DETECTION ET CATEGORISATION D'OBJETS EN MOUVEMENT DANS UNE VIDEO

Youssef Zinbi

UNIVERSITÉ DE
CAEN BASSE NORMANDIE

Unité de Formation et de Recherche Sciences
École doctorale : SIMEM

THÈSE

Discipline : Informatique
présentée et soutenue par :

Mr ZINBI Youssef
Le 14/01/2009

pour obtenir le titre de :
Docteur en Sciences
de l'Université de Caen Basse Normandie

Titre:

DETECTION ET CATEGORISATION D'OBJETS EN MOUVEMENT DANS UNE VIDEO

Mr Abderrahim EL MOATAZ Professeur, Université de Caen (Directeur de thèse)
Mr Youssef CHAHIR Maître de conférence, Université de Caen (Co-Directeur)
Mr Liming CHEN Professeur, Ecole Centrale de Lyon (Examinateur)
Mme Florence SEDES Professeur, IRIT de Toulouse (Rapporteur)
Mme Su RUAN Professeur, IUT de Troyes (Rapporteur)
Mr Christophe ROSENBERGER, Professeur GREYC-Caen (Examinateur)

Table des matières

Liste des figures

Détection et catégorisation d'objets en mouvement dans une vidéo

Résumé

Dans le contexte de l'analyse vidéo, il est important d'avoir des méthodes de segmentation intelligentes et rapides pour fournir un aperçu rapide du contenu des séquences vidéo. Dans le cadre de cette thèse, nous nous intéressons particulièrement à des problèmes d'extraction et de catégorisation des objets vidéos.

Pour l'extraction, nous proposons d'utiliser l'approche par contours actifs globaux basés régions qui permet de localiser rapidement les objets d'intérêt. Pour cela, nous avons utilisé des critères de segmentation qui prennent en compte l'homogénéité et les attributs perceptuels pour définir une compétition entre la région d'intérêt et le fond. Pour améliorer la méthode de détection et de suivi de données vidéo, nous avons étendu la formulation énergétique de notre modèle des contours actifs globaux en incluant une force supplémentaire issue du calcul du flot optique.

Dans une seconde partie, nous abordons le problème de l'interprétation du comportement humain (mouvement et gestuelles) dans les séquences vidéo. Les buts poursuivis sont multiples. D'un côté, nous procédons à une analyse du mouvement humain. Le terme "analyse" concerne ici l'extraction d'informations bas-niveau, tels que la silhouette de la personne, la localisation de son visage, l'extraction et classification de son expression faciale. D'un autre côté, on propose une méthode de catégorisation qui faciliterait la réduction de données et de dimensionnalité des données, ainsi que l'interprétation du comportement humain. Il s'agit de la reconnaissance de démarches (marche, course etc.), de postures (debout, accroupi, etc.), ou entre des personnes (gestes, attitudes etc.).

Mots clés : Objet visuel, segmentation, contours actifs, surfaces actifs, expression faciale, composantes faciales, classification spectrale, comportement humain, reconnaissance des postures

CHAPITRE 1 Introduction

1 Introduction

1.1 Problématique

Actuellement, la technologie numérique est omniprésente dans nos vies, dans divers moyens de communication tels que les téléphones portables, l'Internet à haut débit et la télévision numérique. La variété de contenus de ces documents peuvent être soit des journaux télévisés, des émissions sportives, des films, des documentaires, ou bien des enregistrements de vidéosurveillance. Chaque type de document possède sa propre structure qui le distingue. Dans ce contexte, il est apparu nécessaire d'avoir une représentation de bas niveau et de haut-niveau qui permettra de fournir une analyse du contenu. La représentation bas-niveau vise à extraire les objets visuels à l'aide de critères tels que la couleur, la texture, et le mouvement. La représentation haut-niveau vise à fournir une description sémantique du contenu de la vidéo, en définissant le comportement de ces objets. La complexité de la tâche a poussé les chercheurs à se pencher sur deux sujets de recherche importants qui sont l'extraction d'objets visuels et leur catégorisation.

Un document vidéo n'est pas qu'une suite d'images, il contient une structure hiérarchique équivalent d'un ouvrage. En général, on peut décomposer la vidéo en plusieurs niveaux la figure 1.1 illustre les niveaux suivants: objet, plan, scène et séquence qui sont considérés comme les niveaux de base pour l'analyse des documents vidéos.

- L'objet vidéo se rapporte à un objet 3D du monde réel projeté sur un plan.
- Une image peut contenir plusieurs objets sémantiques mais seulement quelques objets d'intérêt.
- Un plan est défini comme une séquence d'images prises sans interruption temporelle par la caméra.
- Les scènes sont définies comme des collections de plans, adjacents dans le temps, et qui sont liées sémantiquement.
- Une séquence est un ensemble de scènes appartenant au même élan de narration et d'émotion cinématographique.

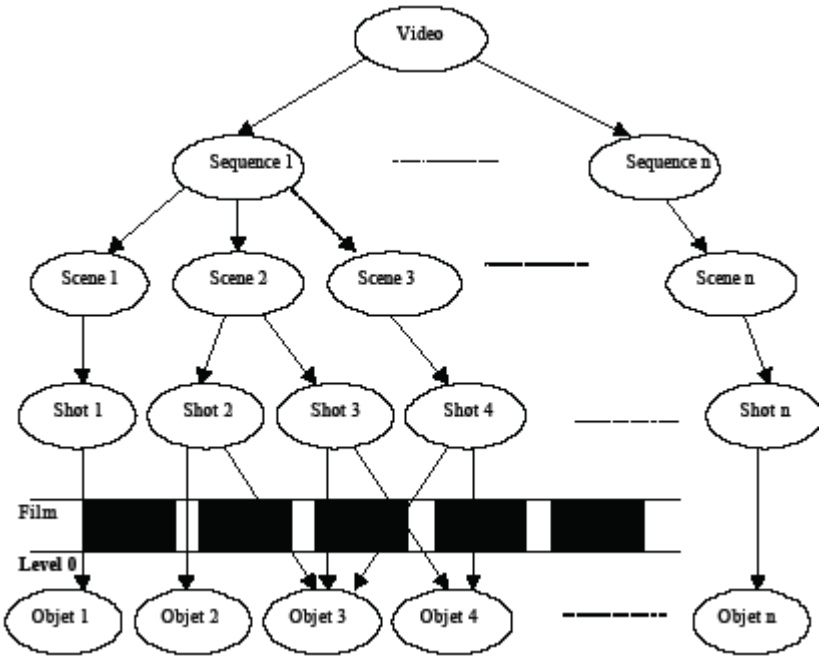


Figure 1-1: Structure cinématographique d'une vidéo

1.2 Aperçu et contributions

Dans le cadre de cette thèse, nous nous intéressons particulièrement à des problèmes d'extraction et de catégorisation des objets vidéos. Notre premier objectif est de proposer des méthodes de segmentation intelligentes et rapides pour fournir un aperçu rapide du contenu des séquences vidéo. Le but est de procéder à une segmentation d'objets d'intérêt dans des images et des séquences vidéo.

Nous proposons pour cela d'utiliser l'approche par contours actifs globaux puisqu'elle permet d'obtenir directement et rapidement l'objet d'intérêt. Nous nous intéressons particulièrement aux contours actifs basés régions, c'est-à-dire ceux qui tiennent compte des propriétés des régions à segmenter. Pour cela nous nous proposons d'élaborer des critères de segmentation qui prennent en compte les vraies distributions des caractéristiques de l'image afin d'approcher au mieux les données de l'image. Dans le cas de critères d'homogénéité comme l'entropie, ces régions seront plus ou moins homogènes, l'entropie permettant une certaine variabilité des caractéristiques considérées, notamment dans le cas de l'utilisation d'une compétition entre la région d'intérêt et le fond.

Pour améliorer la méthode de détection et de suivi de données vidéo, nous avons étendu la formulation énergétique de notre modèle des contours actifs globaux en incluant une force supplémentaire issue du calcul du flot optique. Nous parlons alors d'une segmentation active spatio-temporelle.

Nous nous sommes également intéressés à l'interprétation du comportement humain (mouvement et gestuelles) dans les séquences vidéo. Les buts poursuivis sont multiples. D'un

côté, nous procédons à une analyse du mouvement humain. Le terme “analyse” concerne ici l’extraction d’informations bas-niveau. Ces données bas-niveau peuvent être, par exemple, la silhouette de la personne, la localisation de son visage, l’extraction et classification de son expression faciale. D’un autre côté, on propose une méthode de catégorisation qui faciliterait l’interprétation du mouvement humain. Quand on parle d’interprétation du mouvement ou du comportement humain, le champ de recherches est très vaste, il peut s’agir de la reconnaissance de démarche (marche, course etc.), de postures (debout, accroupi, etc.), d’interactions avec des objets (poser, prendre etc.) ou entre des personnes (gestes, attitudes etc.). Dans notre cas, nous nous intéressons à la catégorisation des expressions faciales et des postures.

Le visage est un moyen de communication important et complexe. Il émet en permanence des signes qui renseigne sur l’état émotionnel de la personne. Pour cela, nous nous sommes intéressés à la détection du visage, et à la localisation de ses attributs. Cette étape nous a permis d’effectuer la catégorisation des expressions faciales à partir d’un modèle de diffusion par marche aléatoires sur graphe. L’objectif d’une classification automatique des visages et de reconnaître les informations caractéristiques d’une catégorie de visage afin d’identifier l’expression ou la classe d’appartenance du visage analysée. Ces expressions innées correspondent aux sept émotions suivantes: la neutralité, la joie, la tristesse, la surprise, la peur, la colère et le dégoût. C’est plus spécialement sur ces expressions que notre travail va porter.

Reconnaître un comportement humain (facial ou gestuel) est une tâche complexe à accomplir par un système de vision par ordinateur à cause de la grande variabilité entre les individus. L’analyse du comportement humain suscite l’intérêt de plusieurs communautés de recherche. Cela consiste à estimer le mouvement des personnes au cours du temps et de reconnaître leur postures, afin d’avoir une interprétation des gestes ou des comportement précis. En outre, deux questions s’imposent :

- Quels sont les indices pertinents qui doivent être extraits d’un visage, d’une posture ou d’un comportement ?
- Comment le comportement de ces indices peut être modélisé et traduit pour la catégorisation des expressions faciales, des postures et des comportements?

Nous avons essayé d’apporter notre contribution pour répondre à ces deux questions, avec une démarche qui comporte quatre étapes de traitement:

- segmentation spatio-temporelle.
- Suivi et caractérisation du mouvement spatio-temporel 2d+t.
- localisation et analyse du visage.

– Catégorisation des expressions faciales et des postures statiques ou dynamiques par analyse spectrale.

1.3 Structure du document

Après une introduction qui présente le contexte de notre travail et nos contributions liées à ce sujet de thèse. Ce mémoire de thèse aborde quatre problématiques, chaque chapitre correspondant à l'un des quatre principaux problèmes abordés qui sont les suivants :

La première partie du chapitre 2 est théorique et propose un état de l'art des contours actifs (2D/2D+T) et d'estimation de mouvement. Ensuite, nous présentons les différents contours actifs basées régions, ou basées contours. Ils sont présentés selon les trois critères fondamentaux représentation, évolution et attache aux données. Puis, nous présentons nos modèles qui sont dédiés à la segmentation d'images bidimensionnelles et volumiques. Enfin, Nous proposons des améliorations de l'algorithme en combinant à la fois une segmentation basée régions et l'estimation de mouvement par flot optique.

Dans le chapitre 3, on dresse un état de l'art des différents algorithmes de réduction de données ACP, LLE etc…Ensuite, on décrit notre algorithme qui sera utilisé pour la catégorisation des visages, des expressions faciales et des vidéos.

Dans le chapitre 4, on présente une première expérimentation concernant l'analyse et l'interprétation des visages. Nous proposons d'abord un module d'extraction des composantes faciales, dont la qualité des résultats influent directement sur la catégorisation des expressions faciales.

Dans le chapitre 5, on présente une deuxième série d'expérimentations sur l'interprétation des comportements humains. Nous présentons les caractéristiques spatio-temporels utilisés ainsi que les résultats obtenus.

Le chapitre 6 résume l'apport essentiel de cette thèse. Nous tirons aussi le bilan de nos approches ainsi que les perspectives de recherche suggérées par ce travail.

CHAPITRE 2 Segmentation d'objets vidéo par contour actif

2 Segmentation d'objets vidéo par contour actif

2.1 Introduction

La segmentation est une étape primordiale en traitement d'images et de vidéo. La segmentation d'images consiste à déterminer les régions sémantiquement importantes, c-à-d les objets, en calculant les régions «homogènes» et les contours de ces régions. La segmentation d'une image en régions consiste à rassembler les pixels ayant des propriétés communes, comme une couleur ou une texture similaire, ou un mouvement cohérent. Tandis que la segmentation d'une image en objets consiste à rassembler les régions qui ont un sens sémantique commun, même si leurs propriétés sont différentes. L'importance de la segmentation croît avec celle de l'image dans notre société. En imagerie médicale, la segmentation peut aider le médecin dans son diagnostic. En compression vidéo, elle permet de traiter différemment une zone d'intérêt, qui bénéficiera d'une plus grande précision, du reste de l'image qui pourra être plus fortement compressé. En indexation, la segmentation sert à extraire un objet que l'on souhaiterait retrouver dans d'autres images. En post-production cinématographique, un personnage segmenté peut être replacé dans un autre décor, ce qui est le cas notamment pour les présentateurs de la météo. En vidéo-surveillance, la détection d'objets en mouvement peut révéler la présence d'intrus, ou évaluer la fluidité d'un trafic routier. Les applications sont nombreuses et la liste des exemples cités est loin d'être exhaustive.

Dans ce chapitre, nous passons en revue divers travaux liés au traitement de la segmentation active (contours actifs) et de l'estimation du mouvement. Les contours actifs sont des modèles de classification permettant d'extraire un ou plusieurs objets d'intérêt d'une séquence d'images. Cette segmentation est itérative, elle met en jeu un ensemble de courbes et de régions dans une phase dynamique qui sera guidée par des forces internes et externes. Ces forces peuvent être formalisées mathématiquement par une équation d'évolution qui exprime la vitesse du contour actif. Dans la littérature, il existe plusieurs façons de formuler l'équation d'évolution soit par des approches basées frontières, soit par des approches basées régions. Nous présentons ensuite différentes méthodes d'estimation du mouvement, et de détection d'objets en mouvement. Nous abordons d'abord les problèmes liés à la détermination du flot optique, ensuite les quelques modèles de segmentation à partir de la connaissance du flot optique qui ont été proposés.

2.2 Contours actifs

À ce jour, il existe de nombreuses méthodes de segmentation, certaines permettent de définir des régions à partir des gradients de l'image. D'autres approches consistent à définir directement les régions que l'on souhaite segmenter, à partir de caractéristiques communes.

Une dernière approche basée sur les contours actifs, consiste à définir un contour fermé et à le faire évoluer vers l'objet d'intérêt. Cette approche est intéressante car la recherche de frontières d'objets peut se faire en intégrant des contraintes d'intensité et des contraintes géométriques de l'objet. Cela nécessite d'avoir une idée de l'objet ou de la région que l'on souhaite segmenter. Pour définir cet objet, il faut choisir un critère qui détermine si un pixel fait partie de l'objet ou non. Ce critère doit contenir la description des propriétés de l'objet. Le contour actif peut par exemple évoluer vers des zones de fort gradient, tout en essayant d'entourer une région la plus homogène possible au niveau de la couleur. De nombreuses propriétés peuvent être prises en compte dans le critère, et le contour actif évolue en effectuant un compromis entre elles. En cas d'objets multiples, il peut également se séparer; ou au contraire, deux contours peuvent fusionner.

Un contour actif est une courbe qui évolue d'une forme initiale vers les frontières d'un objet d'intérêt, sous l'action d'une force (voir figure 2.1).

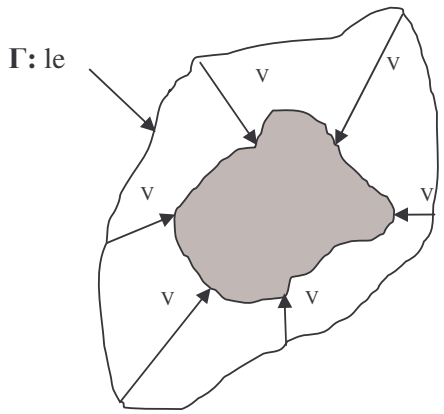


Figure 2.1: Principe des contours actifs : l'évolution de la courbe Γ par le vecteur V s'effectue de façon à ce qu'elle épouse l'objet d'intérêt.

On considère une courbe Γ représentant le contour actif donné par des points p ∈ [a, b] et d'un paramètre d'évolution τ ∈ [0, T] telle que:

$$\Gamma : [a,b]\times[0,T] \rightarrow R^2$$
$$(p,\tau) \rightarrow \Gamma(p,\tau) = X(p,\tau) = \begin{pmatrix} x(p,\tau) \\ y(p,\tau) \end{pmatrix} \qquad (2.1)$$

L'évolution du contour est régie par une équation de forme générale:

$$\frac{\partial \Gamma(p,\tau)}{\partial(p,0)} = V(p,\tau) \qquad (2.2)$$
$$\Gamma(p,0) = \Gamma_0(p)$$

Où $\Gamma_0(p)$ est le contour initial et V une force qui peut dépendre des caractéristiques géométriques ou photométriques de l'image. On décompose ce vecteur V selon ses deux composantes normale $V_N N$ et tangentielle $V_T T$ (cf. figure 2.2).

$$V = V_N N + V_T T \qquad (2.3)$$

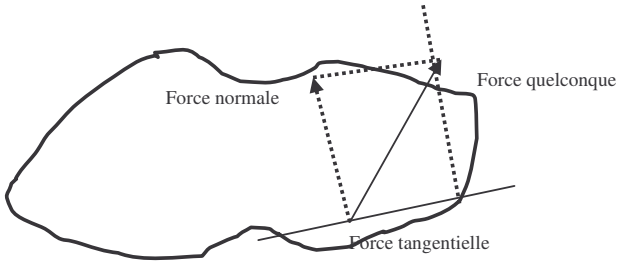


Figure 2.2: Contour actif 1D évolue sous l'action d'une force décomposable d'une composante tangentielle et normale

Dans [Sapiro01], Sapiro et al ont montré que seule la composante normale a une influence sur la déformation des contours actifs.

Par souci de simplicité, seule la composante normale est prise en compte, et l'équation d'évolution d'un contour actif s'écrit généralement:

$$\frac{\partial \Gamma(p,\tau)}{\partial \tau} = V_N(p,\tau) N(p,\tau) \qquad (2.4)$$
$$\Gamma(p,0) = \Gamma_0(p)$$

On parle d'une approche variationnelle, quand l'équation d'évolution du contour actif se déduit de la minimisation d'une énergie modélisant l'objet d'intérêt. De nombreuses techniques de segmentation ont été proposées. Parmi lesquelles, on distingue deux familles principales: Les méthodes orientées contours. et les méthodes orientées régions. La première famille met en œuvre des caractéristiques globales du contour, comme sa longueur ou sa rigidité, elle n'utilise que des propriétés locales de l'image, comme son intensité ou son gradient en un point. Tandis que la deuxième s'attache à caractériser les régions délimitées par

le contour. Nous présenterons brièvement dans les sections si dessous quelques-unes de ces méthodes.

2.2.1 Snake

Les premiers modèles variationnels ont été introduits par Kass, Witkin et Terzopoulos en 1988 [Kass88]. Le contour actif (Snake) proposé est une courbe paramétrée $\Gamma:[a,b]\to\Re^2$. L'équation d'évolution J de ce modèle est obtenue en minimisant la fonctionnelle suivante:

$$J(\Gamma)=\underbrace{\alpha\int_a^b\left|\frac{\partial\Gamma(s)}{\partial s}\right|^2 ds+\beta\int_a^b\left|\frac{\partial^2\Gamma(s)}{\partial s^2}\right|^2 ds}_{(1)}-\underbrace{\lambda\int_a^b\left|\nabla I(\Gamma(s))\right|^2 ds}_{(2)} \tag{2.5}$$

Où s paramètre le contour $\Gamma:[a,b]\to\Re^2$, et α, β , λ sont des constantes positives. ∇I désigne le gradient de l'image. Les deux premiers termes (1) représentent les forces internes qui imposent des contraintes de régularisation du contour qui déterminent son élasticité et sa rigidité. Le dernier terme est un terme d'attache aux données. Il attire le contour vers les zones de forts gradients de l'image.

Cette approche est toutefois limitée par plusieurs inconvénients. Le critère n'est pas intrinsèque, c'est-à-dire qu'il dépend de la paramétrisation du contour. De plus, le contour initial doit être proche de l'objet pour que l'algorithme de minimisation converge.

Cette première approche de contours actifs a été très utilisée, néanmoins ce modèle comporte quelques inconvénients. Tout d'abord, la fonctionnelle $J(\Gamma(s))$ dépend du paramétrage de s du contour actif, ce qui la rend non intrinsèque. D'autre part, la courbe ne peut pas changer de topologie et détecter plusieurs objets, puisque le terme de régularisation contraint la courbe à détecter la forme d'un seul objet. Enfin, ce modèle dépend aussi de l'initialisation du contour qui doit être très proche de la région d'intérêt à segmenter dans l'image.

Pour résoudre certaine de ces inconvénients, plusieurs méthodes ont été proposées tels que les méthodes Dual-Snake [Gunn97] qui utilisent deux contours, le premier est placé à l'intérieur de l'objet à localiser et le deuxième est mis à l'extérieur de la région d'intérêt. La méthode T-Snake [McInerney95] a été développée aussi, dans l'objectif de contourner le problème des changements topologique.

2.2.2 Contours géodésiques

Les contours actifs géodésiques proposés par Caselles et al. [Caselles95] [Caselles97], ont été introduits comme une autre alternative des Snakes. La fonctionnelle est modifiée par la

suppression du terme $\frac{\partial^4 \Gamma(s)}{\partial s^4}$ avec $\beta = 0$, jugé trop contraignant à cause des instabilités numériques qu'il provoque. Ils ont montré que ce modèle revient à chercher une courbe géodésique dans un espace de Riemann dont la métrique est induite par le contenu de l'image. La fonctionnelle d'énergie est réécrite de la façon suivante:

$$J(\Gamma) = \alpha \int_a^b \left| \frac{\partial \Gamma(s)}{\partial s} \right|^2 ds + \lambda \int_a^b g^2(|\nabla I(\Gamma(s))|) ds \tag{2.6}$$

On remarque l'introduction d'une fonction de détection g. La fonction est une fonction strictement décroissante qui tend vers 0 en +∞. La fonction $g(x) = \frac{1}{1+\rho x^m}$, m=1 ou 2 est fréquemment choisie et ρ est une constante positive.

Caselles et al. proposent de minimiser l'énergie suivante:

$$J(\Gamma) = \int_0^{L(\Gamma)} g(|\nabla I(\Gamma(s))|) ds \tag{2.7}$$

Le problème s'interprète alors géométriquement comme la minimisation de la longueur du contour $L(\Gamma)$ dans une métrique prenant en compte les caractéristiques de l'image. La nouvelle fonctionnelle est intrinsèque, c'est-à-dire qu'elle ne dépend pas de la paramétrisation.

L'équation d'évolution déduite du critère s'écrit comme suit:

$$\frac{\partial \Gamma}{\partial \tau} = (g(|\nabla I(\Gamma)|)\kappa - \nabla g(|\nabla I(\Gamma)|).N)N \tag{2.8}$$

où κ la courbure de la courbe Γ et N le vecteur unitaire normal intérieur au contour.

Sur les zones homogènes, où le gradient de l'image est faible, le premier terme de l'équation d'évolution est prépondérant et le contour évolue suivant:

$$\frac{\partial \Gamma}{\partial \tau} = \kappa N \tag{2.9}$$

Comme κ est une constante positive, et le vecteur N est dirigé vers l'intérieur du contour, le contour actif aura donc tendance à rétrécir.

Sur les contours, où le gradient de l'image est de forte amplitude, le deuxième terme de l'équation d'évolution est prépondérant et le contour évolue suivant:

$$\frac{\partial \Gamma}{\partial \tau} = -\nabla g(|\nabla I(\Gamma)|).NN \tag{2.10}$$

Le contour actif s'approche de la frontière de l'objet, qu'il soit à l'intérieur ou à l'extérieur de l'objet.

2.2.3 Approches basées régions

Les modèles de contours actifs basés régions sont de plus en plus utilisés. L'idée de base consiste à introduire des informations géométriques ou topologique tels que les statistiques à l'intérieur ou à l'extérieur de la courbe. On présente dans cette section une formulation générique des contours actifs basées région.

Il s'agit d'un modèle généralisant le modèle de Mumford et Shah, ainsi que d'autres modèles [Chan01] [Jehan-Besson03] [Paragios02] [Zhu96] Fuzzy Region Competition], qui donne une signification physique à chacun des termes de la fonctionnelle intégrée sur les régions: ce sont les descripteurs des régions.

Une formulation encore plus générique peut être adoptée dans le cadre des contours actifs multiples et de la classification d'image. Mais, pour la clarté, on suppose qu'on souhaite partitionner une image en deux régions Ω_{int} et Ω_{ext}. Ω_{int} est la région contenant les objets à segmenter et Ω_{ext} la région du fond. La fonctionnelle d'énergie à minimiser doit s'écrire comme une intégrale sur la région et devient alors:

$$J(\Gamma, \alpha_1, \alpha_2) = \mu \int_{\Gamma} g(\Gamma(s))ds + \lambda \left[\int_{\Omega_{int}} r_1^{\alpha 1}(x)dx + \int_{\Omega_{ext}} r_2^{\alpha 2}(x)dx \right] \tag{2.11}$$

où $\Omega \subset \Re^n$ est le domaine de l'image.

Les fonctions $r_i^{\alpha} : \Omega \to \Re$ désignent les descripteurs géométriques de chaque région qui dépendent des paramètres de régions $\alpha = (\alpha_1, \alpha_2)$, tels que des scalaires [Chan01], des vecteurs [Paragios02] [Zhu96] ou des fonctions [Tsai01] [Chan02]. Ainsi, $r_1^{\alpha 1}$ désigne le descripteur des objets à segmenter et $r_2^{\alpha 2}$ le descripteur de la région du fond.

- Quand $\lambda = 0$, on retrouve le contour actif géodésique.
- Pour $r_i^{\alpha i} = \log P_i(I \setminus \alpha_i)$, on retrouve le modèle de Paragios et Deriche [Paragios02].

– Pour g=Id et $r_i^{\alpha i}(x) = (I_{(x)} - s_i)^2 + |\nabla s_i|$, on retrouve le modèle de Mumford-Shah. [Tsai01][Chan02] [Mumford89], où s1 et s2 sont deux fonctions approximant l'image à l'intérieur et à l'extérieur des régions.

– Pour $r_i^{\alpha i}(x) = (I_{(x)} - c_i)^2$, on retrouve le modèle de Chan et Vese [Chan01] qui consiste à minimiser la fonctionnelle suivante:

$$J(\Gamma) = \mu \int_{\Gamma} g(\Gamma(s))ds + \lambda \left[\int_{\Omega_{int}} (I_{(x)} - c_1)^2 dx + \int_{\Omega_{ext}} (I_{(x)} - c_2)^2 dx \right] \quad (2.12)$$

En particulier quand g=Id ($g(x) = 1$)

$$J(\Gamma) = \mu \int_{\Gamma} ds + \lambda \left[\int_{\Omega_{int}} (I_{(x)} - c_1)^2 dx + \int_{\Omega_{ext}} (I_{(x)} - c_2)^2 dx \right] \quad (2.13)$$

où c_1, c_2 représentant les moyennes de l'image I à l'intérieur et à l'extérieur de la courbe.

La solution de (2.15) se fait de deux manières:

1) Soit en fixant Γ, donc c_1, c_2 sont calculés avec les moyennes de l'image des Ω_{int} et Ω_{ext},

2) Soit en fixant c_1, c_2, en minimisant $J(\Gamma)$, on retrouve l'équation d'évolution :
$\frac{\partial \Gamma}{\partial \tau} = \{\kappa + \lambda[(I - c_1)^2 - (I - c_2)^2]\}N$ avec κ est la courbure

2.2.4 Implémentation par ensembles de niveaux

La méthode des level set est un cadre de travail analytique travaillant sur l'évolution géométrique d'objets. Au lieu d'employer une représentation Lagrangienne classique pour décrire les géométries, la méthode des level set les décrit à travers une fonction scalaire U définie sur une grille fixe. Elle considère une surface à n dimensions, le contour de l'objet est décrit par le niveau zéro d'une surface tridimensionnelle. 0n parle alors, des ensembles de niveaux (level sets) proposés par Osher et Sethian [Osher88].

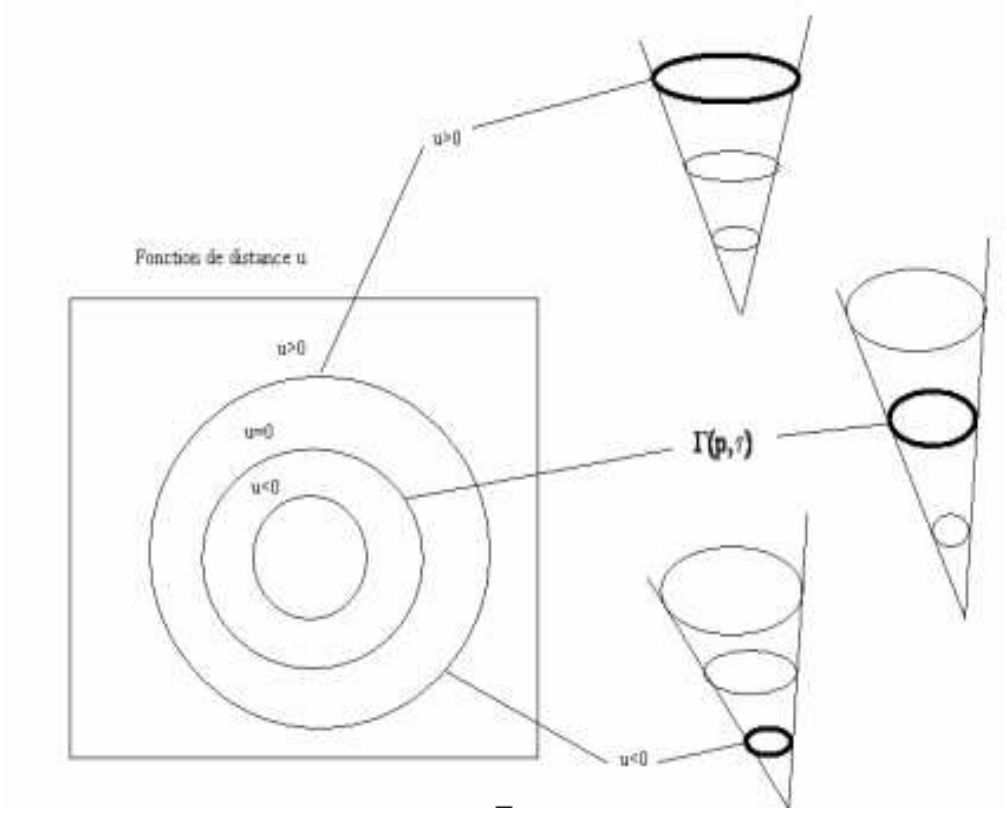


Figure 2.3: Exemple de changement de topologie permis par les ensembles de niveaux

L'idée est donc est de faire correspondre à la courbe Γ une représentation implicite par une courbe de niveaux d'une fonction U définie sur tout le support de l'image.

Considérons que la représentation implicite d'une hyper surface $\Gamma(t)$ dans R^n est une fonction de niveaux $U(x,t)$.

Le niveau de la fonction U_0 coïncide avec Γ_0. Il est choisi comme la fonction distance à l'hyper surface. Il est positive à l'extérieur de Γ_0 et négative à l'intérieur:

$$U(.,t=0)=\begin{cases}\text{dist}(x,\Gamma_0) & \text{si } x\in \text{int}(\Gamma_0)\\ 0 & \text{si } x\in\Gamma_0\\ -\text{dist}(x,\Gamma_0) & \text{si } x\in \text{ext}(\Gamma_0)\end{cases} \quad \text{où } \text{dist}(p,\Gamma)=\min_{p\in\Gamma}|p-p_\Gamma|$$

Le modèle de Chan et Vese [Chan01] consiste à partager le domaine Ω en intérieur et extérieur de Γ. On cherche le contour représenté par une courbe C qui minimise une fonctionnelle de la forme:

$$J(\Gamma, c_1, c_2) = \mu \int_{\Gamma} ds + \lambda \left[\int_{\Omega_{int}} (I_{(x)} - c_1)^2 dx + \int_{\Omega_{ext}} (I_{(x)} - c_2)^2 dx \right] \quad (2.14)$$

où c_1, c_2 sont des constantes représentant les valeurs moyennes de l'image I dans les régions intérieure et extérieure à la courbe C (c_1 = moyenne(I) sur $U \geq 0$) et (c_2 = moyenne(I) sur $U < 0$). Le paramètre $\lambda > 0$ est une constante pondérant le terme de régularisation.

Minimisant le deuxième terme d'approximation de la fonctionnelle, le modèle permet la recherche de la meilleure partition en deux régions de l'image prenant les valeurs c_1 et c_2, avec un seul contour Γ séparant les deux régions.

Formulé en utilisant la méthode des ensembles de niveaux, ce problème revient à minimiser la fonctionnelle suivante:

$$J(U, c_1, c_2) = \mu \int_{U=0} ds + \lambda \left[\int_{U>0} (I_{(x)} - c_1)^2 dx + \int_{U<0} (I_{(x)} - c_2)^2 dx \right] \quad (2.15)$$

ou encore:

$$J(U, c_1, c_2) = \mu \int_{\Omega} \delta(U) |\nabla U| + \lambda . \left(\int_{\Omega} |I_{(x)} - c_1|^2 \, Heav(U) dx + \int_{\Omega} |I_{(x)} - c_2|^2 (1 - Heav(U)) dx \right) \quad (2.16)$$

On définit alors la fonction Heaviside Heav:

$$Heav(x) = 1 \text{ si } x \geq 0,$$

$$Heav(x) = 0 \text{ si } x < 0,$$

La résolution est faite en deux étapes. La fonctionnelle $J(U, c_1, c_2)$ est d'abord minimisée relativement aux constantes c_1 et c_2, pour une fonction de niveaux U supposée fixe. Cette minimisation permet d'estimer c_1 et c_2. Ensuite la fonctionnelle est minimisée relativement à U, pour les valeurs de c_1 et c_2 fixées. Les valeurs de c_1 et c_2 sont données par les formules:

$$c_1(U) = \frac{\int_{\Omega} I_{(x)} . Heav(U_{(x)}) dx}{\int_{\Omega} Heav(U_{(x)}) dx} \quad \text{et} \quad c_2(U) = \frac{\int_{\Omega} I_{(x)} . (1 - Heav(U_{(x)})) dx}{\int_{\Omega} Heav(U_{(x)}) dx}$$

Les équations d'Euler-Lagrange et la méthode de descente du gradient permettent d'aboutir à un modèle géométrique défini par l'équation d'évolution suivante:

$$\frac{\partial U}{\partial t} = \left(\mu \nabla . \frac{\nabla U}{|\nabla U|} - \lambda . (I - c_1)^2 + \lambda . (I - c_2)^2 \right) \tag{2.17}$$

Avec c_1 et c_2 deux valeurs correspondant à la moyenne de l'image sur les régions $U \geq 0$ et $U \leq 0$.

On exprime la longueur e la surface de la manière suivante:

$$\text{Longueur}\{ U = 0 \} = \int_{\Omega} |\nabla \text{Heav}(U(x))| dx = \int_{\Omega} \delta(U(x)) |\nabla U(x)| dx ,$$

$$\text{Aire}\{ U \geq 0 \} = \int_{\Omega} \text{Heav}(U(x)) dx$$

La fonction Dirac est définie par: $\delta(x) = \frac{d}{dx} \text{Heav}(x)$

Dans ce modèle, les courbes de niveaux évoluent avec une force dépendant de la courbure et de la moyenne de l'image à l'intérieur et à l'extérieur du contour actif évoluant. De ce fait, ce modèle peut être appliqué à l'extraction d'objets, même s'ils ne sont pas caractérisés par un contraste important sur leurs frontières, à condition que ces objets soient caractérisés par leur distribution moyenne par rapport aux objets environnants.

Comme on peut le constater, la fonction d'évolution définie peut changer de signe au cours de son évolution. Ce modèle est donc relativement insensible au sens de l'évolution.

Malgré les avantages de ce modèle, il reste très coûteux en temps de calcul, puisqu'il faut réinitialiser la fonction distance sur toute l'image, à chaque itération, pour maintenir la représentation de Γ par une fonction de niveau U régulière.

2.3 Modèles de contours actifs globaux

Tous les modèles présentés et d'autres qui lui sont similaires de part leur construction sont fortement non convexes, et les méthodes de descente de gradient pour la minimisation d'énergie qui aboutissent aux équations d'évolution ne garantissent pas la convergence vers un minimum global. Elles sont donc très sensibles à l'initialisation.

Pour pallier ces problèmes, récemment de nouvelles méthodes de contours actifs ont été proposées [Jianhong06].

Il s'agit donc de convexifier des énergies de la forme, ensuite trouver une solution globale qui est identique à la solution au problème. Rappelons notre problème d'optimisation:

$$\min_{\Gamma}\left\{J(\Gamma,\alpha_1,\alpha_2)=\mu\int_{\Gamma}g(\Gamma(s))ds+\lambda\left[\int_{\Omega_{int}}r_1^{\alpha 1}(x)dx+\int_{\Omega_{ext}}r_2^{\alpha 2}(x)dx\right]\right\} \quad (2.18)$$

Cette optimisation peut s'écrire comme la minimisation de la fonction:

$$\min_{\chi}\left\{J(\chi,\alpha_1,\alpha_2)=\mu\int_{\Gamma}g(x)|\nabla\chi(x)|dx+\lambda\left[\int_{\Omega}\chi(x)r_1^{\alpha 1}(x)dx+\int_{\Omega}(1-\chi(x))r_2^{\alpha 2}(x)dx\right]\right\} \quad (2.19)$$

Avec $\chi(x)=\chi_{int}(x)=\begin{cases}1 & \text{si } x\in\Omega_{int}\\ 0 & \text{sinon}\end{cases}$

Sous cette forme, nous proposons d'étendre (équation 2.19) en un problème qui soit convexe en remplaçant χ par une fonction u d'appartenance à un ensemble convexe $\{0,1\}$.

En posant $u=\chi$, la fonctionnelle devient:

$$\min_{u\in\{0,1\}}\left\{J(u,\alpha_1,\alpha_2)=\mu\int_{\Gamma}g(x)|\nabla u(x)|dx+\lambda\left[\int_{\Omega}u(x)r_1^{\alpha 1}(x)dx+\int_{\Omega}(1-u(x))r_2^{\alpha 2}(x)dx\right]\right\}$$

Pour résoudre ce problème, la convexification consiste à prendre le problème relaxé qui s'écrit sous la forme suivante:

$$\min_{u\in[0,1]}\left\{J(u,\alpha_1,\alpha_2)=\mu\int_{\Gamma}g(x)|\nabla u(x)|dx+\lambda\left[\int_{\Omega}u(x)r_1^{\alpha 1}(x)dx+\int_{\Omega}(1-u(x))r_2^{\alpha 2}(x)dx\right]\right\} \quad (2.20)$$

$$=\min_{u\in[0,1]}\left\{J(u,\alpha)=\mu\int_{\Gamma}g(x)|\nabla u|(x)dx+\lambda\int_{\Omega}r^{\alpha}(x)u(x)dx\right\}$$

$$=\min_{u\in[0,1]}\{J(u,\alpha)=\mu R(u)+\lambda F(u)\}$$

Avec:

$R(u)=\int_{\Gamma}g(x)|\nabla u|(x)dx$ comme terme de régularisation

$$F(u)=\int_{\Omega}r^{\alpha}(x)u(x)dx=\int_{\Omega}u(x)r_1^{\alpha 1}(x)dx+\int_{\Omega}(1-u(x))r_2^{\alpha 2}(x)dx$$

Cette fonction est convexe et possède donc un minimum global. Soit u^* la solution de (équation 2.18). Il a été montré [Nikolova04] que:

$u^{*t}(x) = \begin{cases} 1 & \text{si } u^*(x) > t \\ 0 & \text{sinon} \end{cases}$ est la solution de la fonction dans l'équation (2.20) avec $t \in [0,1]$.

La minimisation de l'équation (2.20) en fixant α est équivalent à la minimisation de l'équation suivante :

$$\mu \int_{\Omega} g(x)|\nabla u(x)|dx + \lambda \int_{\Omega} u(x)r(x)dx \tag{2.21}$$

L'équation (2.21) est sous la contrainte de u et r, $0 \le u \le 1$ où $r = r_1^{\alpha 1} + r_2^{\alpha 2}$

Ainsi, ce problème sous contrainte a le même ensemble de minimiseurs que le problème non contraint de l'équation suivante:

$$\min_{u \in [0,1]} \left\{ \tilde{J}(u) = \int_{\Omega} \mu g(x)|\nabla u(x)| + \lambda u(x)r(x) + \alpha\theta(u(x))dx \right\} \tag{2.22}$$

Avec $\alpha > \frac{1}{2}|r|_{\infty}$ et le terme de pénalité convexe $\theta(u) = \max(0, \min(1, u))$ pour contraindre la solution $0 \le u(x) \le 1$.

La minimisation de l'équation (2.20) étant un problème d'optimisation contrainte , pour le résoudre on la transforme en problème d'optimisation non contrainte (équation 2.22) qui peut être résolu numériquement par des méthodes de descente du gradient basée sur l'équation d'Euler-Lagrange ou par la méthode de projection de Chambolle.

L'énergie $J(\Gamma, \alpha_1, \alpha_2)$ peut être globalement optimisé, et sa minimisation permet d'extraire l'objet et le fond. Dans cette partie nous allons décrire une technique d'optimisation par reformulation de l'équation (2.18) comme un problème d'optimisation convexe.

Une région est représentée implicitement par des fonctions caractéristiques $u : V \to \{0,1\}$, c'est-à-dire $u = 1_{\Omega ext}$ et $1 - u = 1_{\Omega int}$ avec Ω_{int} est les régions contenant les objets à segmenter et Ω_{ext} les régions appartenant au fond.

$$V = \Omega_{ext} \bigcup \Omega_{int}$$

En résumé, l'optimisation présentée dans la section 2.3 peut être décomposée en deux parties:

– Trouver un minimiseur u de l'équation 2.22

– Seuillage du résultat: $\Omega_{ext} = \{x \in V \mid u(x) < \mu \quad \text{avec} \quad \mu \in (0,1)\}$

Une condition nécessaire pour un minimum de l'équation 2.22 est indiqué par l'équation d'Euler-Lagrange:

$$\frac{\partial u}{\partial t} = 0$$

$$0 = -\mu g.\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) - \left\langle \nabla g, \frac{\nabla u}{|\nabla u|} \right\rangle + \lambda f(x) + \alpha\theta_{\varepsilon}^{'}(u)$$

$$0 = \lambda f(x) - \mu \mathrm{div}\left(g\frac{\nabla u}{|\nabla u|}\right) + \alpha\theta_{\varepsilon}^{'}(u) \; **$$

où $\langle -,- \rangle$ représente le produit scalaire entre deux variables et θ_{ε} est une version régularisée de la dérivée de θ.

On peut utiliser une forme à priori comme un terme de fidélité additionnelle, qui rend la segmentation plus robuste aux occlusions, et aux données hétérogènes. De façon générale, la contrainte de forme est introduite par une métrique permettant de comparer le contour actif à l'instant t avec la forme à priori. Dans le cadre des approches variationnelles, cette métrique est utilisée pour la formulation d'une énergie de contrainte de forme F_{forme} qui est alors ajoutée à celle relative aux données: $F = f_{image} + \lambda F_{forme}$

La forme à priori est définie comme la distance entre le vecteur des caractéristiques de la région, tels que les moments de la région qui évolue et la forme de référence: $d(\Omega, \Omega_{ref}) = \|(m(\Omega) - m(\Omega_{ref})\|$

2.3.1 Minimisation rapide par SOR

La discrétisation de l'équation d'Euler-Lagrange (**) nous ramène à un système d'équation non linéaire qui peut être résolu par la méthode de descente de gradient. Cependant, cette méthode converge très lentement. C'est pour cela, que nous avons utilisé la méthodes d'itération à point fixe qui transforme le système non linéaire en une série de systèmes linéaires. Ces derniers peuvent être résolus efficacement avec des solutionneurs itératifs, tels que la méthode de Gauss-Seidel, successive over-relaxation (SOR), ou des méthodes multi grilles.

En négligeant $\alpha\theta_\varepsilon^{'}(u)$, la seule source de non linéarité dans (**) est la diffusion $g:\frac{1}{|\nabla u|}$.

On commence par l'initialisation de $u^{\circ} = 0.5$, on peut calculer g et le garder constant.

Pour g constant , (**) devient un système d'équation linéaire qui peut être résolut par la méthode SOR. Ce qui conduit à calculer itérativement u du voxels i.

$$u_i^{l,k+1} = (1-\omega)u_i^{l,k} + \omega\frac{\nu\sum_{j\in N(i),j<i}\rho_j g_{i\sim j}^{l}u_j^{l,k+1} + \nu\sum_{j\in N(i),j>i}\rho_j g_{i\sim j}^{l}u_j^{l,k} - b_i}{\nu\sum_{j\in N(i)}\rho_j g_{i\sim j}^{l}} \qquad (2.23)$$

Avec: N(i) voisinage de i et $g_{i\sim j}^{l}$ est la diffusion entre le voxels i et ses voisin j. Le vecteur b_i contient la partie constante de (**) qui ne dépend pas de u c'est-à-dire le terme de fidélité de $b_i = r^{\alpha}$.

Le paramètre over-relaxation ω doit être choisi dans l'intervalle [0,2] pour que la méthode converge.

La valeur optimale dépend du système linéaire à résoudre.

Empiriquement, nous obtenons une convergence rapide autour de $\omega = 1{,}85$

Dans la section suivante nous allons montrer quelques choix du terme de fidélité $b_i = (r_1^i - r_2^i)$ et $b_i = (\log P_1^i - \log P_2^i)$

2.3.2 Extraction d'objets 2d/2d+t par contours actifs globaux

Dans cette partie nous présentons notre approche de segmentation d'objets 2d/3D et d'objets en mouvement dans des séquences vidéo. Tout d'abord dans la section 2.4.1, nous présentons une généralisation de la méthode de segmentation d'objets par contour actif (2d à 2d+t). Ensuite, nous présentons nos approches de segmentation d'objets en mouvement en utilisant une approche mixte combinant les contours actifs et le flot optique. La première, segmente les objets en mouvement en utilisant un champ dense mais la précision de cette segmentation repose sur la qualité du flot optique obtenu et de ce fait, certaines portions des objets peuvent être laissés de côté. Dans la seconde, nous proposons une autre méthode de segmentation d'objets mobiles où nous cherchons à estimer conjointement le mouvement des objets et leur segmentation. Pour ce faire nous minimisons une fonctionnelle spatio-temporelle prenant en compte le mouvement et l'intensité des objets d'intérêt. Ces méthodes ont l'avantage de combiner le spatial et le temporel et permettent de segmenter correctement des objets mobiles ou en mouvement.

2.3.2.1 Formulation générale pour la segmentation

Dans cette partie, nous allons présenter notre approche de segmentation binaire rapide basée sur les contours actifs globaux. Les approches proposées sont applicables sur des images 2D et facilement extensible à toute autre dimension. Le but est d'extraire des objets vidéo d'intérêt dans une série d'images 2d ou volume 2d+t.

Dans ce paragraphe, nous allons montrer des expérimentations sur des images 2D et 3D. Rappelons l'énergie à minimiser :

$$\min_{u \in [0,1]} \left\{ J(u,\alpha) = \mu \int_{\Gamma} g(x) |\nabla u|(x) dx + \lambda \int_{\Omega} r^{\alpha}(x) u(x) dx \right\}$$

- En posant, g=1 et $r_i^{\alpha}(x) = (I_{(x)} - c_1)^2 - (I_{(x)} - c_2)^2$, $\alpha = (c_1, c_2)$, nous avons cherché à minimiser la fonctionnelle de Chan et Vese [Chan01] d'une manière efficace , rapide par les contours actifs globaux.

Le problème devient [Zinbi04]:

$$\min_{u \in [0,1]} \left\{ J(u,c_1,c_2) = \mu \int_{\Gamma} |\nabla u| + \lambda \int_{\Omega} u\left[(I-c_1)^2 - (I-c_2)^2\right] \right\} \text{ avec } (c_1,c_2) \in R^2$$

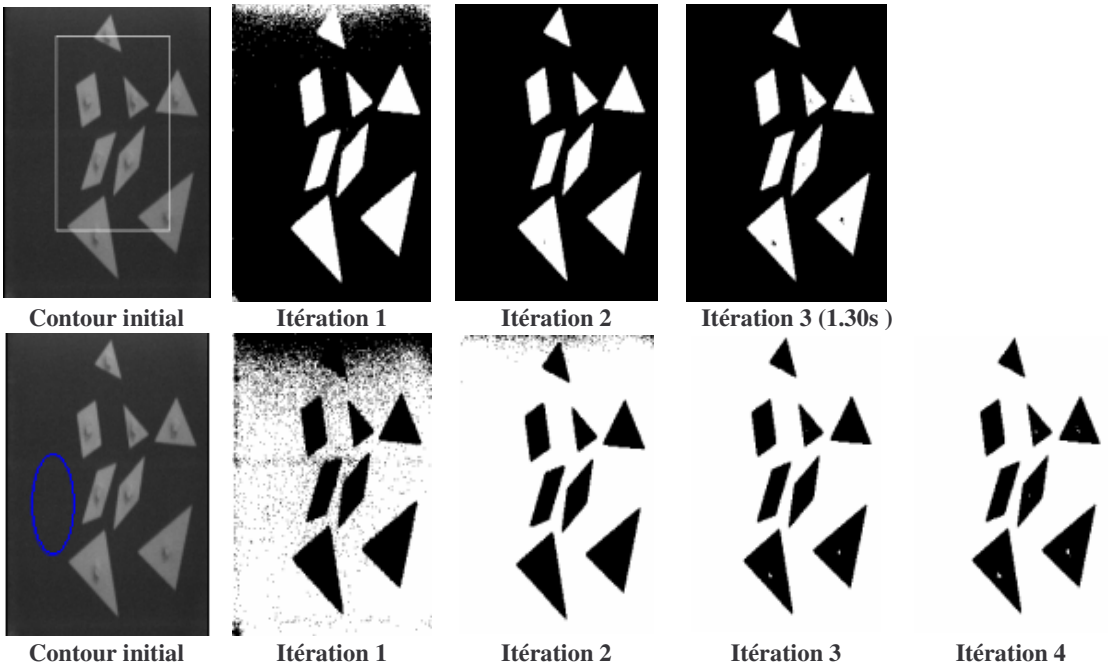


Figure 2.4: Segmentation de l'image Tangrame [256*256]

Ces valeurs sont calculées de la même manière pour le cas 3D. L'énergie possède un minimum global. Le minimum trouvé à l'itération finale correspond au coût de la segmentation optimale. Cette valeur doit être calculée à l'itération finale quelque soit le contour initial donné et le nombre d'itérations réalisées (figure 2.4). On peut remarquer que le temps d'exécution est corrélé directement avec la distance euclidienne entre la solution actuelle et la solution optimale (figure 2.5)

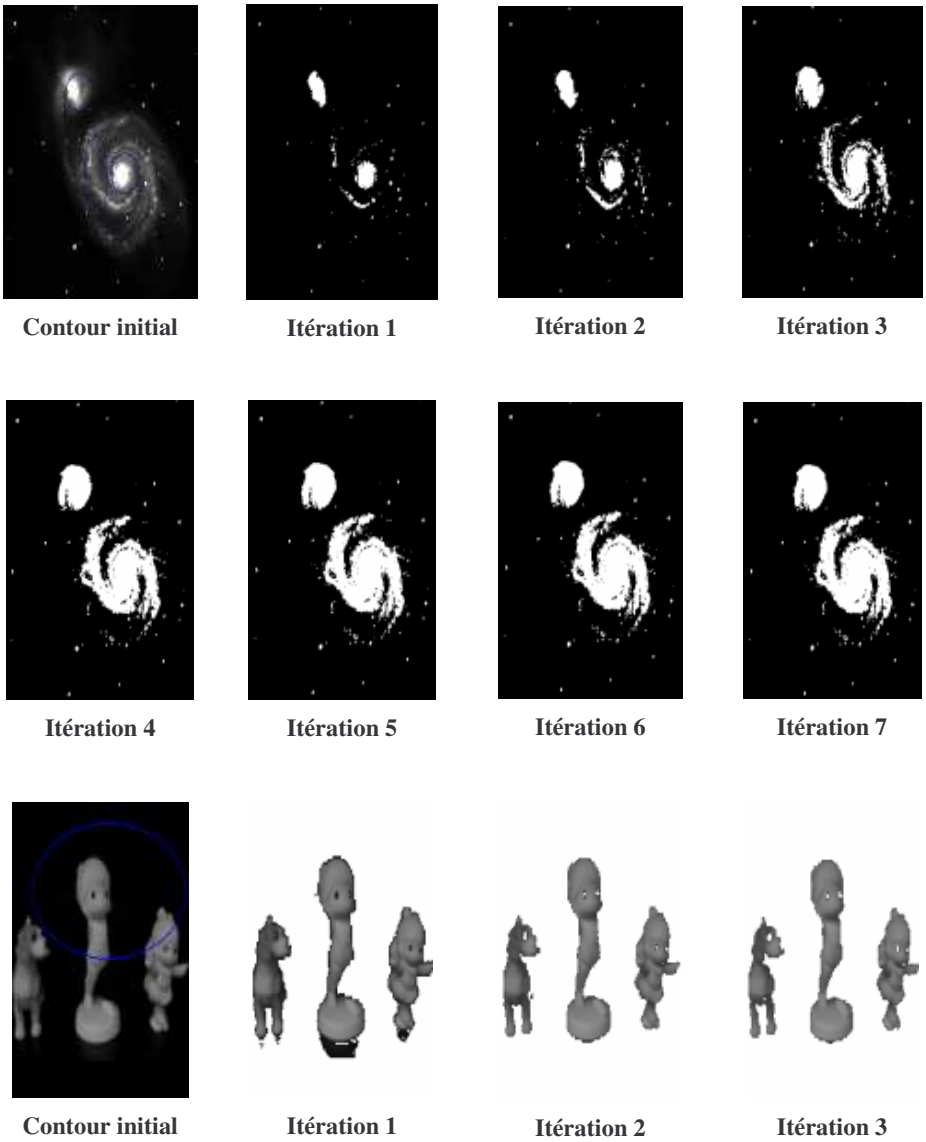


Figure 2.5: Convergence vers l'optimum globale en un certain nombre d'itérations

- En posant, $r_i^{\mu i,\sigma i}(x) = \log P_1(x) - \log P_2(x)$, $\alpha_i = (\mu_i, \sigma_i)$, nous avons voulu améliorer le modèle précédant en introduisant une fonction générale de la gaussienne qui permet de mieux tenir compte des caractéristiques divers de texture

et de couleur dans l'image, qui est la densité de la probabilité d'appartenance ou non à la région délimitée par le contour comme en [Zhu96] [Paragios02].

Soit Γ la frontière entre Ω_{int} et Ω_{ext}, on parvient alors à une segmentation active en deux classes [Zinbi05] en minimisant une fonctionnelle d'énergie globale J, qu'on définit comme suit:

$$\min_{u \in [0,1]} \left\{ J(u, c_1, c_2) = \mu \int_{\Gamma} |\nabla u| + \lambda \int_{\Omega} u \left[\log P_1 - \log P_2 \right] \right\} \tag{2.24}$$

$P_i(x) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{\frac{(I(x) - \mu_i)^2}{2\sigma_i^2}}$ la fonction de densité de probabilité de la région Ω_i

Où $I(x)$ est l'intensité du pixel/voxel x, μ_i est la moyenne de la région Ω_i et σ_i son écart type. Soit $P_{\Omega in}$ et $P_{\Omega out}$ respectivement dans les deux régions Ω_{int} et Ω_{ext}.

Les images de la figure 2.6 montrent des résultats de segmentation par l'approche basée sur la densité de probabilité.

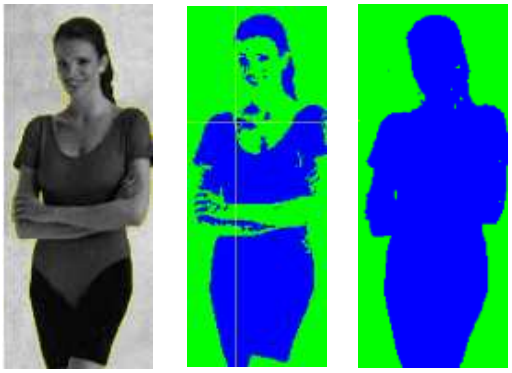


Figure 2.6: Segmentation de l'image en utilisant la moyenne, comme descripteur statistique (à gauche), et avec utilisation de la probabilité de densité (à droite).

Ce modèle est intéressant pour sa robustesse au bruit grâce notamment à la prise en compte des statistiques du bruit et d'informations a priori sur la segmentation recherchée (homogénéité, textures) (figure.2.7).

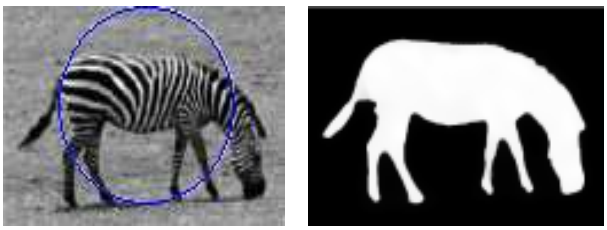


Figure 2.7: Segmentation d'une image texturée avec avec utilisation de la probabilité de densité

2.3.2.2 La sensibilité au contour initial

La minimisation classique des contours actifs par les EDP est coûteuse en temps de calcul et elle se fait en un nombre élevé d'itérations pour converger vers la solution finale et cela devient assez grand pour les contours initiaux loin des objets désirés.

Pour étudier la sensibilité de notre approche, nous utilisons plusieurs contours différents figure 2.8, en général avec quelques itérations rapides seulement (deux itérations sur l'image zèbre) figure.2.7 on peut arriver à la solution finale.

Initialisation | **Segmentation finale avec u=10** | **Segmentation finale avec u=20**

Figure 2.8: Segmentation de l'image du Cameraman [256*256] par différents contours initiaux

Contour initial	Nombre d'itération
C 1	5 itérations
C 2	4 itérations
C 3	6 itérations
C 4	3 itérations

Tableau 2-1: Sensibilité au contour initial

Le contour initial donne une première idée sur «l'objet» et le «fond», l'intérêt d'un contour proche des objets par rapport un contour éloigné est essentiellement pour accélérer la segmentation dans les applications, avec le contour initial C4 sur figure 2.8 la segmentation est faite après 3 itérations et le temps d'exécution 2 fois plus rapide que avec le contour initial C3. tableau2-1.

Image initial **Itération 1** **Itération 2**

Figure 2.9: Image zèbre, Stabilité (après deux itérations)

2.3.2.3 Application sur des images couleurs, et images 3D

Le principe reste le même pour les images couleurs, dans ce cas une image I est composée d'un vecteur (Ix, Iy, Iz) de trois élément chacun prend la valeurs sur une composante. La variance de l'image devient un vecteur de variances $\delta^2 = (\delta_x^2, \delta_y^2, \delta_z^2)$ calculés par rapport à chaque composante; La distance de similarité est donnée par:

$$d(Ip, Ic) = (d(Ip_x, Ic_x)^2 + d(Ip_y, Ic_y)^2 + d(Ip_z, Ic_z)^2)$$

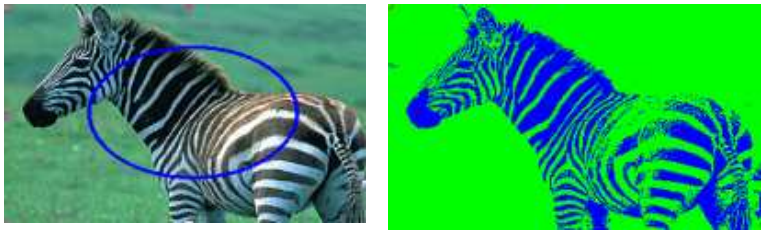


Figure 2.10: Segmentation d'image en couleur, zèbre 1.88s

Dans le cas 3D, le contour initial est donné comme un volume sur la séquence, la région trouvée à chaque itération représente une hyper surface qui divise le volume en deux régions volumiques R_{in} et R_{ext} (figure 2.11).

Figure 2.11: Application sur des séquences 3D

La non nécessité d'un facteur stoppant confère plusieurs avantages à ce modèle. Il détecte les contours avec ou sans gradient important. Il détecte automatiquement les contours intérieurs des objets. Il est peu sensible à la position des contours initiaux. Il réalise une partition binaire de l'image : les objets intéressants et le fond les entourant.

2.3.2.4 Segmentation active spatio-temporelle

La segmentation d'objets en mouvement dans des séquences vidéo est une phase importante dans l'analyse d'une scène. Elle engendre un grand nombre d'applications en télésurveillance, en imagerie médicale, en météorologie et plusieurs autres application. Néanmoins, les techniques utilisées ne sont pas toujours appropriées pour la segmentation et le suivi d'objets en mouvement. Nous pouvons séparer les méthodes de détection de mouvement en trois grandes familles. Une revue de ces méthodes peut être trouvée dans [Zhang01]. Une des méthodes les plus simples lorsque la caméra est fixe, s'effectue par le calcul de la différence entre deux images. D'autres méthodes s'appuient sur l'estimation d'une image représentant le fond statique de la scène, appelé fond de la séquence. Le fond d'une séquence est généralement une moyenne des n images de la séquence vidéo.

Jehan-Besson et al. proposent un critère de détection de mouvement engendré par la différence entre le fond estimé et l'image à traiter. Afin d'éliminer les valeurs correspondant aux objets en mouvement, les auteurs introduisent dans leur critère un M-estimateur et minimisent la différence pondérée entre le fond et chacune des images par un algorithme de minimisations alternées. Wu et Kittler [Wu93] utilisent un modèle de mouvement affine pour représenter le mouvement de la caméra et détectent les objets en mouvement par un seuillage. De son coté, Salembier [Salembier99] propose une approche morphologique pour la segmentation d'objets en mouvement. Certains ont utilisé des méthodes d'estimation du fond de la séquence vidéo (la partie statique) afin de segmenter les objets en mouvement. Le fond peut aussi être calculé par une approche statistique en modélisant l'image de différences comme une mixture de laplaciennes [Sifakis01]. En caméra mobile, on appelle ce fond une mosaïque [Gastaud02] [Irani96].

2.3.2.4.1 Estimation du Flot Optique

L'estimation du mouvement représente un des aspects essentiels pour extraire les objets visuels dans des images de séquences vidéo. L'estimation du mouvement s'avère un problème

très difficile à résoudre, suite à la variété des types de mouvement dans les séquences vidéo. La figure 2.12 illustre les différents cas d'objets visuels en mouvement qu'on peut trouver dans des séquences vidéo. Dans l'image (1) Akiyo le mouvement est très faible, l'image (2) les bras de la personne ont le même contraste que le fond, l'image (3) deux objets (girafe) qui se croisent et l'un occulte le mouvement de l'autre, dans l'image (4) de hall on a une personne qui se déplace vers la caméra.

Figure 2.12: Quelques exemples des différents types de mouvement dans différentes images.

Toutes les méthodes d'estimation du flot optique intègrent les informations sur un voisinage spatial et spatio-temporel pour parvenir à calculer l'estimation locale du mouvement. Ces techniques peuvent être classées en trois catégories: méthodes de mise en correspondance (MC), méthodes fréquentielles (MF) et méthodes différentielles (MD).

L'hypothèse d'invariance en temps de l'intensité lumineuse le long de la trajectoire du mouvement est exprimée par l'équation des différences entre les images déplacées, c'est-à-dire entre les images à l'instants t et t+1.

$$DFD = I(x + d_x, y + d_y, t + \Delta t) - I(x, y, t)$$

En récrivant cette équation, on obtient:

$$\frac{dI(x,y,t)}{dt} = 0 \tag{2.25}$$

où x et y varient dans le temps, le long de la trajectoire du mouvement. Sous l'hypothèse de différentiabilité spatio-temporelle de l'intensité lumineuse et en utilisant les règles de différentiation, on obtient:

$$I(x + dx, y + dy, t + \Delta t) = I(x, y, t)$$
$$+ \frac{\partial I(x,y,t)}{\partial x} dx + \frac{\partial I(x,y,t)}{\partial y} dy + \frac{\partial I(x,y,t)}{\partial t} \Delta t$$
$$DFD = \frac{\partial I(x,y,t)}{\partial x} dx + \frac{\partial I(x,y,t)}{\partial y} dy + \frac{\partial I(x,y,t)}{\partial t} \Delta t = 0$$

Si on divise l'équation si dessous par rapport à la distance en temps inter-images t, on obtient l'équation de la contrainte du mouvement ECMA nommé aussi l'équation du flux optique (EFO) [Horn81].

$$\frac{\partial I(x,y,t)}{\partial x} Vx(x,y,t) + \frac{\partial I(x,y,t)}{\partial y} Vy(x,y,t) + \frac{\partial I(x,y,t)}{\partial t} = 0 \tag{2.26}$$

Où $Vx(x,y,t) = d_x \Delta t$, $Vy(x,y,t) = d_y \Delta t$ sont les deux composantes de la vitesse. Le calcul du flot optique consiste donc à résoudre l'équation du flot optique, appelée aussi équation de Contrainte du mouvement apparent (ECMA):

$$uI_x + vI_y + I_t = 0 \iff \nabla I(x) + I_t = 0 \tag{2.27}$$

Où le $\nabla I(x)$ désigne le gradient spatial de l'image et I_t sa dérivée temporelle du flot optique, cette équation appelée l'équation de contrainte du flot optique.

La méthode de Horn et Schunck est une méthode différentielle itérative adaptée à l'estimation des petits mouvements et basée sur le développement en série de Taylor du DFD, des gradients spatiaux et temporaux. L'objectif est de trouver les champs des vecteurs qui satisfaits l'équation de contrainte du mouvement en chaque pixel. Soit une image $I(x,y,t)$, on pose les hypothèses classiques de L'ECMA:

Avec les composantes $\frac{\partial I(x,y,t)}{\partial x} = I_x$, $\frac{\partial I(x,y,t)}{\partial y} = I_y$, $\frac{\partial I(x,y,t)}{\partial t} = I_t$, on obtient l'équation. suivante:

$$\begin{aligned} &I_t + V.\nabla I = 0 \\ &V_x I_x + V_y I_y = -I_t \\ &(I_x, I_y).(V_x, V_y) = I_t \end{aligned} \tag{2.28}$$

Finalement la contrainte d'intensité de l'image s'exprime par F_H (Ω représente l'ensemble des positions spatiales):

$$F_H = \iint_\Omega (V_x I_x + V_y I_y + I_t)^2 dxdy \tag{2.29}$$

Horn et Schunck introduisent une contrainte supplémentaire appelée contrainte de lissage de l'image. Mathématiquement, cette contrainte peut s'exprimer comme la minimisation de la quantité F_S suivante:

$$\iint F_S = \iint_\Omega ((\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2)^2 \tag{2.30}$$

Horn et Schunck introduisent une fonction de coût dont le minimum est obtenu pour le champ de vitesse recherché. Cette fonction $F_C = F_H + \alpha^2 F_S$ est une combinaison des contraintes d'intensité et de lissage. α^2 contrôle l'influence du terme de lissage. En dérivant F_C selon x puis selon y, on obtient le système suivant:

$$\begin{cases} I_x^2 V_x + I_x I_y V_y = \alpha^2 \nabla^2 V_x - I_x I_t \quad (1) \\ I_y^2 V_y + I_x I_y V_x = \alpha^2 \nabla^2 V_y - I_y I_t \quad (2) \end{cases} \tag{2.31}$$

En effectuant $I_y(1) - I_x(2)$, on obtient:

$$I_y(I_x^2 V_x + I_x I_y V_y) - I_x(I_y^2 V_y + I_x I_y V_x) = I_y(\alpha^2 \nabla^2 V_x - I_x I_t) - I_x(\alpha^2 \nabla^2 V_y - I_y I_t)$$

$$(I_y I_x^2 V_x + I_x I_y^2 V_y) - (I_x I_y^2 V_y + I_x^2 I_y V_x) = (I_y \alpha^2 \nabla^2 V_x - I_y I_x I_t) - (I_x \alpha^2 \nabla^2 V_y - I_x I_y I_t)$$

$$\Leftrightarrow I_y \alpha^2 \nabla^2 V_x = I_x \alpha^2 \nabla^2 V_y \Leftrightarrow \frac{I_y}{I_x} \nabla^2 V_x = \nabla^2 V_y \Leftrightarrow \nabla^2 V_x = \nabla^2 V_y \frac{I_y}{I_x}$$

Selon l'approximation du Laplacien $\nabla^2 V_x = \overline{V_x} - V_x$, on a:

$$\Rightarrow \nabla^2 V_x = \frac{I_x}{I_y}(\overline{V_y} - V_y) \Rightarrow \overline{V_y} = \frac{I_y}{I_x} \nabla^2 \overline{V_x} - V_y$$

En remplaçant $\nabla^2 V_x$ dans (1), on obtient:

$$I_x^2 V_x + I_x I_y V_y = \alpha^2 \nabla^2 V_x - I_x I_t$$

$$\Rightarrow V_x = \overline{V_x} - I_x \frac{(I_x \overline{V_x} + I_y \overline{V_y} + I_t)}{(\alpha^2 + I_x^2 + I_y^2)} \text{ De même } V_y = \overline{V_y} - I_y \frac{N}{(\alpha^2 + I_x^2 + I_y^2)}$$

2.3.2.4.2 Approche mixte CAFO: Contour Actif et Flot Optique

Les méthodes d'estimation et de segmentation conjointe du mouvement définissent un critère dépendant à la fois de la région et du mouvement à estimer. La minimisation de ce critère conjoint a pu être traitée comme une minimisation hiérarchique non linéaire [Mémin02], comme un problème aux valeurs propres ou par l'évolution d'un contour actif [Paragios00].

Les contours actifs sont particulièrement bien adaptés pour la segmentation et le suivi d'objets en mouvement parce que les objets bougent et se déforment peu entre deux images. De ce fait il est fréquent d'initialiser la segmentation d'une image par le contour final de l'image précédente, et l'algorithme converge d'autant plus vite que le contour initial est proche de l'objet d'intérêt. Dans [Staib92], les auteurs formulent l'hypothèse d'un modèle de mouvement global pour toute la région d'intérêt. Les auteurs utilisent la valeur absolue de l'image différence.

Nous présentons dans cette section notre approche de segmentation des objets en mouvement dans une vidéo, une approche qui étend les nombreux travaux sur le sujet. Un calcul du flot optique par la méthode de Horn & Schunk est préalablement effectué. La norme du flot calculé est ensuite incorporée dans un modèle de contour actif basé sur la minimisation d'une fonctionnelle d'énergie qui prend aussi en compte l'information photométrique.

Les objets d'intérêt ne sont pas toujours très visibles et très distinguables du fond. Toutes les méthodes de segmentation trouvent leurs limites quand il y a une occlusion ou quand l'objet possède les mêmes caractéristiques statistiques tels que la moyenne et la texture que son voisinage ou le fond. La phase d'estimation de mouvement par flot optique permet de séparer les objets qui bougent d'un fond statique. La figure 2.13 montre l'estimation du mouvement de pixels entre deux images consécutives. Pour une bonne visibilité, nous avons opté pour l'affichage d'un masque binaire (paramétrable en fonction du gradient) qui montre les pixels qui ont bougé. Néanmoins, pour l'exploitation du flot dans notre approche, c'est l'amplitude des déplacements qui est exploitée. On remarque d'ores et déjà, que seules les pixels qui ont changé de couleur, sont gardés.

Figure 2.13: Deux images consécutives de séquences à gauche (a et b) et l'estimation du flot optique (c)

Rappelons notre énergie à minimiser qui est de la forme générale.

$$\min_{u \in [0,1]} \left\{ J(u, c_1, c_2) = \mu \int_{\Gamma} g|\nabla u| + \lambda \int_{\Omega} u\left[r^1 - r^2\right] \right\}$$

Les fonctions r^1 et r^2 décrivent l'a priori sur les statistiques des données sur Ω_{int} et Ω_{ext}. Pour la détection des objets en mouvement, l'importance se situe au niveau du terme décrivant la région extérieure au contour.

- Estimation du fond de la séquence d'images : ($g = 1$, $r^1 = \alpha$ et $r^2 = |B - I|$).

Quand $g = 1$, $r^1 = \alpha$ et $r^2 = |B - I|$, on retrouve le modèle proposé dans Jehan Besson, Barlaud et Aubert [Jehan-Besson 01]. Ce descripteur est basé sur la différence entre une image et le fond estimé à partir de plusieurs images de la séquence (B).

Nous avons proposé, une approche CAFO [Zinbi06] qui combine à la fois les contours actifs et le flot optique et qui utilisent les paramètres suivants: ($g = 1$, $r^1 = \alpha$ et $r^2 = \|F\|$) avec F le vecteur du flot optique de la séquence et $\|v(x)\|$ sa norme. Cela nécessite parfois du seuillage et l'utilisation d'opérateurs morphologiques pour mettre en évidence certaines zones d'activité ou pour éliminer du bruit.

Figure 2.14: Propagation du contour sur une image d'Akiyo, α=40 et $\lambda = 10$

- Détecteur de bords dans une zone d'activité:

Quand $g(x) = \frac{1}{1+x^2}$, $r^1 = \alpha$ et $r^2 = \|F\|$, on retrouve le modèle proposé par F. Ranchin, et F. Dibos [Ranchin04]. La différence étant que l'on ne considère pas une image de fond et que l'on introduit le détecteur de bords $g(\|\nabla I\|)$. L'idée est donc de contrôler la norme du flot sur la région extérieure, tout en contrôlant la régularité de la frontière et en empêchant celle-ci de traverser des zones homogènes de l'image où $\nabla I = 0$ et $g(\|\nabla I\|) = 0$.

La constante α représente un seuil de référence de l'amplitude du flot au-delà de laquelle on considère qu'il y a mouvement. $\mu g(\|\nabla\|)$ vise à régulariser la frontière tout en l'attirant vers un bord.

Le détecteur de bords permet de contrebalancer quelques effets indésirables de la segmentation au sens du mouvement, à savoir l'obtention d'un flot optique d'amplitude trop faible sur les zones homogènes et bien sûr une localisation approximative des objets en mouvement (l'algorithme de Horn & Schunk est effectué en 2d+t sur la totalité de la vidéo et mélange donc les informations obtenues sur chacune des images).

En pratique, le seuil α est choisi ad hoc. L'initialisation correspond précisément au masque obtenu en seuillant la norme du flot au niveau α. Les figures suivantes montre le masque de mouvement obtenu par Horn & Schunk en 2d+t sur des images extraites de vidéos différentes.

Figure 2.15: Propagation du contour sur quelques séquences: α*=40* **et** $\lambda = 10$

Sur les exemples figure 2.16 et la figure 2.17, nous montrons l'efficacité de cette méthode sur un corpus de vidéos d'activités humaines (Courir, Sauter,…) qui sera présenté dans le chapitre V.

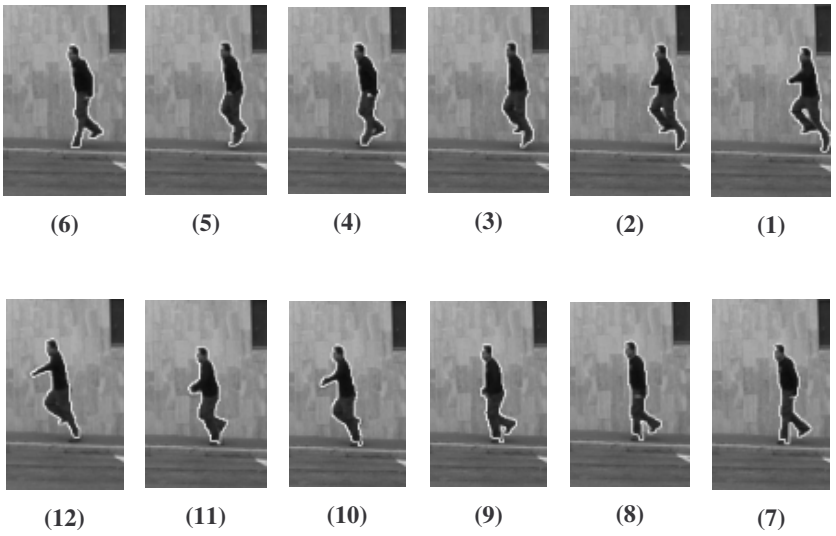


Figure 2.16: La séquence vidéo du mouvement Shahar (jump), pour α=120, λ=15

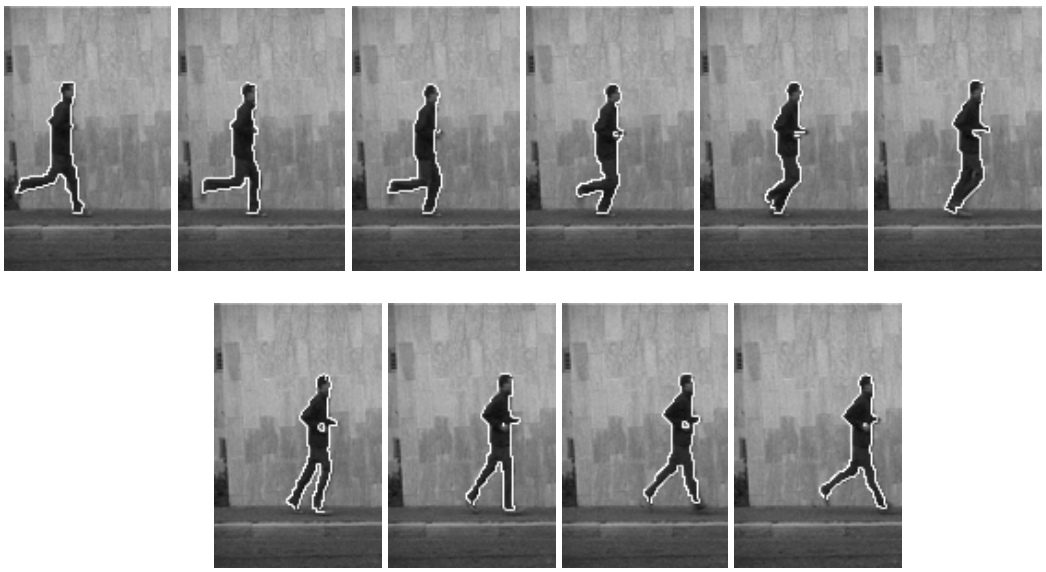


Figure 2.17: La séquence vidéo du mouvement Ido (run) avec α=120, λ=20

Nous avons aussi proposé une méthode de segmentation au sens du mouvement, basée sur la segmentation par contours actifs globaux dans des zones d'activités. Ces dernières sont déterminées par des masques dynamiques issues du calcul du flot optique global [Zinbi08a]. La figure 2.18 montre un exemple de segmentation d'objets en mouvement.

Figure 2.18: Les zones d'activité extraites par FO

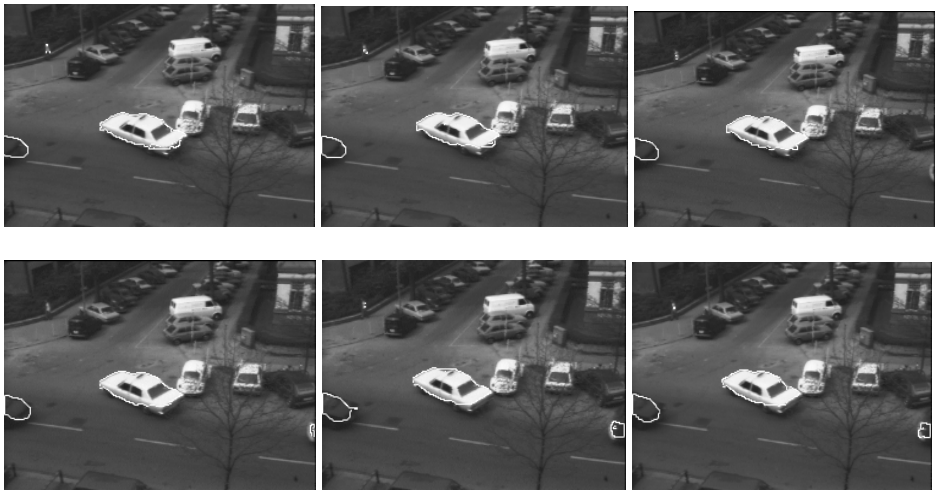


Figure 2.19: Segmentation d'objets dans les zones d'activité (ex. Séquence du taxi)

2.4 Conclusion et perspectives

Dans ce chapitre, nous avons situé la problématique de la segmentation d'objets en mouvement sur des images 2d et 2d+t. Par la suite, nous avons présenté dans ce chapitre la méthode des contours actifs globaux avec une technique d'optimisation qui a montré un grand potentiel pour résoudre plusieurs problèmes en vision. Nous avons expérimenté cette méthode dans le cas de segmentation binaire rapide où elle est très efficace et robuste.

Le couplage de cette méthode avec le flot optique donne un algorithme de segmentation 2D + t. La formulation énergétique du modèle de contour actif a été étendue par l'intégration d'un critère supplémentaire issu du calcul du flot optique. Les avantages constatés: changement de topologie automatique, plus stable numériquement, aussi moins sensible aux contour initiale et au bruit, elle est applicable en 3D ou toute autre dimension de la même façon que les coupes de graphe. En général, le temps d'exécution est plus rapide par rapport aux approches basées sur les EDP. L'application sur les mouvements peut être réalisée sur chaque image 2D et le contour final trouvé et utilisé comme un contour initial de l'image

suivante. Les résultats peuvent être spatialement incohérents. Nous avons préféré l'application sur la séquence comme un volume unique (N-D) utilisant les contraintes dures temporelles avec un terme de gradient spatial.

L'évaluation a montré que l'approche proposée donne de bons résultats, en terme de temps de calcul et d'extraction d'objets vidéo complexes, sur un corpus varié de vidéos. Néanmoins, plusieurs critères caractérisant les propriétés des objets d'intérêt restent à prendre en compte, tels que les descripteurs de forme, de texture ou même de type de mouvement. L'utilisation d'une énergie de la forme à priori en 3D par template de la forme [Freedman05] ou des poses [Bray06] restes possible.

CHAPITRE 3 Catégorisation par l'analyse spectrale des graphes

3 Méthodes spectrales d'analyse en imagerie vidéo

3.1 Introduction

Face à l'accroissement et à la diversité des informations disponibles, la catégorisation des objets et des contenus s'impose de plus en plus comme une technologie clé dans la gestion de l'intelligence. Elle n'en reste pas moins un domaine scientifique et technique complexe qui requiert des connaissances avancées en matière de technologies de traitement des informations textuelles, audio-visuelles et langagières. L'analyse de données nécessite des méthodes de catégorisation qui visent à regrouper en classes homogènes un ensemble d'observations. Ces dernières années, les besoins d'analyse de données et en particulier de classification ont augmenté significativement. En effet, de plus en plus de domaines scientifiques nécessitent de catégoriser leurs données dans un but descriptif ou décisionnel.

Les méthodes de classification se divisent généralement en deux sous problèmes distincts: la classification supervisée, appelée également analyse discriminante, et la classification non supervisée, dénommée aussi classification automatique. Les premières approches qui ont été proposées en classification étaient algorithmiques, heuristiques ou géométriques et reposaient essentiellement sur la dissimilarité entre les objets à classer. L'approche statistique, plus récente, se base sur des modèles probabilistes qui formalisent l'idée de classe. Cette approche permet en outre d'interpréter de façon statistique la classification obtenue. D'autre part, les processus d'acquisition des données ayant aussi progressé rapidement, la dimension des données à étudier est devenue très grande. Le monde scientifique d'aujourd'hui fournit des données qui sont chaque jour plus nombreuses et de plus grande dimension. On peut citer l'analyse d'image et de vidéos où les données sont également de grande dimension, voir de très grande dimension si l'on considère les résolutions actuelles des appareils photos numériques.

Nous proposons dans ce chapitre d'étudier les variétés ainsi que les méthodes de regroupement permettent de réduire le nombre des variables pour l'analyse de données, dans le but d'extraire par la suite des structures homogènes. Cela devrait introduire des solutions plus robustes (moins sensibles au bruit et aux données aberrantes) et permettre l'analyse visuelle de la structure de l'information. Nous présentons synthétiquement plusieurs méthodes que nous avons étudiées, basées sur une décomposition spectrale. On pourra trouver un «tour d'horizon» détaillé des méthodes de réduction de dimension existantes dans [Carreira97] [Fodor02]. Ensuite, nous allons illustrer par la mise en œuvre d'une nouvelle approche de catégorisation appliquée sur données réelles en imagerie vidéo.

3.2 Fléau de la grande dimension

Dans la littérature, le terme de «fléau de la dimension» est abondamment utilisé pour caractériser les différentes manifestations de la grande dimension. Le « fléau de la dimension»

est un terme que l'on doit à Bellman [Bellman57] qui l'utilisa comme principal argument en faveur de la programmation dynamique. Bellman utilisa ce terme pour parler de la difficulté d'optimiser une fonction par une recherche exhaustive de l'optimum dans un espace discrétisé.

Certains problèmes, tels que la reconnaissance de visages, fournissent des données en très grande dimension (un millier de dimensions) et le nombre n d'observations disponibles est généralement beaucoup plus faible que la dimension p. Il est clair que, dans ce cas, la dimension des données est artificiellement augmentée par le processus d'acquisition. En effet, d'un point de vue géométrique, n points vivent dans un espace de dimension au plus (n−1). Cet exemple met en évidence le fait que la dimension acquise est, en général, nettement supérieure à la dimension intrinsèque des données. Ce raisonnement induit qu'une grande part des variables est corrélée et donc qu'une grande part de l'information est redondante. Par conséquent, si nous parvenons à nous ramener à un système de d variables indépendantes, alors le fléau de la dimension sera fonction de la dimension intrinsèque d qui peut être très faible devant p.

D'autre part, on observe généralement un phénomène intéressant dans le cadre de la classification: plus la dimension de l'espace est grande, plus la classification des données est facile avec un classifieur adapté. En général, avec un classifieur adapté, la tâche de classification est plus facile dans un espace de grande dimension que dans un espace de faible dimension. Ce phénomène est en particulier exploité par les méthodes de discrimination SVM, présentées plus tard, qui augmentent artificiellement la dimension des données pour faciliter leur discrimination.

Une des solutions qui peut être mise en œuvre pour limiter les effets du «fléau de la dimension» est de réduire la dimension des données avant de les traiter. C'est en effet la solution la plus naturelle puisqu'elle prend le problème à la source : la dimension est trop grande, alors réduisons-la ! De fait, il est théoriquement possible de réduire la dimension de l'espace à d dimensions et ce sans entraîner de perte d'information. L'enjeu est donc d'identifier les dimensions (ou les combinaisons de dimensions) qui sont porteuses d'informations redondantes. Les techniques de réduction de dimension sont traditionnellement divisées en deux catégories:

– les méthodes d'extraction de caractéristiques (feature extraction),
– les méthodes de sélection de caractéristiques (feature selection).

Les méthodes d'extraction de caractéristiques construisent, à partir des p variables (dimensions) originales, d nouvelles variables qui contiennent la plus grande part possible de l'information initiale. Parmi toutes les techniques d'extraction de caractéristiques existantes, la plus connue et la plus utilisée est très certainement l'analyse en composantes principales (ACP) qui est une méthode linéaire.

Les méthodes de sélection de caractéristiques, quant à elles, cherchent un sous-ensemble de d variables parmi les p variables originales. La recherche peut-être optimale en utilisant une méthode de sélection exhaustive si le nombre de dimensions de l'espace original n'est pas trop grand. En pratique, ces méthodes de recherche exhaustive ne sont pas utilisables avec les données modernes qui sont décrites par un trop grand nombre de dimensions. En effet, le nombre de sous-ensembles possibles est égal à C_p^d :

$$C_p^d = \frac{p!}{(p-d)!d!}$$

On comprend vite la nécessité d'introduire des méthodes sous-optimales. D'autre part, les différentes méthodes de sélection de variables se différencient les unes des autres de par le choix du critère mesurant la pertinence du sous-ensemble de variables. Ces méthodes sont détaillées dans [Guyon03] [Webb02].

Un point essentiel de la réduction de dimension, par extraction ou sélection de variables, qui est le nombre de dimensions qu'il faut retenir, ne possède malheureusement pas de solution explicite. Nous allons présenter uniquement ici des critères permettant de déterminer le nombre de dimensions à retenir dans le cadre de méthodes d'extraction de caractéristiques tels que ACP. La plupart des techniques de recherche du nombre d'axes à retenir sont basées sur les valeurs propres de la matrice de covariance $\sum$ des données. Cette approche se justifie par le fait que chaque valeur propre de $\sum$ représente la variance portée par le vecteur propre associé.

3.3 Classification non-Supervisé de Variétés

L'apprentissage non supervisé vise à caractériser la distribution des données, et les relations entre les variables, sans discrimination entre les variables observées et les variables à prédire. Les formes principales d'apprentissage non supervisé sont les suivantes:

– L'estimation de fonction de densité ou de fonction de probabilité. C'est la forme la plus générale d'apprentissage non supervisé. On a un critère clair, la log-vraisemblance (mais certains remettent cela en question). On apprend explicitement la fonction p(x).

– La découverte de classes naturelles, ou clustering (e.g., l'algorithme K-moyennes), qui cherche à découvrir les modes principaux de la distribution, les «prototypes», les catégories principales, etc.. Cela donne une forme de réduction de dimensionnalité qui associe un entier à chaque exemple.

– L'apprentissage de variétés de faible dimension, c'est à dire de surfaces (planes ou non-linéaires) prés desquelles se retrouvent la majorité des données, en haute dimension. On obtient ainsi une représentation de faible dimension des données, ce qui peut être une étape importante pour visualiser les données et/ou comme prétraitement avant

l'apprentissage supervisé. Très utile en finance pour réduire la dimensionnalité et éviter l'overfitting.

L'objectif d'une méthode de classification déborde le cadre strictement exploratoire. C'est la recherche d'une typologie ou segmentation c'est-à-dire d'une partition ou répartition des individus en classes, ou catégories.

Les problèmes de réduction et d'analyse de données sont classiquement abordés par les techniques de sélection de variables et de réduction de dimension, qui visent à trouver des structures intrinsèques de dimension plus réduite cachée dans les observations dimensionnelles, dont l'analyse en composantes principales (ACP) est un exemple bien connu. Récemment, une nouvelle famille de méthodes est apparue, fondée sur des approximations globales de données Isomap, LLE, qui font l'objet de nombreuses recherches.

3.3.1 Réduction linéaire de dimension

3.3.1.1 Analyse en Composantes Principales (ACP)

L'ACP est une méthode mathématique d'analyse des données qui consiste à rechercher les directions de l'espace qui représentent le mieux les corrélations entre n variables aléatoires. Dans la plupart des situations, on dispose de plusieurs observations sur chaque individu constituant la population d'étude. On a donc à prendre en compte p variables par individu, p étant strictement supérieur à 1. L'étude séparée de chacune de ces variables donne quelques informations mais elle est insuffisante car elle laisse de côté les liaisons entre elles, ce qui est pourtant souvent ce que l'on veut étudier.

L'ACP est une bonne méthode pour étudier les données multidimensionnelles, lorsque toutes les variables observées sont de type numérique, de préférence dans les mêmes unités, et que l'on veut voir si il y a des liens entre ces variables. Dans la littérature, on trouve deux approches différentes de l'ACP. Elle peut être présentée comme la recherche d'un ensemble réduit de variables non corrélées, combinaisons linéaires des variables initiales résumant avec précision les données (approche anglo-saxonne). Une autre interprétation repose sur la représentation des données initiales à l'aide de nuage de points dans un espace géométrique. L'objectif est alors de trouver des sous-espaces (droite, plan,...) qui représentent au mieux le nuage initial. C'est cette dernière approche que nous aborderons par la suite. On part d'un tableau de données rectangulaire, représentant toutes les données, en plaçant en ligne les individus et en colonnes les variables. Soit X ce tableau de données.

$$X = \text{individus....} \underbrace{\begin{bmatrix} x_1^1 & & x_1^p \\ & & \\ x_1^n & & x_p^n \end{bmatrix}}_{\text{variable}}$$

Par convention, on placera en exposant ce qui se rapporte aux individus: le premier individu est donc x1, et en indice ce qui se rapporte aux variables: x1 désigne la première variable. X11 désigne la valeur numérique prise par la première variable, pour le premier individu. Et le tableau de données regroupe toutes les valeurs prises par tous les individus (de 1 à n) par p variables, soit encore:

$$X = (x_j^i)_{(1\le i\le n, 1\le j\le p)} \text{ et } \forall (i,j), x_j^i \in R \tag{3.1}$$

Le but est de trouver le sous-espace vectoriel E_k de dimension k ($k < p$ souvent $k = 2$) tel que I_{E_K}, inertie du nuage N par rapport à l'espace E_k soit minimum.

$$I_{E_K} = \sum_{i=1}^{n} p_i d_M{}^2(x^i, E_K) \tag{3.2}$$

p_i : pondération sur les individus. Chaque individu i est muni du poids p_i. La plupart du temps, on se place dans un cadre d'équipondération: tous les individus ont le même poids.

d_M : distance définie par la métrique M. En pratique, on considère deux métriques différentes: I, la métrique identité, ou la métrique $D=1/\sigma^2$ qui réduit les variables. Réduire un tableau de données consiste à calculer l'écart type pour chacun des caractères et à exprimer toutes les cases en nombre d'écarts types (positif ou négatif). L'écart type devient ainsi une mesure unique commune à tous les caractères et les unités dans lesquels s'expriment initialement les données n'ont plus d'importance. $d_M(x^i, E_K)$ désigne la distance entre x_i et E_k soit la distance entre xi et son projeté sur E_k.

On procède alors de la manière suivante :

- Recherche d'un axe Δu_1 maximisant l'inertie $I_{\Delta u_1^{\perp}}$
- Recherche d'un axe Δu_k, M-orthogonal à E_{k-1} maximisant l'inertie $I_{\Delta u^{\perp}_k}$

On note V la matrice d'inertie du nuage N, qui est aussi la matrice de covariance des caractères $(x_i, ..., x_p)$. La solution est alors obtenue en utilisant les propriétés spectrales des matrices: les vecteurs propres normés de la matrice VM ordonnés suivant les valeurs propres décroissantes fournissent les axes Δu_1, Δu_k, appelés axes factoriels. De plus, les inerties expliquées par ces axes sont égales aux valeurs propres Δk. Les u_i forment une base M-orthonormée de E_k : les vecteurs u_i sont par définition normés et par ailleurs, la matrice VM étant symétrique, ses vecteurs propres sont orthogonaux.

Le problème initial était d'obtenir une représentation du nuage N dans des espaces de dimension réduit. On connaît maintenant les axes définissant ces espaces. Pour pouvoir obtenir les différentes représentations, il suffit de déterminer les coordonnées de la projection de tous les points du nuage sur chaque axe factoriel. Soit $C_1^i,.....,C_n^i$ ces n coordonnés pour l'axe i.

Le vecteur $C^i = \begin{pmatrix} C_1^i \\ .... \\ C_n^i \end{pmatrix}$ est appelé $i^{ème}$ composante principale.

On peut alors obtenir «l'image» du nuage N dans un plan factoriel quelconque (u_i,u_j) grâce aux composantes principales c_i et c_j. La représentation dans le premier plan factoriel est obtenue grâce à c_1 et c_2. En utilisant conjointement la représentation du plan, on peut «voir» le nuage dans le sous-espace E3. Le calcul des composantes principales se fait par changement de base. Il suffit de faire une projection orthogonale sur les nouveaux vecteurs de base. Ainsi, pour la $i^{ème}$ composante principale, on a:

$$c^i = (c_i^j)_{1\le j\le n} \text{ avec } c_i^j = M(u_i, x^j)$$

D'où l'expression de la composante principale: $c^i = XMu_i$

3.3.1.2 Le Multi-Dimensional Scaling (MDS)

Parfois on a pas les coordonnées des exemples, mais seulement les distances (ou autre mesure de similarité) entre chaque paire d'exemples. Le MDS classique trouve une représentation des exemples qui correspond exactement à l'ACP, mais en partant de ces distances D_{ij}. L'algorithme est le suivant:

1. Moyennes par rangées: $u_i = \frac{1}{n}\sum_j D_{ij}$,
2. Double centrage (distance vers produit scalaire): $P_{ij} = \frac{1}{2}(D_{ij} - u_i - u_j + \frac{1}{n}\sum_i u_i)$
3. Calcul des vecteurs propres v_j et valeurs propres λ_j principales de la matrice
4. La $i^{ème}$ coordonnée réduite de l'exemple j est $\sqrt{\lambda_i}v_{ij}$.

Tableau 3-1: Algorithme MDS classique

Notons bien que seuls les exemples d'apprentissage reçoivent une coordonnée réduite. On peut généraliser le MDS pour permettre des variétés non-linéaires mais on obtient une fonction de coût non-convexe qui peut être difficile à optimiser. On peut généraliser le MDS pour permettre des variétés non-linéaires mais on obtient une fonction de coût non-convexe qui peut être difficile à optimiser.

3.3.2 Méthodes non linéaires de réduction de dimension

L'ACP et MDS considèrent les distances euclidiennes dans l'espace d'observation. Si on Suppose que les données ont une dimension intrinsèque m plus faible que la dimension d'observation n, l'ACP ne pourra trouver un système de coordonnées en m dimensions exact que si la variété à partir de laquelle sont tirées les données est en fait un sous espace linéaire. Si les données sont plutôt tirées d'une variété non linéaire, comme à la figure 3.1.A), l'ACP sera incapable d'exprimer les caractéristiques de cette variété et ne pourra pas par conséquent créer une représentation en plus faible dimension respectant les longueurs des géodésiques sur la variété.

La plupart de ces algorithmes d'apprentissage linéaires ne font intervenir que des produits scalaires entre les observations. Ces algorithmes peuvent être généralisés afin de considérer les relations entre les fonctions non linéaires des observations. Il suffit de remplacer chaque produit scalaire $\langle x, y \rangle$ par un noyau $k(x, y)$ dans l'algorithme. Le choix de k permettra alors de considérer des produits scalaires (ou relations de covariance) dans un espace de caractéristiques possiblement non linéaires plutôt que dans l'espace original d'observation.

Récemment, de nombreuses méthodes de réduction de dimension non-linéaire ont été proposées.

- Une première catégorie de méthodes se propose d'étendre l'ACP linéaire classique au cas non-linéaire. Parmi ces méthodes, Kernel-PCA (KPCA) [Schölkopf98] qui utilise les fonctions noyaux des SVM pour transformer les ou de «surfaces principales» [Delicado01] [Girard05] [Hastie89] recherchent, non plus un hyper-plan comme en ACP, mais une hyper-surface paramétrée et lisse qui approche au mieux les données originales avant d'appliquer une ACP classique sur les données transformées. Les méthodes dites de «courbes principales»

- La seconde catégorie de méthodes de réduction de dimension non-linéaire est basée sur l'idée que les données sont disposées sur une variété non-linéaire de dimension intrinsèque d dans l'espace de dimension p. Ces méthodes ont généralement comme principal objectif de permettre la visualisation des données de grande dimension. Pour cela, elles cherchent à «déplier» la variété sur laquelle vivent les données. Ces méthodes font parties des méthodes neuronales dans le sens où les points (qui jouent le rôle de neurones) cherchent leur position dans l'espace de sortie tout en respectant (tout au moins localement) la topologie d'entrée.

La première méthode de ce type qui a été proposée est le Multi-Dimensional Scaling (MDS) dont on pourra trouver la technique détaillée dans de nombreux livres tels que [Hastie01]. Ces dernières années, plusieurs méthodes dérivées ont vu le jour qui s'opposent principalement sur la question du critère de similarité entre les topologies d'entrée et de sorties. Parmi ces

extensions, nous pouvons citer, la Locally Linear Embedding (LLE) [Roweis00] et la méthode Isomap [Tenenbaum00] que nous présentons dans le paragraphe suivant.

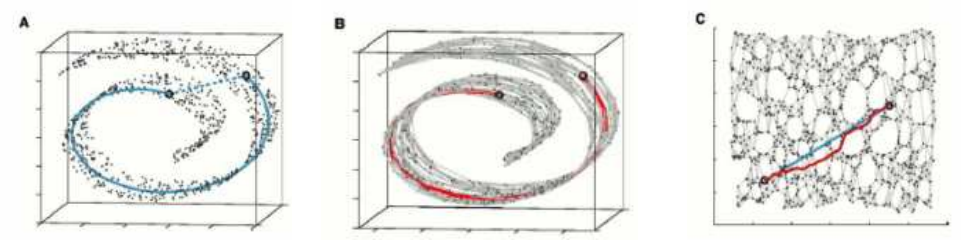


Figure 3.1: Principe de l'algorithme Isomap : (A) Variété non linéaire en deux dimensions de données synthétiques. (B) Distance géodésique. (C) Projection dans un espace réduit où la distance euclidienne est représentative [Tenenbaum00].

La figure 3.1 illustre comment Isomap exploite les géodésiques pour faire une réduction non linéaire de la dimensionnalité. Dans la figure 3.1-A, la distance euclidienne (en bleue pointillé) ne correspond pas à la notion de distance souhaitée (bleue trait plein). Dans (B), la géodésique entre deux points correspond à la distance, cette distance est définit comme une longueur du plus cours chemin dans un graphe reliant les plus proches voisins de chaque point. Dans (C) l'image montre que la distance du plus court chemin est bien préservée par le plongement dans un espace bi-dimensionnel où la distance euclidienne est plus proche des longueurs des géodésiques dans l'espace d'origine.

3.3.2.1 ACP à noyau (Kernel PCA)

Schématiquement l'analyse en composantes principales (PCA) est un changement de repère qui vise à privilégier les axes de variance maximale par rapport à un ensemble de données. Les axes où la variance des données est réduite peuvent être éliminés pour atteindre une réduction de la dimensionnalité avec une perte minimale d'information. La transformation est, par essence, linéaire (matrice de passage orthogonale). Or, pour les vecteurs issus de nos applications en vidéo, il est souhaitable de pouvoir atteindre des relations non-linéaires; la méthode Kernel PCA est une première extension de PCA qui l'envisage.

Kernel PCA réalise une analyse en composantes principales dans l'espace K appelé espace des caractéristiques à l'aide d'une fonction noyau. L'algorithme d'analyse en composantes principales à noyau est détaillé au tableau 3.2. Cette procédure revient à effectuer une analyse en composantes principales dans un espace de caractéristiques de haute dimension.

– La matrice de covariance C dans l'espace des $\phi(x)$ (qui peut maintenant être de dimension infinie) est $C = \frac{1}{n}\sum_{i=1}^{n}(\phi(x_i)-u)(\phi(x_i)-u)'$, avec $u = \frac{1}{n}\sum_i \phi(x_i)$. Les vecteurs propres u_j de cette matrice sont liés aux vecteur propres v_j de la matrice de Gram K.

$K_{ij} = K(x_i, x_j) - \overline{K}_i - \overline{K}_j + \overline{\overline{K}}$ et $\overline{K}_i = \frac{1}{n}\sum_j k(x_i, x_j), \overline{\overline{K}} = \frac{1}{n}\sum_i \overline{K}_i$

– On obtient la projection de $\phi(x)$ sur le $j^{ème}$ vecteur principal de C (donc la $j^{ème}$ coordonnée d'un exemple quelconque x) avec

$$u_i\phi(x) = \frac{1}{\lambda_j}\sum_j v_{ji} K(x, x_i)$$

où λ_j est la $j^{ème}$ valeur propre de la matrice de Gram K et v_{ji} le $i^{ème}$ élément du $j^{ème}$ vecteur propre.

Tableau 3-2: Analyse en composantes principales à noyaux permettant d'exprimer les données en d dimensions

3.3.2.2 Local Linear Embedding (LLE)

L'algorithme LLE modélise une variété comme une union de petits espaces linéaires. Il exploite la géométrie locale des points x_i dans l'espace original pour la reproduire dans un espace de plus faible dimension. Chaque point x_i a un voisinage $N(i)$ et l'idée consiste à exprimer x_i comme une combinaison linéaire de ses voisins $N(i)$ et de construire son image dans le nouvel espace y_i en respectant cette relation.

C'est une méthode qui permet de découvrir une variété non-linéaire basée sur l'approximation dans un espace de faible dimension des relations géométriques locales dans chaque région délimitée par les k plus proches voisins d'un exemple.

Algorithme:

– Trouver les m plus proches voisins x_j de chaque exemple x_i.

– Trouver pour chaque exemple i les poids w_{ij} sur les voisins j qui minimisent l'erreur de régression $(x_i - \sum_j w_{ij}x_j)^2$, avec la contrainte $\sum_j w_{ij}x_j = 1$. On prend $w_{ij} = 0$ si j n'est pas un voisin de i.

– Transformer la matrice sparse de poids W en $M = (I-W)'(I-W)$ symétrique.

– Calculer les k vecteurs propres v_j de plus petite valeur propre (excluant la plus petite, qui est 0), ce qui donne les coordonnées réduites des exemples d'apprentissage, $(v_{1i}, v_{2i}, ..., v_{ki})$.

Notez que les vecteurs propres sont les coordonnées réduites y_i pour chaque exemple x_i qui minimisent l'erreur.

$$\sum_i \left\| y_i - \sum_j w_{ij} \right\|^2 \tag{3.3}$$

sous la contrainte que $\sum_i y_{ij}{}^2 = 1$ (normalisation des coordonnées, sinon la solution est $y_i = 0$) et $\sum_i y_{ij} y_{ik} = 1_{j=k}$ (sinon il y a une infinité de solutions correspondant à des rotations des coordonnées). On cherche donc à reproduire la structure géométrique locale, mais avec un système de coordonnées de faible dimension.

3.3.2.3 Isomap

Isomap est une généralisation non-linéaire de l'algorithme d'échelle multidimensionnelle (MDS). MDS permet à partir des distances euclidiennes entre points, de déterminer un système de coordonnées réduit qui préserve les distances. L'idée fondamentale du MDS est la définition d'un produit à partir de la distance entre les vecteurs.

Comme LLE, cet algorithme se base sur les relations linéaires locales entre voisins pour capturer la structure de la variété, mais il a aussi une composante «globale», en essayant de préserver les distances le long de la variété. Pour cela on essaie d'approximer la distance géodésique sur la variété par la distance minimale dans un graphe dont les nœuds sont les exemples et les arcs seulement entre voisins sont associés aux distances locales.

Isomap [Tenenbaum00] possède une optimalité globale et la garantie de convergence asymptotique de l'ACP et de MDS tout étant capable d'apprendre une grande classe de variété non linéaire. Isomap est une généralisation de MDS qui préserve la géométrie intrinsèque des données, capturée par les longueurs des géodésiques passant par la variété à partir de laquelle sont tirées les données.

Pour des points voisins, les distances euclidiennes dans l'espace d'observation sont de bonnes approximations des longueurs des géodésiques. On construit un graphe reliant chaque point à ses k plus proches voisins. Les longueurs des géodésiques entre deux points éloignés sont alors estimées en trouvant la longueur du plus court chemin entre ces deux points dans le graphe. Il suffit finalement d'appliquer MDS aux distances obtenus pour obtenir un positionnement des points dans un espace de dimension réduite. Les trois étapes fondamentales d'Isomap sont énumérées au tableau 3.3.

La première étape détermine quels points sont voisins sur la variété V en se basant sur les distances $d_\chi(i, j)$ entre les paires de points i, j dans l'espace d'observation X afin de construire un Graph G approximant la variété.

La seconde étape consiste essentiellement à approximer les longueurs des géodésiques $d_v(i, j)$ entre tous les points i, j sur la variété. Le calcul du plus court chemin

$d_g(i,j)$ fournit cette estimation. L'algorithme proposé à l'étape 2 du tableau 3.3 pour ce faire peut être remplacé par tout autre algorithme de plus court chemin dans un graphe.

L'étape finale applique MDS à la matrice de distance D_G afin de fournir un positionnement des observations en d dimensions respectant le plus possible les longueurs des géodésiques sur V. L'algorithme est le suivant:

- Définir le graphe G sur l'ensemble des observations en connectant i et j $d_\chi(i,j) < \varepsilon$ (ε-Isomap) ou si i est des k plus proches voisins de j en fonction de $d_\chi(i,j)$ (k-Isomap)
- Initialiser $d_g(i,j) = d_\chi(i,j)$ si i et j sont connectés par un arc et $d_g(i,j) = \infty$ sinon.

  Calculer la longueur $D(i,j)$ du chemin le plus court dans le graphe entre chaque paire d'exemples: $D(i,j) = \min_p \sum_k d(p_k, p_{k+1})$ où p est un chemin $(p_1, p_2, ..., p_l)$ entre i et j dans le graphe $(p_1 = i, p_l = j)$.
- Appliquer l'algorithme MDS sur la matrice des distances géodésiques, $D_{ij} = D(i,j)$, ce qui donne les coordonnées réduites pour les exemples d'apprentissage.

Tableau 3-3 : Algorithme Isomap

L'algorithme Isomap prend en entrée les distances entre toutes les paires i, j de p points de donnée dans l'espace d'observation de haute dimension X. Cette distance peut être la distance euclidienne usuelle ou toute métrique spécifique à un domaine d'application particulier. L'algorithme produit des vecteurs de sortie $y_j \in \Re^n$ qui représentent le mieux possible la géométrie intrinsèque des données . Le seul paramètre libre ε ou k apparaît à l'étape 1.

3.3.2.4 Clustering spectral

L'extraction de caractéristiques se révèle souvent plus efficace en apprentissage supervisé lorsqu'on cherche à construire un classifieur mais demande à l'utilisateur un effort important d'interprétation pour comprendre la nouvelle représentation de ses données. Ensuite, d'une part, les techniques classiques utilisables en apprentissage non supervisé comme l'analyse factorielle en composantes principales (ACP), ou le positionnement multidimensionnel (MDS) sont limitées par leur caractère linéaire. D'autre part, les méthodes non linéaires comme Isomap ou le plongement localement linéaire (LLE) ont une complexité trop importante pour être utilisées sur de grandes masses de données; ce dernier point est toutefois à nuancer depuis l'apparition récente de méthodes incrémentales de calcul.

Récemment, l'intérêt pour le traitement de données s'est tourné vers les méthodes de clustering spectrales, en raison de nombreux succès [Chung97] [Geng05]. Ces méthodes

emploient le contenu spectral d'une matrice de similarité (distance entre chaque paire de données) pour réaliser la réduction et la partition d'un jeu de données. Plus spécifiquement, les vecteurs propres sont vus comme un outil fournissant une représentation (visualisation) des données dans un espace où les données sont bien séparées et peuvent facilement être groupées (classées).

L'approche spectrale est un algorithme de clustering qui consiste en deux étapes: d'abord la transformation des données en coordonnées réduites (correspondant à des variétés non linéaires), suivi d'une étape classique de clustering (comme l'algorithme k-moyennes).

L'algorithme est présenté dans le tableau ci-dessous:

- Appliquer un noyau $K(x_i, x_j)$ à chaque paire d'exemples (x_i, x_j)
- Normaliser le noyau: $K_{ij} = \frac{K(x_i, x_j)}{\sqrt{\mu_i \mu_j}}$, $\mu_i = \frac{1}{n}\sum_{i=1}^{n} K(x_i, x_j)$ est la moyenne par rangée.
- Calculer les vecteurs propres principaux v_j de la matrice K.
- Obtenir les coordonnées de norme 1 pour chaque exemple $i : (v_{1i}, v_{2i}, \ldots, v_{ki}) / \sqrt{\sum_i v_{ji}}$
- Appliquer un algorithme de clustering classique (k-moyennes) sur les exemples dans ce nouveau système de coordonnées.

Tableau 3-4: Algorithme de clustering spectral

La méthode des k plus proches voisins (k-ppv) est une approche non paramétrique qui ne fait appel a aucune hypothèse sur la répartition des différentes classes ou la nature des surfaces séparatrices. Le volume de la région r autour d'une forme xi est choisi de manière à contenir exactement k points de l'échantillon observe. On choisit généralement des volumes réguliers centres en x_i (hyper cubes, hyper sphères). Les performances de la méthode dépendent de la valeur de k, le nombre des plus proches voisins. Elle consiste a rechercher pour un nouveau vecteur a classer le sous-ensemble des k plus proches vecteurs de l'ensemble d'apprentissage au sens d'une distance (distance euclidienne, métrique adaptative) qui détermine le type de voisinage, puis a affecter a ce vecteur la classe majoritairement représentée dans le sous-ensemble.

3.3.2.5 Laplacian EigenMaps

Cet algorithme [Belkin03] est une variante et mélange les méthodes précédentes. Une représentation graphique des données est obtenue en considérant comme nœuds les points x_i et comme poids sur les arêtes, W_{ij} . Les distances sont calculées avec un noyau Gaussien ou

un noyau de type les k-plus proches voisins. Si t est la matrice diagonale avec éléments, la fonction à minimiser est:

$$\sum_{ij}(y_i - y_j)^2 W_{ij} \tag{3.4}$$

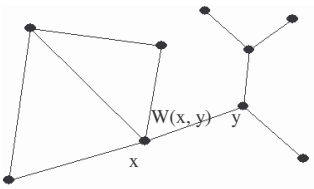
De manière similaire au LLE, la minimisation est forte sous les contraintes d'une projection centrée, de variance unitaire. La solution est trouvée grâce aux plus faibles valeurs propres.

3.4 Diffusion géométrique par marches aléatoires sur graphe

Soit (X,μ) un espace d'ensemble fini d'objets, où $X = \{x_1, x_2, ..., x_N\}$ un ensemble de jeu de m points de données tel que $\forall x_i \in X$, $x_i \in \Re^n$ (x_i est un élément de dimension n) et μ est la mesure qui compte représenter la distribution de ces points dans le jeu de données. Pour simplifier, nous supposerons que μ est fini. Dans cette section, nous rappellerons la structure de diffusion comme décrit dans [Lafon05], en commençant par la construction du noyau de diffusion, puis par la présentation de ses propriétés spectrales pour analyser la géométrie des données. Nous dans le chapitre suivant l'implémentation de cette approche dans le cas discret avec des applications pour la réduction, la réorganisation et la visualisation des images et vidéos.

3.4.1 Construction du noyau de diffusion

L'idée de base, issue de la théorie des graphes, est de représenter la variété de données X comme un graphe $G = (V, E)$ qui consiste en un ensemble fini de sommets $V = \{v_1, v_2, ..., v_n\}$ et un ensemble fini d'arêtes $E \subseteq V \times V$:



Supposons que la géométrie de X est définie par le noyau $w_\varepsilon(x, y)$. La notion de similarité entre deux points de données x et y est donc définie par :

$$w_\varepsilon(x, y) = exp\left(-\|x - y\|^2 / \varepsilon\right)$$

où ε est un paramètre d'échelle.

Le noyau w satisfait les conditions suivantes :

- w est symétrique : $w(x, y) = w(y, x)$ et non négative : pour tout x et y de X, $w(x, y) \geq 0$

- w est défini semi-positive :

$$\forall f \in L^2(X) \text{ non null } : \iint_{XX} w(x,y)f(x)f(y)d\mu(x)d\mu(y) \geq 0$$

Intuitivement, ces propriétés révèlent quelques notions :

- Le noyau w définit une notion de voisinage. C'est-à-dire le voisinage de xi correspond à tous les points x de X qui agissent réciproquement avec y. Numériquement, ce critère est significatif dans le sens où le noyau définit la géométrie locale de X.

- La propriété de positivité permet de re-normaliser w dans un noyau de Markov, et de définir une marche aléatoire sur les données.

La construction d'un processus de diffusion sur le graphe est un sujet classique dans la théorie de graphe spectrale, et la procédure consiste à re-normaliser le noyau $w(x, y)$ comme suit : pour tout $x \in X$,

Soit, $q^2(x) = \int_X w(x,y)d\mu(y)$, alors, le noyau re-normalisé est $\tilde{w}'(x,y) = \frac{w(x,y)}{q^2(x)}$ est un noyau stochastique (avoir une somme de 1 sur chaque ligne de la matrice de similarité)

$$\int_X \tilde{w}'(x,y)d\mu(y) = 1.$$

De plus, pour tout x ,y de X, $\tilde{w}'(x,y) \geq 0$. Comme une conséquence, $\tilde{w}'(x,y)$ peut être interprété comme la matrice de transition d'un processus de Markov homogène sur X. Cette procédure montre qu'on peut associer une marche aléatoire à chaque noyau admissible sur l'ensemble X.

L'opérateur $\tilde{W}'f(x) = \int_X \tilde{w}'(x,y)f(y)d\mu(y)$ correspond à ce noyau qui est défini semi-positif:

si $f \geq 0$ alors $\tilde{W}'f(x) \geq 0$.

Puisque on s'intéresse aux propriétés spectrales de cet opérateur, il est préférable de travailler avec un conjugué symétrique de $\tilde{W}'$: $\tilde{w}'$ est conjugué par q.

$\tilde{w}(x,y) = \frac{w(x,y)}{q(x)q(y)} = q(x)\tilde{w}'(x,y)\frac{1}{q(y)}$ et $\tilde{W}f(x) = \int_X \tilde{w}(x,y)f(y)d\mu(y)$.

En conséquence, le nouveau noyau est conjugué en noyau stochastique, et partage le même spectre, et leur fonction propre est obtenue par la conjugaison par q.

3.4.2 La décomposition spectrale du noyau de diffusion (distance de diffusion)

Après avoir construit la matrice de diffusion, on s'attache maintenant aux propriétés spectrales liées à cette matrice, et plus précisément aux distances de diffusion. Soit l'opérateur de diffusion W avec le noyau symétrique w tel que [Kheir06]:

$W : L2(X) \rightarrow L2(X)$

$Wf(x) = \int_X w(x,x')f(x')d\mu(x')$ est défini semi positif, c-à-d pour tout $f \in L2(X, d\mu)$, on a :

$$\forall f \in L^2(X,d\mu),\ \langle Wf, f \rangle = \iint_{XX} w(x,x')\frac{f(x)}{q(x)}\frac{f(x')}{q(x')}d\mu(x)d\mu(x') \text{ et } \|W\| = 1.$$

Soient $\psi_j \in L_2(X)$ les fonctions propres de W solutions de : $(W\psi_j)(x) = (\lambda_j\psi_j)(x)$ normalisées et orthogonales associées aux valeurs propres $\lambda_j \geq 0$, triées en ordre décroissant. On a alors : $(\lambda_j)_{j\in M} \in l_1$ et $w(x,x') = \sum_{j=1}^{M}\lambda_j\psi_j(x)\psi_j(x')$ est valide pour tout couple (x,x'). Soit w(t)(x,x') le noyau de l'opérateur de diffusion W(t) . Par conséquent, on a :

$$w^{(t)}(x,x') = \sum_{j=1}^{M}\lambda^{(t)}{}_j\,\psi_j(x)\psi_j(x')$$

Au niveau des éléments de X, le noyau $w(t)(x,x')\,d\mu(x')$ a une interprétation probabiliste . En clair, w(t)(x,x') est la probabilité pour un marcheur aléatoire d'atteindre x' partant de x en t étapes.

$p(x^{t} = x' \mid x^{t} = x) = w(t)(x,x)d\mu(x')$

De même, les fonctions propres peuvent être considérées comme des nouvelles coordonnées sur les données. Enfin, une nouvelle représentation des données de l'ensemble X dans l'espace euclidien $l^m(R)$est obtenue en considérant la configuration des fonctions propres suivantes :

$$\Psi_t(x) : X \rightarrow \Re^m$$
$$x \mapsto \Psi^t(x) = (\lambda_1^t\psi_1(x), \lambda_2^t\psi_2(x), \ldots, \lambda_m^t\psi_m(x))\ \ T.$$

Chaque fonction propre est interprétée comme une coordonnée sur le jeu de données. Cette configuration prend ainsi des entités concrètes et fournit une représentation des données comme des points dans un espace euclidien. La question appropriée est : qu'est-ce qui caractérise cette configuration ?

Pour répondre à cette question, soit une famille de métriques {Dt}t>=1 sur l'ensemble X:

$D^2t(x,x') = w(t)(x, x) + w(t)(x', x') - 2w(t)(x, x')$

Sous la contrainte définie semi-positive de ce noyau,

$$D_t^2(x,x') = (1 \ -1) \begin{pmatrix} w^{(t)}(x, x) & w^{(t)}(x, x') \\ w^{(t)}(x, x') & w^{(t)}(x', x') \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

La quantité Dt(x, x') a une interprétation fonctionnelle. D'abord, elle peut être considérée comme une distance de diffusion entre x et x' : elle mesure le taux de connectivité entre les points du jeu de données. Moins il y aura de chemins entre ces deux points, plus le taux sera petit. Plus il y aura de chemins entre ces deux points, plus le taux sera grand.

À la différence de la distance géodésique, la distance de diffusion est robuste au bruit et aux courts-circuits topologiques, parce que c'est une moyenne de tous les chemins connectant deux points (voir la Figure 3.2). Dans cet exemple, le jeu est composé de points pris au hasard sur deux ensemble disjoints et la distance géodésique de A à B n'est pas beaucoup plus grande que celle d'entre B et C. Du point de vue métrique de diffusion, les points B et C sont connectés par plusieurs chemins et sont donc très proches l'un de l'autre. Par contre, les points A et B sont connectés par peu de chemins, laissant ces points très éloignés l'un de l'autre. La distance de diffusion est donc capable de séparer les deux disques.

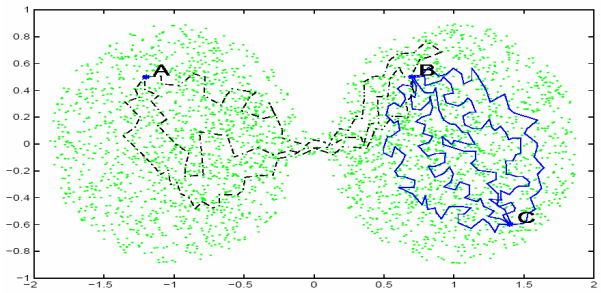


Figure 3.2: Différence entre la distance de diffusion et la distance géodésique

Étant donné la définition des marches aléatoires, nous dénotons la distance de diffusion comme une distance mesurée à l'instant t entre deux sommets, ou la distance euclidienne entre les colonnes d'index x et x' dans la matrice w(t) .

$$D_t^2(x,x') = \int_X \left| w^{(t)}(x,z) - w^{(t)}(x',z) \right|^2 d\mu(z) = \left\| w^{(t)}(x, .) - w^{(t)}(x', .) \right\|^2.$$

Par conséquent, un effet remarquable est que cette quantité complexe peut être simplement mesurée dans l'espace de fixation l2 (N), par la relation:

$$D_t^2(x,x') = \sum_{j \geq 0} \lambda_j^t (\psi_j(x) - \psi_j(x'))^2$$

En d'autres termes, la diffusion métrique peut être calculée comme une distance pondérée euclidienne dans l'espace de fixation, les poids étant λ_t^0, λ_t^1 , etc.

Cette proposition donne une réponse à la question posée précédemment : la configuration Ψ fournit une représentation des données comme les points d'un espace euclidien de telle façon que la distance pondérée dans cet espace est égale à la distance de diffusion des données.

3.4.3 Cas de densité des données non uniformes

La métrique sur X dépend de l'espace ambiant Rn. Nous supposerons que μ a une densité de la mesure de probabilité Riemannienne dx sur X (c-à-d, $d\mu(x) = p(x)dx$). Cette densité p(x) peut être considérée comme la densité des points types du jeu de données. Dans la construction de l'opérateur de diffusion expliqué précédemment, l'information de la géométrie locale est concrétisée par le fait que le noyau w et la distribution des points sur X, donné par $d\mu(x) = p(x)dx$, sont combinés. En revanche, cette combinaison échoue quand la densité est non uniforme. Pour cela, une simple modification de cette normalisation manipule le cas de densité non uniforme. En d'autre terme, il est capable de séparer la distribution des points de la géométrie intrinsèque de X. Pour cela, l'opérateur Laplace-Beltrami est seulement défini par la géométrie. Donc, au lieu d'appliquer la procédure de normalisation au noyau w(x; y), nous pourrions plutôt employer le noyau $\frac{w_\varepsilon(x,y)}{q(x)q(y)}$ pour séparer la géométrie de X de la distribution des points. Dans la pratique, cela suppose que l'on connaît q, ce qui n'est pas souvent le cas. Cependant, cette densité peut être rapprochée (jusqu'à un facteur de multiplication) lorsqu'on couple le noyau à une mesure sur l'ensemble de points X.

$$q_\varepsilon(x) = \int_X w_\varepsilon(x, y)q(y)dy.$$

Dans ce cas, on peut remplacer w_ε par le noyau

$$\tilde{w}_\varepsilon(x,y) = \frac{w_\varepsilon(x,y)}{q_\varepsilon(x)q_\varepsilon(y)}$$

Ainsi, le noyau stochastique $q^2_\varepsilon(x) = \int_X \tilde{w}_\varepsilon(x,y)q(y)dy.$

et l'opérateur $\tilde{W}_\varepsilon$, définie sur L2(X), correspond à ce noyau

$$\tilde{W}_\varepsilon f(x) = \frac{1}{q^2{}_\varepsilon(x)} \int_X \tilde{w}_\varepsilon(x,y)f(y)q(y)dy.$$

3.4.4 Calcul du graphe Laplacien pondéré avec l'opérateur Laplace-Beltrami

Puisque nous sommes dans le cas discret, toutes les intégrales de la mesure empirique q(x)dx des données sont calculées comme des sommes discrètes, c'est-à-dire $q_\varepsilon(x) = \sum_y w_\varepsilon(x,y)$ et

$\tilde{W}_\varepsilon f(x) = \sum_y \tilde{w}_\varepsilon(x,y)f(y)$ où $\tilde{w}(x,y)$ est obtenu par normalisation de w(x, y).

Pour notre problématique, l'ensemble d'origine $X = \{x_1, x_2, ..., x_N\}$ est considéré comme l'ensemble de sommets du graphe pondéré avec w(·,·). Le but étant de représenter chaque $x_i \in R^d$ par un point $y_i \in R^m$ où m<<d, de sorte que l'ensemble $Y = \{y_1, y_2, ..., y_N\}$ capture toute l'information géométrique intrinsèque de l'ensemble d'origine. Pour cela, Belkin et Niyogi [Belkin03] ont montré que l'utilisation de l'opérateur de Laplace Beltrami, constitue un "bon" choix pour la prise en compte de l'information géométrique. En particulier, l'information locale est bien préservée par l'utilisation de ces fonctions propres (eigenmaps).

Lafon, Keller et Coifmann [Lafon06] ont montré que l'utilisation d'un noyau gaussien w_ε produit une représentation des données qui est fortement corrélée à la distribution des échantillons des données. Il est définit comme suit:

$$w_\varepsilon(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|^2 / \varepsilon\right)$$

où ε est un paramètre d'échelle et $\|\ \|$ désigne la distance euclidienne standard.

Dans le tableau 3.5, nous résumons le principe du calcul du Graphe Laplacien Pondéré et l'approximation de l'opérateur Laplace-Beltrami :

- Entrée : $X = \{x_1, x_2, ..., x_N\} \subset R^d, t, \varepsilon, m$
- Sortie : $Y = \{y_1, y_2, ..., y_N\} \subset R^m$
- Construction de la matrice de similarité w_ε en utilisant le noyau gaussien :

$$w_\varepsilon : X \times X \to \Re$$

$$w_\varepsilon(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|^2 / \varepsilon\right)$$

- Première normalisation de w_ε pour la densité : (avec choix d'opérateurs)

$$\tilde{w}_\varepsilon(x_i, x_j) = \frac{w_\varepsilon(x_i, x_j)}{[q_\varepsilon(x_i) q_\varepsilon(x_j)]^\alpha}$$

  $\alpha = 0$: Graphe de Laplacien.
  $\alpha = 0.5$: le propagateur de Fokker-Plank.
  $\alpha = 1$: Opérateur de Laplace-Beltrami

- Re-normalisation pour obtention du noyau de diffusion :

$$p(x_i, x_j) = \tilde{\tilde{w}}_\varepsilon(x_i, x_j) = \frac{\tilde{w}_\varepsilon(x_i, x_j)}{\sqrt{\tilde{q}_\varepsilon(x_i)\tilde{q}_\varepsilon(x_j)}} \text{ Avec } q_\varepsilon(x_i) = \sum_{x_k \in X} w_\varepsilon(x_i, x_k)$$

- Diagonalisation de la matrice P. Le résultat est le spectre du l'opérateur de diffusion tel que

$$p(x_i, x_j) = \sum_{s \geq 0} \lambda_s (\psi_s(x_i) - \psi_s(x_j))^2 \quad with\ 1 = \lambda_0 \geq \lambda_1 \geq ... \geq 0$$

- Espace de diffusion :

$$x \to y = (\lambda_1^t \varphi_1(x), \cdots, \lambda_{m(}^t \varphi_{m(}(x))^T$$

Tableau 3. 5: Calcul du Graphe Laplacien Pondéré et approximation de l'opérateur Laplace-Beltrami

Reprenons notre graphe $G = (V,E)$. Deux sommets v_i et v_j sont adjacents si l'arête. $C(v_i, v_j) \in E$ et les deux nœuds sont alors appelés des nœuds voisins. Le graphe est considéré comme graphe pondéré, de poids $w : V \times V \to R^+$ qui satisfait les conditions suivantes pour chaque sommet $v_i, v_j \in V : w(v_i, v_j) = w(v_j, v_i)$ et $w(v_i, v_j) \geq 0$

La matrice des poids obtenus W est appelée matrice d'affinité ou matrice de similarité. Elle est définie par : $(W)_{ij} = w(v_i, v_j)$ si les sommets v_i et v_j sont adjacents et $w(v_i, v_j) = 0$ dans le cas contraire. Les sommets étant liés à eux-mêmes nous avons pour tout sommet $v_i : w(v_i, v_i) = 1$.

$q(v_i) = \sum_{v_j \in V} w(v_i, v_j)$ est définit comme étant le degré du sommet v_i. On définit la matrice diagonale des degrés des sommets D avec: $(D)_{ii} = D(v_i, v_i) = q(v_i)$ et $D(v_i, v_j) = 0$ pour $v_i \neq v_j$

La première normalisation de la matrice de similarité permet de trouver une représentation indépendante de la distribution.

$\tilde{w}(v_i, v_j) = \tilde{w}_{ij} = \frac{w(v_i, v_j)}{q(v_i) q(v_j)} = \frac{w_{ij}}{q_i q_j}$ dans le cas de l'opérateur Laplace-Beltrami.

Nous nous sommes intéressé à un processus de marche aléatoire (ou de diffusion dans le graphe G). Le temps est discrétisé $t = (0,1,2,...)$. A chaque instant, un marcheur est localisé sur un sommet et se déplace à l'instant suivant vers un sommet choisi aléatoirement et uniformément parmi les sommets voisins. La suite des sommets visités est alors une marche aléatoire, et la probabilité de transition du sommet v_i au sommet v_j est définie à chaque étape par:

$$p_{ij} = p(v_i, v_j) = \frac{\tilde{w}(v_i, v_j)}{\sqrt{\tilde{q}(v_i)\,\tilde{q}(v_j)}}. \qquad (3.5)$$

Avec $\tilde{q}(v_i) = \sum_{v_j \in V} \tilde{w}(v_i, v_j)$

Ceci définit la matrice de transition P, $(P)_{ij} = p(v_i, v_j)$ de la chaîne de Markov correspondant à la marche aléatoire. La matrice P est stochastique, en effet: $\forall v_i, \forall v_j, 0 \leq p(v_i, v_j) \leq 1$ et $\sum_{v_j \in V} p(v_i, v_j) = 1$.

Considérons $p_t(v_i, v_j)$ le noyau correspondant à la t^{ème} puissance de P : P^t. $p_t(v_i, v_j)$ peut être interprété comme la probabilité pour un marcheur d'atteindre le sommet v_j en partant du sommet v_i en t étapes. L'intérêt d'introduire cette matrice de transition est que l'exploration du graphe par la marche aléatoire qu'elle engendre permet de déterminer des propriétés topologiques du graphe [Ingve04], reliées aux propriétés spectrales de P.

P est généralement symétrique et pour chaque colonne la somme des éléments est égale à 1. Cette matrice est intéressante car elle reflète la géométrie intrinsèque des données [Robles04]. Une marche aléatoire correspond à une chaîne de Markov homogène puisque les probabilités de transition restent les mêmes à chaque fois que l'on revient sur un nœud du graphe.

Les chaînes de Markov sont définies en terme d'états et de transitions entre ces derniers. Les états sont dans notre cas les nœuds du graphe. Dans une chaîne de Markov, deux états i et j sont dits communicants si l'on peut atteindre l'un à partir de l'autre avec une probabilité finie; ce qui signifie que le graphe est connexe. Si l'on veut décrire la probabilité de transition $p_t(v_i, v_j)$ d'un nœud v_i à un nœud v_j en t sauts, il suffit de considérer des voisinages plus larges, ce qui correspond à élever la matrice P à la puissance t. Si le graphe est connexe et non bipartite, alors la marche aléatoire converge vers une distribution $\pi = [\pi_1, ...\pi_n]$ satisfaisant $n\pi^t = \pi$ avec $\pi_i = \frac{d_i}{vol(V)}$ avec $Vol(V) = \sum_{i \in V} d_i$.

C'est-à-dire $\lim_{t \to \infty} p_t(v_i, v_j) = \pi_i$

La décomposition spectrale de la matrice de transition P donne un ensemble de valeurs propres $1 = |\lambda_0| \geq |\lambda_1| \geq |\lambda_2| \geq ... \geq 0$ engendrant un ensemble de vecteurs propres $\{\varphi_0, \varphi_1, \varphi_2, ..., \varphi_m\}$, solutions de:

$$P\varphi_m = \lambda_m^t \varphi_m \tag{3.8}$$

Ainsi, on peut définir la famille des distances de diffusion $\{D_t\}_{t \geq 1}$ par:

$$D_t^2(x, y) = \sum_{j \geq 0} \lambda_j^t (\varphi_j(x) - \varphi_j(y))^2 \tag{3.9}$$

où le paramètre d'échelle t contrôle la sensibilité de la distance de diffusion D_t aux valeurs propres φ_j. $D_t(x, y)$ mesure le taux de connectivité entre les données x et y par les chemins de longueur t .

Dans notre travail, nous avons considéré des marches aléatoires de pas fixe et suffisamment court t=1 afin d'atteindre des longueurs suffisamment grandes $\tilde{w}(x,y)_{\epsilon}$ [0 1] pour collecter des informations globales sur le voisinage du point de départ de la marche.

Considérons la transformation suivante $\{\psi_t\}_{t\geq 1}$:

$$\psi_t : R^n \rightarrow R^{m(t)} \qquad (3.10)$$

$$x \rightarrow \psi_t(x) = (\lambda_0^t\varphi_0(x), \lambda_1^t\varphi_1(x), \lambda_2^t\varphi_2(x), \cdots, \lambda_{m(t)}^t\varphi_{m(t)}(x))^T$$

avec φ_0 est un vecteur constant $\varphi_0(x) = (1,1,...,1)$.

Cette transformation est communément utilisée maintenant pour l'analyse et la réduction de dimensionnalité [Lafon06]. Elle permet donc de passer d'un espace de mesure de dimension n à un espace de représentation homogène de dimension $m(t)$ plus réduit représentant toutes les informations ainsi que les propriétés structurales du graphe. $m(t)$ est le nombre pour lequel les valeurs propres $\{\lambda_j^t\}_{j\geq m(t)}$ sont numériquement insignifiants. On a généralement $m(t) \leq 3$. Ces informations sont principalement captées par les premiers vecteurs propres de P^t liés aux plus grandes valeurs propres. La distance de diffusion (1) peut être définie par:

$$D_t^2(x,y) = \left\|\psi_t(x) - \psi_t(y)\right\| \qquad (3.11)$$

Les vecteurs propres de la matrice de transition P^t peuvent être interprétés comme la généralisation des fonctions de Fourrier sur un graphe. Ainsi les valeurs propres de faibles valeurs correspondent aux vecteurs propres de hautes fréquences, et celles de fortes valeurs correspondent aux vecteurs de basses fréquences.

Remarquons que l'on ne considère pas le vecteur propre car il ne porte aucune information. Notons également, que le signe de ϕ_1 permet de catégoriser l'ensemble des données X en deux ou plusieurs classes et que toute distance entre deux points dans l'espace de mesures sera transformée en une distance euclidienne dans l'espace de représentation, facilitant ainsi la comparaison de données et leur fusion.

Ce second vecteur propre ϕ_1 est connu comme le vecteur de Fiedler [Levy06] et peut être utilisé pour ordonner l'ensemble des données X [Higgs06].

Nous exploitons les informations des coordonnées: $(\lambda_1^t\psi_1, \lambda_2^t\psi_2, ..., \lambda_i^t\psi_i)$ pour identifier les regroupements de sommets du graphe. Grâce à ce type de méthodes, nous pouvons détecter

des structures de faibles dimensions dans un espace initial de très grande dimension: ceci est désigné par le terme de manifold learning dans la communauté.

On résume dans le schéma suivant notre démarche de catégorisation:

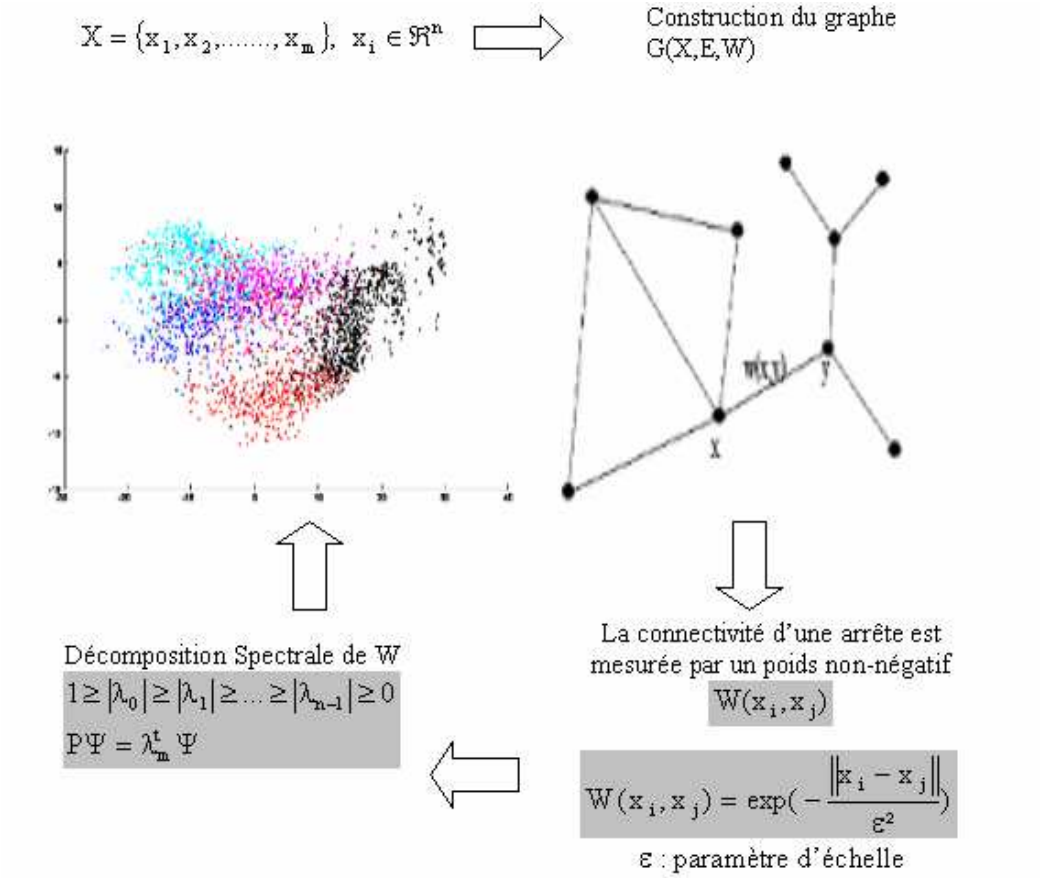


Figure 3.3: Différentes étapes de la catégorisation par marches aléatoires sur graphe

SVD (Singular Value Decomposition) d'une matrice est un outil important de factorisation des matrices rectangulaires réelles ou complexes. Le théorème spectral énonce qu'une matrice normale peut être diagonalisée par une base orthonormée de vecteurs propres.

L'utilisation de la décomposition en valeurs singulières est la représentation explicite de l'image et du noyau d'une matrice. Le calcul explicite, analytique de la décomposition en valeurs singulières d'une matrice est difficile dans le cas général. Nous avons utilisé GNU Scientific Library (GSL) pour calculer le SVD. GNU GSL propose trois alternatives; l'algorithme de Golub-Reinsh, l'algorithme modifié (plus rapide pour les matrices possédant bien plus de lignes que de colonnes) et l'orthogonalisation de Jacobi.

3.4.5 Applications en imagerie vidéo

Après avoir présenté, théoriquement, dans le chapitre précédent les propriétés spectrales des marches aléatoires sur graphe et leurs différences avec les deux autres approches (Isomap et LLE), nous avons testé leurs adéquations pour la révélation de l'information géométrique liées à un ensemble de données, qui est de nature non linéaire.

3.4.5.1 Réduction, Réorganisation et Visualisation de bases d'images

Nous présentons dans cette section deux applications démonstratives qui ont été mise en œuvre. La première est liée à la visualisation des données et l'autre à leur réorganisation. Notre jeu de données dans ces applications est un ensemble d'images de visage [1] ordonnées de gauche à droite. Chaque image a une résolution de 112× 92 pixels, c'est-à-dire un vecteur de données de dimension 10304. L'opérateur de diffusion utilisé est Laplace-Beltrami de pas t=1, avec comme mesure de similarité:

$$w(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / \varepsilon) \tag{3.12}$$

où $\|\ \|$ désigne la métrique euclidienne standard et ε le paramètre d'échelle. Nous construisons un graphe complet et à partir de la décomposition spectrale de P nous utilisons les coordonnées $(\lambda_1\psi_1, \lambda_2\psi_2)$ pour la visualisation 2D. Dans nos expériences, le paramètre ε a été fixé comme suit:

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N} \min\left\{\|x_i - x_j\|^2 / \|x_i - x_j\|^2 > 0, j = 1,2,...N\right. \tag{3.13}$$

L'ensemble des images a été traité comme si elles étaient désordonnées pour voir la capacité qu'ont ces trois méthodes pour paramétrer ce nuage de points.

[1] http://images.ee.umist.ac.uk/danny/database.html

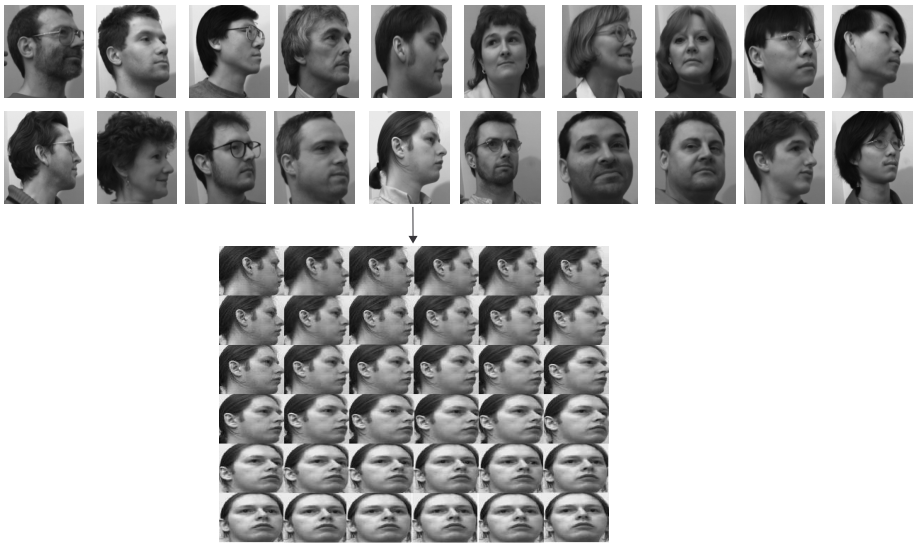


Figure 3.4: Corpus d'images ordonnées de visages (profils)

Figure 3.5: Corpus d'images de visages ordonnées de gauche et l'ensemble désordonnée à droite.

(a) *(b)*

Figure 3.6: Corpus d'images de visages ordonnées de gauche et l'ensemble désordonnée à droite.

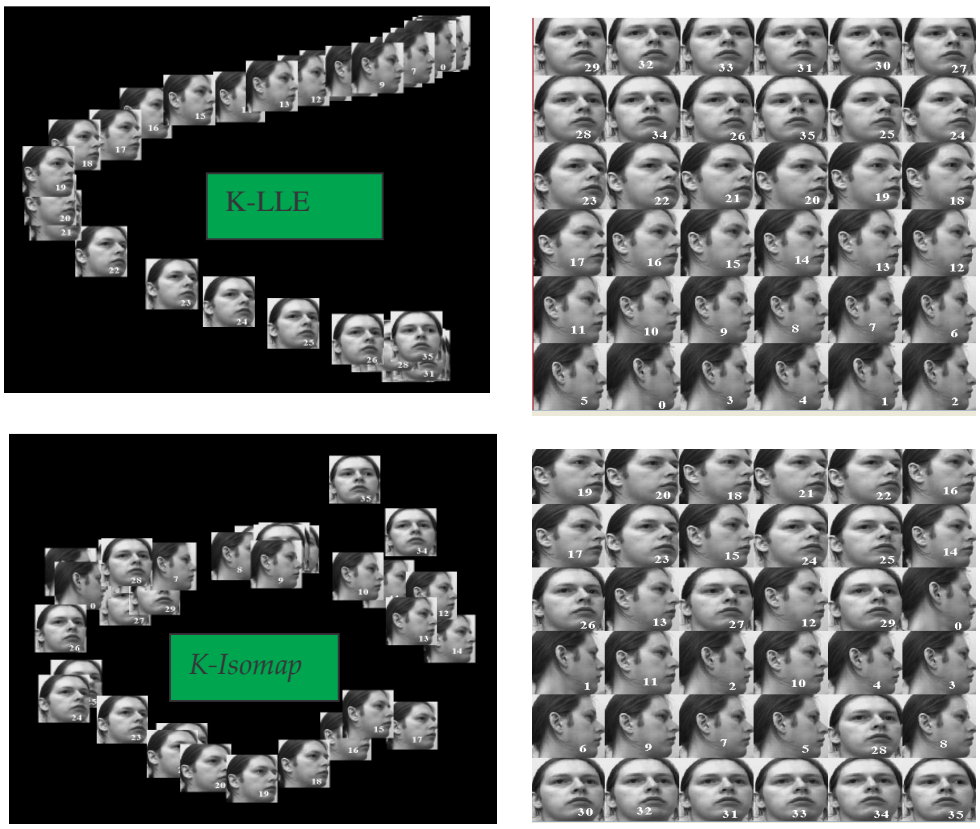


Figure 3.7: Les méthodes de k-Isomap et k-LLE. (a) La visualisation des images selon les axes ψ_1 et ψ_2 (b) La réorganisation : le tri du premier vecteur propre ψ_1

On reconnaît dans la figure 3.7 l'efficacité du processus de diffusion par marches aléatoires pour la révélation géométrique de l'ordre des visages et dans le sens de rotation de gauche à

droite en formant une courbe à deux limites. Contrairement aux k-Isomap et k-LLE, le processus de diffusion permet de mieux évaluer la justesse de capture des petites variations d'intensité entre chaque paire de visage.

Le tri du premier vecteur propre forme une courbe qui ressemble à une courbe de cosinus entre [0, π/2]. Cette localisation permet d'exposer les trente-six images de visage sur la carte de diffusion aux cordonnées correspondant à la position du visage. C'est-à-dire qu'elle permet de récupérer l'organisation des données suivant l'angle de rotation du visage qui est entre [0 π/2] (figure 3.8).

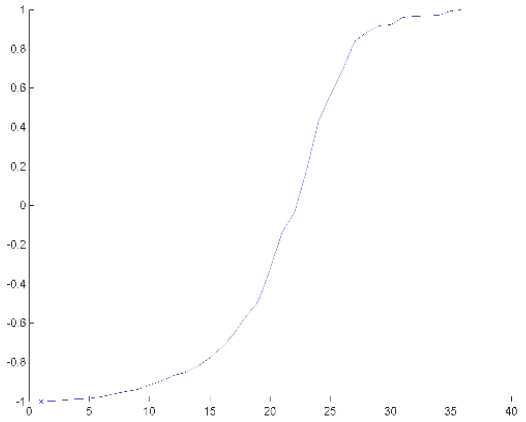


Figure 3.8: La courbe correspondante à ψ_1

En conséquence, avec cette précision de localisation, les marches aléatoires sur graphe ont tendance à visualiser les différentes positions des images de n'importe quel type. La figure 3.9 montre une réorganisation de la base d'images en fonction de la position de la tête Il s'agit de la position de départ sur la courbe correspondante à ψ_1.

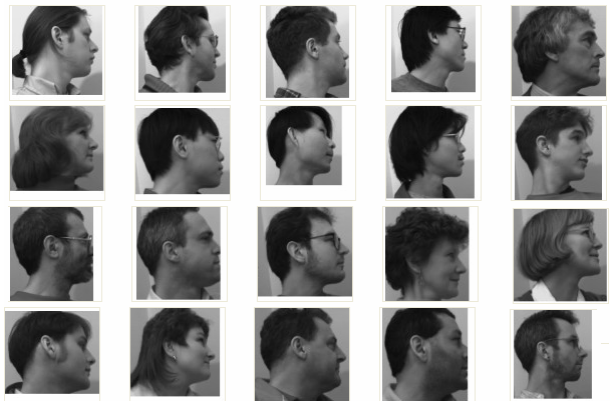
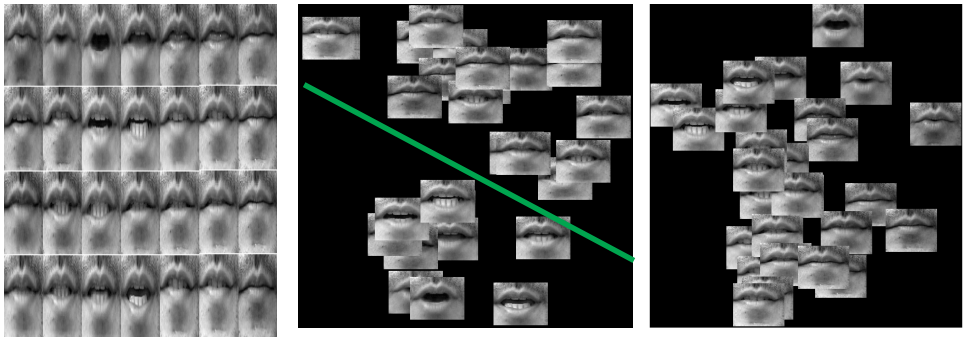


Figure 3.9: Réorganisation des positions des images (images de départ)

Dans la figure 3.10, nous avons visualisé un ensemble de bouches de différentes formes (ouvertes/fermées), où la matrice de diffusion est formée avec des descripteurs unidimensionnels formés de l'intensité des images (vecteur données de 10304 dimension), et multidimensionnel en combinant cette intensité avec les informations du contour et le gradient (vecteur données de 10304*3 dimension).



(a) (b) (c)

Figure 3.10: Marches aléatoires sur graphe sur un ensemble de bouche. (a) Les groupement des bouches selon la position fermée/ouverte avec un descripteur unidimensionnel (intensité). (b) Le groupement des bouches avec les différentes formes avec un descripteur multidimensionnel (intensité+gradient).

Après avoir vu l'adéquation des marches aléatoires sur graphe pour la révélation de l'information géométrique intrinsèque aux jeux des données, nous espérons aller au-delà pour appliquer notamment la diffusion géométrique sur graphe à la segmentation de composantes faciales et aussi la catégorisation de vidéos.

3.4.5.2 Caractérisation visuelle de la texture

Dans la littérature, on trouve différents descripteurs qui consistent à calculer un certain nombre de paramètres mathématiques caractéristiques de la texture, permettant de quantifier les différents niveaux de gris présents dans une image en terme d'intensité et de distribution, notamment les filtres de Gabor, les descripteurs à base ondelettes, la matrice de co-occurrence, et les champs de Markov aléatoires. Chahir et al. [Chahir07a] ont proposé un descripteur de texture qui repose sur la caractérisation des interactions entre pixels dans un voisinage local, basées sur les marches aléatoires locales sur graphe. Ils ont montré son utilité pour des applications de détection d'objet, et de caractérisation de texture. L'idée consiste à transformer l'image en un graphe dont les sommets sont des pixels, ou des motifs autour de pixels ou tout vecteur caractérisant la texture localement. Les arrêtes sont alors définies en terme de mesures de similarité entre les propriétés locales des pixels ou des motifs.

A chaque pixel est associé un patch (carré, rectangle ligne, rectangle colonne) à partir duquel sera calculé le descripteur.

Soit : $X = \{x_1, x_2, \ldots, x_m\}$,tel que : $x_i \in \Re^n$. Nous considérons un voisinage centré au pixel xi , de taille w × w, noté $V_{W\times W}(x_i)$ tel que :

$$\begin{cases} V_{W\times W}(x_i) = \{ x_{i1}, x_{i2}, \ldots, x_{ik} \}. & V_{W'\times W'}(x_{ij}) = \{ x_{ij,1}, x_{ij,2}, \ldots, x_{ij,k,\ldots} \} \\ \quad et & avec \\ \forall x_{ij} \in V_{W\times W}(x_i),\ \exists\ V_{W'\times W'}(x_{ij}) & V_{W'\times W'}(x_{ij}) : \text{Patch de formes différentes} \end{cases}$$

Ce nouveau descripteur est basé sur l'exploration du graphe par analyse spectrale. Il permet de capturer les interactions locales et non locales entre pixels ou motifs composant une texture. Cette méthode a été exploitée pour la caractérisation des composantes faciales et les actions faciales sur les différents types d'image (intensité, couleur). L'idée de base de ce descripteur est d'analyser les valeurs propres formées directement à partir de fenêtres de niveaux de gris ou de couleur locaux. Les valeurs singulières $\lambda_1, \lambda_2, \ldots, \lambda_n$ fournissent l'information sur la dépendance entre les lignes et les colonnes de la matrice markovienne locale correspondant à chaque pixel, et reflète donc la connectivité entre les sommets du graphe. Soit une image $X = \{x_1, x_2, \ldots, x_n\}$. Pour chaque $V_{W\times W}(x_i)$

– Construction et décomposition de la matrice de transition de Markov de chaque fenêtre locale de x_i . Le résultat est les fonctions propres de la matrice de transition :

$$\begin{cases} 1 = \lambda_0 \geq \lambda_1 \geq \ldots \geq 0 \\ \quad et \\ \{\psi_j\}_{j=i,w} \text{ tel que } \quad P\psi_j = \lambda_j \psi_j \end{cases}$$

– Remplacer x_i par la somme des valeurs singulière : $\Gamma(x_i) = \sum_{h=r}^{w} \lambda_h$

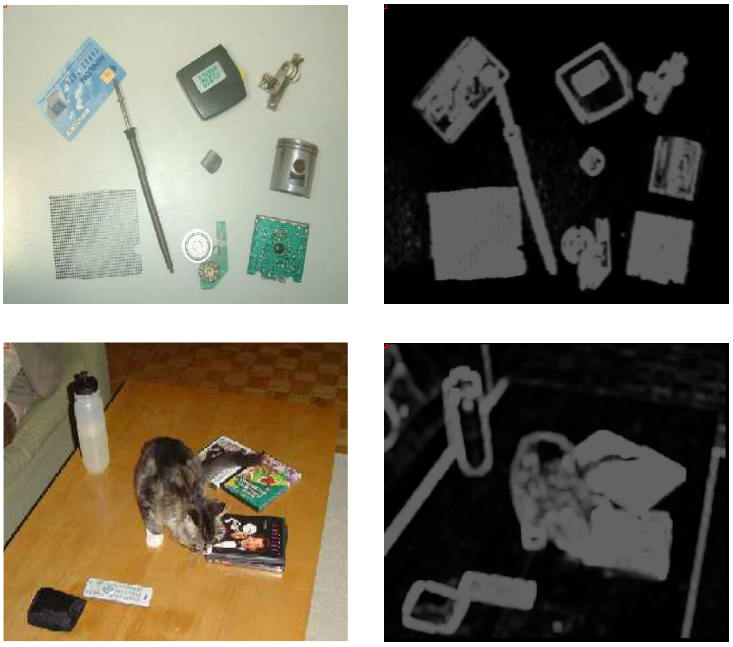
Les valeurs singulières $\lambda_1, \lambda_2, \ldots, \lambda_n$ fournissent l'information sur la dépendance entre les lignes et les colonnes de la matrice markovienne locale correspondant à chaque pixel, et reflète donc la connectivité entre les sommets du graphe. Le descripteur peut être calculé sans passer par la décomposition spectrale de la matrice de transition P , ce qui offre, d'un point de vue pratique, un avantage en temps de calcul. En effet : $\Gamma(x_i) = \sum_{h=r}^{w} \lambda_h = \sum_{i=1}^{W} P(i,i)$

Le tableau suivant résume les principales étapes de l'algorithme:

Soit une image $X = \{x_1, x_2, \ldots, x_n\}$
Pour chaque $V_{W\times W}(x_i)$
– Construction du noyau de diffusion P.
– Remplacer x_i par la trace de la matrice de diffusion:

Tableau 3-6: Algorithme de caractérisation par diffusion géométrique locale

La figure 3.11 montre les résultats de la caractérisation sur des images réelles contenant des objets divers et variés. Pour ce faire, nous avons utilisé un patch 5x5 avec une fenêtre composée de 5 lignes autour du point.



(a) *(b)*

Figure 3.12 :(a)Images originales. (b) Caractérisation locale $\Gamma_{5\times5,5\times5}$.

Nous pouvons donc voir que le descripteur Γ sera faible s'il y a peu de variation lumineuse entre les pixels des lignes ou les colonnes qui composent la fenêtre w×w(ex : bouteille qui est sur la table). Cependant, cette variation produit des trous qui sont définis comme des régions du fond, et qui ne sont entourés que d'une seule autre région. Donc, ces trous peuvent être agglomérés dans la région qui les entourent.

3.5 Conclusion et perspectives

Contrairement à la transformation linéaire (eigen-transform [Tavakoli06]), l'estimation des attributs locaux est liée à un voisinage de quelques pixels (ex :11*11), ce qui domine en général la structure réelle des objets, même à une petite échelle d'image. Généralement, lors de la segmentation d'une image en régions homogènes au sens de la texture, on ne cherche pas à regrouper les pixels semblables en niveau de gris, mais plutôt celles ayant un agencement spatial sembable (patch). Nous avons utilisé aussi cet opérateur pour la caractérisation des composantes faciales [Chahir07b] (voir chapitre suivant).

Une texture pourra être perçue de différentes façons en fonction des variations des descripteurs présents sur l'image. C'est à dire on peut calculer, à l'aide des valeurs des pixels (intensité) de chaque rectangle, un certain nombre de paramètres mathématiques

caractéristiques de la texture. La transformée locale présentée ici met en evidence certaines textures. On peut enrichir le vecteur caractéristique d'autres paramètres les plus souvent utilisés pour caractériser une texture tels que la moyenne, la variance, le "skewness" et le "kurtosis" . L'approche de caractérisation loale permet l'adaptabilité avec la nature des descripteurs et permet aussi l'extraction d'éléments saillants dans une image.

Pour extraire la structure des objets saillants dans l'image, on effectue le traitement suivant :

- Caractérisation locale par marches aléatoires
- Classification des pixels (k-means, Fuzzy clustering, etc.) ou le seuillage ;
- Bouchage de trous dans les régions. Un trou est défini comme une région du fond (label=0) qui n'est entourée que d'une seule autre région.
- Extraction des objets .

La figure suivante montre des exemples d'extraction d'objets par caractérisation visuelle locale par marches aléatoires sur graphe.

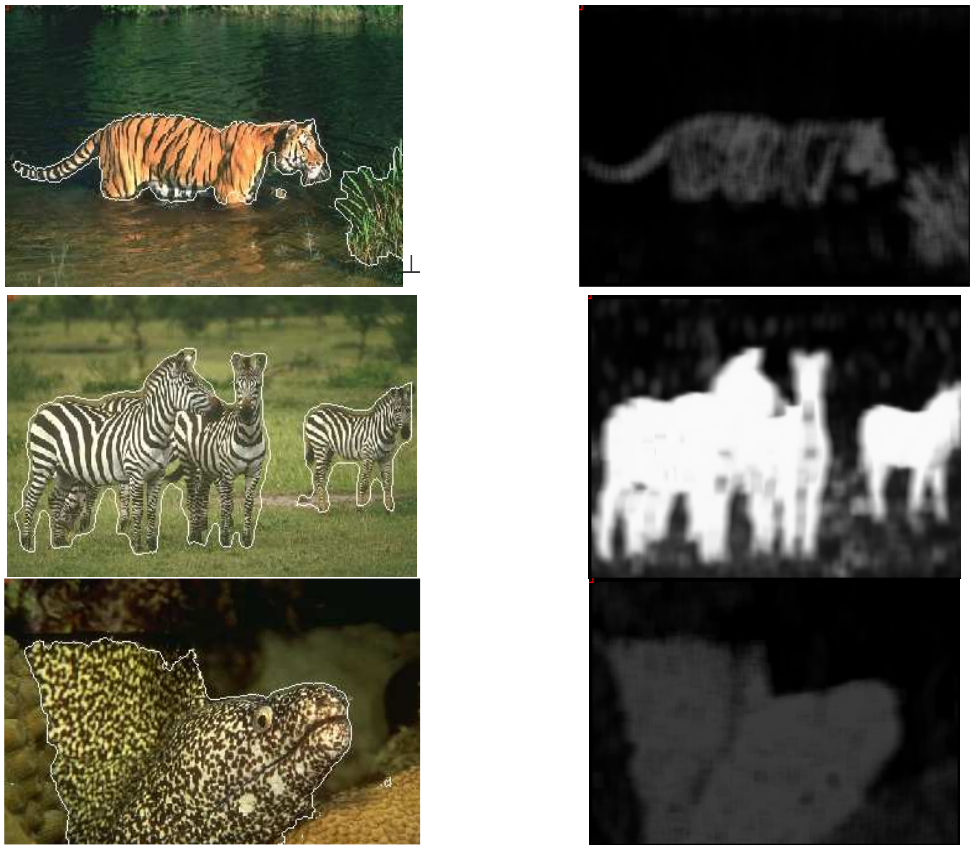


Figure 3 .13 : (a) Résultat de segmentation par classification des pixels en deux classes, (b) Transformée de markov.

CHAPITRE 4 Analyse et catégorisation des expressions faciales

4 Analyse et catégorisation des expressions faciales

4.1 Introduction

L'analyse automatique des expressions faciales constitue un outil important pour la recherche dans les domaines des sciences de l'étude du comportement et de la psychologie, ainsi que dans les domaines de la compression d'images et de l'animation de visages synthétiques [Hoch94]. Le visage est un élément prépondérant dans l'analyse du comportement humain de par sa richesse en information sociale. Les attributs du visage jouent le rôle le plus important dans le processus de communication. Ils fournissent des informations de l'état émotionnel d'une personne. Selon Albert Mehrabian, psychologue américain [Mehrabian68], dans les situations de communication 55% du message est exprimé par l'expression faciale, 38% par le message verbal prononcé et à 7% par le sens des mots. Avant de procéder à l'analyse d'une expression faciale sur une image ou sur une séquence vidéo, il convient de détecter ou de suivre le visage observé puis d'en extraire les informations pertinentes pour l'analyse de l'expression faciale. L'extraction des composantes faciales consiste généralement à détecter la présence [1] et les caractéristiques des composantes [2].

Mercier, Ekman et Friesen [Mercier06] ont établi qu'il existe un nombre limité d'expressions reconnues par tous, indépendamment de la culture .La terminologie utilisée généralement consiste à décrire les expressions faciales à partir des informations telles que la position et la géométrie des actions faciales (déformations faciales des yeux et de la bouche). L'intensité des actions faciales peut être mesurée en déterminant la déformation géométrique des structures faciales ou la densité des rides apparaissant dans certaines régions du visage telle que les yeux et la bouche. En outre, des expériences psychologiques ont montré que les contours des yeux, des sourcils et des lèvres, sont des informations qui doivent être prises en compte dans le processus de catégorisation des expressions faciales. Comme tout autre comportement humain, reconnaître une expression faciale est une tâche complexe à accomplir par un système de vision par ordinateur à cause de la grande variabilité entre les individus. De nombreux travaux en reconnaissance et interprétation des expressions faciales et gestuelles ont tenté de répondre à deux questions récurrentes importantes :

- Quels sont les indices pertinents qui doivent être extraits d'un visage ?
- Comment le comportement de ces indices peut être modélisé et traduit pour la catégorisation des expressions faciales?

Un système de reconnaissance et de classification des expressions faciales nécessite la résolution de trois problèmes :

- la détection et la localisation des visages dans la séquence d'images,

---

1 certaines composantes peuvent être cachées

2 le nombre de composantes et leurs caractéristiques respectives sont dépendants du domaine d'application

- la détection et la localisation des composantes faciales,
- l'extraction de caractéristiques à partir de ces composantes faciales
- et la classification des expressions faciales.

Dans ce chapitre, nous présentons notre travail d'analyse, de visualisation et de catégorisation des expressions faciales. Dans la première partie, nous présenterons les principaux concepts liés à l'expression faciale ainsi que les principaux formalismes de leurs descriptions. Dans une seconde partie, nous évoquerons en détail deux méthodes dédiées à la détection et la segmentation des structures faciales. Dans la troisième partie, nous décrirons dans un premier temps l'algorithme utilisé pour la catégorisation des expressions faciales et dans un second temps, l'application de cet algorithme sur l'ensemble des vecteurs d'informations issus des composantes faciales extraites.

4.2 Formalisme et descriptions

Dans notre travail, nous avons opté pour une approche géométrique où les caractéristiques du visage sont détectées directement à partir de ses composantes. Ensuite, des données issues de mesures relatives à certains points de ces composantes, permettent de définir une représentation graphique du visage. Le choix de l'ensemble des données caractéristiques est crucial pour la suite de l'analyse. La figure 4.1 illustre une schéma général utilisé en reconnaissance et catégorisation des expressions du visage.

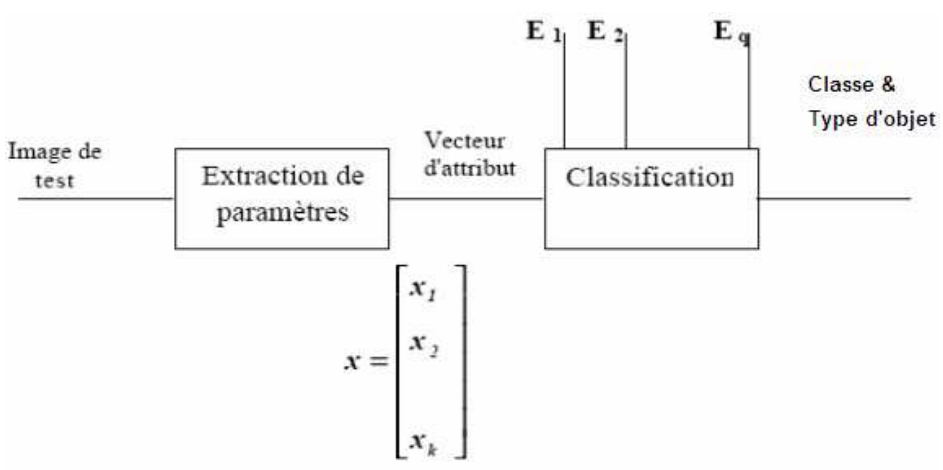


Figure 4.1 : Schéma général de catégorisation et de reconnaissance

La description des expressions du visage est un ensemble d'interprétations successives d'indices visuels. Un sens (dépendant du domaine d'application) est construit à partir de mesures de bas niveaux (présence ou non d'une composante, position éventuelle). Ces mesures sont combinées successivement spatiallement, puis de manière temporelle pour former le sens attendu (émotion sous-jacente par exemple). On introduit ici un vocabulaire nécessaire à la description:

a) **Attribut facial** : un attribut facial est une propriété élémentaire «centrée objet» caractérisant un visage. La position des yeux est un attribut facial. La présence de barbe est un autre attribut. Les attributs faciaux directement visibles sont dit de premier ordre. Les attributs qui ne peuvent être mesurés qu'à partir d'autres attributs de premier ordre, sont des attributs de second ordre et ainsi de suite [Hammal06].

b) **Indice visuel** : un indice visuel est une propriété élémentaire «centrée observateur» du visage : c'est un attribut facial qui est observé et visible. Certains attributs ne sont pas visibles chez certaines personnes (barbe, moustache, sourcils; certains ne sont visibles qu'à certains moments (un œil peut être caché lors d'une rotation de la tête par exemple).

c) **Composante faciale** : une composante faciale est une partie du visage. Le découpage en composantes est celui du langage naturel : les yeux, le nez, la bouche, les joues, les sourcils, la barbe, etc. Bien que certaines puissent être entièrement caractérisées par un ensemble d'attributs faciaux (les yeux peuvent être caractérisés par leur forme, leur couleur, la présence et la longueur des cils, etc.), d'autres ne sont que des mesures floues et sont difficiles à caractériser à partir d'indices visuels, objectifs et élémentaires. C'est le cas par exemple des joues dont les limites sont difficiles à fixer, même pour un observateur humain. Cependant, ces mesures sont importantes puisqu'un humain a plus de facilités à manipuler des données floues que des données précises (les joues sont gonflées).

d) **Action faciale** : une action faciale est un ensemble d'indices visuels intégrés de manière temporelle. Le relèvement des sourcils est par exemple une action faciale composée d'un ensemble de positions successives des sourcils. Une action faciale est généralement décrite par sa dynamique: le relèvement des sourcils consiste en une position actuelle des sourcils plus haute que sa position précédente. Les actions faciales sont généralement caractérisées par leur «profil temporel»: durée d'attaque, durée de maintien et durée de relâchement.

   En 1978, Ekman et Friesen présentent un système de codification manuelle des expressions du visage, composé de 46 Actions Unitaires (Action Unit , qui correspond aux actions faciales définies plus haut) qui décrivent les mouvements élémentaires des muscles. N'importe quelle mimique observée peut donc être représentée sous le forme d'une combinaison d'Actions Unitaires. Ce système de codage est connu sous le nom de «Facial Action Coding System» (FACS). FACS s'est imposé depuis comme un outil puissant de description des mimiques du visage, utilisé par de nombreux psychologues.

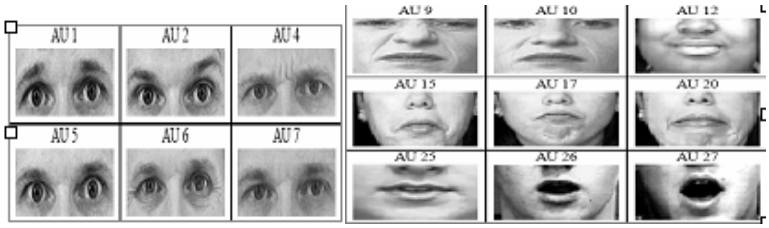


Figure 4.2: Exemples d'actions faciales (Action Units) du haut/bas du visage.

4.3 Analyse du visage

4.3.1 Introduction

Le cerveau humain effectue deux types d'analyses du visage: l'une dite globale où le visage est traité comme un tout et l'autre dite par composantes où le visage est vu comme un ensemble de composantes faciales (yeux, nez, bouche, etc.). Le processus de détection d'un visage consiste à isoler une zone qui ressemble à un visage générique. L'approche globale semble être la plus naturelle au problème de détection du visage, bien qu'il soit tout à fait possible de détecter un visage par une approche plus locale, en détectant le clignement des yeux par exemple. A l'inverse, reconnaître une expression fait appel généralement à une analyse locale (par composantes). En effet, il semblerait que le modèle utilisé par les humains pour reconnaître une expression puisse se résumer à une indication sur la forme des composantes faciales. Ainsi un ensemble restreint de caractéristiques de chaque composante faciale suffit pour qu'un humain reconnaisse une expression. Bien que l'approche globale semble plus adaptée à la détection et la reconnaissance du visage et l'approche par composantes plus adaptée à l'analyse des expressions, les méthodes utilisées dans la pratique sont généralement une combinaison des deux approches. Les systèmes holistiques traitent l'image comme un tout, et en définissent certaines caractéristiques. Ils tirent de l'image une représentation sous-dimensionnée. Dans la littérature de nombreuses approches ont été proposées en analyse du visage, qu'on peut classer en 3 classes :

Approches structurelles à base de connaissances: Cette méthode s'intéresse au visage et ses composantes comme la bouche, le nez et les yeux. Par exemple, la composante faciale comme les yeux sont symétriques entre eux. Ces informations connues sont utilisées dans un processus qui permet de localiser les composantes faciales nez, bouche, yeux, et puis d'analyser les relations spatiales entre elles à partir de certaines règles prédéfinies. Une phase de vérification est nécessaire pour éliminer les fausses détections. Les inconvénients de cette approche résident dans la difficulté de traduire les connaissances humaines afin de tenir en compte de tous les cas possibles qui permettent de parvenir à une détection correcte du visage. Il est difficile aussi de prolonger cette approche à des situations où le visage a différentes postures car il est difficile d'énumérer tous les cas.

Approches basées sur des caractéristiques visuelles invariables: L'idée principale de cette approche est de chercher des caractéristiques invariantes pour la détection du visage quelque soit les conditions d'éclairage et de posture du visage dans des séquences d'images tels que la texture, la couleur chair , niveau de gris, etc….. L'information de texture nous permet de localiser le visage et ses composantes faciales puisque chaque zone du corps humain à une texture différente de celle du visage. Dans [Krieg02], on peut trouver un état de l'art très exhaustif sur les différentes méthodes de détection des visages.

Approches basées sur les modèles d'apparence: Elles utilisent des techniques d'apprentissage pour trouver des caractéristiques discriminantes du visage et du non visage. Ces techniques donnent de bons résultats lorsque la personne est de face.

4.3.2 Carte de composantes faciales

Afin de distinguer les axes de couleurs les plus discriminants, nous avons testé les différentes combinaisons d'axes provenant de différents espaces de couleurs et nous avons exploité des outils de Datamining qui permettent de dégager des règles de décision pour classer un pixel en pixel valide pour une composante faciale ou non valide, par exemple de peau ou non-peau [Hammamil05]. Finalement, la segmentation des régions de peau dans la séquence d'images et leur analyse spatiale et géométrique permet de détecter et de localiser chacune des composantes faciales. Notre démarche se résume donc comme suit:

- Construction d'un modèle visuel des composantes faciales
- Utilisation de la fouille de donnés (Datamining)
- Filtrage des régions de peau dans l'image.
- Localisation des composantes faciales

Il est admis que le choix de l'espace de couleur utilisé dans le cadre d'un traitement d'images est crucial car il influe directement sur les résultats. Cependant, ce choix est rendu difficile par la multitude d'espaces existant comme le montre ce chapitre. Le choix de l'espace de couleur dépend de l'image à analyser d'une part et du choix de l'algorithme utilisé d'autre part. C'est pourquoi, on propose d'analyser l'image à partir d'un espace couleur hybride qui regroupe des composantes couleur pouvant être issues de différents espaces.

– Masque du visage

Lorsque les images d'entrées sont en couleurs, il est avantageux d'utiliser cette information pour isoler les régions susceptibles de contenir des visages. En effet, plusieurs auteurs ont développé des règles de décision dans divers espace de couleur RGB, YcbCr, HSV et TSL. Peer et al [Peer99], Ahlves et al [Ahlvers] et Jones et al. [Jones02] ont défini des règles resp. sur l'espace RGB, YcbCr/HSV et TSL qui ont permis de localiser les régions du visage. La figure 4.3 illustre le processus général d'extraction du visage. En général, l'espace HSV présente un avantage pour la détection des couleurs qui réside dans le fait qu'un des canaux

représente la luminance (Value V). Cette particularité permet d'exprimer adéquatement les couleurs sans se soucier des variations de luminosité. Ainsi, les pixels peau seront extraits en observant seulement la teinte (Hue H) et la saturation (S) des pixels.

Figure 4.3: Comparaisons des résultats de différents règles d'espaces couleurs

On peut remarquer des fausses détections pour certaines zones des cheveux, mais aussi d'autres régions non détectées qu'on peut expliquer par les réfléchissement de la lumière sur certaines régions du visage qui sont ainsi plus éclairées que d'autre régions du visage ce qui change la couleur des régions. Ainsi, le canal S est amoindri alors que le canal V est élevée, la couleur des pixels de ces régions tendent vers le blanc, et sort de l'intervalle de détection.

La détection des composantes du visage peut se faire à partir des résultats de l'extraction de la peau. Une des méthodes envisageable consiste à analyser les trous contenus dans la zone de peau, comme par exemple ceux générés par les yeux. La validation peut alors être réalisée en respectant différentes règles de positionnement (des trous entre eux et par rapport au visage) et de taille.

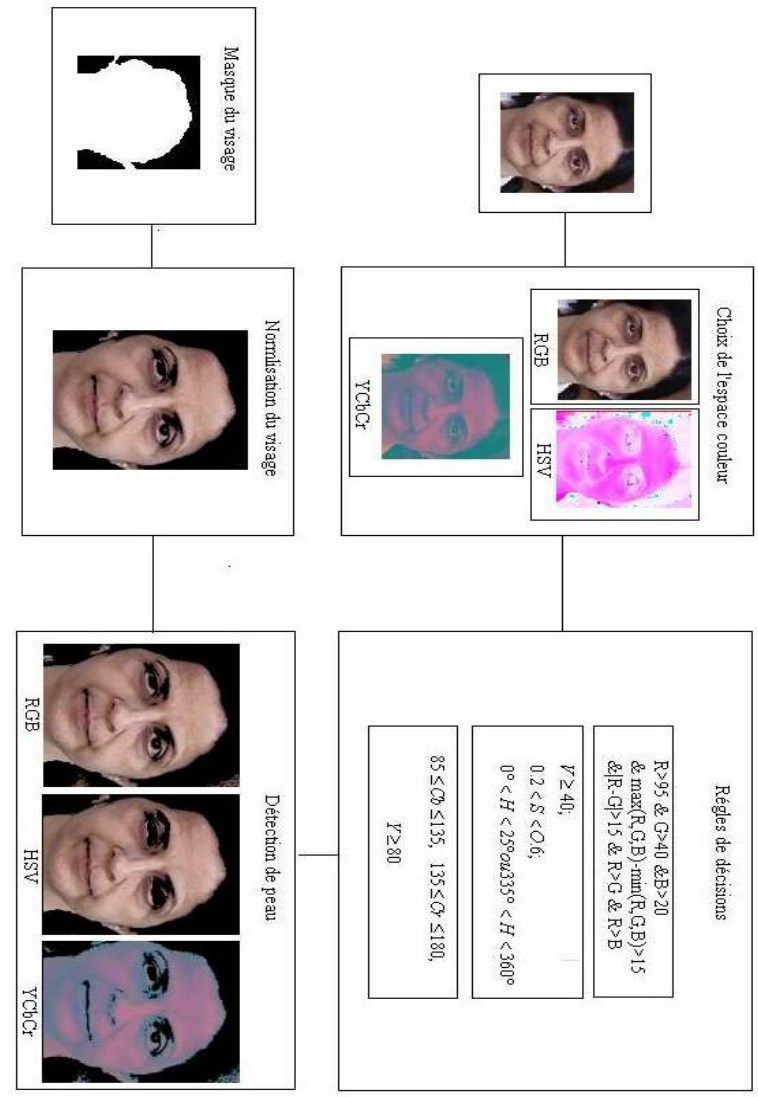


Figure 4.4: Processus de détection d'un visage dans les espaces de couleur RGB, YUV et HSV.

– Détection des lèvres

Pour détecter la bouche, on s'est basé sur la rougeur de la zone des lèvres. Après un lissage intensif des composantes Cr et Cb, elles sont normalisées. La méthode consiste donc à évaluer une fonction de la couleur, prenant des fortes valeurs sur la composante rouge et de faible valeurs sur la composante bleue. Selon la fonction ci dessus, on obtient l'image MouthMap.

$$\mathrm{MouthMap} = \mathrm{Cr}^2 \cdot (\mathrm{Cr}^2 - \eta \cdot \mathrm{Cr}/\mathrm{Cb})^2$$

$$\text{où } \eta = 0.95 * \frac{\frac{1}{n}\sum C_r^2}{\frac{1}{n}\sum (C_r / C_b)}$$

La structure suivante est utilisée, pour recréer l'effet décrit dans la formule ci-dessus.

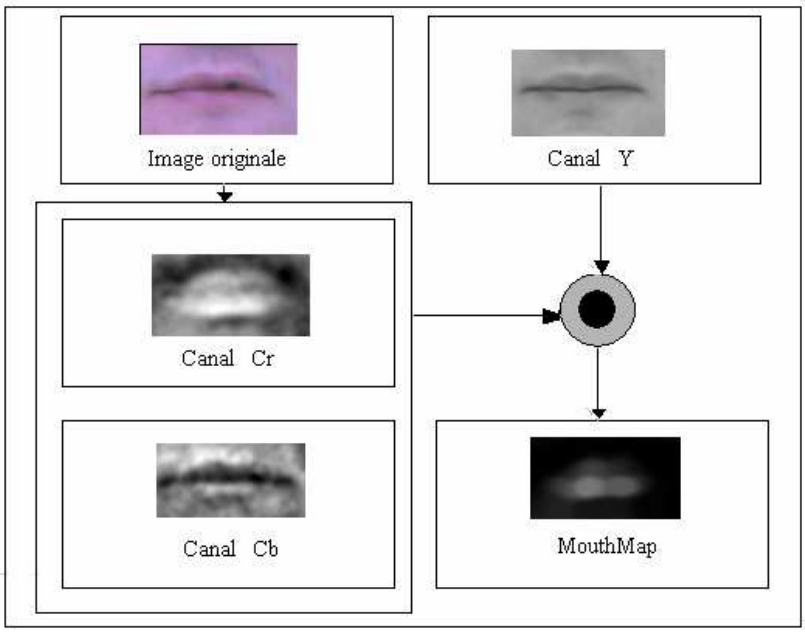


Figure 4.5: Modèle de détection des lèvres

Nous procédons ensuite à un seuillage de MouthMap en conservant de 1 à 10% des pixels avec les valeurs les plus élevés. La figure 4.6 montre quelques résultats de la localisation des lèvres.

Figure 4.6: Les résultats de la localisation des lèvres Avec MouthMap

– Détection du nez

La détection des trous de nez est inspirée de la détection des yeux présenté précédemment. Elle exploite des propriétés de luminance de Y d'une image de nez (YcrCb), et des propriétés

de luminance /couleur de V de la même image de nez dans l'espace HSV. Ces deux images sont lissées et normalisés. Ainsi NoseMap est calculée selon la formule suivante:

$$\text{MoseMap} = (255 - V)^3 * (255 - Y)^3 \quad \text{normalisé } [0..255]$$

La structure suivante est utilisée, pour recréer l'effet décrit dans la formule ci-dessus.

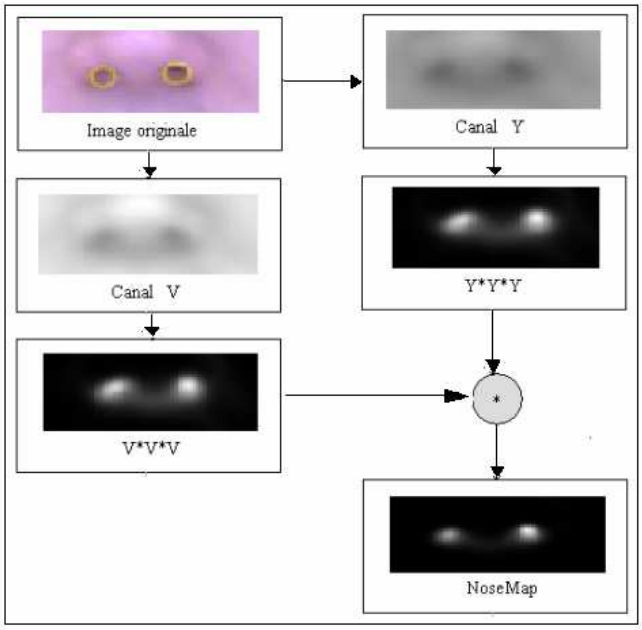


Figure 4.7: Modèle de détection du nez

– Détection des yeux

Les régions des yeux et de la bouche sont définis en effectuant des opérations dans l'espace de couleur YCbCr. Ces opérations produisent une image au niveau de gris avec des forts contraste dans les régions des traits du visages.

La chrominance de la carte des yeux EyeMap C est calculée aussi après un certain nombre d'étapes de prétraitement. D'abord on lisse le canal de Cr et le canal de Cb tout à fait intensivement, pour éliminer le bruit. Particulièrement le canal bleu (Cb) a beaucoup de bruit dans l'image originale (les sondes de CMOS produisent beaucoup de bruit dans le domaine bleu de couleur). Ensuite, les deux canaux sont normalisés, avec des valeurs séparées.

L'analyse des composantes de chrominance indique, qu' au tour des contours des yeux, on a des fortes valeurs de Cb et de faible valeurs Cr. Afin de détecter les yeux on a utilisé une fonction de la couleur pour chaque pixels de l'image.

$$E_{chrom} = \text{EyeMapC} = \frac{1}{3}\left\{(C_b^2 + \overline{Cr}^2 + (C_b / C_r)\right\}$$

$\overline{Cr}$ est l'inverse de Cr (i.e. 255-Cr). Les pixels dont $Cr=0$ ne sont pas calculés. Le résultat E_{chrom} est normalisé entre [0,255].

La structure suivante est utilisée, pour recréer l'effet décrit dans la formule ci-dessus.

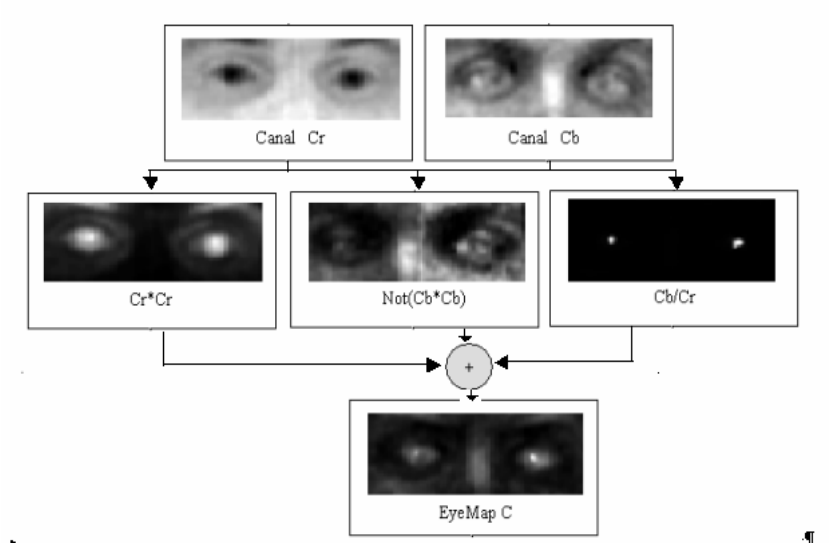


Figure 4.8: Modèle de détection des yeux

L'étape de normalisation n'est pas nécessaire après le calcul de C_b^2 et $\overline{Cr}^2$ à cause de la normalisation des valeurs initiales et pendant de l'étape de prétraitement. Le résultat pour C_b/C_r permet d'assurer que toutes les valeurs soient dans une limite légale. La figure 4.9 illustre quelques résultats de EyeMap C.

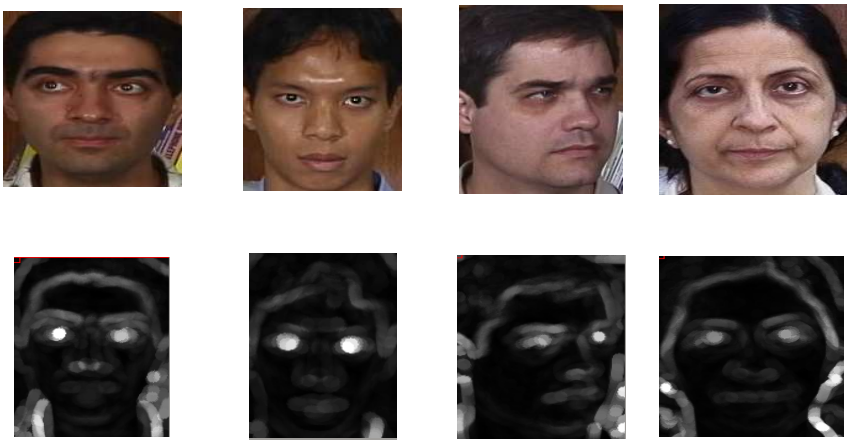


Figure 4.9: Les résultats Avec EyeMap C.

La chrominance (Cr, Cb) et l'information de la luminance (Y) peuvent être exploités pour localiser les deux régions des yeux. Les résultats de nos testes montre que la zone autour des yeux a des valeurs colorimétriques spécifiques. Le but de cette étape est d'accentuer les valeurs de pixel les plus lumineux des yeux, en utilisant les canaux Cb de Cr de chrominance et de la luminance (Y).

La Luminance de la carte des yeux (Luminance Eyemap) Avant de calculer EyeMapL, il faut au préalable un certain nombre d'étapes de prétraitement sont appliquées. Ces étapes de prétraitement sont nécessaires pour compenser la qualité plus ou moins parfaite des images de webcam. D'abord l'image est lissée pour enlever le bruit. En second lieu l'image est normalisée, pour éviter les phénomènes de reflet et d'ombrage.

Après ces étapes de prétraitement, on met en évidence les zones de luminance des yeux par la formule suivante :

$$E_{lum} = \text{EyeMapL} = \frac{Y \oplus g_\sigma}{Y \otimes g_\sigma + 255}$$

où la $\oplus$ et $\otimes$ sont des opérations de dilatation et d'érosion sur une fonction f avec un élément structurant circulaire g_σ. La structure suivante est utilisée, pour recréer l'effet décrit dans la formule ci-dessus.

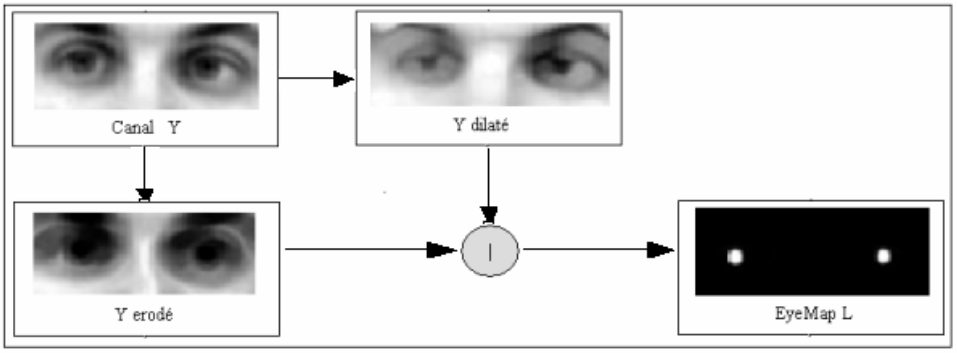


Figure 4.10: Modèle de détection des yeux

La détection des yeux se base sur deux images, l'image de la luminance Y, dans la quelle on a autour des yeux des zones d'ombres et des zones très lumineuses. La deuxième est obtenu à partir des composantes de chrominance.

Un exemple du canal de Y est dilaté en utilisant un élément structurant de taille 5. Un deuxième exemple du canal de Y est érodé, également en utilisant le même élément structurant . L'image EyeMap L est obtenue par la division des deux images une image de luminance Y dilaté, et une image de luminance érodée.

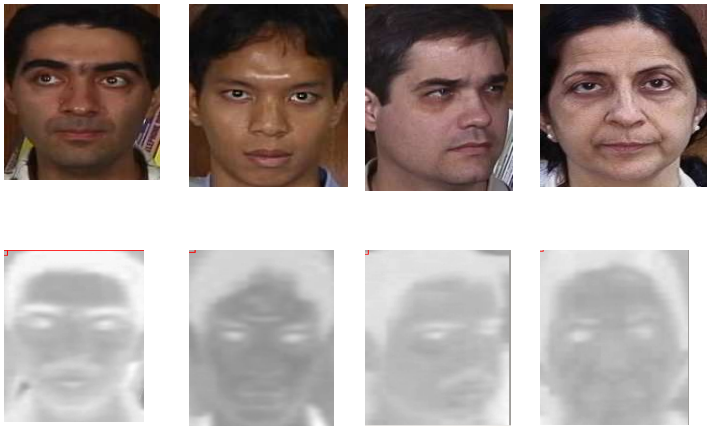


Figure 4.11: Les résultats de EyeMap L

L'image EyeMapL est multipliée par une autre image qui exploite les informations de chrominance autour des yeux, pour localiser les régions des yeux. En multipliant les deux images EyeMapC et EyeMapL, on obtiendra donc une image ou les yeux auront un fort contraste. Une première version de EyeMap peut être créé en multipliant EyeMapL par EyeMapC. Enfin quelques étapes de post-traitement sont faites pour augmenter le résultat final.

$$E_{map1} = \text{EyeMap}_1 = \text{EyeMapL} * \text{EyeMapC}$$

Les yeux contient des zones lumineuses (blanches) et foncées (l'iris). Pour séparer la forme ou les bords de ces deux régions on utilise l'espace RGB. La première étape consiste à utiliser le filtre de sobel pour extraire les contours. Le résultat obtenu est ensuite lissé par un filtre moyenneur. Ceci peut être écrit comme suit :

$$E_{edge} = P(\left|P(\frac{(R+G+B)}{3}, S_h)\right| + \left|P(\frac{(R+G+B)}{3}, S_v)\right|, A)$$

Avec P(X,A) qui représente le filtre spatial de X par A. Sh and Sv est le filtre de sobel horizontal resp. vertical et A le filtre moyenneur.

Une 4ème carte des yeux (Eyemap) est construite avec le constat suivant qui est que les yeux sont toujours des zones sombres dans les trois composantes R,G et B. RGB Eyemap est calculé en multipliant les inverses des composantes R,G et B. Ensuite le résultat est filtré avec le même filtre présenté précédemment. Ainsi :

$$E_{RGB} = P((\overline{R} \times \overline{G} \times \overline{B}), A)$$

Carte finale des yeux :

Un simple Eyemap ne peut pas localiser la position des yeux. D'où la proposition d'étendre la construction de Eyemap aussi à partir de E_{edge} et de E_{RGB} . Il suffit donc de combiner les quatre selon la formule suivantes:

$$E_{map2} = E_{map1} \times E_{edge} \times E_{RGB}$$

Une multiplication est proposée plutôt qu'une addition car elle est plus déterministe et permet de supprimer les faux positifs à partir de Eyemap simple E_{map1} .Cependant, la multiplication accentue également les faux négatifs et les expériences montrent qu'elles restent largement moins que les faux positifs. Les différentes cartes et leur produit final sont illustrées dans la figure 4.12.

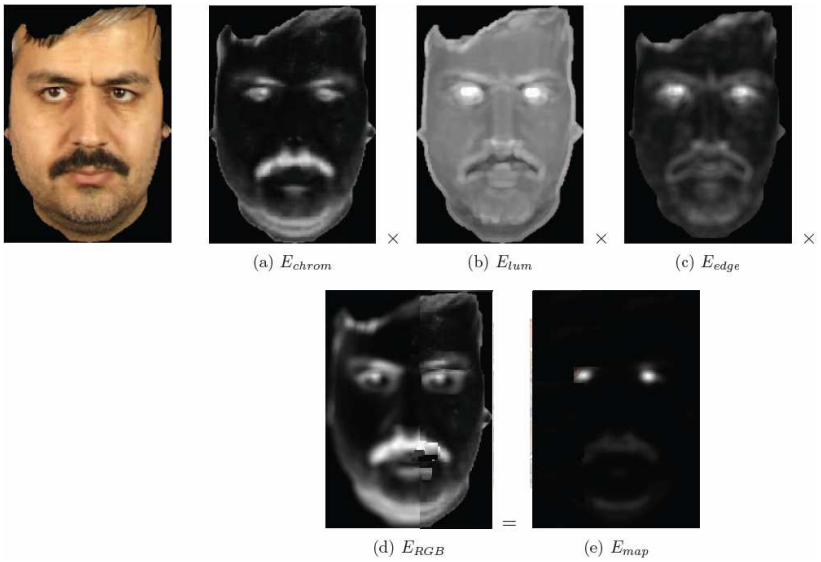


Figure 4.12: Les résultats de EyeMap L [Louw07]

Une fois le visage, les lèvres et les yeux détectés, les composantes faciales de chaque trame de la vidéo peuvent alors être projetées dans l'espace de visage obtenu par l'approche spectrale.

– Position des yeux

On considère que les yeux dans l'image E_{map} sont bien alignés verticalement. Sinon, un étape de calcul de l'axe de rotation est nécessaire.

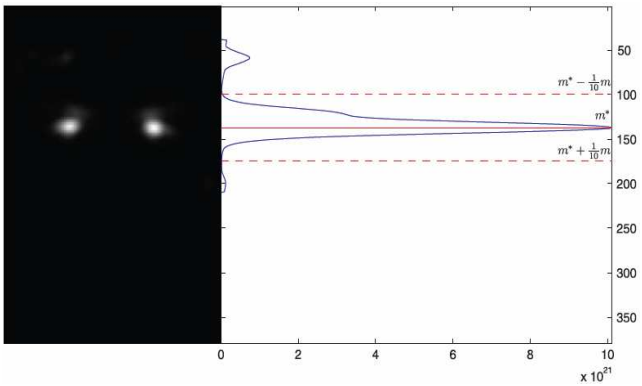


Figure 4.13: Position des yeux [Louw07]

4.3.3 Extraction des actions faciales

L'approche principale reposent essentiellement sur la segmentation par classification de pixels et l'extraction des composantes connexes (composantes faciales) par une méthode de croissance de régions. Les points caractéristiques correspondant aux extrémités des rectangles englobants de chaque composante faciale, sont ensuite extraits. Afin de segmenter et d'extraire les composantes faciales (zones d'intérêts), une chaîne de traitement a été mise au point fondé sur les étapes suivantes[Chahir07b]:

- Normalisation de l'image afin d'atténuer les changements d'apparence du visage qui sont causés par la position de la tête ou les variations d'éclairage.
- Caractérisation des composantes faciales par :
    - Analyse spectrale
    - EueMap/MouthMap
- Classification et segmentation des composantes : elle sert à séparer les structures détectées les plus contrastés. La segmentation sert à étiqueter les structures détectées.
- Extraction des points caractéristiques de chaque composante faciale.
- Utilisation d'opérateurs de morphologie mathématique pour éliminer le bruit et fusionner les structures proches et voisines.
- Les actions faciales en question seront les deux composantes de taille plus grande.

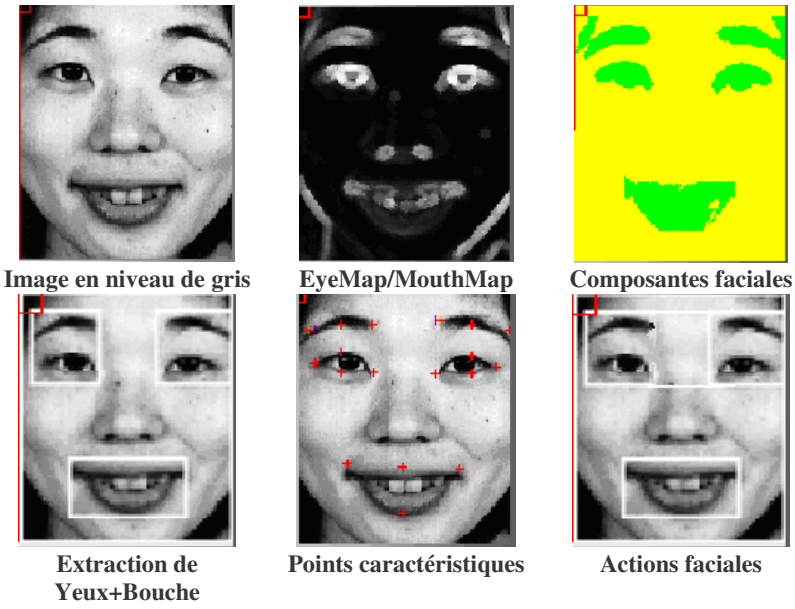


Figure 4.14: Processus d'extraction d'actions faciales

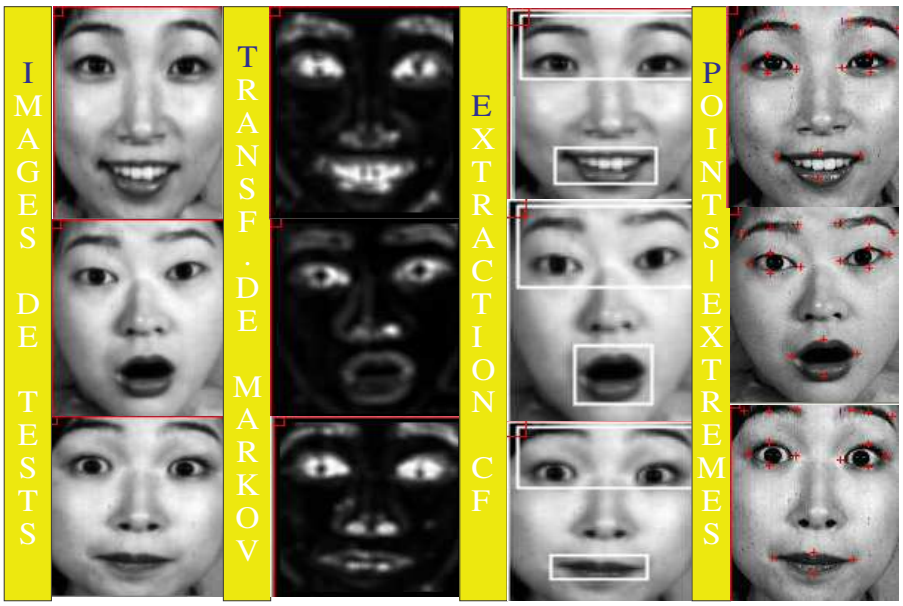


Figure 4.15: Processus d'extraction d'actions faciales par l'intermédiaire de l'analyse spectrale

4.4 Expressions faciales

Dans le passé, les travaux de recherche sur l'analyse des expressions faciales se situaient principalement dans le cadre de la psychologie. Les progrès effectués dans des domaines connexes tels que la détection, le suivi et la reconnaissance de visages [Zhao00] ainsi que dans le traitement d'images et la reconnaissance de formes, ont apporté une contribution

significative à la recherche dans le domaine de l'analyse, de la synthèse et de la reconnaissance d'expressions faciales.

En comparaison de la reconnaissance faciale, il y a moins de travaux sur la reconnaissance des expressions faciales. Ils se divisent en deux catégories suivant qu'on travaille sur des séquences d'images ou sur des images statiques.

a) **Extraction des informations. // Segmentation des composantes faciales :**

Les caractéristiques les plus importantes utilisées pour l'identification des visages humains semblent être [Dai98]: la région des yeux et des sourcils, la bouche et la lèvre inférieure, le nez, le menton. Turk and Pentland [Turk91] représentent les images faciales par un ensemble de composants faciaux standardisés (les "eigenfaces") et effectuent une analyse linéaire des composants principaux. Padgett et Cottrell [Padgett97] utilisent une approche similaire, utilisant les "eigenfaces" sur certaines régions du visage (les deux yeux et la bouche). De leur côté, Lanitis et al. [Lanitis97] tiennent compte à la fois des formes et des niveaux de gris en utilisant des descripteurs déformables paramétriques et en effectuant une analyse statistique sur un ensemble d'entraînement d'images faciales. Black et Yacoob [Black97] utilisent des modèles locaux paramétriques pour représenter le mouvement des visages. Ils estiment le mouvement relatif des traits caractéristiques dans le repère du visage. Les paramètres de ce mouvement servent par la suite à représenter l'expression faciale.

De manière identique, Cohn et al. [Cohn98] utilise un algorithme hiérarchique pour effectuer le suivi des traits caractéristiques par estimation du flux optique.Les vecteurs de déplacement représentent l'information sur les changements d'expression faciale. De même, Padgett et Cottrell [Padgett98] utilisent des gabarits d'oeil et de bouche, calculés par Analyse en Composantes Principales d'un ensemble d'apprentissage, en conjonction avec des réseaux de neurones. D'autre part, Hong et al. [Hong98] utilise un modèle global basé sur des graphes étiquetés construits à partir de points de repère distribués sur le visage. Les nœuds de ces graphes sont formés par des vecteurs dont chaque élément est la réponse à un filtrage de Gabor extraite en un point donné de l'image. Certains [Lien98] utilisent des modèles de Markov cachés, ou automate à état caché de Markov, un procédé utilisé généralement en reconnaissance de formes ou en traitement automatique des langues. D'autres utilisent [Bartlett98] l'analyse en composantes principales et indépendantes ou les réseaux de neurones. Cootes et al. [Cootes01] utilise une représentation par modèle actif d'apparence (AAM) pour extraire automatiquement des paramètres caractérisant le visage. D'autres [Zhan04] des filtres se basant sur des ondelettes de Gabor, permettant de construire des modèles pour les expressions faciales.

b) **Reconnaissance d'expressions**

Après avoir détecté le visage et extrait les informations pertinentes, l'étape suivante consiste à identifier l'expression faciale affichée. Pour classer l'expression faciale dans l'une des six catégories de base en plus de la catégorie neutre, Hong et al. [Hong98] part du principe que deux personnes qui se ressemblent affichent la même expression faciale de manière similaire. Un graphe étiqueté est attribué à l'image de test puis la personne connue la plus proche est déterminée à l'aide d'une méthode de mise en correspondance de graphes élastiques. La galerie personnalisée de cette personne est alors utilisée pour reconnaître l'expression faciale de l'image de test. Un graphe étiqueté par des réponses de filtres de Gabor est par ailleurs utilisé par Lyons et al. [Lyons98] et Bartlett et al. [Bartlett03]. L'ensemble des graphes construits sur un ensemble d'apprentissage est ensuite soumis à une ACP puis analysé à l'aide d'une analyse discriminante linéaire (ADL) afin de séparer les vecteurs dans des classes ayant des attributs faciaux différents. Le graphe étiqueté de l'image testée sera alors projeté sur les vecteurs discriminants de chaque classe afin de déterminer son éventuelle appartenance à cette classe.

Essa et Pentland [Essa] extraient des gabarits spatiotemporels de l'énergie du mouvement du visage pour chaque expression faciale. Le critère de similarité repose sur la distance Euclidienne entre ces gabarits et l'énergie du mouvement de l'image observée. Heisele, Ho et Poggio [Heisele01] utilisent des machines à vecteurs de support (SVM) dans le cadre de la reconnaissance de visages par des méthodes globales ainsi que par des méthodes reposant sur des traits caractéristiques. De manière identique, l'algorithme de reconnaissance de visages FaceIt est basé sur une technique d'analyse locale des traits caractéristiques (LFA) développée par Penev et Atick [Penev96]. Draper et al. [Draper] compare les performances de l'analyse en composantes principales et de l'analyse en composantes indépendantes (ICA) pour la reconnaissance de visages et d'expressions faciales en se basant sur le système de codage FACS [Ekman]. Par contre, Yang [Yang02] utilise une représentation par noyaux (KPCA) pour la reconnaissance de visages. Finalement, Edwards, Cootes et Taylor [Edwards98] utilisent le modèle actif d'apparence pour reconnaître l'identité d'un individu observé de manière robuste par rapport à l'expression faciale ainsi que l'illumination et la pose. Pour ceci, le critère de similarité utilisé repose sur la distance de Mahalanobis, et une ADL est appliquée afin de maximiser la séparation des classes.

c) **Synthèse d'expressions**

La synthèse d'expressions faciales est une tâche difficile compte tenu de la complexité de la forme et de la texture des visages. De plus le visage présente un grand nombre de rides et de plis ainsi que des variations subtiles de forme et de texture qui ont une importance cruciale dans la compréhension et la représentation des expressions faciales. Dans cette perspective, les techniques d'interpolation offrent une approche intuitive pour l'animation de visages. Pighin et al. [Pighin96] utilise des techniques de morphing 2D combinées avec des

transformations d'un modèle géométrique 3D, pour créer des modèles faciaux réalistes tridimensionnels à partir de photographies, et pour construire des transitions lisses entre les différentes expressions faciales. Dans la même optique, Chen et al. [Chen95] applique le morphing au cas 3D. En outre, dans le cadre du "Video-Rewrite", Bregler et al. [Bregler97] utilise des techniques de suivi de points 2D sur la bouche d'un orateur dans une séquence d'apprentissage et des techniques de morphing pour combiner ces mouvements dans une vidéo finale montrant une personne différente prononçant les même paroles. Dans une finalité analogue, Ezzat et al. [Ezzat02] utilisent une représentation par modèle déformable multidimensonnel et une technique de synthèse de trajectoire pour modifier les mouvements de la bouche d'un visage parlant. Cette représentation est capable de synthétiser des configurations inconnues de lèvres parlantes à partir d'une séquence initiale, en utilisant des techniques de morphing. Chuang et al. [Chuang]] utilise une ACP combinée à un modèle de factorisation bilinéaire pour synthétiser une nouvelle expression sur un visage parlant. Kang et al. [Kang98] utilise le modèle actif d'apparence en conjonction avec des techniques de régression linéaire pour annuler l'expression faciale d'un visage dans le but d'améliorer les performances de la technique de reconnaissance de visages par AAM décrite dans [Edwards98].

4.4.1 Catégorisation des composantes/actions faciales

Etant donné la matrice d'affinité de notre ensemble de données, nous allons nous intéresser à un processus de marche aléatoire dans le graphe G construit. Un marcheur est localisé sur un sommet et se déplace vers un sommet choisi aléatoirement et uniformément parmi les sommets voisins. La suite des sommets visités est alors une marche aléatoire, et la probabilité de transition du sommet i au sommet j est à chaque étape: $P_{ij} = \frac{W_{ij}}{d(i)}$

(où W est la matrice d'affinité du graphe G et d(i) le degré du sommet i).

Ceci définit la matrice de transition P de la chaîne de Markov correspondante. Ensuite, on s'attache aux propriétés spectrales liées à ce graphe, et plus précisément aux intérêts mathématiques des vecteur propres. La décomposition spectrale de la matrice P donne un ensemble de valeurs propres.

$1 = \|\lambda_0\| \geq \|\lambda_1\| \geq ..... \geq \|\lambda_k\| = 0$, engendrent elles même un ensemble de vecteurs propres, soient: $\psi_j \in L_2(V)$ solutions de: $P\psi_j = \lambda_j \psi_j$

Après avoir présenté, les propriétés spectrales des marches aléatoires sur graphe dans le chapitre précédant , nous présentons dans cette section deux démonstrations de catégorisation basées sur les vecteurs d'intensités et les vecteurs de distances. Nous avons testé notre

approche sur la base de visage JAFFE3 (résolution : 140*115) (figure 4.16) répartie en sept classes d'expressions faciales universelles.

Figure 4.16: Base de visage JAFFE

Après avoir obtenues les composantes faciales sous forme de vecteurs d'intensités (EyeMap/Spectral) et de distances (points caractéristiques), on s'intéresse ici à trouver une projection de ces vecteurs, qui sont distribues dans un espace de grande dimension , dans un espace réduit 2D.

– Similarité basée sur les vecteurs d'intensité:

Motivé par l'interprétation du paragraphe (4.2.4) (indépendance des muscles), nous avons analysé seulement deux zones (actions faciales): la zone des yeux et celle de la bouche. En se basant uniquement sur le vecteur de caractérisation (intensité) de la zone bouche, deux catégories d'expressions faciales se mettent en évidence : «Sourire» et «Surprise» (voir figure 4.17).

Ce résultat s'explique par la différence d'intensité qui existe entre les deux formes correspondant à la composante bouche ouverte. En revanche, le reste des images avec des bouches fermées est dispersé dans l'ensemble des classes selon la forme des contours de la bouche.

De même, la projection de la composante faciale des yeux (figure 4.18), montre une organisation selon la forme, et cela est du au fait que la déformation entre l'œil et le sourcil est ici estimée par une simple différence d'intensité. Il s'avère important de prendre en compte la variation de distance entre les deux mimiques faciales sourcils et yeux, et de lier cette

[3] JAFFE : http://www.kasrl.org/jaffe.html

distance à la déformation de la bouche. D'où la nécessité de prendre en compte des distances issues de la l'information des points caractéristiques de chaque composante faciale.

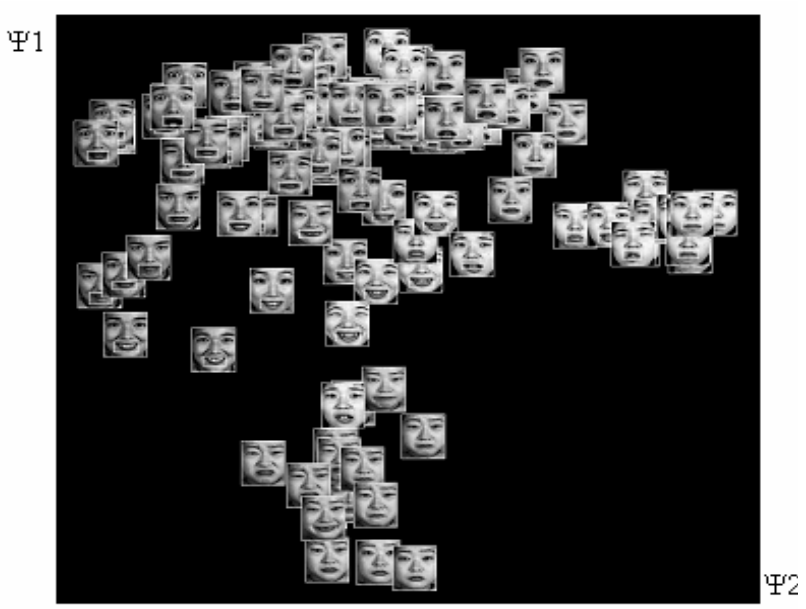


Figure 4.17: La projection des deux premiers vecteurs propres dans l'espace 2D selon les contours de la bouche

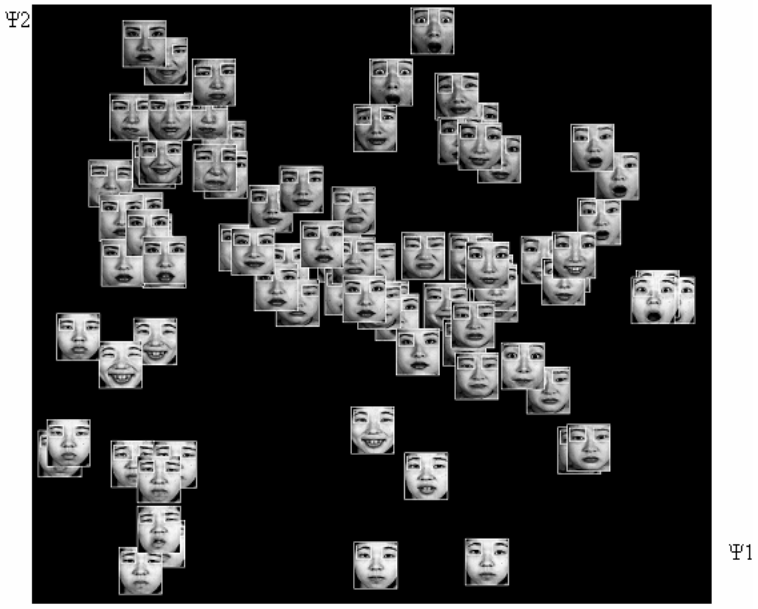


Figure 4-1: La projection des deux premiers vecteurs propres dans l'espace 2D selon la forme des yeux.

– Similarité basée sur les distances MPEG4

La norme MPEG-4 fournit une description des transformations subies par chacun des traits du visage lors de la production de chacune des six émotions universelles. Cette description est la suivante [Hammal06]:

– Joie : la bouche s'ouvre, les commissures se retirent en arrière en direction des oreilles, les sourcils sont décontractés ;

– Tristesse : les coins intérieurs des sourcils se courbent vers le haut, les yeux se ferment légèrement, la bouche est décontractée;

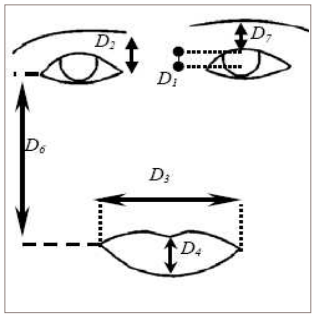
– Colère : les coins intérieurs des sourcils s'abaissent ensembles, les yeux s'ouvrent largement, les lèvres se serrent l'une contre l'autre ou bien elles s'ouvrent pour laisser apparaître les dents;

– Peur : les sourcils se lèvent ensemble et leur partie intérieure est courbée vers le haut, les yeux sont contractés et en état d'alerte;

– Dégoût : la lèvre supérieure se lève et se courbe souvent de manière asymétrique, les sourcils et les paupières sont décontractées;

– Surprise: les sourcils se lèvent, les paupières supérieures s'ouvrent, la bouche s'ouvre, les paupières inférieures sont relâchées;

Afin de traduire numériquement toutes ces descriptions, un ensemble de distances particulières sur chaque squelette a été défini : la figure 4.19 décrit toutes les distances Di considérée. D2 et D7 donnent une mesure de la distance entre les yeux et les sourcils. D6 mesure la distance entre les yeux et la bouche. D3 et D4 mesurent le degré d'ouverture de la bouche, D1 mesure le degré d'ouverture des yeux.



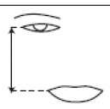
D1		Distance entre la paupière supérieure et inférieure
D2		Distance entre le coin intérieur de l'œil et le celui du sourcil
D3		Largeur de bouche.
D4		Ouverture de bouche en hauteur
D5		Distance entre un coin de la bouche et celui du coin extérieur de l'œil

Figure 4.18: Définition des distances Di

La liste des critères associées à chaque description d'émotions :

– Joie: {D4 augmente}, {D3 diminue ET D6 diminue}, {les autres distances restent constantes}

– Tristesse: {D2 augmente ET D7 diminue}, {D1 diminue}, {les autres distances restent constantes}

– Colère : {D2 diminue}, {D1 augmente}, {D4 diminue ou D4 augmente}

– Peur : {D2 augmente ET D7 augmente mais plus que D2}, {?}

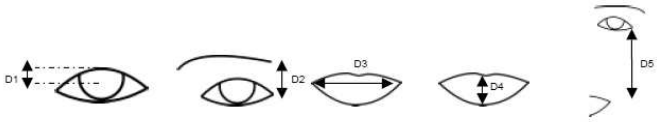
– Dégoût : {D3 augmente ET D4 augmente}, {les autres distances restent constantes}

– Surprise : {D2 augmente}, {D1 augmente}, {D4 augmente}, {les autres distances restent constantes}.

À partir de ces distances caractéristiques cités en haut, on va ajouter des états symboliques où on associe l'un des trois états symboliques suivants à chacune des valeurs de distances.

Dans leur travaux [Hammal05] Hammal et al. ont proposé de définir des distances caractéristiques et ont établi une base de règles pour la modélisation des expressions faciales. Chaque expression faciale est caractérisée par une combinaison d'états symboliques.

La table extraite de leur travaux est présentée ci-dessous (cf. tableau 4.1).



Sourire	C-	S / C-	C +	C+	C -
Surprise	C+	C+	C -	C+	C+
Dégoût	C -	C -	S / C+	C +	S / C -
Colère	C +	C -	S	S / C -	S
Tristesse	C -	C +	S	S	S
Peur	S / C+	S / C+	S/C-	S/C+	S/C+
Neutre	S	S	S	S	S

Tableau 4-1: Etats symbolique associés à chaque expression

– Etat C+ pour lequel la distance Di est plus grande que celle pour l'expression neutre.

– Etat S pour lequel la distance Di est de même ordre de grandeur que celle pour l'expression neutre.

– Etat C- pour lequel la distance Di est plus petite que celle pour l'expression neutre.

En considérant qu'une expression faciale est capturée par l'ensemble de ces distances, nous avons calculé la similarité entre une expression donnée et une expression neutre. En se basant sur ces états symboliques, nous avons construit une matrice de similarité appropriée où les lignes représentent les expressions neutres de chaque visage et les colonnes représentent l'ensemble de visages qui vont vérifier les états décrits dans le tableau ci-dessus. Par conséquent, le nouveau calcul de la matrice de similarité se fera de la manière suivante :

Le nouveau calcul de la matrice de similarité :

– $w(x,y) = \exp\left(-\frac{\|x-y\|^2}{\varepsilon}\right) \quad x \in \Omega_1, y \in \Omega_2$

Ω_1 : ensemble des composantes faciales correspondantes aux visages d'expression neutre,

Ω_2 : ensemble des composantes faciales correspondantes aux visages de différentes expression,

– $W' = WW^t$

La figure 4.20 montre un résultat de catégorisation par simple projection sur les deux premiers axes, en utilisant les distances mpeg4. Visuellement, la répartition des visages donne un résultat plus au moins acceptable dans le cas d'expression neutre, joie et surprise. En revanche, dans certains cas, des erreurs de confusion sont apparues (figure 4.21) entre les différents types d'expression. Ces confusions sont dues principalement à l'imprécision de la détection automatique de points caractéristiques de chaque composante faciales Les fausses détections de ces points sont suscités par des erreurs de segmentation des composantes faciales, notamment avec les composantes faciales des bouches ouvertes et les sourcils.

Un grand avantage, c'est que l'expression neutre se trouve sous forme d'un groupe au milieu du nuage de points, ce qui confirme la validité de la proposition concernant le nouveau calcul de la matrice de similarité décrite précédemment.

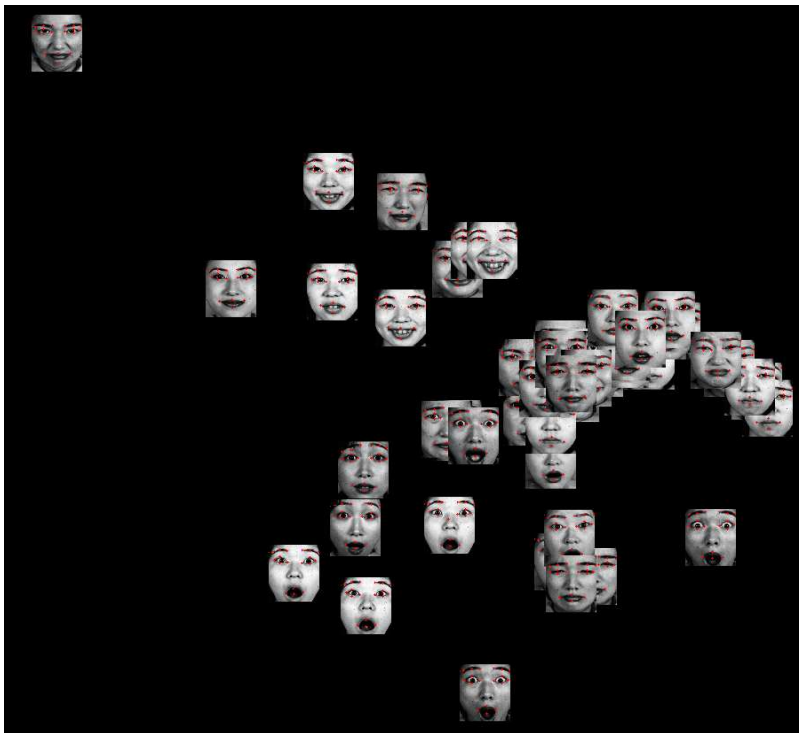


Figure 4.19: Projection des deux premiers vecteurs propres dans l'espace 2D selon les distances entre les points caractéristiques.

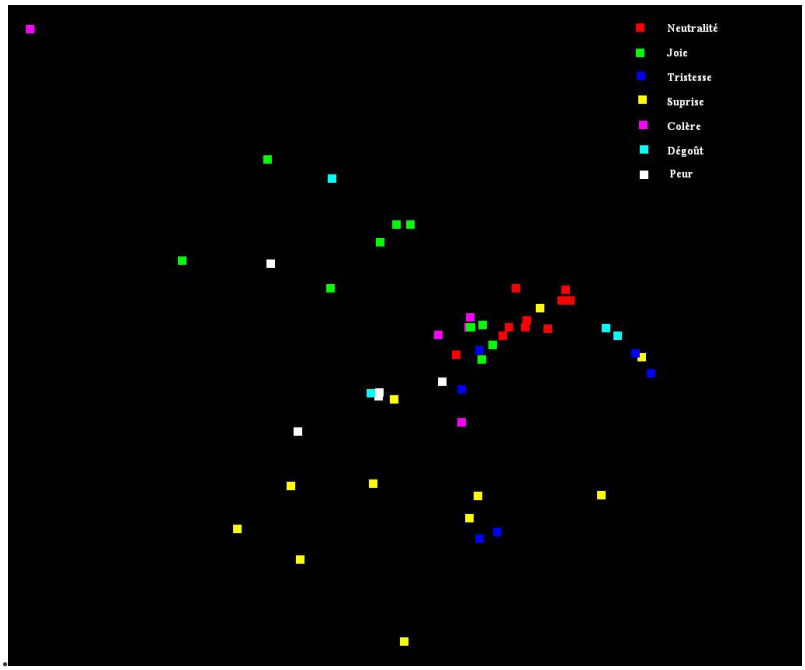


Figure 4.20: Visualisation des expressions dans l'espace 2D

– Approche mixte

L'idée consiste à combiner les deux distances, celle qui est basée sur l'intensité et les distances caractéristiques mpeg4. Pour cela, nous avons crée un vecteur caractéristique

composé de l'union des deux vecteurs. Dans la figure 4.22, où il s'agit d'utilisation combinant les deux distances, la répartition des visages donne un résultat meilleur, et l'expression neutre demeure au milieu du nuage de points.

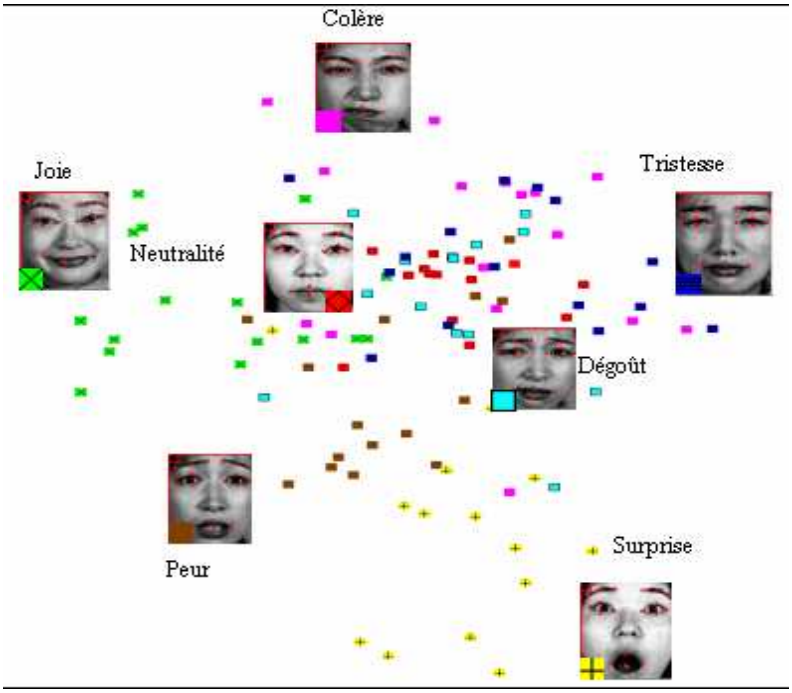


Figure 4.21: Répartition des composantes faciales

Le tableau suivant résume le taux de reconnaissance des expressions faciales de notre base, après classification en 7 classes :

Expression	Colère	Joie	Neutralité	Tristes	Dégout	Surpris	peur
Taux (%)	68,75	76,47	92,30	75	85,71	86,66	68,75

Le Tableau montre une bonne catégorisation automatique des expressions faciales des individus. En particulier pour l'expression neutre qui a été reconnue avec un taux maximum, malgré que l'approche soit basée sur des caractéristiques extraites d'une manière automatique.

4.5 Conclusion et perspectives

La détection du visage est une étape particulièrement importante dans un système d'analyse du visage. Plusieurs techniques de détection ont été présentées au cours de ce chapitre, chacune possédant ses forces et ses faiblesses. Compte tenu de la masse de données vidéo on a choisit d'utiliser une méthode basée couleur alliant l'atout d'être plus rapide que certaines techniques de détection du visage et facilitant la localisations des composantes faciales. Notre système de détection détermine en premier temps les zones du visage à l'aide d'un algorithme d'extraction des pixels de peau dans plusieurs espaces de couleurs HSV,

YCbCr, RGB. La combinaison des différents règles de décisions permet d'améliorer la localisation du visage potentiel. Par la suite, nous appliquons une approche basée sur la luminance et la chrominance de l'image couleur(EyeMap / MouthMap) pour localiser les composantes faciales (yeux, lévres, nez). Des résultats expérimentaux ont également été présentés, montrant des résultats intéressants et suffisants pour la phase d'interprétation et de catégorisation des expressions faciales.

Nous avons présenté notre contribution en essayant de répondre aux deux questions liées à la catégorisation des expressions faciales. Nous avons présenté un algorithme de détection et de segmentation des composantes faciales des yeux et de la bouche. Les deux méthodes utilisés ont montré une meilleurs efficacité pour la détection mieux que pour la segmentation.

Le premier modèle qui est basé sur la partition locale de graphe de grande dimension a montré son efficacité du fait que ce modèle caractérise non seulement les composantes faciales mais aussi les déformations musculaires autour de ces composantes. Ce qui fait de ce modèle un outil pour caractériser les actions faciales notamment dans la vidéo. Le second modèle basé sur EyeMap/MouthMap permet aussi de localiser les composantes faciales. En revanche cette méthode montre certaines limites notamment dans le cas de bouches ouvertes (zones très sombres).

En se basant sur les composantes faciales issues de la phase de la segmentation, un algorithme de clustering spectral (Graphe du Laplacian) a été choisi afin de visualiser la répartition de ces composantes. Ainsi, pour réaliser cette tâche, deux informations essentielles ont été prises en compte: l'information d'intensité(niveau de gris), les mesures de distances trouvées à partir de la déformation des points extrême correspondant aux composantes faciales.

La projection des deux vecteurs propres, calculés par la décomposition matricielle de la matrice de similarité, calculé d'une manière différente du mode classique, construite à partir des vecteurs d'information des composantes faciales, a montré de bons résultats de classification des expressions faciales 'Sourire',' Surprise' et 'Neutre'. D'un autre côté, pour palier aux fausses détections, il apparaît nécessaire d'exploiter les contours caractéristiques du visage par des approches de contours actif.

CHAPITRE 5 Structuration et catégorisation de vidéos

5 Structuration et catégorisation de vidéos

Le traitement d'images et de vidéos a connu ces dernières années une grande activité grâce à l'avènement de techniques permettant la reconnaissance de formes (personnes, objets, visages, etc) et leur suivi dans le temps (tracking). Lors de ces traitements, le but principal consiste à réduire le flux gigantesque de données fournies par les caméras. Le but est de comprendre le comportement et le geste d'une personne. Une telle analyse permet de synthétiser l'information visuelle sous une forme très compacte et avec un contenu sémantique important (de la position d'une personne, on passe au but poursuivi par cette personne). Il s'agit d'un domaine privilégié de la recherche car il permet d'expliquer le sens des actions d'une personne observée via une caméra, et ce de manière entièrement automatique. Dans le contexte social, les émotions ont un rôle prépondérant aussi bien en communication orale que non-verbale. Leur perception étant multimodale, l'aptitude d'un interlocuteur à les identifier à travers une variété de comportements tels que les mouvements du visage et les gestes constitue une base essentielle pour l'initiation de ses propres actions et réponses car le jugement et la prise de décision sont influencés par l'humeur, les sentiments mécanismes facilitant l'adaptation humaine et l'intégration sociale.

L'analyse de comportements peut aussi améliorer la précision et la robustesse des traitements d'images « bas-niveaux » via un mécanisme rétro-actif qui s'inspire des méthodes utilisées dans la reconnaissance de la parole. Par exemple, lorsqu'une personne montre un objet du doigt, une estimation erronée de l'orientation du doigt peut changer la signification du geste. La connaissance d'une information contextuelle provenant de l'analyse de comportements permettra de lever les ambiguïtés engendrées par le traitement « bas niveau.» En effet, en poursuivant les actions accomplies par la personne au fil du temps, ces actions prennent un sens et on peut mieux comprendre les choix effectués, voir même deviner quelques détails qui sont passés inaperçus à la première analyse. Ce point est très important pour des applications telles que les jeux interactifs où le joueur communique avec l'ordinateur via une caméra car il permet de rendre l'interface gestuelle plus robuste. Les applications privilégiées vont de la vidéosurveillance et la reconnaissance de gestes, à l'interaction visuelle homme machine, l'annotation de séquences sportives et la synthèse et animation des gestes et comportements.

Dans ce chapitre, nous allons présenter les résultats de structuration de vidéos par l'approche spectrale présentée dans le chapitre précédent. Une étape indispensable consiste à extraire le vecteur caractéristique de chaque vidéo. Pour les vidéos d'activités humaines, il fallait d'abord extraire et suivre les personnes en mouvement. Pour cela, nous avons utilisé notre approche mixte présentée dans le chapitre 2.

5.1 Activités humaines

Ces dernières années, le problème de reconnaissance et de classification d'activités humaines a suscité l'intérêt de communautés de recherche plus larges, allant des neurosciences du mouvement, la biomécanique, l'informatique, les sciences de la communication et les sciences de l'ingénierie. Dans plusieurs applications, telles que la vidéo surveillance, l'archivage et l'indexation de vidéos, il est important de reconnaître les mouvements des personnes pour pouvoir interpréter leurs comportements. Cette reconnaissance d'activité nécessite l'extraction de données multiples, l'interprétation automatique des séquences vidéos, et fait appel à des techniques d'analyse vidéo (perception visuelle, estimation de mouvement,…) et des méthodes d'analyse et de classification de données. Ce problème d'identification devient crucial quand on a un nombre croissant d'individus sous différents points de vue de caméras, et dans des environnements complexes.

5.1.1 Introduction

Les méthodes d'identification des activités humaines sont généralement basées sur les modèles d'apparence 2D ou 3D. Une catégorie des travaux consiste à détecter les différentes parties du corps telles que la tête, les mains, les pieds ainsi que d'autres parties du corps telles que les articulations [Gav99, Zhao01]). Haritaoglu et al. [Har99] proposent un système de reconnaissance globale d'actions qui est basé sur les projections horizontales et verticales de la silhouette de la personne, et de son orientation par rapport à la caméra (vue de face, vue de côté gauche, ...). Iwasawa et al [Iwas00] ont proposé une méthode qui consiste d'abord à déterminer le centre de gravité de la silhouette, ensuite qui calcule l'orientation de la moitié supérieure du corps, et enfin d'estimer les différentes parties significatives du corps en utilisant une analyse heuristique du contour de la silhouette. D'autres travaux, cherchent à suivre et interpréter le mouvement humain dans l'action. Efros et al. [Efros03] comparent deux actions en se basant sur les caractéristiques extraites, dans l'espace spatio-temporel, à partir du flot optique. Manor et Irani [Manor01] proposent une analyse multi-échelle de distributions du gradient temporel. Laptev et Lindeberg [Laptev03] comparent deux actions par appariement de points d'intérêt (Harris).

Plusieurs techniques d'estimation de mouvement, ont été utilisées dans le problème d'identification des actions. Yang et al. [Yang02] propose de suivre la trajectoire de la main et de la tête, qui est un mouvement affine. Une distance entre deux actions est ensuite calculée en comparant les délais, par réseau de neurones. Blank et al [Blank05] utilisent une pile de points de silhouettes qui sont extraites et évaluées en utilisant l'équation de Poisson pour chacun des points. La comparaison de deux actions se fait par distance euclidienne entre les vecteurs caractéristiques. Bobick et Davis [Bobick01] proposent d'utiliser les images d'énergie du mouvement et celles de l'historique du mouvement (MHI) , et la distance de Mahalanobis entre les moments de Hue 2D pour comparer entre deux actions.

Une action humaine étant fortement liée au mouvement, nous proposons dans cette thèse de suivre l'objet en mouvement et de former un volume dans l'espace 3D (2d+t). Ce volume qui représente une action donnée sera caractérisé par des moments géométriques 3D qui sont invariants à la translation et au changement d'échelle. Nous utilisons notre approche de catégorisation des actions basée sur la diffusion géométrique par marches aléatoires sur graphe. L'idée de base est de considérer l'ensemble des actions (vidéos) comme un graphe pondéré, où les sommets du graphe sont représentés par les volumes 3D (séquences des actions), et les arêtes connectés représentent la similarité entre les nœuds. Cette mesure de similarité sera calculée par une distance euclidienne entre les vecteurs caractéristiques des actions.

5.1.2 Extraction de caractéristiques de personnes en activité

La base de séquences que nous avons utilisé comporte 8 actions:

a. marcher (walk)
b. courir (run)
c. sauter sur les 2 pieds en se déplacant (jump)
d. sauter sur les 2 pieds sans se déplacer (jump in place)
e. toucher le sol avec la main droite et se remettre debout (bend)
f. lever la main gauche sans se déplacer (one-hand wave)
g. lever les 2 mains sans se déplacer (two-hands wave)
h. mouvement a pas chassé (gallop sideways)

Chaque action est exécutée par 9 personnes différentes (Ido, Shahar, …etc). On a donc un total de 72 vidéos d'une centaine de frames chacune, avec des fonds uniformes. La résolution de chaque vidéo est de 180 * 144, 25 fps. Une action peut être exécuté de la gauche vers la droite ou de la droite vers la gauche par des personnes différentes.

– **Segmentation spatio-temporelle:**

La première étape consiste à segmenter la personne en mouvement. La figure 5.1 présente un exemple d'images représentatives d'une vidéo de l'action « Run ». Notre objectif étant de segmenter l'objet 3D en mouvement, nous avons utilisé pour cela notre méthode mixte de segmentation et de suivi d'objets qui combinant à la fois une segmentation par contour actif, basée régions et l'estimation de mouvement par flot optique [Zinbi06]. L'idée principale de l'approche mixte consiste à utiliser une segmentation active des objets 3D par une résolution simultanée du problème d'estimation de mouvement et de segmentation active d'une image vidéo, par minimisation d'énergie.

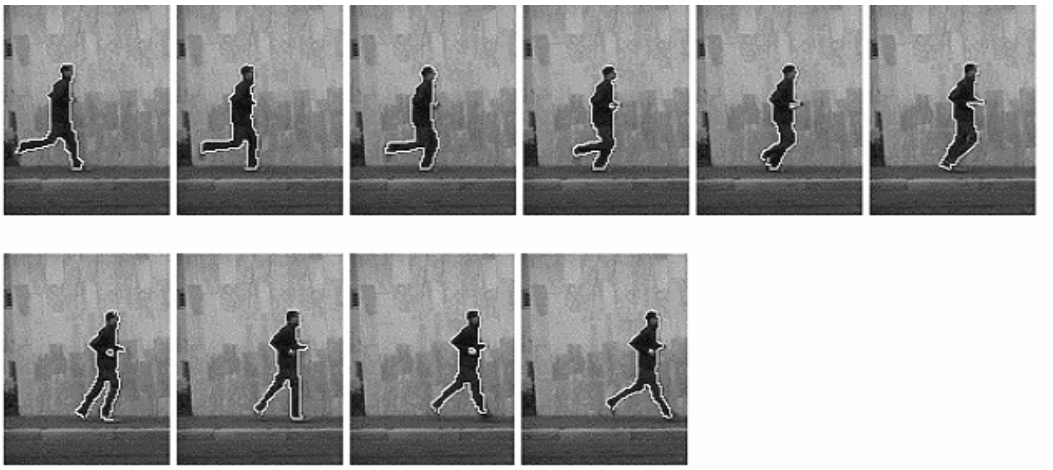


Figure 5.1: Images représentatives de l'activité « Run » de la personne Ido

La figure 5.2 montre un autre exemple de segmentation d'une autre personne en mouvement de la base. Il y a d'autres séquences d'activité telles que « se pencher », « lever une main », et « lever les deux mains », où le mouvement ne capture que des parties des mains, notre approche capture l'objet d'intérêt (personne) en plus de son mouvement, contrairement aux autres approches basées uniquement sur le mouvement.

Figure 5.2: Segmentation de la personne « Shahahr » qui saute

A partir du volume des images binaires, il faut extraire des caractéristiques représentatives de l'action.

– **Energie du mouvement (MEI) :**

L'énergie du mouvement est essentiellement une image du mouvement cumulé. Elle indique l'emplacement spatial du mouvement. Elle est calculée comme suit:

$$E_r(x,y,t) = \begin{cases} 0 & \text{si} \quad B(x,y,t) = 0 \\ 1 & \text{sinon} \end{cases} \quad t \in \{t-r,...t\} \tag{5.1}$$

r est le temps de capture de la séquence d'image .

– **Historique du mouvement (MHI):**

Les caractéristiques temporelles du mouvement sont importantes pour l'analyse du mouvement. L'historique du mouvement caractérisant la séquence temporelle est défini par :

$$H_r(x,y,t) = \begin{cases} r & \text{si} \quad B(x,y,t) = 1 \\ \text{Max}(0, H_r(x,y,t-1)) & \text{sinon} \end{cases} \tag{5.2}$$

Le résultat est une fonction du mouvement de chaque pixel. La brillance d'un pixel est proportionnelle au changement de l'intensité, donc à la séquence du mouvement .

La figure 5.3b illustre l'historique des actions, qui est la projection 2D du volume. On peut remarquer, d'ores et déjà qu'il y a des informations, telles que la durée d'une action, son rythme, et le sens d'une trajectoire qui sont importants. On peut espérer que la forme, la durée et le rythme d'une action soit capturé par le vecteur caractéristique du volume.

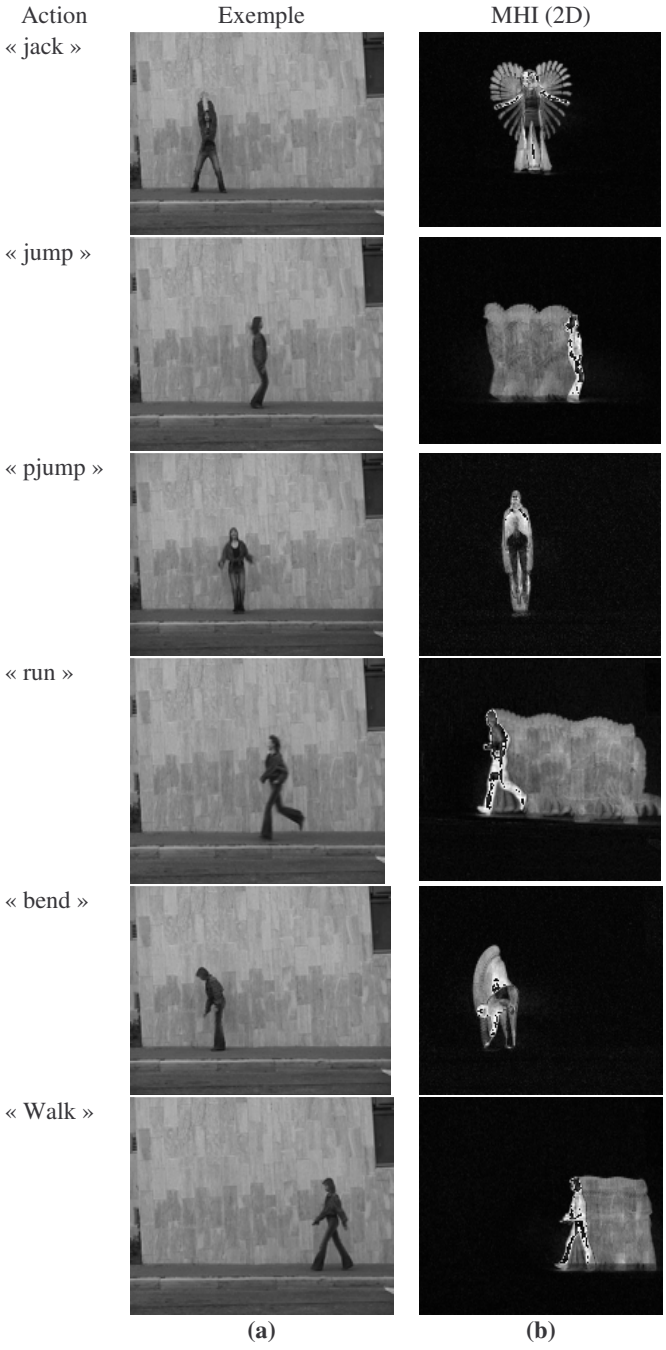


Figure 5.3: Exemples de la base des actions et de leur historique de mouvement

– Silhouette 3D

L'utilisation de la morphologie mathématique en détection et en estimation du mouvement a donné lieu a des développements intéressants pour les systèmes de détection, de poursuite et

de reconnaissance d'objets. L'utilisation d'éléments structurants temporels ou spatio-temporels permet d'aborder différemment la différentiation trame a trame. Elle permet à la fois de calculer des filtres spatio-temporels intéressants, et d'intégrer des changements temporels par accumulation (lorsque l'élément structurant est allongé dans l'axe temporel). Elle permet d'autre part de discriminer un déplacement donné en orientant l'élément structurant dans la direction correspondante. La figure 5.4 montre, une autre alternative à notre approche de segmentation, qui est une implémentation par la Ligne de Partage des Eaux (LPE).

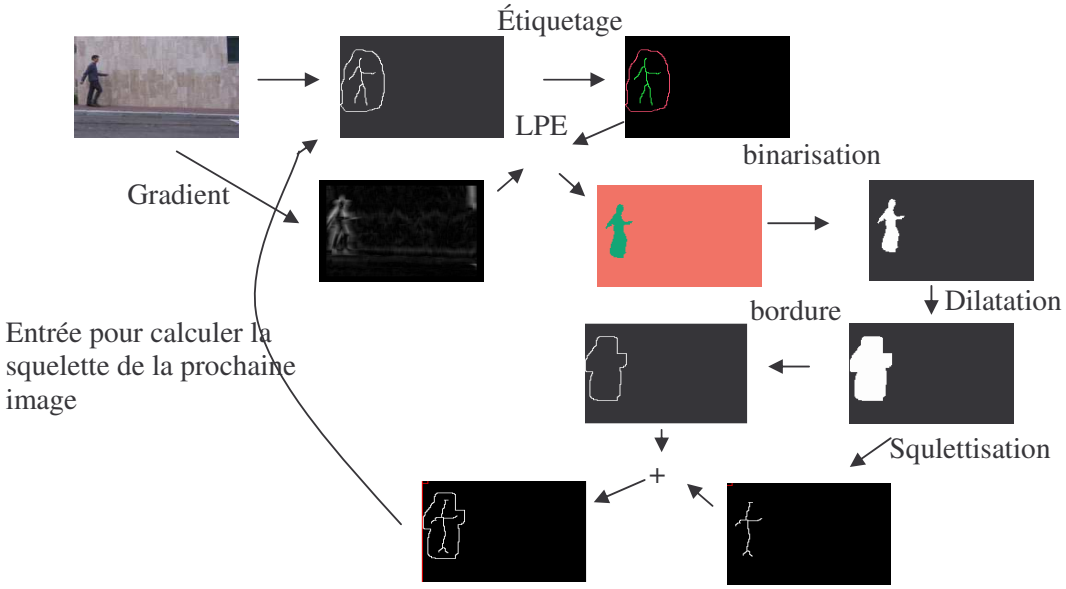


Figure 5.4: Technique de construction d'une squelette d'une action 3D

Dans la littérature, il a été proposé une mesure basée sur la distance entre les maximums pour évaluer la similarité entre 2 silhouettes. Les points squelettiques sont sur le centre da la silhouette. Définissons SD comme l'ensemble qui contient les points du squelette de la silhouette détectée, et SMi l'ensemble des points du squelette du modèle de l'action i. la mesure entre les 2 squelettes SD et SMi est donnée par :

$$M_i = \sum_{pd \in SD} \min_{pm \in M_i} (|pd, pm|) \qquad (5.3)$$

Avec |.,.| est la distance euclidienne. L'action qui minimise cette mesure est choisie comme solution.

Une autre manière de représenter une silhouette est d'utiliser ses projections horizontales et verticales [Haritaoglu98] [Panini03] [Boulay05]. Une fois que nous avons la silhouette binaire de la personne nous la représentons par sa projection horizontale et verticale. La projection

horizontale (verticale) sur l'axe de référence est obtenue en comptant la quantité de pixels du mouvement correspondant à la personne détectée pour chaque rangée (colonne) de l'image. Ils projettent le modèle 3D sur une image pour chaque action de référence qui a été produite pour toutes les orientations possibles. Alors nous comparons la projection (H&V) de ces images avec la projection (H&V) de la silhouette détectée. Ils proposent une comparaison basée sur les secteurs non recouverts des projections (H&V) . Ils définissent 2 rapports:

$$R_0(H) = \frac{\sum_{ir \in I_0} (H_{ir}^0 - H_{ir}^m)^2}{\sum_{ir} (H_{ir}^0)^2} \tag{5.4}$$

Ce qui représente la somme de la différence au carré des projections calculées sur l'intervalle I0, normalisé par la somme des valeurs au carré de la projection horizontale de la personne détectée (H^d) .

$$R_m(H) = \frac{\sum_{ir \in I_m} (H_{ir}^0 - H_{ir}^m)^2}{\sum_{ir} (H_{ir}^m)^2} \tag{5.5}$$

Ce qui représente la somme des différences au carrée des projections calculées sur l'intervalle I0, normalisé par la somme des valeurs au carré de la projection horizontale du modèle produit (H^m) . La distance entre la silhouette détectée Sild et la silhouette modèle Silm est donnée par :

$$\text{dist}(S_{ilm}, S_{ild}) = \frac{1}{4} R_0(H) + R_m(H) + R_0(V) + R_m(V) \tag{5.6}$$

Cette distance appartient à l'intervalle [0,1] pour laquelle 0 correspond aux silhouettes semblables. Le modèle de l'action qui donne la distance minimum est choisi pour l'action da la personne étudiée.

5.1.3 Reconnaissance de la forme par des moments statistiques

On distingue deux grandes approches de description de formes. L'approche contour qui caractérise la forme à partir de son contour sans tenir compte de la texture et l'approche globale qui étudie la forme dans son ensemble. Ici, nous ne décrivons que des représentations globales basées sur les moments : Les moments 2D de hu et les moments géométriques 3D.

– **Moments de Hu :**

La représentation des formes par des moments statistiques est une technique classique dans la littérature. Ces moments sont basés sur les moments polynomiaux 2D :

$$m_{pq} = \iint x^p y^q P(x,y)\, dxdy \tag{5.7}$$

Où P = 1 si le pixel appartient à la silhouette
0 sinon

Afin de rendre les moments invariants à la translation, les moments sont centrés :

$$u_{pq} = \iint (x-\bar{x})^p (y-\bar{y})^q P(x,y) dxdy) \tag{5.8}$$

avec $\bar{x} = \frac{m_{10}}{m_{00}}$ et $\bar{y} = \frac{m_{01}}{m_{00}}$, ou m_{00} est la surface de l'objet

Plus loin les moments sont calculés de sorte qu'ils soient invariants à l'échelle:

$$n_{pq} = \frac{u_{pq}}{u_{00}^{\frac{p+q}{2}+1}} \tag{5.9}$$

Finalement pour ces moments invariants à la rotation, les 7 moments de hu sont calculés :

$$s_1 = n_{20} + n_{02}$$
$$s_2 = (n_{20} - n_{02})(n_{20} - n_{02}) + 4n_{11}n_{11}$$
$$s_3 = (n_{30} - 3n_{12})(n_{30} + 3n_{12}) + (n_{03} - 3n_{21})(n_{03} - 3n_{21})$$
$$s_4 = (n_{30} - n_{12})(n_{30} + n_{12}) + (n_{03} + n_{21})(n_{03} + n_{21})$$
$$s_5 = (n_{30} - 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})(n_{30} + n_{12}) - 3(n_{03} + n_{21})(n_{03} + n_{21})] + (3n_{21} - n_{03})(n_{03} + n_{21})[3(n_{30} + n_{12})(n_{30} + n_{12}) - (n_{03} + n_{21})(n_{03} + n_{21})]$$
$$s_6 = (n_{20} - n_{02})[(n_{30} + n_{12}) - (n_{30} + n_{12}) - (n_{03} + n_{21})(n_{03} + n_{21}) + 4n_{11}(n_{30} + n_{12})(n_{03} + n_{21})]$$
$$s_7 = (3n_{21} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})(n_{30} + n_{12}) - 3(n_{21} + n_{03})(n_{21} + n_{03})] - 3(n_{21}+n_{03})(n_{21} + n_{03})] - (n_{30} - 3n_{12})(n_{21} + n_{02})[3(n_{30} + n_{12})(n_{30} + n_{12}) - (n_{21} + n_{03})(n_{21} + n_{03})]$$

– Moments géométriques 3D

A partir de toutes les images binaires obtenues, il faut extraire des caractéristiques représentatives de la séquence. On a opté pour une représentation globale des actions afin de simplifier le processus de reconnaissance et d'amener plus de robustesse lors de cette étape. Pour cela, une action est tout d'abord représentée par le volume 3D constitué par tous les points (x, y, t) détectés en mouvement. Ce volume spatio-temporel contient beaucoup d'informations dont la silhouette de la personne à chaque image et la dynamique de l'action: est ce que la forme s'agrandit au cours du temps, se déplace vers la gauche, etc. Pour caractériser ce volume, sans avoir à extraire (et séparer) les différentes informations présentes (posture, mouvement,…), nous utilisons les moments géométriques 3D[Chahir09].

Soit {x,y,t} l'ensemble des points appartenant au volume binaire où x et y représentent les coordonnées spatiales et t, la coordonnée temporelle. Le moment d'ordre (p+q+r) de ce volume est déterminé par:

$$A_{pqr} = E\{x^p y^q t^r\} \tag{5.10}$$

Où E{x} représente l'espérance mathématique de x. Afin d'utiliser des caractéristiques invariantes en translation, ils utilisent les moments centrés définis par :

$$Ac_{pqr} = E\{(x - A_{100})^p (y - A_{010})^q (t - A_{001})^r\} \tag{5.11}$$

Ces moments doivent aussi être rendus invariants par rapport à l'échelle pour préserver une invariance à la distance de l'action ou à la taille des personnes [Mokhber05]. Une normalisation directe sur les différents axes, en divisant chaque composante par l'écart type correspondant n'est pas souhaitable car elle va amener une grosse perte d'informations quant à la forme des silhouettes binaires qui va s'arrondir. Aussi, une normalisation identique est effectuée sur les deux premiers axes, tandis que le troisième (le temps) sera normalisé séparément. La normalisation réalisée en conservant le ratio largeur/hauteur des silhouettes binaires est donc obtenue avec :

$$M_{pqr} = E\{\left(\frac{x - A_{100}}{Ac_{200}^{1/4} Ac_{020}^{1/4}}\right)^p \left(\frac{y - A_{010}}{Ac_{200}^{1/4} Ac_{020}^{1/4}}\right)^q \left(\frac{t - A_{001}}{Ac_{002}^{1/2}}\right)^r\} \tag{5.12}$$

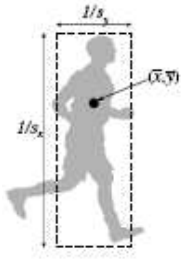
5.1.4 Catégorisation des activités humaines

La durée des actions est variable dans le corpus, et donc nous nous sommes restreint à une dizaine de frames par vidéo.

– Distance entre moments:

Une première tentative consiste à comparer uniquement les distances entre les vecteurs composés de moments 3D . La reconnaissance est réalisée en cherchant pour chaque vecteur caractéristique d'une action de la base, un vecteur plus proche dans la même base, au sens de la distance euclidienne. Ce qui revient à construire une matrice de distance entre toutes les vidéos de la base. Ensuite un kmeans est utilise afin de regrouper les distances les plus proches. Nous utilisons un vecteur de caractéristique M compose des moments d'ordre 2 et 3, soit 14 moments:

M={M200, M011, M101, M110, M300, M030, M003, M210, M201, M120, M021, M102, M012, M111}.



Les 14 moments de action courir (2) de la base :

1.31474 0.0667942 1.10797 0.0110768 0.000652416 -0.000166253 0.000220814 0.000700451 0.000547358 6.48052e-05 8.47032e-05 0.000374271 6.14959e-05 0.000244231 1.00002

Sur le tableau suivant, sont présentés les taux de reconnaissance obtenus lors de la comparaison des distances.

	1	2	3	4	5	6	7	8	9
1Walk	50	37.5	37.5	50	50	37.5	37.5	50	50
2Run	62.5	62.5	62.5	25	25	62.5	62.5	62.5	25
3Jump	25	25	50	50	50	50	25	50	50
4Pjump	100	100	88.88	100	88.88	100	100	88.88	100
5Bend	100	88.88	100	100	88.88	100	100	100	100
6Wave1	88.88	100	100	88.88	88.88	88.88	100	88.88	88.88
7Wave2	100	100	100	100	100	88.88	88.88	100	100
8Side	50	50	50	37.5	37.5	37.5	50	37.5	50

Tableau 5-1: Taux de reconnaissance obtenus par comparaison des distances

Les taux de reconnaissance moyens sur les 8 actions effectuées chacune par 9 personnes varient de 44,44 a 97,52. Nous constatons que les taux de reconnaissance moyens des personnes qui font des mouvements en se déplaçant (action 1,2,3,8), est faible par rapport aux personnes qui font des mouvements sans se déplacer (action 4,5,6,7). On peut donc conclure que les actions effectuées par des personnes qui font des mouvements sans se déplacer sont bien reconnues. Nous avons constaté aussi que le mouvement des personnes effectuant

l'action de la gauche vers la droite n'est pas classé de la même manière que les personnes effectuant l'action de la droite ver la gauche, de même mouvement, c'est ce qui explique le taux de reconnaissance faible dans les action 4,5,6,7 et provoque des confusions avec d'autres mouvements effectués dans la même direction. Nous présentons la matrice de confusion moyenne, à partir des distances, obtenue sur les 8 actions:

	1walk	2run	3jump	4pjump	5bend	6wave1	7wave2	8side
1Walk	44.44	0	34.72	0	0	0	0	20.84
2Run	0	50	20.32	0	0	0	0	29.69
3Jump	0	4.18	41.66	0	0	0	0	54.16
4Pjump	0	0	0	96.29	3.71	0	0	0
5Bend	0	0	0	0	97.52	0	2.48	0
6Wave1	0	0	0	0	0	95.58	4.42	0
7Wave2	0	0	0	0	2.48	0	97.52	0
8Side	0	19.45	36.11	0	0	0	0	44.44

Tableau 5-2: Matrice de confusion

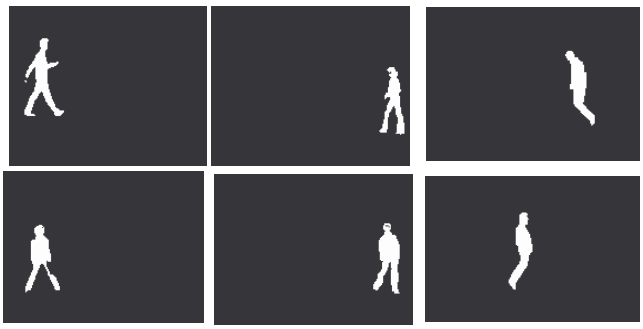


Figure 5.5: Même actions effectuées dans des direction différentes

Par exemple, l'action marcher (1) est exécutée par 9 personnes différentes, 5 personnes parmi les 9 marchent de la gauche ver l'adroite, et les 4 autres de la droite ver la gauche. En comparant les moments de la première personne qui marche avec les moments de toutes la base, il a été classé comme étant plus proche de 4 personnes qui marchent et ces personnes ont la même direction que lui, par contre il y a eu confusion avec 3 personnes de l'action 3 et une personne de l'action 8, qu'ont toujours la même direction du mouvement que lui. Il n'a pas reconnue les 4 personnes qui marchent de la droite vers la gauche. Même constat fait pour toutes les autres actions.

– Similarité basée sur MHIs:

Les figures 5.6 et 5.7, montrent un premier résultat de l'utilisation des marches aléatoires sur graphe. Il s'agit de la réorganisation des historiques de mouvement en triant le 1er vecteur propre dans l'espace de diffusion, en utilisant la surface (différence de pixel). Dans cet

exemple, pour l'affichage nous nous sommes limités aux 4 classes parmi les 8 actions. On peut remarquer que les 8 actions, des personnes, de même nature se suivent et donc peuvent être classés à part. En fait, il s'agit des classes : « bend », « pjump » ,« wave1 » et « wave2 ».

Dans la figure suivante, on présente le résultat de tri du premier vecteur propre, selon une mesure de similarité basée sur la différence de pixels.

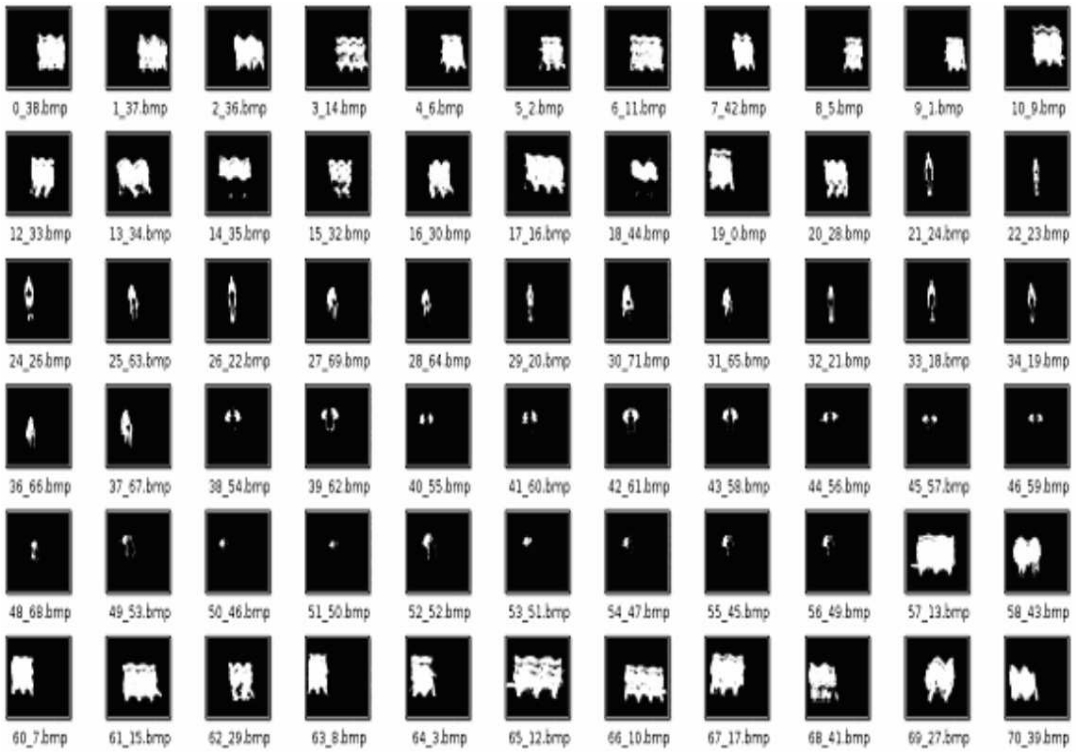


Figure 5.6: Tri du 1er vecteur propre selon une différence de pixels

D'après le premier vecteur propre on remarque trois classes :

* Une première classe des personnes qui se déplacent de la droite vers la gauche indépendamment de leur mouvement. (images 0-20)

* Une 2ème classe des personnes qui font des mouvements sans se déplacer, qui sont d'ailleurs bien classés. On remarque juste une confusion entre le mouvement 4 et 5. Par contre le mouvement 7 (image 38 à 46) et 8 (images 46 à 56) sont reconnues à 100% .

* Une 3ème classe qui regroupe des personnes qui se déplacent de la gauche vers la droite.

Nous avons effectué le même travail (figure 5.7) , mais en comparant les surfaces générées par l'historique des actions.

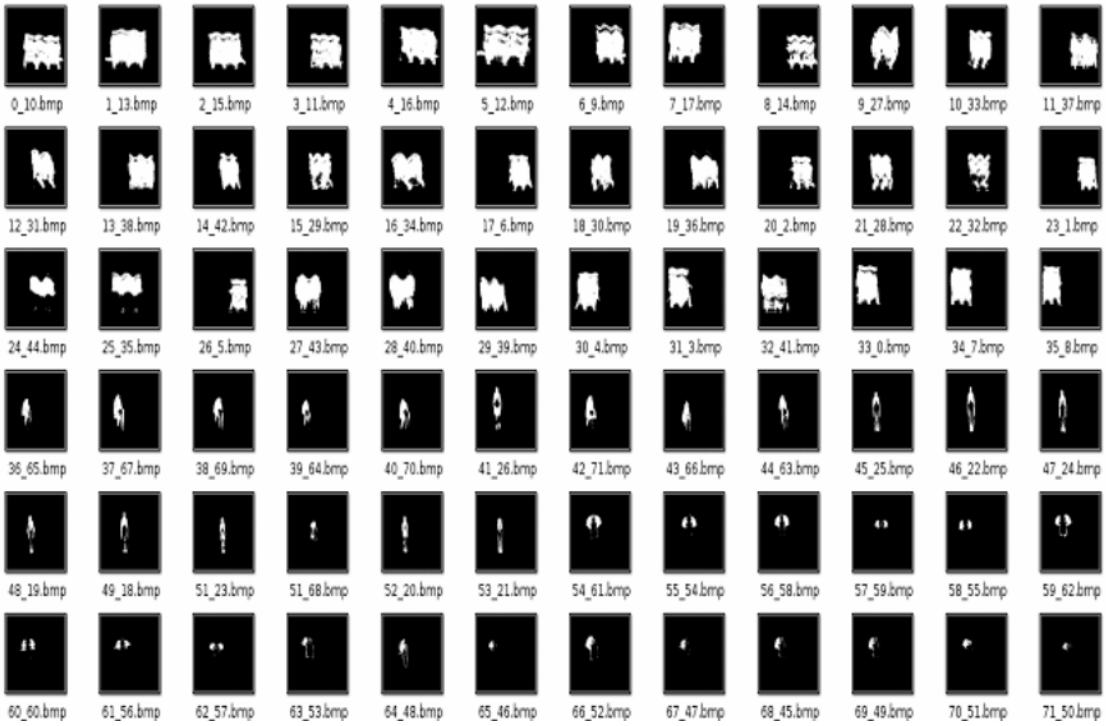


Figure 5.7: Réorganisation selon les surfaces des historiques

A première vue des résultats, on remarque deux classes:

* une première classe, regroupe des personnes qui se déplacent en effectuant le même mouvement.

* une 2ème classe, regroupe des personnes qui font des mouvements sans se déplacer.

En détaillant le vecteur propre, on remarque que l'action 2 (courir) a été détectée à 100% cela est dû à la particularité de l'action, qui est dense. Par contre les autres actions sont confuses. En ce qui concerne les personnes qui effectuent le mouvement sans se déplacer, l'action 7 et 8 sont reconnues à 100%, on remarque juste une confusion entre le mouvement 4 et 5.

La figure 5.8 montre une projection sur les deux premiers vecteurs propres V1 et V2. Quatre classes C1, C2, C3 et C4 sont bien détectées qui sont respectivement bend, pjump, wave1, wave2.

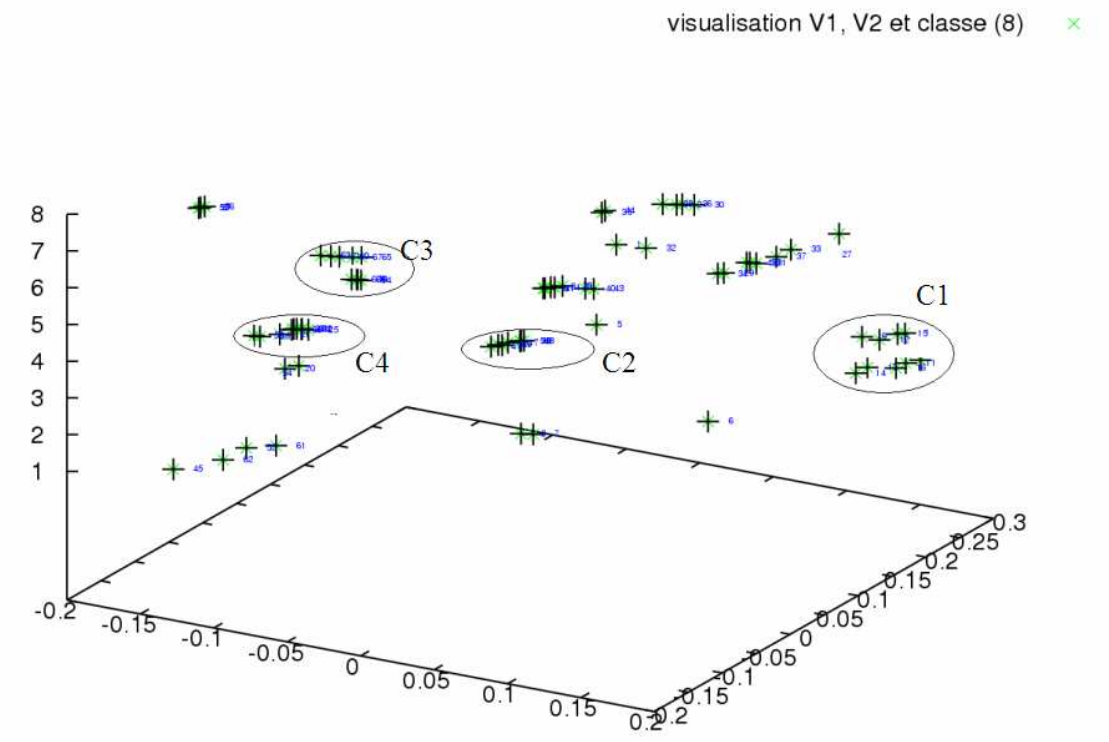


Figure 5.8: Catégorisation des actions en 8 classes, sur $(\lambda_1^t\psi_1, \lambda_2^t\psi_2)$

– Similarité basée sur les moments géométriques 3D:

La figure 5.9 montre un premier résultat de la réorganisation des historiques de mouvement en triant les données associées au 1er vecteur propre dans l'espace de diffusion, en utilisant les moments 3D. Dans cet exemple, pour l'affichage nous nous sommes limités aux trois classes parmi les dix actions, qui sont « bend », « pjump » et « jack ». Les actions sont classées de gauche à droite, de haut en bas. Cependant, il y a un problème, signalé par *, qui indique que cette image ne fait pas partie de la classe en question. Il s'agit d'une action de « sauter » qui a permuté avec une autre de « se pencher ».

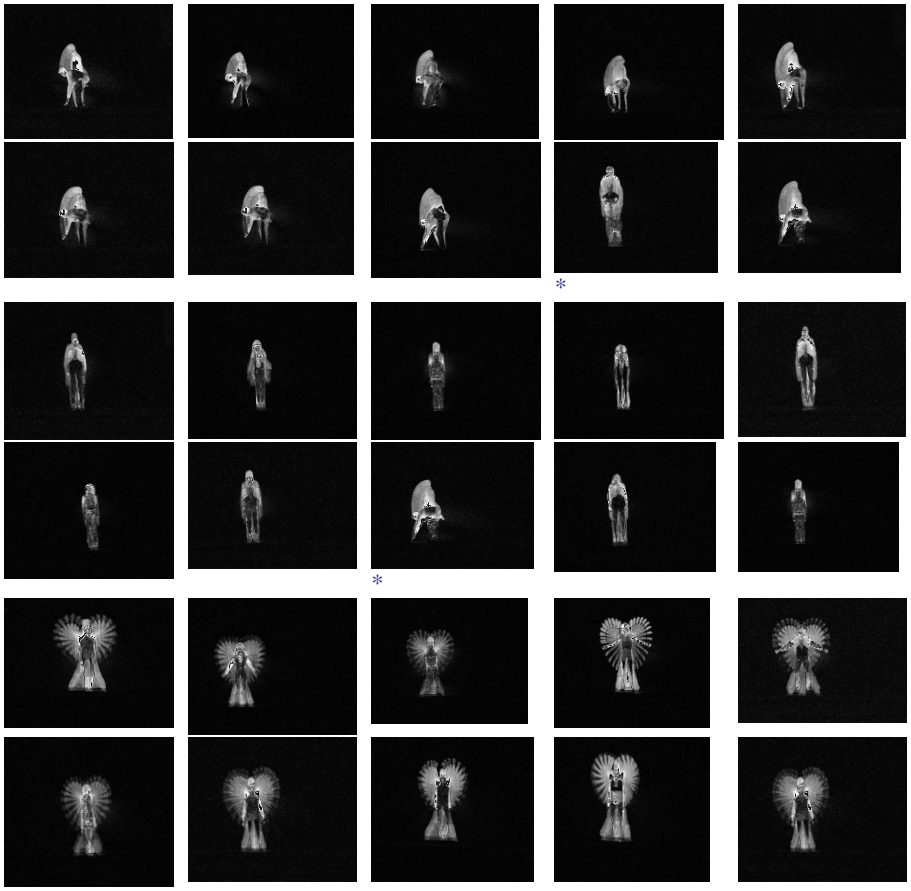


Figure 5.9: Réorganisation des historiques de mouvements par analyse spectrale

La figure 5.10, montre le résultat de la catégorisation des actions en utilisant la classification par k-means en 10 classes, sur les deux projections:

$$(\lambda_1\varphi_1, \lambda_2\varphi_2)$$

La projection des actions montre l'émergence de certaines classes, mais avec certaines fausses détections. Le point positif qu'on peut relever est le fait que dans chaque classe, il y a plus d'actions similaires. Pour améliorer, la classification des actions, on doit à la fois contrôler l'extraction des volumes en 3D, mais aussi tenir compte des critères calculables à partir du volume tels que le rythme.

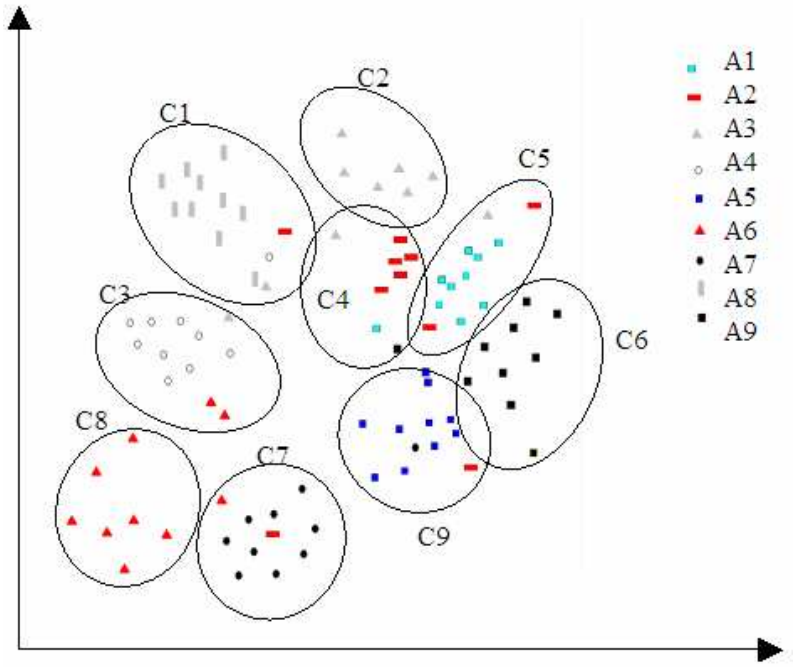


Figure 5.10: Catégorisation des actions en utilisant les moments 3D

5.2 Structuration de home vidéos

Dans le cadre d'une vidéo, il s'agit de détecter des groupes d'images proches selon une mesure de similarité donnée qui va favoriser des regroupements qu'on peut classer en plans ou en scènes. Une première application consiste à détecter les plans (ou cuts) qui généralement représentent des prises de caméra continues et une constance de luminosité.

Ensuite, les plans qui partagent la même sémantique peuvent être groupés en scènes. Ici, on suppose que la sémantique des plans vidéo est suffisamment bien exprimée par les caractéristiques de bas niveau tel que les informations de couleur, ou des informations topologiques tel que le mouvement.

Dans le cadre d'un corpus de vidéos, il s'agit aussi de classer des vidéos entières dans des catégories bien définies. Pour mettre en évidence le cadre unifié de l'approche basée sur la diffusion sur graphe pour la structuration, visualisation et catégorisation des ces objets à savoir : une image, un plan et une scène pour une vidéo donnée, nous avons construit un graphe complet entre les images d'une vidéo où la mesure de similarité est définie par : $w(x_i, x_j) = 1$ si x_i, x_j appartiennent à la même classe, sinon $w(x_i, x_j) = 0$. La décomposition spectrale de la matrice de transition P détermine les valeurs propres et les vecteurs propres de la matrice P : $\{\lambda_1^t\psi_1, \lambda_2^t\psi_2, \ldots, \lambda_i^t\psi_i \ldots\}$.

Il s'agit de retrouver une structure hiérarchique dans la vidéo. Tout d'abord, détecter et réordonner des classes d'images ordonnées qui ont le même contenu visuel. Ensuite, découper

ces ensembles en scènes de plans très proches tels que les plans alternatifs. La détermination d'une mesure de similarité entre ensembles est un problème qui se pose dans beaucoup d'applications: extraction d'information, indexation automatique, etc. Dans le cadre des données vidéo, la mesure de similarité dépend de ce qu'on cherche à catégoriser et de comment on souhaite le faire. Par exemple, la mesure de similarité entre images diffère de celles entre les plans ou entre les scènes. Aussi, il y a le choix des caractéristiques audiovisuelles qu'on souhaite prendre en compte: couleur, texture, mouvement, forme, rythme.

Zeeshan et Shah [Zee05] proposent une mesure de similarité entre deux plans qui est exprimée selon deux mesures: la similarité visuelle (couleur) basée sur la maximisation de vraisemblance (SimVis) et la similarité entre les images successives (SimMouv) qui tient compte des intervalles temporels mais aussi du flot optique entre les deux séries d'images.

$$d_S(x, y) = \alpha.\mathrm{SimVis}(x, y) + \beta.\mathrm{SimMouv}(x, y) \tag{5.13}$$

où α et β sont les poids qu'on utilise tels que $\alpha+\beta=1$ (par défaut à $\alpha=\beta=0.5$) .

La similarité entre deux ensembles d'images images i et j définies sur le critère de couleur est définie par :

$$\mathrm{SimVis}(i, j) = \max_{p \in k_i, q \in k_j} (\mathrm{SimCol}) \text{ avec } \mathrm{SimCol} = \sum \min(h_p, h_q) \tag{5.14}$$

où p et q sont les images des plans i et j avec h_p et h_q leurs histogrammes couleur respectifs. D'un autre côté, la similarité entre plans i et j en se basant sur l'aspect spatio-temporel, peut être définie de plusieurs manières dont :

$$\mathrm{SimMouv}(i, j) = \frac{2.\min(\mathrm{Mouv}_i, \mathrm{Mouv}_j)}{\mathrm{Mouv}_i, +\mathrm{Mouv}_j} \tag{5.15}$$

$$\text{avec } \mathrm{Mouv}_i = \frac{1}{b-a} \sum_{x=a}^{b-1} (1 - \mathrm{SimCol}(x, x+1))$$

Nous avons construit un graphe dont les nœuds sont constitués par les plans (vidéos) et nous avons expérimenté notre approche de catégorisation basée sur les marches aléatoires sur graphe [Zinbi08b]. Dans nos expériences, nous avons d'abord implémenté la méthode proposée dans [Zee05], ensuite une similarité simple entre deux plans x et y basée sur les histogrammes cumulées :

$$d_S(x,y) = \left\| hcum(x_i) - hcum(x_j) \right\|^2 \tag{5.15}$$

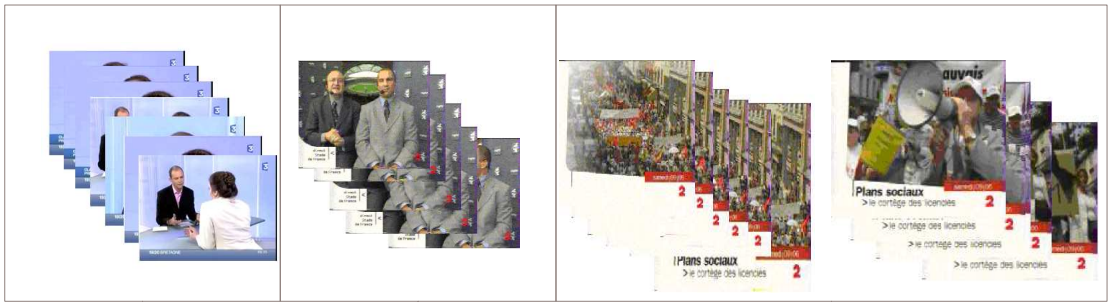
Mais pour la série d'expérimentation qui suit, nous avons utilisé une mesure, proposée par Odobez et al. [Odo03]combinant une similarité visuelle et temporelle. La matrice d'affinité W est définie par :

$$W_{ij} = W_{ij}^v W i_j^t \text{ avec } W_{ij}^v = e^{-\frac{d_v^2(f_i,f_j)}{2\sigma_v^2}} \text{ et } W_{ij}^t = e^{-\frac{d_t^2(f_i,f_j)}{2\sigma_t^2}} \tag{5.16}$$

où W_{ij} représente la similarité entre les images clés f_i et f_j, d_v et d_t sont les distances de similarités visuelles et temporelles, et σ_v^2 et σ_t^2 sont des estimations utilisées comme facteurs d'échelles visuel et temporel. La similarité visuelle est calculée à partir du coefficient Bhattacharyya, qui est reconnu pour sa robustesse lors de comparaison de distributions de couleur.

$d_v(f_i, f_j) = \sqrt{1-\rho_{BT}}$ avec $\rho_{BT} = \sum_k \sqrt{h_{ik} h_{jk}}$ qui est le coefficient de Bhattacharyya calculé à partir des histogrammes.

$d_t(f_i, f_j) = \frac{\left| |f_j| - |f_i| \right|}{|vc|}$ où $|f_j|$ représente le nombre de frames du plan représenté par f_j, et $|vc|$ représente la durée totale de la vidéo.



A b c d

Figure 5.11: Exemple de séquences vidéo d'un corpus de journal télévisé avec 4 plans différents(a-c) et une scène (c+d)

La figure 5.11 montre quatre plans différents. Les deux derniers constitue une scène. Pour la visualisation des plans similaires dans le repère $(\lambda_1\psi_1, \lambda_1\psi_2)$, nous avons représenté chaque plan par son image représentative(image clé). Cette dernière a été sélectionné en prenant l'image du plan qui a la valeur la plus élevée du vecteur ψ_1.

Nous avons testé notre algorithme sur une petite vidéo de 8 minutes., composée de 52 plans identifiés, que nous avons classé en 6 scènes (S1,...,S6). Chaque scène regroupe un ensemble de plans similaires.

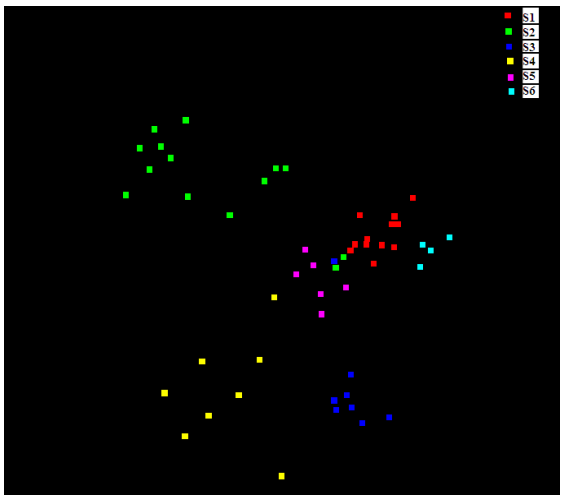


Figure5.12: Visualisation de catégories (scènes) dans l'espace de diffusion (2 premiers axes)

Les plans de chaque scène sont représentés par une couleur. Comme on peut le voir dans cette figure, la plus part des plans similaires de même couleur sont proches.

5.3 Conclusion et perspectives

Nous avons présenté dans ce travail le processus de détection, de reconnaissance et de représentation des données liées actions humaines tel que « sauter », « marcher »…etc. Nous avons dans un premier temps utilisé notre méthode se segmentation présentée dans le chapitre 2 pour l'extraction des personnes en action. Ensuite, nous avons résumé chaque action par une image représentative qui illustre l'action spatio-temporelle. Dans une seconde partie, nous avons étudié l'opérateur de caractérisation volumique de forme qui sont les moments géométrique 3D, afin de caractériser chaque action. Enfin, nous avons présenté le résultat de processus d'exploration du graphe des actions par marches aléatoires, lié aux méthodes spectrales afin de réduire la dimension de représentation des données. Ce processus de diffusion fournit une structure de représentation de données adéquate. Nous avons exploité ce processus également pour la structuration de home vidéos, en utilisant des vecteurs caractéristiques appropriés.

CHAPITRE

6 Conclusions & Perspectives

6 Conclusions & Perspectives

6.1 Conclusions

Nous avons présenté dans ce mémoire de thèse un système permettant de réaliser l'extraction, l'analyse et l'interprétation d'expressions et de mouvement humain dans des séquences vidéo. Les méthodes de segmentation d'images sont nombreuses; toutes présentent des avantages mais ne donnent pas entière satisfaction. Toutes doivent être adaptées en fonction des applications que l'on se propose de réaliser. Dans une première partie, nous avons étudié les contours actifs rapides et les avons replacés dans le cadre du traitement d'images et particulièrement parmi les méthodes de segmentation spatio-temporelles. Il s'agit, de la minimisation d'une somme d'énergies diverses qui combinent à la fois les contours actifs et le flot optique. Cette énergie doit être définie en fonction du problème à traiter. La contribution principale de ce travail est la définition d'une énergie permettant le suivi d'objets en mouvement, ainsi que l'introduction d'une procédure d'initialisation rendant ce suivi automatique. Une méthodologie du choix des différents paramètres régissant l'évolution des contours actifs est également proposée. Nous avons proposé, pour ce problème de minimisation, d'utiliser les contours actifs globaux qui ne nécessite aucune connaissance structurelle a priori de l'image.

En seconde partie, nous avons abordé les méthodes non linéaires de réduction de dimension, en particulier une approche géométrique globale pour la réduction de dimensionnalité non-linéaire basée sur les marches aléatoires sur graphe. L'idée est de modéliser l'ensemble de données comme une surface paramétrée très complexe, vivant dans un espace de grande dimension. Les méthodes classiques utilisent l'*Analyse en Composantes Principales*, qui réduit la dimension de façon linéaire, et ne permet pas d'analyser correctement une surface courbe. Les algorithmes présentés du type *LLE* et *IsoMap* proposés récemment permettent au contraire d'extraire cette structure complexe.

Dans les deux derniers chapitres, nous avons présenté quelques applications de l'approche de catégorisation par marches aléatoires sur graphe. Une première série de tests concerne l'extraction des composantes faciales et leurs catégorisations. Il faut pouvoir analyser de nombreux paramètres : illumination, pose, déformations ... Ensuite, nous avons abordé le problème d'extractions d'indices caractérisant les mouvements humains (vidéos), et la recherche de classes de comportements à partir de ces indices. L'idée consiste à modéliser cette masse d'images/vidéos comme une surface paramétrée très complexe, vivant dans un espace de grande dimension. L'approche de catégorisation par marches aléatoires sur graphe

constitue un cadre unifié à la réduction de dimensions, au traitement des données, leur organisation et leur visualisation.

Nous avons présenté un cadre complet pour l'apprentissage non supervisé de variétés non linéaires et pour la projection de nouveaux points sur cette variété. La méthode consiste en deux étapes, la première étant un processus de réduction des données permettant d'obtenir des coordonnées réduites, la seconde étant une mise en correspondance linéaire entre les coordonnées réduites et les coordonnées originales, permettant une projection sur la variété. Les performances de la méthode proposée méritent d'être comparée à d'autres méthodes de classification non-supervisés .

6.2 Perspectives

De nombreuses perspectives concernant chaque étape de traitement ont été présentées dans les chapitres correspondants. Elles concernent généralement les limitations de ces étapes et décrivent des solutions ou des approches qui pourraient améliorer les résultats relatifs à chaque étape. Nous ne les rappellerons donc pas ici. Nous allons présenter quelques perspectives concernant les deux approches : méthode d'extraction active d'objets 2d+t et la méthode de catégorisation par marches aléatoires sur graphe.

D'une part, les contours actifs constituent un modèle pertinent des objets caractérisant bien leur forme et leur surface. En utilisant des descripteurs de forme invariants tels que les moments de Legendre, nous pouvons étendre notre modèle pour qu'il soit capable de contraindre l'évolution d'un contour actif vers une forme de référence (un a priori), augmentant la robustesse de la segmentation en présence de bruit, de fonds texturés et d'occultations.

D'autres améliorations peuvent être apportées à la catégorisation, notamment avec l'utilisation de méthodes statistiques d'apprentissage actif qui font partie du vaste champ de recherche qu'est l'intelligence artificielle. L'idée est de propager quelques vérités terrain dans le voisinage pour aider à une convergence rapide ou sûre. D'autre part, l'approche spectrale proposée peut fournir une série de solutions nouvelles pour différents domaines :

- Catégorisation de documents (texte/image/son ...)
- Analyse d'événement dans une vidéo,
- Fouille du Web: réseau sociaux,
- Intelligence économique

6.3 Publications

- Conférences internationales avec actes et comité de lecture

[1] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Visual Object Detection Using General Gaussian in Active Region Model. IEEE, International Conference on Complex Systems, Intelligence and Modern Technology Applications - CSIMTA . Cherbourg, France, September 19-22, (2004)

[2] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Visual object detection by an active region model, 3ème Conférence Internationale IEEE : Sciences Electroniques, Technologies de l'Information et des Télécommunications - IEEE SETIT,ISBN 9973-41-902-2, actes sur CDROM,27-31 Mars , Sousse, Tunisie, (2005)

[3] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Moving object Segmentation using optical flow with active contour model,
The International Conference on Information & Communication Technologies: from Theory to Applications , ICTTA'08, April 7 - 11, Damascus, Syria, (2008)

[4] Youssef CHAHIR, Youssef ZINBI, Abderrahim ELMOATAZ. A random walk through human behavior, IS&T/SPIE, Int. Conf. Multimedia Content Access: Algorithms and Systems III , January 2009

- Conférences nationales avec actes et comité de lecture

[5] Youssef ZINBI, Youssef CHAHIR et Abder ELMOATAZ, Extraction d'objets vidéo : une approche combinant les contours actifs et le flot optique Sixièmes journées Extraction et Gestion des Connaissances,- EGC'06 Revue des Nouvelles Technologies de l'Information, ISBN 2-85428-718-5, pp. 41-46 , Lille, France, 17-20 janvier (2006).

[6] Youssef CHAHIR, Youssef ZINBI, Kheir-Edine AZIZ, Catégorisation des expressions faciales par marches aléatoires sur graphe 12 èmes journées d'étude et d'échange Compression et représentation des Signaux Audiovisuels Montpellier, 8-9 novembre, (2007)

- Workshop avec comité de lecture

[7] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Video Structuring by Diffusion Maps,Spectral and fuzzy clustering techniques: application to signal and image segmentation. ClasSpec'2008 , Lens – October, 2008

- Autres communications

[8] Suivi d'objets visuels par contour actif. Journée du GDR, Indexation et Recherche d'Informations, 24 Septembre 2004, INSA Lyon.

[9] Segmentation active d'objets visuels. Réunion de l'équipe Données, Document et Langue, 17 juin 2005, GREYC, Caen.

CHAPITRE 7 Références

7 Références

[Ahlvers] U. Ahlvers, U. Zölzer, R. Rajagopalan. "Model-free Face Detection and Head Tracking with Morphological Hole Mapping", EUSIPCO'05, Antalya, Turkey.

[Baker77] Baker, C. (1977), The numerical treatment of integral equations, Oxford: Clarendon Press.

[Bartlett03] M. S. Bartlett, G. Littlewort, I. Fasel, J. R. Movellan, Real Time Face Detection and Facial Expression Recognition:Development and Applications to Human Computer Interaction, IEEE workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (in conjunction with IEEE Intl. Conf. on CVPR), Madison, U.S.A., June, 2003

[Bartlett98] M. Bartlett. Face image analysis by unsupervised learning and redundancy reduction. Springer, 1998.

[Belkin 03] Belkin, M. and Niyogi, P. (2003) Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", Neural Computation, Vol. 6, no. 15, pp. 1373–1396

[Belkin03] M. Belkin. and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation, 15(6) :1373–1396, 2003.

[Bellman57] R. Bellman. Dynamic programming. Princeton University Press, 1957.

[Black97] M. J. Black and Y. Yacoob, Recognizing Facial Expressions in Image Sequences Using Local Parametrized Models of Image Motion, International Journal of Computer Vision, Vol. 25, Number 1, pp. 23–48, 1997.

[Blank05] Blank, M. Gorelick, L. Shechtman, E. ,Irani, M. and Basri, R., (2005). Actions as space-time shapes, in ICCV

[Bobick01] Bobick, A. and J. Davis, "The recognition of human movement using temporal templates," IEEE Transaction on Pattern Analysis & Machine Intelligence, 23(3), March 2001.

[Boulay05] Boulay, B., Bremond, F., Thonnat, M., 2005. Posture recognition with a 3D human model. In: ICDP 2005, pp. 135–138.

[Bouveyron06] C Bouveyron, Modélisation et classification des données de grande dimension : application à l'analyse d'images. Thèse soutenue le 28/09/06 à l'université Joseph Fourier

[Bray06] M. Bray, P. Kholi, P. Torr. PoseCut: Simulataneous segmentation and 3d pose estimation of humans using dynamic graph cuts . In ECCV. 2006.

[Bregler97] C. Bregler, M. Covel, and M. Slaney, Video Rewrite: Driving Visual Speech with Audio, Siggraph proceedings, pp. 353–360, 1997.

[Carreira97] M. Carreira-Perpinan. A review of dimension reduction techniques. Technical Report CS-96-09, Dept. of Computer Science, University of Sheffield, 1997.

[Caselles95] Caselles (V.), Kimmel (R.) et Sapiro (G.). Geodesic active contours. In IEEE International conference on Computer Vision, pp. 694{699, Boston, USA, Juin 1995.

[Caselles97] Caselles (V.), Kimmel (R.) et Sapiro (G.). Geodesic active contours. International Journal of Computer Vision, 22:61{79, 1997.

[Chahir07a] Chahir, Y. Elmoataz, A. et Aziz, K. Caractérisation de la texture par marches aléatoires locales sur graphe, GRETSI , Troyes, 2007

[Chahir07b] Chahir,Y. Zinbi, Y. et Aziz, K. Catégorisation des expressions faciales par marches aléatoires sur graphe, CORESA , Montpellier, 2007

[Chahir09b] Chahir Y. Zinbi Y. et Elmoataz A. A random walk through human behavior, IS&T/SPIE, Int. Conf. Multimedia Content Access: Algorithms and Systems III , January 2009

[Chan01]T.F. Chan and L.A. Vese. Active contours without edges. IEEE Trans. on Image Processing, 10(2):266–277, February 2001.

[Chan02] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. Int. Journal of Computer Vision, 50(3):271–293, 2002.

[Chen95] D.T. Chen, A. State, and D. Banks, Interactive Shape Metamorphosis, Siggraph proceedings, pp. 43–44, 1995.

[Chuang] E. S. Chuang, H. Deshpande and C. Bregler, Facial Expression Space Learning.

[Chung97] F. R. K. Chung, Spectral graph theory, CBMS regional conference series in mathematics, AMS 1997

[Cohn 98] J. Cohn, A. Zlochower, and J. J. Lien and T. Kanade, Feature-point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression, Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 396–401, 1998.

[Cootes01] T.F. Cootes and G.J. Edwards and C.J. Taylor, Active Appearance Models, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 681–685,2001.

[Dai98] D. Q. Dai, P. C. Yuen and FENG G. C.: A multi-resolution decomposition method for human face recognition. Proceedings/actes Vision Interface '98, p. 301-307, Vancouvert, British columbia, June 1998.

[Delicado01] P. Delicado. Another look at principal curves and surfaces. Journal of Multivariate Analysis, 77(1) :84–116, 2001.

[Draper] B. Draper, K. Baek, M.S. Bartlett and R. Beveridge, Recognizing Faces with PCA and ICA, Computer Vision and Image Understanding

[Edwards98] G.J. Edwards, T.F. Cootes, and C.J. Taylor, Face Recognition Using Active Appearance Models, Proceedings of the European Conference of Computer Vision, pp. 581–695, 1998.

[Edwards98] J. Edwards, T.F. Cootes, and C.J. Taylor, Face Recognition Using Active Appearance Models, Proceedings of the European Conference of Computer Vision, pp. 581–695, 1998.

[Efros03] Efros, A. Berg, A. Mori, G. Malik, J. (2003) Recognizing action at a distance, in: ICCV

[Ekman] P. Ekman and W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement.

[Essa] I. Essa and A. Pentland, Coding, Analysis Interpretation Recognition of Facial Expressions, IEEE Transactions on Pattern Analysis andMachine Intelligence,

[Ezzat02] T. Ezzat, G. Geiger and T. Poggio, Trainable Videorealistic Speech Animation, ACM Transactions on Graphics (TOG), Vol. 21, Number 3, pp. 388–398,2002.

[Fodor02]I. Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Livermore, Canada, 2002.

[Fowlkes01] C. Fowlkes, S. Belongie, et J. Malik. Efficient spatiotemporal grouping using the Nyström method. In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 231–238, Kauai, Hawaii, 2001.

[Freedman05] D. Freedman, T. Zhang. Interactive Graph cut Based Segmentation With Shape Priors. Computer Science Department, Rensselaer Polytechnic Institue, Troy, NY 12180.

[Gastaud02] M. Gastaud et M. Barlaud. Video segmentation using active contours on a group of pictures. Dans IEEE International Conference on Image Processing, volume 2, pages 81-84, Rochester, N.Y., septembre 2002.

[Gav99] Gaveau, B. Lesne, A. Schulman, L.S. (1999) Spectral signatures of hierarchical relaxation, Physical Letters A: 258, pp 222-228

[Geng05] X. Geng, De-Chuan Zhan, and Zhi-Hua Zhou, Member, IEEE.Supervised Nonlinear Dimensionality Reduction for Visualization and Classification. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics.

[Girard05] S. Girard and S. Iovleff. Auto-associative models and generalized principal component analysis Journal of Multivariate Analysis, 93(1) :21–39, 2005.

[Gunn97] S. R. Gunn and M. S. Nixon. A robust snake implementation ; a dual active contour. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(1) :63–68, 1997.

[Guyon03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Journal of Machine Learning Research, 3 :1157–1182, 2003.

[Ham04] J. Ham, D. Lee, S. Mike, , and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In Proceedings of the Twenty First International Conference on Machine Learning, pages 369–376, 2004.

[Hammal06] Zakia Hammal, Facial Features Segmentation, Analysis and Recognition of Facial Expressions using the Transferable Belief Model, thèse de Zakia Hammal, Juin 2006.

[Hammami05] These Mohamed Hammamil Modèle de peau, classification sémantique d'images et filtrage intelligent. Co-supervised with Pr. Liming Chen (LIRIS, Ecole Centrale de Lyon)

[Haritaoglu98] Haritaoglu, I. Harwood, D. and Davis, L. S. (1998) A Real Time System for Detecting and Tracking People, CVPR'98 Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,p:962, Washington, DC, USA

[Hastie01] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. Springer, New York, 2001.

[Hastie89] T. Hastie and W. Stuetzle. Principal curves. Journal of the American Statistical Association, 84 :502–516, 1989.

[Heisele01] B. Heisele, P. Ho and T. Poggio, Face Recognition with Support Vector Machines: Global Versus Component-based Approach, , International Conference on Computer Vision (ICCV'01), Vancouver, Canada,Vol. 2, pp. 688–694, 2001.

[Higgs06] B.W. Higgs, J. Weller,. and , J.L. Solka (2006) Spectral embedding finds meaningful (relevant) structure in image and microarray data, BMC Bioinformatics, Vol 7, 74-84

[Hoch94] M. Hoch, G. Fleischmann, and B. Girod, Modeling and Animation of Facial Expressions based on BSplines, The Visual Computer, pp. 87–95, 1994.

[Hong98]. Hong, H. Neven and C. von der Malsburg, Online Facial Expression Recognition based on Personalized Gallery, Intl. Conference on Automatic Face and Gesture Recognition, IEEE Comp. Soc, pp. 354–359, 1998.

[Horn81] Horn, B.K.P. and Schunk, B.G., "Determining Optical Flow", Artificial Intelligence, Vol. 17, pp. 185-201, 1981.

[Ingve04] Ingve, S. et al. (2004)Diffusion on complex networks : a way to probe their large-scale topological structures. Physica A: Statistical Mechanics and its Applications, 336(1-2) : 163–173

[Irani96] M. Irani, P. Anandan, J. Bergen, R. Kumar, et S. Hsu. Efficient representations of video sequences and their applications. Signal Processing :Image Communication, 8(4) :327-351, mai 1996.

[Iwas00] Iwasawa, S. et al. (2000) Real-time, 3D Estimation of Human Body Postures from Trinocular Images,' Faculty of engineering, Seikei University

[Jehan-Besson01] S. Jehan-Besson, M. Barlaud, G. Aubert ,Contours actifs basés régions pour la segmentation des objets en mouvement dans les séquences à caméra fixe ou mobile, GRETSI, Toulouse, september 2001.

[Jehan-Besson03] S. Jehan-Besson, M. Barlaud, and G. Aubert. Dream2s: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. Int. Journal of Computer Vision, 53(1):45–70, 2003.

[Jianhong06] Jianhong (Jackie) Shen. A stochastic-variational model for soft mumford-shah segmentation. International Journal of Biomedical Imaging, 2006(92329).

[Jones02] M. J. Jones, J. M. Rehg. Statistical Color Models with Application to Skin Detection, International Journal of Computer Vision, v. 46, (2002), 81 p.

[Kang98] H. Kang, T.F. Cootes and C.J. Taylor, Face Expression Detection and Synthesis using Statistical Models of Appearance, Measuring Behavior, pp. 126–128, 2002.

[Kass88] M. Kass, A. Witkin, et D. Terzopoulos. Snakes : Active contour models. International Journal of Computer Vision, 1 :321-332, 1988.

[Kheir06] Kheir-Eddine AZIZ, Diffusion géométrique par marche aléatoires sur graphe: Applications en analyse d'image et à la réduction de données de grande dimension Rapport de Master LID- 2006). GREYC , Caen

[Krieg02] M. H. Yang, D. Kriegman, and N. Ahuja, Detecting Faces in Images: a Survey, IEEE transactions on pattern analysis and machine intelligence, January 2002.

[Lafon05] S. Lafon, R. R. Coifman, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker.Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. 7426–7431,PNAS,May 24, 2005 . Vol. 102 , no. 21

[Lafon06] Lafon, S. Keller Y. and Coifman. R. (2006) Data Fusion and Multi-Cue Data Matching by Diffusion Maps. In IEEE transactions on Pattern Analysis and Machine Intelligence, Vol 28, 11 , 1784-1797

[Lanitis 97] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7) :743–756, 1997.

[Levy06] Lévy, B. , Laplace-Beltrami Eigenfunctions: Towards an algorithm that 'understands' geometry, in IEEE International Conference on Shape Modeling and Applications p. 13

[Lien98] J. J. Lien. Automatic recognition of facial expressions using hidden Markov models and estimation of expression intensity. Pittsburgh, Pa. : Carnegie Mellon University, The Robotics Institute, 1998.

[Louw07] Louw Lioyd A.B , Automated face detection and recognition for a login system, (2007), Master of Science Engineering at the University of Stellenbosch

[Lyons98].J. Lyons and J. Budynek and S. Akamatsu, Automatic Classification of Single Facial Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, pp. 1357–1362, 1999.

[Manor01] Lihi Zelnik-Manor, Michal Irani: Event-Based Analysis of Video. CVPR (2) 2001: 123-130

[McInerney95] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In ICCV, pages 840–845, 1995.

[Mehrabian 68] A. Mehrabian, Communication without Words, Psychology Today, Vol. 2, Number 4, pp. 53–56, 1968.1

[Mémin02] E. Mémin et P. Pérez. Hierarchical estimation and segmentation of dense motion fields. International Journal of Computer Vision, 46(2) :129-155, 2002.

[Mercier06] H. Mercier, Analyse automatique des expressions du visage. Application à la Langue des Signes.Travail de DEA 2006, TOULOUSE.FRANCE.

[Mokhber 05] Mokhber, A. Achard, C. Qu, X. and Milgram, M. (2005) Action Recognition with Global Features, A. Human Computer Interaction, Workshop of ICCV

[Mumford89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. Comm. on Pure and Applied Mathematics, 42(5):577–685, 1989.

[Nikolova04] T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. UCLA CAM Report 04-54, 2004.

[Odo03] Jean-marc Odobez, Daniel Gatica-perez, Mael Guillemot. VIDEO SHOT CLUSTERING USING SPECTRAL METHODS, In Proc. of 3rd International Workshop on Content-Based Multimedia Indexing (CBMI)

[Osher88] S. Osher et J.A. Sethian. Fronts propagating with curvature-dependent speed : Algorithms based on hamilton-jacobi formulation. Journal of Computational Physics, 79 :12-49, 1988.

[Padgett01] C. Padgett and G. Cottrell. Representing face images for emotion classification.Advances in Neural Information Processing Systems, pages 894–900, 1997.

[Padgett97] C. Padgett and G. Cottrell. Identifying emotion in static images. In Proceedings of the 2nd Joint Symposium on Neural Computation, volume 5, pages 91–101, 1997.

[Padgett98] C. Padgett, G. Cottrell and R. Adolphs, Categorical Perception in Facial Emotion Classification, Siggraph proceedings, pp. 75–84 1998.

[Panini03] L. Panini and R. Cucchiara, A machine learning approach for human posture detection in domestics applications, ICIAP (2003), pp. 103–108

[Paragios00] N. Paragios et R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(3) :266-280, 2000.

[Paragios02] N. Paragios and N. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. Int. Journal of Computer Vision, 46(3):223–247, 2002.

[Peer99] P. Peer, F. Solina. An automatic human face detection method, Proc. of 4th Computer Vision Winter Workshop (CVWW), Rastenfeld (1999), 122 page

[Penev96] P. Penev and J. Atick, Local Feature Analysis: A general Statistical Theory for Object Representation, 1996.

[Pighin96] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D.H. Salesin, Synthesizing Realistic Facial Expressions from Photographs, Proceedings of Cognitive Science Conference, 1996.

[Platt04] J. Platt. Fast embedding of sparse similarity graphs. Advances in Neural Information Processing Systems,2004.

[Ranchin04] F. Ranchin et F. Dibos. Moving objects segmentation using optical flow estimation. Dans Mathematics, Image and Analysis, Paris, France, 2004.

[Robles04] A. Robles-Kelly and E.R. Hancock. 3string Edit Distance, Random Walks and graph matching", IJPRAI, 18(3), pages 315-327, 2004

[Roweis00] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500) :2323–2326, 2000.

[Salembier99] P. Salembier et F. Marquès. Region-based representations of image and video : segmentation tools for multimedia services. IEEE Transactions on Circuits and Systems for Video Technology, 9(8) :1-20, décembre 1999.

[Sapiro01] G. Sapiro, Geometric partial differential equations and image analysis, Cambridge University Press, 2001.

[Schölkopf98] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10 :1299–1319, 1998.

[Scott83] D. Scott and J. Thompson. Probability density estimation in higher dimensions. In Fifteenth Symposium in the Interface, pages 173–179, 1983.

[Sifakis01] E. Sifakis et G. Tziritas. Moving object localisation using a multilabel fast marching algorithm. Signal Processing : Image Communication, 16 :963-976, 2001.

[Staib92] L.H Staib et J.S. Duncan. Boundary finding with parametrically deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 14(11) :1061-1075, 1992.

[Tavakoli06] A. T. Tavakoli, A.Shademan: Clustering of singular value decomposition of image data with applications to texture classification. In: VCIP. (2003) 972–979

[Tenenbaum00] J. Tenenbaum, V. De Silva, and J. Langford. A global geometric framework for non linear dimensionality reduction. Science, 290(5500) :2319–2323, 2000.

[Tsai01] A. Tsai, A. Yezzi Jr., and A.S. Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. IEEE Trans. on Image Processing, 10(8):1169–1186, 2001.

[Turk91] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1) :71–86, 1991.

[Webb02] A. Webb. Statistical pattern recognition. Wiley, New York, 2 edition, 2002.

[Williams00]C Williams, M. SEEGER (2000), The effect of the input density distribution on kernel based classifiers, Proceedings of the seventeeth International Conference on machine Learning, Morgan Kaufmann,

[Williams01] C. Williams, .(2001), On a connection between kernel psa and metric multidimensional scaling, Advances in Neural Information Processing Systems, The MIT Press, p.675-681.

[Wu93] S. F. Wu et J. Kittler. A gradient-based method for general motion estimation and segmentation. Journal of Visual Communication and Image Representation, 4(1) :25-38, mars 1993.

[Yang02] M. H. Yang, Face Recognition Using Kernel Methods, Advances in Neural Information Processing Systems 14 (NIPS 14), pp. 215–220, 2002.

[Zee05] Zeeshan Rasheed and Mubarak Shah, Fellow, Detection and Representation of Scenes in Videos IEEE Trans. On Multimedia, Vol. 7, no. 6, December, 2005

[Zhan04]. Ye, Y. Zhan, and S. Song. Facial expression features extraction based on gabor wavelet transformation. IEEE International Conference on Systems, Man and Cybernetics, 2004.

[Zhang01] D. Zhang. Segmentation of moving objects in image sequence : A review. Circuits, Systems, and Signal Processing, 20(2) :143-183, 2001.

[Zhao00] W. Zhao, R. Chellappa, A. Rosenfeld and P.J. Phillips. Face recognition: A literature survey. CVL Technical Report, University of Maryland,, October 2000.

[Zhao01] Zhao, T. Nevatia R. and Lv, F. (2001) Segmentation and Tracking of Multiple Humans in Complex Situations," Computer Vision and Pattern Recognition

[Zhu96] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. On Pattern Analysis and Machine Intelligence, 18(9):884–900, September 1996.

[Zinbi04] Youssef CHAHIR, Youssef ZINBI and Abder ELMOATAZ, Visual Object Detection Using General Gaussian in Active Region Model. IEEE, International Conference on Complex Systems, Intelligence and Modern Technology Applications - CSIMTA . Cherbourg, France, September 19-22, (2004)

[Zinbi05] Youssef CHAHIR, Youssef ZINBI and Abder ELMOATAZ, Visual object detection by an active region model, 3ème Conférence Internationale IEEE : Sciences Electroniques, Technologies de l'Information et des Télécommunications - IEEE SETIT,ISBN 9973-41-902-2, actes sur CDROM,27-31 Mars ,Â Sousse, Tunisie, (2005)

[Zinbi06] Youssef ZINBI, Youssef CHAHIR et Abder ELMOATAZ, Extraction d'objets vidéo :une approche combinant les contours actifs et le flot optique Sixièmes journées Extraction et Gestion des Connaissances,- EGC'06 Revue des Nouvelles Technologies de l'Information, ISBN 2-85428-718-5, pp. 41-46 , Lille, France, 17-20 janvier (2006).

[Zinbi08a] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Moving object Segmentation using optical flow with active contour model, The International Conference on Information & Communication Technologies: from Theory to Applications , ICTTA'08, April 7 - 11, Damascus, Syria, (2008)

[Zinbi08b] Youssef ZINBI, Youssef CHAHIR and Abder ELMOATAZ, Video Structuring by Diffusion Maps,Spectral and fuzzy clustering techniques: application to signal and image segmentation . ClasSpec'2008 , Lens – October, 2008