

Segmentation d'images polarimétriques sous éclairage non uniforme

Marc KARNOUKIAN¹, Sylvain FAISAN¹, Christian HEINRICH¹, Alex LALLEMENT¹, Jihad ZALLAT¹

¹ICube, Université de Strasbourg, CNRS ; 300 boulevard Sébastien Brant, BP 10413, 67412 Illkirch, France

karnoukian@unistra.fr

Résumé – Les algorithmes de segmentation dédiés à l'imagerie de Mueller ne sont pas robustes aux inhomogénéités de l'éclairage, qui conduisent à des erreurs de classification ou à des problèmes de sur-segmentation. Pour résoudre ces problèmes, ces images sont modélisées ici par un modèle adapté de mélanges gaussiens qui prend en compte explicitement l'inhomogénéité de l'éclairage dans le modèle de formation des images. Les paramètres du modèle sont estimés grâce à un algorithme d'espérance-maximisation (EM). L'algorithme EM est inséré à l'intérieur d'une procédure itérative qui permet d'estimer le nombre de classes. Une fois les paramètres calculés, la carte de segmentation est obtenue en utilisant une modélisation markovienne classique. Les résultats obtenus sont comparés avec une approche en deux temps qui consiste à normaliser les données, puis à les segmenter. Ils montrent l'intérêt de prendre en compte l'inhomogénéité de l'éclairage au cœur de l'algorithme de segmentation.

Abstract – The proposed segmentation algorithms dedicated to Mueller matrix images are not robust to illumination inhomogeneities that may lead to misclassifications and over segmentation issues. To overcome these drawbacks, such images are modeled by an adapted Gaussian mixture distribution that takes explicitly into account illumination inhomogeneities in the image formation model. The model parameters are estimated with the Expectation-Maximization (EM) algorithm. The EM procedure is inserted into an iterative framework that allows the estimation of the number of classes. Based on the estimated parameters, segmentation is achieved by modeling the segmentation map as a Markov random field. Results obtained are compared with a two-step approach that relies on a normalization step followed by a segmentation one. They demonstrate the interest of accounting for lightning non-uniformity in the core of the estimation process.

1 Introduction

L'imagerie polarimétrique permet d'obtenir des informations sur la nature des interactions entre le milieu et la lumière et dévoile ainsi des propriétés inaccessibles classiquement. Elle est complémentaire des approches traditionnelles pour différentes applications telles que le contrôle non-destructif et la détection dans un fouillis... Les propriétés polarimétriques d'un milieu matériel (linéaire) sont décrites par une matrice dite de Mueller de taille 4×4 . Sa mesure se fait grâce à un polarimètre de Mueller qui acquiert N images ($N \geq 16$) de la scène sous différentes conditions. En chaque pixel \mathbf{s} , la matrice de Mueller $\mathbf{M}(\mathbf{s})$ est reliée aux N mesures d'intensité $\mathbf{I}(\mathbf{s})$ par :

$$\mathbf{I}(\mathbf{s}) = \mathbf{P} \cdot \underline{\mathbf{M}(\mathbf{s})} (+ \text{bruit}), \quad (1)$$

où \mathbf{P} est la matrice de mesure polarimétrique (\mathbf{P} est connue et bien conditionnée) et où le sous-tiret est l'opérateur de vectorisation qui convertit une matrice en un vecteur colonne (obtenu en empilant les colonnes de la matrice).

Peu d'algorithmes de segmentation d'images de Mueller ont été proposés [7, 1]. L'approche développée dans [1] permet uniquement de segmenter un objet d'intérêt du fond et le défaut de [7] est que la non-uniformité de l'éclairage affecte largement la qualité des résultats de segmentation. Pour s'affranchir de ce problème, nous proposons un modèle original (section 2) qui prend directement en compte l'inhomogénéité de l'éclairage. Ces paramètres sont estimés sans prendre en compte l'informa-

tion spatiale grâce à un algorithme d'espérance-maximisation (section 3). Puis, une carte de segmentation est obtenue en utilisant une régularisation spatiale (section 4).

2 Modélisation

Si le pixel \mathbf{s} est associé à la classe k (on notera $y(\mathbf{s}) = k$), nous considérons le modèle d'observation :

$$\begin{aligned} \mathbf{I}(\mathbf{s}) &= \kappa(\mathbf{s})\mathbf{P}\mathbf{M}_k + \epsilon(\mathbf{s}) \\ &= \kappa(\mathbf{s})\boldsymbol{\Psi}_k + \epsilon(\mathbf{s}), \end{aligned} \quad (2)$$

où \mathbf{M}_k est la matrice de Mueller de la classe k et $\boldsymbol{\Psi}_k$ les mesures d'intensité associées ($\boldsymbol{\Psi}_k = \mathbf{P}\mathbf{M}_k$). De plus, $\epsilon(\mathbf{s}) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k)$ où $\boldsymbol{\Sigma}_k$ est la matrice de variance-covariance associée à la classe k . $\boldsymbol{\Sigma}_k$ est supposée diagonale car les bruits observés sur les différents canaux d'observation (les composantes de \mathbf{I}) sont indépendants. Le terme $\kappa(\mathbf{s})$ modélise les non-uniformités de l'éclairage : on suppose qu'il suit une loi impropre constante sur les réels positifs. Par la suite, un facteur $\kappa(\mathbf{s})$ différent est utilisé pour chaque classe. Cependant, nous n'utilisons pas la notation $\kappa_k(\mathbf{s})$ dans l'Eq. 2 car le terme ne fait pas partie du modèle (cf fin de la section). Avec une telle modélisation, on peut obtenir une infinité de modèles équivalents : il suffit de multiplier $\kappa(\mathbf{s})$ par un facteur et de diviser la matrice \mathbf{M}_k par le même facteur. Pour enlever cette ambiguïté, l'élément supérieur gauche de \mathbf{M}_k est fixé à 1 (cet élément est strictement positif pour toutes les matrices de Mueller).

Finalement, le modèle est décrit par le nombre de classes K et par le vecteur de paramètres $\Theta = \{\Theta_k\}_{k=1\dots K} = \{\pi_k, \mathbf{M}_k, \Psi_k, \Sigma_k\}_{k=1\dots K}$ où $\pi_k = P(y(\mathbf{s}) = k)$. Comme \mathbf{M}_k et Ψ_k sont liés par la relation $\Psi_k = \mathbf{P}\mathbf{M}_k$, uniquement l'une de ces deux variables est nécessaire mais elles sont toutes les deux introduites dans le modèle pour faciliter l'écriture des équations. La vraisemblance $L(\Theta)$ s'écrit sous l'hypothèse d'indépendance des pixels :

$$L(\Theta) = \prod_{\mathbf{s}} P(\mathbf{I}(\mathbf{s})|\Theta), \quad (3)$$

où $P(\mathbf{I}(\mathbf{s})|\Theta)$ est proportionnel à

$$\begin{aligned} & \sum_{k=1}^K \int_{\kappa(\mathbf{s})=0}^{+\infty} \pi_k P(\mathbf{I}(\mathbf{s})|y(\mathbf{s}) = k, \kappa(\mathbf{s}), \Theta_k) d\kappa(\mathbf{s}) \\ &= \sum_{k=1}^K \int_{\kappa(\mathbf{s})=0}^{+\infty} \pi_k \mathcal{N}(0, \Sigma_k, \mathbf{I}(\mathbf{s}) - \kappa(\mathbf{s})\Psi_k) d\kappa(\mathbf{s}), \end{aligned} \quad (4)$$

$\mathcal{N}(\mu, \Sigma, \mathbf{x})$ est la densité de probabilité normale de moyenne μ et de matrice de variance-covariance Σ évaluée en \mathbf{x} .

3 Estimation

L'algorithme d'espérance-maximisation (EM) est utilisé pour estimer les paramètres du modèle (section 3.1). Il est inséré dans un paradigme [6] qui permet d'estimer le nombre de classes. Il fonctionne suivant ce principe : l'algorithme démarre avec une unique classe ($K = 1$). Après estimation des paramètres, le nombre de classes est incrémenté si au moins une classe parmi les K ne satisfait pas à des critères statistiques (section 3.2). Cette procédure (estimation des paramètres, tests statistiques, et incrémentation du nombre de classes) est répétée jusqu'à ce que toutes les classes satisfassent les critères. L'algorithme EM étant sensible à l'initialisation, la procédure d'initialisation des paramètres est décrite en annexe pour les cas $K = 1$ et $K > 1$.

3.1 Algorithme d'espérance-maximisation

L'algorithme EM est souvent utilisé pour estimer les paramètres des modèles de mélanges gaussiens [2]. Il introduit une fonction auxiliaire $Q(\Theta, \Theta')$ qui satisfait à la propriété : si $Q(\Theta, \Theta') \geq Q(\Theta', \Theta')$ alors $L(\Theta) \geq L(\Theta')$. L'algorithme EM procède alors en définissant une série de modèles $\{\Theta_1, \Theta_2, \dots\}$ vérifiant $Q(\Theta_{i+1}, \Theta_i) \geq Q(\Theta_i, \Theta_i)$ et donc $L(\Theta_{i+1}) \geq L(\Theta_i)$. Dans le cadre des modèles de mélanges gaussiens, la fonction auxiliaire est construite à partir des données cachées $y(\mathbf{s})$. Nous conservons ici ce principe. De plus, le terme $\kappa(\mathbf{s})$ est considéré également comme une donnée cachée. La fonction auxiliaire s'écrit alors :

$$Q(\Theta, \Theta') = \sum_y \int_{\kappa} \log \left[P(\mathbf{I}, y, \kappa | \Theta) \right] P(y, \kappa | \mathbf{I}, \Theta') d\kappa. \quad (5)$$

Des manipulations similaires à celles décrites dans [2] permettent d'obtenir :

$$\begin{aligned} Q(\Theta, \Theta') &= \sum_{\mathbf{s}} \sum_{k=1}^K \int_{\kappa(\mathbf{s})=0}^{+\infty} P\left(y(\mathbf{s}) = k, \kappa(\mathbf{s}) | \mathbf{I}(\mathbf{s}), \Theta'\right) \\ &\quad \log \left[P(\mathbf{I}(\mathbf{s}), y(\mathbf{s}) = k, \kappa(\mathbf{s}) | \Theta_k) \right] d\kappa(\mathbf{s}). \end{aligned} \quad (6)$$

En notant $w'_k(\mathbf{s}) = P(y(\mathbf{s}) = k | \mathbf{I}(\mathbf{s}), \Theta')$, on a :

$$\begin{aligned} Q(\Theta, \Theta') &= \sum_{\mathbf{s}} \sum_{k=1}^K w'_k(\mathbf{s}) \int_{\kappa(\mathbf{s})=0}^{+\infty} P(\kappa(\mathbf{s}) | y(\mathbf{s}) = k, \mathbf{I}(\mathbf{s}), \Theta') \\ &\quad \log \left[P\left(\mathbf{I}(\mathbf{s}), y(\mathbf{s}) = k, \kappa(\mathbf{s}) | \Theta_k\right) \right] d\kappa(\mathbf{s}), \end{aligned} \quad (7)$$

où

$$\begin{aligned} w'_k(\mathbf{s}) &\propto \pi'_k \int_{\kappa(\mathbf{s})=0}^{+\infty} P\left(\mathbf{I}(\mathbf{s}) | y(\mathbf{s}) = k, \kappa(\mathbf{s}), \Theta'_k\right) d\kappa(\mathbf{s}) \\ &\propto \pi'_k \sqrt{\frac{\alpha'_k}{|\Sigma_k|}} \exp\left(\frac{\alpha'_k \beta_k'^2 - \gamma'_k}{2}\right) \left(1 + \operatorname{erf}\left(\beta'_k \sqrt{\frac{\alpha'_k}{2}}\right)\right), \end{aligned} \quad (8)$$

avec $\alpha'_k = (\Psi_k'^T \Sigma_k'^{-1} \Psi_k')^{-1}$, $\beta'_k = \Psi_k'^T \Sigma_k'^{-1} \mathbf{I}(\mathbf{s})$ et enfin $\gamma'_k = \mathbf{I}(\mathbf{s}) \Sigma_k'^{-1} \mathbf{I}(\mathbf{s})$. Le coefficient de proportionnalité (Eq. 8) peut être facilement déterminé car $\sum_{k=1}^K w'_k(\mathbf{s}) = 1$. A notre connaissance, l'étape de maximisation (Eq. 7) ne peut pas être réalisée de manière analytique. Nous avons recours à un algorithme EM stochastique [3]. Le principe est d'approximer la fonction auxiliaire en générant des données cachées conduisant ainsi à l'optimisation de la log-vraisemblance totale. En chaque pixel \mathbf{s} et pour chaque classe k , les données cachées $\kappa_k(\mathbf{s})$ sont générées suivant :

$$\kappa_k(\mathbf{s}) | y(\mathbf{s}) = k, \mathbf{I}(\mathbf{s}), \Theta' \sim \mathcal{N}^+(\alpha'_k \beta'_k, \alpha'_k), \quad (9)$$

où \mathcal{N}^+ est une loi normale à support positif. Une fois les données $\kappa_k(\mathbf{s})$ générées, l'optimisation de l'Eq. 7 conduit à :

$$\hat{\Theta} = \arg \max_{\Theta} \sum_{\mathbf{s}} \sum_{k=1}^K w'_k(\mathbf{s}) \log[\pi_k P(\mathbf{I}(\mathbf{s}) | y(\mathbf{s}) = k, \kappa_k(\mathbf{s}), \Theta_k)] \quad (10)$$

Le passage de l'Eq. 7 à l'Eq. 10 revient à effectuer une intégration de Monte-Carlo avec la loi instrumentale $P(\cdot | y(\mathbf{s}) = k, \mathbf{I}(\mathbf{s}), \Theta')$ pour le calcul de chaque intégrale. L'optimisation de l'Eq. 10 est réalisée comme suit. Tout d'abord, on a :

$$\hat{\pi}_k = \frac{1}{\#\text{pixels}} \sum_{\mathbf{s}} w'_k(\mathbf{s}). \quad (11)$$

L'équation 10 s'écrit alors (pour chaque k) :

$$\begin{aligned} \{\hat{\Psi}_k, \hat{\Sigma}_k\} &= \arg \min_{\Psi, \Sigma} \sum_{\mathbf{s}} w'_k(\mathbf{s}) \\ &\quad \left[\log(|\Sigma|) + \left(\mathbf{I}(\mathbf{s}) - \kappa_k(\mathbf{s})\Psi \right)^T \Sigma^{-1} \left(\mathbf{I}(\mathbf{s}) - \kappa_k(\mathbf{s})\Psi \right) \right]. \end{aligned} \quad (12)$$

La valeur optimale pour Ψ_k notée $\bar{\mathbf{I}}_k$ est :

$$\bar{\mathbf{I}}_k = \frac{1}{\sum_{\mathbf{s}} w'_k(\mathbf{s}) \kappa_k^2(\mathbf{s})} \sum_{\mathbf{s}} w'_k(\mathbf{s}) \kappa_k(\mathbf{s}) \mathbf{I}(\mathbf{s}). \quad (13)$$

Cependant, il n'y a aucune garantie que $\bar{\mathbf{I}}_k$ soit une solution acceptable. Pour être acceptable, il doit exister une matrice de Mueller \mathbf{M}_k vérifiant $\Psi_k = \mathbf{P}\mathbf{M}_k$. On peut démontrer que le critère de l'équation 12 s'écrit également :

$$\{\hat{\mathbf{M}}_k, \hat{\Sigma}_k\} = \arg \min_{\mathbf{M}, \Sigma} \left(\sum_{\mathbf{s}} w'_k(\mathbf{s}) \right) \log(|\Sigma|) + \left(\sum_{\mathbf{s}} w'_k(\mathbf{s}) \kappa_k^2(\mathbf{s}) \right) \|\Sigma^{-1/2}(\bar{\mathbf{I}}_k - \mathbf{P}\mathbf{M})\|^2. \quad (14)$$

Une variante de l'algorithme EM consiste à trouver un Θ vérifiant $Q(\Theta, \Theta') > Q(\Theta', \Theta)$ (mais Θ n'est pas un minimiseur de $Q(\cdot, \Theta')$) [2]. Nous proposons donc, à partir de l'équation 14, d'estimer \mathbf{M}_k en fixant Σ_k égal à Σ'_k , et ensuite d'estimer Σ_k en fixant \mathbf{M}_k égal à son estimation. Ces deux optimisations successives permettent d'accroître la fonction auxiliaire (ou au pire de ne pas la modifier).

Avec $\Sigma_k = \Sigma'_k$, le critère de l'équation 14 devient :

$$\hat{\mathbf{M}}_k = \arg \min_{\mathbf{M}} \|\Sigma_k'^{-1/2}(\bar{\mathbf{I}}_k - \mathbf{P}\mathbf{M})\|^2, \quad (15)$$

dont le minimiseur est $(\mathbf{P}^t \Sigma_k'^{-1} \mathbf{P})^{-1} \mathbf{P}^t \Sigma_k'^{-1} \bar{\mathbf{I}}_k$. Ce dernier convient s'il représente une matrice de Mueller. Si ce n'est pas le cas, le critère de l'équation 15 est optimisé en utilisant une procédure d'optimisation sous contrainte [4].

Finalement, l'estimation de Σ_k ne pose pas de problème particulier (connaissant $\hat{\Psi}_k = \mathbf{P}\hat{\mathbf{M}}_k$) :

$$\hat{\Sigma}_k = \frac{1}{\sum_{\mathbf{s}} w'_k(\mathbf{s})} I_{N \times N} \odot \left(\sum_{\mathbf{s}} w'_k(\mathbf{s}) \left(\mathbf{I}(\mathbf{s}) - \kappa_k(\mathbf{s}) \hat{\Psi}_k \right) \left(\mathbf{I}(\mathbf{s}) - \kappa_k(\mathbf{s}) \hat{\Psi}_k \right)^T \right) \quad (16)$$

où \odot est le produit de Hadamard et où $I_{N \times N}$ est la matrice identité de taille $N \times N$.

En résumé, une itération de l'algorithme EM procède comme suit : les poids $w'_k(\mathbf{s})$ (Eq. 8), puis les *a priori* (Eq. 11) sont calculés. Les données cachées $\kappa_k(\mathbf{s})$ sont générées suivant l'Eq. 9. $\hat{\mathbf{M}}_k$ est alors obtenu en minimisant le critère 15, puis $\hat{\Sigma}_k$ est obtenu à partir de l'Eq. 16. Enfin, les éléments de $\hat{\mathbf{M}}_k$ sont divisés par son élément supérieur gauche de manière à ce que ce dernier soit égale à 1. On peut enfin calculer $\hat{\Psi}_k$ comme étant égal à $\mathbf{P}\hat{\mathbf{M}}_k$.

3.2 Augmentation du nombre de classes

Le nombre de classes est incrémenté si au moins une classe (parmi les K) ne satisfait pas à un critère statistique basé sur le résidu \mathbf{r}_k . Pour une classe k , et au pixel \mathbf{s} , il s'écrit :

$$\hat{\mathbf{r}}_k(\mathbf{s}) = \mathbf{I}(\mathbf{s}) - \hat{\kappa}_k(\mathbf{s}) \Psi_k \quad (17)$$

où $\hat{\kappa}_k(\mathbf{s})$ est estimé de manière à maximiser son *a posteriori* (Eq. 9). Si le nombre de classes est trop petit, on peut s'attendre à ce qu'au moins la distribution d'un résidu (suivant une composante) ne soit pas unimodale. Ainsi, nous avons utilisé un test d'unimodalité [5]. Du point de vue de la classe k , les pixels ne partagent pas les mêmes poids $w_k(\mathbf{s}) = P(y(\mathbf{s}) = k | \mathbf{I}(\mathbf{s}), \Theta)$

si bien que les résidus n'ont pas tous la même importance. Nous utilisons la stratégie définie dans [6] pour estimer correctement la distribution des différents résidus (suivant chaque composante) à partir de $\hat{\mathbf{r}}_k(\mathbf{s})$.

Un test statistique est réalisé pour chacune des composantes conduisant à $N.K$ p-valeurs (il y a N composantes par classe). Le problème des comparaisons multiples est résolu avec la correction de Sidak. L'hypothèse nulle est rejetée au niveau α (α est fixé à 0.05 dans la suite) si la plus petite p-valeur $p_{min} = \min\{p_{k,j}\}_{k=1\dots K, j=1\dots N}$ est inférieure à $1 - (1 - \alpha)^{1/NK}$. Si l'hypothèse nulle est rejetée, le nombre de classes K est incrémenté et l'initialisation de l'algorithme EM est obtenue en découpant en 2 la classe associée à la p-valeur minimale (cf annexe).

4 Segmentation

La carte de segmentation I_0 est modélisée comme un champ markovien de Potts de paramètre β . La segmentation est obtenue avec le critère :

$$\{\hat{I}_0, \hat{\beta}\} = \arg \max_{\{I_0, \beta\}} \log P(I_0 | \beta) + \sum_{\mathbf{s}} \log P(\mathbf{I}(\mathbf{s}) | I_0(\mathbf{s}), \Theta) \quad (18)$$

Pour son optimisation, nous construisons une série de cartes de segmentation $\{I_0^{(1)}, I_0^{(2)}, \dots\}$ et de $\beta : \{\beta^{(1)}, \beta^{(2)}, \dots\}$ jusqu'à convergence. La carte de segmentation $I_0^{(n)}$ est obtenue à partir de l'équation 18 en posant β égal à $\beta^{(n)}$ (optimisation par coupes minimales), alors que $\beta^{(n)}$ est calculée à partir de l'équation 18 en posant I_0 égal à $I_0^{(n-1)}$ (optimisation similaire à celle développée dans [7]). Pour l'initialisation, $\beta^{(1)}$ est fixé à 0 (pas de régularisation).

5 Résultats

La méthode proposée a été testée et validée grâce à des données synthétiques (non montrées) et des images réelles obtenues avec un polarimètre fonctionnant en transmission. Nous comparons les résultats avec ceux obtenus par [7]. Avec cet algorithme, le nombre de classes doit être fixé et, pour prendre en compte l'inhomogénéité de l'éclairage, une étape de normalisation des données est effectuée : on estime l'image de transmittance moyenne (élément supérieur gauche de la matrice de Mueller) et on divise les N images acquises par cette dernière. L'objet considéré par la suite est représenté schématiquement Fig. 1a. Il est composé d'un film transparent de cellophane sur lequel sont placés trois films polaroïd dichroïques avec des orientations différentes (ils ont donc chacun une matrice de Mueller différente) et du papier. On peut observer, Fig. 1b, l'image « d'intensité classique » de la scène ainsi que la présence d'un halo lumineux dû à l'inhomogénéité de l'éclairage. Les autres figures représentent la carte de segmentation obtenue avec la méthode proposée (fig. 1c, environ 20 minutes de calcul sur un ordinateur standard), et la méthode de [7] sans

normalisation des données avec 5 et 6 classes (Fig. 1d et 1e) et avec normalisation (Fig. 1f).

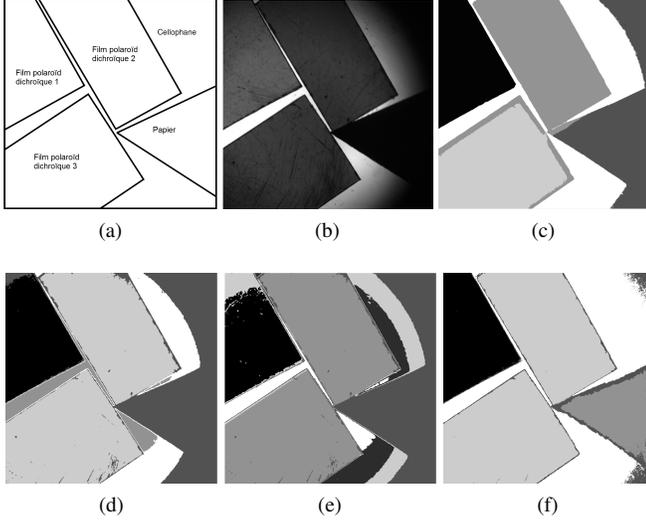


FIGURE 1 – Composition de l’objet-test (a), image « d’intensité classique » (b) et cartes de segmentation obtenues avec la méthode proposée (c) et [7] sans normalisation des données (5 et 6 classes (d,e) et avec normalisation (5 classes, f)).

Sans normalisation, l’approche de [7] est sensible à l’éclairage. Par exemple, le cellophane qui est une classe pour laquelle l’éclairage est très peu homogène est découpé en 3 ou 4 classes (Fig. 1d et Fig. 1e). Avec normalisation des données (Fig. 1f), les résultats sont plus satisfaisants mais les trois films polaroïd ne sont pas tous distingués. Cela vient du fait que le bruit présent dans les images normalisées possède une distribution de Cauchy (sous l’hypothèse qu’il est gaussien dans les images originales), rendant ainsi l’algorithme de segmentation moins adapté. Pour pouvoir distinguer les trois films polaroïd avec l’approche de [7], il est nécessaire de normaliser les données et d’utiliser 9 classes pour la segmentation (résultats non montrés).

Au contraire, l’approche proposée converge automatiquement vers le nombre attendu de classes, à savoir 5 et permet de distinguer les trois films polaroïd. On peut noter toutefois que le bord de l’image à droite est mal segmenté. Cela provient du fait que l’éclairage est très faible à cet endroit.

En conclusion, les résultats obtenus montrent qu’un éclairage non uniforme rend les algorithmes de segmentation classiques peu appropriés. Au contraire, l’approche proposée prend en compte l’éclairage à l’intérieur du modèle d’observation. Les résultats obtenus sont très satisfaisants.

A Initialisation de l’algorithme EM

Dans le cas $k = 1$ ($\Theta_1 = \{\pi_1 = 1, \mathbf{M}_1, \Psi_1, \Sigma_1\}$), $\hat{\kappa}(\mathbf{s})$ est fixé à l’élément supérieur gauche de $\mathbf{M}(\mathbf{s})$ (estimé par pseudo-inverse). Si $\hat{\kappa}(\mathbf{s})$ est plus petit qu’un seuil ($2 \cdot 10^{-16}$), $\hat{\kappa}(\mathbf{s})$ est

fixé égal à ce seuil. Ψ_1 et \mathbf{M}_1 sont alors estimés comme suit :

$$\Psi_1 \propto \frac{\sum_{\mathbf{s}} \mathbf{I}(\mathbf{s})}{\sum_{\mathbf{s}} \hat{\kappa}(\mathbf{s})} \text{ et } \mathbf{M}_1 = (\mathbf{P}^t \mathbf{P})^{-1} \mathbf{P}^t \Psi_1 \quad (19)$$

où le coefficient de proportionnalité est calculé pour que l’élément supérieur gauche de \mathbf{M}_1 soit égal à 1. On peut noter que \mathbf{M}_1 peut ne pas être une matrice de Mueller et que Ψ_1 peut ne pas être égal à $\mathbf{P} \mathbf{M}_1$ (si $N > 16$). Ces contraintes seront prises en compte dès la première itération de l’algorithme EM. Enfin, Σ_1 est calculé à partir des résidus $\mathbf{I}(\mathbf{s}) - \hat{\kappa}(\mathbf{s}) \Psi_1$.

Si $k > 1$, on note $(k^*, n^*) = \arg \min_{k=1 \dots K, j=1 \dots N} \{p_{k,n}\}$ (cf section 3.2). Le modèle $\Theta = \{\Theta_k\}_{k=1 \dots K+1}$ est initialisé à partir du modèle courant $\Theta^{(p)} = \{\Theta_k^{(p)}\}_{k=1 \dots K}$ en découpant en deux la classe k^* de $\Theta^{(p)}$. On pose ainsi $\Theta_k = \Theta_k^{(p)}$ pour $k = 1 \dots K, k \neq k^*$. Les paramètres Θ_{K+1} et Θ_{k^*} sont alors estimés en deux étapes. Tout d’abord, les pixels de l’image sont regroupés en deux classes C_{K+1} et C_{k^*} en seuillant le résidu associé à la classe k^* pour la n^* -ème composante. Le seuillage s’effectue en considérant les poids $w_{k^*}^{(p)}(\mathbf{s}) = P(y(\mathbf{s}) = k^* | \mathbf{I}(\mathbf{s}), \Theta^{(p)})$ [6]. Les paramètres Θ_{K+1} et Θ_{k^*} sont alors estimés de manière similaire au cas $K = 1$ avec 3 différences : (i) uniquement les pixels de C_{K+1} et de C_{k^*} sont utilisés pour l’estimation de Θ_{K+1} et Θ_{k^*} , respectivement, (ii) les moyennes et les variances sont pondérées par les poids $w_{k^*}^{(p)}(\mathbf{s})$ et (iii), les *a priori* sont fixés comme suit :

$$\pi_{K+1} = \pi_{k^*}^{(p)} \frac{\sum_{\mathbf{s} \in C_{K+1}} w_{k^*}^{(p)}(\mathbf{s})}{\sum_{\mathbf{s} \in C_{K+1} \cup C_{k^*}} w_{k^*}^{(p)}(\mathbf{s})}, \text{ and } \pi_{k^*} = \pi_{k^*}^{(p)} - \pi_{K+1} \quad (20)$$

Références

- [1] G. ANNA, N. BERTAUX, F. GALLAND, F. GOUDAIL et D. DOLFI : Joint contrast optimization and object segmentation in active polarimetric images. *Opt. Lett.*, 37(16):3321–3323, 2012.
- [2] J. BILMES : A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Rapport technique, International Computer Science Institute, 1998.
- [3] G. CELEUX et J. DIEBOLT : The SEM algorithm : a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational statistics quarterly*, 2(1):73–82, 1985.
- [4] S. FAISAN, C. HEINRICH, G. SFIKAS et J. ZALLAT : Estimation of Mueller matrices using non-local means filtering. *Opt. Express*, 21(4):4424–4438, Feb 2013.
- [5] J. A. HARTIGAN et P. M. HARTIGAN : The Dip test of unimodality. *Ann. Statist.*, 13(1):1–433, 1985.
- [6] D. VERVERIDIS et C. KOTROPOULOS : Gaussian mixture modeling by exploiting the Mahalanobis distance. *Signal Processing, IEEE Transactions on*, 56(7):2797–2811, July 2008.
- [7] J. ZALLAT, C. HEINRICH et M. PETREMAND : A bayesian approach for polarimetric data reduction : the Mueller imaging case. *Optics Express*, 16(10), 2008.