HAL
open science

# Sérialisation du k-means pour la segmentation des images en couleur

Yann Leydier, Frank Le Bourgeois, Hubert Emptoz

## To cite this version:

Yann Leydier, Frank Le Bourgeois, Hubert Emptoz. Sérialisation du k-means pour la segmentation des images en couleur. 8ième colloque international francophone sur l'écrit et le document, CIFED, Jun 2004, La Rochelle, France. pp.191-196. sic_00001217

**HAL Id: sic_00001217**

**https://archivesic.ccsd.cnrs.fr/sic_00001217v1**

Submitted on 7 Dec 2004

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Sérialisation du *k*-means pour la segmentation des images en couleur: Application aux images de documents et autres

Yann LEYDIER[1,2] – Frank Le BOURGEOIS[2] – Hubert EMPTOZ[2]

[1] Archimed Lyon,
4 quai des Étroits
69005 Lyon

y.leydier@archimed.fr

[2] LIRIS,
20 av. Albert Einstein
69621 Villeurbanne Cedex

{frank.lebourgeois|hubert.emptoz}@liris.cnrs.fr

**Résumé** : *Cet article présente un système de segmentation adaptatif qui a été créé pour l'analyse d'images de documents en couleur. Cette méthode est basée sur une sérialisation de l'algorithme k-means (ou* nuées dynamiques*) appliqué séquentiellement sur l'image dans une fenêtre glissante. Lors du déplacement de la fenêtre, notre algorithme réutilise des informations provenant des nuées calculées dans la fenêtre précédente et les ajuste automatiquement pour adapter le classificateur à toute variation locale des couleurs. Afin d'améliorer les résultats, nous proposons de définir plusieurs nuées dans l'espace des caractéristiques pour chaque classe logique. Notre méthode a été testée avec succès sur des images de manuscrits médiévaux, des images de vidéos et d'autres images, naturelles ou non, présentant de lourdes détériorations, des variation d'illumination ou des problèmes de transparence. L'algorithme proposé est suffisamment générique pour être utilisé sur un très large panel d'images à des fins de segmentation couleur ou bien encore de binarisation.*

**Mots-clés** : segmentation couleur, binarisation, images de documents, manuscrits médiévaux

## 1 Introduction

Chaque année des milliers de manuscrits sont numérisés dans le monde. Une partie de ces images est accessible dans des bibliothèques électroniques sur l'Internet mais la plupart d'entre elles a été indexée manuellement à grand coût si bien que l'automatisation de cette tâche est devenue une nécessité. Un certain nombre de systèmes d'indexation a déjà été proposé mais la plupart d'entre eux n'est efficace que sur des images de documents récents. La difficulté d'analyse automatique des manuscrits s'explique par la complexité de la segmentation dûe à la mauvaise qualité des images et à la complexité de leur contenu [LEB 04].
Les algorithmes de segmentation d'images couleurs génériques sont difficilement applicables aux images de documents. Il est donc difficile d'effectuer une segmentation globale sur une image de document sans avoir à faire face à des surcharges de mémoire et des ralentissements considérables de par la haute résolution de numérisation. De plus, les images de documents présentent certaines détériorations qui déroutent les algorithmes de segmentations classiques. La figure 1 illustre les détériorations les plus communes que l'on trouve dans les images de manuscrits médiévaux telles que des taches, des trous, des traces d'humidité, la dégradation de l'encre et des traits fin, ou bien encore des variations la couleur du papier ou des problèmes de transparence…
Dans la plupart des outils de traitement d'images décrits dans la littérature, une phase de restauration peut apparaître avant la phase de segmentation. L'algorithme que nous proposons peut être utilisé pour restaurer et/ou segmenter les images.

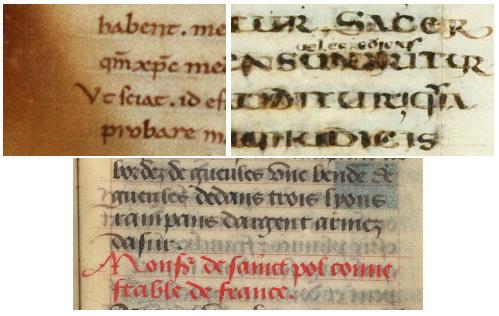

FIG. 1 – Exemples d'images de manuscrits médiévaux.

## 2 Proposition

### 2.1 Sérialisation de l'algorithme *k*-means

Les méthodes de segmentation adaptatives sont généralement celles qui donnent les meilleurs résultats sur l'analyse d'images. Il notamment a été démontré que les algorithmes de binarisation adaptatifs, comme ceux de Niblack ou Sauvola, sont plus performants que des algorithmes non adaptatifs, comme ceux de Otsu ou Fisher. Cela s'explique par les détériorations spécifiques aux images de documents, comme les taches, les variation d'illumination

ou bien les couleurs estompées qui doivent être traitées par des algorithmes particulièrement adaptatifs. En général, on utilise des informations locales calculées dans une fenêtre glissante centrée tour à tour sur chaque pixel.
Le principe de notre méthode est d'appliquer un classificateur non supervisé, comme par exemple *k*-means, sur une fenêtre glissante pour rendre la segmentation dépendante du contexte et permettre au classificateur d'adapter subtilement les nuées à tout variation locale des couleurs. Nous avons choisi l'algorithme *k*-means car c'est le classificateur non supervisé le plus simple et le plus efficace qui puisse être aisément modifié et contrôlé. D'autres classificateurs peuvent cependant être utilisés.

Dans notre algorithme, chaque point $P$ de l'image est étiqueté par un classificateur *k*-means initialisé avec les centres des nuées de la classification précédente et entraîné sur une fenêtre $w_P$ centrée sur $P$. La dépendance entre deux classifications successives est justifiée par le fait que la plupart de l'information est similaire entre deux fenêtres successives. La sérialisation du *k*-means a trois propriétés notables :

- Une importante réduction du nombre d'itérations nécessaires à la stabilisation des centres des nuées.
- Une plus grande adaptativité de la segmentation car le centre de chaque nuée peut lentement se déplacer.
- Chaque classe n'est *pas* nécessairement représentée dans chaque fenêtre.

Nous avons choisi de sérialiser le *k*-means le long de l'axe des $x$ car la plupart de nos images représentent des livres ouverts et sont donc plus larges que hautes. Des tests nous ont montré que la sérialisation le long de l'axe des $y$ n'apporte pas de meilleurs résultats et peut conduire à des temps de calculs plus longs. Au début de chaque ligne, les centres sont initialisés avec les centres originaux (voir section 2.2) puis, le long de la ligne, nous nous assurons que les centres ne se croisent pas (voir section 2.3).

Notre *k*-means sérialisé est indépendant du choix des caractéristiques et de la dimension de l'espace des caractéristiques. L'expérience nous a montré que nous avons besoin du maximum d'information pour séparer, par exemple, des caractères du verso visible en transparence de caractères dégradés du recto qui ont à peu près les mêmes teinte et luminosité. Pour cela nous avons utilisé les canaux RVB [WES 01], YUV[1], TSL (Teinte, Saturation, Luminosité) [ZHA 00], qui est une transformation non linéaire de l'espace RVB, et des caractéristiques de textures [EGL 04].

## 2.2 Initialisation de l'algorithme

Les images de documents étant bruitées et inégalement illuminées, certains pixels d'une même classe situés à différents endroits d'une image peuvent être très différents. De plus, certaines classes peuvent avoir une frontière très ténue (par exemple le texte du verso et du texte estompé du recto) que nous devons décrire avec précision, c'est à dire, en définissant plusieurs nuées par classe logique. Nous utilisons donc une classification hiérarchique : des méta-classes composées de plusieurs nuées dans l'espace des caractéristiques [PUZ 00, LAM 03].

[1] l'encodage luminance/chrominance des télévisions PAL

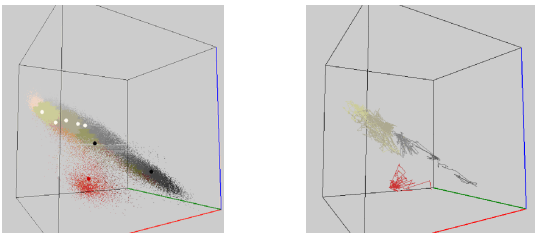

FIG. 2 – a) Couleurs dans l'espace RVB et centres des nuées, b) Déplacement des centres des nuées le long d'une ligne.

En général, la répartition des couleurs d'images de documents dans les nuées n'est pas gaussienne [TOD 00], si bien qu'il est difficile de déterminer le nombre de nuées nécessaire. Cela explique pourquoi c'est l'utilisateur qui doit décider du nombre de classes et de nuées ainsi que de leurs centres initiaux. Dans notre application, une interface utilisateur permet de définir le nombre de classes logiques et de sélectionner les échantillons de couleurs qui initialiseront les nuées de chaque classe que nous noterons $\left\{C_i^{\text{init}}\right\}_{i\in[1\ldots k]}$. Ces échantillons de couleurs sont la moyenne des caractéristiques des pixels contenus dans des portions de l'image désignées par l'utilisateur. Cette initialisation n'est effectuée qu'une seule fois pour toutes les images d'un même livre. Le nombre $k$ de nuées est égal au nombre d'échantillons désignés et non pas au nombre $M$ de classes logiques. L'utilisateur désignera au moins une nuée par classe logique ($k \geq M$). La figure 2a représente les couleurs de l'image figure 4a et les centres des nuées dans l'espace 3D RVB. Cette image est décrite par trois classes ($M = 3$) définies par différents nombres de nuées ($\omega_1$ représente le texte rouge avec un centre, $\omega_2$ représente le texte noir avec 2 centres et $\omega_3$ représente le papier et le verso avec 5 centres).

## 2.3 Protection contre le croisement des classes

Les classificateurs non supervisés comme l'algorithme *k*-means peuvent facilement laisser les centres des nuées se croiser. Dans le cas de la sérialisation, cela crée des inversions entre les nuées et les étiquettes des classes. Nous protégeons notre algorithme contre ce phénomène après la stabilisation des centres dans chaque fenêtre pour garantir la convergence de la classification et la validité de l'étiquetage. Pour cela, dans chaque fenêtre $w_P$, après l'adaptation du *k*-means et avant l'étiquetage de $P$, nous calculons la distance entre chaque nouveau centre de nuée $C_i^{w_P}$ et un ensemble de centres de référence $\left\{C_i^{\text{ref}_{w_P}}\right\}_{i\in[1\ldots k]}$. Si le centre de référence le plus proche n'est pas celui portant la même étiquette, alors $C_i^{w_P}$ est déplacé vers le centre de référence $C_i^{\text{ref}_{w_P}}$ :

$$\forall i \in [1\ldots k],\quad \underset{j\in[1\ldots k]}{\operatorname{argmin}}\ d\left(C_i^{w_P}, C_j^{\text{ref}_{w_P}}\right) \neq i \Rightarrow C_i^{w_P} \leftarrow C_i^{\text{ref}_{w_P}}$$

Les centres de référence, qui sont recalculés dans chaque fenêtre, sont des combinaisons linéaires[2] des

[2] ou circulaires dans le cas de la Teinte

centres précédents $\left\{C_i^{w_{P^-}}\right\}_{i\in[1\ldots k]}$ et des centres originaux $\left\{C_i^{\text{init}}\right\}_{i\in[1\ldots k]}$ que l'utilisateur a sélectionné lors de la phase d'initialisation.

$$\forall i \in [1\ldots k], C_i^{\text{ref}_{w_P}} = (1-\lambda)\cdot C_i^{\text{init}} + \lambda \cdot C_i^{w_{P^-}},\ \lambda \in [0,1]$$

De meilleurs résultats sont obtenus avec $0 \leq \lambda \leq 0,7$.
Cette limitation des mouvements des centres ne restreint pas l'adaptation des nuées. La figure 2b illustre l'adaptation des centres dans l'espace RVB le long d'une ligne de l'image figure 4a en utilisant des centres de référence égaux aux centres précédents.

## 2.4 Étiquetage des pixels

En temps normal, on affecte à un point $P$ l'étiquette du centre de nuée le plus proche. Cela pose des problèmes dans le cas des images tramées car une couleur apparente peut être le résultat de la juxtaposition de deux couleurs différentes.
Considérons une image avec deux couleurs, $c_1$ et $c_2$ et un niveau de tramage. Nous allons tenter de segmenter cette image en trois classes. Comme notre algorithme initialise les centres des nuées avec la moyenne de vecteurs de caractéristiques calculée dans des fenêtres données par l'utilisateur, les centres seront $C_1^{\text{init}}$ de couleur $c_1$, $C_2^{\text{init}}$ de couleur $c_2$ et $C_3^{\text{init}}$ de couleur $\mu_f(c_1,c_2)$, avec $\mu_f$ la fonction moyenne adaptée à l'espace des caractéristiques choisi. Si nous ne faisons pas attention, aucun pixel ne sera classifié dans la nuée $C_3$ car tous les pixels présents dans l'image sont de couleur $c_1$ ou $c_2$. C'est un problème qu'aucun classificateur classique de peut résoudre, même s'il est adaptatif et sérialisé. En partant du principe que, dans une fenêtre donnée, $C_3^{w_P}$ ne s'est vue assigner aucun pixel, si les nuées $C_1^{w_P}$ et $C_2^{w_P}$ ont approximativement le même nombre d'échantillons, nous calculons les barycentres spatiaux $B_1$ et $B_2$ des pixels de la fenêtre qui sont de couleur $c_1$ et $c_2$ (voir la figure 3). De ce calcul, nous pouvons conclure que :

1. si $B_1$ et $B_2$ sont distincts, la fenêtre est centrée sur la frontière entre deux régions uniformes de couleurs $c_1$ et $c_2$,
2. si $B_1$ est très proche de $B_2$, la fenêtre est centrées sur une partie tramée de l'image *ou* la fenêtre est centrée sur un type particulier de figure symétrique. Une analyse fréquentielle peut nous amener à une séparation plus précise de ces deux derniers cas (les tramages ont une signature fréquentielle particulière).

Dans le second cas, nous décidons d'appliquer un algorithme spécifique pour traiter les images tramées. Cela peut conduire à une mauvaise classification de certaines figures symétriques mais c'est un moindre mal que de laisser $C_3$ sans aucun échantillon attribué.
En général, on utilise un algorithme de lissage pour traiter les images tramées, mais dans notre cas, cela détruirait les plus fines lignes des caractères et accollerait certaines figures géographiquement proches. Nous adaptons donc la classification dans le second cas en choisissant l'étiquette d'un pixel selon le critère suivant :

$$l = \underset{i\in[1\ldots k]}{\text{argmin}}\ \left\{d\left(f(P),C_i^{w_P}\right), d\left(f\left(\mu_\sigma(w_P)\right),C_i^{w_P}\right)\right\}$$

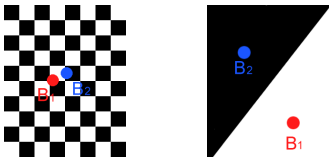


FIG. 3 – Barycentres spatiaux d'une image tramée et d'une image lisse.

où $l$ est l'étiquette donnée au pixel $P$, $\{C_i^{w_P}\}$ les centres des nuées dans la fenêtre $w_P$, $f$ la fonction qui extrait les caractéristiques d'un point et $\mu_\sigma(w_P)$ la moyenne gaussienne $w_P$ d'écart-type $\sigma$ et centrée sur $P$.
Le paramètre $\sigma$ permet de traiter des problèmes très locaux en lissant la classification des pixels sans lisser l'image. Les images bruitées ou tramées sont généralement traitées avec $0,7 \leq \sigma \leq 1,4$ pour lisser la classification alors que les images de documents propres sont traitées sans lissage ($\sigma \leq 0,5$).

Un autre paramètre, $\rho$, permet de contrôler la sérialisation de l'algorithme. Il définit la distance entre un pixel et le centre le plus proche de ce dernier au delà de laquelle ce pixel n'est plus pris en compte pour calculer le nouveau centre. Donc, si $\rho = 0$, la sérialisation est bloquée et l'algorithme est un simple $k$-means séquentiel. Si $\rho \to \infty$ alors l'algorithme est complètement sérialisé.
Il est utile de pouvoir contrôler la sérialisation lorsque l'on traite des images tramées car dans ce cas, si l'algorithme est complètement sérialisé, il s'adaptera à chaque couleur du tramage et le résultat sera aussi tramé. En général, pour les images tramées, on utilise $\sigma > 0,5$ et $\rho = 0$ et pour les images lisses présentant des variations d'illumination, on utilise $\sigma \leq 0,5$ et $\rho \to \infty$.

## 2.5 Résumé

Soit $f$ la fonction que l'on applique à un point $P$ pour obtenir son vecteur de caractéristiques. Pour chaque $w_P$ centrée sur $P$, nous calculons d'abord les nouveaux centres des nuées. C'est un calcul de $k$-means traditionnel auquel on ajoute la condition de $\rho$ :

$$\forall i \in [1\ldots k], C_i^{w_P} =$$

$$\mu_f\left(\left\{f(Q), Q \in w_P / \left\{\begin{array}{l} \underset{j\in[1\ldots k]}{\text{argmin}}\ d\left(f(Q),C_j^{w_{P^-}}\right) = i \\ \underset{j\in[1\ldots k]}{\min}\ d\left(f(Q),C_j^{w_{P^-}}\right) < \rho \end{array}\right.\right\}\right)$$

Ensuite, nous empêchons les centres de se croiser en les ramenant aux centres de référence s'ils bougent trop :

$$\forall i \in [1\ldots k], C_i^{\text{ref}_{w_P}} = (1-\lambda)\cdot C_i^{\text{init}} + \lambda \cdot C_i^{w_{P^-}},\ \lambda \in [0,1]$$

$$\forall i \in [1\ldots k],\ \underset{j\in[1\ldots k]}{\text{argmin}}\ d\left(C_i^{w_P}, C_j^{\text{ref}_{w_P}}\right) \neq i \Rightarrow C_i^{w_P} \leftarrow C_i^{\text{ref}_{w_P}}$$

Nous calculons les barycentres spatiaux des nuées les plus peuplées :

$$a = \underset{i\in[1\ldots k]}{\text{argmin}}\ \text{card}\left(C_i^{w_P}\right),\ b = \underset{i\in[1\ldots k]\setminus a}{\text{argmin}}\ \text{card}\left(C_i^{w_P}\right)$$

$$B_1 = \mu\left(\left\{Q \in w_P / \underset{i\in[1\ldots k]}{\operatorname{argmin}} \; d(f(Q), C_i^{w_P}) = a\right\}\right)$$

$$B_2 = \mu\left(\left\{Q \in w_P / \underset{i\in[1\ldots k]}{\operatorname{argmin}} \; d(f(Q), C_i^{w_P}) = b\right\}\right)$$

Finalement, nous classifions le point $P$ selon que la partie de l'image dans $w_P$ est tramée ou non :

- $\begin{cases} \frac{\text{card}\, a}{\text{card}\, b} \approx 1 \\ d(B_1, B_2) < \varepsilon \end{cases} \Rightarrow$

$$l(P) = \underset{i\in[1\ldots k]}{\operatorname{argmin}} \; \{d(f(P), C_i^{w_P}), d(f(\mu_\sigma(w_P)), C_i^{w_P})\}$$

- $\begin{array}{l} \frac{\text{card}\, a}{\text{card}\, b} \neq 1 \\ \text{ou} \quad d(B_1, B_2) \geq \varepsilon \end{array} \Rightarrow$

$$l(P) = \underset{i\in[1\ldots k]}{\operatorname{argmin}} \; d(f(P), C_i^{w_P})$$

# 3 Résultats

## 3.1 Segmentation d'images de documents

Nous avons validé notre algorithme sur des images de manuscrits médiévaux de la bibliothèque *Mazarine* qui nous ont été fourni par l'IRHT[3]. Ces images contiennent du texte noir et rouge et des enluminures. Le texte du verso est visible en transparence et les caractères du recto sont partiellement estompés (figure 4a). Une binarisation traditionnelle ne peut pas mener à une segmentation correcte de ces images. Pour illustration, la figure 4b montre le résultat de l'application de l'algorithme de Sauvola [SAU 97]. Nous utilisons l'initialisation décrite dans la figure 2. La segmentation en 3 classes est donnée ci-dessous dans les figures 4c et 4d (le fond n'est pas représenté ici). Les paramètres sont une fenêtre $6 \times 6$, $\lambda = 0,5$, $\sigma = 0,5$ et $\rho = 50000$ et les caractéristiques sont les canaux RVB et TSL. Une fois filtrées, ces images peuvent servir de masque pour créer une image artificielle du document et partiellement le restaurer. Le fond est remplacé par sa moyenne et les couches rouge et noire sont copiées telle quelles (figure 4e).

## 3.2 Binarisation

Nous avons aussi étudié la capacité de notre algorithme à binariser directement les images en couleur et l'avons comparé à des algorithmes de binarisation appliqués à des images de luminance. La figure 5b montre l'application de l'algorithme de Sauvola sur le canal de luminance de l'image figure 5a dans une fenêtre $18 \times 18$. Il reste des artefacts sur les bords de la tâche. Notre algorithme, appliqué sur une fenêtre de seulement $6 \times 6$, avec $\lambda = 0$, $\sigma = 0,5$ et $\rho = 50000$ sur les canaux RVB et TSL (figure 5c) avec deux classes (une nuée par classe), tire avantage de l'analyse directe de la couleur et démontre son adaptativité.

[3] Institut de Recherche sur l'Histoire des Textes

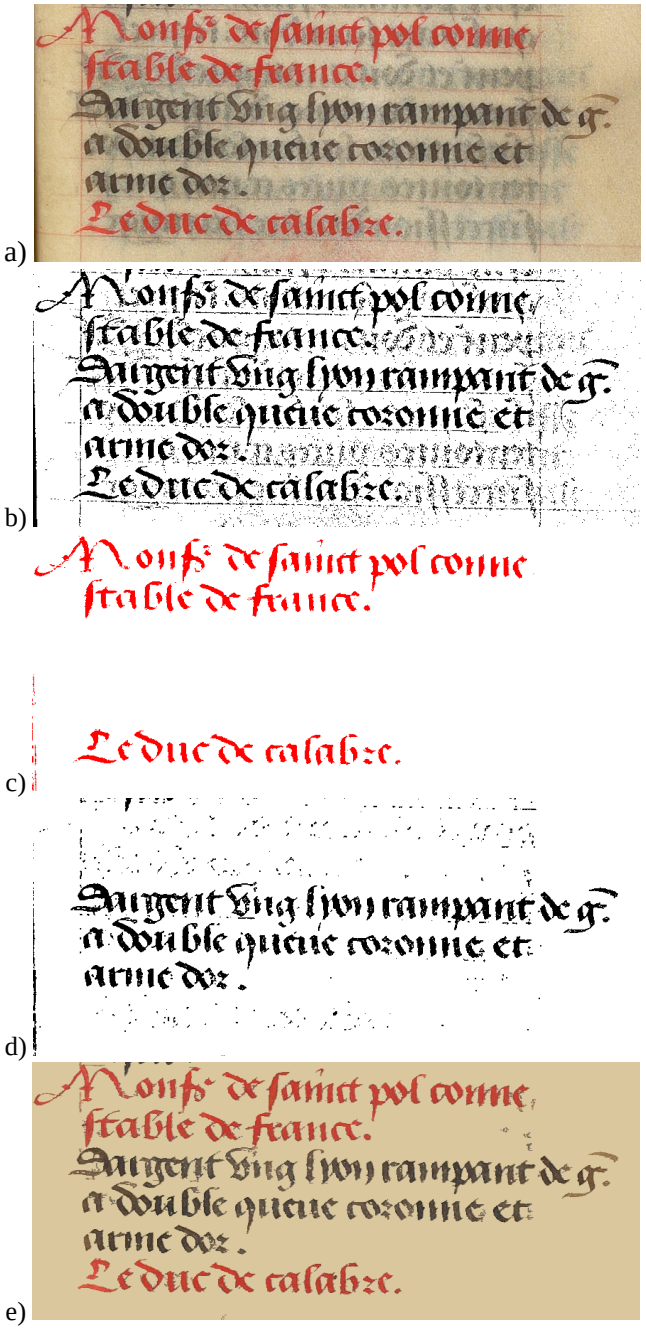


FIG. 4 – a) Image originale de manuscrit médiéval de la bibliothèque *Mazarine*, b) Application de l'algorithme de Sauvola, c) Texte rouge extrait par *k*-means sérialisé, d) Texte noir extrait par *k*-means sérialisé, e) Image restaurée.

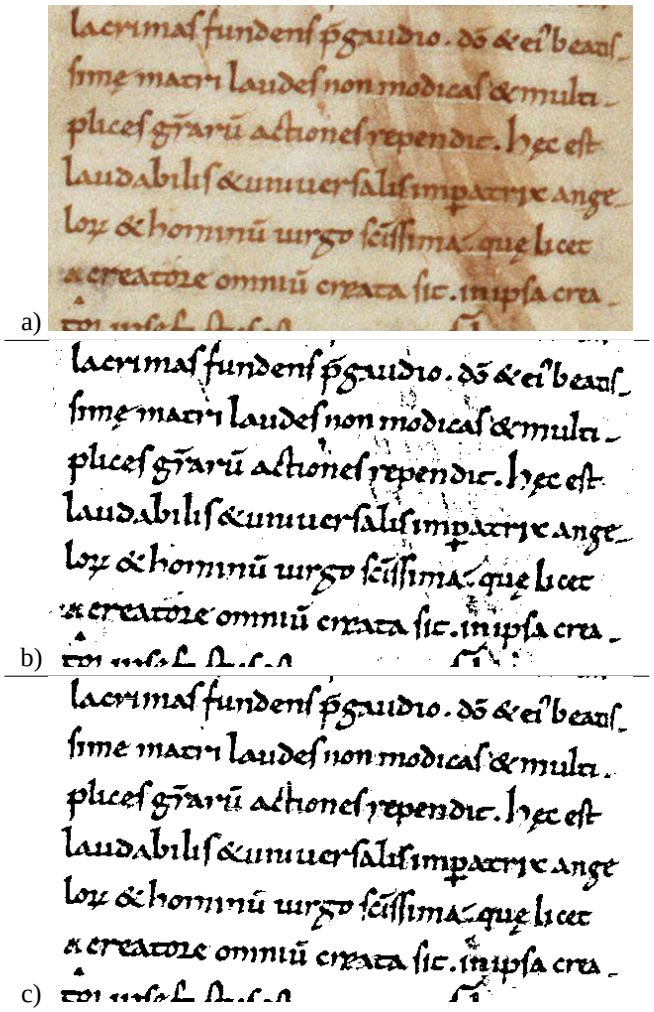


FIG. 5 – a) Image originale de manuscrit médiéval de la bibliothèque de *Vendôme*, b) Binarisation par l'algorithme de Sauvola, c) Binarisation par 2-means sérialisé.

FIG. 6 – Application du *k*-means sérialisé aux images de vidéo.

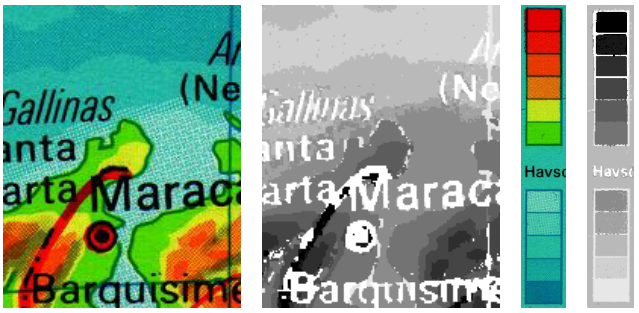


FIG. 7 – Indexation d'une image de carte géographique tramée. La légende a été utilisée pour initialiser le processus.

### 3.3 Segmentation d'images de vidéo

Notre algorithme est suffisamment générique pour être appliqué sur une grande variété d'images dont, par exemple, les images bruitées issues de vidéos. Son adaptativité lui permet d'ignorer le bruit de la compression MPEG (figure 6). Les paramètres sont $\lambda = 0,5$, $\sigma = 1$ et $\rho = 50000$ dans une fenêtre $6 \times 6$ et les caractéristiques sont les canaux RVB et TSL.

### 3.4 Segmentation couleur d'images tramées

La figure 7 montre un exemple d'image de carte géographique tramée que nous segmentons et indexons. Il s'agit de l'image "`P03048.jpg`" de la base *MediaTeam*. Les caractéristiques sont les canaux RVB et TSL. Les paramètres sont une fenêtre $6 \times 6$, $\lambda = 0$, $\sigma = 0,8$ et $\rho = 0$. Nous avons utilisé la légende en couleur pour initialiser les nuées. Notre algorithme a su gérer la mauvaise qualité de l'image, notamment les cinq couleurs bleues apparentes réalisées à partir de tramages de deux couleurs.

### 3.5 Performances

La réutilisation des centres de la classification précédente sur une même ligne nous permet de réduire le nombre d'itérations pour stabiliser les *k*-means d'un facteur allant jusqu'à 17% de nombre d'itérations pour un *k*-means séquentiel. Le nombre moyen d'itérations pour chaque déplacement de fenêtre se situe entre 2,2 et 3. Cependant, l'algorithme que nous proposons, comme toutes les méthodes de segmentation adaptatives, peut devenir très lourd en calculs selon la taille de la fenêtre et la dimension de l'espace des caractéristiques. Le traitement d'une image

en $3000 \times 2000$ avec une fenêtre de $6 \times 6$, un espace de caractéristiques de dimension 6, $\lambda = 0$, $\sigma = 0.5$ et $\rho = 50000$ est effectué en 500 secondes sur un PC à 1,5GHz.

# 4 Conclusion

Nous avons présenté un algorithme de segmentation adaptative pour les images de documents en couleur basé sur une sérialisation de l'algorithme *k*-means appliqué séquentiellement sur chaque ligne. Nous avons aussi proposé de représenter chaque classe logique par plusieurs nuées dans l'espace des caractéristiques. L'initialisation des centres des nuées n'est pas une phase critique de par l'adaptativité de l'algorithme. Le nombre de nuées par classe et la taille de la fenêtre ont une influence limitée sur les résultats de la segmentation. Une table sera produite pour aider l'utilisateur à choisir les paramètres $\lambda$, $\sigma$ et $\rho$ qui ont un impact important sur la pertinence de la segmentation et sur le temps de calcul. Les résultats sont assez bons sur toutes les images de manuscrits médiévaux que nous avons traités. Nous avons aussi testé notre algorithme sur de nombreux types d'images telles que des images de vidéo, des images naturelles et des cartes. Dans un futur proche, nous travaillerons à simplifier l'utilisation de cet algorithme et à assister l'utilisateur en définissant automatiquement certains paramètres tels que le nombre de classes et les échantillons pour initialiser les nuées.

# Références


[EGL 04] EGLIN V., BRES S., Analysis and interpretation of visual saliency for document functional labeling, *à paraître dans International Journal of Document Analysis and Recognition (IJDAR)*, , 2004.

[LAM 03] LAMBERT P., GRECU H., A quick and coarse color image segmentation, *International Conference on Image Processing (ICIP)*, Barcelone, 2003, pp. 965-968.

[LEB 04] LEBOURGEOIS F., AL., Document Images Analysis solutions for Digital libraries, *First International Workshop on Document Image Analysis for Libraries (DIAL)*, Palo Alto, 2004.

[PUZ 00] PUZICHA J., BELONGIE S., Model-based halftoning for color image segmentation, *International Conference on Pattern Recognition (ICPR)*, Barcelone, 2000, pp. 3633-3637.

[SAU 97] SAUVOLA J., SEPPÄNEN T., HAAPAKOSKI S., PIETIKÄINEN M., Adaptive Document Binarization, *International Conference on Document Analysis and Recognition*, vol. 1, Ulm, 1997, pp. 147–152.

[TOD 00] TODORAN L., WORRING M., Segmentation of Color Document Images, *ISIS technical report series*, vol. 21, 2000.

[WES 01] WESOLKOWSKI S., TOMINAGA S., DONY R., Shading and Highlight Invariant Color Image Segmentation, *SPIE*, vol. 4300, 2001, pp. 229-240.

[ZHA 00] ZHANG C., WANG P., A new method of color image segmentation based on intensity and hue clustering, *International Conference on Pattern Recognition (ICPR)*, Barcelone, 2000, pp. 3617-3621.