Approche EM pour la construction de régions de teinte homogènes

EM approach for construction of homogeneous color regions

Sébastien Marcel Olivier Bernier Daniel Collobert

France Télécom CNET

2, Avenue Pierre Marzin, 22307 Lannion cedex
sebastien.marcel@cnet.francetelecom.fr

Résumé

Ce papier décrit un modèle probabiliste gaussien de régions homogènes de teinte chair (blob). Les paramètres des blobs sont calculés par un algorithme EM (Expectation-Maximisation) et ces derniers sont fusionnés sur un critère spatio-colorimétrique. Les blobs obtenus sont moins nombreux et plus stables que les zones de pixels de teinte chair connexes dont les blobs sont issus.

Mots Clef

teinte chair, algorithme EM, régions homogènes, fusion.

Abstract

This paper present a probabilistic gaussian model of homogeneous skin color regions (blob). Blobs parameters are adaptated by an Expectation-Maximisation algorithm and blobs are merged using a space-color criteria. The obtained blobs are less numerous and more stable than the connex skin color pixel regions from which the blobs are issued.

Keywords

skin color, EM algorithm, homogeneous regions, merging.

1 Introduction

La détection et l'analyse des personnes est un problème fondamental en vision par ordinateur. Généralement, les objets importants tels que le visage ou les mains sont détectés à l'aide de la couleur de la peau. Il existe de nombreux modèles pour représenter et segmenter la couleur de la peau, mais encore faut-il regrouper les pixels pour former des régions pertinentes autour des objets et éviter que ces dernières ne fusionnent entre elles ou avec le fond.

Ce papier décrit une formalisation des blobs définissant un modèle probabiliste gaussien dont les paramètres sont ajustés aux données de l'image par une procédure EM. Les séquences d'images que nous traitons sont issues d'une caméra , sont de type visiophone et sont au format CIF (384x288 pixels).

2 Le traitement de l'image

Dans ces images, nous nous intéressons à des objets particuliers, les visages et les mains (figure 1). Ces derniers possèdent une teinte caractéristique qu'il est possible de séparer du reste de l'image [13].

FIG. 1 – *Image de type visiophone à analyser*

2.1 Le filtrage de la teinte chair

Il existe de nombreuses méthodes pour modéliser la teinte chair, principalement à l'aide de gaussiennes [12] ou de mélanges de gaussiennes [11]. De plus, divers espaces colorimétriques RVB [12], HSV [11] ou Lab [8] peuvent être utilisés. Nous avons choisi l'espace colorimétrique YUV, car c'est le format issu de la caméra et il est relativement robuste au changement de luminosité grâce à la séparation de la luminance (Y) de la chrominance (UV).

Nous avons choisi de filtrer l'image en utilisant une table de correspondance des pixels de teinte chair (figure 2). Cette technique bien que rudimentaire est efficace et très rapide.

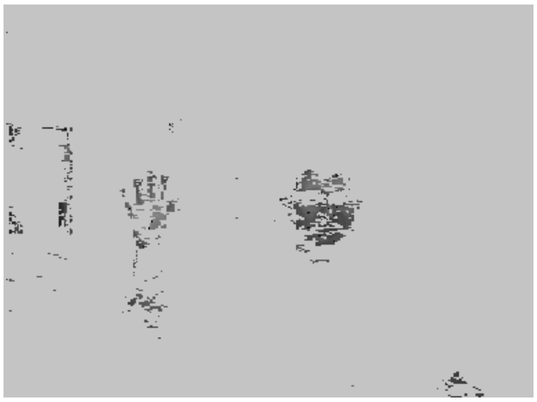

FIG. 2 – *Image des pixels de teinte chair*

Une fois les pixels de teinte chair séparés du reste de l'image, nous cherchons à les regrouper en régions.

2.2 Zonage par regroupement des pixels connexes

Une première approche pour regrouper les pixels est de tenir compte de leur connexité. Un algorithme d'expansion permet de déterminer des zones de pixels de teinte chair connexes (figure 3). De plus, on élimine les zones de hauteur ou de largeur inférieure à 10 pixels. Ceci explique pourquoi certaines zones n'apparaissent pas dans le fond (figure 3 et 4).

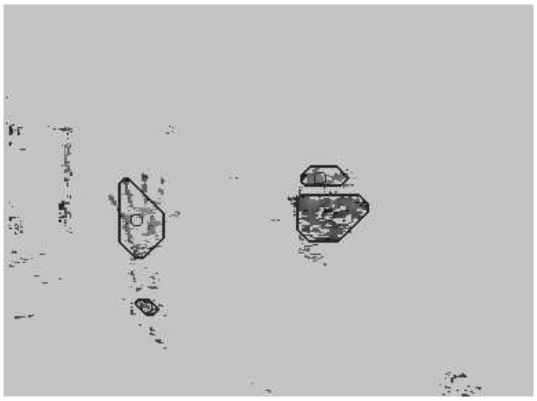

FIG. 3 – *Zones connexes de teinte chair à l'image i*

Cependant, le bruit de la caméra et la variation de l'illumination dans la scène rend ces zones géométriquement instables. Certaines zones apparaissent ou disparaissent, d'autres peuvent changer de taille ou fusionner ensemble (figure 3 et 4).

Il est nécessaire d'utiliser une méthode non plus basée uniquement sur la connexité mais sur les composantes spatiales et colorimétriques des pixels pour déterminer des régions homogènes.

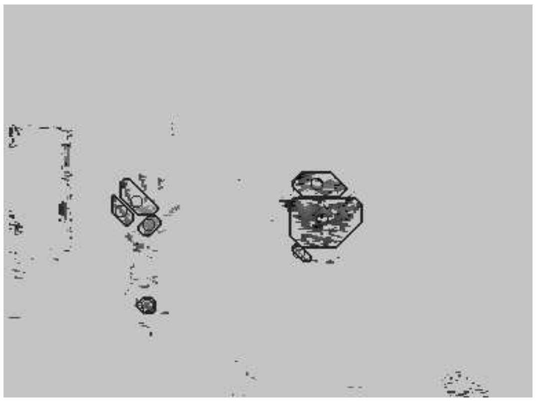

FIG. 4 – *Zones connexes de teinte chair à l'image i+1 : d'une image à l'autre le nombre de zones et leur forme changent (cf. la main)*

3 Les blobs

De nombreux modèles de blobs ont déjà été utilisés pour la détection des personnes [10], en reconnaissance des gestes de la main [7] ou pour représenter des modèles déformables [9].

3.1 Le modèle des blobs

Dans cette section, nous définissons un blob comme un modèle probabiliste gaussien sur les dimensions spatiales (x,y) et colorimétriques (Y,U,V) de ses pixels de teinte chair.

Introduisons les notations suivantes :

- P : nombre de pixels de teinte chair dans l'image,
- $\mathcal{I} = \{\mathbf{x}_i\}$: l'ensemble des pixels de teinte chair de l'image, avec $i = 1 \ldots P$
- $\mathbf{x}_i = {}^T[x_i, y_i, u_i, v_i, l_i]$
  - x_i, y_i : position du pixel dans l'image,
  - u_i, v_i : chrominance du pixel dans l'image,
  - l_i : luminance du pixel dans l'image.
- B : nombre de blobs fixé au départ,
- $\mathcal{B} = (\{\boldsymbol{\mu}_j\}, \{\boldsymbol{\sigma}_j\})$: ensemble des blobs, avec $j = 1 \ldots B$
  - $\boldsymbol{\mu}_j$: vecteur centre des pixels du blob j,
  - $\boldsymbol{\sigma}_j$: inverse de la matrice de covariance des pixels du blob j,
- $\boldsymbol{\Omega}_j$: ensemble des pixels du blob j.

Dans ce modèle, on suppose connu a priori le nombre de blobs B. On recherche les paramètres $\{\boldsymbol{\mu}_j\}$ et $\{\boldsymbol{\sigma}_j\}$ qui permettent d'obtenir les meilleurs blobs étant donnés les pixels de teinte chair procurés par le filtrage de l'image, c'est-à-dire ceux qui maximisent la vraisemblance des pixels observés.

3.2 La vraisemblance des blobs

Soit Θ la collection des paramètres $\{\boldsymbol{\mu}_j\}$ et $\{\boldsymbol{\sigma}_j\}$. On peut alors introduire la fonction de vraisemblance L (Equation 1) des paramètres des blobs Θ sur l'ensemble des données $\mathcal{I}$, c'est-à-dire les pixels de teinte chair :

$$\begin{aligned} L(\Theta, \mathcal{I}) &= \Pr(\mathcal{I} \mid \Theta) \\ &= \prod_{i=1}^{P} \Pr(\mathbf{x}_i \mid \Theta) \end{aligned} \tag{1}$$

L'équation 1 peut être maximisée par l'algorithme EM.

3.3 L'algorithme EM

L'algorithme EM (Expectation Maximisation) [3] cherche à maximiser la fonction de log-vraisemblance l (Equation 2) sur les paramètres Θ, l'ensemble des données $\mathcal{I}$ étant donné.

$$l(\Theta, \mathcal{I}) = log L(\Theta, \mathcal{I}) \tag{2}$$

Pour simplifier ce problème, l'hypothèse EM est d'introduire un nouvel ensemble de paramètres $\mathcal{H}$ dits cachés. Ainsi, nous obtenons un nouvel ensemble de données $\mathcal{D} = (\mathcal{I}, \mathcal{H})$, appelé l'ensemble complet des données, de fonction de log-vraisemblance $l(\Theta, \mathcal{D})$. Cependant, cette fonction ne peut pas être maximisée directement car $\mathcal{H}$ est inconnu.

Il a été déjà démontré [3] que l'estimation itérative de la fonction auxiliaire Q (Equation 3) maximise $l(\Theta, \mathcal{D})$ en utilisant les paramètres $\hat{\Theta}$ de l'itération précédente.

$$Q(\Theta, \hat{\Theta}) = E_{\mathcal{H}}[l(\Theta, \mathcal{D}) \mid \mathcal{I}, \hat{\Theta}] \tag{3}$$

L'algorithme EM est le suivant :

- Pour $k = 1 \dots K$
  - Estimation : calcul de $Q(\Theta, \Theta^{(k-1)}) = E_{\mathcal{H}}[l(\Theta, \mathcal{D}) \mid \mathcal{I}, \Theta^{(k-1)}]$
  - Maximisation : $\Theta^{(k)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(k-1)})$

Le calcul analytique n'est pas toujours possible. On tente alors d'estimer les paramètres Θ indirectement, on parle d'algorithme GEM (Generalized Expectation Maximisation). En effet, il a été démontré qu'il est possible de garantir la convergence vers un maximum local de la fonction de vraisemblance en réalisant une mise à jour des paramètres $\Theta^{k+1} = M(\Theta^k)$ où la fonction $M(.)$ est telle que $Q(M(\Theta^{k+1}), \Theta^k) \geq Q(\Theta^k, \Theta^k)$ [3] [1].

Dans cet article, nous avons choisi de développer la maximisation analytique en cherchant à annuler les dérivées partielles $\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\mu}_j}$ et $\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\sigma}_j}$.

3.4 Ajustement des blobs

Les variables cachées.

L'ajout de variables cachées doit faciliter la résolution du problème. Dans notre cas, ces variables cachées vont nous permettre d'associer un pixel à un et un seul blob. Cette représentation est celle des cartes de support [10]. Soit les paramètres cachés $\mathcal{H} = h_1 \dots h_P$, où h_i est le blob associé au pixel i. Ainsi, nous avons l'ensemble complet des données $\mathcal{D}$ défini par :

$$\mathcal{D} = (\mathcal{I}, \mathcal{H}) = (\mathbf{x}_1 \dots \mathbf{x}_P, h_1 \dots h_P)$$

La fonction de vraisemblance sur $\mathcal{D}$ est :

$$\begin{aligned} L(\Theta, \mathcal{D}) &= \Pr(\mathcal{I}, \mathcal{H} \mid \Theta) \\ &= \prod_{i=1}^{P} \Pr(\mathbf{x}_i, h_i \mid \Theta) \\ &= \prod_{i=1}^{P} \Pr(\mathbf{x}_i \mid h_i, \Theta) \Pr(h_i \mid \Theta) \end{aligned} \tag{4}$$

Posons z_{ij} la variable aléatoire définie par :

$$z_{ij} = \begin{cases} 1 & : \quad h_i = j \\ 0 & : \quad h_i \neq j \end{cases}$$

alors on peut réécrire l'équation 4 comme :

$$L(\Theta, \mathcal{D}) = \prod_{i=1}^{P} \prod_{j=1}^{B} \Pr(\mathbf{x}_i \mid h_i = j, \Theta)^{z_{ij}} \Pr(h_i = j \mid \Theta)^{z_{ij}}$$

La log-vraisemblance est :

$$\begin{aligned} l(\Theta, \mathcal{D}) &= \log L(\Theta, \mathcal{D}) \\ &= \sum_{i=1}^{P} \sum_{j=1}^{B} z_{ij} \left[\log \Pr(\mathbf{x}_i \mid h_i = j, \Theta) \right. \\ &+ \left. \log \Pr(h_i = j \mid \Theta) \right] \end{aligned} \tag{5}$$

Etape d'Estimation.

En utilisant l'équation 3 et l'équation 5, la fonction auxiliaire s'écrit :

$$\begin{aligned} Q(\Theta, \hat{\Theta}) &= E_{\mathcal{H}}[l(\Theta, \mathcal{D}) \mid \mathcal{I}, \hat{\Theta}] \\ &= \sum_{i=1}^{P} \sum_{j=1}^{B} E_{\mathcal{H}}[z_{ij} \mid \mathcal{I}, \hat{\Theta}] \\ &\quad [\log \Pr(\mathbf{x}_i \mid h_i = j, \Theta) + \log \Pr(h_i = j \mid \Theta)] \\ &= \sum_{i=1}^{P} \sum_{j=1}^{B} \Pr(h_i = j \mid \mathcal{I}, \hat{\Theta}) \\ &\quad [\log \Pr(\mathbf{x}_i \mid h_i = j, \Theta) + \log \Pr(h_i = j \mid \Theta)] \end{aligned}$$

Nous avons supposé précédemment qu'un pixel appartient à un et un seul blob. Nous supposons alors que tous les blobs sont équiprobables d'où :

$$\Pr(h_i = j \mid \Theta) = \frac{1}{B}$$

De plus, on pose que la fonction de densité de probabilité d'un pixel de teinte chair $\mathbf{x}_i$ connaissant le blob j est une gaussienne, ainsi :

$$\begin{aligned}\Pr(\mathbf{x}_i \mid h_i = j, \Theta) &= \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j) \\ &= \sqrt{\frac{\mid \boldsymbol{\sigma}_j \mid}{(2\pi)^5}}\, e^{-\frac{1}{2}{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)}\end{aligned}$$

D'autre part :

$$\begin{aligned}Pr(h_i = j \mid \mathcal{I}, \hat{\Theta}) &= \frac{\Pr(h_i = j, \mathcal{I} \mid \hat{\Theta})}{\Pr(\mathcal{I} \mid \hat{\Theta})} \\ &= \frac{\Pr(\mathcal{I} \mid h_i = j, \hat{\Theta})\Pr(h_i = j \mid \hat{\Theta})}{\Pr(\mathcal{I} \mid \hat{\Theta})} \\ &= \frac{1}{B}\frac{\prod_{k=1}^{P}\Pr(\mathbf{x}_k \mid h_i = j, \hat{\Theta})}{\prod_{k=1}^{P}\Pr(\mathbf{x}_k \mid \hat{\Theta})}\end{aligned}$$

or $\forall k \neq i, \Pr(\mathbf{x}_k \mid h_i = j, \hat{\Theta}) = \Pr(\mathbf{x}_k \mid \hat{\Theta})$ car un pixel appartient à un et un seul blob, d'où :

$$\begin{aligned}Pr(h_i = j \mid \mathcal{I}, \hat{\Theta}) &= \frac{1}{B}\frac{\Pr(\mathbf{x}_i \mid h_i = j, \hat{\Theta})}{\Pr(\mathbf{x}_i \mid \hat{\Theta})} \\ &= \frac{1}{B}\frac{\Pr(\mathbf{x}_i \mid h_i = j, \hat{\Theta})}{\sum_{l=1}^{B}\Pr(\mathbf{x}_i \mid h_i = l, \hat{\Theta})}\end{aligned}$$

Finalement, nous avons :

$$\begin{aligned}Q(\Theta, \hat{\Theta}) &= \sum_{i=1}^{P}\sum_{j=1}^{B}\frac{\Pr(\mathbf{x}_i \mid h_i = j, \hat{\Theta})}{\sum_{l=1}^{B}\Pr(\mathbf{x}_i \mid h_i = l, \hat{\Theta})} \\ &\quad \left[\log \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j) + \log\frac{1}{B}\right] \\ &= \sum_{i=1}^{P}\sum_{j=1}^{B} E_{ij}\, \log \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j) \\ &+ \sum_{i=1}^{P}\sum_{j=1}^{B} E_{ij}\, \log\frac{1}{B} \qquad (6)\end{aligned}$$

avec :

$$E_{ij} = \frac{\mathcal{G}(\mathbf{x}_i, \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\sigma}}_j)}{\sum_{l=1}^{B}\mathcal{G}(\mathbf{x}_i, \hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\sigma}}_l)} \qquad (7)$$

L'étape d'estimation de l'algorithme EM s'achève sur le calcul de l'espérance des variables cachées E_{ij}, sachant les données $\mathcal{I}$ et les paramètres précédents $\hat{\Theta}$.

Si nous isolons ce terme, c'est par commodité de calcul car il va se propager tout au long de l'étape de maximisation.

Etape de Maximisation.

On recherche les paramètres Θ de sorte à maximiser la fonction auxiliaire $Q(\Theta, \hat{\Theta})$. Pour cela, on calcule les paramètres $\boldsymbol{\mu}_j$ et $\boldsymbol{\sigma}_j$ tels que $\frac{\partial Q(\Theta,\hat{\Theta})}{\partial \boldsymbol{\mu}_j} = 0$ et $\frac{\partial Q(\Theta,\hat{\Theta})}{\partial \boldsymbol{\sigma}_j} = 0$.

Commençons par dériver $Q(\Theta, \hat{\Theta})$ par rapport à $\boldsymbol{\mu}_j$:

$$\begin{aligned}\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\mu}_j} &= \sum_{i=1}^{P} E_{ij}\frac{\partial \log \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}{\partial \boldsymbol{\mu}_j} \\ &= \sum_{i=1}^{P} E_{ij}\frac{\partial \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}{\partial \boldsymbol{\mu}_j} \\ &= -\frac{1}{2}\sum_{i=1}^{P} E_{ij}\frac{\partial\left({}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)\right)}{\partial \boldsymbol{\mu}_j} \\ &= \frac{1}{2}\sum_{i=1}^{P} E_{ij}\left[\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j) + {}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j\right]\end{aligned}$$

or $\boldsymbol{\sigma}_j$ est symétrique d'où :

$$\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\mu}_j} = \sum_{i=1}^{P} E_{ij}\,(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j$$

ainsi si l'on cherche à annuler les dérivées :

$$\begin{aligned}&\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\mu}_j} = 0 \\ \Leftrightarrow\ &\sum_{i=1}^{P} E_{ij}\,(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j = 0 \\ \Leftrightarrow\ &\boldsymbol{\sigma}_j\sum_{i=1}^{P} E_{ij}\,\mathbf{x}_i - \boldsymbol{\sigma}_j\sum_{i=1}^{P} E_{ij}\,\boldsymbol{\mu}_j = 0\end{aligned}$$

On obtient une somme des pixels pondérés par l'espérance E_{ij} :

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{P} E_{ij}\,\mathbf{x}_i}{\sum_{i=1}^{P} E_{ij}} \qquad (8)$$

A présent, il nous reste à trouver $\boldsymbol{\sigma}_j$. Pour cela, dérivons $Q(\Theta, \hat{\Theta})$ par rapport à $\boldsymbol{\sigma}_j$:

$$\begin{aligned}&\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\sigma}_j} \\ &= \sum_{i=1}^{P} E_{ij}\frac{\partial \log \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}{\partial \boldsymbol{\sigma}_j} \\ &= \sum_{i=1}^{P} E_{ij}\frac{1}{\mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}\frac{\partial \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}{\partial \boldsymbol{\sigma}_j}\end{aligned}$$

$$= \sum_{i=1}^{P} E_{ij} \frac{1}{\mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)}$$

$$[\frac{\frac{\partial\sqrt{|\boldsymbol{\sigma}_j|}}{\partial \boldsymbol{\sigma}_j}}{\sqrt{(2\Pi)^5}} e^{-\frac{1}{2}{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)}$$

$$+ \frac{\sqrt{|\boldsymbol{\sigma}_j|}}{\sqrt{(2\Pi)^5}} \frac{\partial e^{-\frac{1}{2}{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)}}{\partial \boldsymbol{\sigma}_j}]$$

Dans cette expression, il est nécessaire de dériver le déterminant de la matrice $\boldsymbol{\sigma}_j$. Soit A une matrice symétrique définie positive, alors on a :

$$\frac{\partial\sqrt{|A|}}{\partial A} = \frac{1}{2\sqrt{|A|}} \frac{\partial |A|}{\partial A}$$

de plus, on sait que :

$$\frac{\partial |A|}{\partial A} = |A| A^{-1}$$

d'où :

$$\frac{\partial\sqrt{|A|}}{\partial A} = \frac{1}{2}\sqrt{|A|} A^{-1} \tag{9}$$

ainsi $\frac{\partial Q(\Theta,\hat{\Theta})}{\partial \boldsymbol{\sigma}_j}$ devient :

$$\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\sigma}_j}$$

$$= \sum_{i=1}^{P} E_{ij} \frac{1}{\mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)} [\frac{1}{2}(\boldsymbol{\sigma}_j)^{-1}\mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)$$

$$+ \mathcal{G}(\mathbf{x}_i, \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j) \frac{\partial - \frac{1}{2}{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\boldsymbol{\sigma}_j(\mathbf{x}_i - \boldsymbol{\mu}_j)}{\partial \boldsymbol{\sigma}_j}]$$

$$= \frac{1}{2}\sum_{i=1}^{P} E_{ij} \left[(\boldsymbol{\sigma}_j)^{-1} - (\mathbf{x}_i - \boldsymbol{\mu}_j)\,{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\right]$$

ainsi :

$$\frac{\partial Q(\Theta, \hat{\Theta})}{\partial \boldsymbol{\sigma}_j} = 0$$

$$\Leftrightarrow \quad \frac{1}{2}\sum_{i=1}^{P} E_{ij} \left[(\boldsymbol{\sigma}_j)^{-1} - (\mathbf{x}_i - \boldsymbol{\mu}_j)\,{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)\right] = 0$$

$$\Leftrightarrow \quad (\boldsymbol{\sigma}_j)^{-1}\sum_{i=1}^{P} E_{ij} - \sum_{i=1}^{P} E_{ij}(\mathbf{x}_i - \boldsymbol{\mu}_j)\,{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j) = 0$$

On obtient :

$$\boldsymbol{\sigma}_j = \left(\frac{\sum_{i=1}^{P} E_{ij}\,(\mathbf{x}_i - \boldsymbol{\mu}_j)\,{}^T(\mathbf{x}_i - \boldsymbol{\mu}_j)}{\sum_{i=1}^{P} E_{ij}}\right)^{-1} \tag{10}$$

On obtient pour $\boldsymbol{\sigma}_j$ (Equation 10) l'inverse de la matrice de corrélation pondérée par les espérances E_{ij}.

L'algorithme EM appliqué aux blobs devient à présent :

- Pour k = 1 ...K, avec K étant le nombre d'itérations EM
  - Calculer les E_{ij} d'après l'équation 7,
  - Calculer les $\{\boldsymbol{\mu}_j\}$ d'après l'équation 8,
  - Calculer les $\{\boldsymbol{\sigma}_j\}$ d'après l'équation 10.

A présent, il nous manque une procédure pour initialiser les blobs avant de débuter l'algorithme EM.

4 Initialisation des blobs

On suppose que le nombre de blobs, qui est constant durant l'algorithme EM, doit être convenablement initialisé. Or on ne le connaît pas a priori. Aussi, nous choisissons de privilégier une stratégie qui sur-estime le nombre de blobs au début, mais qui permet de les fusionner par la suite. Ainsi, les blobs sont initialisés, tout d'abord, à partir des zones connexes de teinte chair de la manière suivante :

- $\boldsymbol{\mu}_j^{(0)}$: vecteur centre initial des pixels du blob j,

$$\boldsymbol{\mu}_j^{(0)} = \frac{1}{Card\,\Omega_j} \sum_{\mathbf{x}_k \in \Omega_j} \mathbf{x}_k$$

- $\boldsymbol{\sigma}_j^{(0)}$: inverse de la matrice de covariance initiale des pixels du blob j,

$$\boldsymbol{\sigma}_j^{(0)} = (\frac{1}{Card\,\Omega_j} \sum_{\mathbf{x}_k \in \Omega_j} (\mathbf{x}_k - \boldsymbol{\mu}_j^{(0)})\,{}^T(\mathbf{x}_k - \boldsymbol{\mu}_j^{(0)}))^{-1}$$

Puis, on fusionne les nouveaux blobs $(\{\boldsymbol{\mu}_j^{(0)}\}, \{\boldsymbol{\sigma}_j^{(0)}\})$ avec les blobs de l'image précédente.

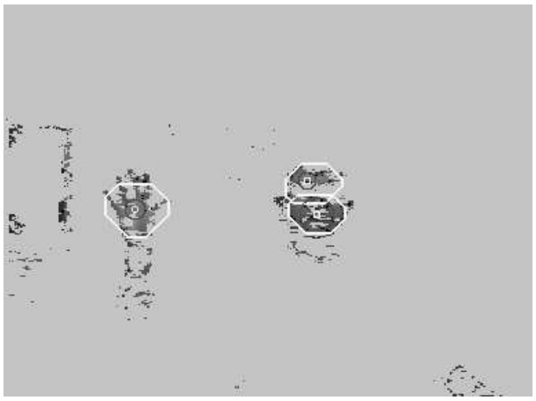

FIG. 5 – *Blobs avant la fusion*

Cette approche permet le plus souvent de sur-estimer le nombre de blobs. Une fois l'ensemble des blobs ci-dessus obtenu, on applique l'algorithme EM. A l'issue de celui-ci, les blobs ont convergé vers un état stable et se sont répartis sur les objets.

Cependant, comme il est possible d'avoir deux blobs sur le même objet (figure 5), il faut introduire un mécanisme pour fusionner les blobs.

5 Fusion des blobs

Il existe plusieurs manières de fusionner des blobs. On peut tout d'abord fusionner les blobs qui se superposent, ou bien ceux qui sont à une distance minimale. Dans tous les cas, il faut introduire un seuil qui le plus souvent est fixé a priori. Nous avons choisi une approche intermédiaire où chaque blob possède un "rayon d'attraction".

La fusion s'effectue sur un critère spatial et colorimétrique à l'aide d'un "rayon d'attraction" intra-blob ς recalculé après la maximisation finale des $\{\boldsymbol{\mu}_j\}$. On pose ς_j comme étant l'écart moyen spatio-colorimétrique (Equation 11) des pixels au centre du blob $\boldsymbol{\mu}_j$.

$$\varsigma_j = \frac{1}{Card\,\Omega_j} \sum_{\mathbf{x}_k \in \Omega_j} \| \mathbf{x}_k - \boldsymbol{\mu}_j \| \qquad (11)$$

On décide de fusionner deux blobs j et k (figure 6) si $\| \boldsymbol{\mu}_j - \boldsymbol{\mu}_k \| \leq \varsigma_j + \varsigma_k$.

FIG. 6 – *Blobs après la fusion*

En utilisant cette méthode, on permet à deux blobs suffisamment proches dans l'espace spatio-colorimétrique de fusionner ensemble. On peut éviter également qu'un blob-main ne fusionne avec un blob-visage lorsque la main passe près du visage ou vient toucher le menton.

Cependant, la fusion n'est pas garantie, on peut très bien observer temporairement deux blobs sur le même objet (figure 5). Mais, lorsque les objets bougent, on constate que, le plus souvent, l'image apporte de nouvelles informations de teinte chair pour permettre aux blobs de fusionner.

6 Résultats

Nous obtenons des blobs (figure 7 et 8) plus robustes que les zones de pixels de teinte chair connexes. En effet, on constate que le nombre de blobs est beaucoup plus constant que le nombre de zones. De plus, les blobs sont géométriquement plus stables que les zones.

FIG. 7 – *Exemple de blobs du visage et de la main sur fond uniforme*

FIG. 8 – *Exemple de blobs du visage et de la main sur fond complexe*

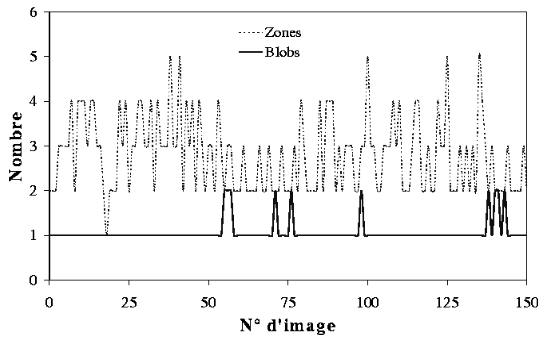

FIG. 9 – *Variation du nombre de zones et de blobs sur une séquence visage seul*

Prenons l'exemple d'une séquence ne contenant que le visage, le nombre de blobs est nettement plus régulier que le nombre de zones (figure 9). De même, la variation de la surface d'un blob-visage est moindre que celle d'une zone-visage (figure 10).

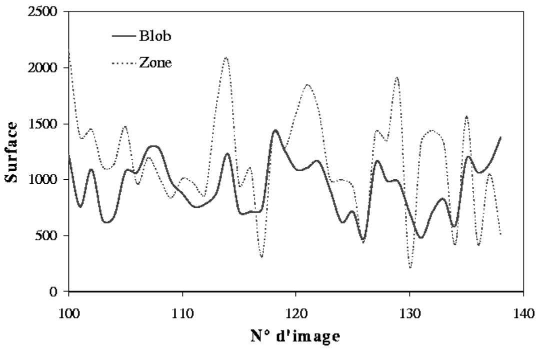

FIG. 10 – *Variation de la surface d'une zone et d'un blob sur une séquence visage seul*

7 Application à la détection du visage et de la main

LISTEN est un système temps-réel de vision par ordinateur qui détecte et suit un visage dans une séquence d'images vidéo provenant d'une caméra. Dans ce système, les visages sont détectés par un réseau de neurones modulaire [4] dans des zones de teinte chair [2].

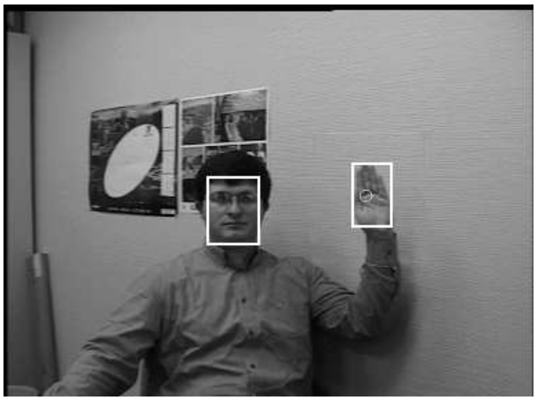

FIG. 11 – *Détection du visage et reconnaissance d'une posture de la main après obtention des blobs*

Il existe une extension de ce système intégrant les blobs pour détecter les visages et reconnaître les gestes de la main (figure 11) [6]. Les postures de la main sont détectées à l'aide de réseaux de neurones dans un espace corporel centré sur le visage [5].

8 Conclusion

Dans cet article, nous avons présenté un modèle probabiliste gaussien de blobs dont les paramètres sont obtenus par un algorithme EM. Les blobs sont initialisés à partir des zones de pixels de teinte chair connexes. De plus, un mécanisme de fusion de blobs est proposé. On constate que le nombre des blobs est moins important que le nombre de zones de pixels de teinte chair connexes. D'autre part, on observe que les blobs sont géométriquement plus stables que les zones.

Actuellement, les blobs sont utilisés pour la détection des visages et la reconnaissance des postures de la main. Cependant, les travaux futurs s'orientent vers l'analyse des gestes de la main au travers de leurs propriétés géométriques et de leurs trajectoires dans l'image. Il nous faut donc des blobs toujours plus stables.

Aussi, nous envisageons d'étudier un nouveau modèle, où le nombre de blobs n'est plus fixé a priori mais estimé dynamiquement. De plus, nous pensons améliorer l'initialisation des blobs afin de ne plus s'appuyer sur les zones, et d'élargir les conditions de fusion des blobs.

Références

[1] Baum, L.: An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. Inequalities, vol. 3, 1-8, 1972

[2] Collobert, M. et Feraud, R. et Le Tourneur G. et Bernier, O. et Viallet, J.E. et Mahieux, Y. et Collobert, D.: LISTEN: A System for Locating and Tracking Individual Speakers. 2nd Int. Conf. on Automatic Face and Gesture Recognition, 283-288, 1996

[3] Dempster, A.P. et Laird, N.M. et Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. Journal of Royal Statistical Society, vol. 39, 1-938, 1977

[4] Feraud, R.: PCA, Neural Networks and Estimation for Face Detection. NATO ASI, Face Recognition: from Theory to Applications, 424-432, 1997

[5] Marcel, S.: Hand Posture Recognition in a Body-Face centered space. CHI'99, Extended Abstracts, 302-303, 15-20 mai, Pittsburgh PA, USA, 1999

[6] Marcel, S. et Bernier, O. et Collobert, D.: Reconnaissance de la main pour les interfaces gestuelles. CORESA'99, 14-15 juin, Sophia-Antipolis, 1999

[7] Pavlovic, V.I. et Sharma, R. et Huang, Thomas S.: Visual Interpretation of Hand Gestures for Human-Computer Interation: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, 677-695, 1997

[8] Richard P. Schumeyer et Kenneth E. Barner: A Color-Based Classifier for Region Identification in Video. SPIE, Proceedings of Visual Communications and Image Procesing, vol 3309, pp 189-200, 1998

[9] Stan Sclaroff et John Isidoro: Active Blobs: Proceedings of the International Conference on Computer Vision, India, 1998

[10] Wren, C. et Azarbayejani, A. et Darrell, T. et Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, 780-785, 1997

[11] Ming-Hsuan Yang et Narenda Ahuja: Gaussian Mixture Model for Human Skin Color and Its applications in Image and Video Databases. SPIE, Conference on Storage and Retrieval for Image and Video Databases, San Jose, 1999

[12] Jie Yang et Weier Lu et Alex Waibel: Skin color modeling and adaptation. Proceedings of the 3^{rd} Asian Conference on Computer Vision, vol 2, pp 687-694, 1998

[13] Ming-Hsuan Yang et Narenda Ahuja: Detecting faces in color images. Proceedings of the International Conference on Image Processing, vol 1, pp 127-130, Chicago, 1998