le critère de choix issu de la validation croisée, noté CV, doit permettre de maximiser le taux de bonne classification des pixels appartenant aux zones Z^p, Z^c et Z^f.

$$CV_k = \frac{1}{l} \sum_{e=1}^{l} \sum_{\omega \in Ne} \delta\left(\hat{C}(\omega), C^*(\omega)\right),$$

où l est le nombre de groupe de jeux de données d'apprentissage et de tests, e l'indice de l'étape de test, Ne le nombre d'échantillons contenus dans l'étape e de test, $\hat{C}(\omega)$ est l'étiquette de l'échantillon centré en ω, obtenue après classification via nos descripteurs avec k fixé, et enfin $C^*(\omega)$ est la valeur de l'étiquette fournit par la vérité terrain.

Si $\hat{C}(\omega) = C^*(\omega)$, alors l'étiquette obtenue après classification est identique à celle de la vérité terrain, nous avons $\delta\left(\hat{C}(\omega), C^*(\omega)\right) = 1$, sinon $\delta\left(\hat{C}(\omega), C^*(\omega)\right) = 0$.

Nous calculons donc pour chaque valeur de k la valeur du critère de validation croisée CV_k. La valeur de k optimale pour la généralisation de la classification correspondant à celle qui maximise ce critère.

A ce stade nous venons de construire un descripteur de texture fondé sur l'idée de construire une base adaptée au signal dans laquelle, nous choisissons le nombre de vecteurs optimaux à employer pour optimiser la classification. Ces travaux sont proches de ceux proposés par Mairal *et al.*. Dans la prochaine section, nous proposons de les comparer.

3.2.3 Comparaison des descripteurs de texture

Mairal *et al.* [52], dont nous avons présenté le travail précédemment, illustrent la qualité de leur descripteur en proposant une segmentation d'images en niveaux de gris. Cette méthode s'appuie sur la comparaison des résultats de reconstruction obtenus avec les dictionnaires des différentes classes.

Pour comparer les descripteurs de Mairal *et al.* avec les notres, nous proposons d'utiliser le critère de segmentation fondé sur la minimisation de l'erreur de reconstruction. La comparaison des descripteurs sera induite par la comparaison des résultats de chacune des cartes de segmentation.

3.2.3.1 Protocole d'évaluation

En règle générale, évaluer la qualité d'une segmentation sur différentes images similaires ou comparer deux segmentations de la même image, n'est pas une chose aisée.

Dans [74] Unnikrishnan et Hebert proposent de nouveaux critères d'évaluation et de comparaison de résultats de segmentation. Ils travaillent sur la comparaison des résultats de segmentation à une ou plusieurs segmentations manuelles. Les auteurs soulignent le

fait que pour une même image, on peut obtenir plusieurs segmentation manuelles. Ils proposent une nouvelle fonction de similarité qui prend en compte l'ambiguïté due aux divers résultats de segmentation possibles pour une même image et la propriété qu'à proximité des frontières de régions l'erreur d'étiquetage est moins importante.

Dans [7] Chabrier *et al.* proposent une étude sur les critères d'évaluation qui leur ont semblé les plus pertinents, dans le cadre de la segmentation d'images. L'objet de cette étude est de comparer les critères afin d'évaluer leur efficacité ainsi que leur potentielle complémentarité. Une perspective de ce travail est de combiner les meilleurs critères d'évaluation, afin d'optimiser leur utilisation dans différents contextes.

Dans nos travaux, nous proposons un critère d'évaluation fondé sur une approche supervisée. En effet, un expert créé une vérité terrain en segmentant manuellement les images. Dans les images naturelles, la résolution, le contraste, la forme des frontières varient en fonction du protocole d'acquisition. Pour un problème de segmentation, les structures à détecter sont plus ou moins visibles en fonction de leur taille et de leur contraste. : une variabilité intra- et inter-expert est donc présente, et ne facilite pas l'établissement d'un résultat optimal dont l'algorithme de segmentation automatique doit se rapprocher, même si de récentes publications montrent que l'établissement d'un tel étalon est possible.

Nous comparons les résultats des descripteurs de texture en deux temps :
- la qualité de classification,
- le caractère discriminant des dictionnaires (mesure d'ambigüité).

Qualité de classification :

La qualité de classification des pixels de l'image est mesurée à partir des termes de rappel/précision. Ces termes sont définis à partir des données issues d'une vérité terrain et du résultat de la classification.

Le rappel mesure le nombre de pixels correctement détectés dans l'image au regard du nombre de pixels définis par la vérité terrain. La précision est définie par le nombre de pixels correctement détectés rapporté au nombre total de pixels détectés.

Les indicateurs globaux de rappel et de précision d'une classification en 3 classes sont respectivement définis comme la moyenne des indicateurs de rappel et de précision sur les classes.

Une classification idéale détecte la totalité des pixels qui auraient dû être détectés (rappel = 1) et ne se trompe pas dans la classification (précision = 1). En pratique, nous souhaitons à la fois être précis (maximiser l'indicateur de précision) et performant (maximiser l'indicateur de rappel). Pour synthétiser l'information contenue dans ces deux indicateurs en une seule mesure, nous proposons d'utiliser la F-mesure [75].

$$F_\beta = \frac{(1+\beta^2) \times (\text{rappel} \times \text{précision})}{\beta^2 \times \text{précision} + \text{rappel}}$$

Pour avoir un indicateur qui mesure autant la précision que la pertinence de la classification, nous posons $\beta = 1$.

$$F_1 = \frac{2 \times \text{rappel} \times \text{précision}}{(\text{rappel} + \text{précision})}$$

Des valeurs de β en valeurs absolues supérieures à 1 conduiraient à privilégier le terme de rappel par rapport au terme de précision. Si la valeur de β en valeur absolue est inférieure

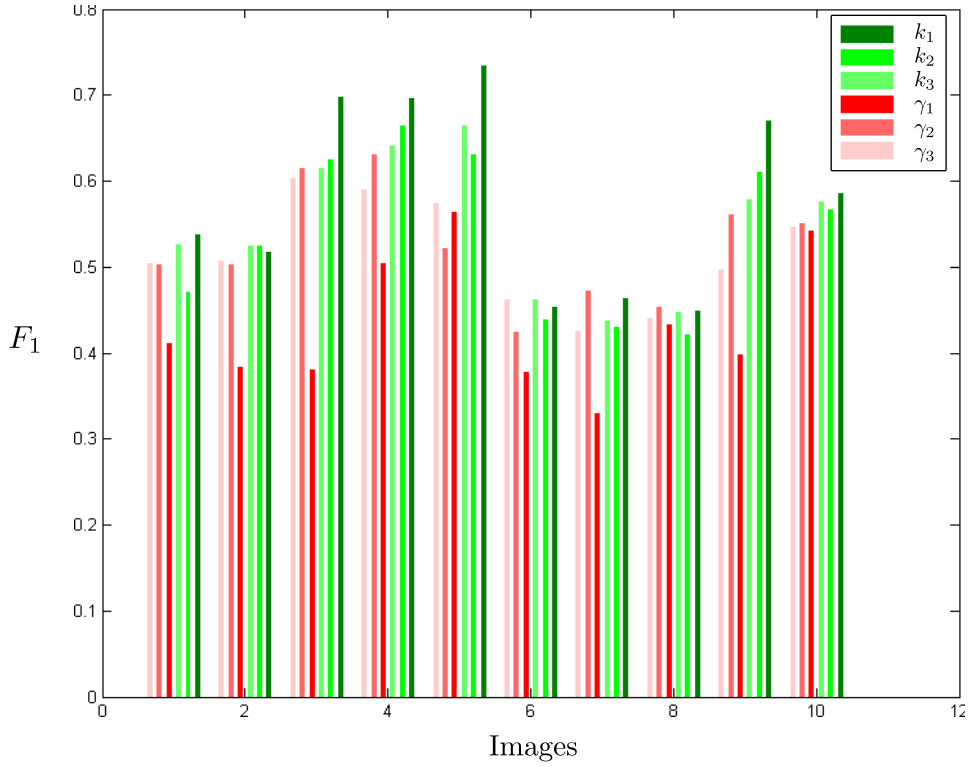


FIGURE 3.11 – Évaluation de la qualité de 6 segmentations via l'indicateur F_1 sur un échantillon de 10 images : 3 segmentations obtenues par ACP pour 3 valeurs de k et 3 segmentations obtenues en utilisant la construction de dictionnaires discriminants pour 3 valeurs du poids γ.

à 1 alors ce serait le terme de précision qui serait mis en avant par rapport au terme de rappel.

Plus F_1 s'approche de la valeur 1 meilleurs seront les résultats de la classification.

La Figure 3.11 illustre (en ordonnée) la qualité de la classification pour un échantillon de 10 images (en abscisse). Sur chacune des images, nous comparons 6 résultats de segmentation grossière obtenue par classification. Les trois résultats illustrés en vert correspondent à notre méthode de description de texture par ACP testé pour différentes valeurs de k (nombre de vecteurs propres utilisés avec le classifieur introduit à la section 3.2.2). Les trois autres résultats illustrés en rouge sont obtenus en appliquant la méthode de construction de dictionnaires discriminants due aux travaux de Mairal *et al.* en testant différentes valeurs de γ (poids permettant de gérer le degré d'implication du terme de discrimination par rapport au terme d'attache aux données, *cf.* équation 3.9).

– **Notre méthode par ACP** (en vert) : Les 3 résultats de classification sont paramétrés par la quantité de vecteurs propres que nous utilisons pour la caractérisation de la texture. Le paramètre k_1 correspond au nombre de vecteurs obtenus par validation croisée sur les zones Z^p, Z^c et Z^f. Les valeurs attribuées à k_2 et k_3 sont choisies de manière empirique. Le paramètre k_2 vaut la largeur de l'échantillon ($k_2 = \sqrt{n}$) et k_3 le double de cette largeur ($k_3 = 2\sqrt{n}$).
  Nous constatons que parmi ces trois résultats, la classification obtenue en utilisant k_1 est meilleure ou comparable aux autres résultats, au sens de l'indicateur F_1.
– **La méthode des dictionnaires discriminants** (en rouge) : Les 3 résultats sont pa-

ramétrés par le poids γ qui permet de pondérer l'erreur résiduelle et le score de discrimination des dictionnaires. Plus γ est grand plus on favorise la minimisation de l'erreur résiduelle et moins le caractère discriminant des dictionnaires est important. Les valeurs de γ testées sont hiérarchisés de la manière suivante : $\gamma_1 << \gamma_2 << \gamma_3$. Nous constatons que lorsque γ est très petit la qualité de la classification est faible cela s'explique par le fait que la fonctionnelle ne prend pas assez en compte le terme lié à l'erreur résiduelle. Pour γ_2 et γ_3 la qualité de classification s'améliore mais reste inférieure à notre proposition.

En résumé, la classification qui donne les meilleurs résultats sur la qualité de la classification (au sens de l'indicateur F_1) pour un échantillon de données composé des 10 images est obtenue avec notre modélisation par ACP dont le nombre de vecteurs propres est défini par la méthode de validation croisée. Nous nous posons aussi la question de savoir si la décision lors de la classification est ambigüe ou pas.

L'ambigüité de classification :

L'ambigüité permet de qualifier l'erreur que l'on a pu commettre en étiquetant le pixel avec la deuxième classe la plus pertinente. Plus formellement, nous mesurons l'ambigüité associée au pixel ω à partir de la variable $A(\omega)$ défini par la formule :

$$A(\omega) = |E^1(\omega) - E^2(\omega)|$$

Avec $E^1(\omega)$ et $E^2(\omega) \in \mathbb{R}$ les erreurs de reconstruction en ω en utilisant respectivement la classe la plus pertinente et la deuxième la plus pertinente. Plus la valeur de A est grande, moins la classification est ambigüe. Nous calculons la valeur de A sur l'ensemble des pixels des 10 images et nous ré-ordonnons les valeurs de l'ambigüité A par ordre décroissant. La Figure 3.12 présente la critère d'ambigüité. Sur l'axe des ordonnées, nous présentons la valeur du critère en un pixel. Pour des questions de lisibilité du critère nous ordonnons l'ensemble des pixels de toutes les images suivant l'ordre décroissant de la valeur de $A(\omega)$. Sur l'axe des abscisses nous avons donc l'indice du pixel après ordonnancement. Cette figure montre que la méthode la moins ambigüe (valeurs de A les plus grandes) est celle utilisant les dictionnaires discriminants dans la configuration où γ est faible (*cf.* courbe rouge γ_1). Ce résultat est raisonnable car il correspond à une mise en équation appliquant une faible contribution du terme énergétique associé à l'erreur de reconstruction, donc *a fortiori* un plus fort poids pour le caractère discriminant des dictionnaires ce qui entraîne une réduction de l'ambigüité.

Notre méthode arrive en deuxième position sur le critère d'ambigüité avec k_1 optimisé par validation croisée.

Si nous associons le critère sur la qualité de la classification et celui sur l'ambigüité, nous constatons que la méthode la moins ambigüe est également celle qui donne la moins bonne classification. Cette méthode est la méthode développée par Mairal *et al.* pour $\gamma = \gamma_1$. En revanche, la méthode qui obtient le meilleur compromis entre qualité de classification et ambigüité, est la notre. De plus, nous notons que notre méthode est directe, contrairement à celle proposée par Mairal *et al.* qui doivent définir la valeur de la constante γ et de la suite λ.

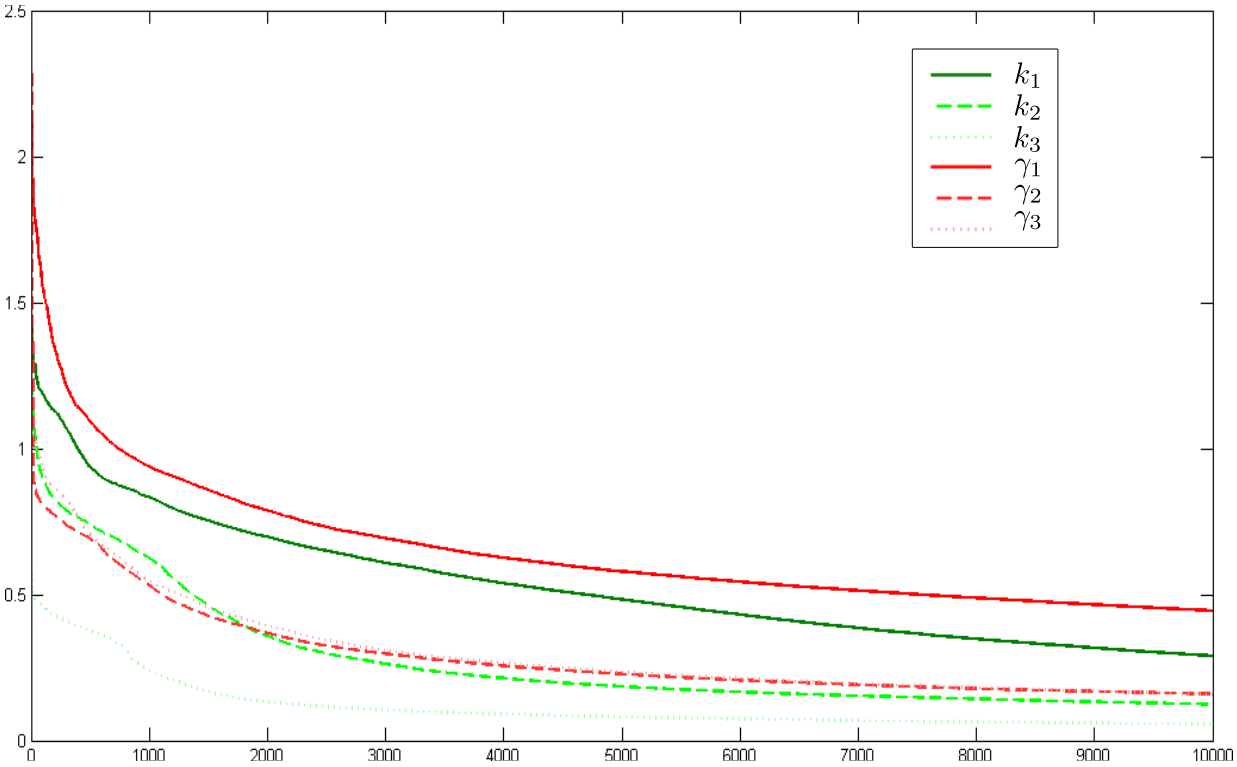


FIGURE 3.12 – Évaluation de l'ambigüité de 6 segmentations.

3.3 Modélisation et segmentation

Dans le chapitre 2 à la section 2.1.2.1, nous avons présenté une série d'évaluations des nombreux espaces de couleur pour la détection de peau. Si certains espaces offrent des performances clairement en retrait, plusieurs espaces de couleur se comportent de manière similaire pour la tâche considérée. Pour notre cas d'étude, il est difficile de statuer sur le choix d'un espace de couleur, puisque nous souhaitons exprimer simultanément les caractéristiques colorimétriques de la peau, des cheveux et du fond dans les images à forte variabilité. Nous proposons d'utiliser l'espace *YCbCr* qui est adapté à la détection de la Peau. De plus, la séparation de la chrominance et de la luminance, permet de travailler uniquement avec la chrominance et ainsi de s'affranchir de l'illumination de la scène (*cf.* Figure 3.13).

En résumé, pour chaque pixel de l'image $\omega \in \Omega$, nous notons $X_c(\omega)$, les composantes colorimétrique du descripteur, et $X_t(\omega)$ les composantes texturales. Les composantes colorimétriques sont définies par :

$$\begin{array}{rcl} X_c : \Omega & \longrightarrow & \mathbb{R}^2 \\ \omega & \longmapsto & X_c(\omega) = \begin{bmatrix} C_b(\omega) \\ C_r(\omega) \end{bmatrix} \end{array} \tag{3.13}$$

Les composantes texturales sont définies par les trois composantes représentant les erreurs résiduelles obtenues à partir des k premiers vecteurs des trois bases issues de

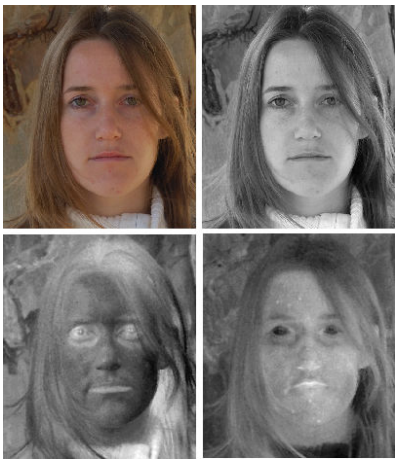


FIGURE 3.13 – Décomposition de l'image (en haut à gauche) en les canaux Y (en haut à droite), Cb (en bas à gauche) et Cr(en bas a droite).

notre méthode en ACP : B^p, B^c et B^f. Pour chaque pixel $\omega \in \Omega$, nous notons :

$$\begin{array}{rcl} X_t : \Omega & \longrightarrow & \mathbb{R}^3 \\ \omega & \longmapsto & X_t(\omega) = \begin{bmatrix} E^p(\omega) \\ E^c(\omega) \\ E^f(\omega) \end{bmatrix} \end{array} \tag{3.14}$$

Après avoir modélisé les données couleur et texture, nous combinons nos résultats en construisant une image vectorielle, notée X, provenant de la concaténation de X_c et X_t :

$$X = \begin{bmatrix} X_c \\ X_t \end{bmatrix} \qquad X \in \mathbb{R}^5. \tag{3.15}$$

A cette étape de l'étude, nous venons de construire en chaque pixel ω un descripteur contenant de l'information de couleur et de texture. Par construction, puisque nous avons caractérisé la texture des trois classes en utilisant les zones *a priori*, cette information est adaptée aux caractéristiques de l'image.

Ces zones *a priori* vont une nouvelle fois intervenir pour construire le modèle décisionnel qui nous permettra d'obtenir la carte de décision finale C. Prendre une décision sur la classe d'appartenance d'un pixel se décompose en deux phases. Nous allons d'abord utiliser un modèle pour représenter la distribution des descripteurs sur les trois classes puis, nous définirons un outils de mesure qui permet de statuer sur la classe la plus adaptée au descripteur.

3.3.1 Modélisation de la distribution des descripteurs

Dans cette section, nous présentons et argumentons notre choix sur la modélisation des distributions de chacune des classes, et plus précisément les distributions des descripteurs de dimensions $d = 5$ conditionnelles aux classes : Peau, Fond et Cheveux.

Nous avons choisi des fonctions densité de probabilité gaussiennes que nous notons G^p, G^f, G^c pour chaque classe Peau, Fond et Cheveux.

$$G^i(\mathbf{x}, \mathbf{m_i}, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{(\mathbf{x}-\mathbf{m_i})^t\Sigma_i^{-1}(\mathbf{x}-\mathbf{m_i})}{2}\right) \tag{3.16}$$

Nos expériences montrent que les distributions des vecteurs caractéristiques X_c et/ou X_t sont suffisamment uni-modales pour ne pas trop s'écarter d'une distribution gaussienne. Un test du χ^2 permet de valider cette hypothèse. Des lois statistiques plus sophistiquées (modélisation par mélange de gaussiennes par exemple) pourraient aisément s'intégrer dans notre modélisation. En utilisant le vecteur caractéristique X (défini dans la section 3.3) et les zones Z^p, Z^c, Z^f, nous apprenons les valeurs des paramètres $\mathbf{m_i}, \mathbf{\Sigma_i}$ des trois distributions gaussiennes décrivant les classes Peau, Cheveux et Fond. Nous supposons que chacune des zones *a priori* contient un échantillon représentatif de la distribution de sa classe associée.

3.3.2 Les modèles décisionnels

Dans cette section, nous présentons trois types de modèles décisionnels que l'on enrichit étape par étape. Nous comparerons leurs résultats de classification.

Nous commençons par proposer une segmentation pixel à pixel avec un modèle bayésien en utilisant uniquement les propriétés des distributions des descripteurs. Dans une deuxième phase, nous ajoutons un *a priori* global sur la décision. Enfin, nous présentons un modèle décisionnel complet qui reprend les étapes citées précédemment et qui prend en compte le contexte local pour réduire le bruit sur la carte de décision.

3.3.2.1 Le modèle décisionnel sans *a priori*

Nous sommes en possession de trois fonctions densité de probabilité G^p, G^c, G^f et du descripteur X associé à l'image I. A partir de ces données, nous pouvons décider pour chaque pixel $\omega \in \Omega$ (domaine de l'image) une étiquette Peau, Cheveux ou Fond à partir de la probabilité *a posteriori*.

$$\begin{aligned} P[C^i(\omega)|X(\omega)] &= \frac{p(X(\omega)|C^i(\omega))\ P[C^i(\omega)]}{\sum\limits_{i\in\{p,c,f\}} p(X(\omega)|C^i(\omega))P[C^i(\omega)]} \qquad \forall i \in \{p,c,f\} \\ &\propto p(X(\omega)|C^i(\omega))\ P[C^i(\omega)] \end{aligned} \tag{3.17}$$

$P[C^i(\omega)|X(\omega)]$ est la probabilité *a posteriori* de la classe $C^i(\omega)$ ayant mesuré le vecteur descripteur $X(\omega)$.

$p(X(\omega)|C^i(\omega))$ est la fonction de vraisemblance des mesures $X(\omega)$ conditionnelle à la classe $C^i(\omega)$ ou fonction densité de probabilité de la distribution de la classe C^i appliquée en $X(\omega)$.

$P[C^i(\omega)]$ est la probabilité *a priori* de la classe C^i. Dans cette sous-section, on s'abstrait de tout *a priori*, nous supposons que pour chaque classe C^i, la probabilité *a priori* $P[C^i(\omega)] = \frac{1}{3}$ est équiprobable.

Le théorème de Bayes nous permet d'obtenir trois cartes de probabilité : $P[C^p|X(\omega)]$, $P[C^c|X(\omega)]$ et $P[C^f|X(\omega)]$. En appliquant la décision Bayésienne simple, nous pouvons définir la carte de décision finale C comme étant :

$$C(\omega) = i^*$$

avec

$$E = \underset{i^*}{\text{argmax}} P[C^i(\omega)|X(\omega)].$$

(a) Image à segmenter (b) Segmentation sans *a priori*

FIGURE 3.14 – Résultat de segmentation

Sur la figure 3.14, nous présentons un résultat de segmentation. Les pixels en surbrillance rouge sont classifiés Peau, les pixels en surbrillance verte sont classifiés Cheveux et les pixels en surbrillance bleue sont classifiés Fond. Cette méthode de segmentation/classification, bien que naïve, présente déjà des résultats très intéressants. Nous constatons que les erreurs de classification sont peu nombreuses entre le Fond et les Cheveux et pourtant ces deux classes sont colorimétriquement très proches. Le descripteur de texture joue très bien son rôle en séparant les deux classes.

Enfin de réduire les aberrations de classification (classifier Fond un pixel au milieu du visage), dans la section suivante, nous mettons en place un modèle décisionnel prenant en compte un *a priori* global de la scène.

3.3.2.2 Le modèle décisionnel avec un *a priori* global

Afin de réduire, voire supprimer, les aberrations de classification qui sont par exemple de classifier les pixels comme étant du Fond au milieu de pixels étiquetés Peau, nous introduisons dans le modèle décisionnel des cartes de probabilité *a priori* de segmentation pour chaque classe. Pour cela, nous avons une première phase d'apprentissage hors ligne des cartes *a priori* de segmentation. Dans un deuxième temps, nous discutons de l'insertion de l'*a priori* dans le modèle décisionnel.

Ces cartes de probabilité *a priori* de segmentation sont issues d'une phase d'apprentissage de cartes de segmentation manuelles de Peau, Cheveux et Fond.

Apprentissage des cartes *a priori* de segmentation Nous construisons une base de données d'images représentatives. Elle est composée de 100 images de Portraits faisant face à la caméra, et dont la variabilité des coupes de cheveux est représentative. Nous devons être en mesure de traiter des coupes courtes comme des coupes longues de cheveux. Du fait de cette grande variabilité, nous décidons, comme dans l'article de Lee *et al.* [45] de construire un modèle *a priori* de segmentation. Plus précisément, nous reconstruirons trois modèles : un pour les cheveux courts, un pour les cheveux mi-long et un pour les cheveux longs. Le choix de trois modèles, nous est apparu comme un bon compromis entre l'ajout de complexité (le nombre de cartes) et la fidélité à la modélisation des données.

Pour établir les cartes *a priori* de probabilité de chaque modèle, nous recensons les images de la base de données qui correspondent au modèle. Puis, nous appliquons à l'ensemble des images des homographies qui normalisent géométriquement toutes les images par rapport à la position des yeux. Cette étape sera décrite ou précisée à la section 5.2.1 du Chapitre 5 . Cette étape est nécessaire à l'exploitation des données, elle les rend comparables entre elles. Ensuite, nous segmentons manuellement en trois régions les images normalisées : Peau, Cheveux et Fond. Nous récupérons pour chaque image et pour chaque classe deux cartes de segmentation correspondant à la segmentation manuelle et à son image miroir. Enfin, le calcul de la carte de probabilité *a priori* pour chaque classe C^i, noté $P(C^i(\omega))$, est le ratio pixel à pixel entre le nombre d'images qui étiquettent C^i le pixel ω et le nombre total d'images.

Pour les trois modèles : cheveux courts, cheveux mi-longs et cheveux longs, nous obtenons les trois cartes de probabilité présentées sur la Figure 3.15. Chaque ligne correspond à un modèle et chaque colonne à une classe.

Nous constatons que la base d'image sélectionnée pour le modèle cheveux courts ne semble pas encoder le sous ensemble de la chevelure de type "frange" contrairement à un modèle de cheveux mi-longs ou longs. Cela semble raisonnable car les cheveux courts sont en grande majorité portés par les garçons et ces derniers ne possèdent le plus souvent pas de frange.

La mise en équation du modèle décisionnel avec un *a priori* global Si nous réutilisons l'équation 3.17, en prenant en compte les modèles *a priori* $P_M(C^i(\omega))$; l'équation devient :

$$P[C^i(\omega_j)|X(\omega_j), M] = \frac{p(X(\omega)|C^i(\omega))\ P_M[C^i(\omega)]}{\sum_{i\in\{p,c,f\}} p(X(\omega)|C^i(\omega))\ P_M[C^i(\omega)]}. \quad (3.18)$$

La figure 3.16(c) illustre le résultat de la classification pixel à pixel utilisant un *a priori* global. Nous pouvons constater que les résultats de classification se sont améliorés par rapport à la segmentation naïve sans *a priori*. Par exemple les éléments du visage comme le nez, les yeux sont maitenant étiquétés Peau plutôt que Cheveux. Toutefois, les cartes de Peau, Cheveux et Fond restent "bruitées". Pour éliminer cela, nous proposons, dans le paragraphe suivant, d'ajouter un terme de régularisation spatiale des étiquettes.

3.3.2.3 Le modèle décisionnel avec un *a priori* global et une régularisation

Régularisation des cartes de probabilité *a posteriori* Dans cette partie, nous disposons des trois cartes de probabilité *a posteriori* notées C^p, C^c et C^f. Elles sont obtenues en appliquant une formulation bayésienne pixel à pixel et un *a priori* global décrit dans la section précédente. Jusque là cette technique ne prend pas en compte les informations du voisinage. Ici, nous proposons de réduire les erreurs d'étiquetage, notamment des décisions incertaines, en utilisant le voisinage dans le but de limiter les incohérences de classification. Par exemple, il semble absurde de classifier un pixel comme étant de la Peau lorsque, dans son voisinage, tous les pixels sont étiquetés Fond. Pour minimiser l'apparition de ces incohérences, nous choisissons de post-traiter les cartes en introduisant la notion de voisinage. Nous choisissons de lisser c'est à dire régulariser les probabilités obtenues là où nous aurions pu également utiliser des champs de Markov pous régulariser les étiquetages. Notre but était d'obtenir un modèle dont la résolution soit la plus rapide

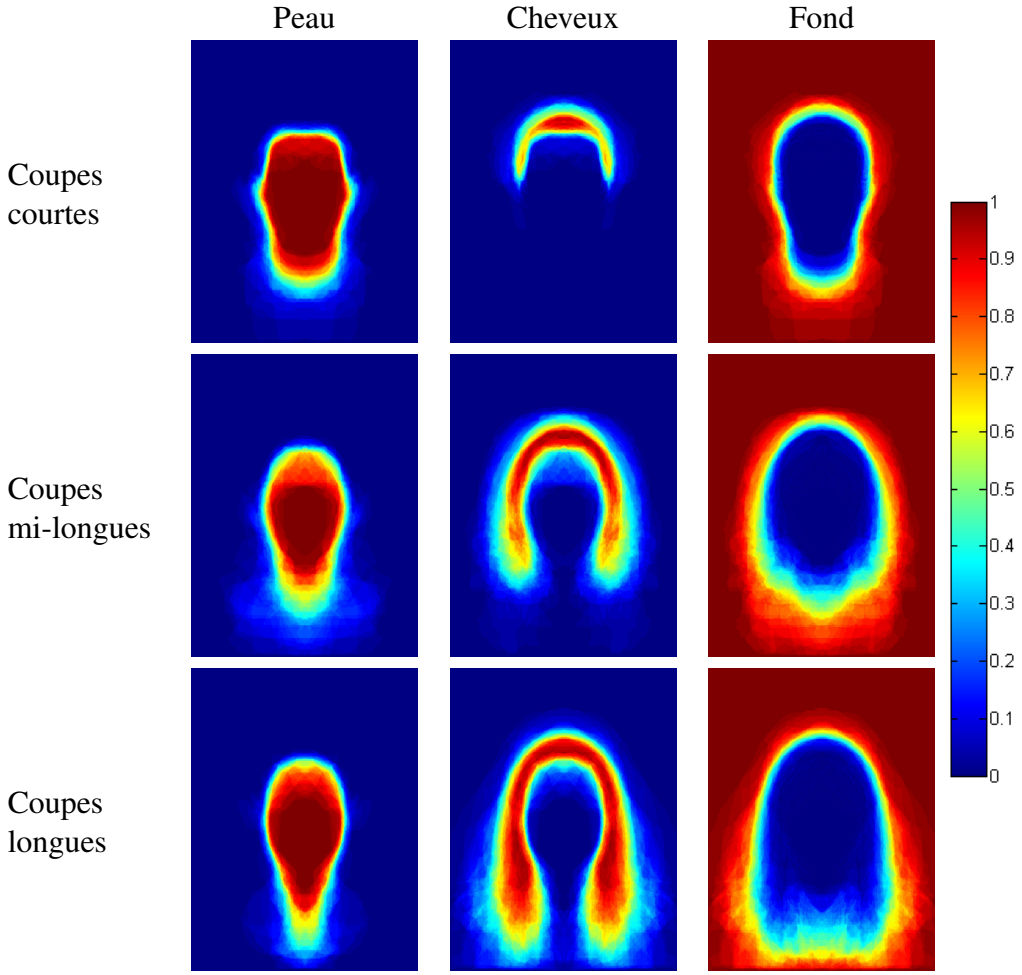


FIGURE 3.15 – Nous présentons sur la première colonne les cartes de probabilité *a priori* de Peau, sur la deuxième celles de Cheveux et celles de Fond. Sur la première ligne nous présentons les trois cartes à priori pour une modélisation de type cheveux courts. Sur la deuxième ligne, c'est une modélisation pour les cheveux mi-long et sur la dernière ligne c'est une représentation pour les cheveux longs.

possible et les approches par champ de Markov sont souvent peu performantes dans ce domaine.

L'objectif est de lisser la carte de décision C en régularisant les cartes de probabilité *a posteriori*, notés R^p, R^c, R^f en faisant intervenir un terme de voisinage.

Nous devons nous assurer que les versions régularisées des cartes de probabilité *a posteriori* restent des cartes de probabilités ce qui implique que des contraintes de positivité et de somme à 1 des éléments de R^i sont bien respectées. La différence entre les cartes C^i et R^i vient du fait que les R^i doivent prendre en compte le voisinage de chaque pixel et donc minimiser la différence entre un pixel et son voisinage tout en gardant les cartes R^i semblable à C^i. Dans la suite, nous définissons une énergie à optimiser puis nous développons la méthode de résolution. Enfin nous présentons des résultats de régularisation.

Les nouvelles cartes notés R^i doivent remplir les conditions suivantes :

1. La matrice R^i doit être semblable à C^i

(a) Image à segmenter (b) Segmentation sans *a priori* (c) Segmentation avec un *a priori* global

FIGURE 3.16 – Résultat de classification en 3 classes : Peau, Cheveux et Fond.

2. La valeur de $R^i(\omega)$ doit dépendre de la valeur de ses quatre plus proches voisins.
3. La matrice R^i doit être à valeurs positives.
4. Les coefficients à l'emplacement ω des matrices R^i doivent se sommer à 1.

On définit le terme d'attache aux données de la manière suivante :

$$E_1(R^p, R^c, R^f) = \sum_{R^i=\{R^p,R^c,R^f\}} \sum_{\omega\in\Omega} (R^i(\omega) - C^i(\omega))^2 \tag{3.19}$$

On définit le terme de lissage en prenant en compte les 4 plus proches voisins, nous obtenons :

$$E_2(R^p, R^c, R^f) = \sum_{R^i=\{R^p,R^c,R^f\}} \sum_{\omega\in\Omega} \sum_{\omega'\in\mathcal{N}(\omega)} (R^i(\omega) - R^i(\omega'))^2 \tag{3.20}$$

On note $\mathcal{N}(\omega)$ les 4 plus proche voisins de ω. Pour le pixel de coordonnées (i, j), $\mathcal{N}(i, j) = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$.

Sans contrainte, l'énergie s'écrit :

$$E(R^p, R^c, R^f) = \sum_{i=1}^{N} \sum_{\omega\in\Omega} (R^i(\omega) - C^i(\omega))^2 + \alpha \sum_{i=1}^{N} \sum_{\omega\in\Omega} \sum_{\omega'\in\mathcal{N}(\omega)} (R^i(\omega) - R^i\omega'))^2 \tag{3.21}$$

Nous avons vu précédemment que les contraintes étaient nécessaires pour garantir l'obtention de trois cartes de probabilité. La première contrainte de somme à 1 s'écrit $\sum_i R^i(\omega) = 1$. On introduit cette dernière dans l'énergie en utilisant les multiplicateurs de Lagrange.

$$\begin{aligned} E(R^p, R^c, R^f) &= \sum_{i=1}^{N} \sum_{\omega\in I} (R^i(\omega) - C^i(\omega))^2 + \alpha \sum_{i=1}^{N} \sum_{\omega\in I} \sum_{\omega'\in\mathcal{N}(\omega)} (R^i(\omega) - R^i(\omega'))^2 \\ &+ \sum_{\omega\in I} \beta(\omega) \left(\sum_{i=1}^{N} R^i(\omega) - 1 \right) \end{aligned} \tag{3.22}$$

Pour que les terme en $(R^i)^2$ restent comparables au terme en R^i, nous avons choisit d'introduire un poids variable β qui dépend de R^i et donc à fortiori de la position du pixel ω. Ce poids sera calculé par la suite (*cf.* équation 3.24).

Pour la contrainte $R^i \in [0,1]$, nous utilisons une fonction barrière. Pour éviter que la minimisation de l'équation 3.22 entraîne un lissage semblable à un lissage gaussien sur les cartes régularisée, nous proposons une fonction barrière qui pénalise fortement les valeurs intermédiaires de $R^i(\omega)$.

La fonction "barrière" $g : x \longmapsto \frac{1}{x^2(1-x)^2}$ permet de forcer x à être proche de 0 et 1 (*cf* figure 3.17).

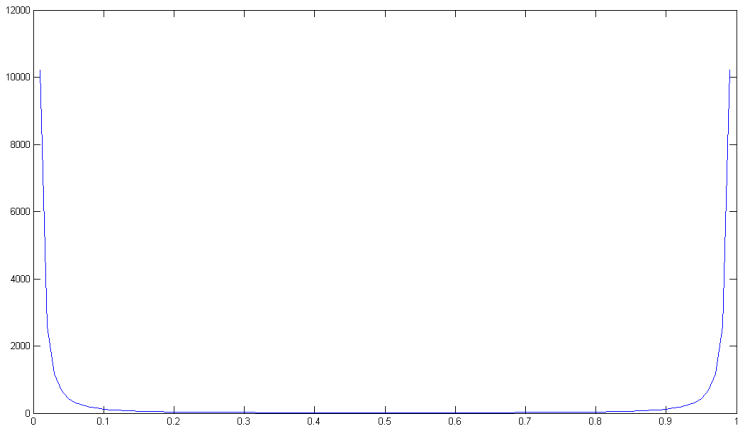


FIGURE 3.17 – Représentation de la fonction barrière

L'énergie à minimiser s'écrit alors :

$$\begin{aligned} E(R^p, R^c, R^f) &= \sum_{i=1}^{N} \sum_{\omega\in\Omega} (R^i(\omega) - C^i(\omega))^2 + \alpha \sum_{i=1}^{N} \sum_{\omega\in\Omega} \sum_{\omega'\in\mathcal{N}(\omega)} (R^i(\omega) - R^i(\omega'))^2 \\ &+ \sum_{\omega\in\Omega} \beta(\omega) \left(\sum_{i=1}^{N} R^i(\omega) - 1 \right) - \gamma \sum_{i=1}^{N} \sum_{\omega\in\Omega} \frac{1}{(R^i(\omega))^2 (1 - R^i(\omega))^2} \end{aligned} \quad (3.23)$$

avec α, β et γ strictement positifs.

Nous obtenons les conditions nécessaires suivantes :

$$\begin{cases} \frac{\partial E}{\partial R^i(\omega)} = & 2(R^i(\omega) - C^i(\omega)) + 16\alpha R^i(\omega) - 4\alpha \sum\limits_{\omega'} R^i(\omega') \\ & +\beta(\omega) - \gamma \left(g'(R^i(\omega)) \right) \\ \sum\limits_{i=1}^{N} R^i(\omega) = & 1 \end{cases}$$

Les facteurs 16α et 4α sont issus du calcul de la dérivée par rapport à $R^i(\omega)$ en tout point de l'image. Nous remarquons que sur la somme des pixels de l'image, $R^i(\omega)$ peut bien évidemment avoir le rôle de pixel central ; mais aussi a son tour il aura le rôle de voisin. Lorsque $R^i(\omega)$ joue le rôle du pixel central ou de voisin, on le retrouve $4 \times 2 \times 2$ fois dans la dérivée :

- 4 : $R^i(\omega)$ intervient 4 fois dans la somme $\sum\limits_{\omega'\in\Omega}$,
- 2 : facteur issu de la dérivée de l'expression de degré égal à 2,
- 2 : facteur issu du fait que $R^i(\omega)$ joue à la fois le rôle de pixel central mais également de voisin.

De part la présence du double produit, nous retrouvons dans la dérivée un terme prenant en compte les pixels voisins $\left(\sum_{\omega' \in \Omega} R^i(\omega')\right)$, on le retrouve 2×2 fois dans la dérivée :

- 2 :facteur issu de la dérivée de l'expression de degré égal à 2,
- 2 :facteur issu du fait que $R^i(\omega)$ joue à la fois le rôle de pixel central mais également de voisin.

Afin de pouvoir résoudre le système, nous proposons une méthode itérative faisant évoluer au cours du temps les valeurs de R^i en tout point ω de l'image. Cette évolution se traduit par l'introduction d'un nouvel indice n, $R^i(\omega)$ sera maintenant noté $R^i_n(\omega)$. Les itérations visent à améliorer $R^i_n(\omega)$ en utilisant $R^i_{n-1}(\omega)$. Nous avons donc chaque terme $R^i_n(\omega)$ qui est défini en fonction de $R^i_{n-1}(\omega)$ et/ou de $R^i_n(\omega)$. nous remarquons que comme la valeur de $\beta(\omega)$ est calculée en fonction de la valeur de $R^i(\omega)$, on re-calculera $\beta(\omega)$ à chaque itération, on notera alors $\beta_n(\omega)$.

Afin de pouvoir résoudre le système, nous linéarisons la dérivée de la fonction g.

$$g'(R^i_n(\omega)) = g'(R^i_{n-1}(\omega)) + g''(R^i_{n-1}(\omega))(R^i_n(\omega) - R^i_{n_1}(\omega))$$

La minimisation du critère E s'effectue par descente de gradient $\Delta R^i = -\frac{\partial E}{\partial R}$

$$\begin{cases} \Delta R^i(\omega) = & -2(R^i_n(\omega) - C^i(\omega)) - 16\alpha R^i_n(\omega) + 4\alpha \sum\limits_{\omega'} R^i_{n-1}(\omega') \\ & -\beta_n(\omega) + \gamma \left(g'(R^i_{n-1}(\omega)) + g''(R^i_{n-1}(\omega))(R^i_n(\omega) - (R^i_{n-1}(\omega))\right) \\ \sum\limits_{i=1}^{N} R^i(\omega) = & 1 \end{cases}$$

$$\begin{cases} R^i_n(\omega) = & \frac{2C^i(\omega) + 4\alpha \sum\limits_{\omega'} R^i_{n-1}(\omega') - \beta_n(\omega) + \gamma\, g'(R^i_{n-1}(\omega)) - \gamma\, g''(R^i_{n-1}(\omega)) R^i_{n-1}(\omega)}{2 + 16\alpha - \gamma\, g''(R^i_{n-1}(\omega))} \\ \sum\limits_{i=1}^{N} R^i(\omega) = & 1 \end{cases}$$

Nous posons :

$$\begin{aligned} A^i_{n-1}(q) &= 2C^i(\omega) + 4\alpha \sum_{\omega'} R^i_{n-1}(\omega') + \gamma g'(R^i_{n-1}(\omega)) - \gamma g''(R^i_{n-1}(\omega)) R^i_{n-1}(\omega) \\ B^i_{n-1}(\omega) &= 2 + 16\alpha - \gamma g''(R^i_{n-1}(\omega)) \end{aligned}$$

$$\begin{cases} R^p_n(\omega) = \frac{A^p_{n-1}(\omega) - \beta_n(\omega)}{B^p_{n-1}(\omega)} \\ R^c_n(\omega) = \frac{A^c_{n-1}(\omega) - \beta_n(\omega)}{B^c_{n-1}(\omega)} \\ R^f_n(\omega) = \frac{A^f_{n-1}(\omega) - \beta_n(\omega)}{B^f_{n-1}(\omega)} \\ \sum\limits_{i=1}^{N} R^i(\omega) = 1 \end{cases}$$

Nous déduisons du système précédent la valeur de $\beta(\omega)$

$$\begin{aligned}\beta_n(\omega) &= \left(\frac{A_n^p(\omega)}{B_n^p(\omega)} + \frac{A_n^c(\omega)}{B_n^c(\omega)} + \frac{A_n^f(\omega)}{B_n^f(\omega)} - 1\right) \\ &\times \frac{B_n^p(\omega)B_n^c(\omega)B_n^f(\omega)}{(B_n^p(\omega) + B_n^c(\omega))(B_n^c(\omega) + B_n^f(\omega))(B_n^p(\omega) + B_n^f(\omega))}\end{aligned} \tag{3.24}$$

Le résultat obtenu pour les trois cartes régularisées R^p, R^c etR^f est illustré sur la Figure 3.18. Plus la couleur du pixel tend vers le rouge plus la probabilité est proche de 1, plus la couleur du pixel tend vers le bleu plus la probabilité est proche de 0. Les trois cartes de probabilité sont complémentaires et la transition entre les classes est quasi inexistante du fait du choix de la fonction barrière.

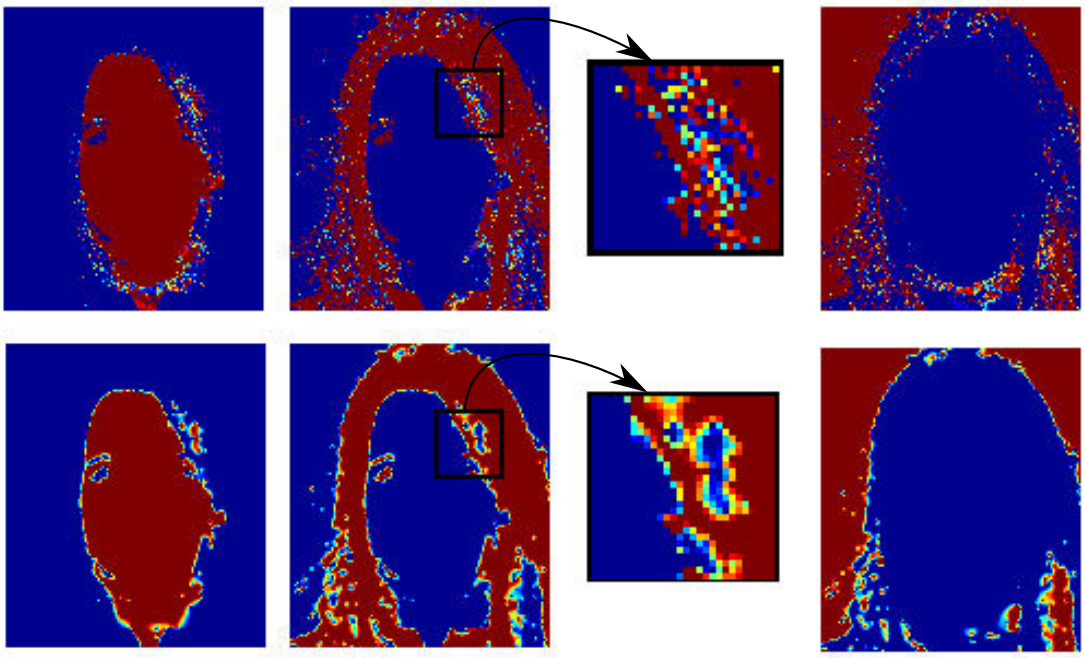
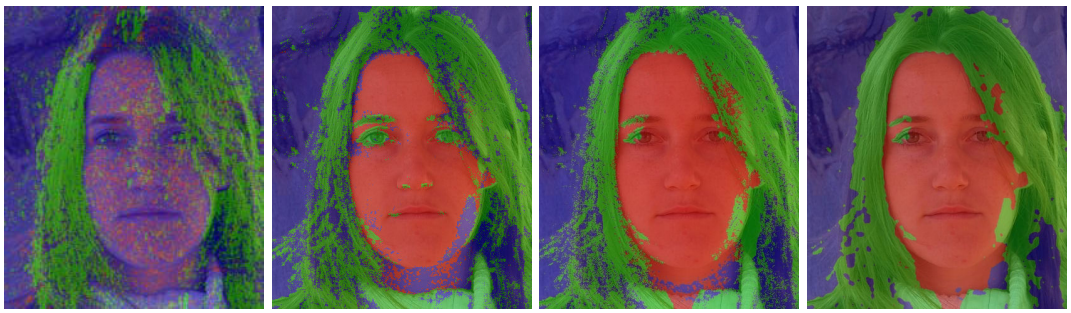


FIGURE 3.18 – Sur la première ligne, les cartes de probabilité des 3 classes Peau, Cheveux et Fond issues du processus bayésien. Sur la deuxième ligne, les résultats de la régularisation des cartes par minimisation de l'équation 3.23.

La figure 3.19(e) illustre la carte décisionnelle C obtenue après classification bayésienne avec *a priori* et régularisation. Nous pouvons constater que le "bruit" de classification a été réduit et que la segmentation finale est proche de la solution finale.

Le terme de régularisation spatial améliore grandement la segmentation. En perspective, nous devons envisager de comparer cette approche avec les méthodes basées sur les champs de Markov où leurs variantes (par exemple les *conditional random field* en s'appuyant sur l'article de Verbeek et Triggs [76]).

(a) Image à segmenter



(b) Texture uniquement (c) Texture et couleur sans *a priori* (d) Texture et couleur avec un *a priori* global (e) Texture et couleur avec un *a priori* global et une régularisation sur le voisinage

FIGURE 3.19 – Résultat de classification en 3 classes : Peau, Cheveux et Fond.

3.4 Résultats

Dans cette section, nous présentons sur un échantillon d'images les résultats de classification issus de nos travaux. L'échantillon d'images sélectionné est représentatif du type d'images téléchargées par les utilisateurs des modules d'essayage virtuel de lunettes. Les images contiennent des personnes ayant les cheveux très courts à longs, des cheveux blonds, des cheveux bruns, des arrières plans variés (un mur blanc, de la végétation, des stores,...).

La figure 3.20 illustre quelques uns de nos résultats de classification. Dans la première colonne, nous pouvons voir l'image traitée. Dans les deux colonnes suivantes, nous présentons les résultats de la classification utilisant un modèle décisionnel maximum de vraissemblance. Les descripteurs utilisés sont dans les deux cas des descripteurs colorimétriques et texturaux. La différence entre les deux résultats est due au choix du descripteur de texture. Dans la colonne de gauche, les résultats sont obtenus en utilisant les dictionnaires discriminants décrits précédemment. Dans la colonne de droite, nous présentons les résultats obtenus avec notre descripteur de texture fondé sur la construction de 3 dictionnaires adaptés aux 3 zones Peau, Cheveux et Fond. Nous pouvons constater visuellement que la qualité des résultats de classification est comparable. Pour notre base d'image, nous avons 78,5% des pixels qui sont correctements étiquetés en utilisant les dictionnaires discriminants et 77,5% en utilisant notre descripteur de texture. Ce qui est un peu décevant par rapport aux résultats de la figure 3.11

Les figures 3.21, 3.22 et 3.21 présentent quelques résultats intermédiaires et finaux de notre algorithme de classification. Sur les trois dernières colonnes, les zones surlignées

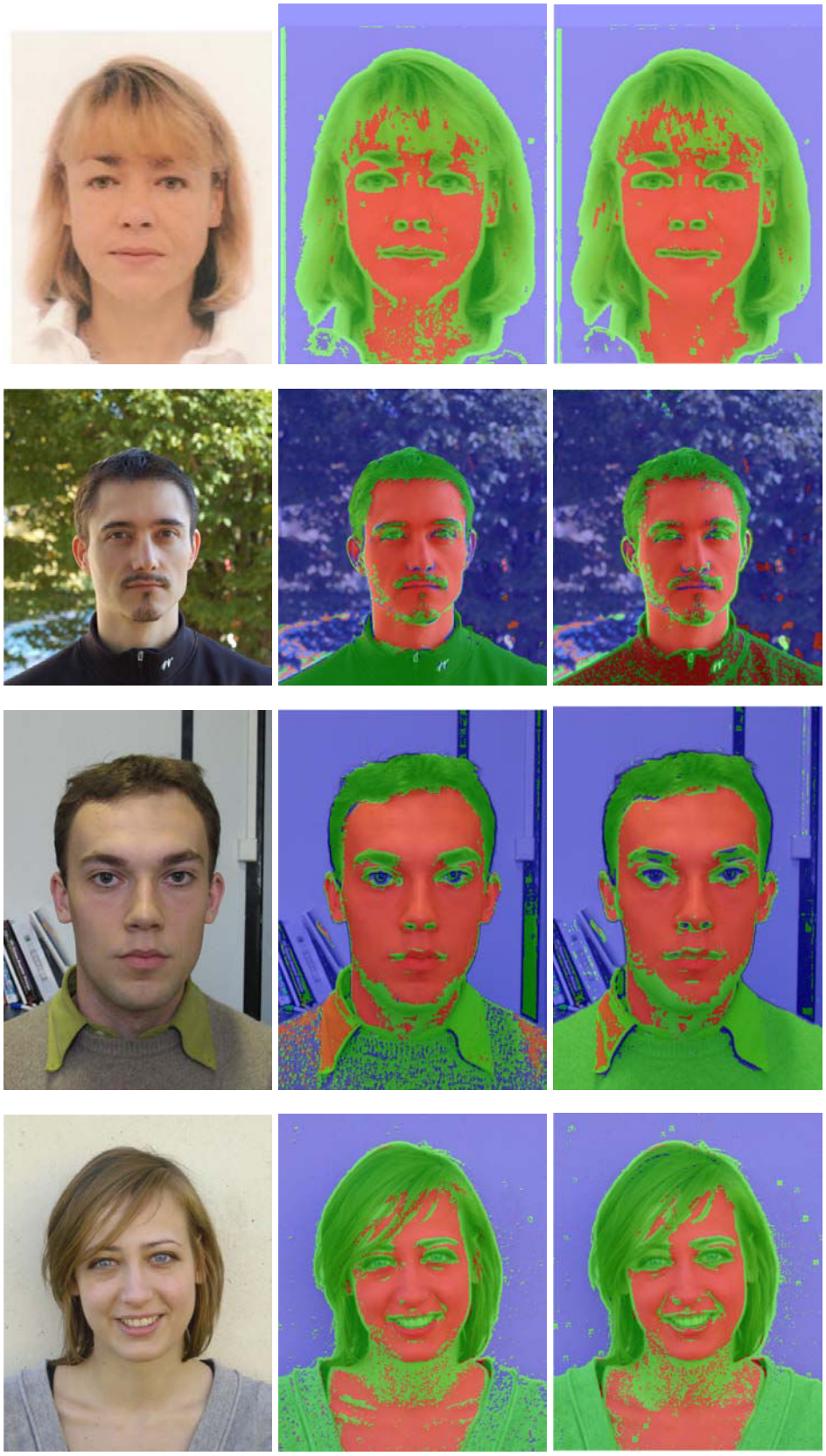


FIGURE 3.20 – Résultats de classification sans *a priori* en utilisant les descripteurs de couleur et de texture. Le descripteur de texture est obtenu en utilisant les dictionnaires discriminants ou les dictionnaires adaptés aux zones Peau, Cheveux et Fond.

de rouge correspondent à la classe Peau ; les zones surlignées de vert correspondent à la classe Cheveux ; et enfin les zones surlignées de bleu correspondent à la classe Fond. La première colonne illustre l'image à traiter. La deuxième colonne illustre les résultats obtenus par classification bayésienne à partir des descripteurs de couleur et de notre descripteur de texture construit par ACP. La troisième colonne illustre comme la précédente les résultats de classification bayésienne mais en y ajoutant un *a priori* global. Finalement la dernière colonne illustre nos résultats les plus aboutis. La méthode de classification employée combine l'utilisation du modèle décisionnel bayésien avec un *a priori* global et une régularisation locale de la carte de décision.

Sur l'ensemble des images, nous pouvons constater que la première phase de classification produit déjà des résultats intéressants notamment concernant les image riches en texture (images 6-7-8-9). Les erreurs de classification qui reviennent le plus souvent concerne la classification des pixels correspondants aux zones de vêtements. C'est pour réduire les aberrations de classification des pixels que nous avons introduit un *a priori* global sur l'image. Sur les images 1-2-3-4-8-9-13, les zones de vêtements qui étaient mal classées appartiennent maintenant à la classe Fond.

Je rappelle que la segmentation finale est utilisée pour un logiciel d'essayage virtuel de lunette. C'est à dire qu'il faut positionner l'image des lunettes entre le calque de la zone de peau et celui de la zone de Cheveux. Suite à la première étape de classification, nous constatons que certains éléments du visage, qui ont des caractérisques texturales et colorimétriques proches de celles des cheveux comme les cils, les sourcils et la barbe, appartiennent au calque des cheveux. Pour assurer un rendu réaliste à l'essayage virtuel de lunette, les éléments du visage cités précédemment doivent appartenir au calque de la peau. En effet, lorsque l'on porte des lunettes les sourcils sont visibles en transparence à travers les lunettes (comme la peau) et non par dessus les lunettes (comme les cheveux). La mise en équation de l'*a priori* global permet de forcer à classifier sous l'étiquette Peau les sourcils, cils et barbe (*cf.* images 1-2-3-6-7).

Enfin la dernière étape de la classification qui consiste à appliquer une régularisation sur les cartes de probabilités *a posteriori* obtenues à l'étape précédente, permet d'homogénéïser spatialement l'étiquetage des pixels. Sur l'ensemble des images, nous constatons que les zones sont plus "compactes".

Les ombres portées ont tendances à être mal classifiées (*cf.* images 10-11). Il peut être envisagé de travailler dans un espace de couleur mieux choisi. Nous avons choisi de travailler avec le composantes Cb et Cr de l'espace de couleur YCbCr pour justement s'affranchir des ombres. Cette modélisation de la couleur est très simpliste il faudrait introduire un modèle plus riche.

Finalement, la figure 3.24 montre que notre descripteur de texture pourtant simpliste donne des résultats de classification aussi satisfaisant que ceux obtenus avec le descripteur de texture utilisant les dictionnaires discrimimants. De plus, l'ajout des informations spatiales globales et locales permet d'améliorer fortement les résultats.

3.5 Conclusion

En conclusion, notre descripteur de texture fondé sur l'analyse en composantes principales permet de décrire simplement et de manière discriminante la texture des trois classes Peau, Cheveux et Fond. En ajoutant un *a priori* global de la scène traitée et un *a priori* local en introduisant un terme de régularisation spatiale, nous obtenons une segmentation

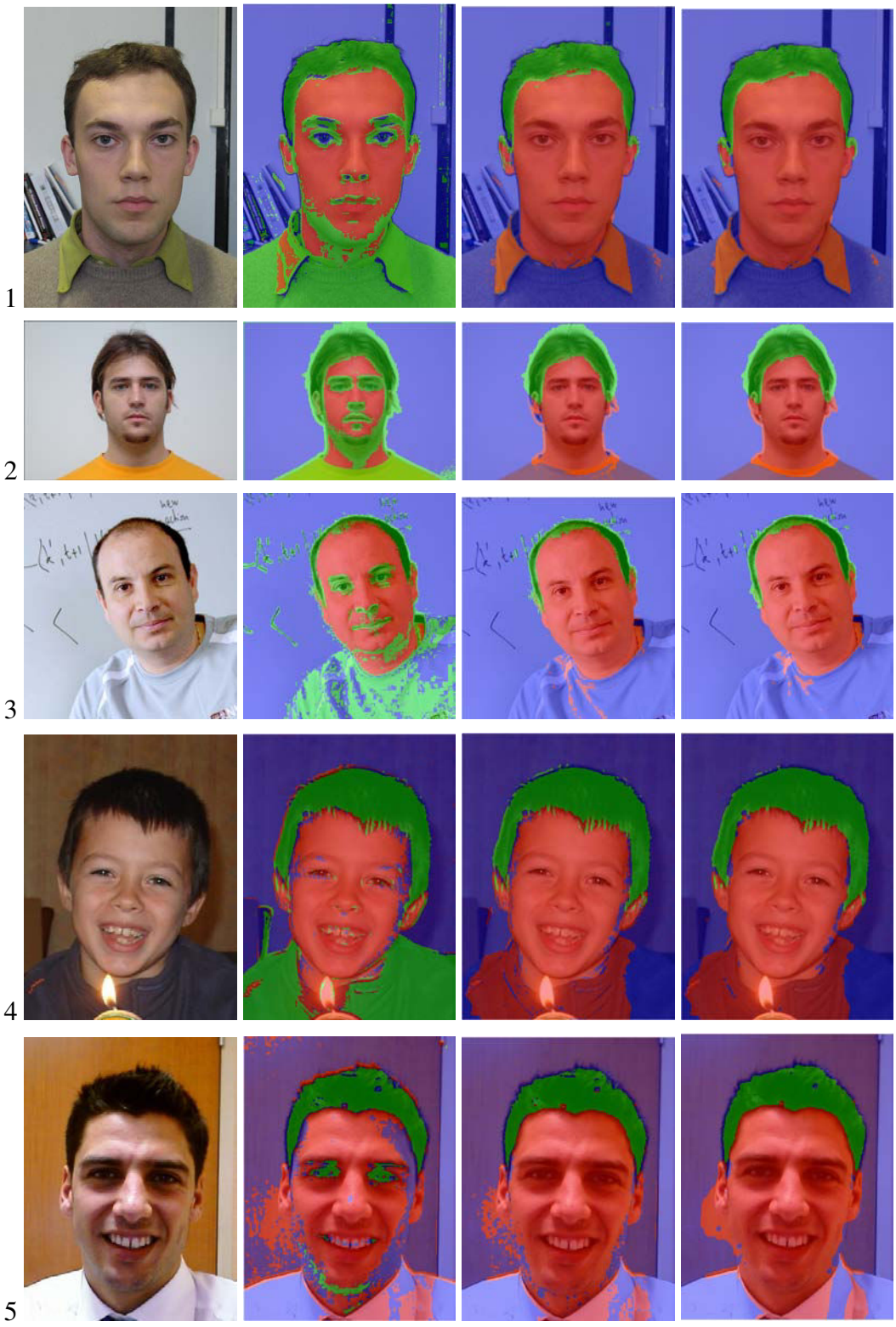


FIGURE 3.21 – Résultats de classification des pixels d'une image portrait

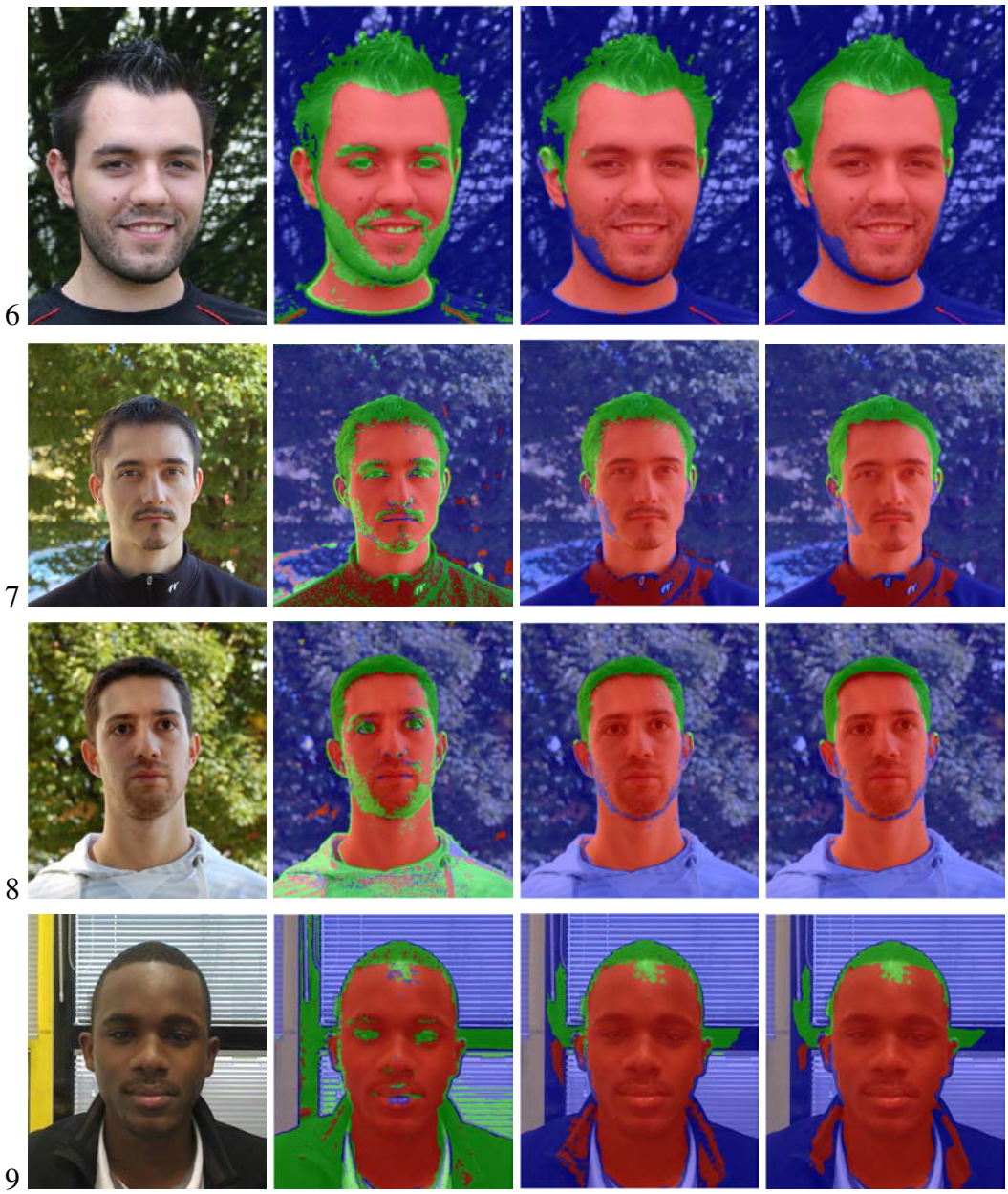


FIGURE 3.22 – Résultats de classification des pixels d'une image portrait

proche de la solution finale. Nous pouvons donc envisager d'utiliser ces résultats pour initialiser une segmentation précise utilisant les méthodes variationnelles.

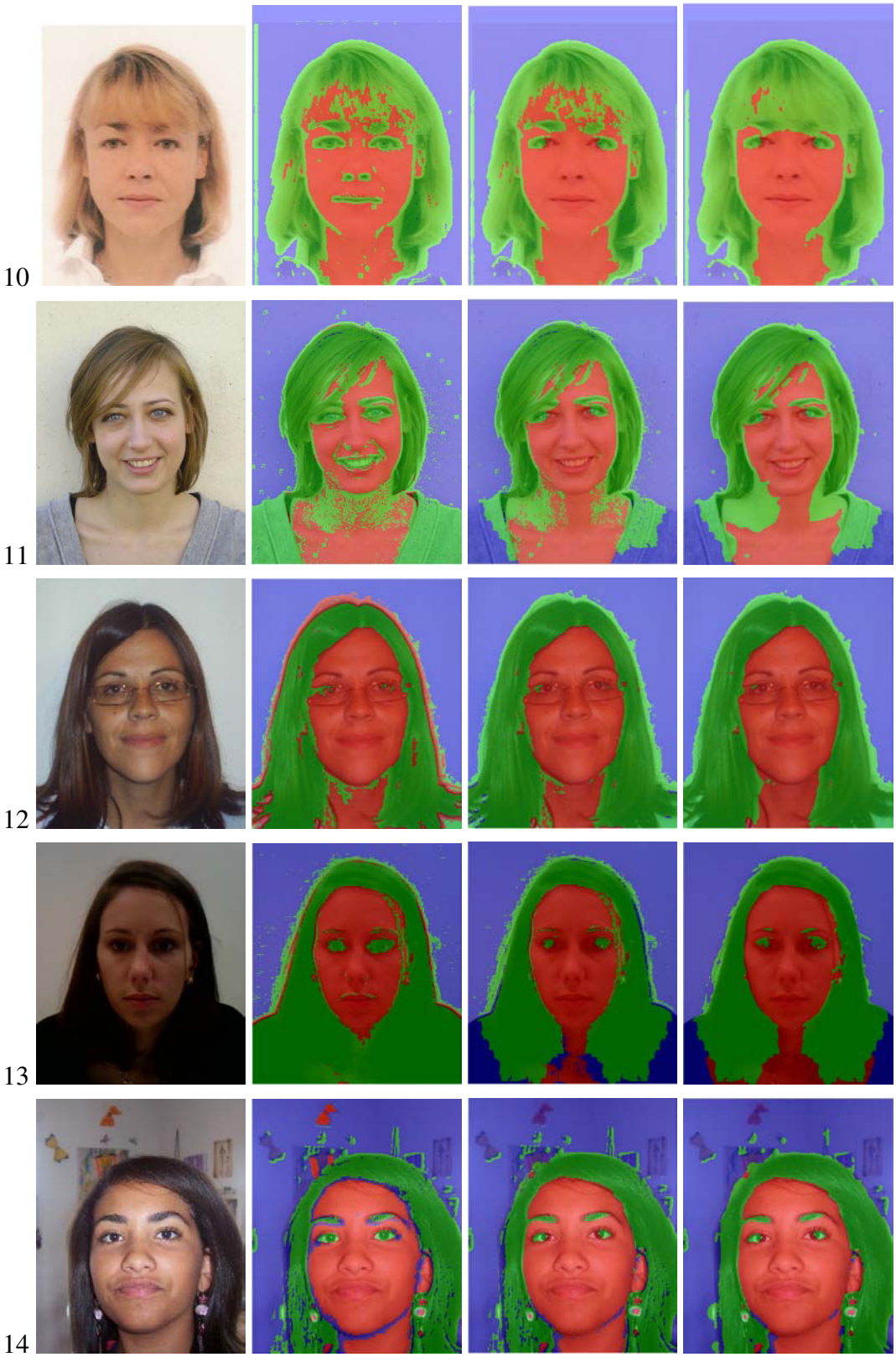


FIGURE 3.23 – Résultats de classification des pixels d'une image portrait

Classification sans *a priori* avec les dictionnaires discriminants	Classification sans *a priori* avec les dictionnaires adaptés aux zones	Classification avec *a priori*, régularisation et dictionnaires adaptés aux zones
78,5 %	77,5 %	89,4 %

FIGURE 3.24 – Pourcentage de pixels bien classifiés.

Chapitre 4

Segmentation en utilisant les méthodes variationnelles

Après avoir introduit une première technique de segmentation des images portrait par classification, nous étudions une seconde approche fondée sur les contours actifs. Les deux outils sont selon nous complémentaires. Les résultats issus de la classification restent grossiers mais permettent l'initialisation d'une méthode variationnelle plus fine. Cette finesse d'analyse est requise lorsque les images sont difficiles à traiter et présentent par exemple des zones de mélanges à la frontière des cheveux et de la peau. Nous décrivons une méthode variationnelle qui manipule donc cinq régions composées, comme précédemment, des trois régions contenant les pixels de la Peau, du Fond, des Cheveux, auxquelles nous ajoutons deux nouvelles régions contenant les pixels issus du mélange de la classe Peau avec la classe Cheveux ou bien de la classe Cheveux avec la classe Fond. Notre modèle s'appuie conjointement sur la texture et la couleur pour les termes d'attache aux données, et utilise également des termes de régularisation et des contraintes topologiques. Le résultat final de segmentation est obtenu lorsque le minimum global de la fonctionnelle proposée est atteint. Nous présentons les étapes de résolution qui nous permettent d'approcher ce minimum global tant théoriquement que numériquement. Enfin, nous montrons des résultats de segmentation sur des images de différentes natures.

Ce chapitre nous permet finalement de valider l'hypothèse que si l'on dispose d'un *a priori* sur la localisation des zones Peau, Fond et Cheveux, nous pouvons obtenir de bons résultats de segmentation pour les images portrait via une approche variationnelle.

4.1 Segmentation précise des images Portrait : 3 classes en 5 régions

4.1.1 La problématique

Dans ce chapitre, nous souhaitons segmenter précisément les images portrait en trois classes : Peau, Cheveux et Fond. Les propriétés de la chevelure sont telles qu'il n'est pas trivial de définir la frontière d'une région associée aux cheveux. Nous ne pouvons pas définir aisément la silhouette d'une chevelure comme nous pourrions par exemple le faire pour des objets rigides (stylo, gomme, tasse à café...). La difficulté est due à trois caractéristiques. La première est que la chevelure est un objet déformable ; en effet, sur une séquence d'image, un simple coup de vent peut modifier grandement sa silhouette. De plus, la chevelure est également un objet à géométrie variable ; nous pouvons rencontrer des coupes de cheveux courtes ou bien longues. Enfin, la principale difficulté pour définir la silhouette réside dans le fait que la chevelure est formée d'une multitude d'éléments de très petite taille, les cheveux. Le diamètre d'un cheveux avoisine les 50 μm. Pour avoir une segmentation précise de la chevelure, il faudrait que le pixel ait une précision inférieure à cette dimension. Hors, il est clair que cela est déraisonnable pour les images portrait. Nous voyons alors apparaître des régions de transition composées de pixels contenant un mélange entre la classe Cheveux et sa classe adjacente (Peau ou Fond). Nous introduisons deux régions correspondant aux mélanges des cheveux avec la peau et des cheveux avec le fond (notées respectivement Ω_{pc} et Ω_{cf}). L'objectif final, de ce chapitre, est de produire une segmentation précise entre les classes Peau, Cheveux et Fond. Pour cela, nous définissions non plus trois mais cinq régions dans l'image Portrait.

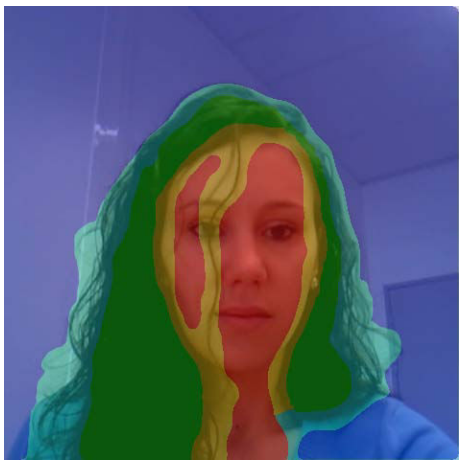


FIGURE 4.1 – Exemple de segmentation en 5 régions : Ω_p en rouge, Ω_{pc} en jaune, Ω_c en vert, Ω_{cf} en turquoise, Ω_f en bleu.

Pour segmenter précisément une image Portrait en 3 classes, nous cherchons 5 régions :

1. Ω_p la région qui regroupe tous les pixels appartenant uniquement à la classe Peau (en rouge sur la figure 4.1),
2. Ω_{pc} la région qui regroupe tous les pixels qui correspondent à un mélange des classes Peau et Cheveux (en jaune sur la figure 4.1),

3. Ω_c la région qui regroupe tous les pixels appartenant uniquement à la classe Cheveux (en vert sur la figure 4.1),
4. Ω_{cf} la région qui regroupe tous les pixels qui correspondent à un mélange des classes Cheveux et Fond (en turquoise sur la figure 4.1),
5. Ω_f la région qui regroupe tous les pixels appartenant uniquement à la classe Fond (en bleu sur la figure 4.1).

4.1.2 Segmentation multi-régions par combinaison de courbes de niveaux

Les cinq régions définies précédemment peuvent contenir une ou plusieurs composantes connexes, c'est en faisant cette remarque que nous nous orientons vers un outil capable d'expliquer différents types de topologie pour une même région. Une représentation implicite des frontières par courbes de niveaux est la solution que nous avons retenue dans ce travail. Dans cette représentation, les frontières changent aisément de topologie. Un critère énergétique de la segmentation, constitué de deux termes l'un d'attache aux données et l'autre de régularisation, permet respectivement de travailler avec l'information contenue dans l'image (homogénéité des régions par exemple) et les propriétés associées aux contours des régions dont on souhaite qu'ils soient lisses. La formulation variationnelle fait évoluer par une descente de gradient les contours alors dit "actifs" vers un minimum du critère énergique (appelé fonctionelle dans la suite).

4.1.2.1 L'initialisation de la segmentation

Un inconvénient de cette modélisation par optimisation d'une fonctionnelle et courbes (ou ensemble) de niveaux est que pour converger vers le minimum global de la fonctionnelle (correspondant à la segmentation optimale), nous devons, en l'absence de convexité, initialiser la segmentation en étant proche de la solution finale pour se trouver dans le bassin de convergence du minimum global. La Figure 4.2 illustre ces propos, l'initialisation 1 (à une distance d_1 du minimum global) converge vers un minimum local de la fonctionnelle ; tandis que l'initialisation 2 (à une distance $d_2 << d_1$ du minimum global) converge vers le minimum global de la fonctionnelle. Dans ce chapitre, les résultats de segmentation sont présentés en utilisant une initialisation manuelle grossière. L'objectif, dans un second temps, est d'utiliser les résultats du chapitre précédent comme une initialisation automatique.

4.1.2.2 Les courbes de niveaux multi-régions

Dans le chapitre 2, nous avons rappelé comment une courbe Γ, étant la frontière entre deux régions Ω_1, Ω_2 peut être représentée de manière implicite par une fonction lipschitzienne Φ_Γ de dimension supérieure. Φ_Γ est la fonction distance signée de chaque pixel à la courbe, avec Ω_1 correspondant à l'ensemble des pixels du domaine de l'image vérifiant $\Phi > 0$ et Ω_2 à ceux vérifiant $\Phi < 0$. La fonction Φ est aussi appelée *fonction courbe de niveau* (en anglais *levelset function*). Nous remarquons que l'utilisation d'une seule fonction Φ permet de segmenter uniquement deux régions (composées d'une ou plusieurs composantes connexes).

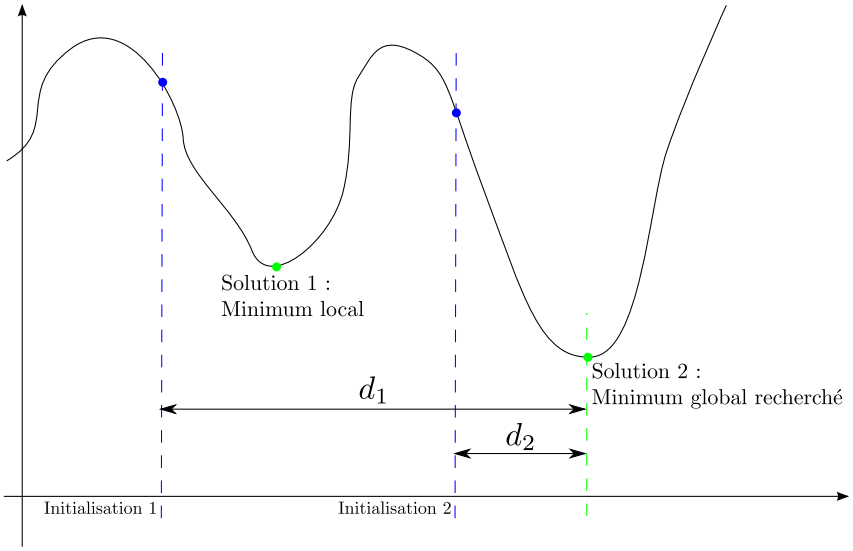


FIGURE 4.2 – Représentation de l'influence de l'initialisation pour une fonctionnelle non convexe.

Pour étendre la modélisation à plus de deux régions, Vese et Chan [77] proposent une segmentation multi-régions en utilisant plusieurs fonctions de distances signées. Avec N fonctions de distances signées, il est possible de segmenter 2^N régions. Pour notre application qui prend en compte 5 régions, nous devons utiliser 3 fonctions courbe de niveau, notées Φ_1, Φ_2, Φ_3 avec $\Phi_i : \Omega \longrightarrow \mathbb{R}$. L'union des niveaux zéro des Φ_i représentent les frontières des régions de l'image. Nous remarquons que l'utilisation de 3 fonctions

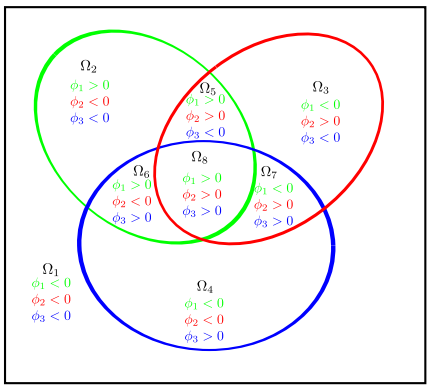


FIGURE 4.3 – Trois courbes de niveau zéro des fonctions Φ_1, Φ_2, Φ_3 qui représentent huit régions

distance signée, Φ_1, Φ_2 et Φ_3, fournit $2^3 = 8$ combinaisons possibles, nous pourrions donc segmenter 8 régions (*cf.* Figure 4.3). Souhaitant détecter uniquement 5 régions, il faut que les 3 régions restantes soient fortement pénalisées dans la fonctionnelle pour qu'elles n'apparaissent pas lors de la segmentation. Dans la section 4.3.1, à l'équation 4.2 nous définirons ce terme de pénalisation.

4.1.2.3 L'intérêt de segmenter 3 classes en 5 régions

L'utilisation des méthodes variationnelles pour la segmentation de l'image en 5 régions est une étape intermédiaire permettant d'obtenir une segmentation précise en appliquant sur les régions de transition une méthode de *matting*. Le *matting* permet de "démélanger" les couleurs à la frontière de l'objet cheveux. Suite à cette remarque, nous mettons en place un critère colorimétrique qui permet de définir les caractéristiques des régions mélange en fonction des propriétés des régions peau, cheveux et fond.

Dans la section 4.2, nous détaillons la mise en équation, la résolution et présentons quelques résultats de segmentation de l'image Portrait en 3 régions : Peau , Peau + Cheveux et le reste. Nous faisons ce choix car l'écriture de la segmentation en 5 régions est lourde. Mais le passage de 3 à 5 régions se fait aisément. Dans la section 4.3.1 nous écrirons la fonctionnelle que nous avons mis en place pour les 5 régions et nous présentons quelques uns de nos résultats à la section 4.3.2.

4.2 Segmentation précise entre la peau et les cheveux

Dans cette section, nous présentons, de façon détaillée, le cas particulier de la segmentation en 3 régions permettant de définir la frontière entre la peau et les cheveux.

4.2.1 Mise en équation

La segmentation précise entre la peau et les cheveux est le résultat d'une minimisation énergétique d'une fonctionnelle que nous pouvons scinder en deux parties :

- le terme d'attache aux données,
- le terme de régularisation.

Le terme d'attache aux données est une énergie sur le choix d'étiquetage de chaque pixel de l'image tandis que le terme de régularisation influe sur les propriétés de la courbe. Dans la littérature, nous rencontrons des termes s'attachant aux propriétés colorimétrique, fréquentielle ou encore d'attraction aux contours de l'image. Dans le cadre de notre étude et compte tenu de part la nature des données nous choisissons de construire cette énergie à partir des données colorimétriques. Juan et Keriven [36] se sont intéressés à la qualification des propriétés colorimétriques de la région mélange en fonction des propriétés des deux régions adjacentes.

Plus précisément, dans [36] Juan et Keriven proposent une technique de segmentation pour extraire de l'arrière plan un objet aux frontières complexes. Ils proposent de définir les coefficients de transparence entre l'objet à segmenter et l'arrière plan en utilisant une méthode de *matting*. Dans notre cas, l'objet est assimilable à la classe Peau, l'arrière plan a tout ce qui n'est pas de la Peau, c'est à dire les Cheveux et le Fond, et nous recherchons les valeurs des coefficients de transparence sur la zone de transition entre la peau et les cheveux.

La segmentation se décompose en 3 régions :

- la région constituée uniquement de pixels peau est appelée Ω_p,
- la région où l'objet à des propriétés de transparence, et où nous constatons un mélange colorimétrique entre la peau et les cheveux Ω_{pc},
- la région constituée de pixels uniquement cheveux et fond Ω_c.

Les auteurs modélisent les deux régions indépendantes Ω_p et Ω_c avec des modèles statistiques, notés p_p et p_c qui sont des fonctions densité de probabilité de type mélange de gaussiennes :

$$\begin{aligned} p_p(\omega) &= \sum_{i=1}^{N_p} \pi_{p_i} G_{\mu_{p_i}\,\Sigma_{p_i}}(\omega) \\ p_c(\omega) &= \sum_{j=1}^{N_c} \pi_{c_j} G_{\mu_{c_j}\,\Sigma_{c_j}}(\omega), \end{aligned}$$

où $G_{\mu\,\Sigma}()$ est une fonction gaussienne centrée en μ et de covariance Σ, nous rappelons que $G_{\mu\,\Sigma}(\omega)$ s'écrit $G_{\mu\,\Sigma}(\omega) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left(-\frac{(\omega-\mu)^t\Sigma^{-1}(\omega-\mu)}{2}\right)$ avec N la dimension des données ω (ici nous travaillons dans l'espace RVB donc $N = 3$). De plus, π_{p_i} et π_{c_j} sont respectivement les poids attribués aux i-ème et j-ème gaussiennes des classes Peau et Cheveux. En pratique, nous supposons que les distributions de Peau, de Cheveux et de Fond exprimées dans l'espace RVB peuvent être modélisées par un mélange de gaussiennes. Nous choisissons de manière empirique le nombre de gaussiennes N_p et N_c associées aux régions Ω_p et Ω_c. En pratique, nous prenons $N_p = N_c = 2$. Ce choix est le résultat d'un compromis entre le fait d'avoir un modèle paramétrique qui approche au mieux les données d'apprentissage en gardant un nombre minimal de paramètres.

La contribution de Juan et Keriven est de définir la fonction densité de probabilité de la région de transition Ω_{pc}, notée p_{pc}, en fonction de celle de la peau p_p et des cheveux p_c.

$$\begin{aligned} \mu_{pc_{ij}} &= \frac{\mu_{p_i} + \mu_{c_j}}{2} \\ \Sigma_{pc_{ij}} &= \frac{1}{3}\left(\Sigma_{p_i} + \Sigma_{c_j}\right) + \frac{1}{12}\left(\mu_{c_j} - \mu_{p_i}\right)\left(\mu_{c_j} - \mu_{p_i}\right)^t \end{aligned}$$

$$p_{pc}(\omega) = \sum_{i=1}^{N_p}\sum_{j=1}^{N_c} \pi_{pc_{ij}} G_{\mu_{pc_{ij}}\,\Sigma_{pc_{ij}}}(\omega)$$

Seuls les poids $\pi_{pc_{ij}}$ ne sont pas directement déduits des paramètres des fonctions densité de probabilité p_p et p_c. Ils sont estimés en appliquant sur la région Ω_{pc} un algorithme EM d'espérance-maximisation.

Le terme d'attache aux données est alors défini par la somme, sur les régions, des log-vraisemblances.

Pour le terme de régularisation, nous prenons en compte l'élasticité de la courbe en influant sur les propriétés de contraction de la courbe et de lissage des irrégularités géométriques. Cela veut dire que nous allons pondérer la longueur de la courbe pour obtenir un contour plus ou moins régulier. Nous notons Γ_1 la courbe permettant de définir la frontière entre les régions Ω_p et Ω_{pc}, et Γ_2 la courbe permettant de définir la frontière entre

Ω_{pc} et Ω_c. Notre fonctionnelle E s'écrit alors :

$$\begin{aligned} E(\Gamma_1, \Gamma_2) = & - \int_{\Omega_p} \log\left(p_p(\omega)\right) \quad d\omega \\ & - \int_{\Omega_{pc}} \log\left(p_{pc}(\omega)\right) \quad d\omega \\ & - \int_{\Omega_c} \log\left(p_c(\omega)\right) \quad d\omega \\ & + \nu\left(\mathcal{L}(\Gamma_1) + \mathcal{L}(\Gamma_2)\right). \end{aligned} \tag{4.1}$$

Dans l'équation 4.1, les trois premiers termes correspondent aux termes d'attache aux données sur les trois régions (Ω_p, Ω_{pc} et Ω_c). Le fait que nous ayons besoin de segmenter trois régions implique l'utilisation de deux courbes de niveau pour le passage à l'écriture implicite. Les deux fonctions de courbes de niveau, notées Φ_1 et Φ_2, peuvent représenter jusqu'à quatre régions. Il faudra alors pénaliser la quatrième région pour qu'elle n'apparaisse pas.

Les régions Ω_p, Ω_{pc} et Ω_c sont défini aux travers des fonctions Φ_1 et Φ_2 de la manière suivante :

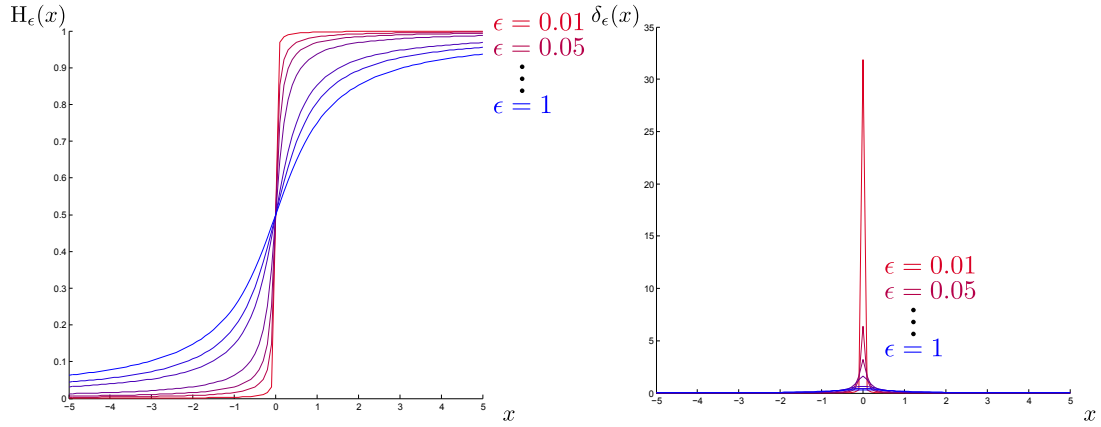
- $\Omega_p = \{\Phi_1 > 0 \cap \Phi_2 > 0\}$ avec sa fonction caractéristique donnée par $\chi_{\Omega_p} = H(\Phi_1)H(\Phi_2)$
- $\Omega_{pc} = \{\Phi_1 < 0 \cap \Phi_2 > 0\}$ avec sa fonction caractéristique donnée par $\chi_{\Omega_{pc}} = (1 - H(\Phi_1))H(\Phi_2)$
- $\Omega_c = \{\Phi_1 < 0 \cap \Phi_2 < 0\}$ avec sa fonction caractéristique donnée par $\chi_{\Omega_c} = (1 - H(\Phi_1))(1 - H(\Phi_2))$

Toujours dans l'équation 4.1, la fonction $\mathcal{L}()$ permet de calculer la longueur de la courbe et ν est un paramètre constant, positif qui permet de régler la régularité de la courbe. Le passage à l'écriture implicite des courbes nous permet d'écrire la fonction suivante :

$$\begin{aligned} E(\Phi_1, \Phi_2) = & - \int_{\Omega} \mathrm{H}(\Phi_1(\omega))\mathrm{H}(\Phi_2(\omega)) \log\left(p_p(\omega)\right) \quad d\omega \\ & - \int_{\Omega} (1 - \mathrm{H}(\Phi_1(\omega)))\mathrm{H}(\Phi_2(\omega)) \log\left(p_{pc}(\omega)\right) \quad d\omega \\ & - \int_{\Omega} (1 - \mathrm{H}(\Phi_1(\omega)))(1 - \mathrm{H}(\Phi_2(\omega))) \log\left(p_c(\omega)\right) \quad d\omega \\ & + \int_{\Omega} \mathrm{H}(\Phi_1(\omega))(1 - \mathrm{H}(\Phi_2(\omega)))\, C \quad d\omega \\ & + \nu \int_{\Omega} \left(|\nabla \mathrm{H}(\Phi_1(\omega))| + |\nabla \mathrm{H}(\Phi_2(\omega))|\right) d\omega \end{aligned} \tag{4.2}$$

où C est la constante permettant de contraindre la segmentation à trois régions en pénalisant fortement la fonctionnelle lors de l'apparition d'une quatrième région, qui n'est pas voulue dans cette mise en équation. $\mathrm{H}(x)$ est la fonction Heaviside (ou échelon), c'est-à-dire lorsque $x \geq 0$ alors $\mathrm{H}(x) = 1$ et lorsque $x < 0$ alors $\mathrm{H}(x) = 0$. $\delta(x)$ est la distribution de Dirac, la dérivée distributionnelle de la fonction Heaviside. Elle peut être considérée comme une fonction qui prend une « valeur » infinie en 0, et la valeur zéro partout ailleurs, et dont l'intégrale sur $\mathbb{R}$ est égale à 1. Le calcul de la longueur de la courbe $\mathcal{L}()$ en utilisant les fonctions implicites mérite quelques explications disponibles en annexe A.

Pour pouvoir résoudre cette équation par descente de gradient, nous approchons la fonction $H(x)$ par une fonction au moins deux fois dérivable, notée $H_\epsilon(x)$. Chan et Vese



(a) Fonction échelon régularisée H_ϵ (b) Fonction Dirac régularisée δ_ϵ

FIGURE 4.4 – Les fonctions échelon et Dirac régularisée pour $\epsilon = [0.01\ 0.05\ 0.1\ 0.2\ 0.5\ 0.7\ 1]$.

[8] proposent une fonction $C^\infty(\Omega)$ c'est-à-dire qui appartient à l'ensemble des fonctions indéfiniment dérivables sur Ω. Nous écrivons :

$$H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{x}{\epsilon}\right)\right),$$

et nous en déduisons :

$$\delta_\epsilon(x) = \frac{dH_\epsilon}{dx} = \frac{\epsilon}{\pi(\epsilon^2 + x^2)}$$

C'est le paramètre ϵ de H_ϵ qui permet de régler la vitesse du saut de la courbe. Lorsque ϵ tend vers 0, H_ϵ tend à devenir la fonction échelon et δ_ϵ vers la distribution de Dirac (*cf.* Figure 4.4).

4.2.2 Résolution

Pour minimiser la fonctionnelle 4.2, nous sommes confrontés à la résolution d'un système d'équations aux dérivées partielles. Avant cela, nous parlons des différentes conditions aux limites que nous pouvons mettre en place.

4.2.2.1 Conditions aux limites

Nous rappelons que $\Omega \in \mathbb{R}^2$ est le domaine fini correspondant au support de l'image. A ce dernier, nous associons son bord extérieur que nous notons $\partial\Omega$. Le long de ce bord, on trouve un champ de vecteurs $\mathbf{n}$ normalisés, orientés vers l'extérieur de Ω, orthogonaux au bord (*cf.* Figure 4.5). Il joue un rôle important dans la dérivation des équations d'Euler-Lagrange et pour la spécification de certaines conditions de bords (Neumann ici).

Les conditions aux limites sont définies en imposant le comportement de la fonction Φ et / ou de ses dérivées en tout point du bord $\partial\Omega$. La Figure 4.6 illustre les différentes configurations des conditions aux limites de la fonction Φ représentées en niveau de gris. La courbe bleue représente la segmentation finale de l'objet, c'est le niveau zéro de la

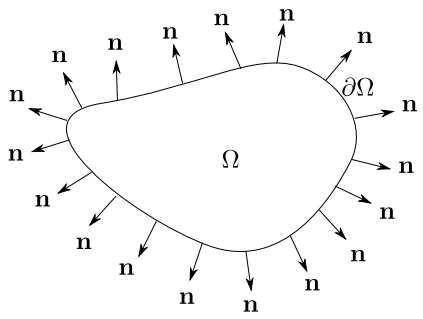
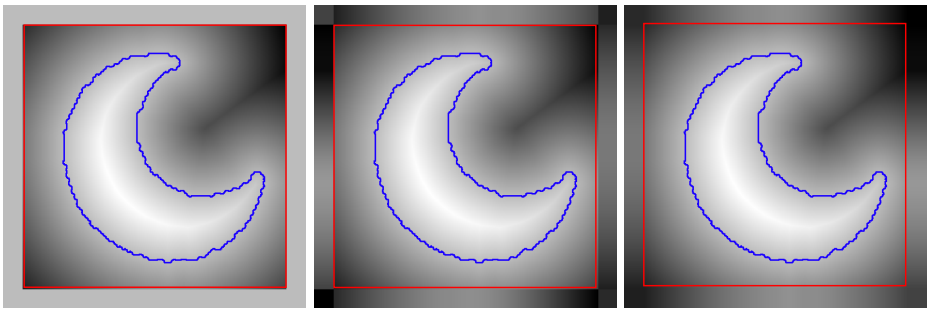


FIGURE 4.5 – Exemple de domaine Ω avec son bord associé $\partial\Omega$ et le champ de vecteurs $\mathbf{n}$

fonction Φ. A l'intérieur du carré rouge, nous sommes sur le domaine image Ω et à l'extérieur, nous représentons les différentes conditions aux limites sur le bord de l'image $\partial\Omega$ décrites ci-dessous.

(a) Image à segmenter



(b) Dirichlet (c) Périodique (d) Neumann

FIGURE 4.6 – Illustration des différentes conditions aux limites de la fonction Φ.

Conditions de Dirichlet :

Les conditions aux limites de Dirichlet sont définies en imposant la valeur de la fonction $\Phi(\omega)$ en tout point ω du bord $\partial\Omega$. En pratique, il est courant de prendre $\Phi(\omega) = 0$ pour $\omega \in \partial\Omega$. Dans notre cas, ce type de condition aux limites est problématique car elle correspondant de fait à ajouter un nouveau bord à cette région également à la frontière de la région que nous cherchons à obtenir.

Conditions périodiques :

Les conditions aux limites périodiques s'affranchissent des frontières de l'image en accolant artificiellement l'information du bas de l'image avec celle du haut et l'information

de la droite de l'image avec celle de la gauche. Dans le cas de la segmentation des images Portrait, le signal n'est pas périodique. Nous ne pouvons pas envisager de coller la première ligne de l'image à la dernière sans qu'il y ait la création d'un contour artificiel. En général, la première ligne de l'image appartient est de l'arrière plan tandis que la dernière ligne est plutôt composée de vêtement. *A priori* ces deux lignes sont dissemblables. Ce type de conditions ne peut donc pas être envisagé pour les images portrait.

Conditions de Neumann :

Les conditions aux limites de Neumann sont celles que nous utilisons par la suite. Elles sont définies en imposant la valeur de la dérivée de la fonction Φ suivant la normale de Ω et en tout point du bord $\partial\Omega$. Il est courant de prendre $\frac{\partial\Phi}{\partial\mathbf{n}}(\omega) = 0$ pour $\omega \in \partial\Omega$. En pratique, cela correspond à recopier sur les bords de l'image les pixels à la frontière.

4.2.2.2 Descente de gradient

Les fonctions de courbe de niveau Φ_1 et Φ_2 qui minimisent l'énergie E définie par l'équation 4.2 doivent remplir une condition d'optimalité. Cette condition nécessaire est que la différentielle de E, $D_{(\Phi_1,\Phi_2)}E$ s'annule en (Φ_1, Φ_2). La différentielle est une forme linéaire qui peut s'écrire vectoriellement en utilisant la notion de gradient. Cela donne l'équation d'Euler-lagrange :

$$\nabla_{(\Phi_1,\Phi_2)}E = 0$$

ou, en écrivant terme à termes, le système d'équations d'Euler-Lagrange de E :

$$\begin{cases} \nabla_{\Phi_1}E = 0 \\ \nabla_{\Phi_2}E = 0 \end{cases}$$

La résolution de ce système n'est pas directe. En effet, on est en présence d'un système d'équation non-linéaire. On contourne la difficulté de résolution directement en proposant une méthode itérative. Nous cherchons une fonction $t \mapsto \Phi(-,t) = (\Phi_1(-,t), \Phi_2(-,t))$, avec $\Phi(-,0) = (\Phi_1(-), \Phi_2(-))$ et satisfaisant l'équation de descente de gradient

$$\frac{d\Phi}{dt} = -\nabla_\Phi(t)E.;$$

où t désigne un temps algorithmique.

L'idée est que, partant d'un point $(\Phi(0); E(\Phi(0)))$ sur le graphe G_E de E, on va descendre le long de la courbe de plus forte pente partant de ce point sur G_E et on espère arriver à un point stationnaire caractérisée par $\frac{d\Phi}{dt} = \vec{0}$ et donc $\nabla_\Phi E = \vec{0}$, qui sera un minimum, au moins local.

Le gradient est défini comme l'unique vecteur représentant la différentielle pour une métrique choisie, plus de détails sont disponible dans l'annexe B. Dans notre cas, le gradient ∇E_Φ est calculé dans l'espace de Hilbert $L^2(\Omega)$, où $L^2(\Omega)$ est l'espace des fonctions de carré sommable sur Ω, muni du produit scalaire usuel. On pose $l_1(t) = E(\Phi_1 + t\mathbf{v})$ avec $\mathbf{v}$ le vecteur indiquant la direction dans laquelle on calcule la dérivée de E en Φ_1. Pour

alléger les notations nous n'écrirons plus les dépendances en ω.

$$\begin{aligned} l_1(t) = & - \int_\Omega \mathrm{H}_\epsilon(\Phi_1 + t\mathbf{v})\mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right) \quad d\omega \\ & - \int_\Omega (1-\mathrm{H}_\epsilon(\Phi_1 + t\mathbf{v}))\mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right) \quad d\omega \\ & - \int_\Omega (1-\mathrm{H}_\epsilon(\Phi_1 + t\mathbf{v}))(1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right) \quad d\omega \\ & + \int_\Omega \mathrm{H}_\epsilon(\Phi_1 + t\mathbf{v})(1-\mathrm{H}_\epsilon(\Phi_2))\, C \quad d\omega \\ & + \nu \int_\Omega (\delta_\epsilon(\Phi_1 + t\mathbf{v})|\nabla(\Phi_1 + t\mathbf{v})| + \delta_\epsilon(\Phi_2)|\nabla\Phi_2|)\, d\omega \end{aligned}$$

Pour les quatre premiers termes, qui sont les termes d'attaches aux données le calcul du gradient est trivial. Le calcul du dernier terme, le terme de régularisation est un peu plus délicat. Nous utilisons la formule de la dérivée de fonctions composées et la formule de Green :

$$\int_\Omega \nabla u \cdot \nabla v \; d\omega = -\int_\Omega u \; \mathrm{div}(v) \; d\omega + \int_{\partial\Omega} \frac{\partial v}{\partial \mathbf{n}} u \; ds.$$

Les conditions de Neumann (annoncées précédemment), nous permettent d'éliminer les termes de bord. Nous pouvons récrire la formule de Green :

$$\int_\Omega \nabla u \cdot \nabla v \; d\omega = -\int_\Omega u \; \mathrm{div}(v) \; d\omega,$$

et nous obtenons :

$$\begin{aligned} l_1'(t) = & - \int_\Omega \delta_\epsilon(\Phi_1 + t\mathbf{v})\; \mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right)\mathbf{v} \; d\omega \\ & + \int_\Omega \delta_\epsilon(\Phi_1 + t\mathbf{v})\; \mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right)\mathbf{v} \; d\omega \\ & + \int_\Omega \delta_\epsilon(\Phi_1 + t\mathbf{v})\; (1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right)\mathbf{v} \; d\omega \\ & + \int_\Omega \delta_\epsilon(\Phi_1 + t\mathbf{v})\; (1-\mathrm{H}_\epsilon(\Phi_2))\; C\; \mathbf{v} \; d\omega \\ & - \nu \int_\Omega \delta_\epsilon(\Phi_1 + t\mathbf{v})\; \mathrm{div}\left(\frac{\nabla(\Phi_1 + t\mathbf{v})}{|\nabla(\Phi_1 + t\mathbf{v})|}\right)\mathbf{v} \; d\omega \end{aligned}$$

$$\begin{aligned} l_1'(0) = & \int_\Omega \delta_\epsilon(\Phi_1)\left[-\mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right) + \mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right)\right. \\ & \left. + (1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right) + (1-\mathrm{H}_\epsilon(\Phi_2))\; C - \nu\; \mathrm{div}\left(\frac{\nabla(\Phi_1)}{|\nabla(\Phi_1)|}\right)\right]\mathbf{v} \; d\omega \\ = & \left\langle \delta_\epsilon(\Phi_1)\left[-\mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right) + \mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right)\right.\right. \\ & \left.\left. + (1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right) + (1-\mathrm{H}_\epsilon(\Phi_2))\; C - \nu\; \mathrm{div}\left(\frac{\nabla(\Phi_1)}{|\nabla(\Phi_1)|}\right)\right], \mathbf{v}\right\rangle \end{aligned}$$

De cette dernière équation, nous pouvons en déduire l'écriture du gradient de la fonctionnelle en Φ_1 : ∇E_{Φ_1}. En appliquant le même procédé, nous pouvons calculer le gradient

∇E_{Φ_2} de la fonctionnelle en Φ_2.

$$\begin{aligned}\nabla E_{\Phi_1} &= \delta_\epsilon(\Phi_1)\Big[-\mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right) + \mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right)\\ &+ (1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right) + (1-\mathrm{H}_\epsilon(\Phi_2))\,C - \nu\,\mathrm{div}\left(\frac{\nabla\Phi_1}{|\nabla\Phi_1|}\right)\Big]\end{aligned}$$

$$\begin{aligned}\nabla E_{\Phi_2} &= \delta_\epsilon(\Phi_2)\Big[-\mathrm{H}_\epsilon(\Phi_1)\log\left(p_p\right) - \mathrm{H}_\epsilon(\Phi_1)\log\left(p_{pc}\right)\\ &+ (1-\mathrm{H}_\epsilon(\Phi_1))\log\left(p_c\right) - \mathrm{H}_\epsilon(\Phi_1)\,C - \nu\,\mathrm{div}\left(\frac{\nabla\Phi_2}{|\nabla\Phi_2|}\right)\Big]\end{aligned}$$

L'évolution de Φ_1 et Φ_2 en fonction du temps est obtenue par descente de gradient. Nous écrivons :

$$\begin{aligned}\frac{d\Phi_1}{dt} &= \delta_\epsilon(\Phi_1)\Big[\mathrm{H}_\epsilon(\Phi_2)\log\left(p_p\right) - \mathrm{H}_\epsilon(\Phi_2)\log\left(p_{pc}\right)\\ &- (1-\mathrm{H}_\epsilon(\Phi_2))\log\left(p_c\right) - (1-\mathrm{H}_\epsilon(\Phi_2))\,C + \nu\,\mathrm{div}\left(\frac{\nabla\Phi_1}{|\nabla\Phi_1|}\right)\Big]\end{aligned} \tag{4.3}$$

$$\begin{aligned}\frac{d\Phi_2}{dt} &= \delta_\epsilon(\Phi_2)\Big[\mathrm{H}_\epsilon(\Phi_1)\log\left(p_p\right) + \mathrm{H}_\epsilon(\Phi_1)\log\left(p_{pc}\right)\\ &- (1-\mathrm{H}_\epsilon(\Phi_1))\log\left(p_c\right) + \mathrm{H}_\epsilon(\Phi_1)\,C + \nu\,\mathrm{div}\left(\frac{\nabla\Phi_2}{|\nabla\Phi_2|}\right)\Big]\end{aligned} \tag{4.4}$$

4.2.2.3 Discrétisation des équations linéaires

Dans l'équation 4.2, nous définissons l'énergie de notre segmentation sur le domaine image Ω qui est en pratique un domaine discrétisé. Pour résoudre les équations 4.3 et 4.4 nous devons les discrétiser sur le domaine de l'image. Nous supposons que Ω est une grille régulière représentée sur la Figure 4.7 avec δh la valeur de la distance horizontale ou verticale entre les points voisins de la grille.

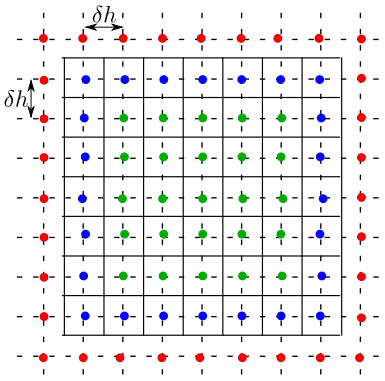


FIGURE 4.7 – Grille de discrétisation de l'image. Les points verts et bleus appartiennent à Ω. Les points bleus sont voisins aux bords. Les points rouges appartiennent aux bords de la grille $\partial\Omega$.

Pour représenter les opérateurs de dérivation $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$ nous utilisons les différences finies suivantes :

Différence avant en x	$D_x^+\Phi(i,j) = \frac{\Phi(i+1,j)-\Phi(i,j)}{\delta h}$
Différence arrière en x	$D_x^-\Phi(i,j) = \frac{\Phi(i,j)-\Phi(i-1,j)}{\delta h}$
Différence centrée en x	$D_x^c\Phi(i,j) = \frac{\Phi(i+1,j)-\Phi(i-1,j)}{2\delta h}$
Différence avant en y	$D_y^+\Phi(i,j) = \frac{\Phi(i,j+1)-\Phi(i,j)}{\delta h}$
Différence arrière en y	$D_y^-\Phi(i,j) = \frac{\Phi(i,j)-\Phi(i,j-1)}{\delta h}$
Différence centrée en y	$D_y^c\Phi(i,j) = \frac{\Phi(i,j+1)-\Phi(i,j-1)}{2\delta h}$

La discrétisation des équations 4.3 et 4.4 révèlent deux unités de discrétisation : spatiale et temporelle. La discrétisation spatiale intervient lors de la définition des différences avant, arrière ou centrée. La discrétisation temporelle introduit l'évolution pas à pas des fonctions Φ. Dans la suite, pour des questions de lisibilité, cette discrétisation temporelle sera renotée avec un indice n en puissance : Φ^n (au lieu de $\Phi(-,n)$ définie précédemment dans la section 4.2.2.2). Cette discrétisation temporelle peut suivre un schéma explicite ou bien semi-implicite. Dans le cas d'une discrétisation avec un schéma explicite, les données à l'instant $n+1$ sont calculées uniquement à partir des données de l'instant n. Dans le cas d'une discrétisation temporelle avec un schéma semi-implicite, les données à l'instant $n+1$ sont calculées à partir des données de l'instant n et des données disponibles de l'instant $n+1$. Nous utilisons le schéma semi-implicite pour résoudre le système car, il est connu qu'il permet d'utiliser un pas de temps plus grand qu'un schéma explicite [83]. Si Φ^n est la valeur de Φ au temps $n\delta t$, où δt est le pas de temps, alors on peut écrire :

$$\frac{\Phi^{n+1}-\Phi^n}{\delta t} = L(\Phi^n, \Phi^{n+1})$$

où $L(\Phi^n, \Phi^{n+1})$ est non-linéaire en Φ^n et linéaire en Φ^{n+1}. Le principe de cette résolution est de fixer, dans la discrétisation du gradient, les termes non-linéaires dépendant de Φ.

L'étape de relaxation par discrétisation avec un schéma semi-implicite aboutit à un système linéaire. En chaque pixel nous pouvons écrire une équation.

Pour aider à la discrétisation, nous ré-écrivons le terme de régularisation, défini dans la section précédente, de la manière suivante (avec $k \in \{1,2\}$) :

$$\operatorname{div}\left(\frac{\nabla\Phi_k^{n+1}}{|\nabla\Phi_k^n|}\right) = \frac{\partial}{\partial x}\left(\frac{1}{\sqrt{\frac{\partial\Phi_k^n}{\partial x}^2 + \frac{\partial\Phi_k^n}{\partial y}^2}}\frac{\partial\Phi_k^{n+1}}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{1}{\sqrt{\frac{\partial\Phi_k^n}{\partial x}^2 + \frac{\partial\Phi_k^n}{\partial y}^2}}\frac{\partial\Phi_k^{n+1}}{\partial y}\right) \tag{4.5}$$

La discrétisation du terme de régularisation s'écrit alors :

$$\operatorname{div}\left(\frac{\nabla\Phi_k^{n+1}}{|\nabla\Phi_k^n|}\right)(i,j) = D_x^-\left(\tfrac{1}{|\nabla\Phi_k^n|}D_x^+\Phi_k^{n+1}(i,j)\right) + D_y^-\left(\tfrac{1}{|\nabla\Phi_k^n|}D_y^+\Phi_k^{n+1}(i,j)\right) \tag{4.6}$$

$$\begin{aligned}\operatorname{div}\left(\frac{\nabla\Phi_k^{n+1}}{|\nabla\Phi_k^n|}\right)(i,j) = {} & \tfrac{1}{(\delta h)^2}\left(\frac{\Phi_k^{n+1}(i+1,j)-\Phi_k^{n+1}(i,j)}{A_k} - \frac{\Phi_k^{n+1}(i,j)-\Phi_k^{n+1}(i-1,j)}{B_k}\right) \\ & + \tfrac{1}{(\delta h)^2}\left(\frac{\Phi_k^{n+1}(i,j+1)-\Phi_k^{n+1}(i,j)}{C_k} - \frac{\Phi_k^{n+1}(i,j)-\Phi_k^{n+1}(i,j-1)}{D_k}\right)\end{aligned} \tag{4.7}$$

Où les C_i sont données par les équations suivantes :

$$A_k = \sqrt{\frac{(\Phi_k^n(i+1,j) - \Phi_k^n(i,j))^2}{(\delta h)^2} + \frac{(\Phi_k^n(i,j+1) - \Phi_k^n(i,j-1))^2}{(2\delta h)^2}},$$

$$B_k = \sqrt{\frac{(\Phi_k^n(i,j) - \Phi_k^n(i-1,j))^2}{(\delta h)^2} + \frac{(\Phi_k^n(i-1,j+1) - \Phi_k^n(i-1,j-1))^2}{(2\delta h)^2}},$$

$$C_k = \sqrt{\frac{(\Phi_k^n(i+1,j) - \Phi_k^n(i-1,j))^2}{(2\delta h)^2} + \frac{(\Phi_k^n(i,j+1) - \Phi_k^n(i,j))^2}{(\delta h)^2}},$$

$$D_k = \sqrt{\frac{(\Phi_k^n(i+1,j-1) - \Phi_k^n(i-1,j-1))^2}{(2\delta h)^2} + \frac{(\Phi_k^n(i,j) - \Phi_k^n(i,j-1))^2}{(\delta h)^2}}.$$

En pratique, nous prenons $\delta h = 1$ pixel. La discrétisation des équations 4.3 et 4.4 s'écrivent :

$$\begin{aligned}
\frac{\Phi_1^{n+1}(i,j) - \Phi_1^n(i,j)}{\delta t} &= \delta_\epsilon(\Phi_1^n(i,j)) \Big[\mathrm{H}_\epsilon(\Phi_2^n(i,j)) \ \log\Big(p_p(I(i,j))\Big) \\
&- \mathrm{H}_\epsilon(\Phi_2^n(i,j)) \ \log\Big(p_{pc}(I(i,j))\Big) \\
&- (1 - \mathrm{H}_\epsilon(\Phi_2^n(i,j))) \ \log\left(p_c(I(i,j))\right) \\
&- (1 - \mathrm{H}_\epsilon(\Phi_2^n(i,j))) \ C \\
&+ \nu \left(\frac{\Phi_1^{n+1}(i+1,j) - \Phi_1^{n+1}(i,j)}{A_1} - \frac{\Phi_1^{n+1}(i,j) - \Phi_1^{n+1}(i-1,j)}{B_1} \right) \\
&+ \nu \left(\frac{\Phi_1^{n+1}(i,j+1) - \Phi_1^{n+1}(i,j)}{C_1} - \frac{\Phi_1^{n+1}(i,j) - \Phi_1^{n+1}(i,j-1)}{D_1} \right) \Big]
\end{aligned} \tag{4.8}$$

$$\begin{aligned}
\frac{\Phi_2^{n+1}(i,j) - \Phi_2^n(i,j)}{dt} &= \delta_\epsilon(\Phi_2^n(i,j)) \Big[\mathrm{H}_\epsilon(\Phi_1^n(i,j)) \ \log\Big(p_p(I(i,j))\Big) \\
&+ (1 - \mathrm{H}_\epsilon(\Phi_1^n(i,j))) \ \log\Big(p_{pc}(I(i,j))\Big) \\
&- (1 - \mathrm{H}_\epsilon(\Phi_1^n(i,j))) \ \log\left(p_c(I(i,j))\right) \\
&+ \mathrm{H}_\epsilon(\Phi_1^n(i,j)) \ C \\
&+ \nu \left(\frac{\Phi_2^{n+1}(i+1,j) - \Phi_2^{n+1}(i,j)}{A_2} - \frac{\Phi_2^{n+1}(i,j) - \Phi_2^{n+1}(i-1,j)}{B_2} \right) \\
&+ \nu \left(\frac{\Phi_2^{n+1}(i,j+1) - \Phi_2^{n+1}(i,j)}{C_2} - \frac{\Phi_2^{n+1}(i,j) - \Phi_2^{n+1}(i,j-1)}{D_2} \right) \Big]
\end{aligned} \tag{4.9}$$

La fonction δ_ϵ étant nulle presque partout, sauf autour du changement de signe des courbes de niveau. Nous ne prendrons pas en compte la valeur des fonctions courbes de niveau qui sont trop loin de la courbe de niveau zéro. Les conditions aux limites que nous avons énoncées précédemment auront alors peu ou pas d'influence sur le résultat.

4.2.2.4 Résolution par relaxation

Nous pouvons remarquer que la mise à jour des courbes de niveaux au point (i, j) dépend uniquement des informations contenues sur ses 4 plus proches voisins $\{(i, j -$

$1), (i, j+1), (i+1, j), (i-1, j)\}$. Pour alléger les notations, on note c le point central jusqu'ici noté (i, j), et $\mathcal{N}(c) = \{n, s, e, o\}$ les 4 plus proches voisins (*cf.* figure 4.8).

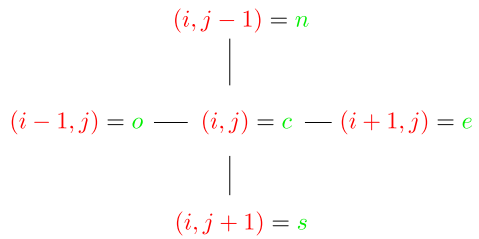


FIGURE 4.8 – Représentation d'un point central et de son voisinage.

En posant $(k, k') = (1, 2)$ pour l'équation discrète 4.8 et $(k, k') = (2, 1)$ pour l'équation discrète 4.9, c'est deux mêmes équations se réécrivent alors :

$$\frac{\Phi_{k,c}^{n+1} - \Phi_{k,c}^{n}}{dt} = \delta_\epsilon(\Phi_{k,c}) f_k(\Phi_{k',c}, C) + \nu\delta_\epsilon(\Phi_{k,c}) \sum_{r\in\mathcal{N}(c)} \beta_{k,r}(\Phi_{k,r}^{n+1} - \Phi_{k,c}^{n+1}) \tag{4.10}$$

avec,

$$f_1(\Phi_{k',c}, C) = \mathrm{H}_\epsilon(\Phi_{2,c}^n)\left(\log\left(p_p(I_c)\right) - \log\left(p_{pc}(I_c)\right)\right) - (1 - \mathrm{H}_\epsilon(\Phi_{2,c}^n))\left(\log\left(p_c(I_c)\right) + C\right),$$

$$f_2(\Phi_{k',c}, C) = \mathrm{H}_\epsilon(\Phi_{1,c}^n)\left(\log\left(p_p(I_c)\right) + C\right) + (1 - \mathrm{H}_\epsilon(\Phi_{1,c}^n))\left(\log\left(p_{pc}(I_c)\right) - \log\left(p_c(I_c)\right)\right),$$

$$\beta_{k,n} = \frac{1}{D_k}, \qquad \beta_{k,s} = \frac{1}{C_k}, \qquad \beta_{k,e} = \frac{1}{A_k}, \qquad \beta_{k,o} = \frac{1}{B_k}.$$

En factorisant dans l'equation 4.10 l'élément $\Phi_{k,c}^{n+1}$ qui doit être mis à jour, on obtient :

$$\Phi_{k,c}^{n+1}\left(1 + \nu\, dt\, \delta_\epsilon(\Phi_{k,c}) \sum_{r\in\mathcal{N}(c)} \beta_{k,r}\right) = \Phi_{k,c}^{n} + dt\, \delta_\epsilon(\Phi_{k,c})\left(f_k(\Phi_{k',c}, C) + \nu \sum_{r\in\mathcal{N}(c)} \beta_{k,r}\Phi_{k,r}^{n+1}\right),$$

et la mise à jour de Φ_k en utilisant la méthode de Gauss Seidel devient :

$$\Phi_{k,c}^{n+1} \leftarrow \frac{\Phi_{k,c}^{n} + dt\, \delta_\epsilon(\Phi_{k,c})\left(f_k(\Phi_{k',c}, C) + \nu\sum\limits_r \beta_{k,r}\Phi_{k,r}^{n+1}\right)}{1 + \nu\, dt\, \delta_\epsilon(\Phi_{k,c})\sum\limits_r \beta_{k,r}}.$$

Les deux courbes de niveau Φ_1 et Φ_2 sont mises à jour en alternant leur résolution en chaque pixel c :

– Pour tous les pixels $c \in \Omega$:

$$1 : \qquad \Phi_{1,c}^{n+1} \leftarrow \frac{\Phi_{1,c}^{n} + dt\, \delta_\epsilon(\Phi_{1,c})\left(f_1(\Phi_{2,c}, C) + \nu\sum\limits_r \beta_{1,r}\Phi_{1,r}^{n+1}\right)}{1 + \nu\, dt\, \delta_\epsilon(\Phi_{1,c})\sum\limits_r \beta_{1,r}},$$

$$2 : \qquad \Phi_{2,c}^{n+1} \leftarrow \frac{\Phi_{2,c}^{n} + dt\, \delta_\epsilon(\Phi_{2,c})\left(f_2(\Phi_{1,c}, C) + \nu\sum\limits_r \beta_{2,r}\Phi_{2,r}^{n+1}\right)}{1 + \nu\, dt\, \delta_\epsilon(\Phi_{2,c})\sum\limits_r \beta_{2,r}}.$$

Cette méthode fait partie des méthodes classiques de résolution d'un système linéaire. Elle consiste à visiter une à une les inconnues $\Phi_{k,c}^{n+1}$ en fixant les autres inconnues $\phi_{k,l}^{n+1}$ avec $l \neq c$.

La méthode Gauss-Seidel [31] utilise immédiatement les termes mis à jour. En pratique, au lieu de sauvegarder les valeurs de Φ_k^{n+1} dans un nouveau vecteur, nous écrasons les données et remplaçons la valeur courante par la nouvelle valeur.

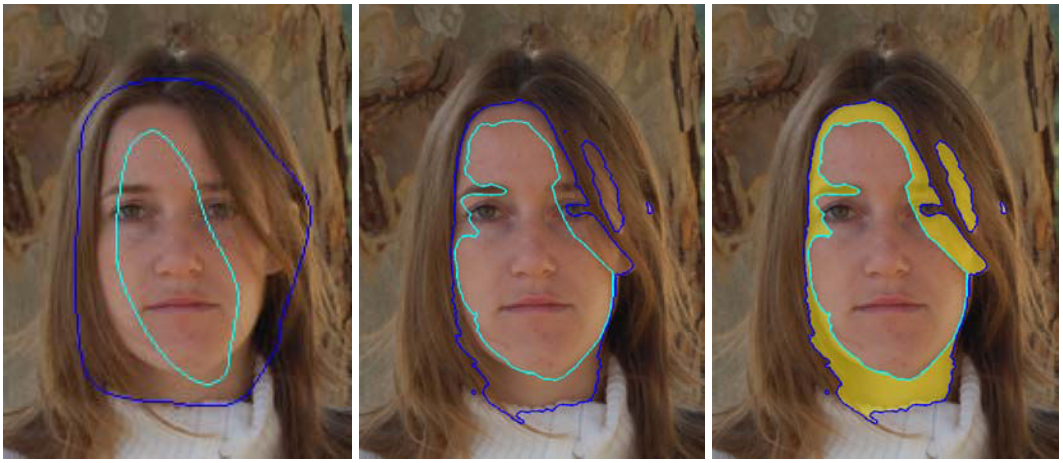
Par rapport à une méthode classique de relaxation du type Jacobi [72], qui met à jour Φ^{n+1} uniquement à partir des données de l'étape précédente Φ^n, l'intérêt de la méthode de Gauss-Seidel est qu'elle utilise deux fois moins de mémoire que la méthode de Jacobi, et qu'elle converge deux fois plus vite [72].

D'autres méthodes telles que la méthode SOR (pour *Successive Over-Relaxation* en anglais) ou bien les méthodes multi-grilles auraient pu être implémentées. La methode SOR est une adaptation de la méthode de Gauss Seidel. Elle introduit un paramètre de *sur-relaxation* pour améliorer la vitesse de convergence, mais dont le choix peut être complexe [71]. La methode multi-grille [5] se fondent sur l'utilisation d'un ensemble de grilles de différentes résolutions. L'idée est d'accélérer la convergence d'une méthode itérative en corrigeant la solution calculée sur une grille fine par celle obtenue rapidement sur une grille plus grossière.

Nous avons fait le choix d'utiliser la méthode de Gauss-Seidel car les résultats sont très satisfaisants et que ainsi nous n'introduisons pas de paramètre supplémentaire.

4.2.3 Résultats

En pratique, il reste à définir le nombre de gaussiennes que nous souhaitons utiliser pour modéliser les régions Ω_p et Ω_c. En plus, il faut déterminer la valeur de ν qui permet de pondérer le terme d'attache aux données avec le terme de régularisation. La valeur de ce paramètre est fixée manuellement par un expert. Des approches de type "validation croisée" sur une large base d'images segmentées manuellement pourraient être utilisées pour apprendre cet hyper-paramètre.



(a) Initialisation (b) Résultat (c) Visualisation de Ω_{pc} en jaune

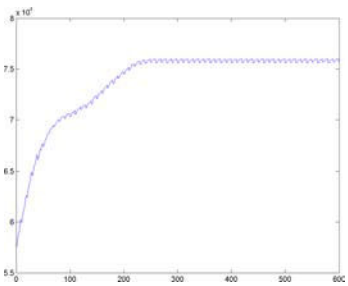
FIGURE 4.9 – Segmentation de la peau et des cheveux en utilisant les méthodes variationnelles

Sur la figure 4.9, les deux courbes en bleu clair et bleu foncé représentent respective-

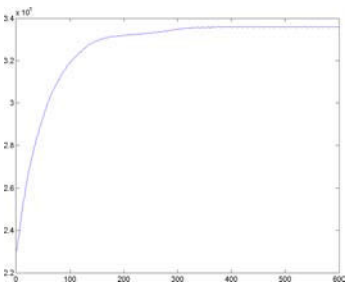
ment, les frontières entre les régions Ω_p et Ω_{pc} et les régions Ω_{pc} et Ω_c. L'image 4.9(a) illustre la phase d'initialisation de la méthode variationnelle. Comme nous l'avons énoncé en introduction l'initialisation est proche de la solution finale. L'objectif de la fonctionnelle est de faire évoluer les courbes jusqu'à ce que l'aire de la régions Ω_{pc} soit minimale.

Les Figures 4.9(b) et 4.9(c) illustrent le résultat de la segmentation. Sur la Figure 4.9(c) la coloration jaune permet de visualiser les pixels appartenant à la région Ω_{pc}. Nous vérifions que cette région contient les pixels correspondants au mélange entre une mèche de cheveux et la tempe droite ou bien une mèche de cheveux et l'oreille droite. Nous constatons que la modélisation des frontières par les fonctions de niveaux Φ permettent un changement de topologie et ainsi faire évoluer la région Ω_{pc} d'une à plusieurs composantes connexes. Ceci permet de segmenter la mèche de cheveux opaque qui est située sur la tempe droite entre deux régions Ω_{pc}.

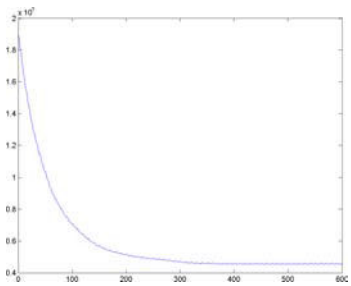
En revanche, nous pouvons voir que les yeux, les sourcils sont des éléments délicats à traiter. Comme nous souhaitons utiliser la segmentation pour améliorer les résultats d'essayage virtuel de lunettes, nous souhaitons que les yeux, tout comme la peau, appartiennent au "calque" inférieur, les cheveux au "calque" supérieur pour pouvoir intercaler le "calque" correspondant aux lunettes (*cf.* Figure 1.17 dans le chapitre 1). Puisque les caractéristiques colorimétriques des yeux sont différentes de celle de la peau. Nous ne pouvons traiter ce problème avec notre modèle actuel. En revanche, une solution à ce problème est envisagée dans le chapitre 6.



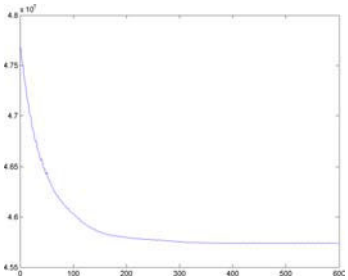
(a) Terme d'attache aux données sur la région Peau



(b) Terme d'attache aux données sur la région Cheveux



(c) Terme d'attache aux données sur la région de mélange



(d) Terme global d'attache aux données

FIGURE 4.10 – Évolution des termes d'attache aux données

La Figure 4.10 illustre l'évolution des termes énergétiques d'attache aux données au cours du temps. Les trois premières courbes représentent respectivement les termes liés aux régions Ω_p, Ω_c et Ω_{pc} ; la dernière courbe représente l'évolution du terme d'attache aux données sur le domaine image Ω. Sur les régions Ω_p et Ω_c, nous constatons que les termes énergétiques augmentent, ceci est dû au fait que le nombre de pixels contenu dans ces régions augmente. Car cela coûte moins cher à l'énergie globale d'attribuer l'étiquette

Peau ou Cheveux à un pixel qui était initialement étiqueté par le mélange peau-cheveux. Même si les énergies correspondant aux régions Ω_p et Ω_c augmentent, celle de la région Ω_{pc} diminue plus rapidement ainsi l'énergie globale d'attache aux données diminue.

(a) $\nu = 0.01$ (b) $\nu = 1$ (c) $\nu = 10$

FIGURE 4.11 – Influence du paramètre ν sur la fonctionnelle 4.2.

Sur la Figure 4.11 les régions Ω_{pc}, colorées en jaune, correspondent à différents résultats de segmentation correspondant à différentes valeurs de ν pour la fonctionnelle présentée à l'équation 4.2. Cette figure illustre le rôle du poids ν qui relie le terme d'attache aux données et le terme de régularisation. Plus la valeur de ν est grande plus le terme de régularisation aura de l'influence sur le résultat final. Le terme énergétique de régularisation agit sur la longueur des courbes de niveau zéro des fonctions Φ_i. L'évolution des courbes est obtenue par une descente de gradient que l'on peut représenter comme un champ de vecteurs représentants la force qui s'applique en chaque point de la courbe. Ce champ de vecteurs est constitué d'une composante représentant la force à appliquer à la courbe pour minimiser sa longueur, cette composante n'est autre qu'un terme de courbure. Nous pouvons interpréter cela par le fait que plus ν est important, moins la courbe oscillera.

4.3 Segmentation précise de l'image portrait

Connaissant le contexte de l'étude nous avons une connaissance *a priori* sur l'organisation des régions. Nous essayons de modéliser les 5 régions discutées dans la section 4.1.1 en ayant la contrainte de ne changer qu'un seul signe des trois fonctions Φ_i pour des régions voisines.

4.3.1 Mise en équation

La fonctionnelle que nous présentons est décomposée en deux énergies, l'une représentant un terme d'attache aux données (E_D) et l'autre un terme de lissage/régularisation de la courbe (E_R).

$$E = E_D + E_R$$

Nous rappelons que nous modélisons le problème comme une segmentation de l'image en cinq régions. Pour définir ces dernières, nous avons besoin de 3 fonctions courbes de niveaux : Φ_1, Φ_2, Φ_3.

Régions	Φ_1	Φ_2	Φ_3	Sélection de la région par produit de fonctions Heaviside
Ω_p	+	+	+	$H(\Phi_1)H(\Phi_2)H(\Phi_3)$
Ω_{pc}	+	+	-	$H(\Phi_1)H(\Phi_2)(1-H(\Phi_3))$
Ω_c	+	-	-	$H(\Phi_1)(1-H(\Phi_2))(1-H(\Phi_3))$
Ω_{cf}	+	-	+	$H(\Phi_1)(1-H(\Phi_2))H(\Phi_3)$
Ω_f	-	-	+	$(1-H(\Phi_1))(1-H(\Phi_2))H(\Phi_3)$

$$
\begin{aligned}
E_D(\Phi_1,\Phi_2,\Phi_3) &= -\int_\Omega \log\left(p_p(\omega)\right)\ H(\Phi_1(\omega))H(\Phi_2(\omega))H(\Phi_3(\omega))\ d\omega \\
&- \int_\Omega \log\left(p_{pc}(\omega)\right)\ H(\Phi_1(\omega))H(\Phi_2(\omega))(1-H(\Phi_3(\omega)))\ d\omega \\
&- \int_\Omega \log\left(p_c(\omega)\right)\ H(\Phi_1(\omega))(1-H(\Phi_2(\omega)))(1-H(\Phi_3(\omega)))\ d\omega \\
&- \int_\Omega \log\left(p_{cf}(\omega)\right)\ H(\Phi_1(\omega))(1-H(\Phi_2(\omega)))H(\Phi_3(\omega))\ d\omega \\
&- \int_\Omega \log\left(p_f(\omega)\right)\ (1-H(\Phi_1(\omega)))(1-H(\Phi_2(\omega)))H(\Phi_3(\omega))\ d\omega \\
&+ \int_\Omega C\left(1-H\left(\Phi_1(\omega)\right)\right)\left(1-H\left(\Phi_2(\omega)\right)\right)\left(1-H\left(\Phi_3(\omega)\right)\right)\ d\omega \\
&+ \int_\Omega C\left(1-H\left(\Phi_1(\omega)\right)\right)H\left(\Phi_2(\omega)\right)\left(1-H\left(\Phi_3(\omega)\right)\right)\ d\omega \\
&+ \int_\Omega C\left(1-H\left(\Phi_1(\omega)\right)\right)H\left(\Phi_2(\omega)\right)H\left(\Phi_3(\omega)\right)\ d\omega
\end{aligned}
\tag{4.11}
$$

$$
E_R = \nu \int_\Omega \left(\left|\nabla H\left(\Phi_1(\omega)\right)\right| + \left|\nabla H\left(\Phi_2(\omega)\right)\right| + \left|\nabla H\left(\Phi_3(\omega)\right)\right|\right) \tag{4.12}
$$

4.3.2 Résultats expérimentaux

La Figure 4.12 illustre nos résultats de segmentation. Les régions en surbrillance rouge, verte et bleu correspondent respectivement aux régions Peau, Cheveux et Fond ; et les régions en surbrillance jaune et turquoise correspondent respectivement aux zones de transition Peau-Cheveux et Cheveux-Fond.

– Dans la zone de transition Peau-Cheveux, nous constatons (comme dans le cas de la segmentation à trois régions) que les pixels peau du front et de l'oreille (*cf.* la partie de l'image délimitée par la frontière jaune sur la Figure 4.13(a)) en partie occultés par les cheveux appartiennent bien à la région Ω_{pc}. Dans cette même région, se trouvent les pixels appartenant aux yeux. En analysant le résultat que nous devons obtenir, nous pouvons remarquer qu'en aucun cas les pixels correspondants à l'image des yeux doivent se superposer sur les lunettes. Ces éléments particuliers du visage devraient appartenir au même "calque" que la peau. Dans les perspectives du chapitre 6, nous proposons une solution à ce problème.

(a) Initialisation (b) Résultat (c) Visualisation des 5 régions

FIGURE 4.12 – Segmentation de la peau, des cheveux et du fond en utilisant les méthodes variationnelles

(a) (b)

FIGURE 4.13 – Localisation des parties de l'image nécessaires pour l'interprétation des résultats.

– Dans la zone de transition Cheveux-Fond, nous constatons qu'une partie des mèches de cheveux se mélangeant au fond (*cf.* la partie de l'image délimitée par la frontière jaune sur la Figure 4.13(b)) ne sont pas contenues dans la zone de transition mais dans la région Fond. Cette erreur est due au critère qui ne prend en compte que l'information colorimétrique, non discriminante dans ce cas de figure. Cependant, pour notre application nous préférons que les faux positifs dans la région Fond que dans la régions Cheveux. Ceci s'argumente par le fait que même si on étiquette Fond une partie de la chevelure, le changement de l'arrière plan peut rester réaliste. Par contre si nous étiquetons Cheveux une partie du fond alors l'incrustation du personnage dans un environnement virtuel apparaît comme irréaliste.

4.4 Conclusion

Dans ce chapitre, nous avons introduit l'utilisation des contours actifs pour la segmentation précise des images de type portrait. A partir des caractéristiques colorimétriques de chacune des classes Peau, Fond et Cheveux, nous avons mis en place une formulation qui permet d'obtenir trois masques complémentaires de segmentation. La précision de la

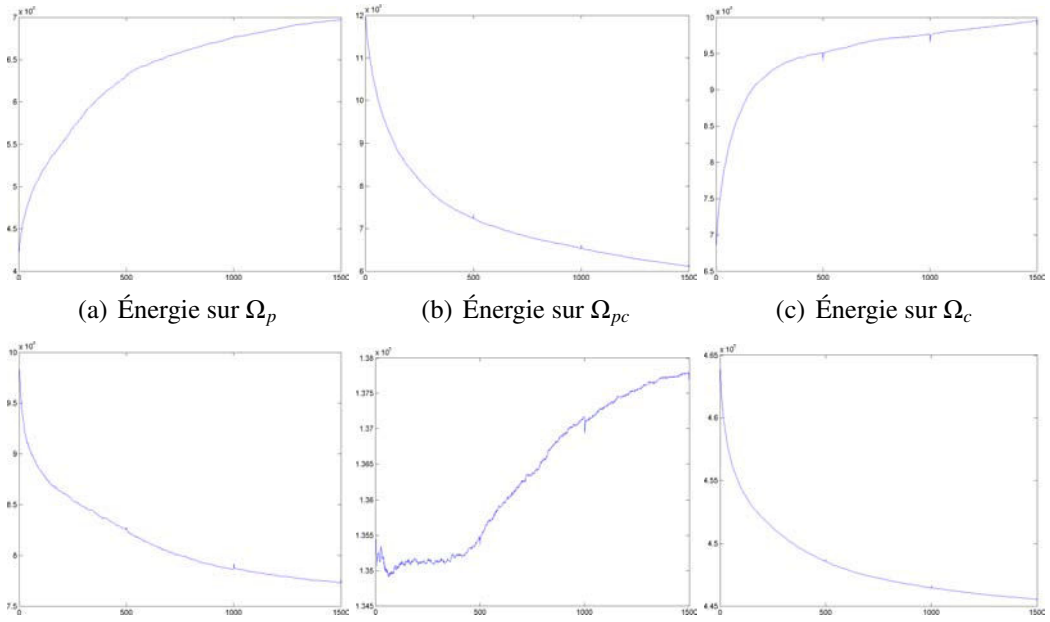
(a) (b) (c)

FIGURE 4.14 – Quelques autres résultats de la segmentation en 5 régions.

segmentation est assurée par la phase de *matting* qui permet de "démélanger" les couleurs associées aux trois classes.

Nous avons essayé de mettre en place une fonctionnelle qui s'adapte le plus possible à la problématique tout ayant le moins de terme possible pour éviter la phase critique de réglage des paramètres. Ce dernier point permet de réduire le nombre de paramètres à régler et donc rend le système plus indépendant des images traitées.

Dans le chapitre 5, nous étudierons plus finement le "contexte" de l'image Portrait, les modélisations par partie d'objet, les modèles de forme dans le but d'introduire automatiquement des connaissances *a priori* sur l'image et supprimer ainsi la phase d'initialisation manuelle utilisée dans le Chapitre 3.



(a) Énergie sur Ω_p (b) Énergie sur Ω_{pc} (c) Énergie sur Ω_c

(d) Énergie sur Ω_{cf} (e) Énergie sur Ω_f (f) Énergie liée aux termes d'attache aux données sur Ω

FIGURE 4.15 – Évolution des différents termes d'attache aux données

(a) Masque de la région Peau. (b) Masque de la région Cheveux. (c) Masque de la région Fond.

FIGURE 4.16 – Résultat de notre segmentation précise après une dernière étape de matting.

(a) Avant segmentation (b) Après segmentation

(c) Avant segmentation (d) Après segmentation

FIGURE 4.17 – Essayage virtuel de lunettes.

Chapitre 5

Utilisation du contexte et des modèles de forme pour la détection des zones Peau, Cheveux et Fond

Dans les chapitres 3 et 4, nous avons présenté deux techniques de segmentation : la segmentation par classification et la segmentation en utilisant les méthodes variationnelles. La première produit une segmentation grossière à partir d'échantillons des régions Peau, Cheveux et Fond, tandis que la seconde produit une segmentation plus précise à la condition, non plus de connaître des échantillons des régions, mais d'être initialisée par une segmentation grossière proche de la solution finale. Nous constatons que les deux méthodes de segmentation se complètent.

Dans ce chapitre, l'objectif est de supprimer l'intervention de l'utilisateur qui, jusqu'alors, venait localiser des échantillons au sein des régions Peau, Fond et Cheveux, que nous appellerons zones. Pour cela, nous nous appuyons sur l'introduction de connaissances *a priori*, préalablement apprises sur une base d'images.

Plus précisément, nous exploitons les caractéristiques géométriques du visage en mettant en place un modèle de forme par parties dans le but d'exploiter le contexte général d'une image Portrait. Nous exploitons par exemple, le fait que les cheveux sont situés autour de la peau, que l'arrière plan est autour du visage et des cheveux...

Les modèles actifs d'apparence permettent de détecter dans l'image Portrait un ensemble de points d'intérêt du visage. A partir de ces données, nous sommes capables de déterminer la position de la zone Peau avec une faible incertitude. En ce qui concerne les zones restantes qui sont celles associées aux cheveux et au fond, nous construisons un modèle statistique de forme pour les cheveux qui permet de détecter la zone Cheveux dans l'image et nous en déduirons la position de la zone Fond. La détection de la zone Cheveux s'appuie sur la minimisation d'une fonctionnelle qui prend en compte simultanément deux types de contours : les contours actifs utilisant les courbes de niveaux et les modèles de forme par point. Cette méthode nous permet d'être robuste en associant une certaine flexibilité des courbes dans un domaine très contraint.

Dans le chapitre 6, nous verrons comment s'articulent l'ensemble des éléments que nous avons présentés au cours des chapitres 3, 4, 5 .

5.1 Problématique

La segmentation des images Portrait pour une application grand public, c'est-à-dire sans contraintes sur l'environnement de prise de vue, est un véritable défi technologique. En raison de la grande variété des images traitées (*cf.* Chapitre 1) , il est difficile de modéliser l'apparence colorimétrique et fréquentielle des trois classes (Peau, Cheveux et Fond) en les rendant séparables indépendamment des images.

Nous avons fait le pari d'être capable de détecter dans chaque image un échantillon de chacune des classes pour apprendre leur apparence en ligne (*cf.* Chapitre 3). Cet apprentissage en ligne permet de s'adapter à l'image traitée. Travaillant dans le riche contexte spatial des images Portrait, nous sommes en mesure de définir certaines caractéristiques géométriques invariantes d'une image à une autre.

Dans la suite, nous faisons un point sur la notion de contexte au sein des images portrait. Ensuite nous définissons et construisons la géométrie de la zone la plus délicate, la zone cheveux. Nous utilisons un modèle actif de forme. Enfin, nous mettons en place l'algorithme de détection des trois zones.

5.2 Contexte pour les objets Peau, Cheveux et Fond d'une image portrait

La notion de contexte joue un rôle crucial pour la perception humaine. Le cerveau humain utilise le contexte pour mieux comprendre une scène ou encore reconnaître des objets. Ce sont ces remarques qui ont motivé les actions de recherche à ce sujet en vision par ordinateur. Nous rappelons que le contexte est composé de trois grands types qui sont le contexte sémantique, le contexte spatial et enfin le contexte d'échelle. Le premier, le contexte sémantique prend en compte la vraisemblance de rencontrer un type objet dans une certaine scène. Le contexte spatial prend en compte la vraisemblance de rencontrer un objet en fonction des autres objets de la scène. Le contexte d'échelle prend en compte la vraisemblance de la taille d'un objet en fonction de celle des autres objet de la scène.

La première référence en vision par ordinateur qui utilise le contexte est probablement le travail de Yakimovsky et Feldman [86], publié en 1973. Il utilise le contexte pour résoudre un problème de segmentation. Plus précisément, les auteurs segmentent des images via une méthode bayésienne en ayant deux objectifs. Ils veulent à la fois segmenter l'image en régions en regroupant les pixels ayant des caractéristiques colorimétriques semblables et introduire des connaissances *a priori* sur la dépendance spatiale inter-régions.

Nous verrons par la suite que le contexte spatial et le contexte d'échelle jouent également un grand rôle pour la détection des zones d'intérêt des images Portrait.

5.2.1 Le contexte d'échelle d'une image Portrait

Le portrait est une scène dont la plage de variation des paramètres géométriques est faible pour une échelle fixée. Les modèles actifs de forme (ASM pour *Active Shape Model* en anglais) exploitent cette propriété. Nous choisissons d'utiliser la position des yeux comme éléments de référence pour définir le contexte d'échelle. Nous définissons la position des zones Peau, Cheveux et Fond en fonction de celle des yeux. Les trois zones

sont ainsi à la même échelle, celle définie par la distance entre les yeux. Pour la détection des 3 zones, nous normalisons géométriquement les images. Pour cela, après détection automatique du visage et des yeux en appliquant l'algorithme de Viola et Jones [79], nous appliquons une similitude $H(\theta, t, s)$ afin d'avoir les yeux toujours aux mêmes positions (cf Figure 5.1), où θ est le paramètre de rotation, t le vecteur de translation dans le plan image et s le facteur d'échelle.

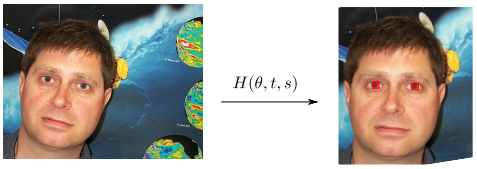


FIGURE 5.1 – Normalisation géométrique des images Portrait par rapport à la position des yeux

5.2.2 Le contexte spatial pour les objets Peau, Cheveux et Fond

La prise en compte du contexte spatial comporte deux objectifs. Le premier est de diminuer voir supprimer les aberrations qui peuvent apparaître lors de la classification pixel à pixel (méthode utilisée dans le Chapitre 3 en utilisant les cartes de probabilité et le terme de régularisation en fonction du voisinage). Le deuxième, qui retient plus particulièrement notre attention dans ce chapitre, vise à définir l'organisation spatiale des objets en présence. **Contraindre des objets ou régions** à vérifier une organisation spatiale connue *a priori* est, dans le domaine de la segmentation, une façon de simplifier le problème et de rendre robuste sa résolution.

En ce qui concerne le cas précis des images Portrait, les objets sont associés aux zones Peau, Cheveux et Fond. Le contexte spatial que nous considérons s'organise comme illustré sur la figure 5.2. Au centre, nous rencontrons l'objet Peau ; au dessus et sur les cotés c'est l'objet Cheveux qui occupe l'espace ; enfin le Fond entoure Peau et Cheveux. Les relations spatiales que nous utilisons ici sont topologiques.

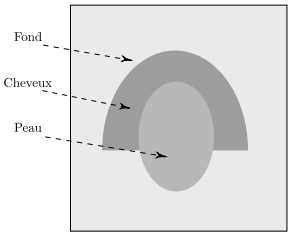


FIGURE 5.2 – Représentation schématique du contexte spatial pour une image de type Portrait

Une étude rapide montre que lorsque l'on présente le schéma de la Figure 5.2 sans étiquetage des régions à un échantillon représentatif de la population française, 77% visualise un portrait. Le contexte spatial est donc très important et permet de contraindre l'espace des solutions de la segmentation. Malgré tout, des sources de variabilité sont à prendre en compte. Elles concernent d'abord les coupes de cheveux.

5.3 Objet Cheveux : une forme complexe

Les frontières de la chevelure sont difficiles à définir et la géométrie de la région associée aux cheveux dans une image est très variable (*cf.* Figure 5.3). Ces trois exemples

(a) (b) (c)

FIGURE 5.3 – Exemples de coupes de cheveux

montrent que la forme de la chevelure varie énormément d'un individu à l'autre. Pour détecter la chevelure dans toutes ces situations, nous cherchons un modèle de forme associé à la chevelure qui révèle les points communs et les différences observées d'un Portrait à l'autre.

5.3.1 Les modèles de forme "existants"

Dans cette section, nous allons établir un bref état de l'art sur les différentes techniques qu'il existe pour faire coïncider l'apparence géométrique d'un objet dans l'image avec un modèle de forme appris préalablement.

Les modèles actifs de forme paramétrés par points Introduits par Tim Cootes *et al.* [16], les modèles actifs de formes (ASM pour *Active Shape Models* en anglais) sont des modèles statistiques de forme d'un objet qui se déforment de manière itérative pour faire correspondre chaque point du modèle avec le point associé de l'objet contenu dans l'image à traiter. La figure 5.4 montre un ensemble de points de contrôle (ou nœud) d'un ASM. Les points peuvent se déplacer avec la déformation du modèle, pour assurer un recalage avec le Portrait traité. Les positions des points du modèle de forme sont contraintes par un modèle obtenu en étudiant la distribution des points spécifiés par un expert sur un jeu de données d'apprentissage. Ce modèle est le résultat d'une analyse en composantes principales. Localiser la meilleure position pour chaque point de la forme contrainte par le modèle peut être obtenu par un critère visant à maximiser la présence de fort contour au niveau des points de contrôle de la forme. Cette technique a été couramment employée, en vision par ordinateur, pour l'analyse du visage [14].

Les modèles actifs de forme paramétrés par des splines Tout comme Cootes *et al*, Cremers *et al* [21] travaillent sur la distribution des points de la silhouette d'un type d'objet. L'originalité de cette méthode vient de la formulation de la fonctionnelle sous-jacente. Les auteurs ont fait le choix d'introduire une contrainte de forme sur l'énergie de Mumford-Shah [56]. Ainsi, en minimisant une seule fonctionnelle, ils sont capables de segmenter une image en maximisant l'homogénéité des intensités des régions tout en s'assurant d'avoir une forme de contour similaire à l'ensemble des formes apprises pour construire le modèle.

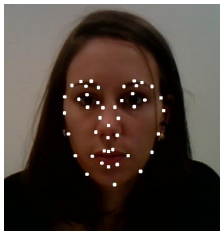


FIGURE 5.4 – Représentation de la paramétrisation par points pour les modèles actifs de forme.

Les modèles actifs de formes représentées par des courbes de niveaux Leventon *et al* [46] proposent d'exprimer les formes par la courbe de niveau zéro de la fonction distance signée. L'objectif est de construire un modèle de forme en étudiant la distribution d'un ensemble de fonctions distances signées. Dans leurs travaux, Leventon *et al* font l'hypothèse que les données sont réparties de manière gaussienne en s'inspirant des travaux cités précédemment sur les ASM de Cootes et Taylor [17]. En appliquant une Analyse en Composantes Principales (ACP) et en ne gardant que les vecteurs propres qui correspondent aux n plus grandes valeurs propres (n est de l'ordre de 5), ils construisent leur modèle de forme. Nous remarquerons que la modélisation par courbe de niveaux est délicate car la combinaison linéaire de deux fonctions distances n'est pas une fonction distance. Cette représentation de la courbe peut créer des aberrations, notamment quand la forme à apprendre en compte est concave, ce qui est généralement le cas pour les objets Cheveux (*cf.* Figure 5.5 deuxième ligne). Nous constatons à travers les deux approches

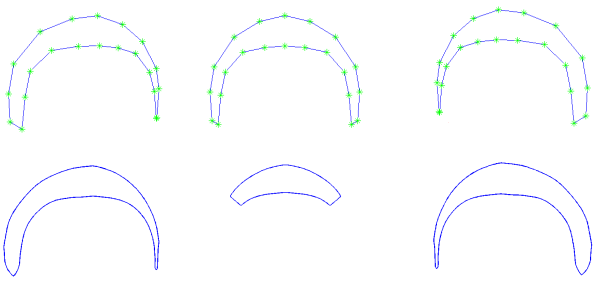


FIGURE 5.5 – Sur la première ligne, nous illustrons la variation d'un modèle actif de forme par point, et sur la deuxième la variation du même mode pour un modèle actif de forme représenté par le niveau zéro de la fonction distance signée.

illustrées à la figure 5.5 que, dans le cas de la modélisation par points, nous utilisons l'information de contour et gardons une certaine continuité dans le déplacement des points de contrôle en passant de la forme en vue de gauche à la forme en vue de droite via la forme en vu de face, alors que pour la modélisation par courbes de niveaux, nous travaillons cette fois-ci avec l'information contenue dans la région. L'utilisation de la fonction distance signée rend instable les pixels proches du contour et de plus en plus stable les pixels qui s'éloignent du contour. Ici ce n'est donc pas l'information de contour qui est primordiale mais la distance au contour.

Les modèles actifs d'apparence Les modèles actifs d'apparence (AAM pour *Active Appearance Models* en anglais), introduit par Cootes *et al.* [15] en 1998, sont des mo-

dèles statistiques qui permettent de synthétiser conjointement la forme et la texture. Ils sont construits à partir d'une Analyse en Composantes Principales (ACP). L'objectif étant d'extraire les points caractéristiques de l'objet traité, qui est le visage dans notre cas. Ceci se fait par recherche de la valeur des meilleurs paramètres, en minimisant la distance entre l'image à traiter et l'image synthétisée du modèle. Cette modélisation énoncée par Cootes *et al.* comporte des faiblesses. Les AAM ne sont pas robustes aux changements d'illumination, d'expression du visage et d'identité. Le Gallou [43] et Pizarro *et al.* [62] proposent de rendre robuste ce modèle. La figure 5.6 illustre les résultats de Pizarro *et al.*. Ils proposent une modélisation qui s'affranchi des changements d'illumination. Nous constatons en effet que la texture recalée est éclairée différemment que la texture du visage à détecter, et que néanmoins l'algorithme converge vers la solution voulue.

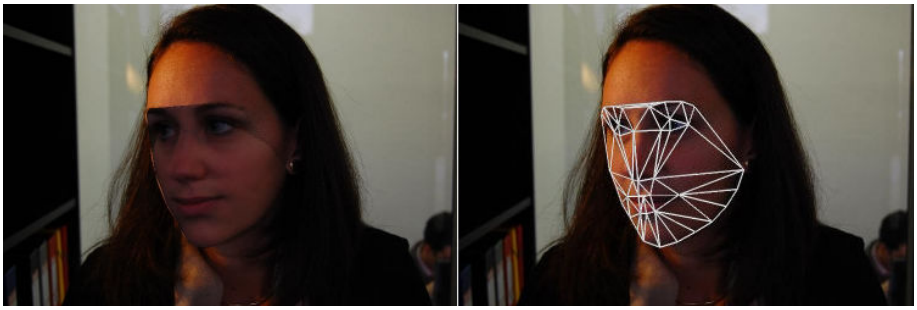


FIGURE 5.6 – Exemple d'AAM robustes aux changements d'illumination [62]

5.3.2 Construction d'un modèle de forme robuste pour les cheveux : UHSM

La détection d'une zone Cheveux (et j'appuie sur l'idée de détection d'un échantillon de cheveux et non de segmentation de l'ensemble de la région Cheveux) doit permettre par la suite d'apprendre les caractéristiques fréquentielles et colorimétriques de la classe Cheveux comme au chapitre 3. Nous souhaitons adopter une méthode robuste qui ne détecte pas forcément la totalité des pixels Cheveux de l'image mais du moins une partie de ces pixels. La première tentative de localisation de la zone cheveux est fondée sur une simple étude statistique des images normalisées. Une carte de probabilité permet de fixer la position de la zone de l'image dans laquelle nous sommes sûrs de rencontrer des cheveux si cheveux il y a (*cf.* Figure 5.7). Pour certains pixels, la probabilité d'appartenir à la chevelure est, dans nos conditions opératoires, élevées (jusqu'à 0.96) . Cette partie correspond à la partie supérieure de la chevelure dont la base correspond aux tempes. Après cette remarque, nous choisissons de construire un modèle de forme qui permet de détecter la position et la forme optimale d'une zone cheveux. Ce modèle est appelé UHSM pour *Upper Hair Shape Model*.

La construction du modèle de forme pour la zone cheveux s'intéressent uniquement à la partie supérieure de la coupe de cheveux. Notre protocole est fondé sur l'utilisation de N images annotées à la main (dans notre cas 60 images et leurs symétriques, soit $N = 120$). La base d'images est composée d'un échantillon des images Portrait représentant la totalité des coupes de cheveux que l'ont peut rencontrer : cheveux courts, mi-long, long, avec ou sans frange... Dans tous les cas, nous nous intéressons uniquement à la chevelure située au dessus des points des tempes. Nous recensons la géométrie de la

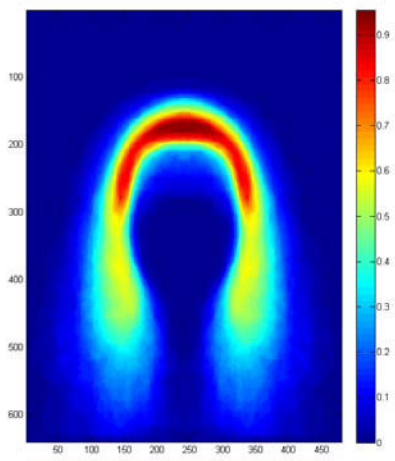


FIGURE 5.7 – Carte de probabilités des pixels cheveux dans le domaine image (plus les couleurs sont chaudes, plus la probabilité est grande).

chevelure en utilisant une modèle par point. Ce dernier est constitué des coordonnées de 22 points représentant la forme de la zone cheveux (soit un vecteur $\in \mathbb{R}^{44}$ illustré à la figure 5.9). Afin d'extraire caractéristiques de la distribution des points de la chevelure pour la base d'apprentissage, nous mettons en forme ces données en une matrice $M \in \mathbb{R}^{44\times N}$ sur laquelle nous appliquons la méthode de l'Analyse en Composantes Principales (ACP).

Nous obtenons alors une base de vecteurs propres dont les quatre premiers, notés v_1, v_2, v_3, v_4, associés aux plus grandes valeurs propres encodent 87% de l'information. Nous remarquons que les vecteurs propres v_1,v_2 , v_3 et v_4 encodent respectivement la variation d'apparence de la forme des cheveux lorsque la position 3D de la tête varie de haut en bas, de droite à gauche, la largeur de la coupe de cheveux sur le haut du crâne et enfin la variation de la position des cheveux à hauteur des points correspondants aux tempes. Ces modes sont illustrés à la figure 5.8. Pour chaque ligne de la figure, la troisième forme correspondant à la forme moyenne de l'ensemble des chevelures.

5.4 Utilisation du contexte via l'UHSM pour la détection des zones Peau, Cheveux et Fond

Le savoir-faire de FittingBox permet de détecter la présence d'un visage dans l'image et également quelques points d'intérêt sur ce dernier comme les points des commissures des yeux et de la bouche, les points des tempes (*cf.* Figure 5.13). C'est à partir de la position de ces points d'intérêt que nous initialisons la position des trois zones d'échantillonage de la Peau, des Cheveux et du Fond notées respectivement Z^p Z^c et Z^f (*cf.* section 5.4.2). Ces trois zones, illustrées sur la figure 5.10, ont les propriétés géométriques suivantes :

– Z^p est une zone fiable que nous fixons par un rectangle dont la taille et la position sont fixées et définies par rapport à la position des yeux. La longueur, notée L^p, du rectangle est égale à la distance entre la position des yeux, sa largeur est définie par $l^p = 0.15\times L^p$. Cette zone est située en dessous de yeux à une distance $d_1 = 0.2\times L^p$

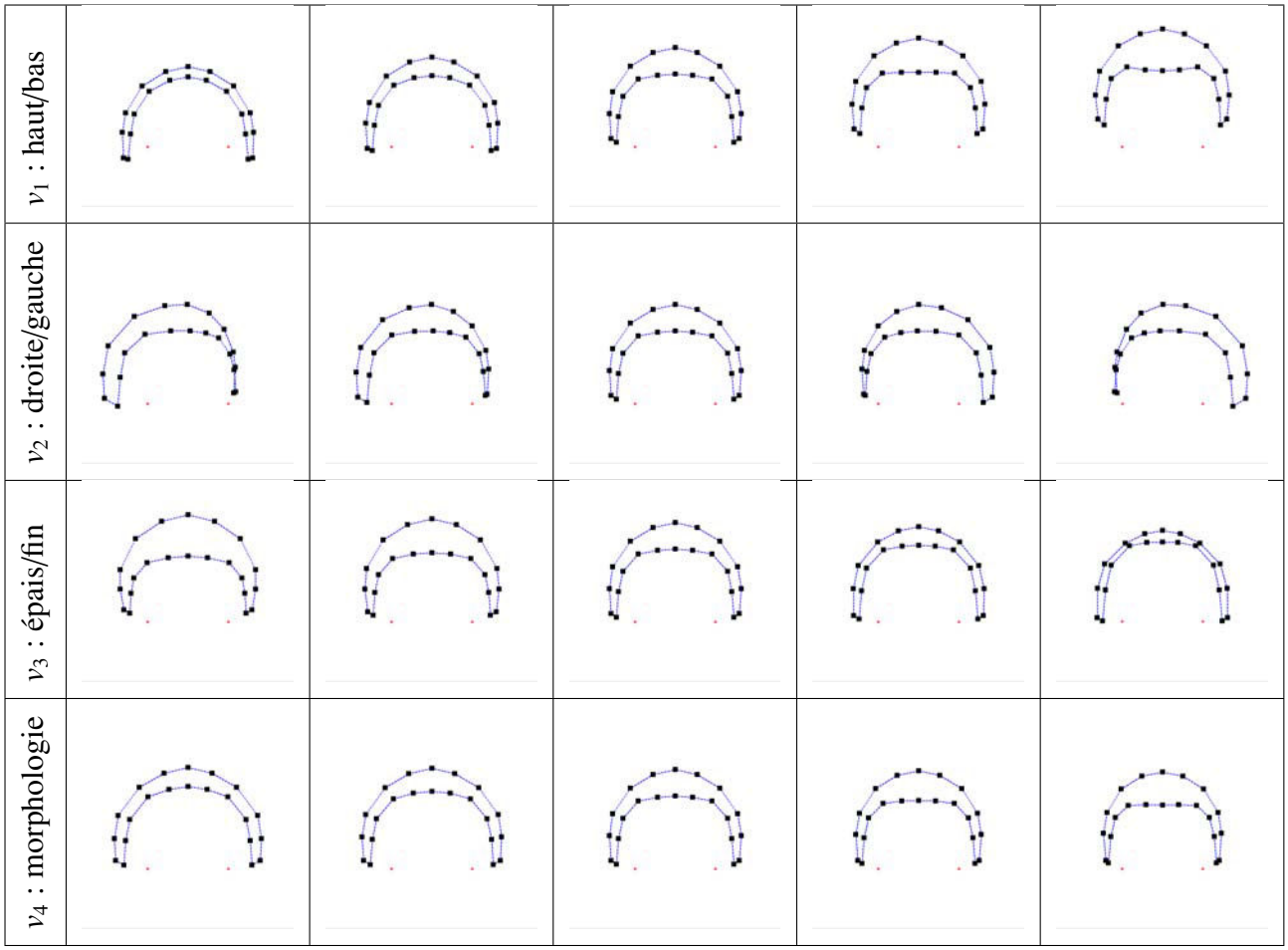


FIGURE 5.8 – Illustation des différents modes du modèle de forme correspondant à la zone Cheveux. Les points rouges correspondent à l'emplacement des commissures extérieures des yeux. La première ligne illustre la variation du premier mode : haut/bas. La deuxième ligne illustre la variation du deuxième mode : droite/gauche. Les troisième et quatrième lignes illustrent la variation des troisième et quatrième modes : chevelure fine/épaisse au niveau du dessus du crâne et la variation de morphologie (distance entre les yeux et les points des tempes).

à la verticale de la position des yeux. Cette zone correspondant à la régions qui à le moins de probabilité d'avoir des cheveux, de la barbe. Mais aussi le fait de récupérer l'information sur le relief du nez donne des informations sur les variations d'incidence de l'éclairage et donc des teintes de couleurs.

– Z^c est la zone définie par notre modèle de forme, l'UHSM. Il est paramétré par le jeu de paramètres (v_1, v_2, v_3 et v_4). Les positions optimales de l'ensemble des points de la forme sont obtenues via la minimisation de l'énergie, sur les paramètres. Cette énergie est introduite dans la section suivante.
– Z^f est également une zone rectangulaire est située à une distance suivant la verticale $d_2 = 0.1 \times L^p$ de l'UHSM. La longueur du rectangle, notée L^f vaut $L^f = 2 \times L^p$, et la largeur vaut $l^f = 0.35 \times L^p$.

L'ensemble de ces données sont définies par expérience, une étude probabiliste pourrait apporter plus d'information.

FIGURE 5.9 – Un extrait de la base d'apprentissage de l'UHSM

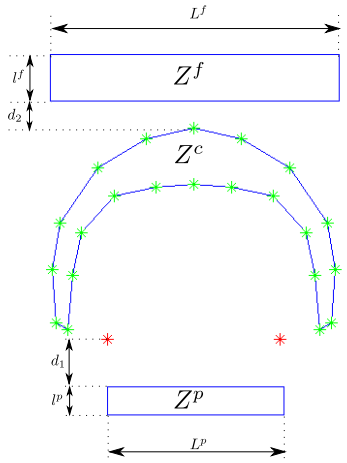


FIGURE 5.10 – Représentation des zones correspondant à la peau, aux cheveux et au fond.

Dans la suite, nous présentons notre modélisation pour la détection des zones cheveux et fond (la zone peau étant fixée). Ensuite, nous développons la phase correspondant à l'initialisation des zones ainsi que la résolution du problème d'optimisation du critère énergétique choisi. Enfin, nous montrons quelques résultats.

5.4.1 Mise en équation

La modélisation de la détection des zones repose sur deux concepts complémentaires, l'un appliqué aux contours des zones de la détection et l'autre à l'information contenue à l'intérieur de ces zones.

Pour le premier concept, c'est suite à la lecture de l'article de Leventon et *al.* [46] qui propose une segmentation en combinant une approche par contour actif et un *a priori* de forme, que nous adaptons cette approche à notre problème. Nous proposons une énergie de détection des zones qui représentent la frontière de la zone cheveux par une combinaison d'une forme active et d'un contour actif. Indépendamment, ces deux méthodes donnent des résultats erronés ; sur la figure 5.11, nous constatons qu'une partie de l'arrière plan est détecté comme appartenant à la zone cheveux. Par contre en combinant les méthodes, la zone détectée ne contient que des pixels cheveux, cette modélisation rend la détection plus robuste.

Pour le second concept, nous mettons en place un terme d'attache aux données qui vise à maximiser les dissemblances inter-zones et maximiser la ressemblance intra-zone.

Nous avons synthétisé ces concepts sur le schéma de la figure 5.12. La forme active

(a) Contour actif (b) Forme active (c) Forme et contour actifs

FIGURE 5.11 – Détection robuste de la zone cheveux en combinant l'approche par contour actif et forme active.

évolue en maximisant la dissemblance colorimétrique des zones et en restant proche de la position du contour actif. Puis, le contour actif évolue à son tour en maximisant l'homogénéïté des zones, en maximisant l'aire de la région associée au contour actif et en restant proche de la position de la forme. Et nous réitérons le procédé jusqu'à convergence.

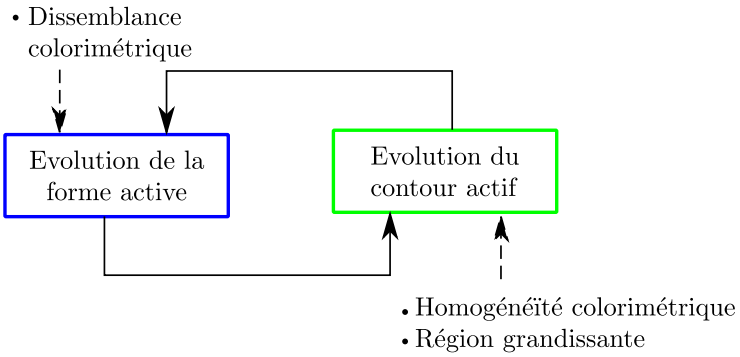


FIGURE 5.12 – Représentation schématique de l'énergie (équation 5.1) à minimiser pour la détection des zones Z^p, Z^c et Z^f.

Le contour actif est formellement défini par le niveau zéro de la fonction distance signée ϕ. La forme active est définie par l'UHSM paramétré par le vecteur ν. L'énergie de détection des zones s'écrit formellement à l'aide de quatre termes :

$$E(c, \nu, \phi) = E_b(\phi) + E_d^1(\phi, c) + E_a(\phi, \nu) + E_d^2(\nu). \tag{5.1}$$

Les trois premiers termes sont des énergies courantes, l'originalité de la fonctionnelle intervient sur le dernier terme. Le terme E_b est un terme de force ballon proposé par Cohen [11] qui encourage la région définie par $\phi > 0$ à grandir. Cette énergie s'écrit alors,

$$E_b(\phi) = -\lambda_b \int_\Omega H_\varepsilon(\phi(\omega))\, d\omega, \tag{5.2}$$

avec H_ε la fonction Heaviside régularisée définie dans le chapitre 4 à la section 4.3.1.

Plus la région cheveux est grande, plus la région vérifiant $\phi > 0$ contient de pixels, plus l'intégrale sur le domaine image de l'image de ϕ par la fonction Heaviside est importante, et donc finalement moins l'énergie est grande.

Le terme E_1^d vise à maximiser l'homogénéïté de la zone Cheveux correspondant à $\phi > 0$ comme dans [8], il s'écrit :

$$E_d^1(\phi, c) = \lambda_{d1} \int_\Omega \|I(\omega) - c\|^2 H_\varepsilon(\phi(\omega))\, d\omega, \tag{5.3}$$

avec I l'image à segmenter et c la valeur moyenne de la couleur sur la région correspondant à $\phi > 0$:

$$c = \frac{\int_\Omega I(\omega)H(\omega)\, d\omega}{\int_\Omega H(\omega)\, d\omega}$$

Le terme E_a pénalise l'écart entre ϕ et ϕ^*, pour coupler les contours actifs qualifiés de "libres", notées ϕ, aux formes actives, notées ϕ^*, paramétrées par v. Nous avons choisi de représenter cela par la formulation :

$$E_a(\phi, v) = \lambda_a \int_\Omega (\phi(\omega) - \phi^*(v, \omega))^2\, d\omega. \tag{5.4}$$

Ce terme relie la dépendance entre le modèle de zone Cheveux décrit dans la section 5.3.2 et l'image. La paramétrisation de l'UHSM et la connaissance de la position des yeux nous permettent d'optimiser les zones Peau, Cheveux et Fond et de les extraire.

Nous attendons de ces trois zones qu'elles aient des propriétés colorimétriques dissemblables. Pour cela, nous calculons sur chaque zone des histogrammes exprimés dans l'espace colorimétrique $YCbCr$. Les histogrammes sont notés h^Z_{CbCr} et h^Z_{YCb} avec $Z = P, C, F$. Le score de dissemblance entre paire d'histogrammes est donné par la distance de Bhattacharyya d_B.Les dissemblances à maximiser sont entre la Peau et les Cheveux et ensuite les Cheveux et le Fond.

Le dernier terme de notre fonctionnelle s'écrit alors :

$$E_d^2(v) = -\lambda_{d2} \sum_{R=B(v),S} d_B(h^{H(v)}_{CbCr}, h^Z_{CbCr}) + d_B(h^{H(v)}_{YCb}, h^Z_{YCb}), \tag{5.5}$$

Nous notons que les quatre paramètres, aussi appelés poids λ_b, λ_{d1}, λ_a et λ_{d2}, sont des constantes strictement positives que nous avons définies de manière empirique.

5.4.2 Initialisation

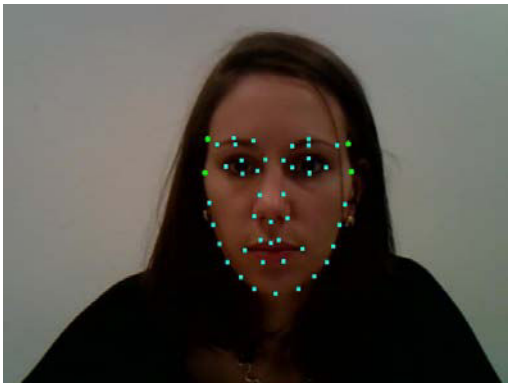


FIGURE 5.13 – Les points d'intérêt du visage détectés par les algorithmes de FittingBox

La méthode proposée par Cristinacce et Cootes [22] fondée sur un modèle local contraint (CLM pour *Constrained Local Model* en anglais) permet de détecter avec robustesse un ensemble de points d'intérêt du visage. Parmi ces points détectés (*cf.* figure 5.13), certains, comme ceux correspondant aux commissures extérieures des yeux, aux tempes, sont communs à notre modélisation par UHSM. Ces jeux de points sont utilisés pour initialiser la position, l'échelle de l'UHSM (via la position des yeux) et le vecteur de paramètre v (via la position des points des tempes, noté t_1 et t_2). Nous remarquons que la position les points des tempes par rapport à celle des yeux nous donne de l'information sur les modes v_1, v_2 et v_4 de l'UHSM correspondant respectivement à la variation haut/bas de la tête, droite/gauche et sa morphologie. Seul le mode v_3, lié à l'épaisseur de la chevelure, n'est pas concerné. Nous faisons le choix d'initilialiser le modèle avec une chevelure fine et d'introduire dans la fonctionnelle le terme énergétique 5.2 visant à agrandir la région délimitée par l'UHSM.

$$v_{init} = \underset{v,\ v_3=0}{\operatorname{argmin}} \left(\|{t_1}^{\mathrm{UHSM}}(v) - {t_1}^{\mathrm{CLM}}\|^2 + \|{t_2}^{\mathrm{UHSM}}(v) - {t_2}^{\mathrm{CLM}}\|^2 \right). \tag{5.6}$$

où ${t_i}^{\mathrm{CLM}}$ et ${t_i}^{\mathrm{UHSM}}(v)$ avec $i \in \{1, 2\}$ sont respectivement les coordonnées des points des yeux et des tempes détectés par l'algorithme CLM, et celles obtenues en appliquant le vecteur de paramètre v à notre UHSM.

5.4.3 Résolution

La dépendance non linéaire de $E(c, v, \phi)$ par rapport à v rend la minimisation de la fonctionnelle non triviale. Le minimum de la fonctionnelle est calculé par itérations successives.

Ayant (c^n, v^n, ϕ^n) à l'itération n, nous itérons les 3 étapes suivantes :

1. Mise à jour de c^{n+1} à partir de la moyenne de I sur la région $\phi^n > 0$.
2. Sachant c^{n+1} et ϕ^n , calculer v^{n+1} en utilisant la méthode du simplex [41] (par exemple en employant la fonction *fminseach* de Matlab). v^{n+1} étant le minimum de $E(c^{n+1}, v, \phi^n)$.
3. Mise à jour de $\phi^{n+1} = \phi^n - dt\nabla_\phi E$ avec dt l'intervalle de temps algorithmique et $\nabla_\phi E$ le gradient de E par rapport à ϕ.

$$\nabla_\phi E = \delta_\varepsilon(\phi) \left(-\lambda_b + \lambda_{d1} \|I - c\|^2 \right) + 2\lambda_a(\phi - \phi^*).$$

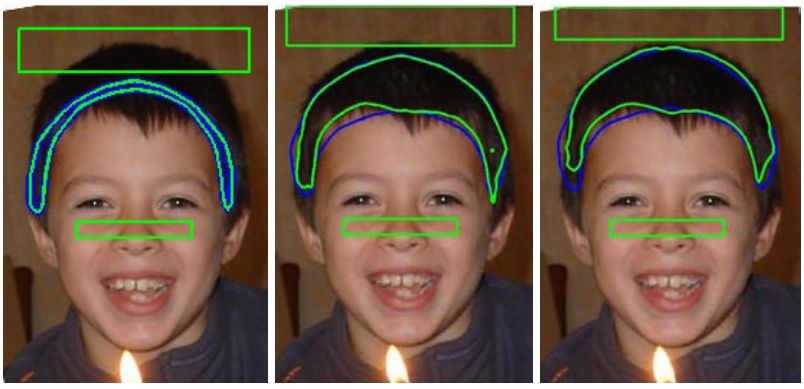
avec $\delta_\varepsilon(.)$ la régularisation de la fonction de Dirac (*cf.* Chapitre 4 section 4.3.1).

La distribution des paramètres de l'UHSM est supposée uniforme. Nous ne voulons pas favoriser ou défavoriser la détection d'un ensemble de forme de cheveux. Nous avons autant de chance de converger vers le jeu de paramètre (0,0,0,0) qui correspond à la forme moyenne des cheveux que vers le jeu de paramètre (1,0,0,0) qui correspond à forme des cheveux tournés vers la droite.

Nous initialisons les paramètre de l'UHSM avec $v^0 = v_{init}$, comme défini dans la section 5.4.2 et $\phi^0 := \phi^*(v^0)$. Nous itérons le processus jusqu'à convergence des paramètres (v) du modèle de forme.

5.4.4 Résultats

Sur la Figure 5.14, nous montrons les différentes phases de notre algorithme de détection des zones Peau, Cheveux et Fond. Nous rappelons que seules la localisation des zones cheveux et fond sont optimisées, la localisation de la zone Peau est fixée. Ces zones sont délimitées par les courbes vertes. Pour la zone correspondant aux cheveux, la courbe vertes correspond au niveau zéro de la fonction ϕ de l'équation 5.1. Quant à la courbe bleue, elle délimite la forme active paramétrée par v. Nous constatons que dès la première itération, nous nous rapprochons fortement de la solution finale. Sur la solution finale, nous constatons que les 2 deux courbes correspondant aux frontières de la zone cheveux sont proches l'une de l'autre. De plus, nous voyons que les pixels contenus dans la forme active des cheveux (courbe bleue) et appartenant à la peau et au fond ne sont pas pris en compte par le contour actif (courbe verte). Le rôle de la forme active est de contraindre la localisation de la zone cheveux et celui du contour actif d'accorder de la flexibilité à la frontière pour supprimer les faux positifs obtenus par l'autre contour.



(a) Initialisation (b) 1ère itération (c) Convergence après 12 itérations

FIGURE 5.14 – Illustration de l'évolution de la détection des zones.

Sur la figure 5.15, nous montrons 6 résultats de détection des zones Peau, Cheveux et Fond minimisant la fonctionnelle 5.1

5.5 Perspectives

Nous avons choisi de mettre en place un modèle de forme robuste et pour cela nous avons restreint le domaine d'étude de la forme en ne modélisant que la partie haute de la chevelure. Nous pourrions envisager d'enrichir notre modèle, sans négliger la robustesse, en modélisant un ensemble de coupes de cheveux possibles. Le modèle que l'on a développé jusqu'à présent serait un modèle de type coupe courte, on pourrait faire de même sur des coupes au carré, des cheveux mi-long, sur des cheveux longs, ... Le choix des différents types de modèles pourrait être inspiré des techniques de dessin tout comme l'ont fait Chen *et al* [9]. Leur principe est de modéliser et détecter les vêtements d'une image en demandant à un artiste de dessiner les vêtements de l'image. Une analyse "grammaticale" des composants du dessin doit conduire à un arbre décisionnel pour la détection des vêtements.

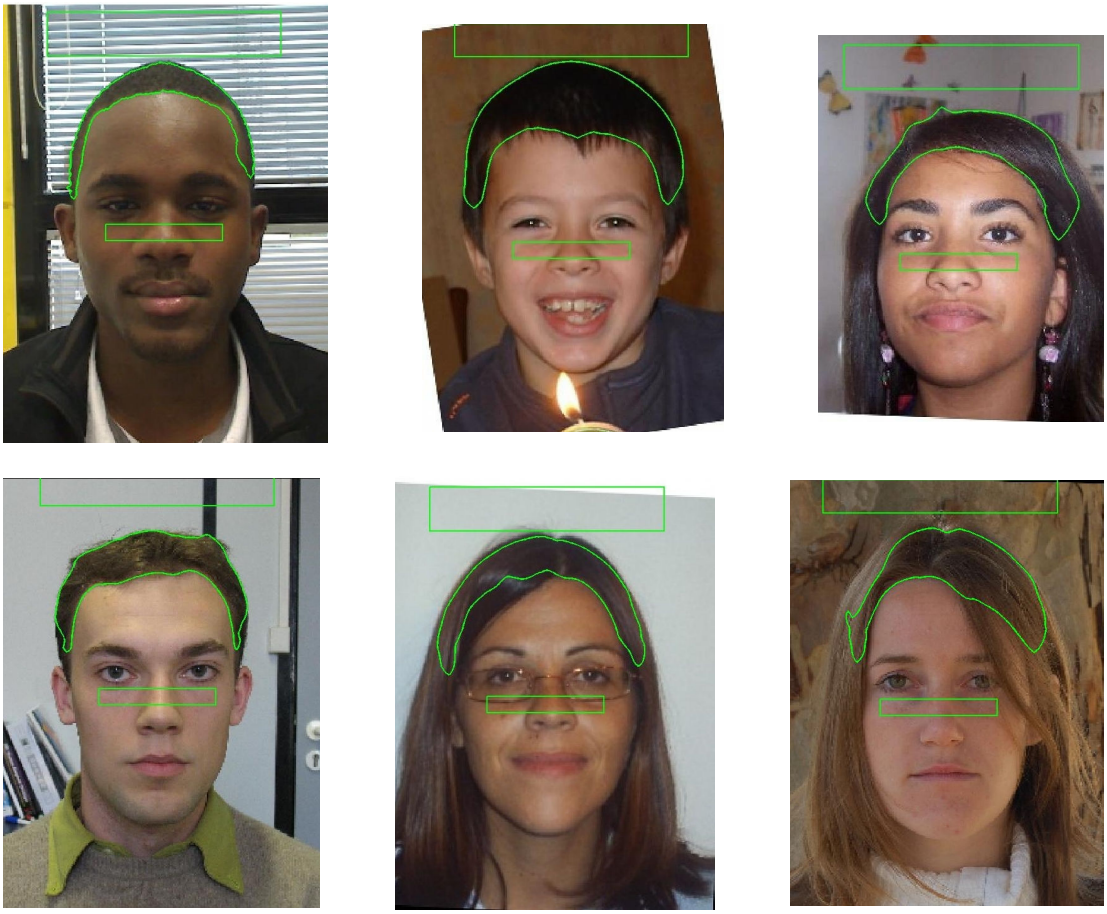


FIGURE 5.15 – Résultats de la détection des zones Peau, Cheveux et Fond.

5.6 Conclusion

Nous avons mis en place une fonctionnelle permettant de détecter nos trois zones d'intérêt. La modélisation prend en compte une formulation paramétrique pour assurer la robustesse des résultats. Cette robustesse est due au fait que la zone de recherche est fortement contrainte par le contexte dans lequel nous travaillons. La modélisation prend également en compte une représentation plus souple des frontières via l'introduction de la fonction courbe de niveau. Cette dernière permet de relâcher la contrainte introduite par la modélisation par points.

Les résultats de cette méthode assurent une détection des zones *a priori* Peau, Cheveux et Fond. Et donc une initialisation fiable pour l'étape de classification détaillée dans le chapitre 3.

Chapitre 6

Conclusion et Perspectives

Dans ce manuscrit de thèse nous avons présenté l'ensemble de nos travaux sur la segmentation des images de type Portrait. Cette segmentation est composée des trois éléments suivants : la peau, la chevelure et l'arrière plan. La segmentation de ce type d'image apporte de réelles solutions pour l'application réaliste d'essayage virtuel de lunettes, mais pas seulement. En effet, ce système peut également être appliqué par les visagistes pour des applications telles que l'essayage de nouvelles coupes de cheveux, de couleurs de cheveux ou encore pour des applications de caractérisation ou de modélisation d'un visage.

Les objectifs visés étaient de présenter une approche pour la segmentation automatique des régions Peau, Cheveux et Fond dans une image Portrait.

À partir des connaissances *a priori* de la scène traitée, nous devions modéliser les régions avec le plus de précision et le moins d'ambiguïté possible. Par précision, nous entendions la définition fidèle d'une segmentation. Par ambiguïté, nous entendions l'impact du choix de la localisation des frontières. Dans cette conclusion, nous proposons de dresser un bilan de notre étude. Puis nous proposerons plusieurs perspectives de recherches pour poursuivre nos travaux, résoudre certains problèmes soulevés par notre étude et explorer certaines pistes que nous n'avons pas eu le temps de traiter au cours de cette thèse.

6.1 Bilan fonctionnel

Dans cette thèse, nous proposons un système complet pour segmenter précisément des images Portraits. Le résultat de cette segmentation se présente en trois calques, chacun contenant la quantité d'information des classes Peau, Cheveux et Fond. Ce système est construit par un enchaînement d'étapes :

Étape 1 : Détection du visage et de ses points d'intérêt.

Étape 2 : Détection des zones *a priori* (*cf.* Chapitre 5).

Étape 3 : Segmentation grossière par classification en trois classes (*cf.* Chapitre 3).

Étape 4 : Segmentation précise par les méthodes variationnelles et l'introduction de régions de transition (*cf.* Chapitre 4).

Étape 5 : "Démélange" des couleurs sur les régions de transition par la technique de *matting*.

Après avoir étudié les différentes étapes dans les chapitres précédents, ici, nous nous concentrons sur les transitions entre les étapes.

En sortie de l'étape 1, nous disposons d'un ensemble de points d'intérêt du visage et en particulier les coordonnées des commissures extérieures des yeux et les coordonnées des points de tempes. Ces informations suffisent à l'initialisation des positions des zones *a priori* Peau, Cheveux et Fond.

En sortie de l'étape 2, nous disposons de la position des zones *a priori* Peau, Cheveux et Fond. Ces informations suffisent à l'initialisation de l'étape 3.

En sortie de l'étape 3, nous disposons d'une segmentation en trois classes/régions et pour l'initialisation de l'étape 4, nous avons besoin d'une segmentation en cinq régions. Nous rappelons que les deux régions ajoutées correspondent aux régions de transitions Peau/Cheveux et Cheveux/Fond. Une méthode naïve consiste à créer une région de transition à partir de la segmentation grossière en dilatant de quelques pixels (le nombre de pixels étant fonction de la distance entre les commissures externes des yeux) la frontière entre deux classes. Ainsi nous faisons à apparaître les deux régions manquantes.

En sortie de l'étape 4, nous disposons d'une segmentation en cinq régions, parmi lesquelles nous comptons 2 régions de transitions. C'est sur ces deux régions de transitions et en s'appuyant sur les 3 autres régions que nous pouvons définir le coefficient d'opacité α entre les régions Peau et Cheveux ou Cheveux et Fond.

Outre la mise en équation des différentes étapes de notre système, nos travaux reposent également sur une phase non négligeable d'analyse du problème et de choix de conception. C'est le résultat de cette phase qui nous a orienté vers un système en cascade.

6.2 Perspectives

Ces travaux de thèse ouvrent de nombreuses perspectives, des améliorations pour notre approche de segmentation, des pistes de recherche pour la caractérisation des frontières des régions, des pistes de nouvelles applications d'essayage virtuel.

La segmentation

À court terme, nous pouvons proposer plusieurs pistes qui permettraient d'améliorer la méthode de segmentation proposée :

- *Mettre en place une approche multi-échelle :* Cette approche permettrait de robustifier les résultats de segmentation et surtout d'accélérer les temps de calcul. Une technique pourrait être de travailler non plus sur chaque pixel mais sur un groupe de pixels. Ce groupement pourrait être obtenu en utilisant les "super-pixels" [64].
- *L'introduction de la région vêtements :* nous pensons que l'introduction de ce segment peut permettre de définir plus précisément le Fond et ainsi réduire l'ambigüité de classification.
- *Traiter indépendamment les élements du visage (les yeux, la bouche, le nez) :* nous envisageons d'utiliser les points d'intérêt du visage (obtenus grâce à la méthode des modèles locaux contraints - CLM) pour que ces éléments ne rentrent pas en compte dans le terme énergétique de la segmentation par méthode variationnelle.
- *La mise en place d'un indicateur :* nous proposons de mettre en place, à chaque étape du système, un indicateur permettant de qualifier le degré de confiance de notre résultat de segmentation.
- *Introduction d'un terme fréquentiel pour la détection des zones a priori :* Actuellement l'étape de détection des zones *a priori* peau, cheveux et fond, est le résultat d'une mise en équation faisant intervenir l'organisation spatiale des objets et leurs propriétés de dissemblance colorimétrique. Nous pourrions envisager d'ajouter un terme visant à exploiter l'information fréquentielle.

L'essayage virtuel

À court terme, la première perspective est l'implémentation de l'algorithme de détection des zones *a priori* et de classification pixel à pixel en langage C pour l'intégrer au logiciel FitPhoto de la société FittingBox. Cette extension à l'application d'essayage virtuel pourra permettre d'améliorer le rendu photoréaliste lors de la phase d'essayage des lunettes. À plus long terme, nous pouvons envisager de réutiliser notre algorithme pour segmenter la chevelure et introduire un module d'essayage virtuel de nouvelles coupes de cheveux.

Annexe A

La variation totale pour le calcul de la longueur de la courbe

L'idée principale est de donner un sens aux discontinuités de la fonction Heaviside (ces discontinuité correpondant aux frontières des régions). Pour cela nous exploitons les propriétés de la fonctionnelle de variation totale (TV pour *Total Variation* en anglais). Étant donnée une fonction $u : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^n$ ouvert, on peut définir, lorsque u est presque partout différentiable sa variation totale de la façon suivante. On définit $|\nabla u|$ par

$$|\nabla u| = \sqrt{\sum_{i=1}^{n} u_{\omega_i}^2}$$

où les u_{x_i} sont les dérivées partielles de u dans la direction ω_i.

$$TV(u) = \int_\Omega |\nabla u| \, d\boldsymbol{\omega} \tag{A.1}$$

Nous supposons que pour tout $\boldsymbol{\omega} \in \Omega$,$\nabla u(\boldsymbol{\omega}) \neq 0$. Nous pouvons alors écrire :

$$\begin{aligned} TV(u) &= \int_\Omega |\nabla u| \, d\boldsymbol{\omega} \\ &= \int_\Omega \frac{\nabla u}{|\nabla u|} \cdot \nabla u \, d\boldsymbol{\omega}. \end{aligned} \tag{A.2}$$

Le champ de vecteurs $\frac{\nabla u}{|\nabla u|}$ est partout de norme 1. On considère sur Ω les champs de vecteurs

$$\boldsymbol{\omega} \mapsto \phi(\boldsymbol{\omega}) = (\phi_1(\boldsymbol{\omega}), \ldots, \phi_n(\boldsymbol{\omega}))$$

Il est facile de voire que l'égalité (A.2) ci-dessus peut se réécrire

$$TV(u) = \sup_{\|\phi\|_\infty = 1} \int_\Omega \phi \cdot \nabla u \, d\boldsymbol{\omega}$$

où

$$\|\phi\|_\infty = \sup_{\boldsymbol{\omega} \in \Omega} |\phi(\boldsymbol{\omega})|, \quad |\phi(\boldsymbol{\omega})| = \sqrt{\sum_{i=1}^{n} \phi_i(\boldsymbol{\omega})^2}.$$

Il est clair qu'on peut remplacer la contrainte $\|\phi\|_{\infty=1}$ par la contrainte plus générale $\|\phi\|_{\infty \leq 1}$ sans que ça ne change la borne supérieure (c'est une conséquence simple de l'inégalité de Cauchy-Schwarz).

Si maintenant ϕ est C^1 sur Ω, on peut alors utiliser la formule de Green et obtenir

$$\int_\Omega \phi \cdot \nabla u \, d\omega = -\int_\Omega \mathrm{div}(\phi) u \, d\omega + \int_{\partial\Omega} (\phi \cdot \mathbf{n}) \, u \, ds \tag{A.3}$$

où $\mathbf{n}$ est le champ des normales extérieures à Ω le long de $\partial\Omega$ et ds la mesure le long de $\partial\Omega$. Bien que dans le cas idéal nous avons $\phi = \frac{\nabla u}{|\nabla u|}$, nous pouvons toujours approcher au plus près ϕ par une fonction qui s'annule sur $\partial\Omega$. Ainsi le terme de bord de l'équation A.3 s'annule.

Nous notons $C_c^1(\Omega)^n$ l'espace des champs ayant n composantes qui sont différentiables sur Ω et qui s'annulent au bord de Ω, nous obtenons alors

$$TV(u) = \sup \left\{ -\int_\Omega u \mathrm{div}(\phi) \, d\omega, \quad \phi \in C_c^1(\Omega)^n, \quad \|\phi\|_\infty \leq 1 \right\}, \tag{A.4}$$

nous remarquons que nous pouvons supprimer le signe "$-$" quitte à remplacer ϕ par $-\phi$.

La formule de l'équation A.4 est alors utilisée comme définition de la variation totale et on dit que $u : \Omega \to \mathbb{R}$ est une fonction à variations bornées si $TV(u) < \infty$.

Dans notre cas particulier nous cherchons à définir la longueur d'une courbe Γ délimitant la frontière entre les régions complémentaires Ω_{int} et Ω_{ext} dans Ω. Nous définissons la fonction caractéristique $\chi_{\Omega_{int}}$ du sous ensemble $\Omega_{int} \subset \Omega$ par :

$$\chi_{\Omega_{int}}(\omega) = \begin{cases} 1 & \text{si } \omega \in \Omega_{int} \\ 0 & \text{sinon.} \end{cases}$$

Grâce à l'équation (A.4), la variation totale de $\chi_{\Omega_{int}}$ s'écrit :

$$\int_\Omega \chi_{\Omega_{int}} \mathrm{div}\phi \, d\omega = \int_{\Omega_{int}} \mathrm{div}(\phi) \, d\omega$$

puisque la fonction $\chi_{\Omega_{int}}$ est nulle hors de Ω_{int}, et en utilisant le théorème de la divergence :

$$\int_{\Omega_{int}} \mathrm{div}\phi \, d\omega = -\int_{\partial\Omega_{int}} \phi \cdot \mathbf{n} \, ds$$

où cette fois $\mathbf{n}$ est la normale extérieure à Ω_{int} le long de $\partial\Omega_{int}$ et ds la mesure le long de $\partial\Omega_{int}$. Pour maximiser l'expression ci-dessus, il suffit de prendre $\phi(\omega) = -\mathbf{n}$ le long de $\partial\Omega_{int}$ et ça donne immédiatement

$$TV(\chi_{\Omega_{int}}) = \int_{\partial\Omega_{int}} ds = \mathcal{L}(\Gamma)$$

la longueur de $\partial\Omega_{int}$ dans Ω.

Ici aussi, la formule (A.4) permets de calculer $TV(\chi_{\Omega_{int}})$ et ainsi donne une façon de calculer $\mathcal{L}(\Gamma)$. Par la suite la fonction caractéristique $\chi_{\Omega_{int}}$ est représentée à l'aide de la fonction composée $H \circ \Phi : \Omega \to \mathbb{R}$, où H est la fonction Heaviside et Φ une fonction courbe de niveau avec si $\omega \in \Omega_{int}$ alors $\Phi(x) > 0$ et sinon $\Phi(x) \leq 0$. Nous pouvons alors écrire :

$$\int_\Omega |\nabla H(\Phi)| \, d\omega = \mathcal{L}(\Gamma).$$

Annexe B

Quelques compléments sur le calcul du gradient

Pour comprendre le calcul du vecteur gradient de la fonction E, ∇E_Φ, nous devons revenir sur la définition de la différentielle. Dans l'espace euclidien, la différentielle de E au point $\Phi \in \mathbb{R}^N$ est défini par l'unique forme linéaire $L_\Phi : \mathbb{R}^N \to \mathbb{R}$, qui vérifie :

$$E(\Phi + h) = E(\Phi) + L_\Phi(h) + \circ(h)$$

Pour calculer cette forme linéaire, nous introduisons la notion de dérivée directionnelle. La dérivée de E en Φ dans la direction du vecteur $\mathbf{v}$ s'écrit :

$$dE_\Phi \mathbf{v} = \lim_{t \to 0} \frac{E(\Phi + t\mathbf{v}) - E(\Phi)}{t}.$$

En posant $l(t) = E(\Phi + t\mathbf{v})$, on remarque à la fois que :

- la forme linéaire peut être définit par : $L_\Phi(\mathbf{v}) = l'(0)$,
- la dérivée directionnelle s'écrit : $dE_\Phi \mathbf{v} = l'(0)$.

On en déduit la simple relation :

$$L_\Phi(\mathbf{v}) = dE_\Phi \mathbf{v}.$$

Les dérivées partielles sont définies à partir de la dérivée directionnelle. Par exemple, la dérivée partielle de E par rapport à la i-ème variable Φ_{ω_i} est définie par :

$$\frac{\partial E}{\partial \Phi_{\omega_i}}(\Phi) := dE_\Phi \ \mathbf{e}_i,$$

où $\mathbf{e}_i$ est le i-ème de la base naturelle de $\mathbb{R}^N$.

Nous remarquons que l'écriture de la dérivée directionnelle et des dérivées partielles sont très proches mais il est important de souligner qu'elles se distinguent dans le choix de la base de $\mathbb{R}^N$ dans laquelle on exprime le vecteur directeur $\mathbf{v}$. Pour $\mathbf{v} = \sum_{i=1}^N v_i \, \mathbf{e}_i$, nous avons la relation suivante :

$$dE_\Phi \mathbf{v} = \sum_{i=1}^{N} v_i \frac{\partial E}{\partial \Phi_{\omega_i}}(\Phi),$$

cette relation est en fait un produit scalaire :

$$\sum_{i=1}^{N} v_i \frac{\partial E}{\partial \Phi_{\omega_i}} = \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial E}{\partial \Phi_{\omega_1}} \\ \vdots \\ \frac{\partial E}{\partial \Phi_{\omega_N}} \end{pmatrix}.$$

En résumé nous pouvons écrire :

$$dE_\Phi \mathbf{v} = \langle \nabla E_\Phi, \mathbf{v} \rangle, \tag{B.1}$$

où ∇E_Φ est le gradient de E dans la base naturelle.

Ce résultat se généralise en dimension infinie. Si Ω est un ouvert de $\mathbb{R}^n$ (dans notre cas $n = 2$) et on note $L^2(\Omega)$ l'espace des fonction de Ω dans $\mathbb{R}$ et de carré sommable avec la convention que deux fonctions égales presque partout sont égales. Nous écrivons alors l'équation suivante :

$$f, g \mapsto \langle f, g \rangle = \int_\Omega fg \, d\omega,$$

elle définie un produit scalaire sur un espace de Hilbert.

Bibliographie

[1] A. A. Abin, M. Fotouhi, and S. Kasaei. Skin segmentation based on cellular learning automata. In *MoMM '08 : Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, pages 254–259, New York, NY, USA, 2008. ACM. *Cité page 26*

[2] M. Aharon, M. Elad, and A. Bruckstein. The k-svd : An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Transactions on Signal Processing*, 54(11) :4311–4322, November 2006. *Cité page 57*

[3] G Aubert, M. Barlaud, and O. Faugeras. Image segmentation using active contours : Calculus of variations or shape gradients. *SIAM Applied Mathematics*, 63 :2003, 2003. *Cité page 50*

[4] R. T. Azuma. A survey of augmented reality. *Presence : Teleoperators and Virtual Environments*, 6(4) :355–385, August 1997. *Cité page 9*

[5] J. H. Bramble. *Multigrid Methods*. Longman Scientic & Technical, 1993. Pitman Research Notes in Mathematics Series #294. *Cité page 102*

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision (IJCV 97)*, 22 :61–79, 1997. *Cité page 47*

[7] S. Chabrier, B. Emile, C. Rosenberger, and H. Laurent. Unsupervised performance evaluation of image segmentation. *Eurasip Journal on Applied Signal Processing*, 15(3) :298–312, March 2006. *Cité page 65*

[8] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10 :266–277, 2001. *3 citations pages 49, 94, et 121*

[9] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. In *CVPR '06 : Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 943–950, Washington, DC, USA, 2006. IEEE Computer Society. *Cité page 123*

[10] C. Cifarelli, G. Manfredi, and L. Nieddu. Statistical face recognition via a k-means iterative algorithm. *Fourth International Conference on Machine Learning and Applications*, pages 888–891, 2008. *Cité page 32*

[11] L. D. Cohen. On active contour models and balloons. In *CVGIP : Image*, 1991. *Cité page 120*

[12] L. D. Cohen, E. Bardinet, and Nicholas Ayache. Surface reconstruction using active contour models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'93)*, 1993. *Cité page 49*

[13] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'93)*, 15 :1131–1147, November 1993. *Cité page 46*

[14] T. F. Cootes, G. Edwards, and C.J. Taylor. Comparing active shape models with active appearance models. In *in Proc. British Machine Vision Conf*, pages 173–182, 1999. *Cité page 114*

[15] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer, 1998. *2 citations pages 28 et 115*

[16] T. F. Cootes and C. J. Taylor. Active shape model search using local grey-level models : A quantitative evaluation. In *In proceedings of the British Machine Vision Conference*, 1993. *Cité page 114*

[17] T. F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1) :38–59, 1995. *2 citations pages 26 et 115*

[18] T.F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision. Technical report, University of Manchester, 2000. *Cité page 28*

[19] C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20 :1–25, 1995. *Cité page 32*

[20] D. Cremers, M. Rousson, and R. Deriche. Review of statistical approaches to level set segmentation : Integrating color, texture, motion and shape. *International Journal of Computer Vision (IJCV'07)*, 72 :215, 2007. *2 citations pages 50 et 53*

[21] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes : Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3) :295–313, 2002. *Cité page 114*

[22] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference (BMVC'06)*, pages 929–938, Edinburgh, UK, 2006. *Cité page 122*

[23] M. Doi and S. Tominaga. Image analysis and synthesis of skin color textures by wavelet transform. *IEEE Southwest Symposium on Image Analysis and Interpretation*, 0 :193–197, 2006. *Cité page 26*

[24] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000. *Cité page 61*

[25] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 895–900, Washington, DC, USA, 2006. IEEE Computer Society. *Cité page 53*

[26] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5 :2443–2446, 1999. *2 citations pages 55 et 56*

[27] A. Ford and A. Roberts. Colour space conversions. Technical report, 1998. *Cité page 25*

[28] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4 :933–969, December 2003. *Cité page 42*

[29] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55 :119–139, August 1997. *Cité page 33*

[30] C. Galleguillos and S. Belongie. Context based object categorization : A critical survey. *Computer Vision and Image Understanding (CVIU)*, 114 :712–722, 2010. *Cité page 19*

[31] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. *Cité page 102*

[32] G. Gomez, M. Sanchez, and L. E. Sucar. On selecting an appropriate colour space for skin detection. In *MICAI '02 : Proceedings of the Second Mexican International Conference on Artificial Intelligence*, pages 69–78, London, UK, 2002. Springer-Verlag. *Cité page 25*

[33] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *International Conference on Computer Vision (ICCV'99)*, 1999. *Cité page 38*

[34] S. Jehan-Besson, M. Barlaud, and G. Aubert. Dreamms : Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *IJCV*, 53 :365–380, 2002. *Cité page 46*

[35] Z. Jiang, M. Yao, and W. Jiang. Skin detection using color, texture and space information. In *Fourth International Conference on In Fuzzy Systems and Knowledge Discovery*, volume 3, pages 366–370, Los Alamitos, CA, USA, 2007. IEEE Computer Society. *Cité page 26*

[36] O. Juan and R. Keriven. Unsupervised segmentation for digital matting. Technical report, Centre Enseignement Recherche Traitement Information Systèmes (CERTIS), 2004. *Cité page 91*

[37] P. Julian, V. Charvillat, C. Dehais, and F. Lauze. On the interest of texture for face segmentation. In *Orasis*, 2009. *Cité page 53*

[38] M. Kampmann. Segmentation of a head into face, ears, neck and hair for knowledge-based analysis-synthesis coding of videophone sequences. In *International Conference on Image Processing*, pages 876–880, 1998. *Cité page 37*

[39] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International journal of computer Vision*, pages 321–331, 1988. *3 citations pages 8, 45, et 46*

[40] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 810–, Washington, DC, USA, 1995. IEEE Computer Society. *Cité page 47*

[41] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9 :112–147, 1998. *Cité page 122*

[42] F. Lauze and M. Nielsen. From inpainting to active contours. *International Journal on Computer Vision (IJCV'08)*, 79 :31–43, August 2008. *Cité page 50*

[43] S. Le Gallou. *Détection robuste des éléments faciaux par Modèles Actifs d'Apparence*. PhD thesis, Université de Rennes 1, 2007. *Cité page 116*

[44] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision (IJCV'89)*, 3 :73–102, 1989. *Cité page 50*

[45] K.C. Lee, D. Anguelov, B. Sumengen, and S.B. Gokturk. Markov random field models for hair and face segmentation. In *Automatic Face and Gesture Recognition (FG08)*, pages 1–6, 2008. *4 citations pages 40, 41, 42, et 71*

[46] M. E. Leventon, W. E. L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Computer Vision and Pattern Recognition (CVPR 00)*, volume 1, page 1316, Los Alamitos, CA, USA, 2000. IEEE Computer Society. *2 citations pages 115 et 119*

[47] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 :228–242, 2008. *Cité page 39*

[48] U. Lipowezky, O. Mamo, and A. Cohen. Using integrated color and texture features for automatic hair detection. In *Convention of Electrical and Electronics Engineers in Israel*, 2008. *Cité page 45*

[49] C. Liu. A bayesian discriminating features method for face detection. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI'03)*, 25 :725–740, 2003. *Cité page 35*

[50] Z.-Q. Liu, J.Y. Guo, and L. Bruton. A knowledge-based system for hair region segmentation. In *Signal Processing and Its Applications, 1996. ISSPA 96., Fourth International Symposium on*, volume 2, pages 575 –576, August 1996. *2 citations pages 37 et 45*

[51] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. *Cité page 31*

[52] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR'08*, pages –1–1, 2008. *3 citations pages 53, 54, et 64*

[53] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. volume 41, pages 3397–3415, 1993. *Cité page 55*

[54] F. Marquès and V. Vilaplana. Face segmentation and tracking based on connected operators and partition projection. *Journal of the pattern recognition society*, 2000. *Cité page 45*

[55] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, Vol E77-D, No.12, 1994. *Cité page 9*

[56] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5) :577–685, 1989. *2 citations pages 49 et 114*

[57] S. Nicolau, X. Pennec, L. Soler, X. Buy, A. Gangi, N. Ayache, and J. Marescaux. An augmented reality system for liver thermal ablation : Design and evaluation on clinical cases. *Medical Image Analysis*, 13(3) :494–506, June 2009. *Cité page 10*

[58] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging,Vision,and Graphics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003. *Cité page 50*

[59] E. Osuna, R. Freund, and F. Girosi. Training support vector machines : an application to face detection. In *Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997. *Cité page 33*

[60] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 555–, Washington, DC, USA, 1998. IEEE Computer Society. *Cité page 29*

[61] N. Paragios and R. Deriche. Geodesic active regions : a new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation, Special Issue on Partial Differential Equations in Image Processing, Computer Vision and Computer Graphics*, 13(1/2) :249–268, march/june 2002. *Cité page 50*

[62] D. Pizarro, J. Peyras, and A. Bartoli. Light-invariant fitting of active appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. *Cité page 116*

[63] M. A. Ranzato, C. Poultney, S. Chopra, and Y. Lecun. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems (NIPS 2006*. MIT Press, 2006. *Cité page 53*

[64] X. Ren and J. Malik. Learning a classification model for segmentation. *IEEE International Conference on Computer Vision (ICCV'03)*, 1 :10, 2003. *Cité page 127*

[65] C Rousset and P. Y. Coulon. Frequential and color analysis for hair mask segmentation. In *International Conference on Image Processing (ICIP 08)*, pages 2276–2279. IEEE, 2008. *2 citations pages 39 et 40*

[66] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Computer Vision on Pattern Recognition (CVPR'03)*, pages 699–704, 2003. *Cité page 53*

[67] S. K. Singh, D. S. Chauhan, M. Vatsa, and Singh R. A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, Vol. 6, No. 4, :227–234, 2003. *Cité page 26*

[68] K. Sobottka and I. Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing : Image Communication*, Vol. 12, No. 3 :263–281, 1998. *Cité page 45*

[69] M. B. Stegmann. Active appearance models : Theory, extensions and cases. Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, aug 2000. *Cité page 28*

[70] M. Subasic, S. Loncaric, and J. Birchbauer. Expert system segmentation of face images. *Expert Syst. Appl.*, 36(3) :4497–4507, 2009. *Cité page 45*

[71] Thomas. *Numerical Partial Differential Equations : Conservation Laws and Elliptic Equations*. Springer, 1 edition, 1999. *Cité page 102*

[72] J.W. Thomas. *Numerical Partial Differential Equations II (Conservation Laws and Elliptic Equations)*. Texts in Applied Mathematics, 1999. *Cité page 102*

[73] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991. *Cité page 29*

[74] R. Unnikrishnan and M. Hebert. Measures of similarity. In *Seventh IEEE Workshop on Applications of Computer Vision*, pages 394–400, January 2005. *Cité page 64*

[75] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979. *Cité page 65*

[76] J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. In *Advances in Neural Information Processing Systems*, volume 20, pages 1553–1560, jan 2008. *Cité page 77*

[77] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50 :271–293, 2002. *Cité page 90*

[78] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1 :511, 2001. *Cité page 31*

[79] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision (IJCV 01)*, 2001. *6 citations pages 15, 29, 36, 37, 39, et 113*

[80] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV 04)*, 57 :137–154, 2004. *Cité page 20*

[81] D. Wang, S. Shan, W. Zeng, H. Zhang, and X. Chen. A novel two-tier bayesian based method for hair segmentation. In *Proceedings of the 16th IEEE international conference on Image processing (ICIP'10)*, pages 2377–2380, Piscataway, NJ, USA, 2010. IEEE Press. *Cité page 42*

[82] N. Wang, H. Ai, and S. Lao. A compositional exemplar-based model for hair segmentation. In *The 10th Asian Conference on Computer Vision (ACCV'10)*, 2010. *2 citations pages 42 et 45*

[83] J. Weickert, C. Feddern, M. Welk, B. Burgeth, and T. Brox. *PDEs for Tensor Image Processing*. 2005. *Cité page 99*

[84] C. Xu and J. L. Prince. Gradient vector flow : A new external force for snakes. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0 :66, 1997. *Cité page 47*

[85] Y. Yacoob and L. S. Davis. Detection and analysis of hair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7) :1164–1169, 2006. *2 citations pages 37 et 38*

[86] Y. Yakimovsky and J. A. Feldman. A semantics-based decision theory region analyzer. In *IJCAI'73 : Proceedings of the 3rd international joint conference on Artificial intelligence*, pages 580–588, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc. *Cité page 112*

[87] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images : a survey. *IEEE Trans. on Pattern Analisys and Machine Intelligence*, 24(1) :35–58, January 2002. *2 citations pages 23 et 36*

[88] B. D. Zarit, B. J. Super, and F. K. H. Quek. Comparison of five color models in skin pixel classification. In *In ICCV'99 Int'l Workshop on*, pages 58–63, 1999. *2 citations pages 25 et 26*

[89] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research, 2010. *Cité page 36*

[90] Z. Zhang, H. Gunes, and M. Piccardi. An accurate algorithm for head detection based on xyz and hsv hair and skin color models. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1644 –1647, 2008. *Cité page 45*