

N° d'ordre : 298/20



Université Sultan Moulay Slimane  
Faculté des Sciences et Techniques  
Béni Mellal



*Centre d'Etudes Doctorales : Sciences et Techniques*

*Formation doctorale : Mathématiques et Physiques Appliquées*

## Thèse

Présentée par

**Meryem AMEUR**

Pour l'obtention du grade de

**Docteur**

*Spécialité : Informatique*

---

## Contributions à la segmentation des images par les modèles de Markov cachés

---

Soutenue le 19/Décembre/2020 devant le jury :

Mohamed FAKIR	Professeur, Faculté des Sciences et Techniques, Béni Mellal	Président
Noureddine EL BARBRI	Professeur, École Nationale des Sciences Appliquées, Khouribga	Rapporteur
Mohamed EL KAMILI	Professeur, École Supérieure de Technologie, Casablanca	Rapporteur
Mohamed BASLAM	Professeur, Faculté des Sciences et Techniques, Béni Mellal	Rapporteur
Karim RHOFIR	Professeur, École Nationale des Sciences Appliquées, Khouribga	Examineur
Najlae IDRISSE	Professeur, Faculté des Sciences et Techniques, Béni Mellal	Co-encadrant
Cherki DAOU	Professeur, Faculté des Sciences et Techniques, Béni Mellal	Directeur de thèse

# Remerciements

Les travaux présentés dans ce document ont été réalisés au sein du laboratoire de Traitement de l'Information et d'Aide à la Décision(TIAD) à la Faculté des Sciences et Techniques, Université Sultan Moulay Slimane, Beni Mellal, Maroc.

Je tiens d'abord à remercier mes encadrants Monsieur **Cherki DAOUI** et Madame **Najlae IDRISSE** qui m'ont beaucoup aidé à réaliser cette thèse, par leurs conseils, leur disponibilité et leurs remarques.

Aussi, j'adresse mes remerciements aux rapporteurs et examinateurs de cette thèse qui ont acceptés de juger mon travail.

Mes remerciements s'adressent également à tous les membres du laboratoire de TIAD et à tous les enseignants du département informatique de la Faculté des Sciences et Techniques de Beni Mellal.

Enfin, je tiens à remercier mes proches, mes amis et toutes les personnes qui ont participé de près ou de loin à la réalisation de cette thèse.

Ce travail est dédié aux personnes qui me sont chères.

# Résumé

L'application des modèles de Markov cachés en traitement d'images, connaissait un essor considérable à partir des années quatre-vingt. Leur succès est dû à leur aptitude à produire, lorsque les divers bruits présents dans l'image considérée sont importants et lorsque les données correspondent bien au modèle utilisé, des résultats spectaculaires dépassant parfois les capacités de l'œil humain. En segmentation d'images, la problématique traitée dans cette thèse, la plupart des images sont de grande taille, ce qui augmente le temps de calcul de paramètres et de processus estimés, et rend les méthodes classiques impraticables. Ainsi, l'objectif principal de ce travail est d'alléger cette malédiction de dimension.

Ce travail est divisé en deux parties : la première comporte un ensemble d'études comparatives entre les estimateurs itératifs des paramètres du modèle chaîne de Markov cachée ; entre les stratégies Bayésiennes utilisées pour estimer la configuration finale du processus caché  $X$  ; et entre les chaînes de Markov cachées utilisées en segmentation d'images. Elle présente, également, des comparaisons entre les chaînes de Markov cachées à processus caché stationnaire et non-stationnaire. Enfin, elle expose les modèles de Markov cachés flous. Nous utilisons ces approches pour segmenter des images : niveau de gris, couleurs, texturées, cérébrales IRM et flous.

La deuxième partie expose quelques nouvelles méthodes algorithmiques réduisant le temps d'exécution du modèle chaîne de Markov cachée à bruit indépendant. Nous exposons trois méthodes, la première est basée sur la technique "diviser pour régner", à estimation indépendante qui divise l'image en blocs de même taille, chaque bloc s'exécute indépendamment des autres blocs d'une manière séquentielle. Et la deuxième, à estimation dépendante où le traitement de chaque bloc courant dépend du bloc précédent. Ces approches sont utilisées pour segmenter des images niveau de gris et couleur de grande taille. Dans la troisième méthode, nous considérons un modèle non-stationnaire d'une chaîne de Markov triplet basé sur l'estimation dépendante des paramètres. Ensuite, nous utilisons le parallélisme pour estimer simultanément les configurations finales du processus caché  $X$  et du processus auxiliaire  $U$ . Cette approche est utile pour segmenter des images cérébrales IRM. En outre, pour l'extraction de la région d'intérêt (la tumeur), nous proposons une nouvelle technique d'extraction basée sur une hybridation entre la technique de seuillage et les opérations de morphologie mathématique.

Finalement, les résultats expérimentaux obtenus sont très encourageants, nous arrivons à réduire clairement le temps d'exécution et à faire l'opération de localisation de la tumeur avec une grande

précision.

Mots-clés : Segmentation d' images, Chaîne de Markov cachée stationnaire, Diviser pour régner, Chaîne de Markov cachée non-stationnaire, Parallélisme, Technique de seuillage, Morphologie Mathématique.

# Abstract

The application of hidden Markov models in image processing was booming in 1980s. Their success is due to their ability to produce, when the various noises present in the image under consideration are important and when the data correspond to the model used, sometimes the spectacular results beyond the capabilities of the human eye. In image segmentation, the problematic treated in this thesis, most images are large, which increases the calculation time of parameters and processes to be estimated, and makes the classical methods impractical. Thus, the main objective of this work is to alleviate this dimensional curse.

This work is divided into two parts : the first one includes a set of comparative studies between the iterative estimators of the parameters of Hidden Markov Chain model ; between the Bayesian strategies used to estimate the final configuration of hidden process  $X$  ; and between the Hidden Markov chains used in image segmentation. It also presents comparisons between stationary and non-stationary Hidden Markov chains. Finally, it exhibits the fuzzy Markov models. We use these approaches to segment images : gray level, color, textured, cerebral MRI and fuzzy.

The second part presents some new algorithmic methods to reduce the execution time of Hidden Markov Chain with independent model. We expose three methods, the first one is based on the "divide and conquer" technique, with independent estimation, which divides the image into blocks of the same size, each block executes independently from the other blocks by sequential way. And the second one, with dependent estimation, where, the processing of each current block depends on the previous block. These approaches are used to segment large gray level and color images. In the third method, we consider a non-stationary model triplet Markov chain based on dependent estimation of parameters. Then, we use the parallelism to estimate the final configurations of the hidden process  $X$  and the auxiliary process  $U$  simultaneously. This approach is useful for segmenting MRI brain images.

In addition, for the extraction of the region of interest (the tumor), we propose a new extraction technique based on a hybrid between the threshold technique and the mathematical morphology operations.

Finally, the experimental results obtained are very encouraging, we arrive to reduce the execution time clearly and to do the tumor localization operation with great precision.

Keywords : Image segmentation, Stationary Hidden Markov Chain, Divide and conquer, Non-Stationary Hidden Markov Chain, Parallelism, Threshold, Mathematical Morphology.

# Publications

## Articles publiés :

1. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Fuzzy Hidden Markov Model in Images segmentation. Soumission acceptée dans : Recent Advances in Intuitionistic Fuzzy Logic Systems and Mathematics, Studies in Fuzziness and Soft Computing 395, [https://doi.org/10.1007/978-3-030-53929-0\\_6](https://doi.org/10.1007/978-3-030-53929-0_6).
2. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Markovian Segmentation of Textured Color Images, pp.1-5, 2020,10.1109/ISCV49265.2020.9204066.
3. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Fast Markovian Images Segmentation, pp.1-6, 2020, DOI :10.1109/ICOA49421.2020.9094479.
4. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Hierarchical Hidden Markov Models in Image Segmentation , Scientific Visualization Journal, Vol. 12, No. 1, pp.22-47, March 2020, DOI : 10.26583/sv.12.1.03, <http://sv-journal.org/2020-1/03/>.
5. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Some Stationaries Hidden Markov Models , International Journal of Advanced Science and Technology, Vol. 29, No. 1, pp. 1043 –1057, January 2020, <http://sersc.org/journals/index.php/IJAST/article/view/3597>.
6. Meryem Ameer, Najlae Idrissi, Cherki Daoui : Color Images Segmentation using Pairwise Markov Chain, pp.1-6, 2018, DOI :10.1109/ICMCS.2018.8525988.
7. Meryem Ameer, Najlae Idrissi, Cherki Daoui : Triplet Markov Chain in Images Segmentation, pp. 1-8, 2018, DOI :10.1109/ISACV.2018.8354055.
8. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Markovian Segmentation of Brain Tumor MRI Images ,pp.155-165, 2017, <http://doi.org/10.11591/ij-ict.v6i3.10079>.
9. Meryem Ameer, Najlae Idrissi, Cherki Daoui : Markovian Segmentation of Color and Gray Level Images,pp.259-264,2016, DOI :10.1109/CGiV.2016.57.

## Articles soumis :

1. Meryem Ameer, Cherki Daoui, Najlae Idrissi : Parallel Hierarchical Triplet Markov Chain.

# Communications orales

1. Markovian Segmentation of Textured Color Images, 4th Edition of International Conference on Intelligent Systems and Computer Vision (ISCV20), Fez, 09-11 June 2020.
2. Fast Markovian Images Segmentation, 6th Edition of International Conference on Optimization and Applications (ICOA20), Beni Mellal, 20-21 April 2020.
3. Fuzzy Hidden Markov Model using in images Segmentation, 6th Edition of International Congress of the Moroccan Society of Applied Mathematics(SM2A19), Beni Mellal, 07-09 November 2019.
4. Color Image Segmentation using Pairwise Markov Chain, 6th Edition of International Conference On Multimedia Computing and Systems(ICMCS18), Rabat, 10-12 May 2018.
5. Triplet Markov Chain in Image Segmentation, 3th Edition of International Conference on Intelligent Systems and Computer Vision (ISCV18), Fez, 02-04 April 2018.
6. Markovian Segmentation of Brain Tumor MRI Images, 3th Edition of International Conference On Business Intelligence (CBI17), Beni Mellal, 29-31 April 2017.
7. Markovian Segmentation of Color and Gray Level Images, 13thEdition of International Conference On Computer Graphics, Imaging and Visualization(CGIV16), Beni Mellal ,30 March- 01 April 2016.

# Table des matières

<b>Remerciements</b>	<b>2</b>
<b>Résumé</b>	<b>3</b>
<b>Abstract</b>	<b>5</b>
<b>Publications</b>	<b>6</b>
<b>Table des figures</b>	<b>11</b>
<b>Liste des tableaux</b>	<b>14</b>
<b>Liste des algorithmes</b>	<b>17</b>
<b>Liste des abréviations</b>	<b>18</b>
<b>Introduction Générale</b>	<b>19</b>
<b>1 Généralités sur la segmentation markovienne des images</b>	<b>24</b>
1.1 Introduction . . . . .	24
1.2 Les modèles de Markov cachés . . . . .	25
1.3 Les phases d'estimation des paramètres . . . . .	26
1.4 Les champs de Markov cachés . . . . .	27
1.4.1 Système de voisinage . . . . .	27
1.4.2 Champs de Markov caché . . . . .	29
1.5 Les chaînes de Markov cachées . . . . .	31
1.5.1 Chaîne de Markov . . . . .	32
1.5.2 Chaîne de Markov cachée . . . . .	33
1.6 Les arbres de Markov cachés . . . . .	34
1.6.1 Arbre de Markov . . . . .	35



1.6.2	Arbre de Markov caché . . . . .	35
1.7	Les modèles de Markov couples . . . . .	36
1.8	Les modèles de Markov triplets . . . . .	37
1.9	Les modèles de Markov Flous . . . . .	39
1.10	Conclusion . . . . .	41
<b>2</b>	<b>Les chaines de Markov cachées en segmentation des images</b>	<b>42</b>
2.1	Introduction . . . . .	42
2.2	Les chaines de Markov cachées à bruit indépendant . . . . .	42
2.2.1	Présentation de modèle . . . . .	43
2.2.2	Estimation des paramètres . . . . .	44
2.2.3	Résultats et expérimentations . . . . .	49
2.3	Les chaines de Markov couples . . . . .	68
2.3.1	Présentation de modèle . . . . .	68
2.3.2	Estimation des paramètres . . . . .	70
2.3.3	Résultats et expérimentations . . . . .	73
2.4	Étude Comparative entre les chaines de Markov cachées stationnaires . . . . .	78
2.4.1	Présentation des modèles . . . . .	79
2.4.2	Estimation des paramètres . . . . .	80
2.4.3	Résultats et expérimentations . . . . .	82
2.5	Les chaines de Markov triplets . . . . .	87
2.5.1	Présentation de modèle . . . . .	87
2.5.2	Estimation des paramètres . . . . .	88
2.5.3	Résultats et expérimentations . . . . .	89
2.6	Les chaines de Markov cachées floues . . . . .	98
2.6.1	Présentation de modèle . . . . .	99
2.6.2	Estimation des paramètres . . . . .	100
2.6.3	Résultats et expérimentations . . . . .	103
2.7	Conclusion . . . . .	120
<b>3</b>	<b>Les chaines de Markov cachées hiérarchiques</b>	<b>121</b>
3.1	Introduction . . . . .	121
3.2	Approche 1 : Approche d'estimation indépendante . . . . .	122
3.3	Approche 2 : Approche d'estimation dépendante . . . . .	125
3.3.1	Approche 2-1 . . . . .	128
3.3.2	Approche 2-2 . . . . .	129
3.4	Résultats et expérimentations . . . . .	131
3.4.1	Résultats Visuels . . . . .	131
3.4.2	Résultats Quantitatifs . . . . .	136

3.5	Conclusion . . . . .	139
<b>4</b>	<b>Les chaines de Markov triplet hiérarchiques</b>	<b>140</b>
4.1	Introduction : . . . . .	140
4.2	Approche 1 : les chaines de Markov triplet hiérarchiques . . . . .	141
4.3	Approche 2 : les chaines de Markov triplet parallèles . . . . .	144
4.4	Approche 3 : les chaines de Markov triplet hiérarchiques parallèles . . . . .	147
4.5	Extraction de la région d'intérêt : Approche proposée . . . . .	148
4.6	Résultats et expérimentations . . . . .	150
4.6.1	Résultats Visuels . . . . .	151
4.6.2	Résultats Quantitatifs . . . . .	156
4.7	Conclusion . . . . .	160
	<b>Conclusion Générale</b>	<b>162</b>
	<b>Bibliographie</b>	<b>163</b>

# Table des figures

1.1	Les modèles de Markov cachés utilisés en segmentation des images . . . . .	26
1.2	La démarche de segmentation des images par les MMCs . . . . .	27
1.3	La grille S . . . . .	28
1.4	Les cliques associées à un système de voisinage quatre connexités et huit connexités . . . . .	29
1.5	Construction du parcours fractal de Hilbert-Peano. . . . .	32
1.6	Graphe de dépendance d'une chaîne de Markov . . . . .	32
1.7	Graphe de dépendance d'une chaîne de Markov cachée . . . . .	33
1.8	Quadarbre . . . . .	35
2.1	Résultats de segmentation d'une image niveau de gris . . . . .	51
2.2	Résultats de segmentation de l'image couleur de dinosaure du data set Wang [103] . . . . .	52
2.3	Résultats de segmentation d'une image texturée 1 avec $K = 2$ . . . . .	55
2.4	Résultats de segmentation d'une image texturée 2 avec $K = 3$ . . . . .	56
2.5	Résultats de segmentation d'une image texturée 3 avec $K = 2$ . . . . .	57
2.6	Résultats de segmentation d'une image texturée 4 avec $K = 3$ . . . . .	58
2.7	Résultats de segmentation d'une image texturée 5 avec $K = 2$ . . . . .	59
2.8	Résultats de segmentation et extraction de la ROI expérience 1 . . . . .	61
2.9	Résultats de segmentation et extraction de la ROI expérience 2 . . . . .	62
2.10	Résultats de segmentation et extraction de la ROI expérience 3 . . . . .	62
2.11	Résultats de segmentation et extraction de la ROI expérience 4 . . . . .	63
2.12	Résultats de segmentation et extraction de la ROI expérience 5 . . . . .	63
2.13	Résultats de segmentation et extraction de la ROI expérience 6 . . . . .	64
2.14	Résultats de segmentation et extraction de la ROI expérience 7 . . . . .	64
2.15	Résultats de segmentation et extraction de la ROI expérience 8 . . . . .	65
2.16	Résultats de segmentation et extraction de la ROI expérience 9 . . . . .	66
2.17	Résultats de segmentation et extraction de la ROI expérience 10 . . . . .	66
2.18	Résultats de segmentation d'une image couleur texturée 1 avec $K = 2$ . . . . .	75
2.19	Résultats de segmentation d'une image couleur texturée 2 avec $K = 2$ . . . . .	76

2.20	Résultats de segmentation d'une image couleur texturée 3 avec $K = 2$ . . . . .	77
2.21	Résultats de segmentation d'une image niveau de gris $K = 2$ . . . . .	84
2.22	Résultats de segmentation d'une image couleur $K = 2$ . . . . .	85
2.23	Résultats de segmentation d'une image niveau de gris bruitée $K = 2$ . . . . .	86
2.24	Résultats de segmentation 1 avec $K = 9$ et $M = 5$ . . . . .	91
2.25	Résultats de segmentation 2 avec $K = 11$ et $M = 6$ . . . . .	92
2.26	Résultats de segmentation 3 avec $K = 3$ et $M = 2$ . . . . .	92
2.27	Résultats de segmentation 4 avec $K = 5$ et $M = 3$ . . . . .	92
2.28	Résultats de segmentation 5 avec $K = 3$ et $M = 2$ . . . . .	93
2.29	Résultats de segmentation 6 avec $K = 3$ et $M = 4$ . . . . .	93
2.30	Résultats de segmentation 7 avec $K = 3$ et $M = 2$ . . . . .	93
2.31	Résultats de segmentation 1 avec $K = 7$ et $M = 3$ . . . . .	95
2.32	Résultats de segmentation 2 avec $K = 3$ et $M = 2$ . . . . .	95
2.33	Résultats de segmentation 3 avec $K = 3$ et $M = 2$ . . . . .	96
2.34	Résultats de segmentation 4 avec $K = 3$ et $M = 2$ . . . . .	96
2.35	Résultats de segmentation 5 avec $K = 3$ et $M = 2$ . . . . .	96
2.36	Résultats de segmentation 6 avec $K = 3$ et $M = 2$ . . . . .	97
2.37	Résultats de segmentation 7 avec $K = 3$ et $M = 2$ . . . . .	97
2.38	Résultats de segmentation 8 avec $K = 3$ et $M = 2$ . . . . .	97
2.39	Résultats de segmentation d'une image couleur floue 1. . . . .	107
2.40	Résultats de segmentation d'une image couleur floue 2. . . . .	109
2.41	Résultats de segmentation d'une image niveau de gris floue 1. . . . .	111
2.42	Résultats de segmentation d'une image niveau de gris floue 2. . . . .	114
2.43	Le temps d'exécution en secondes par estimateur . . . . .	118
2.44	La convergence par estimateur . . . . .	118
3.1	Démarche de l'approche standard avant décomposition . . . . .	122
3.2	Scénario de l'approche d'estimation indépendante . . . . .	123
3.3	Scénario de l'approche d'estimation dépendante . . . . .	125
3.4	Processus d'estimation dépendante : Approche 2 . . . . .	128
3.5	Processus d'estimation dépendante : Approche 2-1 . . . . .	129
3.6	Processus d'estimation dépendante : Approche 2-2 . . . . .	130
3.7	Résultats de la segmentation de l'image médicale du col de l'utérus . . . . .	132
3.8	Résultats de la segmentation de l'image couleur 1 . . . . .	133
3.9	Résultats de la segmentation de l'image couleur 2 . . . . .	134
3.10	Résultats de la segmentation de l'image satellitaire normalisée . . . . .	135
3.11	Le temps d'exécution en secondes par approche proposée . . . . .	137
3.12	Le nombre d'itérations pour atteindre la convergence par approche proposée . . . . .	138

4.1	Démarche de la segmentation d'images cérébrales IRM . . . . .	141
4.2	Démarche de l'approche 1 . . . . .	144
4.3	Pseudo-code python de multiprocessus . . . . .	145
4.4	Pseudo-code python de threads . . . . .	146
4.5	Démarche de l'approche 2 . . . . .	146
4.6	Démarche de l'approche 3 . . . . .	148
4.7	Résultat de segmentation 1. . . . .	151
4.8	Résultat de segmentation 2. . . . .	152
4.9	Résultat de segmentation 3. . . . .	153
4.10	Résultat de segmentation 4. . . . .	154
4.11	Résultat de segmentation 5. . . . .	155
4.12	Le temps d'exécution en secondes par approche proposée : étude comparative 1 . . .	157
4.13	Le temps d'exécution en secondes par approche proposée : étude comparative 2 . . .	158
4.14	La convergence de chaque approche proposée par expérience . . . . .	159

# Liste des tableaux

2.1	Les valeurs de l'indice de PSNR et de taux d'erreur de l'expérience 1	53
2.2	Les valeurs de l'indice de PSNR et de taux d'erreur de l'expérience 2	53
2.3	La convergence et le temps d'exécution par estimateur de l'expérience 1	53
2.4	La convergence et le temps d'exécution par estimateur de l'expérience 2	54
2.5	Le temps d'exécution de MPM et MAP	54
2.6	Les valeurs de l'indice de PSNR	60
2.7	Les valeurs de taux d'erreur	60
2.8	Les valeurs des indices de PSNR et SSIM	67
2.9	Les valeurs de taux d'erreur par estimateur	67
2.10	La convergence d'EM et ECI	68
2.11	Les valeurs de taux d'erreur	78
2.12	Les valeurs de PSNR et taux d'erreur	84
2.13	Les valeurs de PSNR et taux d'erreur	85
2.14	Les valeurs de PSNR et taux d'erreur	86
2.15	Les valeurs des indices PSNR et SSIM	94
2.16	Les valeurs de taux d'erreur	94
2.17	Les valeurs de PSNR et SSIM indices	98
2.18	Les valeurs de taux d'erreur	98
2.19	Les valeurs de PSNR et SSIM pour expérience 1	114
2.20	Les valeurs de taux d'erreur pour expérience 1	115
2.21	Les valeurs de PSNR et SSIM pour expérience 2	115
2.22	Les valeurs de taux d'erreur pour expérience 2	115
2.23	Les valeurs de PSNR et SSIM pour expérience 3	116
2.24	Les valeurs de taux d'erreur pour expérience 3	116
2.25	Les valeurs de PSNR et SSIM pour expérience 4	116
2.26	Les valeurs de taux d'erreur pour expérience 4	117
2.27	Étude comparative entre EM, SEM et ECI en terme de la complexité	119

3.1	Les conditions expérimentales des expériences . . . . .	131
3.2	Les valeurs de l'indice PSNR . . . . .	136
3.3	Les valeurs de l'indice SSIM . . . . .	136
3.4	Les valeurs de taux d'erreur . . . . .	136
4.1	les valeurs de l'indice PSNR . . . . .	156
4.2	les valeurs de l'indice SSIM . . . . .	156
4.3	Les valeurs de taux d'erreur . . . . .	156

# Liste des Algorithmes

1	La démarche de segmentation d'images par CHMC . . . . .	31
2	La démarche de segmentation d'images par CMC . . . . .	34
3	La démarche de segmentation d'images par AMC . . . . .	36
4	La démarche de segmentation d'images par MMCouple . . . . .	37
5	La démarche de segmentation d'images par MMTriplet . . . . .	39
6	La démarche de segmentation d'images par MMF . . . . .	40
7	L'algorithme EM . . . . .	46
8	L'algorithme SEM . . . . .	47
9	L'algorithme ECI . . . . .	48
10	L'algorithme EM pour CMCouple . . . . .	71
11	L'algorithme SEM pour CMCouple . . . . .	72
12	L'algorithme ECI pour CMCouple . . . . .	73
13	L'algorithme EM pour CMC . . . . .	81
14	L'algorithme EM pour CMCouple-BI . . . . .	82
15	L'algorithme EM pour CMT . . . . .	89
16	Initialisation de $V^0(i, j)$ . . . . .	90
17	Initialisation de $PI^0$ . . . . .	91
18	Initialisation de $A^0(i, j)$ . . . . .	91
19	L'algorithme EM pour CMCF . . . . .	101
20	L'algorithme SEM pour CMCF . . . . .	102
21	L'algorithme ECI pour CMCF . . . . .	103
22	Initialisation de $PI^0$ . . . . .	104
23	Initialisation de $A^0$ . . . . .	104
24	Les étapes de l'approche estimation indépendante . . . . .	123
25	Démarche de l'algorithme d'estimation indépendante . . . . .	124
26	Les étapes de l'approche : estimation dépendante . . . . .	126
27	Démarche de l'algorithme d'estimation dépendante . . . . .	127
28	Les étapes de l'approche d'estimation dépendante adaptée au modèle CMTH . . . . .	142



29	L'approche d'estimation dépendante adaptée au modèle CMTH . . . . .	143
30	Les étapes de l'approche adaptée au modèle CMTP . . . . .	144
31	Les étapes de l'approche adaptée au modèle CMTHP . . . . .	147
32	L'algorithme de seuillage(binarisation) . . . . .	149

# Liste des abréviations

MMC	Modèle de Markov Caché
CHMC	Champs de Markov Caché
CMC	Chaîne de Markov Cachée
AMC	Arbre de Markov Caché
MMCouple	Modèle de Markov Couple
MMTriplet	Modèle de Markov Triplet
MMF	Modèle de Markov Floue
EM	Espérance-Maximisation
GEM	Général Espérance-Maximisation
MCEM	Monté Carlo Espérance-Maximisation
SEM	Stochastique Espérance-Maximisation
ECI	Estimation Conditionnelle Itérative
MAP	Maximum <i>A Posteriori</i>
ICM	Mode Conditionnel Itéré
MPM	Mode <i>a posteriori</i> des Marginales
CMC-BI	Chaîne de Markov Cachée à Bruit Indépendant
IRM	Imagerie par Résonance Magnétique
RVB	Rouge Vert Bleu
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Measure
CMCouple	Chaîne de Markov Couple
CMCouple-BI	Chaîne de Markov Couple à Bruit Indépendant
CMT	Chaîne de Markov Triplet
CMMNS-BI	Chaîne de Markov M-Non Stationnaire à Bruit Indépendant
CMCF	Chaîne de Markov Cachée Floue
CMCH	Chaîne de Markov Cachée Hiérarchique
CMTH	Chaîne de Markov Triplet Hiérarchique
CMTP	Chaîne de Markov Triplet Parallèle
CMTHP	Chaîne de Markov Triplet Hiérarchique Parallèle
ROI	Région d'intérêt
MM	Morphologie Mathématique

# Introduction Générale

## Cadre général

L'évolution rapide de la technologie de l'information numérique et multimédia a mis en évidence l'image comme un outil indispensable pour réussir la transmission des messages non verbaux dans le but d'assurer une communication indirecte ou apprentissage à travers son contenu. Elle transfère des messages et des idées, l'image occupe une place de plus en plus large dans plusieurs domaines de la vie quotidienne : la médecine, publicité, industrie, etc..

Le traitement d'images [1] est l'ensemble de techniques permettant de modifier une image numérique, afin d'améliorer sa qualité ou d'en extraire des informations. Il est basé sur quelques transformations mathématiques telles que la technique de filtrage [2], la classification des images [3], la reconnaissance [4, 5], la segmentation et la restauration [6]. La segmentation d'images [7] figure parmi les problèmes les plus importants et les plus compliqués dans un système de traitement d'images. Parfois, elle peut être confondue avec la classification, cette opération permet d'extraire les objets et les composants d'images afin d'aider l'œil humain à analyser les différentes régions de l'image et d'extraire les informations utiles de cette image. Elle consiste à distinguer les pixels de l'image, qui ont des caractéristiques similaires comme la couleur, la texture et l'intensité en régions....

De nombreuses techniques de segmentation existent dans la littérature [1, 7, 8]. Parmi, les plus simples, la technique de seuillage et l'histogramme [9]; le seuillage est basé sur l'indexation des régions similaires de l'image par un seuil, par exemple la binarisation des images qui consiste à définir deux seuils : l'un pour les pixels de couleur blanche et l'autre pour les pixels de couleur noire. L'approche par région [10] et l'approche par contour [11] sont deux approches de segmentation bien connues, la première consiste à diviser l'image en régions homogènes en fonction de la similarité entre les composantes des régions, par contre, la deuxième approche divise l'image en régions en fonction de la discontinuité entre les composantes des régions. En outre, il existe de nombreuses méthodes fondées sur ces deux approches, nous citons la méthode de croissance des régions (Region-Growing) [10] et la méthode division et fusion (Split and Merge) [12] qui sont deux variantes de l'approche par région. L'approche par contour admet des variantes comme le contour actif [13], le contour de Sobel et le contour de Canny [14]. Certaines techniques récentes combinent l'approche

par région et l'approche par contour comme la Morphologie Mathématique(MM) [15, 16, 17] et la Ligne Partage d'Eau (LPE) [18], notons que ces techniques donnent de meilleurs résultats en comparaison avec l'approche par région et l'approche par contour.

L'algorithme de K-moyenne(K-means) est également une autre méthode de segmentation [19], basée sur la technique de partitionnement des données (Clustering) [20]. Il consiste à segmenter l'image en régions en classant les pixels de l'image dans une classe d'appartenance appropriée répondant à certains critères, tels que la distance entre les pixels, les degrés d'appartenance et le nombre de pixels voisins.... Notons que le choix d'une méthode de segmentation dépend de nombreux critères de sélection, par exemple la nature de l'image, les informations dominantes, les caractéristiques des objets à extraire...

Récemment, plusieurs recherches ont été menées pour segmenter les images d'une manière sémantique et rapide. Parmi celles-ci, nous trouvons les coupes de graphes et les réseaux de neurones convolutifs, ces méthodes ont été utilisées pour segmenter les images : couleurs, niveau de gris, 2D, 3D, médicales et satellites de diverses bases d'images. Dans [21], les auteurs ont décrit une nouvelle infrastructure logicielle(framework) pour une extraction des objets d'une manière efficace à partir des images de  $N$  dimensions  $ND$  en utilisant le coupe de graphe  $s/t$ . Les mêmes auteurs ont introduit dans [22] une nouvelle implémentation de l'algorithme Flot Maximal/Coupe Minimale(Max-Flow/Min-Cut), utilisée par la coupe de graphe pour segmenter et restaurer les images. En plus, Sandler et al [23] ont proposé une nouvelle version de l'architecture mobile MobileNet V2, utilisée dans la phase de convolution par les réseaux convolutifs, en se basant sur les résidus inversés et les goulots d'étranglement linéaires pour segmenter et détecter les objets des images. Ensuite, Kniaz [24] a proposé une nouvelle version du réseau convolutif Generative Adversarial Neural Network(GAN), appelée Conditionnel Generative Adversarial Neural Network(CGAN) pour segmenter un ensemble d'images satellites multispectrales. Également, dans un autre travail, Kniaz [25] a proposé un nouveau réseau de Generative Adversarial Neural Network, appelé Hydro Generative Adversarial Neural Network (HydroGAN), ce modèle a été utilisé pour étiqueter les régions hydrographiques dans les images satellites, il est capable d'étiqueter les objets hydrauliques dans les différentes saisons de l'année. Une nouvelle architecture d'encodage et décodage a été développée pour la segmentation d'images appelée SegNet [26]. Dans [27], les auteurs ont comparé cette nouvelle contribution avec d'autres architectures de réseaux convolutifs existantes.

Par la suite, les auteurs [28] se sont concentrés sur les tâches de perception du réseau convolutif en améliorant un système multitâches, appelé Unified Perceptual Parsing Net(UPENet), basé sur l'analyse perceptive unifiée, ils ont évalué cette nouvelle framework en segmentation d'images. Plus récemment, [29] ont étudié la représentation de la résolution des réseaux convolutifs à haute résolution en introduisant une simple modification : par l'augmentation de la représentation à haute résolution en agrégeant, la forme des représentations de toutes les convolutions parallèles.

Dans ce travail, nous nous intéressons à la segmentation d'images par les Modèles de Markov Ca-

chés(MMC) [30]. Ces modèles sont des méthodes de segmentation statistiques non supervisées [31], basés sur la technique de clustering. Ils sont utilisés dans de nombreuses applications telles que l'imagerie médicale [32, 33], l'imagerie sonar [34], l'imagerie industrielle et l'imagerie satellitaire ou aérienne [35]. En particulier, en imagerie satellitaire ou aérienne de telles méthodes ont été utilisées à diverses fins comme, à titre d'exemple, la classification des zones ou l'extraction des structures, l'interprétation des scènes. Les MMC sont des modèles robustes, capables de segmenter les images avec de bons résultats, ils sont très attractifs, ils fournissent un cadre probabiliste très général pour l'analyse d'images ce qui valide notre choix à utiliser ce modèle pour segmenter les images.

Nous distinguons trois types de MMC classiques : les champs, les chaînes et les arbres. Chaque modèle a sa propre procédure de modéliser l'image avant de la segmenter. Les champs de Markov cachés [36] ont pour principe de décomposer l'image en blocs de pixels voisins, chaque bloc contient au moins quatre ou huit pixels voisins. Chaque bloc représente un ensemble de cliques d'ordre différente. Ce modèle markovien donne de bons résultats de segmentation, mais il nécessite un temps et une unité d'exécution importants par rapport aux deux autres modèles markoviens [36, 37]. Les chaînes de Markov cachées transforment l'image  $2D$  en  $1D$  en respectant que deux pixels voisins dans l'image  $2D$  doivent rester voisins dans la chaîne  $1D$ . Il existe de nombreux parcours pour transformer l'image  $2D$  en chaîne  $1D$ , comme le parcours fractal de Hilbert Peano [38], le parcours Zigzaging, le parcours ligne par ligne, le parcours colonne par colonne, .... Les arbres de Markov cachés sont des cas particuliers des chaînes de Markov, nous devons utiliser la représentation quadarbre [39], ou d'autres représentations d'arbres pour convertir une image en forme d'arbre.

En outre, nous trouverons les MMC récents, qui représentent une généralisation aux modèles classiques, tels que les modèles de Markov couples [40, 41], les modèles triplets [36] et les modèles évidentiels [42]. Ces modèles sont utilisés pour résoudre les problèmes de modélisation que les modèles classiques sont incapables de les résoudre. Une autre représentation récente de ces modèles est la version floue, elle est utilisée pour segmenter les images floues [43, 44]. Récemment, plusieurs travaux ont utilisé les MMC dans des applications de segmentation et de restauration des signaux et d'images [45, 46, 47].

## Contributions

À partir des études effectuées sur l'utilisation des MMC en segmentation d'images [48, 49], nous constatons que leur problème principal est la complexité de calcul. Évidemment, les dimensions d'une image segmentée sont très grandes, ainsi que le temps de calcul des paramètres et les dimensions et le nombre des processus à estimer. De ce fait, les méthodes classiques sont impraticables et que même la réduction de la taille de données à traiter peut provoquer une perte totale de l'information. Pour résoudre ce problème des grandes dimensions, nous proposons quelques techniques fondées sur les techniques "diviser pour régner" et "parallélisme".

Tout d'abord, nous présentons deux approches basées sur la technique "diviser pour régner", la

première à estimation indépendante, il s'agit d'une méthode traditionnelle de division qui divise l'image à traiter en un ensemble des sous-blocs de même taille, chaque bloc est segmenté indépendamment des autres à l'aide du modèle chaîne de Markov cachée à bruit indépendant. Chaque bloc composant l'image est traité indépendamment d'une manière séquentielle comme une sous-image. La deuxième approche à estimation dépendante, divise l'image en un ensemble de blocs de même taille, le traitement de chaque bloc dépend du résultat de bloc précédent. Afin de réduire davantage le temps d'exécution de la segmentation, nous proposons deux extensions de la deuxième approche, chaque extension a ses propres principes et procédures. Nous avons utilisé ces approches pour segmenter des images niveau de gris et couleur, de grande taille [50].

Ensuite, l'approche à estimation dépendante a été réutilisée pour accélérer le temps d'exécution du modèle récent chaîne de Markov triplet, nous proposons trois méthodes différentes. La première appelée chaîne de Markov triplet hiérarchique, pour estimer de façon hiérarchique les paramètres du modèle triplet. Dans la deuxième méthode dite chaîne de Markov triplet parallèle, nous gardons le même processus d'estimation que la chaîne de Markov triplet. Les processus cachés  $X$  et  $U$  de ce modèle sont estimés, en parallèle, en utilisant soit les multiprocessoirs ou les threads. La troisième méthode est basée sur les deux dernières approches, elle utilise l'approche à estimation dépendante pour estimer les paramètres du modèle triplet, et dans la phase finale de décision, elle utilise les techniques de parallélisme pour estimer simultanément les processus cachés  $X$  et  $U$ . Ces trois approches ont été appliquées pour segmenter les images cérébrales IRM.

Finalement, pour extraire la région d'intérêt(tumeur) avec précision, nous proposons une nouvelle approche qui applique la technique de seuillage suivie des opérations de la morphologie mathématique. Nous réalisons aussi une étude comparative entre les opérations de la morphologie mathématique telles que l'érosion, la dilatation, l'ouverture et la fermeture, afin de trouver l'opération qui détecte avec précision la forme de la tumeur.

## Plan du mémoire

Ce manuscrit est organisé comme suit :

Le premier chapitre décrit les différents modèles de Markov cachés utilisés en segmentation d'images, à savoir, les modèles classiques : champs, chaînes et arbre ; puis les modèles de Markov récents : modèles de Markov couples, modèles de Markov triplets et modèles de Markov flous.

Le deuxième chapitre présente un ensemble des comparaisons entre les estimateurs itératifs, entre les estimateurs Bayésiens, entre les chaînes de Markov cachées stationnaires, entre les chaînes de Markov cachées stationnaires et non-stationnaires, et finalement nous présentons le modèle de Markov flou : chaîne de Markov cachée floue.

Dans le troisième chapitre, nous exposons nos contributions pour les modèles de Markov classiques. Ce chapitre se divise en deux sections, la première présente l'approche à estimation indépendante. Puis, la deuxième section expose l'approche à estimation dépendante et ses variantes.

Le dernier chapitre présente nos contributions pour accélérer le temps d'exécution du modèle de Markov triplet, ainsi que l'approche proposée pour extraire la région d'intérêt avec précision à partir des images cérébrales IRM segmentées.

Finalement, nous clôturons ce mémoire par une conclusion et des perspectives.

# Chapitre 1

## Généralités sur la segmentation markovienne des images

### 1.1 Introduction

Un modèle de Markov caché est un modèle statistique composé d'états cachés, d'observations et de transitions. Chaque état émet des observations. Nous travaillons donc sur la séquence d'observations générées par les états, et pas sur la séquence d'états. Une transition matérialise la possibilité de passer d'un état à un autre. Un MMC est donc un processus stochastique qui se compose de deux processus, un observé  $Y$  et l'autre caché  $X$ .

La segmentation d'images [51] est une opération de traitement d'images qui regroupe plusieurs approches mathématiques qui servent à étiqueter les pixels ou les voxels d'une image. Les pixels ou voxels ayant la même étiquette représentent une région homogène, l'ensemble des régions homogènes représente l'image segmentée. L'image résultat facilite à l'être humain la connaissance préalable de sa totalité, ce qui facilite l'analyse et la prise de décision. Plusieurs approches mathématiques ont été développées en segmentation d'images. Parmi celles-ci la segmentation markovienne.

La segmentation markovienne d'images [52], est une segmentation basée sur l'approche de segmentation par région qui sert à segmenter l'image selon la similarité de ces régions composantes. Il s'agit d'une segmentation statistique non supervisée basée sur la classification des pixels dans une classe précise. Elle estime l'image résultat de segmentation, notée  $X$ , qui représente la partie cachée du processus markovien à partir de l'image d'entrée, notée  $Y$ , qui représente l'ensemble des observations du processus markovien. Elle cherche la classe d'appartenance  $x_n$  de chaque pixel  $y_n$  qui compose l'image..

La partie cachée  $X = (X_n)_{n=1}^N$  est définie dans l'ensemble de classes d'appartenance  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  avec  $K$  est le nombre de classes d'appartenance, il est initialisé par l'utilisateur. La partie observée  $Y = (Y_n)_{n=1}^N$  est définie dans l'ensemble  $\mathbb{R}$ , avec  $N$  est la taille totale de l'image originale. Les MMC estiment le processus caché  $X$  sachant le processus observé  $Y$  en maximisant la probabilité *a posteriori*  $P(X|Y)$  de  $X$  sachant  $Y$  en utilisant la règle de Bayes [53]. Dans le cas de segmentation



d'images, les MMC considèrent que  $Y$  et  $X$  sont discrets.

$$P(X|Y) = \mathbf{max} \frac{P(Y|X).P(X)}{P(Y)} \quad (1.1)$$

Où :

- $P(Y|X)$  : est la probabilité *a posteriori* de  $Y$  sachant  $X$ .
- $P(X)$  : est la probabilité *a priori* du processus  $X$ .
- $P(Y)$  : est une constante de normalisation  $P(Y) = 1$ .

Pour trouver la classe d'appartenance  $c$  d'un pixel  $y_n$ , il faut calculer :

$$c = \mathbf{argmax}_x (P(Y|X).P(X)) \quad (1.2)$$

Le principe de la segmentation markovienne est de créer une partition  $X$  en sous-ensembles des régions qui possèdent des caractéristiques d'homogénéité à partir de l'image observée  $X$ .

Dans ce chapitre, nous exposons les différents modèles de Markov utilisés en segmentation d'images et restauration de signaux. Nous commençons par les modèles de Markov classiques ; à savoir : champ, chaîne et arbre, puis nous décrivons les modèles récents : couples et triplets. Aussi, nous présentons le modèle de Markov flou et nous terminerons par une conclusion.

## 1.2 Les modèles de Markov cachés

Les modèles de Markov cachés en segmentation d'images sont divisés en deux catégories principales : classique et récente. Dans les modèles classiques le processus caché  $X$  est markovien, homogène et stationnaire, et les observations  $Y$  sont conditionnellement indépendantes du processus caché  $X$ . Pour cette raison, ces modèles sont appelés par les modèles de Markov cachés à bruit indépendant. Les modèles récents sont utilisés pour résoudre les problèmes de modélisation que les modèles classiques ne peuvent pas les résoudre. Par exemple, les modèles couples sont utilisés pour modéliser le bruit ou la texture, ils supposent que le couple  $X$  caché et  $Y$  observé est markovien et stationnaire. Les modèles triplets servent à modéliser les problèmes de la non-stationnarité du processus caché  $X$  ou à modéliser le temps de séjour du processus markovien  $X$ , en introduisant un troisième processus  $U$  qui est auxiliaire, il est utilisé soit pour représenter le changement des transitions du processus caché  $X$  ou pour représenter le temps de séjour du processus markovien  $X$ . Le choix d'un modèle de Markov caché pour segmenter une image dépend de la nature des données à traiter : stationnaires, non-stationnaires, bruitées, texturées.... Les différents modèles de Markov utilisés en segmentation d'images sont résumés dans cette figure :

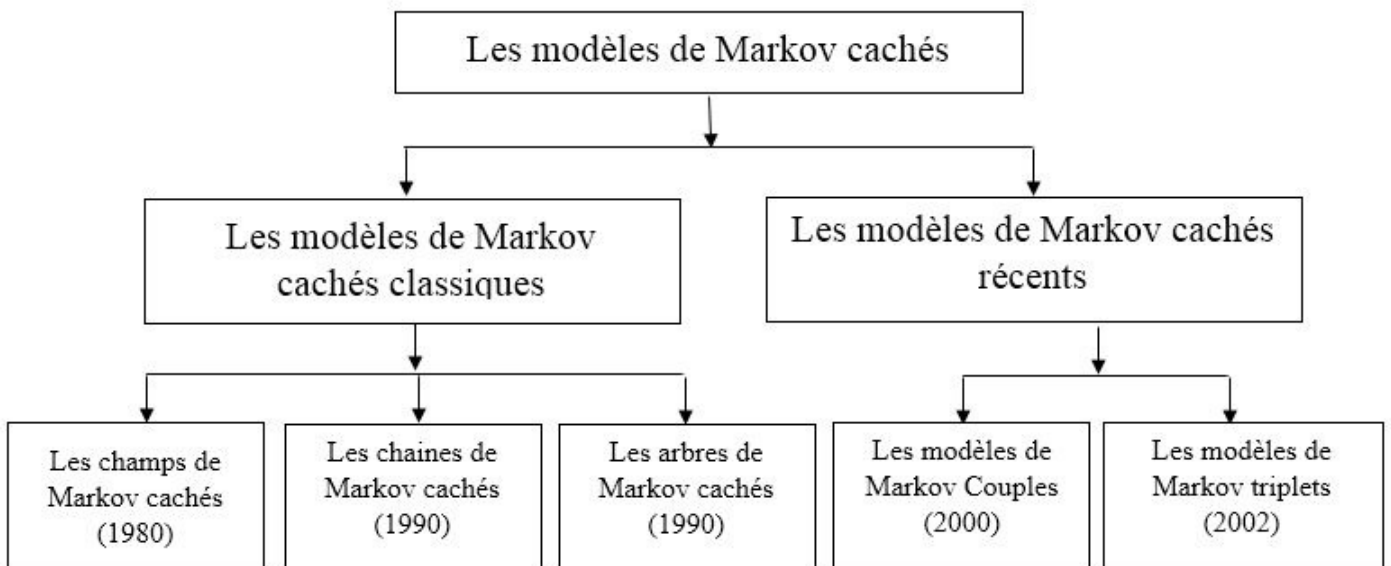


FIGURE 1.1 – Les modèles de Markov cachés utilisés en segmentation des images

### 1.3 Les phases d'estimation des paramètres

L'estimation d'une image résultat de segmentation par les modèles de Markov cachés requiert trois phases d'estimation de paramètres :

1. La phase d'initialisation des paramètres.
2. La phase d'estimation itérative des paramètres.
3. La phase de décision finale.

Dans la phase d'initialisation, on initialise les paramètres des processus  $X$  et  $Y$ , ainsi que, le processus caché  $X^0$  à l'aide d'un algorithme de segmentation qui utilise la technique de clustering. Dans la deuxième phase on estime les paramètres des processus  $X$  et  $Y$  itérativement en utilisant les estimateurs itératifs à savoir, EM(Espérance-Maximisation) en anglais(Expectation–Maximization), SEM(Stochastique Espérance-Maximisation) en anglais(Stochastic Expectation–Maximization), ECI(Estimation Conditionnelle Itérative) en anglais(Iterative conditional Estimation).....Après la convergence de ces algorithmes, les MMC utilisent dans la phase de décision finale les stratégies Bayésiennes telles que l' algorithme, MAP(Maximum *A Posteriori*) en anglais(Maximum *A Posteriori*) et l' algorithme MPM(Mode *a posteriori* des Marginales) en anglais(Marginal Posteriori Mode) pour estimer la configuration finale du processus caché  $X$ . Les stratégies Bayésiennes servent à estimer la configuration finale de  $X$  en maximisant la probabilité *a posteriori*.

Le schéma suivant montre la démarche de segmentation d'images suivie par les MMCs :

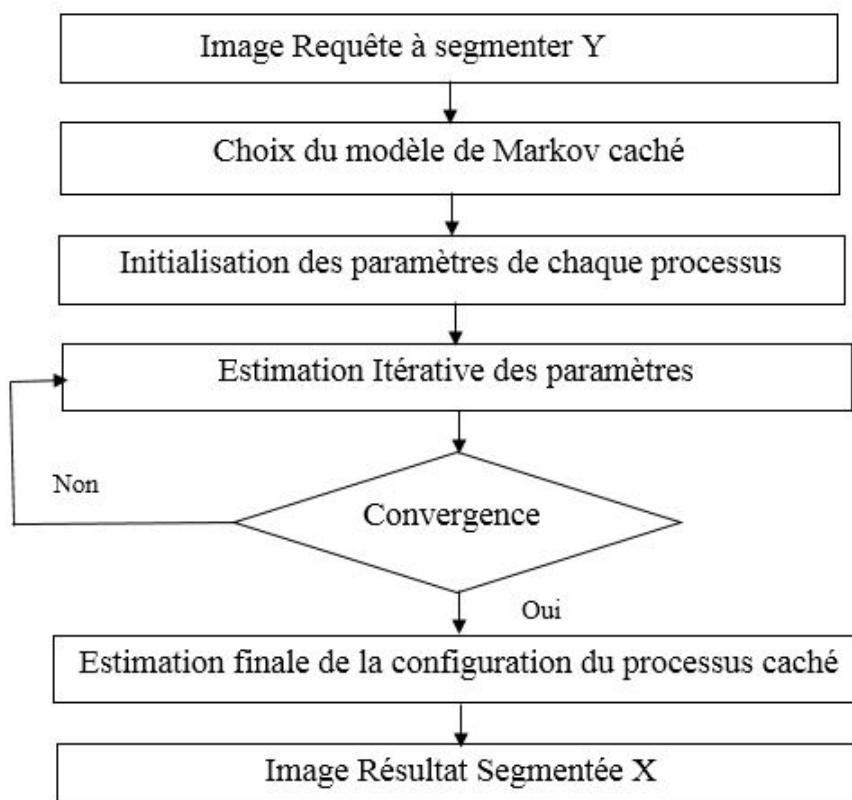


FIGURE 1.2 – La démarche de segmentation des images par les MMCs

## 1.4 Les champs de Markov cachés

Le modèle Champs de Markov Caché (CHMC) [54] est un outil de base de traitement statistique des images, utilisé depuis les années quatre-vingt [55]. Grâce aux progrès des techniques calculateurs qui ont rendu le temps de calcul concevable, l'utilisation de ce modèle est devenue possible. Le CHMC assure la corrélation spatiale entre les pixels de l'image à segmenter [56, p. 84]. Plusieurs travaux dans la littérature ont utilisés ce modèle pour segmenter les images [57, 58, 59].

Une image est considérée comme un ensemble de réalisations d'un champs aléatoire  $Y$  de taille  $N$  qui prennent ses valeurs dans l'ensemble  $\mathbb{R}$ , associé à un champs aléatoire  $X$  de taille  $N$  défini dans  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ . Chaque élément observé  $y_{n=1:N}$  est lié à un élément caché  $x_{n=1:N}$ . Dans un CHMC chaque pixel  $y_n$  est associé à un ensemble de pixel voisin, la relation entre un pixel et ses voisins s'appelle un système de voisinage  $V_s$  [56, p. 84].

### 1.4.1 Système de voisinage

Une image est représentée par une grille bidimensionnelle  $S$  de taille  $n * m$  ou  $S = \{S(i, j), 1 \leq i \leq n, 1 \leq j \leq m\}$  contenant un ensemble de site  $s \in S$ , appelé aussi pixel [56, p. 85].

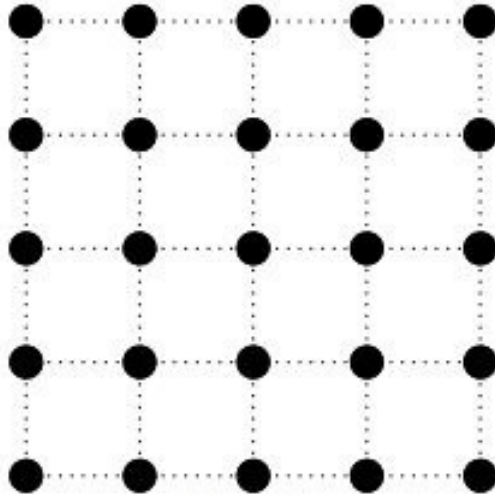


FIGURE 1.3 – La grille S

Tout ensemble  $V = (V_s) \in S$  vérifiant les propriétés suivantes s'appelle système de voisinage :

- $s \notin V_s$  : un site n'est pas voisin à lui-même ;
- $s \in V_t \Leftrightarrow t \in V_s$  représente la relation de voisinage entre  $t$  et  $s$ , elle est symétrique.

L'ensemble des voisins d'un pixel  $s$  est défini par un ensemble de sites voisins inclus dans un rayon  $r$  :

$$V_s = \{t \in S \mid d(s, t) \leq r, t \neq s\}$$

où  $d(s, t)$  est la distance euclidienne entre le site  $s$  et le site  $t$ . Un système de voisinage [60, p. 17]  $V_s$  est défini par un ensemble de clique  $c$ , la clique comporte un nombre de sites  $s$  voisins entre eux. Il existe plusieurs systèmes de voisinage parmi ces systèmes : le système de voisinage de quatre connexités appelé aussi voisinage du premier ordre, et de huit connexités appelé aussi par voisinage du second ordre. La figure 1.4 illustre la notion des cliques dans le cas, de voisinage en quatre connexités et en huit connexités.

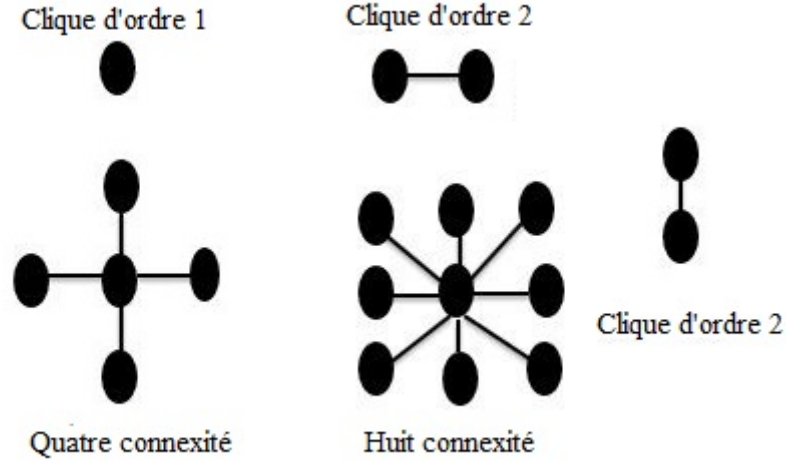


FIGURE 1.4 – Les cliques associées à un système de voisinage quatre connexités et huit connexités

## 1.4.2 Champs de Markov caché

**Définition 1.4.1.** Soit  $X = (X_s)_{s=1}^S$  un processus aléatoire.

On dit que  $X$  est un champ de Markov relativement à un système de voisinage  $V$  si et seulement si la probabilité  $p(x_s|x_t)$  d'un site  $s \in V_s$  ne dépend que de ses voisins  $t \in V_s$

$$p(x_s|x_t, t \neq s) = p(x_s|x_t) \quad (1.3)$$

**Définition 1.4.2.**  $X = (X_s)_{s=1}^S$  est un champ de Gibbs relativement à un système de voisinage  $V$  si et seulement si, sa loi vérifiée :

$$p(x) = \frac{1}{Z} \exp [-U(x)] \quad (1.4)$$

Avec :

$U(x)$  : est l'énergie potentielle d'une clique  $c$ .

$Z$  : est une constante de normalisation.

$p(x)$  : est appelé mesure de Gibbs.

L'énergie potentielle  $U(x)$  du clique  $c$   $\phi_c(x_c)$  est calculée par la sommation des énergies de chaque site en utilisant le symbole de Kronecker  $\phi$  :

$$\phi(x_s, x_t) = \begin{cases} 1 & \text{si } x_s \neq x_t \\ 0 & \text{sinon} \end{cases} \quad (1.5)$$

Donc :

$$U(x) = \sum_{c \in C} \phi_c(x_c) \quad (1.6)$$

$C$  : est l'ensemble des cliques  $c$  définie par un système de voisinage  $V$ .

La constante de normalisation  $Z$  représente la somme de toutes configuration d'une image.

Supposons que nous avons une image binaire  $Y$  dont ses pixels prennent soit la valeur 0 ou 1 de taille  $128 * 128$ . Le nombre de configuration possible est  $2^{128*128} = 2^{16384}$ . Pour simplifier, nous pouvons définir la constante  $Z$  par la somme totale des énergies potentielles des cliques de l'image.

Donc :

$$p(x) = \frac{1}{Z} \exp \left[ - \sum_{c \in C} \phi_c(x_c) \right] \quad (1.7)$$

Un champs de Gibbs est défini par sa loi globale qui s'appelle mesure de Gibbs.

Le théorème de Hammersley-Clifford [61] donne les conditions d'équivalence entre un champ de Markov et un champs de Gibbs.

**Théorème 1.4.1.** *Théorème de Hammersley-Clifford* Un champ de Markov  $X = (X_n)_{n=1}^N$  est un champ de Gibbs  $X$  si et seulement si :

$$p(x) > 0 \quad (1.8)$$

$$p(x) = \frac{1}{Z} \exp [-U(x)] > 0 \quad (1.9)$$

$$p(x) = \frac{1}{Z} \exp \left[ - \sum_{c \in C} \phi_c(x_c) \right] > 0 \quad (1.10)$$

Le théorème nous permet d'accéder à la loi conditionnelle locale  $p(x_s|x_t, t \in V_s)$  à partir de système de voisinage et des énergies potentielles. La loi du probabilité conditionnelle locale  $p(x_s|x_t, t \in V_s)$  s'écrit comme suit :

$$p(x_s|x_t, t \in V_s) = \frac{\exp [-\phi_c(x_s - x_{c-s})]}{\int_{x_s \in S} \exp [-\phi_c(x_s - x_{c-s})]} = \Phi_c(x_s) \quad (1.11)$$

La densité globale se calcule de la manière suivante :

$$p(x) = \prod_{c \in C} \Phi_c(x_s) \quad (1.12)$$

Donc, Il est possible de simuler les réalisations d'un champ de Markov itérativement en calculant la loi conditionnelle locale avec les méthodes de simulation à savoir, l'échantillonneur de Gibbs [55] ou l'algorithme de Métro-polis [62].

Un champ de Markov caché admet deux lois de probabilité : la loi *a priori* du champs caché  $X$  et la loi des observations  $Y$ . Chaque loi admet des paramètres.

la loi du processus caché  $X$  est définie par 1.4, la loi de  $Y$  conditionnellement à  $X$  est strictement positive, elle est définie par :

$$p(y|x) = \prod_{s \in S} p(y_s|x_s) = \exp \left[ \sum_{s \in S} \log(p(y_s|x_s)) \right] \quad (1.13)$$

Nous remplaçons la probabilité  $p(y_s|x_s)$  par la densité  $f_{x_s}(y_s)$  l'équation 1.13 devient :

$$p(y|x) = \prod_{s \in S} f_{x_s}(y_s) = \exp \left[ \sum_{s \in S} \log(f_{x_s}(y_s)) \right] \quad (1.14)$$

Donc la loi de  $(X, Y)$  est comme suit :

$$p(x, y) = p(x)p(y|x) = \frac{1}{Z} \exp[-U(x, y)] \quad (1.15)$$

$$= \exp \left[ - \sum_{c \in C} \phi_c(x_c) + \sum_{s \in S} \log(f_{x_s}(y_s)) \right] \quad (1.16)$$

Pour estimer la classe d'appartenance, il faut calculer la probabilité *a posteriori* de  $X$  sachant  $Y$  en minimisant les énergies potentielles de l'équation 1.16.

Nous résumons la démarche de segmentation d'images en utilisant le modèle CHMC dans l'algorithme suivant :

---

**Algorithme 1** La démarche de segmentation d'images par CHMC

---

- 1: Transformation de l'image d'entrée  $2D$  en un ensemble de cliques  $1D$  ;
  - 2: Initialisation du processus  $X^0$  et des paramètres de chaque processus ;
  - 3: Calcul des énergies potentielles initiales  $U(x)$  de chaque clique ;
  - 4: Calcul de l'énergie potentielle initiale totale  $\sum U(x)$  ;
  - 5: Simulation itérative de la configuration du processus caché  $X$  en utilisant l'algorithme échantillonneur de Gibbs ou l'algorithme de Métro-polis ;
  - 6: Estimation itérative des paramètres soit par l'utilisation des algorithmes à données complètes comme la méthode de décodage, Gradient stochastique [63], pseudo vraisemblance où par les algorithmes à données incomplètes comme EM, SEM, ECI adapté à ce modèle ;
  - 7: Estimation de la configuration finale du champ caché  $X$  en utilisant les estimateurs Bayésiens MAP, ICM ou MPM ;
  - 8: Transformation de l'image en  $2D$  ;
- 

**Remarque 1.** - *La différence entre l'estimation à données complètes et incomplètes est que : dans l'estimation à données complètes les deux processus caché et observé sont disponibles, et dans celle à données incomplètes, nous ne disposons que du processus observé [64, p. 27-28].*

- *Le temps d'estimation des paramètres d'un champ de Markov est coûteux, pour résoudre ce problème, nous pouvons négliger un certain nombre de voisinages par la modélisation de processus  $X$  par un voisinage de deux pixels (le cas d'une chaîne de Markov cachée) ou bien modéliser le processus  $X$  par la structure hiérarchique (le cas d'un arbre de Markov caché) [60, p. 18].*

## 1.5 Les chaînes de Markov cachées

Les Chaînes de Markov Cachées(CMC) sont parmi les modèles de Markov les plus simples, plus utilisés en segmentation des données. Les CMCs ont été utilisées dans plusieurs domaines

d'application de traitement du signal et de l'image : reconnaissance de la parole [65], segmentation d'images [66,67]. D'autres domaines d'application de ces modèles incluent la biologie [68], l'écologie [69] et la finance [70]. Le modèle CMC a été introduite en segmentation dans les années quatre-vingt-dix par [71], il transforme facilement l'image d'entrée à deux dimension  $2D$  à un vecteur à une dimension  $1D$  en parcourant l'image ligne par ligne, colonne par colonne. Nous pouvons, également utiliser le parcours fractal de Hilbert-Peano [72] présenté dans la figure suivante <sup>1</sup> :

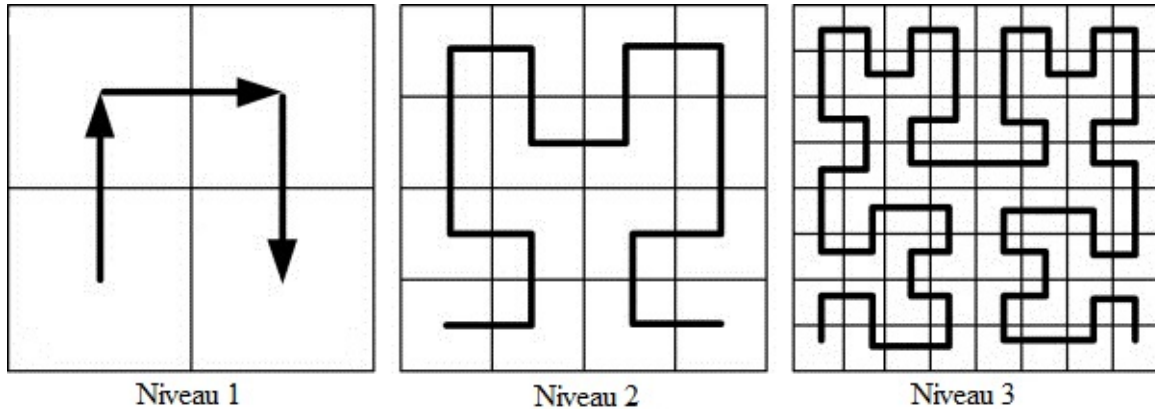


FIGURE 1.5 – Construction du parcours fractal de Hilbert-Peano.

### 1.5.1 Chaîne de Markov

**Définition 1.5.1.** Soit  $X = (X_n)_{n=1}^N$  un processus aléatoire de taille  $N$  à valeurs dans un ensemble fini  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ . On dit que  $X$  est une chaîne de Markov si et seulement si chaque état future  $x_{n+1}$  ne dépend que de l'état du présent  $x_n$ .

$$p(x_{n+1}|x_n, \dots, x_1) = p(x_{n+1}|x_n) \quad (1.17)$$

En mots, l'état courant résume à lui seul tout l'historique du système susceptible d'influencer son évolution future.

Une chaîne de Markov est dite homogène, si les probabilités  $p(x_{n+1}|x_n)$  ne dépendent pas du temps. Une chaîne de Markov est dite stationnaire s'elle est homogène et sa loi ne dépend pas du temps [64, p. 27].

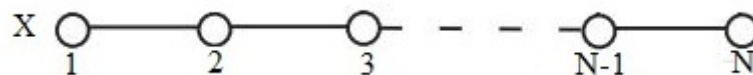


FIGURE 1.6 – Graphe de dépendance d'une chaîne de Markov

1. [https://www.researchgate.net/figure/Example-of-the-Peano-Hilberts-construction-curve-a-the-unit-interval-to-unit\\_fig7\\_257879078/download](https://www.researchgate.net/figure/Example-of-the-Peano-Hilberts-construction-curve-a-the-unit-interval-to-unit_fig7_257879078/download)



## 1.5.2 Chaîne de Markov cachée

**Définition 1.5.2.** Une CMC est un processus stochastique composé de deux processus [73]  $Y$  et  $X$ ,  $Y$  sera supposé à valeurs dans  $\mathbb{R}$ ,  $X$  une chaîne de Markov homogène et stationnaire à valeurs dans un ensemble fini  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  de taille  $N$ , sa loi est :

$$p(x) = p(x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \quad (1.18)$$

Le processus observable  $Y$  est vue comme, une version bruitée de processus caché  $X$ . La loi  $p(y|x)$  est appelée distribution de bruit.

Dans une CMC, la loi de  $Y$  est conditionnellement indépendante de  $X$ , elle est définie par :

$$p(y|x) = p(y_1|x_1) \cdot \prod_{n=1}^{N-1} p(y_{n+1}|x_{n+1}) \quad (1.19)$$

La loi de  $(X, Y)$  sera donc :

$$p(x, y) = p(x_1) \cdot p(y_1|x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot p(y_{n+1}|x_{n+1}) \quad (1.20)$$

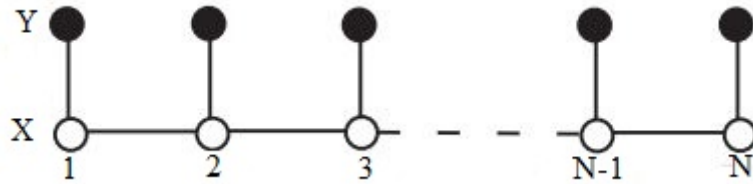


FIGURE 1.7 – Graphe de dépendance d'une chaîne de Markov cachée

En segmentation d'images, la CMC estime l'image résultat que l'on considère comme un processus caché  $X$ , à partir de l'image observée que l'on considère comme un ensemble des observations  $Y$ , en maximisant la probabilité *a posteriori* de  $p(x|y)$ . Une chaîne de Markov cachée admet deux loi de probabilité : la loi *a priori* du processus  $X$  et la loi des observations  $Y$ .

La démarche de segmentation d'images en utilisant le modèle CMC est présentée dans l'algorithme suivant :

---

**Algorithme 2** La démarche de segmentation d'images par CMC

---

- 1: Transformation de l'image d'entrée  $2D$  à une chaîne  $1D$  en utilisant les parcours : ligne par ligne, colonne par colonne, Hilbert peano ;
  - 2: Initialisation du processus caché  $X^0$  et des paramètres de chaque processus ;
  - 3: Estimation des paramètres itérativement en utilisant les algorithmes itératifs EM, SEM ou ECI adapté à ce modèle ;
  - 4: Estimation de la configuration finale de l'image résultat  $X$  en utilisant les estimateurs Bayésiens MAP ou MPM ;
  - 5: Transformation de l'image résultat en  $2D$  ;
- 

## 1.6 Les arbres de Markov cachés

Le modèle Arbre de Markov Caché(AMC) est introduit en segmentation des images par [74] dans les années quatre vingt dix. Il est utilisé dans les applications de segmentation des images multispectrales [75], des images astronomiques multirésolutions [60].

Une image est considérée comme un ensemble de partition d'un processus aléatoire  $Y$  de taille  $N$  à valeurs dans l'ensemble  $\mathbb{R}$ , associé à un processus aléatoire  $X$  partitionné par  $N$  à valeurs dans  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ .

L' AMC [?, p. 12]a12 partitionne les pixels de l' image  $Y$  en  $S^n$  partitions tel que :  $S^1 < S^2 < \dots < S^n$ , chaque partition  $S^i$  représente une génération.

La partition  $S^1$  admet un seul élément, elle représente la racine.

Chaque partition  $S^i$  forme une partition  $S^{i+1}$ .

On note :

- $s^1$  est la racine de l'arbre.
- $s^+ \subset S^i$  chaque partition contient un nombre de nœuds.
- $s^+$  est un enfant de  $s^-$ .
- $s^-$  est un père de  $s^+$ .
- Chaque  $s^-$  admet des enfants  $s^+$ .

L'image à segmenter par AMC peut être transformer un arbre à l'aide d'une représentation arbre comme la représentation quadarbre [60, p. 28-29]. Les quadarbres sont des représentation arbre qui considèrent que chaque pixel à part les feuilles possède quatre enfants.

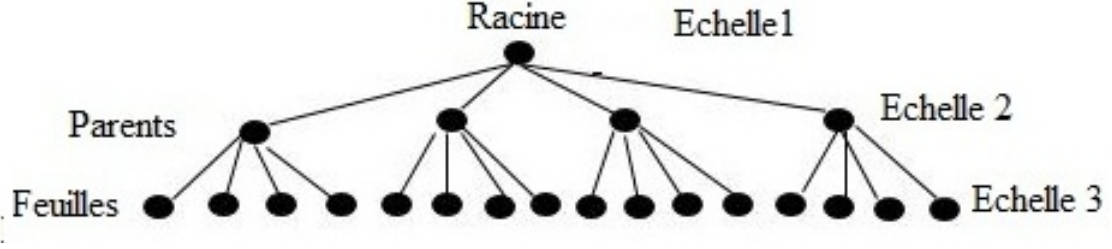


FIGURE 1.8 – Quadarbre

### 1.6.1 Arbre de Markov

**Définition 1.6.1.** Soit  $X = (X_n)_{n=1}^N$  un processus de taille  $N$ , indexé par un arbre à valeurs dans un ensemble fini  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ .  $X$  est un arbre de Markov, si et seulement si la probabilité d'un pixel enfant  $s^-$  ne dépend que du pixel père  $s^+$ , sa loi est :

$$p(x) = p(x_{s^1}) \cdot \prod_{n=1}^{N-1} \prod_{s \in S^i} p(x_{s^+} | x_{s^-}) \quad (1.21)$$

### 1.6.2 Arbre de Markov caché

**Définition 1.6.2.** Soit  $Z = (Z_n)_{n=1}^N$  un processus indexé par un arbre, défini à partir de deux processus [60, p. 22]  $Y = (Y_n)_{n=1}^N$  et  $X = (X_n)_{n=1}^N$  représentant respectivement, les observations et les étiquettes.

$Z$  est un AMC, si et seulement si :

- $X$  est un arbre de Markov, sa loi est :

$$p(x) = p(x_{s^1}) \cdot \prod_{n=1}^{N-1} \prod_{s \in S^i} p(x_{s^+} | x_{s^-}) \quad (1.22)$$

- Les observations  $Y$  sont conditionnellement indépendantes de  $X$ .
- Chaque observation dépend seulement de son état caché.

$$p(y) = \prod_{n=1}^N \prod_{s \in S^i} p(y_{s^+} | x_{s^+}) \quad (1.23)$$

Donc, la loi  $(X, Y)$  s'écrit par :

$$p(x, y) = p(x_{s^1}) \cdot p(y_{s^1} | x_{s^1}) \cdot \prod_{n=1}^{N-1} \prod_{s \in S^i} p(x_{s^+} | x_{s^-}) p(y_{s^+} | x_{s^+}) \quad (1.24)$$

La démarche de segmentation d'images en utilisant un AMC est présentée dans cet algorithme.

---

**Algorithme 3** La démarche de segmentation d'images par AMC

---

- 1: Transformation de l'image d'entrée  $2D$  à un arbre en utilisant la représentation arbre quadarbre ou d'autres représentations arbre ;
  - 2: Initialisation du processus  $X^0$  et des paramètres de chaque processus ;
  - 3: Estimation itérative des paramètres en utilisant les algorithmes itératifs EM, SEM ou ECI adapté à ce modèle ;
  - 4: Estimation de la configuration finale de l'image résultat  $X$  par les estimateurs Bayésiens MAP ou MPM ;
  - 5: Transformation de l'image en  $2D$  ;
- 

**Remarque 2.** - *La démarche de la segmentation par un AMC semble à la démarche de segmentation par une CMC. La différence entre les chaînes et les arbres est au niveau de la modélisation du processus  $X$ . Ou , les chaînes utilisent un voisinage de deux pixels, et les arbres modélisent le processus  $X$  par la structure hiérarchique [60, p. 19].*

## 1.7 Les modèles de Markov couples

Les Modèles de Markov Couples(MMCouples) sont introduits par [40] dans les années deux mille, ils généralisent les modèles classiques, ils servent à résoudre les problèmes de segmentation que les modèles classiques sont incapables de les résoudre comme, la segmentation des images en prenant en compte l'information bruit et les contours des objets de l'image [77]. Les MMCouples considèrent que les observations  $Y$  sont corrélées entre eux [78, p. 22-29] ce qui améliore les résultats de segmentation.

Les modèles de Markov classiques ont été étendus aux modèles couples tels que :

1. Les champs de Markov couples.
2. Les chaînes de Markov couples.
3. Les arbres de Markov couples.

**Définition 1.7.1.** *Le processus  $Z = (X, Y)$  est un processus de Markov couple si et seulement si :*

- *Le processus  $X$  conditionnellement à  $Y$  est un processus de Markov*
- *Le processus  $Y$  conditionnellement à  $X$  est un processus de Markov.*
- *Les observations  $y_n$  sont corrélées entre eux.*

*Sa loi est définie par :*

$$p(z) = p(z_1) \cdot \prod_{n=1}^{N-1} p(z_{n+1}|z_n) \quad (1.25)$$

*Les transitions  $p(z_{n+1}|z_n)$  peuvent s'écrire :*

$$p(z_{n+1}|z_n) = p(x_{n+1}|x_n, y_n)p(y_{n+1}|x_n, x_{n+1}, y_n) \quad (1.26)$$

Donc :

$$p(z) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n, y_n) p(y_{n+1}|x_n, x_{n+1}, y_n) \quad (1.27)$$

**Remarque 3.** Il est remarquable que, les modèles de Markov classiques définis en haut sont des cas particuliers aux modèles couples vérifiant [79, p. 15] :  $p(x_{n+1}|x_n, y_n) = p(x_{n+1}|x_n)$  et  $p(y_{n+1}|x_n, x_{n+1}, y_n) = p(y_{n+1}|y_n)$ .

La modélisation du processus  $X$ , ainsi que les paramètres des modèles couples dépendent des modèles utilisés, à savoir, les champs, les chaînes ou les arbres. La modélisation markovienne du processus  $X$  reste la même selon le modèle utilisé, mais les paramètres diffèrent d'un modèle à un autre. La complexité de calcul des modèles couples est plus élevée que celle des modèles classiques parce qu'ils calculent des paramètres de taille élevée. La démarche du modèle couple se déroule comme suit :

---

**Algorithme 4** La démarche de segmentation d'images par MMCouple

---

- 1: Transformation de l'image d'entrée  $2D$  selon le modèle utilisé champs de Markov couple, chaîne de Markov couple ou arbre de Markov couple ;
  - 2: Initialisation du processus  $X^0$  et des paramètres de chaque processus du modèle couple ;
  - 3: Estimation itérative des paramètres en utilisant les algorithmes itératifs EM, SEM ou ECI adapté au modèle de Markov couple utilisé ;
  - 4: Estimation de la configuration finale du processus caché  $X$  par les estimateurs Bayésiens MAP ou MPM adapté au modèle de Markov couple utilisé ;
  - 5: Transformation de l'image résultat en  $2D$  ;
- 

## 1.8 Les modèles de Markov triplets

Les modèles de Markov Triplets (MMTriplet) sont aussi introduits par [80] dans les années deux mille, ils généralisent les modèles classiques et couples.

On distingue trois représentations : les modèles markoviens non-stationnaires [80, 81], les modèles semi markoviens [81] et les modèles évidentiels [42]. Les modèles de Markov non-stationnaires sont utilisés pour modéliser la non-stationnarité du processus caché  $X$  en introduisant un processus auxiliaire  $U$  qui représente la non-stationnarité du processus caché. Le temps de séjour du processus  $X$  est également représenté par un processus auxiliaire  $U$ , dans ce cas, le modèle de Markov est un modèle semi-Markovien. En outre, le modèle de Markov évidentiel [81] est un modèle de Markov triplet très récent, où, les probabilités  $p$  sont remplacées par, des masses évidentielles  $m$  [81, 82] la fusion de Dempster-Shafer [83] est appliquée pour estimer la configuration finale de l'image résultat. Ce modèle modélise la non-stationnarité, l'ignorance et l'incertain en même temps.

La représentation de la non-stationnarité et le temps de séjour du processus caché  $X$  en même temps nécessite d'introduire deux processus auxiliaires  $U_1$  et  $U_2$  :  $U_1$  pour modéliser les stationnarités de

$X$  et  $U_2$  pour représenter le temps de séjour [84].

Les modèles de Markov classiques et couples ont été étendus aux modèles triplets [79, p. 19-21] :

1. Les champs de Markov triplets.
2. Les chaînes de Markov triplets.
3. Les arbres de Markov triplets.
4. Les champs de Markov couples triplets.
5. Les chaînes de Markov couples triplets.
6. Les arbres de Markov couples triplets.

**Définition 1.8.1.** Soit le processus caché  $X = (X_n)_{n=1}^N$ , prend ses valeurs dans l'ensemble fini  $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ ,  $Y = (Y_n)_{n=1}^N$  le processus observé, prend ses valeurs dans  $\mathbb{R}$  et le processus auxiliaire  $U = (U_n)_{n=1}^N$ , prend ses valeurs dans l'ensemble  $\Lambda = \{\lambda_1, \dots, \lambda_M\}$ . Avec  $M$  le nombre de stationnarités.

$T = (X, U, Y)$  est un modèle de Markov triplet, s'il est de Markov, sa loi est définie par :

$$p(t) = p(t_1) \cdot \prod_{n=1}^{N-1} p(t_{n+1}|t_n) \quad (1.28)$$

$$= p(x_1, u_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}, u_{n+1}, y_{n+1}|x_n, u_n, y_n) \quad (1.29)$$

$$= p(x_1, u_1, y_1) \cdot \prod_{n=1}^{N-1} p(u_{n+1}|u_n) p(x_{n+1}|x_n, u_{n+1}) \cdot p(y_{n+1}|x_{n+1}) \quad (1.30)$$

Dans le cas, où  $U$  gère le temps de séjour du processus caché  $X$ . Le processus  $U$  prend ses valeurs dans  $\mathbb{N}^*$ .

Les paramètres des modèles triplets dépendent de modèle utilisé champs, chaîne, arbre, couple. La modélisation markovienne reste la même, mais les paramètres diffèrent d'un modèle à un autre. La complexité de calcul de ces modèles est plus élevée que celles des classiques, et couples car, ils calculent des paramètres de taille élevée. Ainsi que, les MMT estiment deux processus cachés  $X$  et  $U$ . La démarche du modèle triplet se déroule comme suit :

---

**Algorithme 5** La démarche de segmentation d'images par MMTriplet

---

- 1: Transformation de l'image d'entrée  $2D$  selon le modèle utilisé Champs de Markov triplet, chaîne de Markov triplet, arbre de Markov triplet ;
  - 2: Initialisation du processus  $X^0$  et des paramètres de chaque processus du modèle triplet ;
  - 3: Estimation itérative des paramètres en utilisant les algorithmes itératifs EM, SEM ou ECI adapté au modèle de Markov triplet utilisé ;
  - 4: Estimation de la configuration finale des processus  $X$  et  $U$  par les estimateurs Bayésiens MAP ou MPM ;
  - 5: Transformation de l'image  $1D$  en  $2D$  ;
- 

## 1.9 Les modèles de Markov Flous

Le Modèle de Markov Flou(MMF) est une version floue de modèle de Markov caché introduit par [54], il modélise l'information floue dans l'image en se basant sur la combinaison entre la théorie des probabilités et la théorie des sous-ensembles flous. Les modèles flous utilisent la théorie des probabilités pour modéliser l'incertain, et la théorie de Zadeh [85] pour modéliser l'imprécision [75]. Ils conservent les mêmes propriétés que les modèles de Markov classiques. La différence est au niveau de la modélisation de l'information floue, en calculant les paramètres des classes floues à partir des paramètres des classes dures.

**Définition 1.9.1.** *Considérons le cas, de la segmentation d'une image  $Y = (Y_n)_{n=1}^N$  en deux classes dures 0 et 1 et une classe floue. Chaque pixel  $y_n$  peut appartenir soit à la classe 0 ou à la classe 1 ou à la classe floue [86, p. 38-40].*

*Selon, la théorie des sous-ensembles flous. Nous pouvons définir  $x_n$  par :*

$$x_n = (\varepsilon_n^0, \varepsilon_n^1) \in [0, 1]^2.$$

*ou :*

- $\varepsilon_n^0$  : est le degré d'appartenance du pixel  $x_n$  à la classe 0.
- $\varepsilon_n^1$  : est le degré d'appartenance du pixel  $x_n$  à la classe 1.

*Avec :  $\varepsilon_n^0 + \varepsilon_n^1 = 1$ .*

*Dans le cas flou. Le processus  $X = (X_n)_{n=1}^N$  prend ses valeurs dans l'intervalle  $\Omega_f = [0, 1]$ . Le processus  $X$  est représenté par :*

$$X = \varepsilon_n \tag{1.31}$$

*Avec :*

- $\varepsilon_n = 0$  si le pixel  $x_n$  appartient à la classe 0.
- $\varepsilon_n = 1$  si le pixel  $x_n$  appartient à la classe 1.
- $\varepsilon_n \in ]0, 1[$  si le pixel  $x_n$  est flou.

*La probabilité  $p(x)$  de la loi de  $X$  du modèle flou est définie sur l'intervalle  $\Omega_f$  par une densité qui*

inclue les composants dures qui sont discrètes et une composante floue qui est continue :

$$p(x) = h(\delta_0 + \delta_1 + \zeta) = h\nu \quad (1.32)$$

Avec :

- $\delta_0$  et  $\delta_1$  : sont des masses de Dirac en 0 et 1 .
- $\zeta$  : est la mesure de Lebesgue sur l'intervalle  $]0, 1[$ .

$$\nu = \delta_0 + \delta_1 + \zeta \quad (1.33)$$

Donc :

La distribution de la loi de  $X$  peut être définie par la densité  $h$  sur l'intervalle  $[0, 1]$  selon la mesure de  $\nu$  par :

$$h(0) + h(1) + \int_0^1 h(\varepsilon)d\nu(\varepsilon) = 1 \quad (1.34)$$

Les modèles de Markov classiques, couples, triplets ont été étendus aux modèles flous :

1. Les champs de Markov flous.
2. Les chaînes de Markov floues.
3. Les arbres de Markov flous.
4. Les modèles de Markov couples flous.
5. Les modèles de Markov triplets flous.

La démarche de segmentation des images par le MMF est présentée dans l'algorithme suivant :

---

**Algorithme 6** La démarche de segmentation d'images par MMF

---

- 1: Transformation de l'image d'entrée  $2D$  selon le modèle utilisé champs, chaîne, arbre à la forme  $1D$  ;
  - 2: Initialisation du processus  $X^0$  et des paramètres thématiques et flous ;
  - 3: Estimation itérative des paramètres thématiques et flous en utilisant les algorithmes itératifs EM, SEM ou ECI ;
  - 4: Estimation de la configuration finale du modèle flou  $X$  par les estimateurs Bayésiens MAP ou MPM ;
  - 5: Transformation de l'image résultat en  $2D$  ;
-



## 1.10 Conclusion

Dans ce chapitre, nous avons présenté les différents modèles de Markov utilisés en segmentation des images à savoir, les champs, les chaînes, les arbres, les modèles couples, les modèles triplets et les modèles flous. Les démarches de segmentation utilisées par ces modèles se ressemblent, la différence est au niveau de type, nombre et taille des paramètres ainsi que la manière d'estimation. Dans le chapitre suivant<sup>2</sup>, nous présentons un ensemble de comparaisons entre les estimateurs itératifs, entre les stratégies Bayésiennes, entre les chaînes de Markov cachées stationnaires et entre les chaînes de Markov cachées stationnaires et non-stationnaires utilisées en segmentation d'images niveau de gris et couleur.

# Chapitre 2

## Les chaines de Markov cachées en segmentation des images

### 2.1 Introduction

Dans ce chapitre nous présentons un ensemble d'études comparatives entre les estimateurs itératifs des paramètres [48], entre les stratégies Bayésiennes [48], entre les chaines de Markov stationnaires [87] et entre les chaines de Markov stationnaires et non-stationnaires [49]. Les comparaisons sont en termes de la qualité de segmentation, le temps d'exécution, la convergence et la complexité. Nous commençons ce chapitre par deux études comparatives la première entre les algorithmes d'estimation des paramètres, et l'autre entre les stratégies Bayésiennes dans une application de segmentation des images niveau de gris, couleurs [48, 52], médicales cérébrales IRM [88] en utilisant le modèle de Markov classique chaîne de Markov à bruit indépendant. Ensuite, nous présentons le modèle chaîne de Markov couple, nous comparons ces modèles avec les autres chaines de Markov cachées stationnaires. Nous présentons aussi, le modèle de Markov le plus général en segmentation d'images : chaîne de Markov triplet. Finalement, nous présentons la version floue du modèle chaîne de Markov : chaîne de Markov cachée floue.

### 2.2 Les chaines de Markov cachées à bruit indépendant

Les Chaines de Markov Cachées à Bruit Indépendant (CMC-BI) en anglais (Hidden Markov Chains with Independent Noise) sont parmi les modèles les plus classiques en segmentation d'images [56, p. 8-15]. Ces modèles ne prennent pas en compte l'information bruit dans l'image à segmenter [89].

Dans ce paragraphe, nous présentons le modèle classique chaîne de Markov cachée à bruit indépendant, les estimateurs itératifs des paramètres et les stratégies Bayésiennes. Nous effectuons des études comparatives : entre les estimateurs itératifs et entre les estimateurs Bayésiens pour segmenter des images : niveau de gris, couleurs et cérébrales IRM.

## 2.2.1 Présentation de modèle

Soit le processus  $Z = (X, Y)$ , avec  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$  et  $X = (X_n)_{n=1}^N \in \Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ .  $Z$  est une Chaîne de Markov Cachée à Bruit Indépendant si et seulement si elle vérifie les trois conditions suivantes :

1. le processus  $X$  est une chaîne de Markov, il est stationnaire, homogène, d'ordre 1, sa loi est :

$$p(x) = p(x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \quad (2.1)$$

2. Les observations  $Y$  sont conditionnellement indépendantes de  $X$ .

$$p(y|x) = \prod_{n=1}^N p(y_n|x) \quad (2.2)$$

3. chaque observation  $y_n$  dépend seulement de son état caché  $x_n$ .

$$p(y_n|x) = p(y_n|x_n) \quad (2.3)$$

La loi de  $Z$  est écrite comme suit :

$$p(z) = p(x, y) = p(x_1) \cdot p(y_1|x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot p(y_{n+1}|x_{n+1}) \quad (2.4)$$

Ce modèle admet deux lois de probabilité : la loi du processus caché  $X$  et la loi des observations  $Y$ , chaque loi admet des paramètres à estimer  $\theta = (\theta_x, \theta_y)$  : les paramètres  $\theta_x$  du processus caché  $X$  et les paramètres  $\theta_y$  du processus observé  $Y$ .

Les paramètres  $\theta_x$  de processus caché  $X$  sont : la loi initiale  $PI = p(x_1 = i) \forall i \in \Omega$  de taille  $K$ , et la matrice de transition  $A^0(i, j) = p(x_{n+1} = \omega_j | x_n = \omega_i)$  entre les classes  $i$  et  $j \forall i, j \in \Omega$  de taille  $K * K$ .

Les paramètres  $\theta_y$  de processus observé  $Y$  dépendent de la loi de probabilité suivie par les observations  $Y$ . Dans ce travail, nous supposons que le processus  $Y$  suit la loi gaussienne, donc, les paramètres de  $Y$  sont : la moyenne  $\mu$  de taille  $K$  et la variance  $\sigma^2$  de taille  $K$ .

La loi des observations  $Y$  est définie par la densité gaussienne  $f_i$  pour chaque classe  $i$  :

$$P(Y_n = y_n | X = i) = f_i(y_n) = \frac{1}{\sqrt{2\pi(\sigma_i)^2}} \exp\left[-\frac{(y_n - \mu_i)^2}{(\sigma_i)^2}\right] i \in \Omega. \quad (2.5)$$

nous pouvons remplacer la probabilité  $p(y|x)$  par la densité gaussienne  $f_x(y)$ , la loi de  $Z$  devient :

$$p(z) = p(x_1) \cdot f_{x_1}(y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot f_{x_{n+1}}(y_{n+1}) \quad (2.6)$$

Le modèle CMC-BI a deux types de paramètres  $\theta = (\theta_x, \theta_y)$  les paramètres de processus caché  $X$  :  $\theta_x = (PI, A)$  et les paramètres de processus observé  $Y$  :  $\theta_y = (\mu, (\sigma)^2)$ .

## 2.2.2 Estimation des paramètres

Après la phase d'initialisation des paramètres de chaque processus  $X$  et  $Y$ . Le modèle CMC-BI estime ces paramètres itérativement en utilisant les estimateurs itératifs.

Dans ce paragraphe, nous présentons les différents algorithmes utilisés par le modèle CMC-BI pour estimer les paramètres tel que : EM(Espérance-Maximisation), GEM(Général Espérance-Maximisation), SEM(Stochastique Espérance-Maximisation), MCEM(Monté Carlo Espérance-Maximisation),

ECI(Estimation Conditionnelle Itérative). Ainsi que les critères Bayésiens MAP Maximum *a posteriori* et MPM(Mode *a posteriori* des Marginales).

Les estimateurs itératifs sont des algorithmes qui estiment les paramètres itérativement jusqu'à convergence. La convergence de ces estimateurs dépend d'un certain critère d'arrêt par exemple un seuil d'arrêt, nombre d'itérations [64, p. 33]....

Chaque estimateur a ses propres principes et stratégies pour estimer les paramètres.

L'algorithme EM maximise la probabilité de vraisemblance  $P(x, y|\theta)$ , sa convergence dépend des paramètres initiaux [90]. Deux autres versions d'EM. Ce sont des versions stochastiques SEM [91] et MCEM [92], ces algorithmes utilisent sur l'approximation de Monté Carlo [93, p. 51]. En remplaçant la probabilité de vraisemblance par l'espérance conditionnelle  $\rho = \frac{1}{\tau} \cdot \sum_{\tau=1}^{\tau} .P(x, y|\theta)$ . Ils sont basés sur le principe SIP (Stochastic Imputation Principale) [94], ce principe consiste à générer un processus non observable conditionnellement à un autre processus observable.

GEM [95] est une version générale de EM, que nous pouvons l'utiliser lorsque la maximisation de la vraisemblance devient difficile [96, p. 8]. Un autre algorithme utilisé dans cette étude comparative est : l'algorithme itératif ECI [97], il a un principe identique à celui de MCEM [96, p. 9].

Pour estimer les paramètres de CMC-BI, ces estimateurs utilisent l'algorithme de Baum-Welch conditionnel [56, p. 33-34], pour calculer les probabilités dans la première phase d'estimation ensuite, chaque estimateur utilise ses propres stratégies pour estimer les paramètres du modèle de Markov. Ces stratégies sont soit déterministes ou stochastiques ou combinant les deux.

### a. L'algorithme de Baum-Welch

Baum-Welch est une méthode récursive [98] qui se déroule de la manière suivante en commençant par  $n = 1$  :

**1-** Le calcul des probabilités progressives de Forward  $\alpha_n(x_n) = p(y_1, \dots, y_n, x_n)$  en utilisant l'algorithme de Forward qui se déroule de la manière suivante en deux phases :

1. Initialisation :  $n = 1$

$$\alpha_1(x_1) = p(x_1) \cdot p(y_1|x_1) \tag{2.7}$$

2. Induction : pour  $n > 1$

$$\alpha_n(x_n) = \sum_{x_{n-1}} \alpha_{n-1}(x_{n-1})p(x_n|x_{n-1}) \cdot p(y_n|x_n) \quad (2.8)$$

**2-** Le calcul des probabilités rétrogrades de Backward  $\beta_n(x_n) = p(y_{n+1}, \dots, y_N|x_n)$  en utilisant l'algorithme de Backward qui se déroule de la manière suivante dans la direction opposée commençant par  $n = N$  :

1. initialisation :  $n = N$

$$\beta_N(x_N) = 1 \quad (2.9)$$

2. induction : for  $n < N$

$$\beta_n(x_n) = \sum_{x_{n+1}} \beta_{n+1}(x_{n+1})p(x_{n+1}|x_n) \cdot p(y_n|x_{n+1}) \quad (2.10)$$

**3-** Après avoir calculé les probabilités de Forward, Backward  $\alpha_n(x_n)$  et  $\beta_n(x_n)$ , Baum Welch estime les probabilités conjointes *a posteriori*  $\gamma_n(x_n, x_{n+1}) = p(x_n, x_{n+1}|y_n)$  et les probabilités marginales *a posteriori*  $\xi_n(x_n) = p(x_n|y_n)$  à partir de  $\alpha_n(x_n)$  et  $\beta_n(x_n)$  en utilisant les formules mathématiques suivantes :

$$\gamma_n(x_n, x_{n+1}) = \alpha_n(x_n) \cdot p(x_{n+1}|x_n) \cdot p(y_{n+1}|x_{n+1}) \cdot \beta_{n+1}(x_{n+1}) \quad (2.11)$$

$$\xi_n(x_n) = \sum_{x_{n+1}} \gamma_n(x_n, x_{n+1}) \quad (2.12)$$

ou bien :

$$\xi_n(x_n) = \alpha_n(x_n) \cdot \beta_n(x_n) \quad (2.13)$$

## b. L'algorithme EM

L'algorithme de EM utilise Baum-Welch pour calculer les probabilités qui permettent d'estimer les paramètres des deux processus caché et observé  $\theta = (\theta_x, \theta_y)$  de la CMC-BI en utilisant la stratégie déterministe. EM se déroule comme suit :

---

**Algorithme 7** L'algorithme EM

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (PI^0, A^0, \mu_i^0, (\sigma_i^0)^2, f_i^0)$ **SORTIES:**  $\theta^Q = (PI^Q, A^Q, \mu_i^Q, (\sigma_i^Q)^2, f_i^Q)$ 

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: EM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  :

$$PI^{q+1}(i) = \xi_1^q(i) \quad (2.14)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.15)$$

- 4: EM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  :

$$\mu_i^{q+1} = \frac{\sum_{n=1}^N y_n \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.16)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^{q+1})^2 \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.17)$$

- 5: EM calcule la densité gaussienne  $f$  en utilisant l'équation 2.5
  - 6:  $q = q + 1$
  - 7: **fin pour**
- 

**c. L'algorithme GEM**

L'algorithme GEM a la même procédure qu'EM, sauf que nous ajoutons à l'étape  $M$  une comparaison entre les paramètres obtenus à l'itération précédente  $q$ ,  $(\theta^q)$  et ceux de l'itération actuelle  $q + 1$ ,  $(\theta^{q+1})$ . Finalement, il sélectionne le maximum [96, p. 10].

**d. L'algorithme SEM**

SEM utilise Baum-Welch pour calculer les probabilités qui permettent d'estimer les paramètres de CMC-BI. Il utilise la stratégie d'estimation stochastique pour estimer ces paramètres.

---

**Algorithme 8** L'algorithme SEM

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (PI^0, A^0, \mu_i^0, (\sigma_i^0)^2, f_i^0)$ **SORTIES:**  $\theta^Q = (PI^Q, A^Q, \mu_i^Q, (\sigma_i^Q)^2, f_i^Q)$ 

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: SEM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: SEM simule l'échantillon  $X^q$  aléatoirement une seule fois  $\tau = 1$ .
- 4: SEM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  :

$$PI^{q+1}(i) = \frac{1}{N-1} \sum_{n=1}^N \mathbf{1}[x_n = i] \quad (2.18)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=2}^N \mathbf{1}[x_n = j, x_{n-1} = i]}{N-1} \quad (2.19)$$

- 5: SEM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  :

$$\mu_i^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \quad (2.20)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^{q+1})^2 \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \quad (2.21)$$

- 6: SEM calcule la densité gaussienne  $f$  en utilisant l'équation 2.5
  - 7:  $q = q + 1$
  - 8: **fin pour**
- 

**e. L'algorithme MCEM**

MCEM a la même démarche que le SEM, sauf qu'il simule l'échantillon  $X^q$  plus qu'une seule fois  $\tau > 1$  à chaque itération  $q$  [96, p. 10].

**f. L'algorithme ECI**

Introduit par [99], l'algorithme ECI utilise Baum-Welch pour calculer les probabilités qui permettent d'estimer les paramètres de CMC-BI. Il utilise la stratégie d'estimation déterministe pour estimer les paramètres du processus caché  $X$ , et la stratégie stochastique pour calculer les paramètres du processus observé  $X$ .

---

**Algorithme 9** L'algorithme ECI

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (PI^0, A^0, \mu_i^0, (\sigma_i^0)^2, f_i^0)$ **SORTIES:**  $\theta^Q = (PI^Q, A^Q, \mu_i^Q, (\sigma_i^Q)^2, f_i^Q)$ 

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: ECI calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: ECI simule l'échantillon  $X^q$  aléatoirement une seule fois  $\tau = 1$ .
- 4: ECI estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  :

$$PI^{q+1}(i) = \xi_1^q(i) \quad (2.22)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.23)$$

- 5: ECI estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  :

$$\mu_i^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \quad (2.24)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^{q+1})^2 \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \quad (2.25)$$

- 6: ECI calcule la densité gaussienne  $f$  en utilisant l'équation 2.5
  - 7:  $q = q + 1$
  - 8: **fin pour**
- 

**g. Les stratégies Bayésiennes**

Après la convergence des algorithmes estimateurs, CMC-BI utilise les stratégies Bayésiennes : MAP ou MPM pour estimer la configuration finale du processus caché  $X^Q$ , en utilisant les paramètres de l'itération finale  $Q$   $\theta^Q$ .

**-L'algorithme MAP**

L'estimateur MAP (Maximum *a posteriori*) consiste à pénaliser de la même manière toute valeur de  $\bar{x}$  différente de  $x$ .

Sa fonction coût est [86, p. 11] :

$$L(x, \bar{x}) = 1 - \delta(x, \bar{x}) \quad (2.26)$$

où  $\delta$  : est le symbole de Kronecker

$$\delta(x, \bar{x}) = \begin{cases} 1 & \text{si } x = \bar{x} \\ 0 & \text{sinon} \end{cases}$$



Cet estimateur utilise cette fonction de coût :

$$\bar{x}_{map}(y) = \mathbf{argmax}_x(P(X = x|Y = y)) \quad (2.27)$$

La solution de MAP en segmentation d'images par chaîne de Markov cachée est d'appliquer l'algorithme de Viterbi [100].

### -L'algorithme MPM

Contrairement au MAP, MPM (Mode *a posteriori* des Marginales) estime la configuration de  $x$  proportionnellement au nombre de pixels mal classés. Sa fonction coût est [86, p. 11] :

$$L(x, \bar{x}) = \sum_{n=1}^N (1 - \delta(x_n, \bar{x}_n)) \quad (2.28)$$

Cet estimateur maximise la probabilité marginale locale *a posteriori* pour chaque pixel  $x_n$  :

$$\bar{x}_{nmpm} = \mathbf{argmax}_{x_n}(P(X_n = x_n|Y_n = y_n)) \quad (2.29)$$

Le principe du MPM consiste à estimer pour chaque pixel  $y_n$  une classe d'appartenance  $i$  qui maximise la probabilité  $P(X_n = i|Y_n = y_n)$ .

La solution MPM en segmentation d'images par chaîne de Markov cachée est d'appliquer la formule mathématique suivante [101] :

$$\bar{x}_{nmpm} = \mathbf{argmax}_{i \in \Omega}(\alpha_i(n) \cdot \beta_i(n)) = \mathbf{argmax}_{i \in \Omega}((\xi_i(n))) \quad (2.30)$$

## 2.2.3 Résultats et expérimentations

Ce paragraphe présente une série d'expériences réalisée pour montrer les différences entre les estimateurs itératifs utilisés et entre les stratégies Bayésiennes. Nous avons utilisé le modèle CMC-BI pour segmenter des images niveau de gris, couleurs, couleurs texturées, cérébrales IRM.

### a. Segmentation des images couleurs et niveau de gris

Nous avons effectué des expériences pour segmenter des images niveau de gris et couleurs. Nous avons comparé les cinq estimateurs itératifs en termes de la qualité, la convergence et le temps d'exécution. Pour initialiser les paramètres  $\theta^0 = (\theta_x^0, \theta_y^0)$  nous avons suivi la démarche suivante :

La loi initiale  $PI^0$  et la matrice de transition  $A^0$  sont initialisées par :

$$PI^0(i) = \frac{1}{K} \forall i \in \Omega \quad (2.31)$$

et

$$A^0(i, j) = \begin{cases} A(i, j) = \frac{1}{K} & \text{si } i = j \\ A(i, j) = \frac{1}{2(K-1)} & \text{sinon} \end{cases} \quad \forall i, j \in \Omega \quad (2.32)$$

Nous avons utilisé l'algorithme de K-means pour initialiser le processus initial  $X^0$ . Les paramètres de la loi observé  $\theta_y^0$  sont initialisés en utilisant le processus initial  $X^0$  par :

$$\mu_i^0 = \frac{\sum_{n=1}^N y_n \cdot 1[x_n^0 = i]}{N}. \quad (2.33)$$

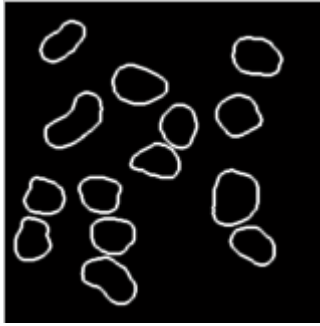
La variance  $\sigma^0$  est calculée à partir de la moyenne initiale  $\mu^0$  par :

$$(\sigma_i^0)^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^0)^2 \cdot 1[x_n^0 = i]}{N}. \quad (2.34)$$

Pour les images couleurs, nous avons calculé la moyenne initiale  $\mu^0$  et la variance initiale  $\sigma^0$  de chaque niveau de couleur RVB(Rouge, Vert, Bleu). Le nombre de classes d'appartenance  $K$  dans les deux expériences égale à 2  $K = 2$ .

● **Les résultats visuels :**

Les figures suivantes représentent les résultats visuels obtenus par chaque estimateur sous MAP et MPM :



(a) Image originale  $Y$



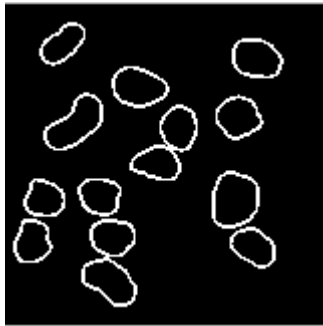
(b) Configuration initiale  $X^0$



(c) EM+MPM



(d) EM+MAP



(e) GEM+MPM



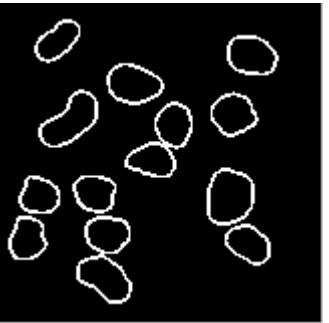
(f) GEM+MAP



(g) SEM+MPM



(h) SEM+MAP



(i) MCEM+MPM



(j) MCEM+MAP



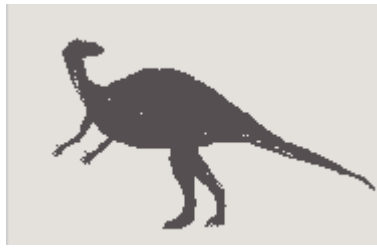
(k) ECI+MPM



(l) ECI+MAP



(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



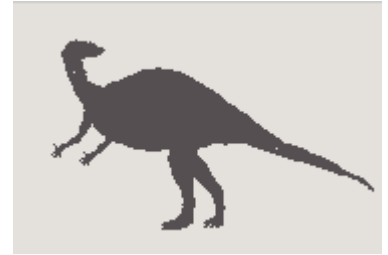
(c) EM+MPM



(d) EM+MAP



(e) GEM+MPM



(f) GEM+MAP



(g) SEM+MPM



(h) SEM+MAP



(i) MCEM+MPM



(j) MCEM+MAP



(k) ECI+MPM



(l) ECI+MAP

FIGURE 2.2 – Résultats de segmentation de l'image couleur de dinosaure du data set Wang [103]

Les résultats visuels montrent que les cinq estimateurs donnent les mêmes résultats de segmentation sous MPM et MAP.

• **Les résultats quantitatifs :**

Pour valider ces résultats, nous avons calculé l'indice de PSNR et le taux d'erreur de chaque estimateur. Le tableau 2.1 représente les valeurs de l'indice de PSNR(Peak Signal-to-Noise Ratio) [102] et du taux d'erreur par estimateur obtenues dans l'expérience 1 et le tableau 2.2 représente les valeurs de PSNR et de taux d'erreur obtenues dans l'expérience 2.

Estimateur	PSNR MPM	PSNR MAP	taux d'erreur MPM	taux d'erreur MAP
EM	14,0023	14,0023	0,00	0,00
GEM	14,0023	14,0023	0,00	0,00
SEM	13,9665	13,9665	0,00	0,00
MCEM	13,9665	13,9665	0,00	0,00
ECI	13,9665	13,9665	0,00	0,00

TABLE 2.1 – Les valeurs de l'indice de PSNR et de taux d'erreur de l'expérience 1

Estimateur	PSNR MPM	PSNR MAP	taux d'erreur MPM	taux d'erreur MAP
EM	22,4572	22,3528	6,2820	6,2820
GEM	22,4572	22,3528	6,2820	6,2820
SEM	22,4572	22,3528	6,2820	6,2820
MCEM	22,4572	22,3528	6,2820	6,2820
ECI	22,4572	22,3528	6,2820	6,2820

TABLE 2.2 – Les valeurs de l'indice de PSNR et de taux d'erreur de l'expérience 2

Les valeurs de l'indice de PSNR et du taux d'erreur confirment les résultats visuels. Pas de différence entre les estimateurs sous MAP et MPM.

Aussi, nous avons calculé la convergence et le temps d'exécution de chaque estimateur. les tableaux 2.2.3 et 2.2.3 présentent respectivement la convergence et le temps d'exécution en secondes de chaque estimateur pour les deux expériences 1 et 2.

Estimateur	La Convergence	Le temps d'exécution
EM	3 itérations	733.289987 s
GEM	2 itérations	944.450694 s
SEM	4 itérations	939.820204 s
MCEM	6 itérations	1257.959682 s
ECI	3 itérations	1129.237750 s

TABLE 2.3 – La convergence et le temps d'exécution par estimateur de l'expérience 1

Estimateur	La convergence	Le temps d'exécution
EM	13 itérations	2128.269554 s
GEM	2 itérations	1976.419812 s
SEM	8 itérations	1927.977150 s
MCEM	8 itérations	2553.556252 s
ECI	7 itérations	2430.018185 s

TABLE 2.4 – La convergence et le temps d'exécution par estimateur de l'expérience 2

Nous remarquons que, GEM converge rapidement après 2 itérations dans les deux expériences, cet algorithme ressemble à un point de convergence fixe, et pas à un maximum. Pour le temps d'exécution, en moyenne, il est similaire pour tous les estimateurs itératifs sauf l'algorithme MCEM qui est vu très lent car, il simule aléatoirement l'échantillon  $X$  plus qu'une seule fois. Le temps d'estimation des paramètres de l'image couleur est lent que celle de l'image niveau de gris car, nous estimons les paramètres du processus observé  $Y$  de chaque bande de couleur (Rouge, Vert, Bleu). Nous avons comparé les estimateurs Bayésiens MPM et MAP en terme de temps d'exécution. Le tableau 2.2.3 représente le temps d'exécution en secondes de MPM et MAP par expérience.

Expérience	MPM	MAP
Expérience 1	2.437575 s	6.213053 s
Expérience 2	3.366573 s	20.904021 s

TABLE 2.5 – Le temps d'exécution de MPM et MAP

MPM est plus rapide que MAP.

Dans les deux expériences précédentes, nous avons obtenu les mêmes résultats. La différence entre EM, GEM, SEM, MCEM et ECI sous MAP et MPM n'est pas remarquable, juste une petite différence au niveau de temps d'exécution et de convergence. De plus, la différence entre les estimateurs MAP et MPM montre que, MPM est plus lent que MAP.

## b. Segmentation des images couleurs texturées

Dans ces expériences, nous avons segmenté quelques images couleurs texturées de la base d'images VisTex de taille  $(128 * 128)$ . Certes que MCEM a la même démarche que SEM, et GEM a la même démarche qu'EM. Nous avons comparé juste les trois algorithmes EM, SEM et ECI sous MPM et MAP. Pour initialiser les paramètres, nous avons suivi la même démarche que celle de la segmentation des images couleurs présentée précédemment dans 2.2.3. La comparaison était au niveau de la qualité de segmentation.

• **Les résultats visuels :**

Les figures suivantes montrent les résultats visuels obtenus par estimateur sous MPM et MAP :



(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



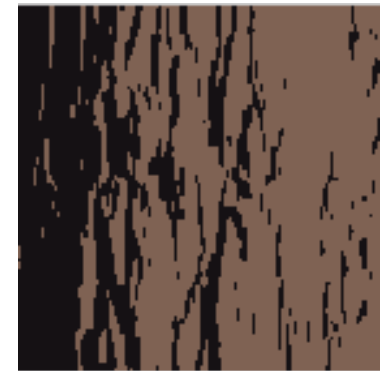
(c) EM+MPM



(d) EM+MAP



(e) SEM+MPM



(f) SEM+MAP



(g) ECI+MPM

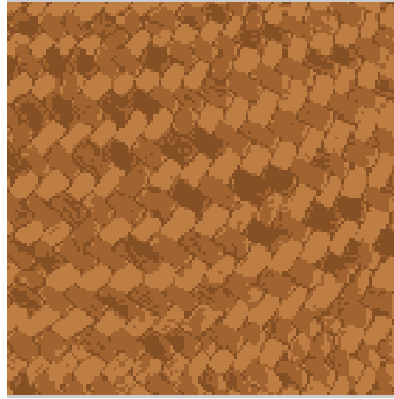


(h) ECI+MAP

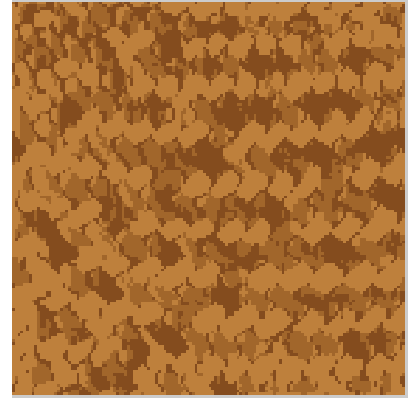
FIGURE 2.3 – Résultats de segmentation d'une image texturée 1 avec  $K = 2$



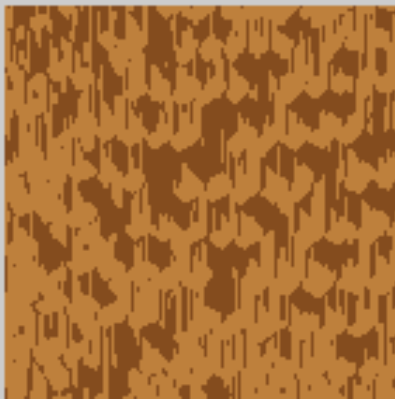
(a) Image originale  $Y$



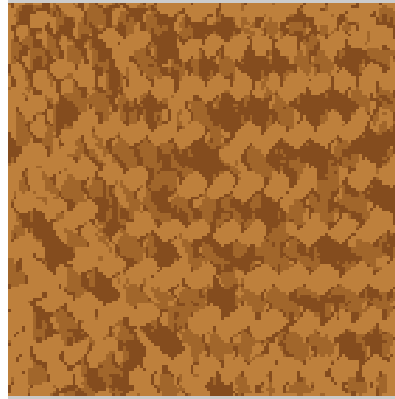
(b) Configuration initiale  $X^0$



(c) EM+MPM



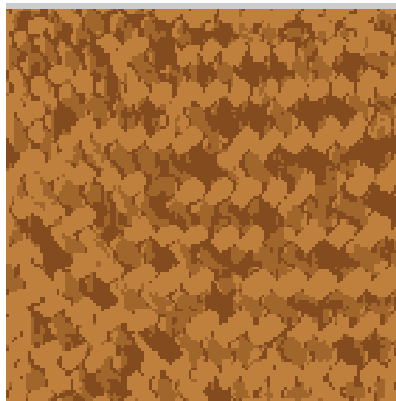
(d) EM+MAP



(e) SEM+MPM



(f) SEM+MAP



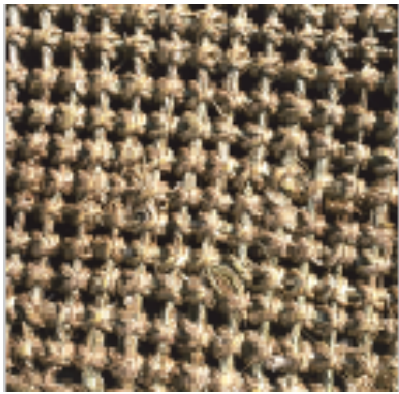
(g) ECI+MPM



(h) ECI+MAP

FIGURE 2.4 – Résultats de segmentation d'une image texturée 2 avec  $K = 3$





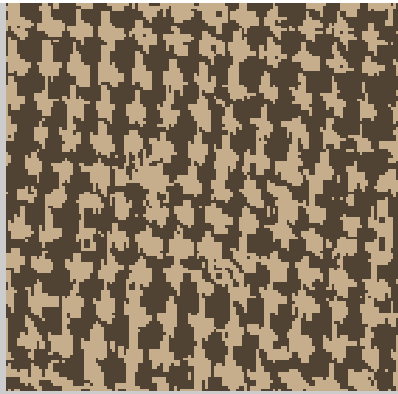
(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



(c) EM+MPM



(d) EM+MAP



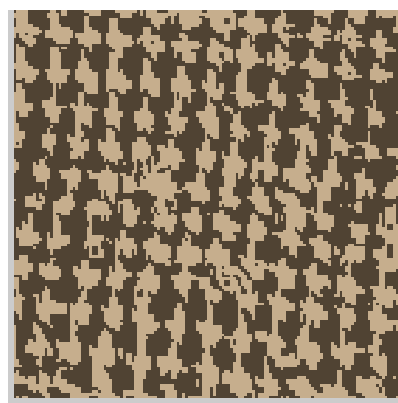
(e) SEM+MPM



(f) SEM+MAP



(g) ECI+MPM

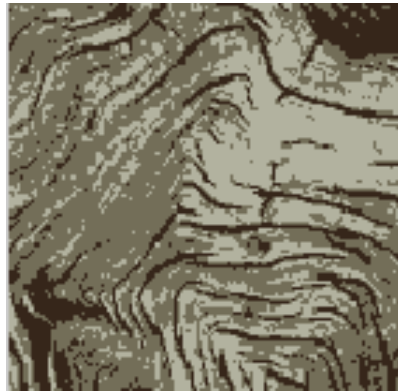


(h) ECI+MAP

FIGURE 2.5 – Résultats de segmentation d'une image texturée 3 avec  $K = 2$



(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



(c) EM+MPM



(d) EM+MAP



(e) SEM+MPM



(f) SEM+MAP

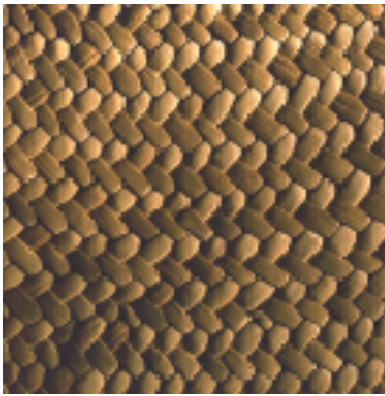


(g) ECI+MPM

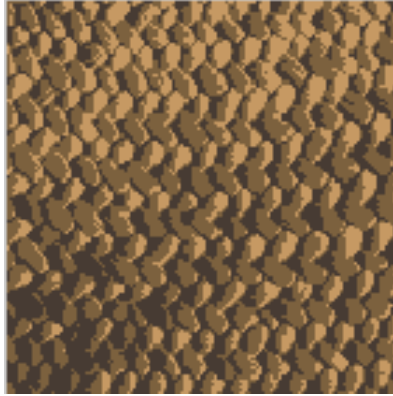


(h) ECI+MAP

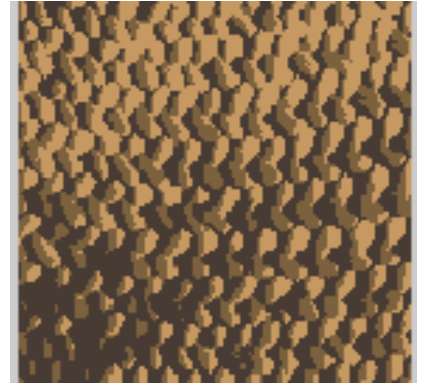
FIGURE 2.6 – Résultats de segmentation d'une image texturée 4 avec  $K = 3$



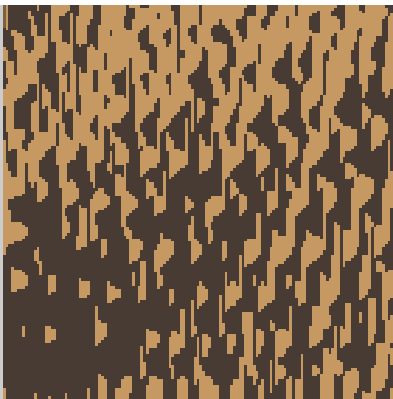
(a) Image originale  $Y$



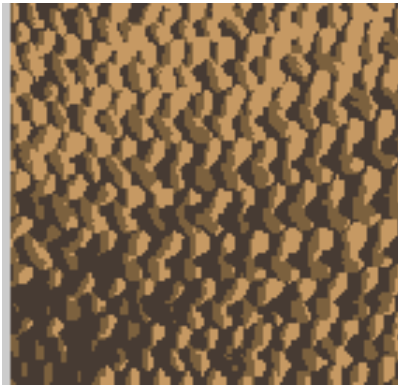
(b) Configuration initiale  $X^0$



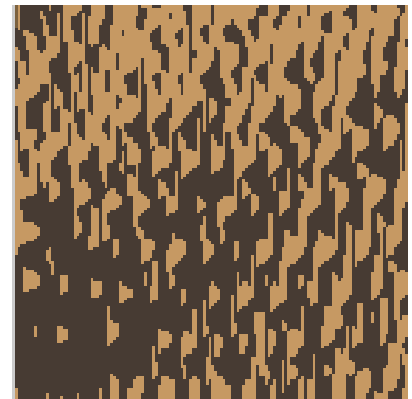
(c) EM+MPM



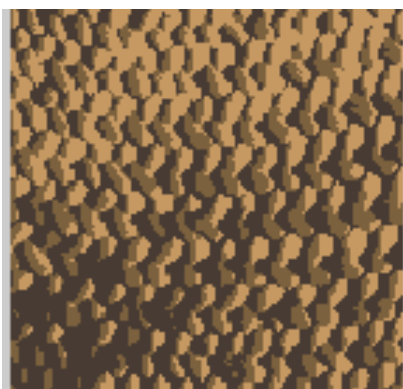
(d) EM+MAP



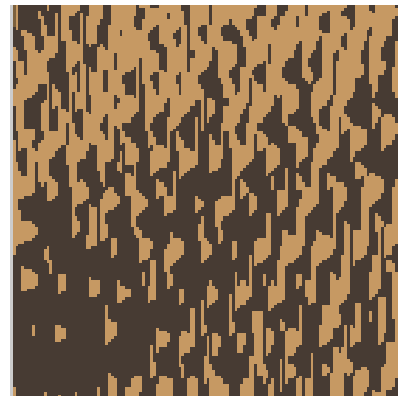
(e) SEM+MPM



(f) SEM+MAP



(g) ECI+MPM



(h) ECI+MAP

FIGURE 2.7 – Résultats de segmentation d'une image texturée 5 avec  $K = 2$

Les résultats de segmentation de ces images montrent que MPM donne de meilleurs résultats que MAP.

• **Les résultats quantitatifs :**

Nous présentons, les valeurs obtenues de l'indice PSNR et du taux d'erreur calculées pour chaque expérience ( $e_1, \dots, e_5$ ) par estimateur. Le tableau 2.6 indique les valeurs de l'indice de PSNR et le tableau 2.7 décrit les valeurs de taux d'erreur.

	EM+MPM	SEM+MPM	ECI+MPM	EM+MAP	SEM+MAP	ECI+MAP
e1	17,9322	17,9322	17,9322	17,6200	17,6200	17,6200
e2	27,3944	27,3944	27,3944	23,8020	23,8020	23,8020
e3	18,2774	18,2774	18,2774	17,9998	17,9998	17,9998
e4	18,6658	18,6658	18,6658	13,5188	13,5188	13,5188
e5	18,3181	18,3181	18,3181	18,0893	18,0893	18,0893

TABLE 2.6 – Les valeurs de l'indice de PSNR

	EM+MPM	SEM+MPM	ECI+MPM	EM+MAP	SEM+MAP	ECI+MAP
e1	1,3590	1,3590	1,3590	1,3853	1,3853	1,3853
e2	2,7588	2,7588	2,7588	4,2562	4,2562	4,2562
e3	1,0412	1,0412	1,0412	1,0417	1,0417	1,0417
e4	4,9388	4,9388	4,9388	10,3733	10,3733	10,3733
e5	2,2186	2,2186	2,2186	3,2247	3,2247	3,2247

TABLE 2.7 – Les valeurs de taux d'erreur

Les valeurs de PSNR et de taux d'erreur d' EM, SEM et ECI sont égaux sous MAP et MPM, pas de différence entre EM, SEM et ECI. Cette différence peut s'expliquer probablement par l'utilisation des mêmes valeurs des paramètres initiaux, ainsi que les valeurs des paramètres finaux estimées de chaque estimateur sont plus proches.

MPM fournit les meilleurs résultats de segmentation que MAP. La différence entre les résultats de MPM et MAP est importante. Nous pouvons expliquer ça par : la fonction coût utilisée par MPM pour estimer la classe d'appartenance de chaque pixel est plus performante que celle de MAP. La fonction coût de MPM estime la configuration de chaque pixel  $y_n$  proportionnellement au nombre de pixels mal classés, contrairement à MAP.

**c. Segmentation des images cérébrales IRM**

Ici nous comparons, les deux algorithmes EM et ECI en termes de la qualité de segmentation et la convergence sous MPM. Nous utilisons ces algorithmes pour segmenter des images cérébrales IRM . Pour initialiser les paramètres, nous avons suivi la démarche suivante : La loi initiale  $PI^0$  et

la matrice de transition  $A^0$  sont initialisées par :

$$\mathbf{PI}^0 = \begin{pmatrix} 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$$

$$\mathbf{A}^0 = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$$

avec  $K = 3$  Le processus initial  $X^0$  est initialisé par K-means .

La moyenne et la variance initiaux  $\mu^0$  et  $\sigma^0$  sont initialisées par les equations 2.33 et 2.34.

• **Les résultats visuels :**

Nous avons réalisé dix expériences pour dix images IRM. Nous supposons que les images IRM utilisées sont filtrées. Après la phase de segmentation. Nous avons pris l'image résultat de la segmentation  $X$  obtenue par ECI et nous avons détecté la région d'intérêt(ROI). En utilisant la technique de seuillage qui consiste à éliminer toutes les pixels de l'image qui n'appartiennent pas la région d'intérêt(ROI) . Pour faciliter le diagnostic du type de tumeur(bénigne ou maligne), nous prenons l'image originale  $Y$  et nous marquons la position de la tumeur par la couleur blanche. Les résultats obtenus, pour chaque expérience sont disponibles dans les figures suivantes :

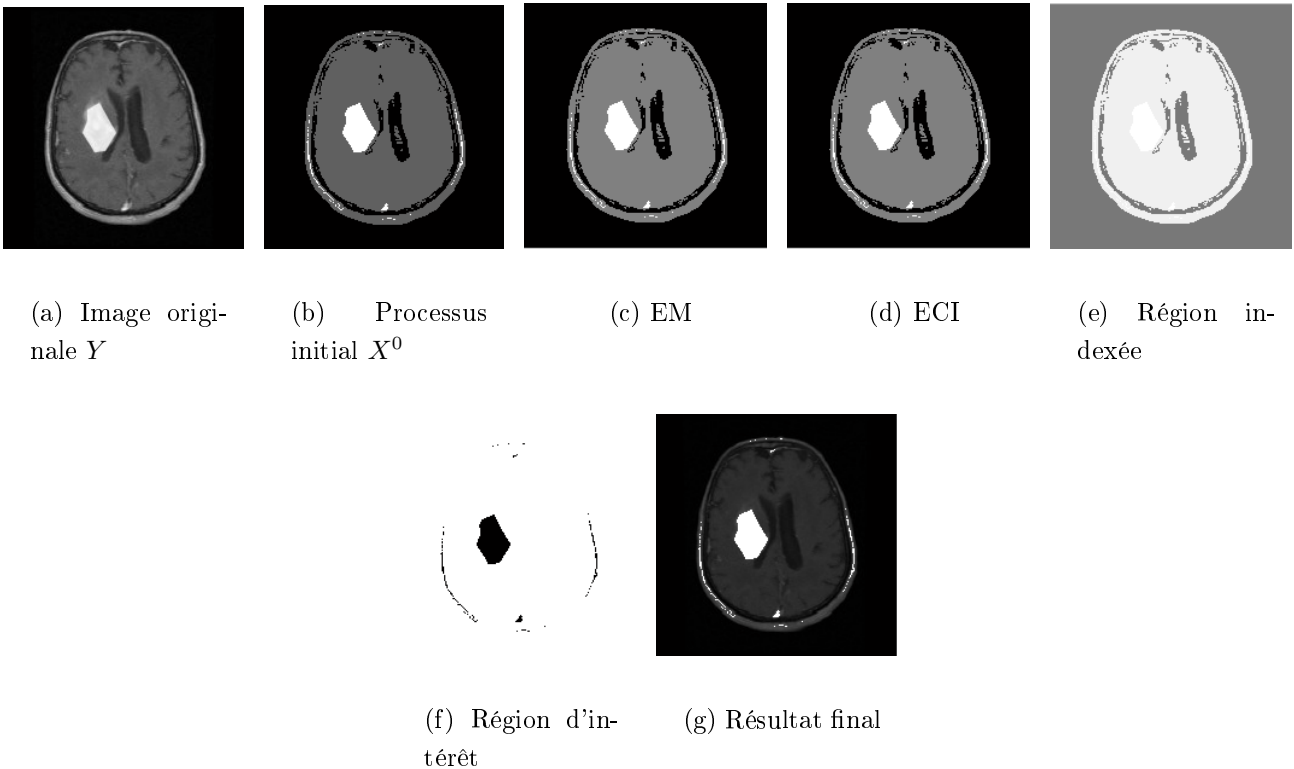


FIGURE 2.8 – Résultats de segmentation et extraction de la ROI expérience 1

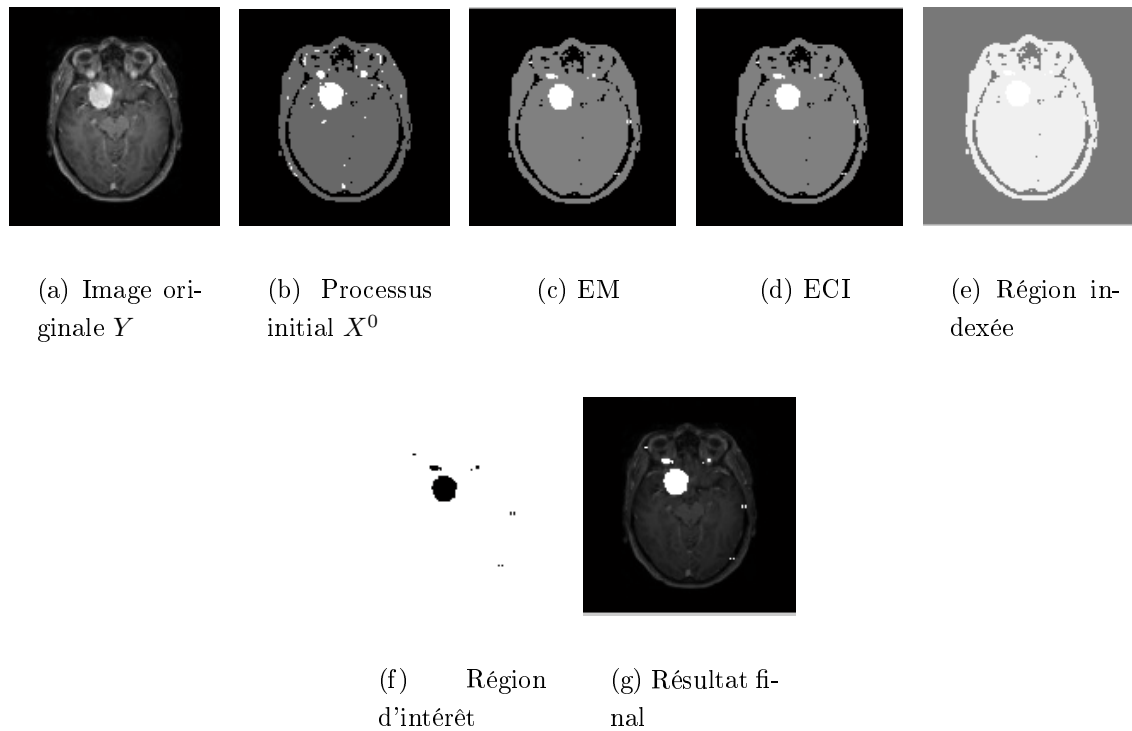


FIGURE 2.9 – Résultats de segmentation et extraction de la ROI expérience 2

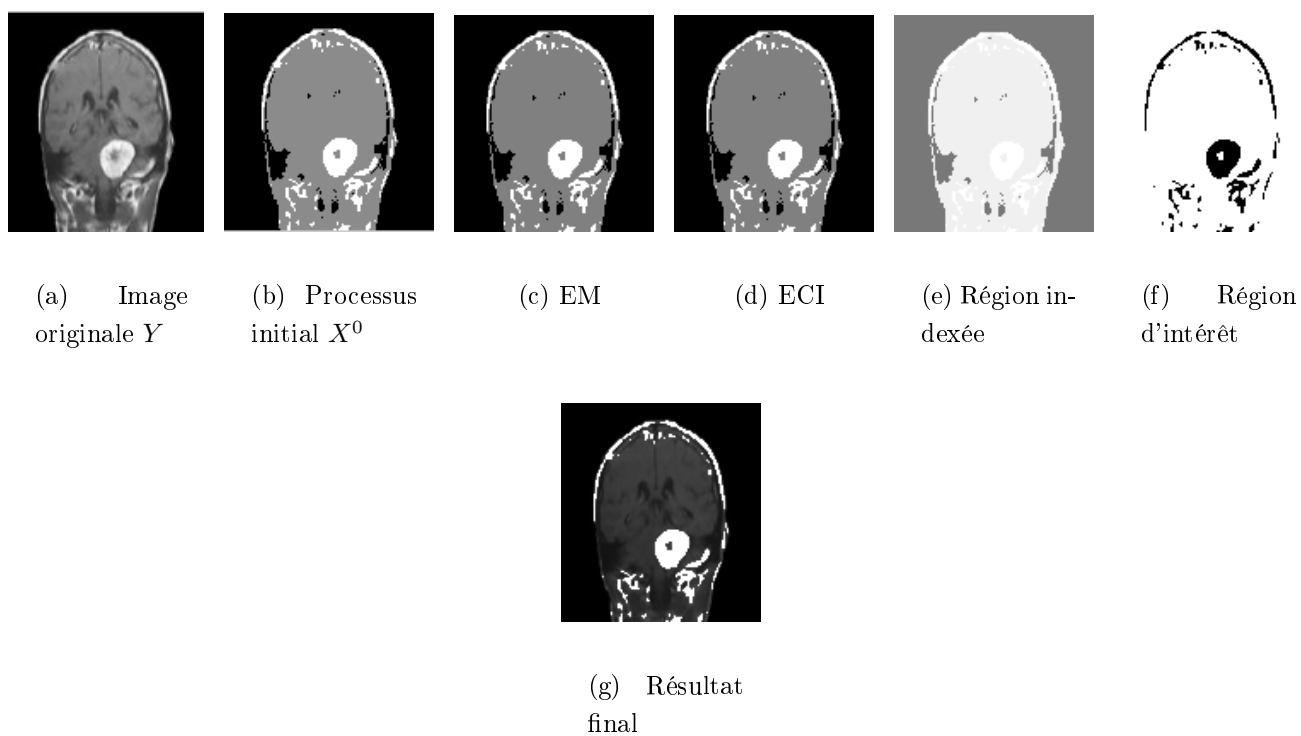


FIGURE 2.10 – Résultats de segmentation et extraction de la ROI expérience 3

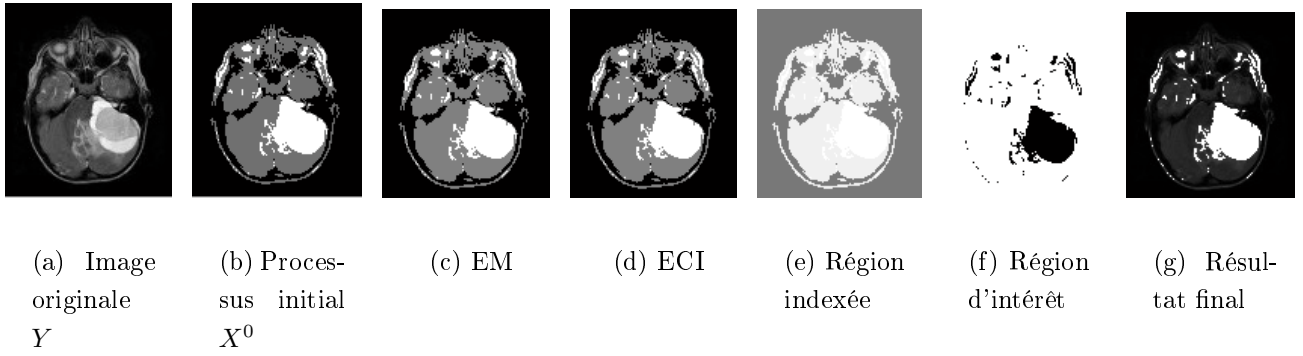


FIGURE 2.11 – Résultats de segmentation et extraction de la ROI expérience 4

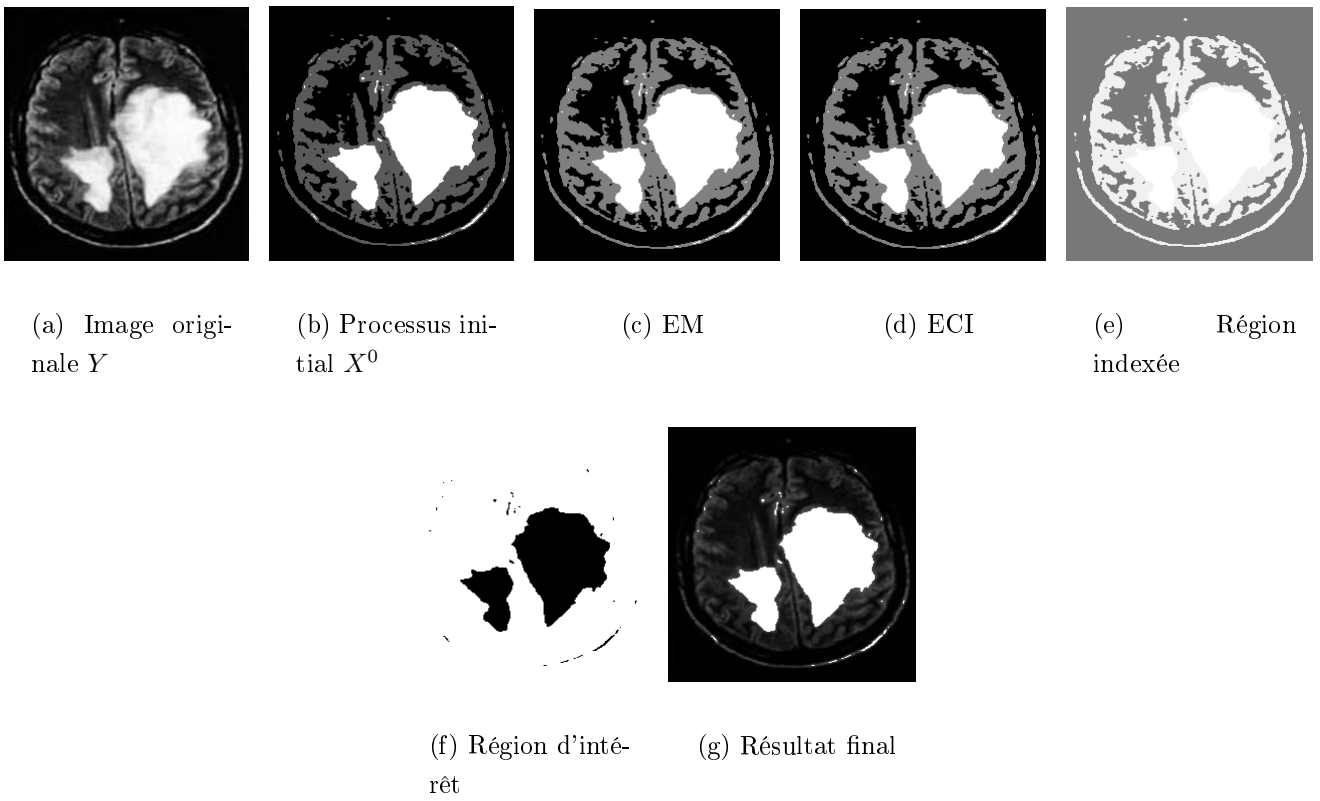


FIGURE 2.12 – Résultats de segmentation et extraction de la ROI expérience 5

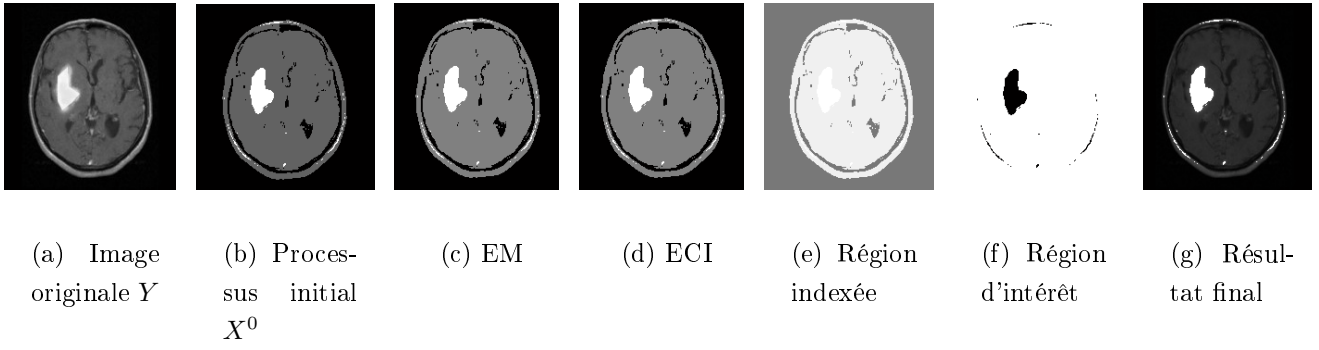


FIGURE 2.13 – Résultats de segmentation et extraction de la ROI expérience 6

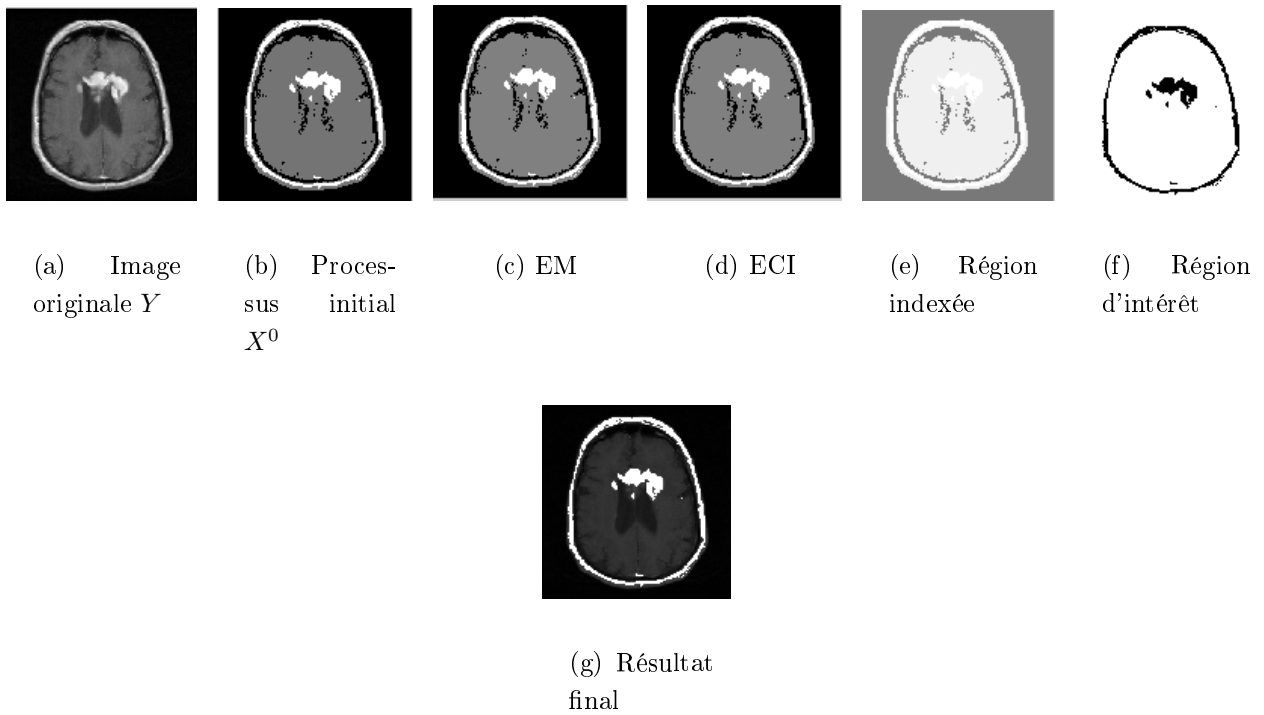
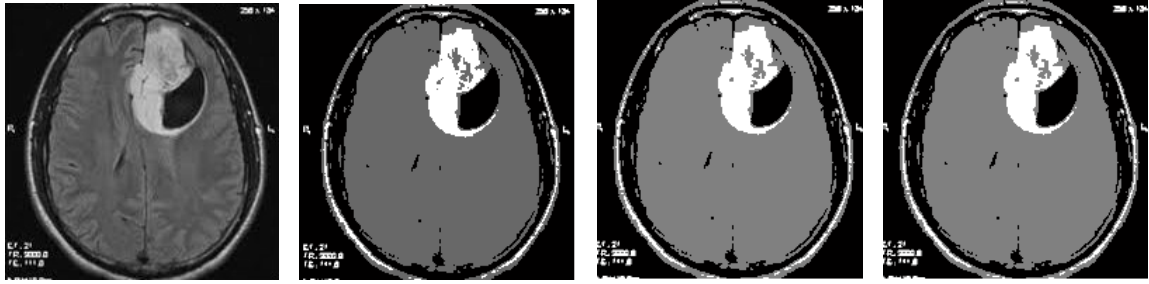


FIGURE 2.14 – Résultats de segmentation et extraction de la ROI expérience 7



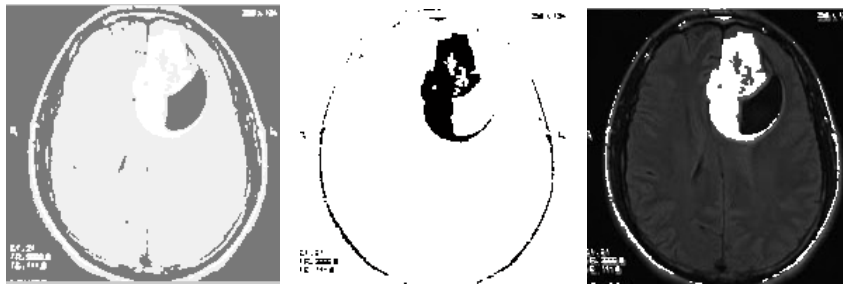


(a) Image originale  
Y

(b) Processus initial  
 $X^0$

(c) EM

(d) ECI



(e) Région indexée

(f) Région d'intérêt

(g) Résultat final

FIGURE 2.15 – Résultats de segmentation et extraction de la ROI expérience 8

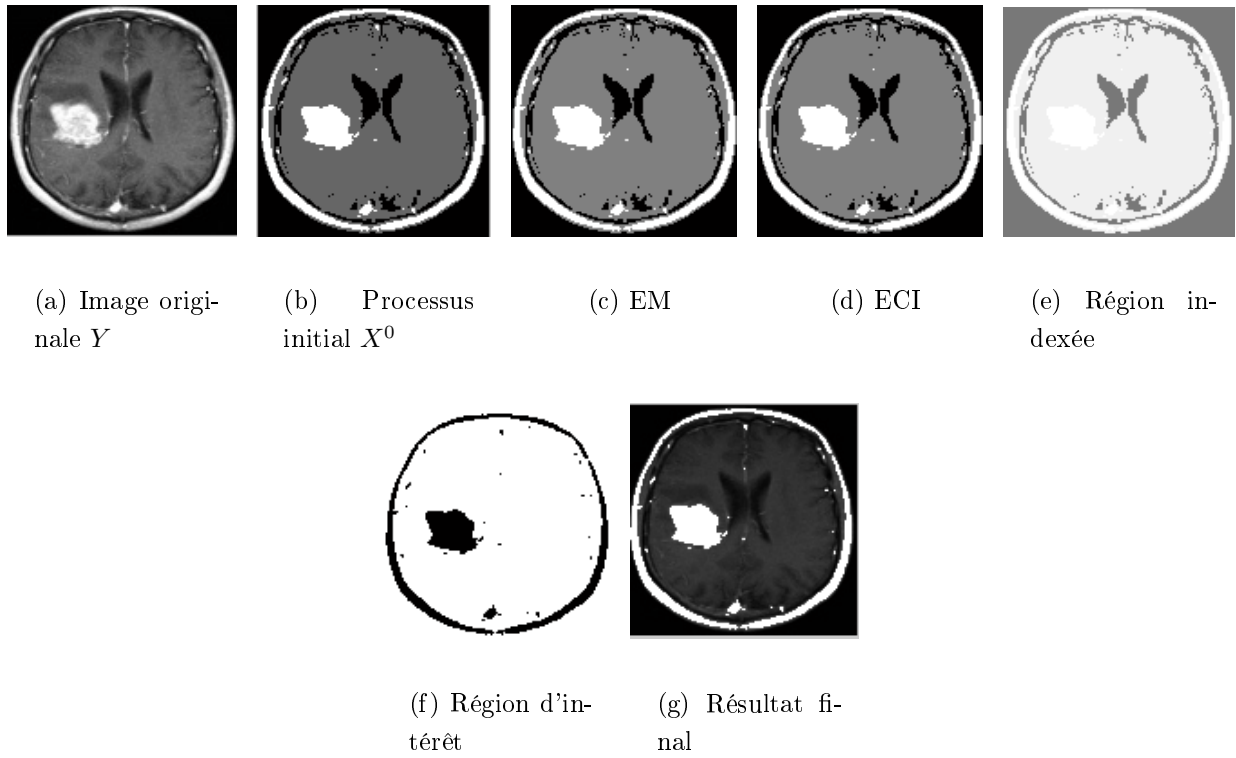


FIGURE 2.16 – Résultats de segmentation et extraction de la ROI expérience 9

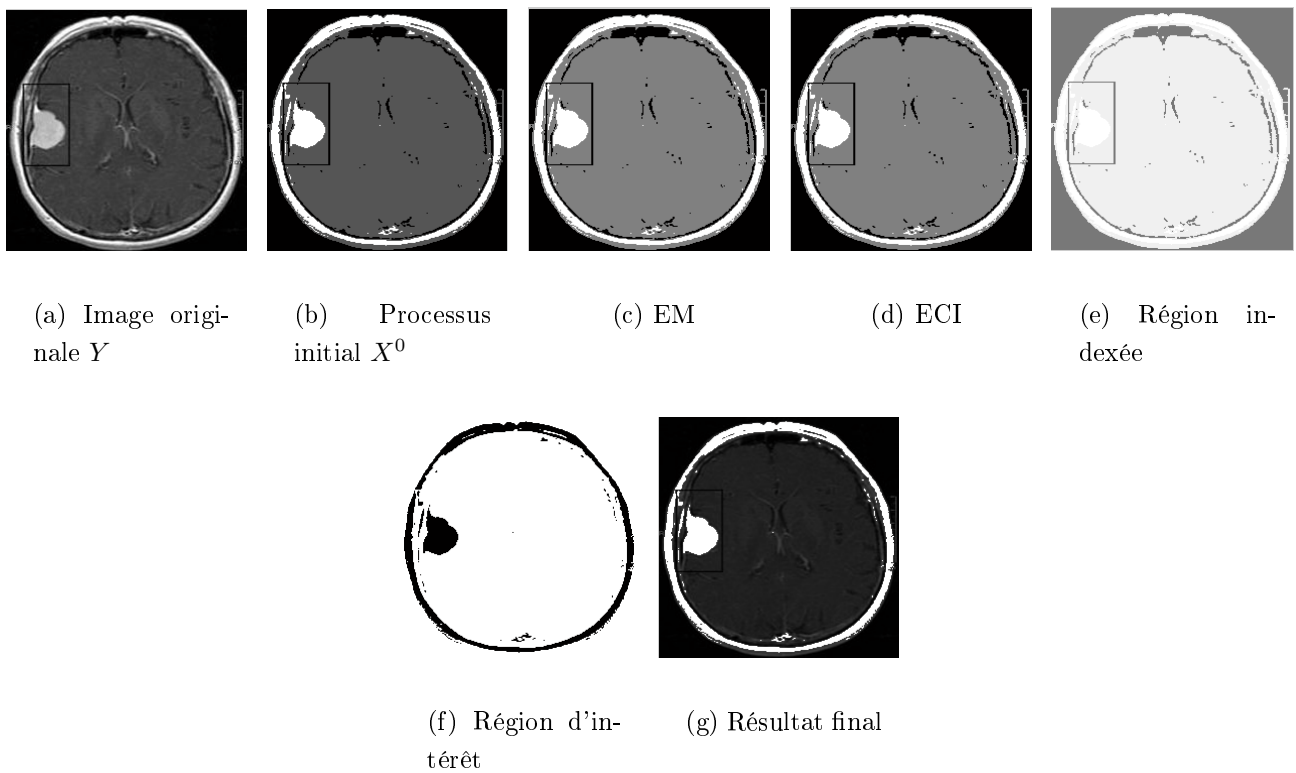


FIGURE 2.17 – Résultats de segmentation et extraction de la ROI expérience 10

A partir de ces figures, nous remarquons que : CMC-BI divise l'image en trois régions, parmi ces régions, nous trouverons les régions contenant la tumeur cérébrale. Visuellement, les méthodes ECI et EM capturent les mêmes détails de l'image réelle dans ces expériences.

● **Les résultats quantitatifs :**

Nous avons repris les résultats obtenus dans les tableaux suivants. Nous avons comparé ces estimateurs dans les dix expériences en termes : d'indice de PSNR, d'indice de SSIM(Structural Similarity Measure), de taux d'erreur et de convergence.

Expérience	PSNR EM	SSIM EM	PSNR ECI	SSIM ECI
Expérience 1	21,9500	0.5397	21,9594	0.5390
Expérience 2	24,0672	0.5697	24,0672	0.5710
Expérience 3	19,9322	0.4847	19,9323	0.4821
Expérience 4	22,1529	0.4977	22,1529	0.4990
Expérience 5	18.4713	0.4784	18.4713	0.4773
Expérience 6	21,8058	0.5150	21,8050	0.5157
Expérience 7	20,3738	0.3922	20,3738	0.3908
Expérience 8	19,0083	0.3506	19,0083	0.3488
Expérience 9	18,0631	0.3843	18,0636	0.3845
Expérience 10	21.7587	0.3572	21.7587	0.3574

TABLE 2.8 – Les valeurs des indices de PSNR et SSIM

Expérience	EM	ECI
Expérience 1	9,2127	9,2127
Expérience 2	8,1357	8,1357
Expérience 3	7,9766	7,9766
Expérience 4	11,4270	11,4270
Expérience 5	9.6075	9.6075
Expérience6	10,0450	10,0450
Expérience 7	9,4128	9,4128
Expérience 8	7,0975	7,0975
Expérience 9	13,0872	13,0872
Expérience 10	12.9558	12.9558

TABLE 2.9 – Les valeurs de taux d'erreur par estimateur

Expérience	EM	ECI
Expérience 1	8 itérations	7 itérations
Expérience 2	7 itérations	6 itérations
Expérience 3	12 itérations	9 itérations
Expérience 4	13 itérations	10 itérations
Expérience 5	8 itérations	6 itérations
Expérience 6	6 itérations	7 itérations
Expérience 7	11 itérations	9 itérations
Expérience 8	9 itérations	7 itérations
Expérience 9	8 itérations	9 itérations
Expérience 10	9 itérations	10 itérations

TABLE 2.10 – La convergence d’EM et ECI

À partir des tableaux 2.8, 2.9 et 2.10, nous remarquons que les valeurs de l’indice de PSNR, de l’indice de SSIM et du taux d’erreur obtenues dans chaque expérience par EM et ECI sont égaux. EM et ECI donnent les mêmes résultats dans toutes les expériences présentées. Malgré qu’ils utilisent des stratégies différentes pour estimer les paramètres. La qualité de la segmentation est comparable pour les deux algorithmes, pas de différence. Pour la convergence EM et ECI convergent rapidement. Mais, ECI est plus rapide qu’EM.

## 2.3 Les chaînes de Markov couples

Les Chaînes de Markov Couples (CMC Couple) en anglais (Pairwise Markov Chains) sont des modèles de Markov stationnaires qui considèrent que : les observations  $Y$  sont corrélées entre eux [104]. Dans cette section, nous présentons ce modèle, ses paramètres, l’estimation des paramètres avec EM, SEM et ECI adaptée à ce modèle, nous effectuons une étude comparative entre ces estimateurs sous le critère Bayésien MPM. Nous appliquons ces approches pour segmenter les images couleurs texturées de la base d’images VisTex [105].

### 2.3.1 Présentation de modèle

Soit le processus  $Z = (X, Y)$ , avec  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$  and  $X = (X_n)_{n=1}^N \in \Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ . Le processus  $Z$  est une chaîne de Markov couple, si et seulement si sa loi est la suivante :

$$p(z) = p(z_1) \cdot \prod_{n=1}^{N-1} p(z_{n+1} | z_n) \quad (2.35)$$

$$= p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}, y_{n+1} | x_n, y_n) \quad (2.36)$$

$$= p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n, y_n) \cdot p(y_{n+1}|x_n, y_n, x_{n+1}) \quad (2.37)$$

$$= p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|z_n) \cdot p(y_{n+1}|z_n, x_{n+1}) \quad (2.38)$$

$Z$  est une CM Couple (Voir [56, p. 25-27]) implique que :

- $X$  conditionnellement à  $Y$  est une chaîne de Markov.
- $Y$  conditionnellement à  $X$  est une chaîne de Markov.
- Les observations  $Y$  dépendent de  $X$ .

Après avoir représenté la probabilité  $p(y_{n+1}|x_n, y_n, x_{n+1})$  par la densité gaussienne  $f_{x_{n+1}, x_n}(y_{n+1}, y_n)$  dans la loi de  $Z$  nous obtenons :

$$p(z) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|z_n) \cdot f_{x_{n+1}, x_n}(y_{n+1}, y_n) \quad (2.39)$$

La CM Couple est un cas général de CMC-BI, et CMC-BI est un cas particulier de cette dernière. La démonstration suivante illustre que la CMC-BI est un cas particulier de CM Couple.

**Démonstration 2.3.1.** Soit  $X = (X_n)_{n=1}^N$  est une chaîne de Markov et  $Y = (Y_n)_{n=1}^N$  représente les observations.

Nous supposons que  $Z = (X_n, Y_n)_{n=1}^N$ .

Et :

$$p(z) = p(x, y) \quad (2.40)$$

$$= p(x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot \prod_{n=1}^N p(y_n|x_n) \quad (2.41)$$

$$= p(x_1) \cdot p(y_1|x_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \quad (2.42)$$

Selon la règle de Bayes, nous avons :

$$p(x_1, y_1) = p(x_1) \cdot p(y_1|x_1) \quad (2.43)$$

et

$$p(x_{n+1}, y_{n+1}|x_n, y_n) = p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \quad (2.44)$$

donc  $(X, Y)$  est une CMC-BI.

Nous avons :

$$p(z) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}, y_{n+1}|x_n, y_n) \quad (2.45)$$

$$= p(z_1) \cdot \prod_{n=1}^{N-1} p(z_{n+1}|z_n) \quad (2.46)$$

A partir de la démonstration  $Z$  est une CMCouple.

Donc CMCouple est le cas général de CMC-BI.

$$p(x_n|x_{n-1}, y_{n-1}) = p(x_n|x_{n-1}) \quad (2.47)$$

et

$$p(y_n|x_n, x_{n-1}, y_{n-1}) = p(y_n|x_n) \quad (2.48)$$

CMC Couple a les paramètres suivants  $\theta = (\theta_x, \theta_y)$  :

Les paramètres de processus caché  $\theta_x$  sont : La matrice de transition  $C_{i,j}$  de taille  $K * K$ ,  $\forall i, j \in \Omega$ .  
et les paramètres de processus observé  $\theta_y$  sont : La matrice de la moyenne  $\mu_{i,j}$  de taille  $K * K$ , la matrice de la covariance  $\Gamma_{i,j}$  de taille  $K * K$ ,  $\forall i, j \in \Omega$ .

### 2.3.2 Estimation des paramètres

Dans ce paragraphe, nous présentons les versions des algorithmes : Baum Welch, EM, SEM et ECI qui estiment les paramètres du modèle CMCouple. Certes que MPM donne des meilleurs résultats de segmentation avec un temps d'exécution meilleur que MAP, nous l'utilisons pour estimer l'image résultat de segmentation dans tout le reste de cette thèse.

#### a. L'algorithme de Baum-Welch

L'algorithme de Baum Welch sous CMCouple se déroule de la manière suivante :

**1-**Le calcul des probabilités de Forward  $\alpha_n(x_n) = p(y_1, \dots, y_n, x_n)$  qui se déroule de la manière suivante :

1. Initialisation  $n = 1$  :

$$\alpha_1(x_1) = p(z_1) \quad (2.49)$$

2. Induction for  $n > 1$  :

$$\alpha_n(x_n) = \sum_{x_{n-1} \in \Omega} \alpha_{n-1}(x_{n-1}) \cdot p(z_n|x_{n-1}) \quad (2.50)$$

**2-** Le calcul des probabilités de Backward  $\beta_n(x_n) = p(y_{n+1}, \dots, y_N|x_n, y_n)$  en deux étapes :

1. Initialisation  $n = N$  :

$$\beta_N(x_N) = 1 \quad (2.51)$$

2. Induction for  $n < N$  :

$$\beta_n(x_n) = \sum_{x_{n+1} \in \Omega} \beta_{n+1}(x_{n+1}) \cdot p(z_{n+1}|x_n) \quad (2.52)$$

**3-**Le calcul des probabilités conjointes *a posteriori*  $\gamma_n(x_n, x_{n+1}) = p(x_n, x_{n+1}|y)$

$$\gamma_n(x_n, x_{n+1}) = \alpha_n(x_n) \cdot p(z_{n+1}|x_n) \cdot \beta_{n+1}(x_{n+1}) \quad (2.53)$$

**4-**Le calcul des probabilité Marginales *a posteriori*  $\xi_n(x_n) = p(x_n|y)$

$$\xi_n(x_n) = \alpha_n(x_n) \cdot \beta_n(x_n) \quad (2.54)$$

## b. L'algorithme EM

---

**Algorithme 10** L'algorithme EM pour CMCouple

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (C_{i,j}^0, \mu_{i,j}^0, (\Gamma_{i,j}^0)^2, f_i^0)$

**SORTIES:**  $\theta^Q = (C_{i,j}^Q, \mu_{i,j}^Q, (\Gamma_{i,j}^Q)^2, f_i^Q)$

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: EM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  par :

$$C_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} \gamma_n^q(i, j)}{N-1} \quad (2.55)$$

- 4: EM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} y_{n:n+1} \gamma_n^q(i, j)}{\sum_{n=1}^{N-1} \gamma_n^q(i, j)} \quad (2.56)$$

$$\Gamma_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1} - \mu_{ij}^{q+1})^2 \gamma_n^q(i, j)}{\sum_{n=1}^{N-1} \gamma_n^q(i, j)} \quad (2.57)$$

5:  $q = q + 1$

6: **fin pour**

---

## c. L'algorithme SEM

L'algorithme de SEM se déroule de la manière suivante :

---

**Algorithme 11** L'algorithme SEM pour CM couple

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (C_{i,j}^0, \mu_{i,j}^0, (\Gamma_{i,j}^0)^2, f_i^0)$

**SORTIES:**  $\theta^Q = (C_{i,j}^Q, \mu_{i,j}^Q, (\Gamma_{i,j}^Q)^2, f_i^Q)$

1: **pour** chaque itération  $q \in Q$  **faire**

2: SEM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.

3: SEM simule l'échantillon  $X^q$  aléatoirement une seule fois  $\tau = 1$

4: SEM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  par :

$$C_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} \mathbf{1}[x_n = i, x_{n+1} = j]}{N-1}, \forall i, j \in \Omega \quad (2.58)$$

5: SEM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} y_{n:n+1} \mathbf{1}[x_n = i, x_{n+1} = j]}{\sum_{n=1}^{N-1} \mathbf{1}[x_n = i, x_{n+1} = j]}, \forall i, j \in \Omega \quad (2.59)$$

$$\Gamma_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1} - \mu_{ij}^{q+1})^2 \mathbf{1}[x_n = i, x_{n+1} = j]}{\sum_{n=1}^{N-1} \mathbf{1}[x_n = i, x_{n+1} = j]}, \forall i, j \in \Omega \quad (2.60)$$

6:  $q = q + 1$

7: **fin pour**

---

#### d. L'algorithme ECI

ECI se déroule de la manière suivante :



---

**Algorithme 12** L'algorithme ECI pour CMCCouple

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (C_{i,j}^0, \mu_{i,j}^0, (\Gamma_{i,j}^0)^2, f_i^0)$

**SORTIES:**  $\theta^Q = (C_{i,j}^Q, \mu_{i,j}^Q, (\Gamma_{i,j}^Q)^2, f_i^Q)$

1: **pour** chaque itération  $q \in Q$  **faire**

2: ECI calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.

3: ECI simule l'échantillon  $X^q$  aléatoirement une seule fois  $\tau = 1$

4: ECI estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  par :

$$C_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} \gamma_n^q(i, j)}{N-1}, \forall i, j \in \Omega \quad (2.61)$$

5: ECI estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} y_{n:n+1} \mathbf{1}[x_n = i, x_{n+1} = j]}{\sum_{n=1}^{N-1} \mathbf{1}[x_n = i, x_{n+1} = j]}, \forall i, j \in \Omega \quad (2.62)$$

$$\Gamma_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1} - \mu_{ij}^{q+1})^2 \mathbf{1}[x_n = i, x_{n+1} = j]}{\sum_{n=1}^{N-1} \mathbf{1}[x_n = i, x_{n+1} = j]}, \forall i, j \in \Omega \quad (2.63)$$

6:  $q = q + 1$

7: **fin pour**

---

## e. L'algorithme MPM

MPM pour les CMCCouples maximise les probabilités marginales *a posteriori*, pour estimer la classe d'appartenance de chaque pixel en utilisant l'équation 2.30.

### 2.3.3 Résultats et expérimentations

Dans cette section, nous évaluons les performances des deux modèles utilisés CMC-BI et CM-Couple. Nous avons réalisé quelques expériences où nous avons segmenté quelques images couleurs de la base d'images VisTex de taille  $(128 * 128)$ . Nous avons comparé ces modèles en terme de taux d'erreur  $\tau$ . Nous avons initialisé la configuration initiale  $X^0$  par l'algorithme de K-means. Nous avons transformé l'image  $Y$  2D en 1D en utilisant le parcours ligne par ligne. Nous avons initialisé les paramètres  $\theta_x^0$  de la loi *a priori* par :

Pour le modèle CMC-BI, nous avons suivi la même démarche des expériences précédentes de la section 2.2.3.

Pour le modèle CMCouple, nous avons suivi cette démarche :

$$C_{ij}^0 = \begin{cases} C_{ij}^0 = 0.45 & \text{si } i = j \\ C_{ij}^0 = 0.05 & \text{sinon} \end{cases} \quad \forall i, j \in \Omega \quad (2.64)$$

Les paramètres de la loi des observations  $\theta_y^0$  sont initialisés par :

Pour chaque niveau de couleur (*Rouge*, *Vert*, *Bleu*) nous avons calculé :

$$\mu_{ij}^0 = \frac{\sum_{n=2}^N y_{n-1:n} \cdot 1[x_n = j, x_{n-1} = i]}{N} \quad \forall i, j \in \Omega \quad (2.65)$$

$$\Gamma_{ij}^0 = \frac{\sum_{n=2}^N (y_{n-1:n} - \mu_{ij}^0)^2 \cdot 1[x_n = j, x_{n-1} = i]}{N} \quad (2.66)$$

### a. Résultats visuels

Nous avons réalisé trois expériences pour trois images couleurs texturées. Les figures suivantes montrent les différents résultats obtenus avec, le nombre de classes d'appartenance  $K$ .



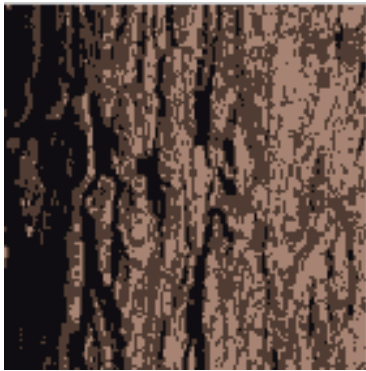
(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



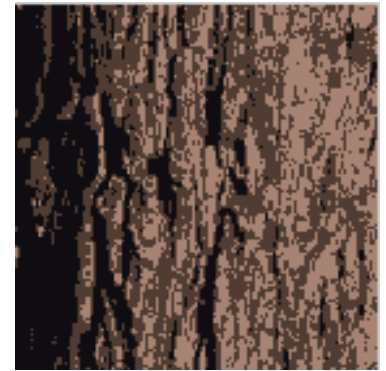
(c) CMC-BI+EM



(d) CMCouple+EM



(e) CMC-BI+SEM



(f) CMCouple+SEM



(g) CMC-BI+ECI

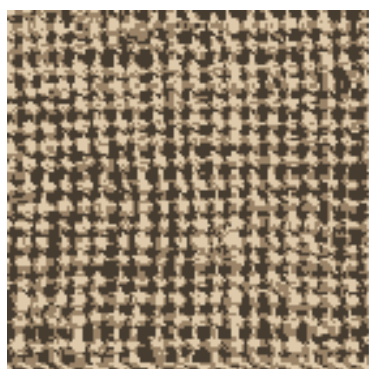


(h) CMCouple+ECI

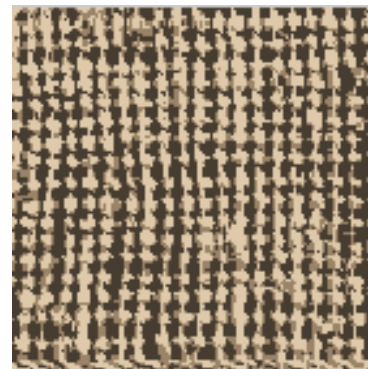
FIGURE 2.18 – Résultats de segmentation d'une image couleur texturée 1 avec  $K = 2$



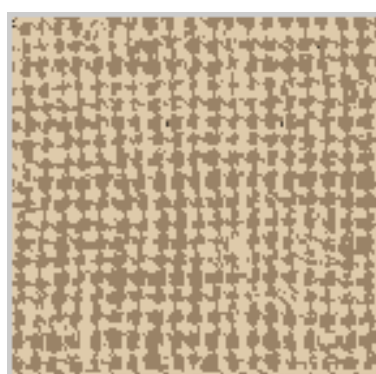
(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



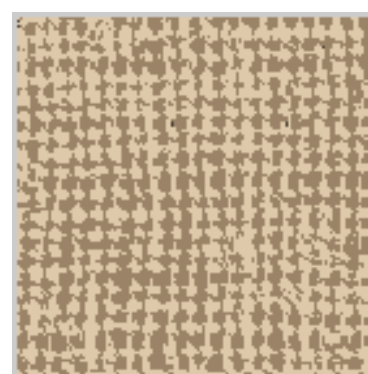
(c) CMC-BI+EM



(d) CMCouple+EM



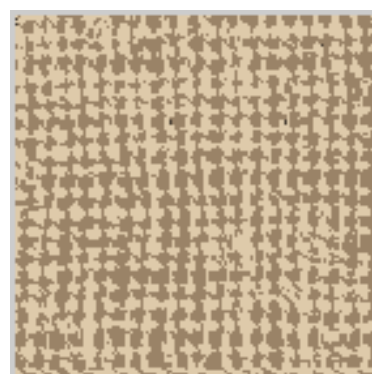
(e) CMC-BI+SEM



(f) CMCouple+SEM

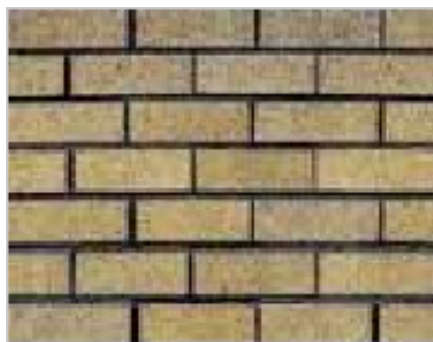


(g) CMC-BI+ECI



(h) CMCouple+ECI

FIGURE 2.19 – Résultats de segmentation d'une image couleur texturée 2 avec  $K = 2$



(a) Image originale  $Y$



(b) Configuration initiale  $X^0$



(c) CMC-BI+EM



(d) CMCouple+EM



(e) CMC-BI+SEM



(f) CMCouple+SEM



(g) CMC-BI+ECI



(h) CMCouple+ECI

FIGURE 2.20 – Résultats de segmentation d'une image couleur texturée 3 avec  $K = 2$

Comme montre ces premières comparaisons visuelles, nous remarquons que toutes les textures segmentées par le modèle CMCouple sont bien segmentées que celles utilisant le modèle classique CMC-BI, dont il ne tient pas compte à l'information bruit, le bruit apparaissant dans les textures segmentées par ce modèle. Dans les textures segmentées par le modèle CMCouple l'information bruit a bien disparu.

## b. Résultats quantitatifs

Le tableau suivant présente les valeurs de taux d'erreur  $\tau$  obtenues dans chaque expérience, par algorithme (modèle+estimateur). D'après le tableau 2.11, nous constatons que : CMCouple est plus

$\tau$	e1	e2	e3
CMC-BI+EM	2.0213	2.3435	3.5991
CMC Couple+EM	1.5755	1.8656	1.1911
CMC-BI+SEM	2.0213	2.3435	3.5991
CMC Couple+SEM	1.5755	1.8656	1.1911
CMC-BI+ECI	2.0213	2.3435	3.5991
CMC Couple+ECI	1.5755	1.8656	1.1911

TABLE 2.11 – Les valeurs de taux d'erreur

puissant que CMC-BI, surtout, lorsque l'image d'entrée est bruitée comme dans la troisième expérience. CMCouple tient en compte la corrélation entre les observations. Par conséquent, CMCouple modélise la texture et le bruit mieux que le modèle classique CMC-BI. Nous pouvons confirmer que la corrélation entre les observations a une influence sur la qualité de la segmentation, elle améliore cette qualité. En bref, CMCouple améliore les résultats de la segmentation non supervisée obtenus par CMC-BI.

Les algorithmes EM, SEM et ECI sous CMC-BI et CMCouple donnent les mêmes résultats de segmentation, même s'ils utilisent des stratégies d'estimation différentes .

## 2.4 Étude Comparative entre les chaînes de Markov cachées stationnaires

Dans ce paragraphe, nous présentons une étude comparative entre les chaînes de Markov stationnaires, nous comparons quatre modèles de Markov. Les Chaînes de Markov Cachées à Bruit Indépendant(CMC-BI) présentées dans la section 2.2, les Chaînes de Markov Couples(CMCouples) que nous avons présenté dans la section 2.3. ainsi que, les Chaînes de Markov Cachées(CMC) et les Chaînes de Markov Couples à Bruit Indépendant(CMCouple-BI) (Voir [96, p. 25-27]). Pour estimer les paramètres des modèles, nous avons utilisé l'algorithme EM et MPM pour estimer la

configuration finale de l'image résultat. Nous utilisons ces approches pour segmenter des images niveau de gris, couleur, et bruitées.

### 2.4.1 Présentation des modèles

Un modèle de Markov caché est stationnaire, si son processus caché  $X$  est stationnaire.

Le processus  $X$  est stationnaire si  $p(x_{n+1}|x_n)$  ne dépend pas de  $n$ .

Certes que nous avons présenté déjà les deux modèles de Markov stationnaires : CMC-BI et CM-Couple. Nous présentons ici juste les deux autres modèles de Markov stationnaires utilisés dans cette compétition CMC et CM-Couple-BI.

#### a. Chaîne de Markov cachée

Dans le modèle CMC-BI, nous supposons que la probabilité  $p(y_{n+1}|x_{n+1}, x_n)$  de l'observation  $y_n$  dépend uniquement de son état caché  $x_n$ , et que les observations ne sont pas corrélées entre eux, pour cela, il est appelé par Chaîne de Markov Cachée avec Bruit Indépendant.

La Chaînes de Markov cachée en anglais (Hidden Markov Chain) ont les mêmes propriétés que CMC-BI, sauf qu'elle considère que les observations  $Y$  sont corrélées entre eux. Elle conserve l'hypothèse de : la probabilité  $p(y_{n+1}|x_{n+1}, x_n)$  de l'observation  $y_{n+1}$  ne dépend que de son état caché  $x_{n+1}$ . elle représente la corrélation entre les observations, avec un nombre de paramètres réduit.

Soit le processus  $Z = (X, Y)$ , avec  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$  et  $X = (X_n)_{n=1}^N \in \Omega$ .

$Z$  est une CMC, elle garde les mêmes propriétés que CMC-BI sauf que :

-Les observations  $Y$  sont corrélées entre eux.

La loi de ce modèle est :

$$p(z) = p(x, y) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot p(y_{n+1}|y_n, x_{n+1}) \quad (2.67)$$

si on remplace la probabilité  $p(y_{n+1}|y_n, x_{n+1})$  par la densité  $f_{x_n}(y_{n+1}, y_n)$  on trouve :

$$p(z) = p(x, y) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \cdot f_{x_{n+1}}(y_{n+1}, y_n) \quad (2.68)$$

CMC a les paramètres suivants  $\theta$  : Les paramètres  $\theta_x$  de la chaîne de Markov  $X$  : la loi initiale  $PI$  de taille  $K$  et la matrice de transition  $A$  de taille  $K^2$ .

Les paramètres  $\theta_y$  du processus observé : la moyenne  $\mu$ , la variance  $\sigma^2$  et la corrélation  $r$  de taille  $K$ . CMC estime  $K^2 + 4K$  paramètres. CMC-BI estime  $K^2 + 3K$  paramètres.

#### b. Chaîne de Markov Couple à Bruit Indépendant

La chaîne de Markov couple est le modèle de Markov stationnaire le plus général, dont le processus caché  $X$  est stationnaire et pas nécessairement markovien, dans ce modèle les observations

$Y$  sont corrélées entre eux, et l'observation  $y_{n+1}$  dépend de  $x_n$  et  $x_{n+1}$   $p(y_{n+1}|x_{n+1}, x_n)$  en même temps.

La Chaîne de Markov Couple à Bruit Indépendant en anglais (Pairwise Markov Chain with Independent Noise) suppose que les observations  $Y$  ne sont pas corrélées entre eux. Et que  $p(y_{n+1}|x_{n+1}, x_n)$   $y_n$  dépend de  $x_{n+1}$  et de  $x_n$  en même temps, le processus caché  $X$  de ce modèle est considéré comme une chaîne de Markov d'ordre 2. Les observations  $Y$  sont conditionnellement indépendantes de  $X$ . Soit le processus  $Z = (X, Y)$ , où le processus caché  $X = (X_n)_{n=1}^N \in \Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$  et le processus observé  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$ .

$Z$  est une CM couple-BI a les mêmes propriétés que la CM couple sauf que :

- $Y$  sont conditionnellement indépendantes de  $X$
- Les observations  $Y$  ne sont pas corrélées entre eux.

La loi de ce modèle s'écrit comme suit :

$$p(z) = p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|x_n, y_n) \cdot p(y_{n+1}|x_{n+1}, x_n) \quad (2.69)$$

$$= p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|z_n) \cdot p(y_{n+1}|x_{n+1}, x_n) \quad (2.70)$$

si on remplace la probabilité  $p(y_{n+1}|x_{n+1}, x_n)$  par la densité  $f_{x_{n+1}, x_n}(y_{n+1})$  on obtient :

$$= p(x_1, y_1) \cdot \prod_{n=1}^{N-1} p(x_{n+1}|z_n) \cdot f_{x_{n+1}, x_n}(y_{n+1}) \quad (2.71)$$

Les paramètres de CM couple-BI sont les suivants :

- Les paramètres de la loi *a priori*  $\theta_x$  sont : La matrice de transition  $C_{i,j}$ .
- Les paramètres de la loi des observations  $\theta_y$  sont : La matrice de la moyenne  $\mu_{i,j}$ , La matrice de la variance  $\sigma_{i,j}^2$ .

CM couple-BI estime  $3K^2$  et CM couple estime  $3K^2$ .

## 2.4.2 Estimation des paramètres

Dans ce paragraphe, Nous présentons les démarches de l'algorithme EM utilisé pour estimer les paramètres de chaque modèle utilisé dans cette comparaison.

Nous avons présenté les versions de l'algorithme EM pour les modèles CMC et CM couple-BI. Pour les versions d'EM pour les modèles CMC-BI et CM couple sont présentées dans 7 et 10.

### a. L'algorithme EM pour CMC

L'algorithme EM pour CMC a la même démarche qu'EM pour CMC-BI. Nous ajoutons juste le calcul de la corrélation. il se déroule de la manière suivante :



---

**Algorithme 13** L'algorithme EM pour CMC

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (PI^0, A^0, \mu_i^0, (\sigma_i^0)^2, r_i^0, f_i^0)$

**SORTIES:**  $\theta^Q = (PI^Q, A^Q, \mu_i^Q, (\sigma_i^Q)^2, r_i^Q, f_i^Q)$

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: EM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  par :

$$PI^{q+1}(i) = \xi_1^q(i) \quad (2.72)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.73)$$

- 4: EM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu_i^{q+1} = \frac{\sum_{n=1}^N y_n \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.74)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^{q+1})^2 \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \quad (2.75)$$

$$r^{q+1}(i) = \frac{\sum_{n=1}^{N-1} y_{n:n+1} \xi_n^q(i)}{(\sigma_i^{q+1})^2 \sum_{n=1}^{N-1} \xi_n^q(i)} \quad (2.76)$$

5:  $q = q + 1$

6: **fin pour**

---

**b. L'algorithme EM pour CMCCouple-BI**

L'algorithme EM pour CMCCouple-BI se déroule de la manière suivante :

---

**Algorithme 14** L'algorithme EM pour CMCouple-BI

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (C_{i,j}^0, \mu_{i,j}^0, (\Gamma_{i,j}^0)^2, f_i^0)$ **SORTIES:**  $\theta^Q = (C_{i,j}^Q, \mu_{i,j}^Q, (\Gamma_{i,j}^Q)^2, f_i^Q)$ 

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: EM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X$  par :

$$C_{ij}^{q+1} == \frac{\sum_{n=1}^{N-1} \gamma_n^q(i, j)}{N-1} \quad (2.77)$$

- 4: EM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu_{ij}^{q+1} = \frac{\sum_{n=1}^{N-1} y_n \gamma_n^q(i, j)}{\sum_{n=1}^{N-1} \gamma_n^q(i, j)} \quad (2.78)$$

$$(\sigma_{ij}^{q+1})^2 = \frac{\sum_{n=1}^{N-1} (y_n - \mu_{ij}^{q+1})^2 \gamma_n^q(i, j)}{\sum_{n=1}^{N-1} \gamma_n^q(i, j)} \quad (2.79)$$

- 5:  $q = q + 1$

- 6: **fin pour**

---

### 2.4.3 Résultats et expérimentations

Cette section montre les différents résultats de segmentation d'images obtenus en utilisant les modèles de Markov stationnaires, nous avons segmenté trois types d'images : une images niveau de gris, une image couleur et une image niveau de gris bruitées. Pour évaluer les performances de ces modèles, nous avons calculé les deux critères d'évaluation : L'indice de PSNR et le taux d'erreur. Les sous-sections suivantes montrent les résultats de segmentation, ainsi que, les valeurs obtenues de l'indice de PSNR et de taux d'erreur.

Pour obtenir la configuration initiale de  $X^0$  nous avons utilisé l'algorithme de K-means.

Les paramètres de la loi *a priori*  $\theta_x^0$  de CMC-BI et de CMC sont initialisés par les equations 2.31 et 2.32 respectivement.

Les paramètres de la loi *a priori*  $\theta_x^0$  de CMCouple et CMCouple-BI sont initialisés par :

$$C_{ij}^0 = \begin{cases} C_{ij} = 0.49 & \text{si } i = j \\ C_{ij} = 0.1 & \text{sinon} \end{cases} \quad (2.80)$$

La moyenne  $\mu^0$  et la variance  $(\sigma^0)^2$  du CMC-BI et CMC sont calculées par 2.33 et 2.34.

La corrélation du modèle CMC  $r^0$  est initialisée par :

$$r^0(i) = \frac{\sum_{n=1}^{N-1} y_{n:n+1} \cdot 1[x_n^0 = i]}{(\sigma^0(i))^2 \cdot N} \quad (2.81)$$

Les paramètres de la loi du processus observé du modèle CMCCouple  $\theta_y^0$  sont initialisés à l'aide des équations 2.65 et 2.66.

Les paramètres  $\theta_y^0$  du modèle CMCCouple-BI sont initialisés par :

$$\mu_{ij}^0 = \frac{\sum_{n=1}^{N-1} y_n \cdot 1[x_{n+1}^0 = j, x_n^0 = i]}{N} \quad (2.82)$$

$$(\sigma_{ij}^0)^2 = \frac{\sum_{n=1}^{N-1} (y_n - \mu_{ij}^0)^2 \cdot 1[x_{n+1}^0 = j, x_n^0 = i]}{N} \quad (2.83)$$

Cette procédure d'initialisation des paramètres  $\theta^0 = (\theta_x^0, \theta_y^0)$  est utilisée dans toutes les expériences présentées dans cette étude. Les sous-sections suivantes montrent les différents résultats de la segmentation de chaque modèle, nous présentons les résultats de la segmentation en trois figures : la figure 2.21 décrit les résultats de la segmentation d'une image niveau de gris, la figure 2.22 représente les résultats de la segmentation d'une image couleur et la figure 2.23 décrit les résultats de la segmentation d'une image niveau de gris bruitée. Pour confirmer les résultats visuels obtenus, nous présentons sous chaque figure, un tableau de valeurs de PSNR et taux d'erreur.

a. Les résultats visuels et quantitatifs 1 :

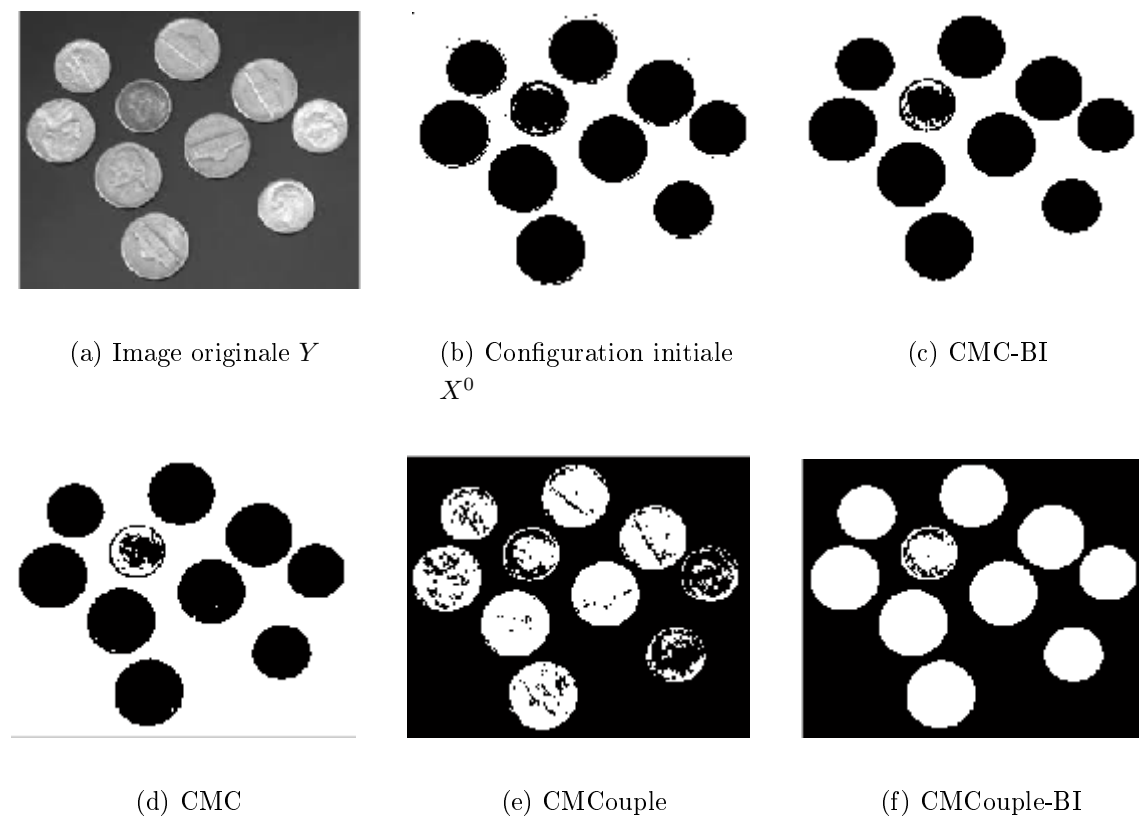


FIGURE 2.21 – Résultats de segmentation d'une image niveau de gris  $K = 2$  .

Modèle	PSNR	taux d'erreur
CMC-BI	25,1119	11,1124
CMC	25,3334	9,8343
CMCouple	26,5531	4,6649
CMCouple-BI	25,3390	9,8354

TABLE 2.12 – Les valeurs de PSNR et taux d'erreur

D'après les résultats visuels et le tableau 2.12, nous remarquons que CMCouple segmente bien l'image niveau de gris, et que CMC et CMCouple-BI donnent les mêmes résultats. CMCouple prend en compte les contours des objets qui composent l'image originale.

b. Les résultats visuels et quantitatifs 2 :

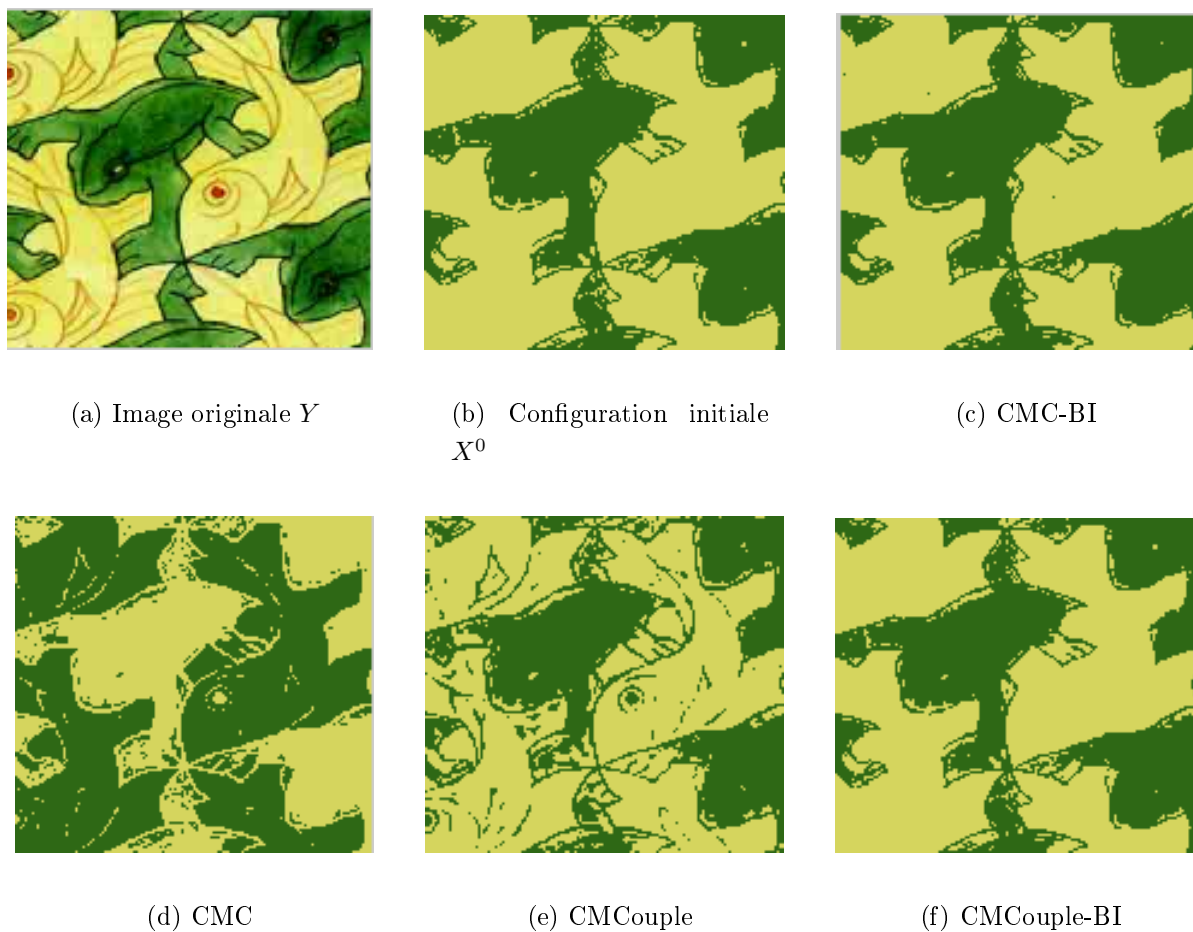


FIGURE 2.22 – Résultats de segmentation d'une image couleur  $K = 2$ .

Modèle	PSNR	taux d'erreur
CMC-BI	26,9661	10,5523
CMC	28,9845	6,1245
CMCouple	29,1254	5,8275
CMCouple-BI	26,9689	10,3523

TABLE 2.13 – Les valeurs de PSNR et taux d'erreur

A partir des valeurs de PSNR et de taux d'erreur et la figure 2.22, nous remarquons que CMCouple fournit le meilleur résultat, suivi par CMC qui donne également un meilleur résultat, ici CMCouple et CMC prennent en compte les contours des objets.

c. Les résultats visuels et quantitatifs 3 :

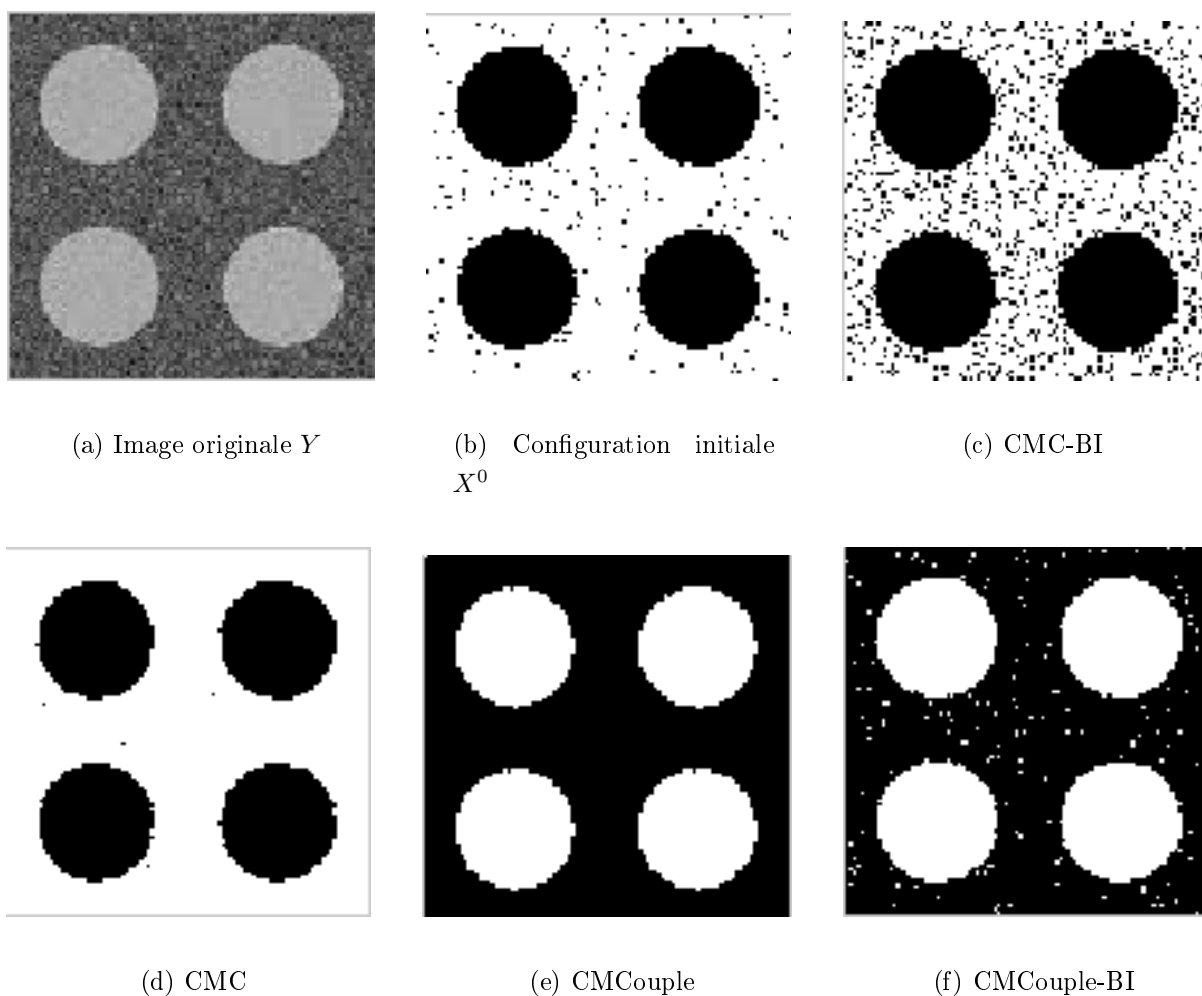


FIGURE 2.23 – Résultats de segmentation d’une image niveau de gris bruitée  $K = 2$  .

Modèle	PSNR	taux d’erreur
CMC-BI	12,4578	21,5623
CMC	24,5816	1,3063
CMCouple	26,4587	0,082
CMCouple-BI	14,5816	11,5663

TABLE 2.14 – Les valeurs de PSNR et taux d’erreur

À partir des résultats obtenus, CMCouple segmente bien l’image bruitée, elle la restaure et segmente, aussi CMC a bien segmenté cette image. CMCouple-BI, CMC-BI et K-means ne prennent pas en compte l’information bruit dans l’image à segmenter. Ils sont segmentés avec bruit. Les résultats obtenus et les valeurs de PSNR et de taux d’erreur montrent que, les modèles CMCouple et CMC fournissent des meilleurs résultats de segmentation, surtout lorsqu’on a des images

bruitées, l'information bruit dans l'image a totalement disparu de 99% avec CM Couple et de 80% avec CMC. Nous pouvons expliquer cela par les raisons suivantes : CM Couple et CMC supposent que les observations sont corrélées entre eux. La corrélation entre observations améliore la qualité de la segmentation en tenant compte aux bruit, texture et au contours des objets. Contrairement aux modèles markoviens à bruit indépendant, ces modèles ne donnent des meilleurs résultats qu'avec les images filtrées.

## 2.5 Les chaines de Markov triplets

Les Chaines de Markov Triplets(CMT) en anglais (Triplet Markov Chains) servent à traiter les problèmes de la non-stationnarité de données à traiter. Pour modéliser la non-stationnarité du processus caché  $X$ . Les CMTs introduisent un processus auxiliaire  $U$ , le rôle de ce processus est : modéliser les stationnarités du processus caché  $X$ . Nous utilisons ce modèles pour segmenter des images texturées niveau de gris de la base d'images Brodatz [107] et des images couleurs texturées de la base d'images Vis Tex [105]. Pour estimer les paramètres, nous avons utilisé l'algorithme EM et MPM.

### 2.5.1 Présentation de modèle

L'utilité du modèle CMT : est de modéliser la non stationnarité du processus  $X$  dans ce cas, ce modèle s'appelle Chaîne de Markov M-Non Stationnaire à Bruit Indépendant(CMMNS-BI) [106]. Ce modèle estime deux processus :le processus caché  $X$  et ses stationnarités  $U$ .

Soit le processus  $T = (X, U, Y)$ , où le processus observé  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$ , le processus auxiliaire  $U = (U_n)_{n=1}^N \in \Lambda = \{\lambda_1, \dots, \lambda_M\}$ , et le processus caché  $X = (X_n)_{n=1}^N \in \Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ , Nous supposons que le processus  $V = (X, U)$ , où  $V$  comprend les deux processus  $X$  et  $U$  en même temps. Le processus  $T$  est une Chaîne de Markov M-Non Stationnaire à Bruit Indépendant, si et seulement, si elle vérifie ces conditions :

- Le processus  $V = (X, Y)$  est une chaîne de Markov stationnaire (Voir [96, p. 45-47]), sa loi est :

$$P(V) = P(V_1 = v_1) \prod_{n=1}^{N-1} P(V_{n+1} = v_{n+1} | V_n = v_n) \quad (2.84)$$

- La transition  $P(X_{n+1} = x_{n+1} | X_n = x_n, U_{n+1} = u_{n+1})$  dépend seulement de son état stationnaire  $u_{n+1}$ .

-  $Y$  sont conditionnellement indépendantes de  $X$ .

- L'observation  $y_n$  ne dépend que de sa classe cachée  $x_n$ .

Les paramètres de ce modèle sont définis par : La matrice de transition entre classes par stationnarité  $V(i, j) \forall i, j \in \Omega * \Lambda$  de taille  $(K * M) * (K * M)$ . La moyenne  $\mu(i) \forall i \in \Omega$  et la variance  $(\sigma^2(i))$  de la taille  $K \forall i \in \Omega$ .

## 2.5.2 Estimation des paramètres

Ce paragraphe expose la procédure de chaque algorithme utilisé pour calculer les paramètres du modèle CMT : l'algorithme BaumWelch, l'algorithme EM et l'algorithme MPM. Nous présentons les versions de ces algorithmes pour le modèle CMT.

### a. BaumWelch pour CMT

BaumWelch se déroule de la manière suivante : **1-**L'algorithme de Forward  $\alpha_n(v_n) = p(y_1, \dots, y_n, v_n)$  est calculé en deux étapes :

1. Initialisation :  $n = 1$

$$\alpha_1(v_1) = p(t_1) \quad (2.85)$$

2. Induction :  $n > 1$

$$\alpha_n(v_n) = \sum_{x_{n-1}} \alpha_{n-1}(v_{n-1}) \cdot p(t_n | t_{n-1}) \quad (2.86)$$

**2-** L'algorithme de Backward  $\beta_n(v_n) = p(y_{n+1}, \dots, y_N | v_n)$  est calculé en deux étapes :

1. Initialisation :  $n=N$

$$\beta_N(v_N) = 1 \quad (2.87)$$

2. Induction :  $n < N$

$$\beta_n(v_n) = \sum_{x_{n+1}} \beta_{n+1}(v_{n+1}) \cdot p(t_{n+1} | t_n) \quad (2.88)$$

**3-** En outre, les probabilités de la loi conjointe *a posteriori*  $\gamma_n(v_n, v_{n+1}) = p(v_n, v_{n+1} | y)$  sont estimées par :

$$\gamma_n(v_n, v_{n+1}) = \alpha_n(v_n) \cdot p(t_n | t_{n-1}) \cdot \beta_{n+1}(v_{n+1}) \forall n \in N \quad (2.89)$$

**4-** Enfin, les probabilités de la loi marginale *a posteriori*  $\xi_n(v_n) = p(v_n | y)$  sont estimées par :

$$\xi_n(v_n) = \alpha_n(v_n) \cdot \beta_n(v_n) \forall n \in N \quad (2.90)$$

$$\xi_n(x_n) = p(x_n | y) = \sum_{u_n \in \Lambda} \xi_n(v_n) \forall n \in N \quad (2.91)$$

$$\xi_n(u_n) = p(u_n | y) = \sum_{x_n \in \Omega} \xi_n(v_n) \forall n \in N \quad (2.92)$$

### b. L'algorithme EM

La procédure pour estimer les paramètres  $\theta = (V, \mu, \sigma^2)$  du modèle CMT avec EM [108], est la même que celle du CMC-BI.



---

**Algorithme 15** L'algorithme EM pour CMT

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (V^0, \mu_i^0, (\sigma_i^0)^2, f_i^0)$ **SORTIES:**  $\theta^Q = (V^Q, \mu_i^Q, (\sigma_i^Q)^2, f_i^Q)$ 

- 1: **pour** chaque itération  $q \in Q$  **faire**
- 2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch.
- 3: EM estime les paramètres  $\theta_v^{q+1}$  du processus caché  $V$  par :

$$V^{q+1}(i, j) = \frac{\sum_{n=1}^{N-1} \gamma_n^q(v_n = i, v_{n+1} = j)}{\sum_{n=1}^{N-1} \xi_n^q(v_n = i)} \forall i, j \in \Omega * \Lambda \quad (2.93)$$

- 4: EM estime les paramètres  $\theta_y^{q+1}$  du processus observé  $Y$  par :

$$\mu^{q+1}(k) = \frac{\sum_{n=1}^{N-1} \sum_{u_n} y_n \cdot \xi_n^q(x_n = k, u_n)}{\sum_{n=1}^{N-1} \sum_{u_n} \xi_n^q(x_n = k, u_n)} \forall k \in \Omega \quad (2.94)$$

$$(\sigma^{q+1}(k))^2 = \frac{\sum_{n=1}^{N-1} \sum_{u_n} (y_n - \mu^{q+1}(k))^2 \cdot \xi_n^q(x_n = k, u_n)}{\sum_{n=1}^{N-1} \sum_{u_n} \xi_n^q(x_n = k, u_n)} \forall k \in \Omega \quad (2.95)$$

5:  $q = q + 1$ 6: **fin pour**

---

**c. L'algorithme MPM**

MPM utilise les formules mathématiques suivantes pour estimer les configurations finales du processus caché  $\bar{X}$  et auxiliaire  $\bar{U}$  :

$$\bar{u}_{nmpm} = \mathbf{argmax}_{\lambda_m \in \Lambda} ((\xi_n^Q(u_n = \lambda_m))) \forall n \in N \quad (2.96)$$

$$\bar{x}_{nmpm} = \mathbf{argmax}_{\omega_i \in \Omega} ((\xi_n^Q(x_n = \omega_i))) \forall n \in N \quad (2.97)$$

**2.5.3 Résultats et expérimentations**

Dans ce paragraphe, nous montrons les différents résultats expérimentaux réalisés, nous avons segmenté des images niveaux de gris et couleurs en utilisant les modèles CMT et CMC-BI. Nous avons effectué une étude comparative entre ces deux modèles en terme de qualité de segmentation : l'indice de PSNR, l'indice de SSIM et le taux d'erreur.

Pour initialiser les paramètres de chaque modèle utilisé dans cette étude, nous suivons cette procédure :

Nous avons programmé l'algorithme suivant pour initialiser la matrice de transition  $V^0(i, j)$  du modèle CMT :

---

**Algorithme 16** Initialisation de  $V^0(i, j)$ 

---

**ENTRÉES:**  $K$  : nombre de classes d'appartenance.    24: **fin tantque**  
**ENTRÉES:**  $M$  : nombre de stationnarités.            25: **sinon**  
**SORTIES:** matrice de transition  $V^0$                 26: **tantque** ( $d \leq M$ ) **faire**  
1:  $l = 1$ ,  $d = 1$                                         27:     **pour** ( $i = ((d-1)*K+1); i = (d*K); i++$ )  
2: **si** ( $K == 2$ ) **alors**                                28:         **faire**  
3:     **tantque** ( $d \leq M$ ) **faire**                        29:             **pour** ( $j = 1; j = (K * M); j++$ ) **faire**  
4:         **pour** ( $i = ((d-1)*K+1); i = (d*K); i++$ )    30:                 **si** ( $\text{mod}(d, 2) == 0$ ) **alors**  
           **faire**                                        31:                      $N = 1./K$ ;  
5:             **pour** ( $j = 1; j = (K * M); j++$ ) **faire**    32:                     **sinon**  
6:                 **si** ( $\text{mod}(d, 2) == 0$ ) **alors**        33:                         **si** ( $i == j$ ) **alors**  
7:                      $N = 1./K$ ;  
8:                     **sinon**                            34:                          $N = 1./2$ ;  
9:                         **si** ( $i == j$ ) **alors**        35:                             **sinon**  
10:                              $N = 0.8$ ;  
11:                             **sinon**                    36:                                 **fin si**  
12:                                  $N = 0.2$ ;  
13:                                 **fin si**  
14:                                 **fin si**                37:                                 **fin si**  
15:                                 **si** ( $(j \geq ((l-1) * K + 1)) \text{et} (j \leq$  38:                                 **si** ( $j \geq ((l-1)*K+1) \text{et} (j \leq l*K)$ )  
   $l * K)$ ) **alors**                                 **alors**  
16:     $V(i, j) = N$ ;  
17:    **sinon**                                 39:     $V(i, j) = N$ ;  
18:     $V(i, j) = 0$ ;  
19:    **fin si**                                 40:    **sinon**  
20:    **fin pour**                                 41:     $V(i, j) = 0$ ;  
21:    **fin pour**                                 42:    **fin si**  
22:     $d = d + 1$ ;  
23:     $l = l + 1$ ;  
  43:    **fin pour**  
  44:    **fin pour**  
  45:     $d = d + 1$ ;  
  46:     $l = l + 1$ ;  
  47:    **fin tantque**  
  48: **fin si**

---

Nous avons appliqué K-means pour estimer le processus initial caché  $X^0$ . Ensuite, nous avons calculé la moyenne et la variance à partir de cette initialisation. en utilisant les equations 2.33 et 2.34.

Les paramètres  $\theta_x^0$  du modèle CMC-BI sont initialisés par les algorithmes suivants : Les algorithmes 17 et 18 décrivent la procédure d'initialisation .

---

**Algorithme 17** Initialisation de  $PI^0$ 

---

**ENTRÉES:**  $K$  : nombre de classes d'appartenance**SORTIES:** La loi initiale  $PI^0(i)$ 

- 1: **pour** ( $i = 1; i = K; i ++$ ) **faire**
  - 2:      $PI^0(i) = 1./K$
  - 3: **fin pour**
- 

---

**Algorithme 18** Initialisation de  $A^0(i, j)$ 

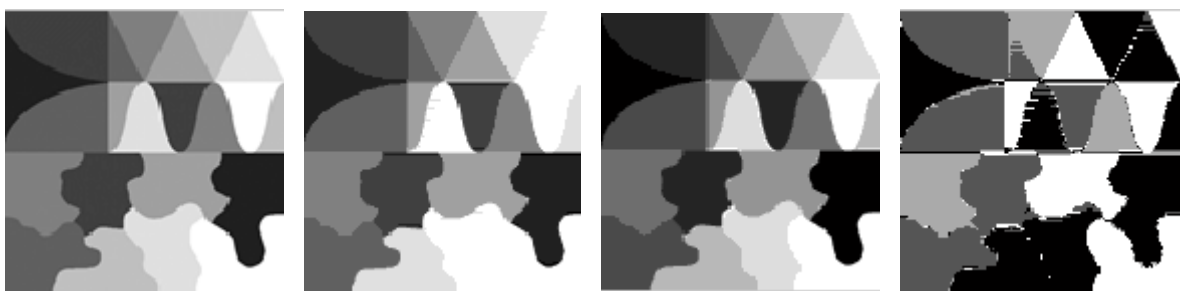
---

**ENTRÉES:**  $K$  : nombre de classes d'appartenance**SORTIES:** La matrice de transition  $A^0$ 

- 1: **pour** ( $i = 1; i = K; i ++$ ) **faire**
  - 2:     **pour** ( $j = 1, j = K; j ++$ ) **faire**
  - 3:         **si** ( $i == j$ ) **alors**
  - 4:              $A^0(i, j) = 1/2$
  - 5:         **sinon**
  - 6:              $A^0(i, j) = 1/2(K - 1)$
  - 7:         **finsi**
  - 8:     **fin pour**
  - 9: **fin pour**
- 

**a. Segmentation des images niveau de gris**

Nous avons réalisé sept expériences, dans chacune, nous avons défini le nombre de classes et le nombre de stationnarités en se basant sur les caractéristiques de l'image à segmenter :



(a) Image originale  
 $Y$

(b) CMC-BI  $X$

(c) CMT  $X$

(d) Le processus  $U$

FIGURE 2.24 – Résultats de segmentation 1 avec  $K = 9$  et  $M = 5$ .

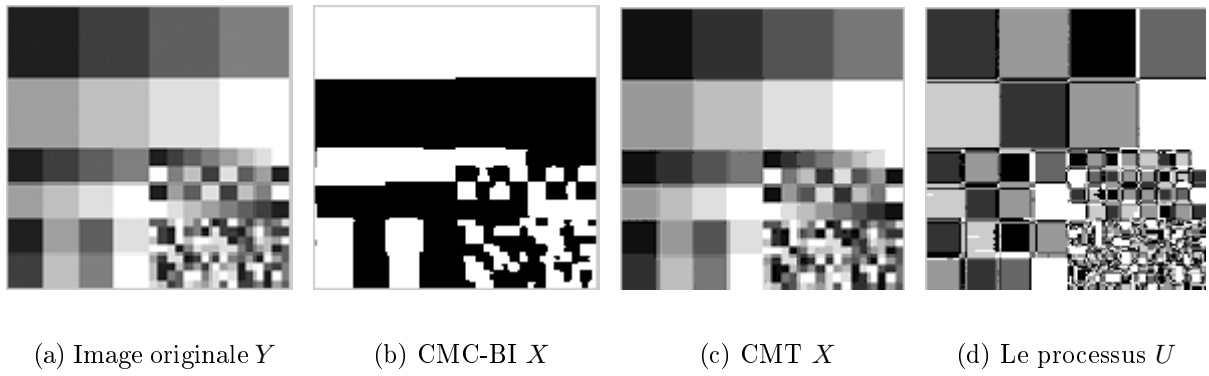


FIGURE 2.25 – Résultats de segmentation 2 avec  $K = 11$  et  $M = 6$ .

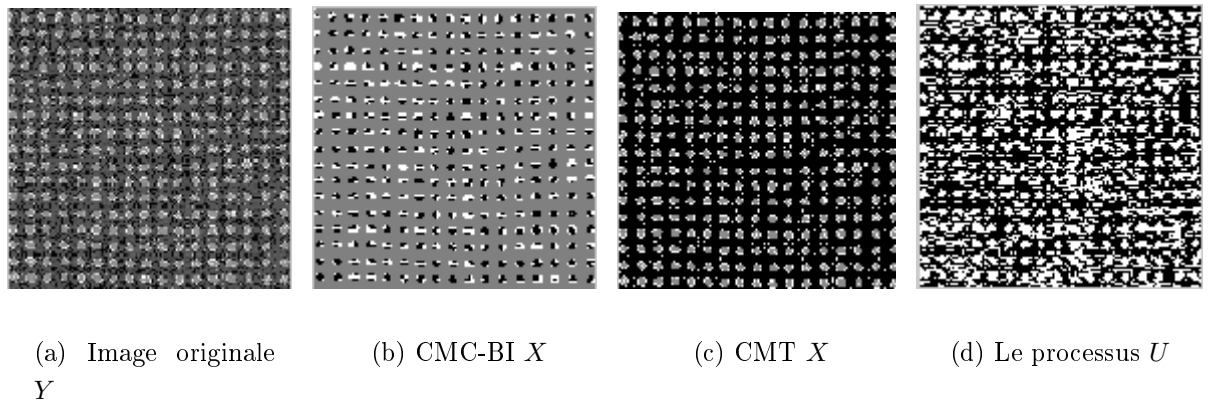


FIGURE 2.26 – Résultats de segmentation 3 avec  $K = 3$  et  $M = 2$ .

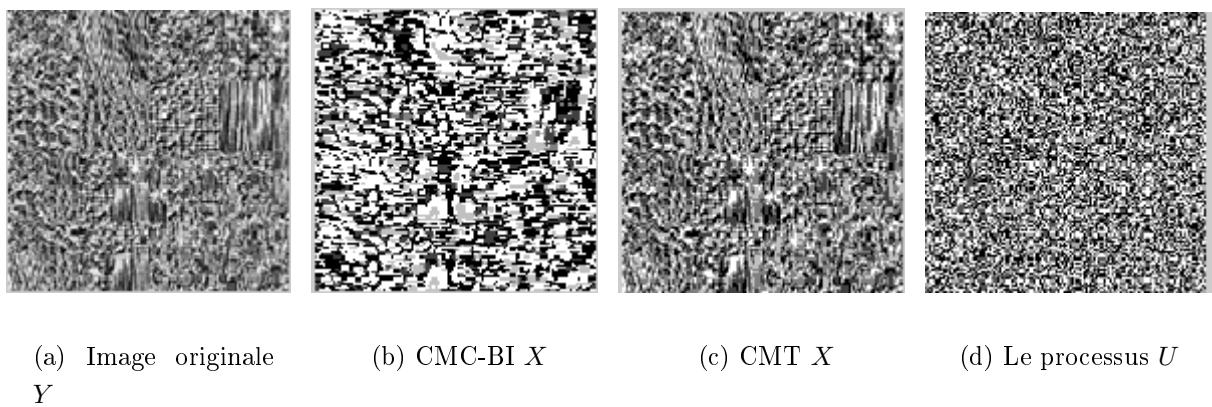


FIGURE 2.27 – Résultats de segmentation 4 avec  $K = 5$  et  $M = 3$ .

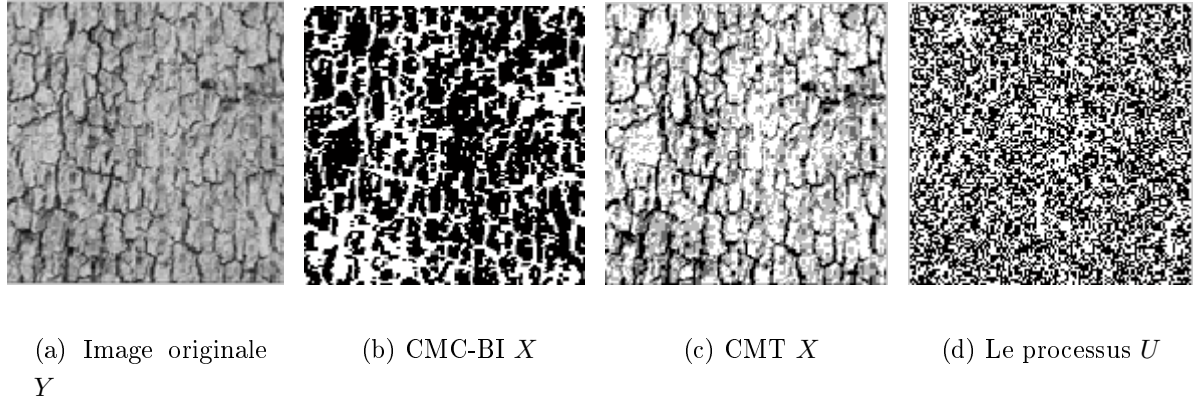


FIGURE 2.28 – Résultats de segmentation 5 avec  $K = 3$  et  $M = 2$ .

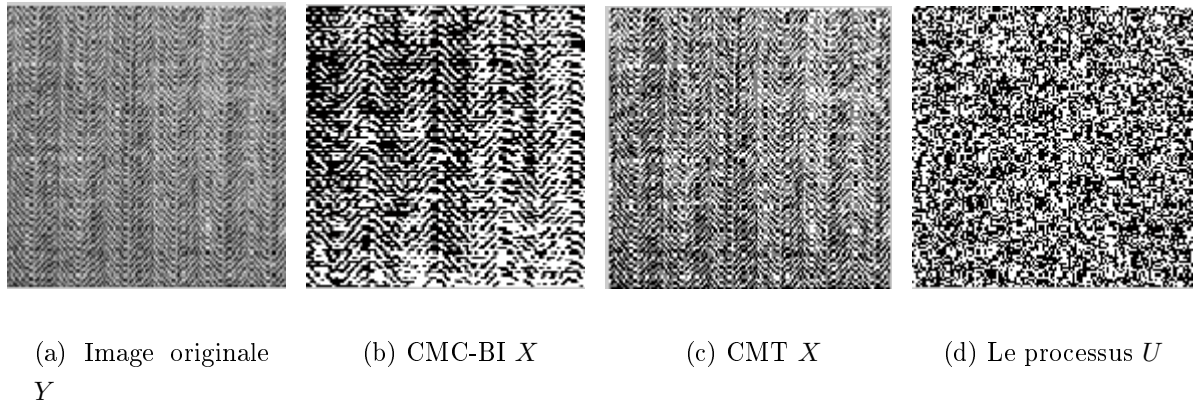


FIGURE 2.29 – Résultats de segmentation 6 avec  $K = 3$  et  $M = 4$ .

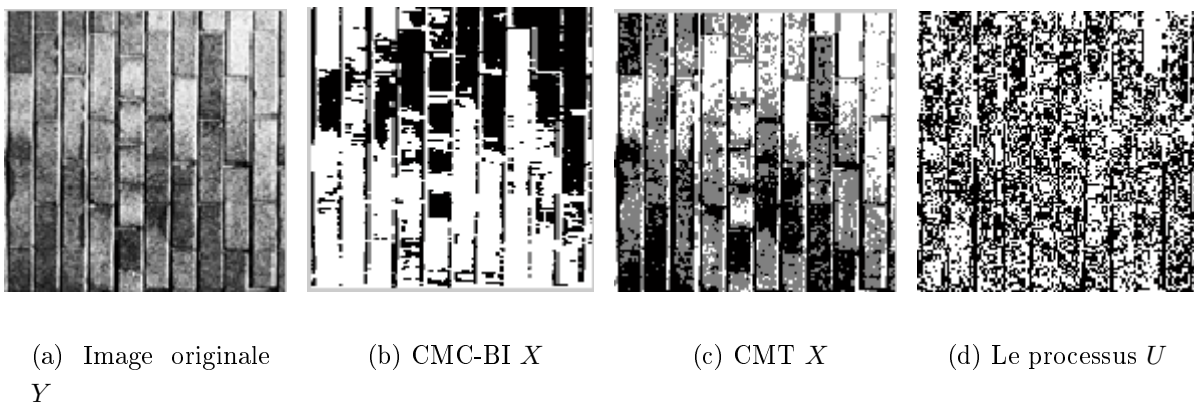


FIGURE 2.30 – Résultats de segmentation 7 avec  $K = 3$  et  $M = 2$ .

Visuellement, les images résultats de segmentation obtenues par CMT sont plus puissantes que celles du CMC-BI.

Pour confirmer les résultats visuels, nous avons comparé CMT et CMC-BI en calculant l'indice de PSNR, l'indice de SSIM et le taux d'erreur. Les deux tableaux suivants présentent cette comparaison.

Expérience	PSNR de CMT	PSNR de CMC-BI PSNR	SSIM de CMT	SSIM de CMC-BI
Expérience 1	33,9860	26,6559	0,8852	0,5932
Expérience 2	29,3677	18,7685	0,7560	0,2680
Expérience 3	23,9470	21,7829	0,5630	0,4645
Expérience 4	27,8879	20,7690	0,8390	0,3768
Expérience 5	25,7756	21,1293	0,7993	0,4560
Expérience 6	25,5675	22,1756	0,7017	0,4167
Expérience 7	23,3738	18,9866	0,5966	0,3526

TABLE 2.15 – Les valeurs des indices PSNR et SSIM

Expérience	Taux d'erreur de CMT	Taux d'erreur de CMC-BI
Expérience 1	2,9854	15,3567
Expérience 2	3,5393	28,5657
Expérience 3	8,5671	12,3453
Expérience 4	11,4270	20,6274
Expérience 5	9,6075	25,6379
Expérience 6	2,3676	10,3394
Expérience 7	6,3893	24,4478

TABLE 2.16 – Les valeurs de taux d'erreur

Dans toutes les expériences de segmentation d'images niveaux de gris, les meilleurs résultats de la segmentation sont ceux obtenus par le modèle CMT, car il prend en compte la non-stationnarité des données à segmenter, il est capable de modéliser cette non-stationnarité en utilisant un processus auxiliaire  $U$ . Par conséquent, l'avantage d'utiliser un processus auxiliaire par les modèles triplets est : d'améliorer la qualité de la segmentation et mieux modéliser l'information de texture dans les images segmentées.

## b. Segmentation des images couleurs

Pour segmenter les images couleurs, nous suivons la même procédure d'initialisation des paramètres que les expériences de segmentation des images niveaux de gris. Nous avons réalisé huit expériences.

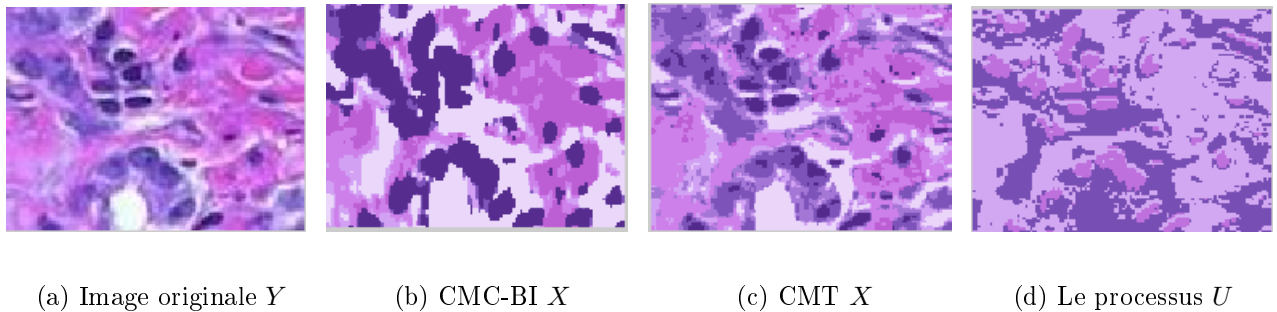


FIGURE 2.31 – Résultats de segmentation 1 avec  $K = 7$  et  $M = 3$ .

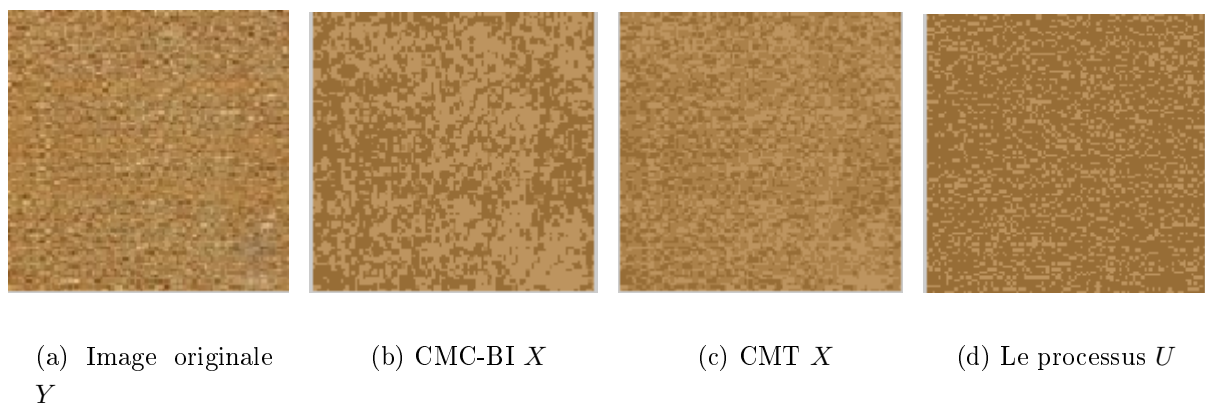
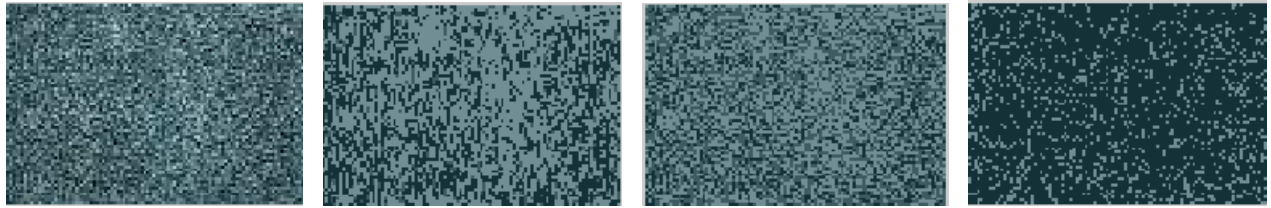
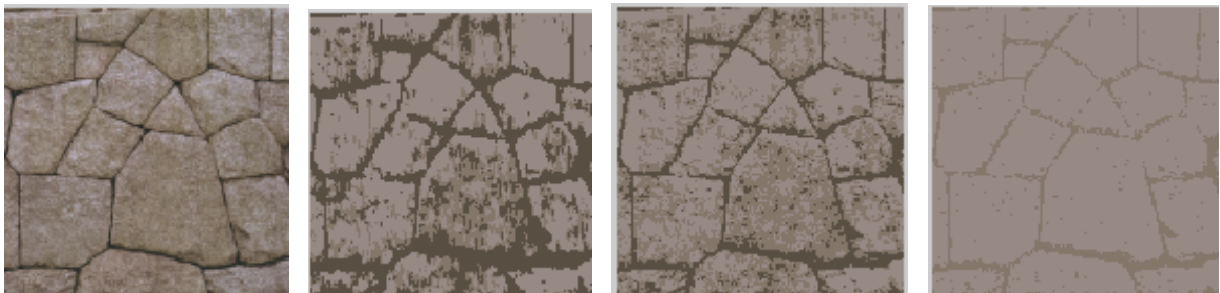


FIGURE 2.32 – Résultats de segmentation 2 avec  $K = 3$  et  $M = 2$ .



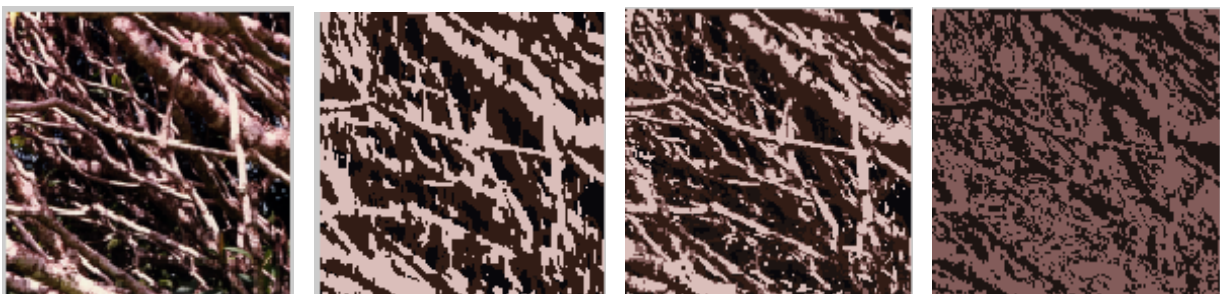
(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

FIGURE 2.33 – Résultats de segmentation 3 avec  $K = 3$  et  $M = 2$ .



(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

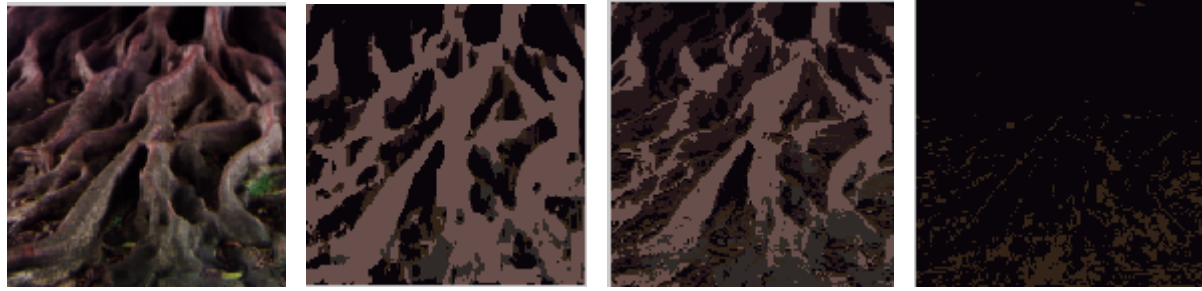
FIGURE 2.34 – Résultats de segmentation 4 avec  $K = 3$  et  $M = 2$ .



(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

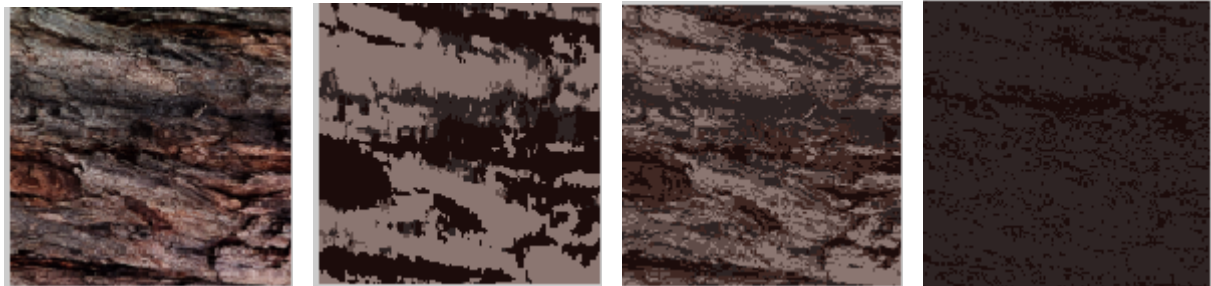
FIGURE 2.35 – Résultats de segmentation 5 avec  $K = 3$  et  $M = 2$ .





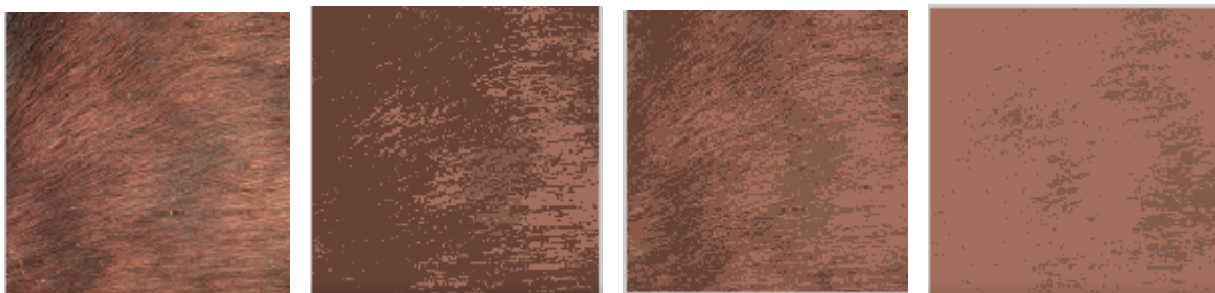
(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

FIGURE 2.36 – Résultats de segmentation 6 avec  $K = 3$  et  $M = 2$ .



(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

FIGURE 2.37 – Résultats de segmentation 7 avec  $K = 3$  et  $M = 2$ .



(a) Image originale  $Y$       (b) CMC-BI  $X$       (c) CMT  $X$       (d) Le processus  $U$

FIGURE 2.38 – Résultats de segmentation 8 avec  $K = 3$  et  $M = 2$ .

Même remarque pour la segmentation des images couleurs, CMT donne de meilleurs résultats de segmentation que CMC-BI. Nous avons résumé les différents résultats quantitatifs obtenus dans les tableaux suivants :

Expériences	PSNR de CMT	PSNR de CMC-BI	SSIM de CMT	SSIM de CMC-BI
Expérience 1	29,767	23.1356	0,8767	0.6986
Expérience 2	29.8217	17.2632	0.7735	0.2197
Expérience 3	25,0393	19.4821	0,7322	0.3047
Expérience 4	27,9653	21.1598	0,6529	0.2274
Expérience 5	23.5618	18.1273	0.6713	0.2284
Expérience 6	25,8678	20.5127	0,7058	0.3757
Expérience 7	23,5479	18.2388	0,6738	0.2922
Expérience 8	29,5479	20.2388	0,7398	0.3922

TABLE 2.17 – Les valeurs de PSNR et SSIM indices

Expérience	Taux d'erreur de CMT	Taux d'erreur de CMC-BI
Expérience 1	8,2127	10,2567
Expérience 2	6.5067	25.1657
Expérience 3	7,9766	24,9859
Expérience 4	8,4750	19,4435
Expérience 5	11.6075	29.7650
Expérience 6	10,1420	27,0450
Expérience 7	14,4725	29,9878
Expérience 8	9,9658	26,4128

TABLE 2.18 – Les valeurs de taux d'erreur

À partir des résultats visuels et quantitatifs, nous concluons que le modèle CMT fournit des meilleurs résultats de segmentation que CMC-BI, parce qu'il considère que les données à traiter sont non stationnaires. Grâce à ce principe, les modèles triplets peuvent résoudre certains problèmes de modélisation que les modèles classiques ne peuvent pas résoudre. Parmi ces problèmes, nous citons : la modélisation de la texture, la texture se caractérise par ses non-stationnarités, dans la majorité des cas, nous trouverons que les mêmes régions texturées se répètent de manière aléatoire dans une image texturée. La répétition d'une certaine région représente une stationnarité différente. Il est préférable d'utiliser les modèles de Markov triplets. Pour segmenter les images texturées.

## 2.6 Les chaînes de Markov cachées floues

Les Chaînes de Markov Cachées Floues (CMCF) en anglais (Fuzzy Hidden Markov Chains) servent à segmenter les images en prenant en compte l'information floue. Ces modèles combinent deux théories : la théorie de probabilité et la théorie des sous-ensembles flous. Nous utilisons ce

modèle pour segmenter des images floues gris et couleur, pour estimer les paramètres, nous proposons une étude comparative entre les estimateurs EM, SEM et ECI. MPM est utilisé pour estimer l'image résultat de segmentation  $X$ .

### 2.6.1 Présentation de modèle

Ce modèle est capable de modéliser l'incertitude et l'imprécision des observations en utilisant une mixture entre la théorie de Bayes et la théorie des ensembles flous. Il utilise l'approche probabiliste pour représenter l'incertain et l'approche floue [109] pour modéliser l'imprécision.

Le modèle CMCF garde les mêmes propriétés que le modèle classique CMC présentées dans 2.2.1.

Soit  $Z = (X, Y)$  où  $Y = (Y_n)_{n=1}^N \in \mathbb{R}$  et  $X = (X_n)_{n=1}^N \in [0, 1]$ .

$Z$  est une CMCF, si et seulement s'il satisfait les mêmes propriétés que celles du modèle CMC citées dans la section 2.2.1.

Pour définir les classes d'appartenance des pixels, CMCF suppose que :

- si  $x_n = 0$ ,  $x_n$  appartient à la classe dure 0.
- si  $x_n = 1$ ,  $x_n$  appartient à la classe dure 1.
- si  $x_n = \zeta_f$ ,  $x_n$  appartient à la classe floue du degré d'appartenance  $\zeta_f$ .

Pour simplifier :

- si  $x_n \in \{0, 1\}$ ,  $x_n$  est un pixel d'une classe dure 0 ou 1.
- si  $x_n \in ]0, 1[$ ,  $x_n$  est un pixel d'une classe floue.

La classe d'appartenance de  $x_n$  est incluse dans l'intervalle  $\Omega_f = [0, 1]$ .

La loi de  $X$  peut être définie par la mesure de densité  $\nu$ .

où  $\nu$  : est une mixture de données discrètes et continues. Cette mesure représente les composantes discrètes par la mesure Dirac  $\delta$  incluse dans  $\{0, 1\}$  comme suit :

- $\delta_0$  : représente la classe dure 0.
- $\delta_1$  : représente la classe dure 1.

Les composantes floues sont définies par la mesure de Lebesgue  $\mu$ . Elle est définie sur l'intervalle ouvert  $]0, 1[$ .

Finalement, la densité de la mesure  $\nu$  est définie par :

$$\nu = \delta_0 + \delta_1 + \mu \tag{2.98}$$

Nous pouvons également utiliser :

$$p(0) + p(1) + \int_0^1 p(\zeta) d\zeta = 1 \tag{2.99}$$

Où :

- $p(0)$  et  $p(1)$  : représentent respectivement la probabilité de classe dure 0 et 1.
- L'intégrale  $\int_0^1 p(\zeta) d\zeta$  : représente la densité des composantes floues.

Numériquement, CMCF discrétise la partie continue des données floues en divisant l'intervalle  $]0, 1[$

approximativement sur un certain nombre de sous intervalles  $F$  tels que :  $[a_0 = 0, a_1 = \frac{1}{F}]$ ,  $[a_1 = \frac{1}{F}, a_2 = \frac{2}{F}]$ ,  $\dots$ ,  $[a_{F-1} = \frac{F-1}{F}, a_F = 1]$ .

Le degré d'appartenance de chaque niveau flou  $\zeta_f$  est obtenu par le calcul de la valeur médiane de chaque sous-intervalle.

$F$  est le nombre de classes floues [75].

CMCF a les mêmes lois et paramètres que le modèle classique [43]. La loi initiale  $PI$ , la matrice de transition  $A$ , la moyenne  $m_0$  et la variance  $\sigma_0^2$  de la classe dure 0, la moyenne  $m_1$  et la variance  $\sigma_1^2$  de la classe dure 1 et la moyenne  $m_f$  et la variance  $\sigma_f$  de chaque classe  $f \in ]0, 1[$ .

Les moyenne  $m_f$  et les variance  $\sigma_f$  des classes floues sont calculées à partir de la moyenne et de la variance des classes dures par :

$$m_f = \zeta_f \cdot m_0 + (1 - \zeta_f) \cdot m_1 \forall f \in ]0, 1[. \quad (2.100)$$

$$\sigma_f^2 = (\zeta_f)^2 \cdot \sigma_0^2 + (1 - \zeta_f)^2 \cdot \sigma_1^2 \forall f \in ]0, 1[. \quad (2.101)$$

## 2.6.2 Estimation des paramètres

Dans ce paragraphe, nous présentons les versions des algorithmes itératifs utilisés pour estimer les paramètres du modèle CMCF. Nous présentons les démarches de EM, SEM, ECI et MPM.

### a. L'algorithme EM

L'algorithme de EM pour CMCF se déroule de la manière suivante :

---

**Algorithme 19** L'algorithme EM pour CMCF

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (PI^0, A^0, m_i^0, (\sigma_i^0)^2)$

**SORTIES:**  $\theta^Q = (PI^Q, A^Q, m_i^Q, (\sigma_i^Q)^2)$

1: **pour** chaque itération  $q \in Q$  **faire**

2: EM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch présenté dans 2.2.2.

3: EM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X \in [0, 1]$  par :

$$PI^{q+1}(i) = \xi_1^q(i) \forall i \in [0, 1] \quad (2.102)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \forall i, j \in [0, 1] \quad (2.103)$$

4: EM estime les paramètres  $\theta_y^{q+1}$  pour les observations  $Y$  par :

$$m_i^{q+1} = \frac{\sum_{n=1}^N y_n \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \forall i \in \{0, 1\} \quad (2.104)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - m_i^{q+1})^2 \cdot \xi_n^q(i)}{\sum_{n=1}^N \xi_n^q(i)} \forall i \in \{0, 1\} \quad (2.105)$$

5: EM estime les paramètres  $\theta_y^{q+1}$  des observations floues  $Y$  en utilisant les équations 2.100 et 2.101  $\forall i \in ]0, 1[$

6: EM calcule la densité gaussienne  $f$  en utilisant l'équation 2.5

7:  $q = q + 1$

8: **fin pour**

---

## b. L'algorithme SEM

L'algorithme de SEM pour CMCF se déroule de la manière suivante :

---

**Algorithme 20** L'algorithme SEM pour CMCF

---

**ENTRÉES:**  $q = 0$

**ENTRÉES:**  $\theta^0 = (PI^0, A^0, m_i^0, (\sigma_i^0)^2)$

**SORTIES:**  $\theta^Q = (PI^Q, A^Q, m_i^Q, (\sigma_i^Q)^2)$

1: **pour** chaque itération  $q \in Q$  **faire**

2: SEM calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch présenté dans 2.2.2.

3: SEM estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X \in [0, 1]$  par :

$$PI^{q+1}(i) = \frac{1}{N-1} \sum_{n=1}^N \mathbf{1}[x_n = i] \forall i \in [0, 1] \quad (2.106)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=2}^N \mathbf{1}[x_n = j, x_{n-1} = i]}{N-1} \forall i, j \in [0, 1] \quad (2.107)$$

4: SEM estime les paramètres  $\theta_y^{q+1}$  pour les observations  $Y$  par :

$$m_i^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \forall i \in \{0, 1\} \quad (2.108)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - m_i^{q+1})^2 \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \forall i \in \{0, 1\} \quad (2.109)$$

5: SEM estime les paramètres  $\theta_y^{q+1}$  des observations floues  $Y$  en utilisant les équations 2.100 et 2.101  $\forall i \in ]0, 1[$

6: SEM calcule la densité gaussienne  $f$  en utilisant l'équation 2.5

7:  $q = q + 1$

8: **fin pour**

---

### c.L'algorithme ECI

L'algorithme de ECI pour CMCF se déroule de la manière suivante :

---

**Algorithme 21** L'algorithme ECI pour CMCF

---

**ENTRÉES:**  $q = 0$ **ENTRÉES:**  $\theta^0 = (PI^0, A^0, m_i^0, (\sigma_i^0)^2)$ **SORTIES:**  $\theta^Q = (PI^Q, A^Q, m_i^Q, (\sigma_i^Q)^2)$ 1: **pour** chaque itération  $q \in Q$  **faire**2: ECI calcule les probabilités  $\alpha^q, \beta^q, \gamma^q$  et  $\xi^q$  en utilisant l'algorithme de Baum-Welch présenté dans 2.2.2.3: ECI estime les paramètres  $\theta_x^{q+1}$  du processus caché  $X \in [0, 1]$  par :

$$PI^{q+1}(i) = \xi_1^q(i) \forall i \in [0, 1] \quad (2.110)$$

$$A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \forall i, j \in [0, 1] \quad (2.111)$$

4: ECI estime les paramètres  $\theta_y^{q+1}$  pour les observations  $Y$  par :

$$m_i^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \forall i \in \{0, 1\} \quad (2.112)$$

$$(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - m_i^{q+1})^2 \mathbf{1}[x_n = i]}{\sum_{n=1}^N \mathbf{1}[x_n = i]} \forall i \in \{0, 1\} \quad (2.113)$$

5: ECI estime les paramètres  $\theta_y^{q+1}$  des observations floues  $Y$  en utilisant les équations 2.100 et 2.101  $\forall i \in ]0, 1[$ 6: ECI calcule la densité gaussienne  $f$  en utilisant l'équation 2.57:  $q = q + 1$ 8: **fin pour**

---

**d. L'algorithme MPM**

L'estimateur MPM estime pour chaque pixel  $y_n$  sa classe d'appartenance thématique ou floue appropriée en maximisant la probabilité *a posteriori*  $P(X_n = x_n | Y_n = y_n)$ . MPM applique la formule mathématique suivante :

$$\bar{x}_n = \mathbf{argmax}_{i \in [0, 1]} (\alpha_i(n) \cdot \beta_i(n)) \quad (2.114)$$

$$= \mathbf{argmax}_{i \in [0, 1]} ((\xi_i(n))). \quad (2.115)$$

**2.6.3 Résultats et expérimentations**

Ce paragraphe présente les différentes expériences réalisées et résultats obtenus, nous avons réalisé quatre expériences, dans la première et la deuxième nous avons segmenté des images couleurs,

dans la troisième et la quatrième nous avons des images niveau de gris. Nous considérons, le cas du nombre de classes dures  $K = 2$ . Nous avons varié le nombre de classes floues  $F$  de  $F = 1$  jusqu'à  $F = 8$ . Chaque expérience est définie par ses résultats visuels et par les valeurs de l'indice de PSNR, de l'indice de SSIM, de taux d'erreur, de temps d'exécution et de convergence. Les paramètres de la CMCF sont initialisés par :

La loi initiale de  $X$ ,  $PI^0$  et la matrice de transition  $A^0$  sont initialisées à l'aide des algorithmes suivants :

---

**Algorithme 22** Initialisation de  $PI^0$

---

**ENTRÉES:**  $K$  : Nombre de classes d'appartenance dures

**ENTRÉES:**  $F$  : Nombre de classes d'appartenance floues

**SORTIES:** La loi initiale  $PI^0$

- 1: **pour** ( $i = 1; i = K + F; i ++$ ) **faire**
  - 2:      $PI^0(i) = 1./K + F$
  - 3: **fin pour**
- 

---

**Algorithme 23** Initialisation de  $A^0$

---

**ENTRÉES:**  $K$  : Nombre de classes d'appartenance dures

**ENTRÉES:**  $F$  : Nombre de classes d'appartenance floues

**SORTIES:** La matrice de transition  $A^0$

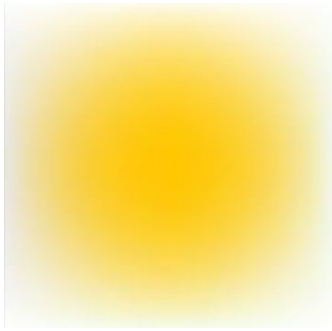
- 1: **pour** ( $i = 1; i = K + F; i ++$ ) **faire**
  - 2:     **pour** ( $j = 1, j = K + F; j ++$ ) **faire**
  - 3:         **si** ( $i == j$ ) **alors**
  - 4:              $A^0(i, j) = 1/2$
  - 5:         **sinon**
  - 6:              $A^0(i, j) = 1/2((K + F) - 1)$
  - 7:         **finsi**
  - 8:     **fin pour**
  - 9: **fin pour**
- 

La moyenne  $m^0$  et la variance  $(\sigma_i^0)^2$  des classes thématiques sont calculées à partir du processus initial  $X^0$  en utilisant les equation 2.33 2.34, ce dernier a été obtenu par K-means. Les moyennes  $m_f^0$  et les variances  $(\sigma_f^0)^2$  initiales floues sont calculées à partir des moyennes et des variances des classes dures à l'aide des équations 2.100 et 2.101.

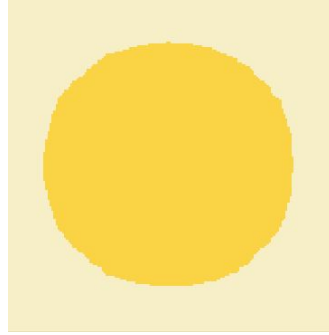
### a. Les résultats visuels

Les résultats visuels obtenus sont présentés dans les figures suivantes :

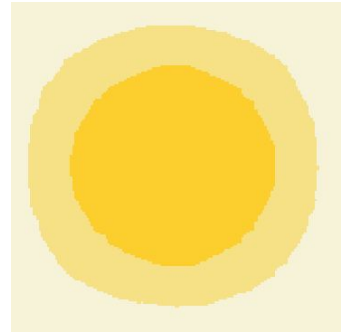




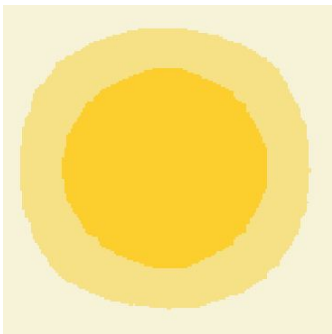
(a) Image originale  $Y$



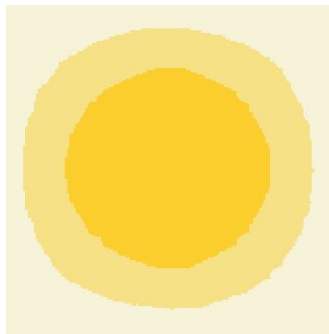
(b) Configuration Initiale  $X^0$



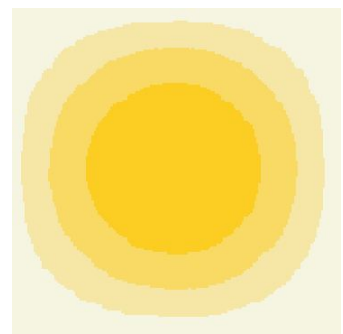
(c) Processus final  $X$  EM  $F=1$



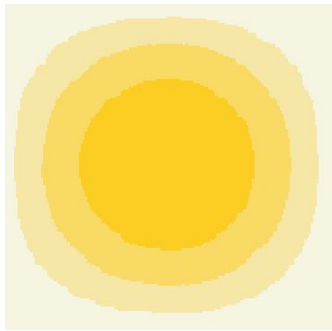
(d) Processus final  $X$  SEM  $F=1$



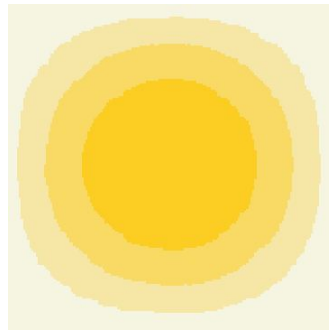
(e) Processus final  $X$  ECI  $F=1$



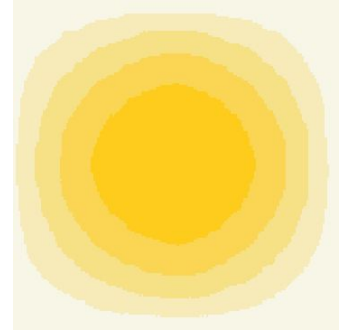
(f) Processus final  $X$  EM  $F=2$



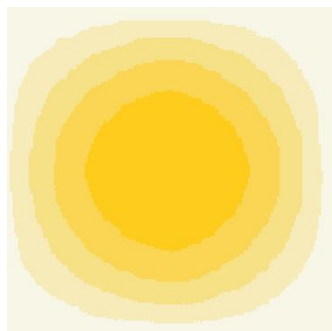
(g) Processus final  $X$  SEM  $F=2$



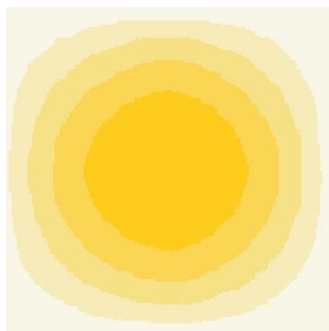
(h) Processus final X ECI  $F=2$



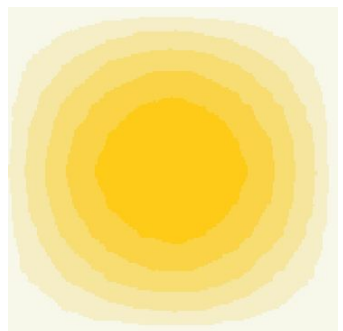
(i) Processus final  $X$  EM  $F=3$



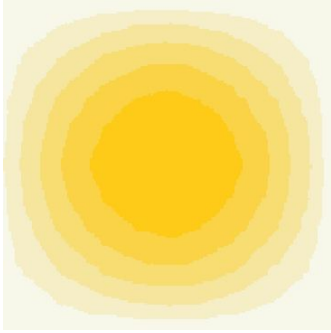
(j) Processus final  $X$  SEM  $F=3$



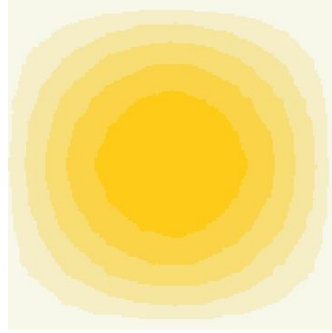
(k) Processus final  $X$  ECI  $F=3$



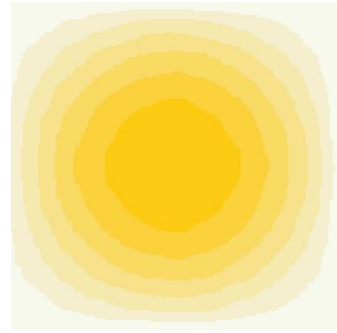
(l) Processus final  $X$  EM  $F=4$



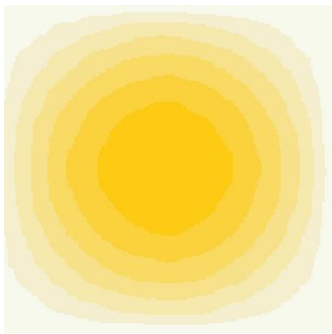
(m) Processus final  
XSEM F=4



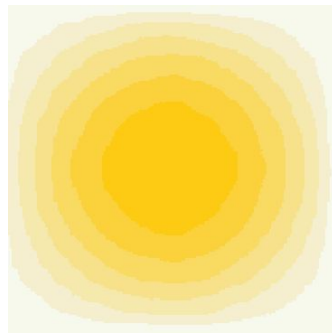
(n) Processus final X  
ECI F=4



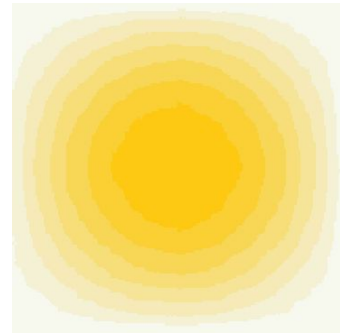
(o) Processus final X  
EM F=5



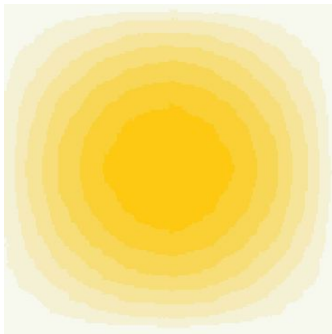
(p) Processus final X  
SEM F=5



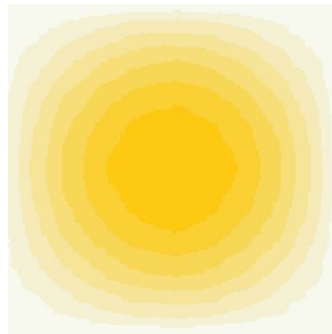
(q) Processus final X  
ECI F=5



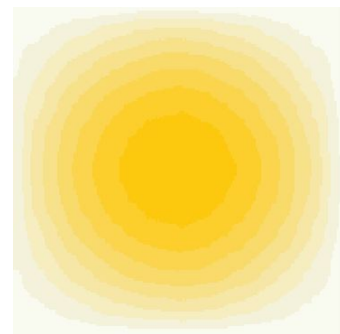
(r) Processus final X EM  
F=6



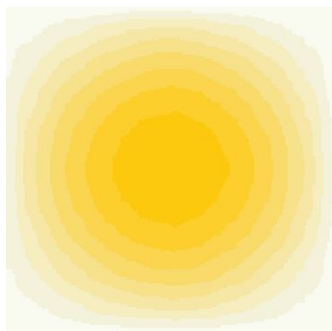
(s) Processus final X  
SEM F=6



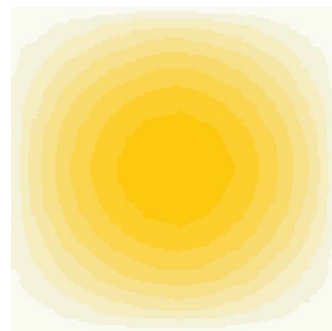
(t) Processus final X  
ECI F=6



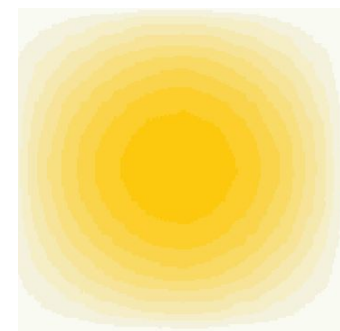
(u) Processus final X EM  
F=7



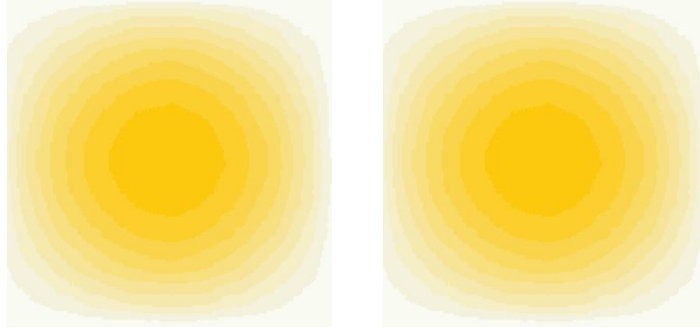
(v) Processus final X  
SEM F=7



(w) Processus final X  
ECI F=7



(x) Processus final X  
EM F=8



(y) Processus final  $X$   
SEM F=8

(z) Processus final  $X$   
ECI F=8

FIGURE 2.39 – Résultats de segmentation d'une image couleur floue 1.



(a) Image originale  $Y$



(b) Configuration Initiale  $X^0$



(c) Processus final  $X$  EM  $F=1$



(d) Processus final  $X$  SEM  $F=1$



(e) Processus final  $X$  ECI  $F=1$



(f) Processus final  $X$  EM  $F=2$



(g) Processus final  $X$  SEM  $F=2$



(h) Processus final  $X$  ECI  $F=2$



(i) Processus final  $X$  EM  $F=3$



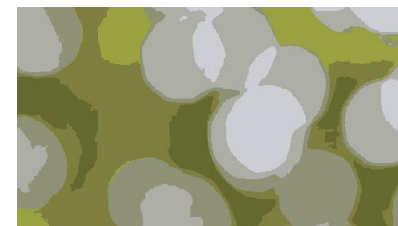
(j) Processus final  $X$  SEM  $F=3$



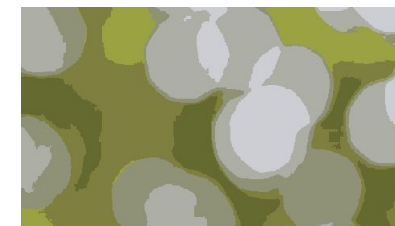
(k) Processus final  $X$  ECI  $F=3$



(l) Processus final  $X$  EM  $F=4$



(m) Processus final  $X$  SEM  $F=4$



(n) Processus final  $X$  ECI  $F=4$



(o) Processus final  $X$  EM  $F=5$



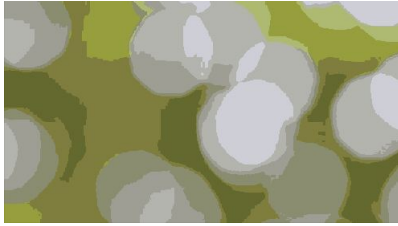
(p) Processus final  $X$  SEM  
F=5



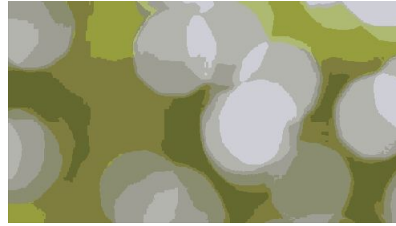
(q) Processus final  $X$  ECI F=5



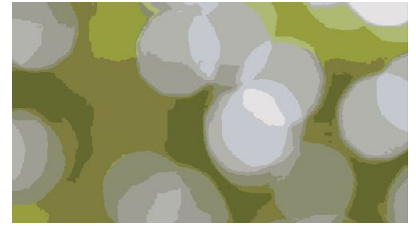
(r) Processus final  $X$  EM F=6



(s) Processus final  $X$  SEM  
F=6



(t) Processus final  $X$  ECI F=6



(u) Processus final  $X$  EM F=7



(v) Processus final  $X$  SEM  
F=7



(w) Processus final  $X$  ECI  
F=7



(x) Processus final  $X$  EM F=8



(y) Processus final  $X$  SEM  
F=8



(z) Processus final  $X$  ECI F=8

FIGURE 2.40 – Résultats de segmentation d'une image couleur floue 2.



(a) Image originale  $Y$



(b) Configuration Initiale  $X^0$



(c) Processus final  $X$  EM  $F=1$



(d) Processus final  $X$  SEM  $F=1$



(e) Processus final  $X$  ECI  $F=1$



(f) Processus final  $X$  EM  $F=2$



(g) Processus final  $X$  SEM  $F=2$



(h) Processus final  $X$  ECI  $F=2$



(i) Processus final  $X$  EM  $F=3$



(j) Processus final  $X$  SEM  $F=3$



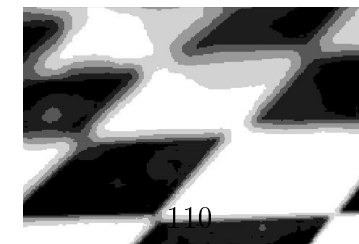
(k) Processus final  $X$  ECI  $F=3$



(l) Processus final  $X$  EM  $F=4$



(m) Processus final  $X$  SEM  $F=4$



(n) Processus final  $X$  ECI  $F=4$



(o) Processus final  $X$  EM  $F=5$



(p) Processus final  $X$  SEM  
F=5



(q) Processus final  $X$  ECI  
F=5



(r) Processus final  $X$  EM  
F=6



(s) Processus final  $X$  SEM  
F=6



(t) Processus final  $X$  ECI  
F=6



(u) Processus final  $X$  EM  
F=7



(v) Processus final  $X$  SEM  
F=7



(w) Processus final  $X$  ECI  
F=7



(x) Processus final  $X$  EM  
F=8

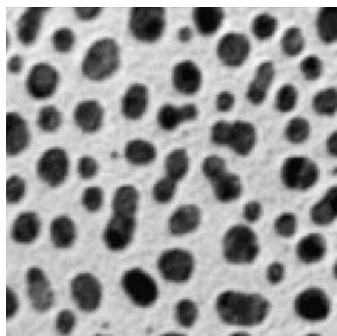


(y) Processus final  $X$  SEM  
F=8

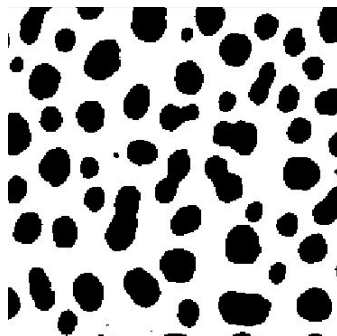


(z) Processus final  $X$  ECI  
F=8

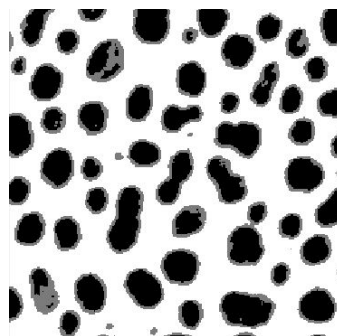
FIGURE 2.41 – Résultats de segmentation d'une image niveau de gris floue 1.



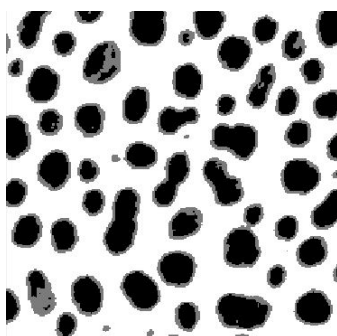
(a) Image originale  $Y$



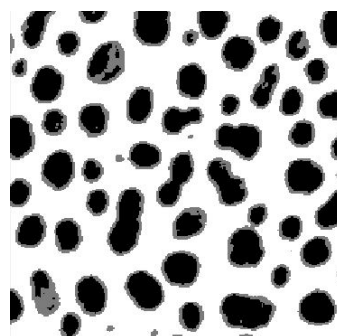
(b) Configuration Initiale  
 $X^0$



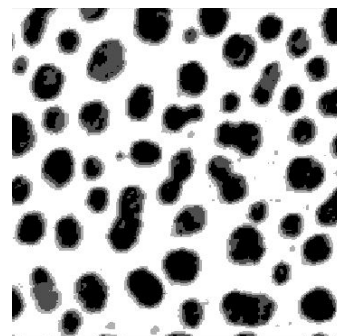
(c) Processus final  $X$  EM  
 $F=1$



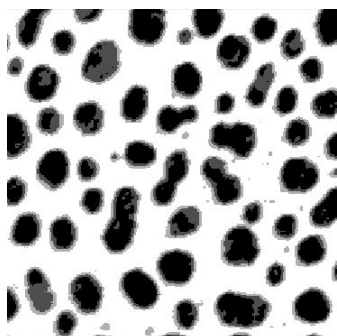
(d) Processus final  $X$   
SEM  $F=1$



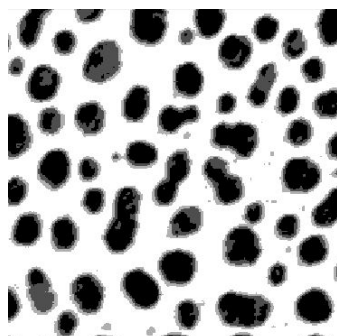
(e) Processus final  $X$  ECI  
 $F=1$



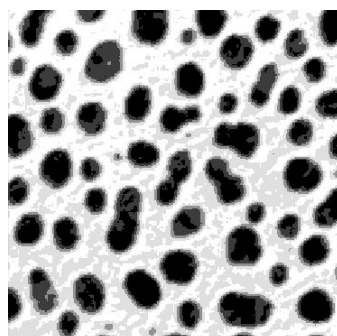
(f) Processus final  $X$  EM  
 $F=2$



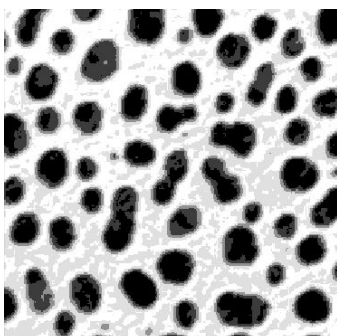
(g) Processus final  $X$   
SEM  $F=2$



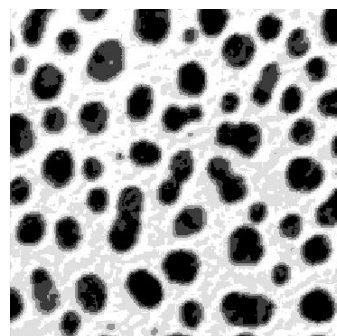
(h) Processus final  $X$  ECI  
 $F=2$



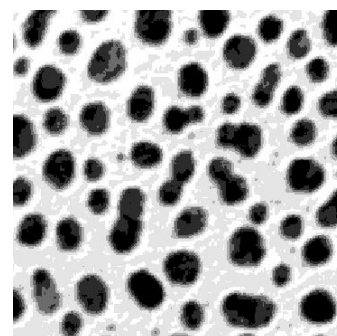
(i) Processus final  $X$  EM  
 $F=3$



(j) Processus final  $X$   
SEM  $F=3$

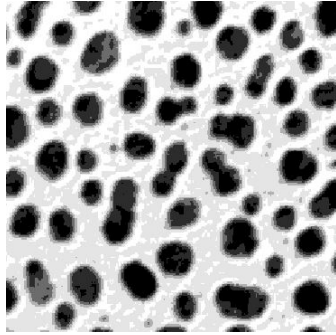


(k) Processus final  $X$   
ECI  $F=3$

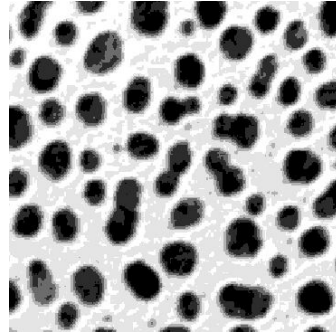


(l) Processus final  $X$  EM  
 $F=4$

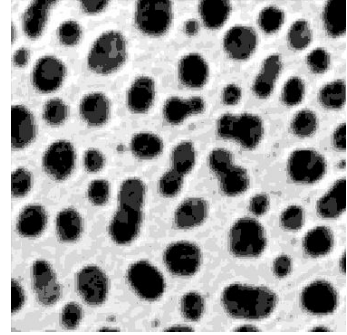




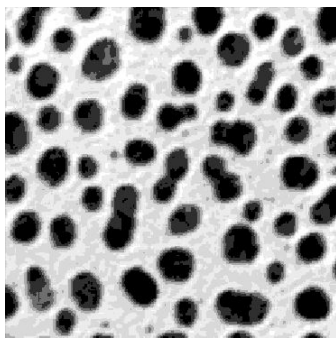
(m) Processus final XSEM F=4



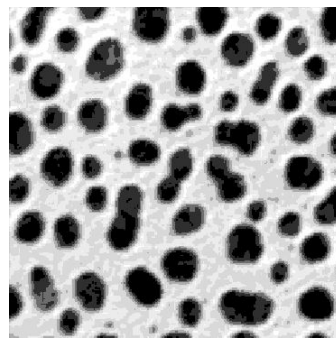
(n) Processus final X ECI F=4



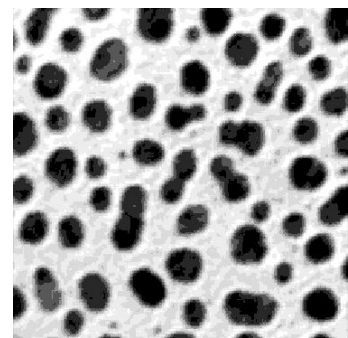
(o) Processus final X EM F=5



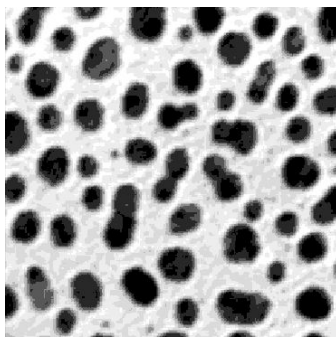
(p) Processus final X SEM F=5



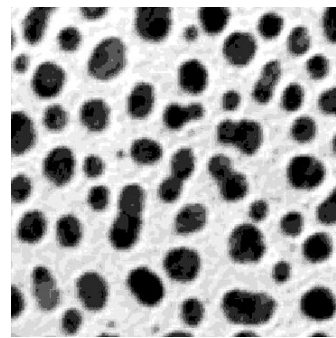
(q) Processus final X ECI F=5



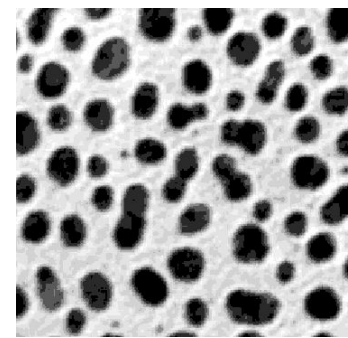
(r) Processus final X EM F=6



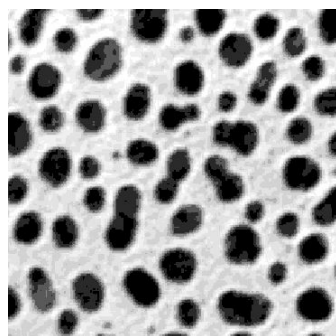
(s) Processus final X SEM F=6



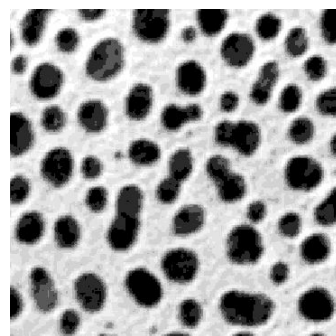
(t) Processus final X ECI F=6



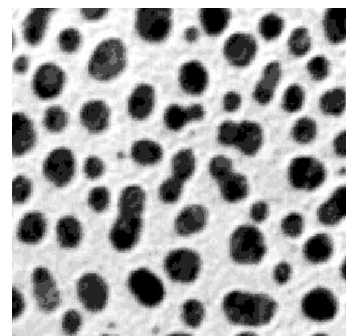
(u) Processus final X EM F=7



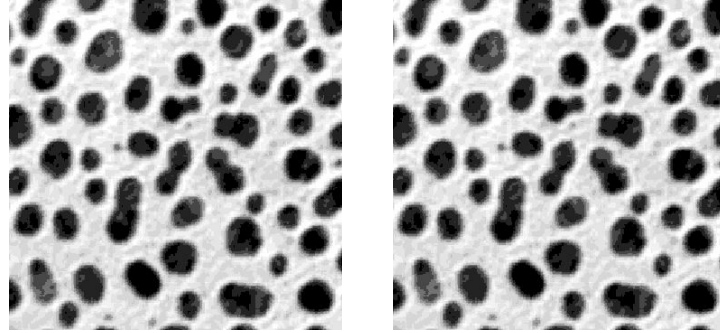
(v) Processus final X SEM F=7



(w) Processus final X ECI F=7



(x) Processus final X EM F=8



(y) Processus final  $X$   
SEM F=8

(z) Processus final  $X$   
ECI F=8

FIGURE 2.42 – Résultats de segmentation d'une image niveau de gris flou 2.

D'après les images résultats obtenues, nous remarquons que, lorsqu'on augmente les niveaux de flous, l'information floue apparait bien.

Il n'y a pas de différence entre les estimateurs.

### b. Les résultats quantitatifs

Pour valider les résultats visuels. Nous avons calculé pour chaque estimateur : les indices de PSNR, SSIM et le taux d'erreur. Les tableaux 2.6.3, 2.6.3, 2.6.3, 2.6.3, 2.6.3, 2.6.3, 2.6.3 et 2.6.3 représentent respectivement les valeurs obtenues dans les expériences 1,2,3 et 4.

Niveau de flou	EM	SEM	ECI	EM	SEM	ECI
F=1	32.7822	32.7822	32.7822	0.8875	0.8875	0.8875
F=2	35.211	35.211	35.211	0.8949	0.8949	0.8949
F=3	36.9479	36.9479	36.9479	0.9129	0.9129	0.9129
F=4	38.6609	38.6609	38.6609	0.9208	0.9208	0.9208
F=5	39.7903	39.7903	39.7903	0.9312	0.9312	0.9312
F=6	41.0536	41.0536	41.0536	0.9442	0.9442	0.9442
F=7	42.0451	42.0451	42.0451	0.9534	0.9534	0.9534
F=8	42.8346	42.8346	42.8346	0.9589	0.9589	0.9589

TABLE 2.19 – Les valeurs de PSNR et SSIM pour expérience 1

Niveau de flou	EM	SEM	ECI
F=1	8.3851	8.3851	8.3851
F=2	7.6520	7.6520	7.6520
F=3	5.129	5.129	5.129
F=4	3.5735	3.5735	3.5735
F=5	2.6564	2.6564	2.6564
F=6	1.9287	1.9287	1.9287
F=7	1.8609	1.8609	1.8609
F=8	1.2245	1.2245	1.2245

TABLE 2.20 – Les valeurs de taux d’erreur pour expérience 1

Niveau de flou	EM	SEM	ECI	EM	SEM	ECI
F=1	28.9551	28.9551	28.9551	0.8697	0.8697	0.8697
F=2	29.6579	29.6579	29.6579	0.8715	0.8715	0.8715
F=3	30.4530	30.4530	30.4530	0.8767	0.8767	0.8767
F=4	31.0568	31.0568	31.0568	0.8855	0.8855	0.8855
F=5	32.4166	32.4166	32.4166	0.8933	0.8933	0.8933
F=6	33.2867	33.2867	33.2867	0.8999	0.8999	0.8999
F=7	34.3880	34.3880	34.3880	0.9094	0.9094	0.9094
F=8	35.5017	35.5017	35.5017	0.9227	0.9227	0.9227

TABLE 2.21 – Les valeurs de PSNR et SSIM pour expérience 2

Niveau de flou	EM	SEM	ECI
F=1	8.9102	8.9102	8.9102
F=2	5.1631	5.1631	5.1631
F=3	3.6205	3.6205	3.6205
F=4	2.6007	2.6007	2.6007
F=5	1.8835	1.8835	1.8835
F=6	1.4897	1.4897	1.4897
F=7	1.1607	1.1607	1.1607
F=8	0.0882	0.0882	0.0882

TABLE 2.22 – Les valeurs de taux d’erreur pour expérience 2

Niveau de flou	EM	SEM	ECI	EM	SEM	ECI
F=1	28.4167	28.4167	28.4167	0.7425	0.7425	0.7425
F=2	29.6297	29.6297	29.6297	0.7950	0.7950	0.7950
F=3	30.0566	30.0566	30.0566	0.8330	0.8330	0.8330
F=4	31.0789	31.0789	31.0789	0.8488	0.8488	0.8488
F=5	31.6134	31.6134	31.6134	0.8686	0.8686	0.8686
F=6	32.2726	32.2726	32.2726	0.8908	0.8908	0.8908
F=7	32.9150	32.9150	32.9150	0.8972	0.8972	0.8972
F=8	33.2435	33.2435	33.2435	0.9115	0.9115	0.9115

TABLE 2.23 – Les valeurs de PSNR et SSIM pour expérience 3

Niveau de flou	EM	SEM	ECI
F=1	5.9939	5.9939	5.9939
F=2	3.2610	3.2610	3.2610
F=3	2.1363	2.1363	2.1363
F=4	1.4876	1.4876	1.4876
F=5	1.1025	1.1025	1.1025
F=6	0.8304	0.8304	0.8304
F=7	0.7191	0.7191	0.7191
F=8	0.6122	0.6122	0.6122

TABLE 2.24 – Les valeurs de taux d’erreur pour expérience 3

Niveau de flou	EM	SEM	ECI	EM	SEM	ECI
F=1	29.0691	29.0691	29.0691	0.6935	0.6935	0.6935
F=2	29.3664	29.3664	29.3664	0.7562	0.7562	0.7562
F=3	29.9264	29.9264	29.9264	0.8356	0.8356	0.8356
F=4	30.3177	30.3177	30.3177	0.8642	0.8642	0.8642
F=5	31.0392	31.0392	31.0392	0.8901	0.8901	0.8901
F=6	31.4969	31.4969	31.4969	0.9115	0.9115	0.9115
F=7	31.8957	31.8957	31.8957	0.9223	0.9223	0.9223
F=8	32.1925	32.1925	32.1925	0.9381	0.9381	0.9381

TABLE 2.25 – Les valeurs de PSNR et SSIM pour expérience 4

Niveau de flou	EM	SEM	ECI
F=1	4.7335	4.7335	4.7335
F=2	3.0220	3.0220	3.0220
F=3	2.1496	2.1496	2.1496
F=4	1.5313	1.5313	1.5313
F=5	1.3740	1.3740	1.3740
F=6	0.9490	0.9490	0.9490
F=7	0.9139	0.9139	0.9139
F=8	0.6614	0.6614	0.6614

TABLE 2.26 – Les valeurs de taux d’erreur pour expérience 4

Les valeurs obtenues pour PSNR, SSIM et le taux d’erreur confirment les résultats visuels, EM, SEM et ECI donnent les mêmes résultats de segmentation. Malgré que, les stratégies d’estimation utilisées sont différentes. Lorsque le nombre de niveaux de flous augmente, les valeurs de PSNR et SSIM augmentent, et les valeurs de taux d’erreur diminuent. La qualité de la segmentation floue augmente lorsque le nombre de niveaux de flou augmente.

Également, nous avons comparé ces estimateurs en termes de temps d’exécution et de convergence. Nous avons calculé le temps d’exécution moyen en prenant la somme des temps d’exécution de tous les niveaux flous de chaque estimateur EM, SEM et ECI par expérience divisé par 8. De la même manière, nous avons calculé la moyenne du nombre d’itérations, les figures 2.43 et 2.32 représentent respectivement le temps d’exécution en secondes et la convergence.

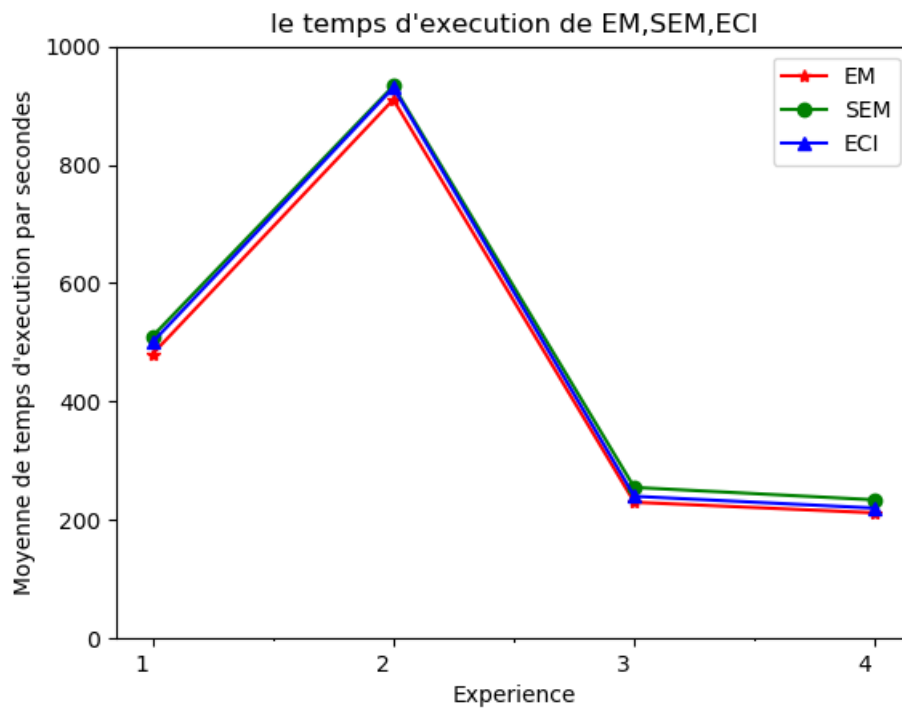


FIGURE 2.43 – Le temps d'exécution en secondes par estimateur

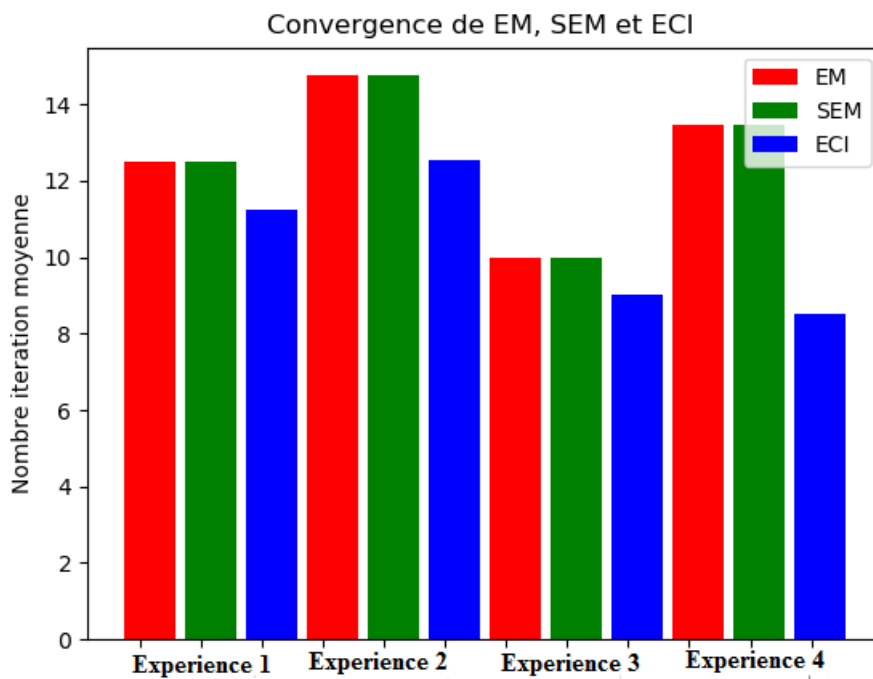


FIGURE 2.44 – La convergence par estimateur

EM est plus rapide que le SEM et ECI avec 15%. Le temps d'exécution d' ECI est plus proche de celle du SEM. ECI est plus rapide que SEM avec 5%. Nous expliquons cela par, les stratégies utilisées par ces estimateurs.

SEM et ECI utilisent la stratégie stochastique pour estimer les paramètres du modèle CMC-BI. Cette stratégie se base sur la simulation du processus caché  $X$  aléatoirement à chaque itération pour calculer les paramètres, ce qui rend la procédure d'estimation plus lente que celle du stratégie déterministe utilisée par EM. Lorsque la taille de données augmente, le temps d'exécution augmente. Le graphe montre que ECI et SEM convergent plus rapidement qu'EM. ECI et SEM ont une vitesse de convergence plus rapide qu' EM. Mais, EM a une vitesse d'exécution plus rapide qu' ECI et SEM. Nous ne pouvons pas conclure l'algorithme le plus performant en termes de la qualité, le temps d'exécution et la convergence.

En outre, nous avons comparé ces estimateurs en terme de niveau de la complexité, pour cela, nous avons calculé la complexité de chaque tâche exécutée par algorithme, le tableau suivant illustre la complexité de EM, SEM et ECI.

Soit :

- $N$  : est la taille totale de l'image d'entrée.
- $K$  : est le nombre de classes dures.
- $F$  : est le nombre de classes floues.

Tache exécutée	EM	SEM	ECI
Probabilité de Forward $\alpha$	$O((K + F)^2 N)$	$O((K + F)^2 N)$	$O((K + F)^2 N)$
Probabilité de Backward $\beta$	$O((K + F)^2 N)$	$O((K + F)^2 N)$	$O((K + F)^2 N)$
Probabilité conjointe <i>a posteriori</i> $\gamma$	$O((K + F)^2 N)$	$O((K + F)^2 N)$	$O((K + F)^2 N)$
Probabilité Marginale <i>a posteriori</i> $\xi$	$O((K + F)N)$	$O((K + F)^2 N)$	$O((K + F)^2 N)$
Simulation du processus $X$	—	$O((K + F)N)$	$O((K + F)N)$
La loi initiale $PI$	$O(K + F)$	$O(K + F)$	$O(K + F)$
La matrice de transition $A$	$O((K + F)^2 N)$	$O((K + F)^2 N)$	$O((K + F)^2 N)$
La moyenne $m$	$O((K + F)N)$	$O((K + F)N)$	$O((K + F)N)$
La variance $\sigma^2$	$O((K + F)N)$	$O((K + F)N)$	$O((K + F)N)$
Calcul de densité $f$	$O((K + F)N)$	$O((K + F)N)$	$O((K + F)N)$
La moyenne floue $m_f$	$O(F)$	$O(F)$	$O(F)$
La variance floue $\sigma_f$	$O(F)$	$O(F)$	$O(F)$

TABLE 2.27 – Étude comparative entre EM, SEM et ECI en terme de la complexité

À partir du tableau 2.6.3, EM est moins complexe que ECI et SEM. ECI et SEM ont le même niveau de complexité. le modèle CMCF est plus complexe que le modèle classique CMC-BI, car, il calcule les paramètres des composantes floues, il calcule la moyenne et la variance de chaque niveau de flou, lorsque, le nombre de niveaux de flou augmente ou la taille des données augmente,

la complexité de calcul augmente.

EM, SEM et ECI ont une complexité linéaire  $O(N)$ .

## 2.7 Conclusion

Nous avons présenté une série d'études comparatives entre les modèles de Markov : chaînes de Markov cachées utilisées en segmentation d'images nous avons comparé les estimateurs des paramètres, les stratégies Bayésiennes utilisées pour estimer les paramètres de ces modèles, nous avons aussi comparé les chaînes de Markov cachées stationnaires entre eux. Ainsi que, les chaînes de Markov stationnaires et non-stationnaires. En plus nous avons présenté la version floue du modèle chaîne de Markov caché. A partir des résultats obtenus, nous constatons que les estimateurs itératifs donnent les mêmes résultats de segmentation, ils ont le même niveau de complexité, ils ont une complexité linéaire  $O(N)$ . La stratégie Bayésienne MPM est plus performante que MAP, la chaîne de Markov couple est plus performante que tous les autres modèles stationnaires, les modèles de Markov non-stationnaires sont meilleurs que les modèles de Markov stationnaires.

Enfin, la technique de seuillage utilisée pour extraire la région d'intérêt(ROI) n'est pas performante car elle détecte des régions qui n'appartiennent pas à la région d'intérêt(tumeur), juste elles ont la même couleur que cette dernière. Dans le chapitre 4 nous proposerons une approche pour résoudre ce problème.



# Chapitre 3

## Les chaînes de Markov cachées hiérarchiques

### 3.1 Introduction

Dans ce chapitre, nous nous concentrons sur la première contribution que nous proposons pour accélérer le temps d'exécution du modèle classique chaîne de Markov cachée à bruit indépendant, présenté dans le chapitre 2. La chaîne de Markov cachée est parmi les modèles markoviens les plus robustes en segmentation d'images, mais, son inconvénient réside dans la complexité temporelle de calcul des paramètres, surtout lorsque la taille des images est grande et leur nombre est assez large. Pour réduire le temps d'exécution, nous proposons des modèles CMC-BI améliorés basés sur la technique "diviser pour régner", cette technique [110, 111] consiste à diviser le problème global initial en un ensemble de sous-problèmes. Chaque sous-problème est résolu indépendamment des autres et ces sous-solutions sont combinées pour construire la solution globale du problème initial. L'idée principale, de ces approches est de diviser le processus d'estimation et de calcul des paramètres du modèle CMC-BI en plusieurs processus à sous-estimation résolus indépendamment. Ensuite, nous combinons toutes les solutions optimales pour générer la solution finale globale.

Dans la première approche, nous divisons les observations  $Y$  en  $M$  blocs de même taille, chaque bloc est traité indépendamment des autres. Dans la deuxième approche, nous divisons également les observations en  $M$  blocs de même taille, mais, le traitement d'un bloc dépend des résultats des blocs précédents, la solution de chaque bloc courant est obtenue en utilisant la solution de son bloc précédent. Cette technique permet de réduire la complexité du calcul par rapport à la première approche proposée. Les sections suivantes montrent les processus des approches proposées.

Le schéma suivant décrit les différentes étapes suivies pour estimer les paramètres et l'image résultat de segmentation :

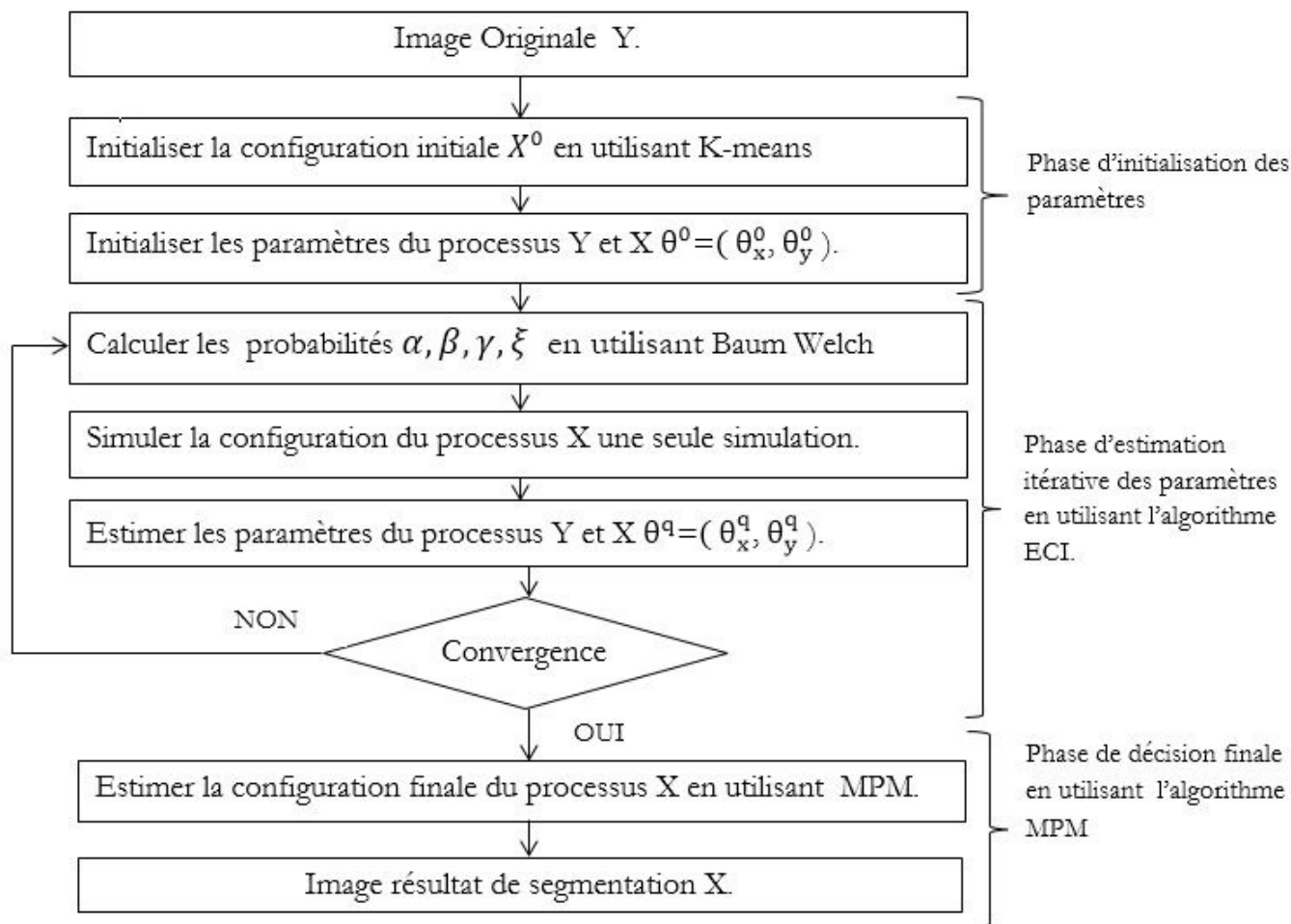


FIGURE 3.1 – Démarche de l'approche standard avant décomposition

### 3.2 Approche 1 : Approche d'estimation indépendante

Il s'agit d'une approche traditionnelle de division, l'approche d'estimation indépendante divise les données à traiter en blocs (sous-processus). Chaque bloc est résolu indépendamment des autres, puis on combine les solutions des blocs, afin de construire la solution globale du problème initial [112].

Soit  $Y = \{Y_1, Y_2, \dots, Y_M\}$  : un ensemble de blocs.

et  $S_G = \{S_1, S_2, \dots, S_M\}$  : un ensemble de solutions.

où :  $S_G$  : est la solution globale.

$M$  : est le nombre de blocs.

*div* : est la taille d'un bloc.

La figure 3.2 montre le scénario de l'approche d'estimation indépendante proposée :

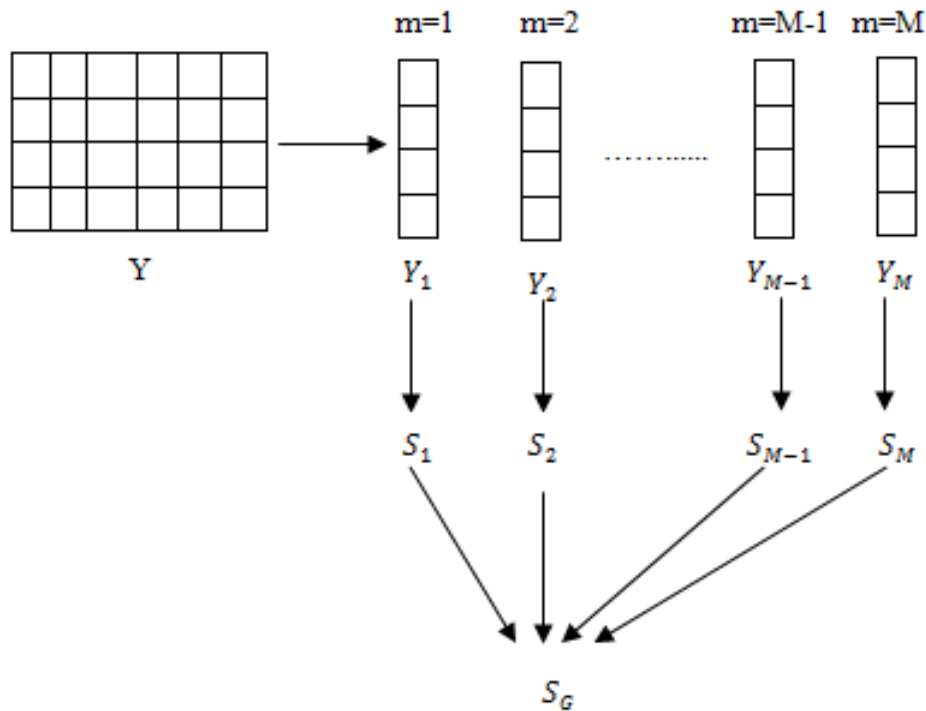


FIGURE 3.2 – Scénario de l’approche d’estimation indépendante

Les observations  $Y$  sont divisées en un ensemble de blocs de même taille. Le nombre de blocs  $M$  est initialisé, soit par la largeur de l’image ou la longueur de l’image ou par un diviseur de la taille totale des observations. Pour estimer les paramètres de chaque bloc, nous suivons la même procédure que l’approche standard. Chaque bloc est traité comme une CMC-BI indépendante. Enfin, nous combinons les configurations finales des blocs, pour obtenir la configuration finale globale de  $X = \cup_{m=1}^M X_m$ . Les étapes suivies par la première approche sont illustrées dans l’algorithme suivant :

---

**Algorithme 24** Les étapes de l’approche estimation indépendante

---

- 1: Initialiser le processus caché  $X^0$  par l’algorithme de K-means ;
  - 2: Initialiser les paramètres de chaque processus  $\theta^0 = (\theta_x^0, \theta_y^0)$  ;
  - 3: Transformer l’image  $Y$  2D en chaîne de Markov 1D en utilisant le parcours ligne par ligne ;
  - 4: Diviser la chaîne obtenue en  $M$  blocs de même taille ;
  - 5: Pour chaque bloc :
    - i-) Calculer les probabilités de Baum Welch ;
    - ii-) Estimer les paramètres et simuler le processus  $X$  jusqu’à convergence ;
    - iii-) Estimer de la configuration finale ;
  - 6: Combiner les configurations finales des blocs et construire l’image segmentée  $X$  ;
-

L'algorithme 3.2 illustre la démarche de l'approche indépendante ; en ajoutant juste une boucle à l'algorithme d'estimation originale pour exécuter les blocs indépendamment d'une manière séquentielle.

---

**Algorithme 25** Démarche de l'algorithme d'estimation indépendante

---

<p><b>ENTRÉES:</b> <math>\theta^0 = (PI^0, A^0, \mu_i^0, (\sigma_i^0)^2)</math></p> <p><b>SORTIES:</b> La configuration finale globale de <math>X</math></p>	<p>11: Calculer les paramètres de processus caché <math>X</math> <math>\theta_x^{q+1}</math> par :</p>
<p>1: Initialisation : <math>m = 1</math></p> <p>2: Initialisation : <math>q = 0</math></p> <p>3: Pour chaque itération <math>q \in Q</math> :</p> <p>4: <b>tantque</b> (<math> \theta^q - \theta^{q+1}  \leq \varepsilon</math>) <b>faire</b></p> <p>5: Calculer <math>\alpha, \beta, \xi, \gamma</math> en utilisant l'algorithme de Baum Welch :</p> <p>6: Initialisation : (<math>n = 1</math>)</p>	$PI^{q+1}(i) = \xi_1^q(i) \forall i \in \Omega \quad (3.7)$ $A^{q+1}(i, j) = \frac{\sum_{n=1}^N \gamma_n^q(i, j)}{\sum_{n=1}^N \xi_n^q(i)} \forall i, j \in \Omega \quad (3.8)$ <p>12: Calculer les paramètres <math>\theta_y^{q+1}</math> des observations <math>Y</math> par :</p> $\mu_i^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}[x_n^q = i]}{\sum_{n=1}^N \mathbf{1}[x_n^q = i]} \forall i \in \Omega \quad (3.9)$ $(\sigma_i^{q+1})^2 = \frac{\sum_{n=1}^N (y_n - \mu_i^{q+1})^2 \mathbf{1}[x_n^q = i]}{\sum_{n=1}^N \mathbf{1}[x_n^q = i]} \forall i \in \Omega \quad (3.10)$
$\alpha_1^q(i) = \frac{PI^q(i) f_i^q(y_1)}{\sum_{j \in \Omega} PI^q(i) f_j^q(y_1)} \forall i \in \Omega \quad (3.1)$ <p>7: Induction : (<math>(n &gt; 1)</math> et <math>(n &lt; div)</math>)</p> $\alpha_{n+1}^q(i) = \frac{f_i^q(y_{n+1}) \sum_{j \in \Omega} \alpha_n^q(j) A^q(i, j)}{\sum_{k \in \Omega} f_k^q(y_{n+1}) \sum_{j \in \Omega} \alpha_n^q(j) A^q(i, j)} \forall i \in \Omega \quad (3.2)$	<p>13: <b>pour</b> (<math>(i = 1; i = K; i++)</math>) <b>faire</b></p> <p>14: <b>pour</b> (<math>(n = 1; n = div; n++)</math>) <b>faire</b></p> <p>15: Calculer la densité gaussienne <math>f_i^q(y_n)</math> de chaque pixel appartient au bloc courant en utilisant l'équation 2.5 ;</p> <p>16: <b>fin pour</b></p> <p>17: <b>fin pour</b></p> <p>18: <math>q = q + 1</math></p> <p>19: Estimer la configuration finale du bloc actuel <math>X_m</math> en utilisant l'algorithme MPM.</p> <p>20: <math>div = m * div</math></p> <p>21: <math>m = m + 1</math></p> <p>22: <b>fin tantque</b></p> <p>23: Combiner les configurations finales des blocs <math>X_m</math> pour construire la configuration finale globale <math>X</math></p>
<p>8: Initialisation : (<math>N = div</math>)</p> $\beta_N^q(i) = 1 \forall i, j \in \Omega \quad (3.3)$ <p>9: Induction : (<math>n &lt; div</math>)</p> $\beta_n^q(i) = \frac{\sum_{j \in \Omega} A^q(i, j) f_j^q(y_{n+1}) \beta_{n+1}^q(j)}{\sum_{k \in \Omega} f_k^q(y_{n+1}) \sum_{j \in \Omega} \alpha_n^q(j) A^q(i, j)} \forall i, j \in \Omega \quad (3.4)$ $\xi_n^q(i) = \alpha_n^q(i) \beta_n^q(i) \forall i, j \in \Omega \quad (3.5)$ $\gamma_n^q(i, j) = \frac{\alpha_n^q(i) A^q(i, j) f_j^q(y_{n+1}) \beta_{n+1}^q(j)}{\sum_{k \in \Omega} f_k^q(y_{n+1}) \sum_{l \in \Omega} \alpha_n^q(l) A^q(l, k)} \forall i, j \in \Omega \quad (3.6)$	<p>10: Simuler le processus caché <math>X_m</math> du bloc actuel <math>Y_m</math> aléatoirement une seule fois ;</p>

---

**Remarque 4.** -Notant que, la division des observations  $y$  en blocs se fait après, la transformation

des observations 2D en vecteur 1D(chaine) en utilisant le parcours ligne par ligne. Ensuite, nous divisons le vecteur 1D en  $M$  blocs de même taille.

-Comme chaque bloc est traité indépendamment des autres, les techniques de parallélisme peuvent être utilisées pour alléger encore les calculs.

### 3.3 Approche 2 : Approche d'estimation dépendante

Contrairement au première approche, la deuxième considère que la solution de chaque bloc de données dépend de la solution du bloc précédent. Les solutions des blocs sont combinées pour construire la solution globale finale du problème initial. La figure suivante présente le scénario de l'approche d'estimation dépendante :

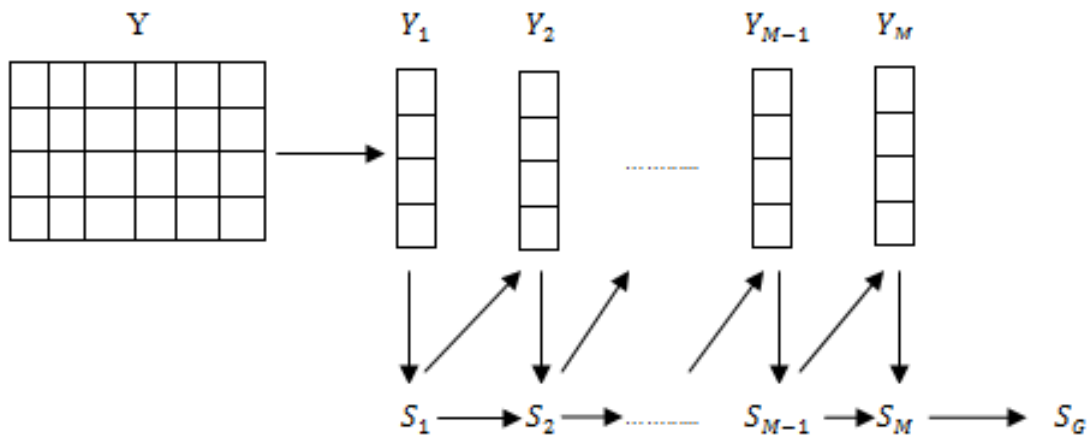


FIGURE 3.3 – Scénario de l'approche d'estimation dépendante

Dans cette approche, nous commençons par diviser l'image  $Y$  en blocs (niveaux), chaque niveau contient un bloc. Le calcul des probabilités d'un bloc dépend du résultat du bloc précédent. Après avoir, calculées les probabilités de tous les blocs, nous combinons ces probabilités pour calculer les paramètres de chaque processus de modèle CMC-BI, nous répétons cette procédure, jusqu'à la convergence de l'algorithme ECI. Enfin, nous prenons les paramètres finaux obtenus pour estimer la configuration finale du processus  $X$  à l'aide de l'algorithme MPM. L'algorithme 26 décrit les étapes suivies par la deuxième approche :

---

**Algorithme 26** Les étapes de l'approche : estimation dépendante

---

- 1: Initialiser le processus caché  $X^0$  par l'algorithme de K-means ;
  - 2: Initialiser les paramètres de chaque processus  $\theta^0 = (\theta_x^0, \theta_y^0)$  ;
  - 3: Transformer l'image  $Y$  en chaîne de Markov en utilisant le parcours ligne par ligne ;
  - 4: Diviser la chaîne obtenue en  $M$  blocs de même taille ;
  - 5: Pour chaque bloc :
    - i-) Calculer les probabilités de Baum-Welch du bloc actuel  $m$  à partir des probabilités du bloc précédent  $m - 1$ .
  - 6: Combiner les probabilités obtenues, et estimer les paramètres globaux du modèle CMC-BI
  - 7: Répéter les étapes 5 et 6 jusqu'à convergence ;
  - 8: Estimer la configuration globale finale du processus  $X$  ;
- 

Considérant que, le temps d'exécution de la phase d'estimation des paramètres est le plus coûteux dans le processus de segmentation des images avec le modèle CMC, nous essayons d'apporter une modification à cette phase . Nous proposons un processus d'estimation qui consiste à estimer les paramètres  $\alpha, \beta, \gamma, \xi$  et à simuler aléatoirement le processus  $X_m$  de chaque bloc. La procédure d'estimation est illustrée dans l'algorithme suivant :

---

**Algorithme 27** Démarche de l'algorithme d'estimation dépendante

---

<p><b>ENTRÉES:</b> <math>M</math> : est le nombre des blocs ;</p> <p><b>ENTRÉES:</b> <math>m</math> : est le numéro du bloc actuel ;</p> <p>1: Initialisation : <math>m = 1</math></p> <p>2: <b>tantque</b> (<math>((m \leq M))</math>) <b>faire</b></p> <p>3:   <b>si</b> (<math>m == 1</math>) <b>alors</b></p> <p>4:     Initialisation : (<math>n = 1</math>)</p> $\alpha_1(i) = \frac{PI(i)f_i(y_1)}{\sum_{j \in \Omega} PI(i)f_j(y_1)} i \in \Omega \quad (3.11)$ <p>5:     Induction : (<math>(n &gt; 1)</math> et (<math>n &lt; div</math>))</p> $\alpha_{n+1}(i) = \frac{f_i(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)} i, j \in \Omega \quad (3.12)$ <p>6:     Initialisation : (<math>N = div</math>)</p> $\beta_N(i) = 1 i \in \Omega \quad (3.13)$ <p>7:     Induction : (<math>n &lt; div</math>)</p> $\beta_n(i) = \frac{\sum_{j \in \Omega} A(i, j) f_j(y_{n+1}) \beta_{n+1}(j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)} i, j \in \Omega \quad (3.14)$ $\xi_n(i) = \alpha_n(i) \beta_n(i) i \in \Omega \quad (3.15)$ $\gamma_n(i, j) = \frac{\alpha_n(i) A(i, j) f_j(y_{n+1}) \beta_{n+1}(j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{l \in \Omega} \alpha_n(l) A(l, k)} i, j \in \Omega \quad (3.16)$ <p>8:     Simuler le processus caché <math>X_1</math> du premier bloc aléatoirement une seule fois ;</p> <p>9:     <math>m++</math></p> <p>10:    <b>finsi</b></p> <p>11:    <b>si</b> (<math>m &gt; 1</math>) <b>alors</b></p>	<p>12:    Initialisation : (<math>n = div + 1</math>)</p> $\alpha_n(i) = \frac{f_i(y_n) \sum_{j \in \Omega} \alpha_{div}(j) A(i, j)}{\sum_{k \in \Omega} f_k(y_n) \sum_{j \in \Omega} \alpha_{div}(j) A(i, j)} i \in \Omega \quad (3.17)$ <p>13:    Induction : (<math>(n &gt; (div + 1))</math> et (<math>n &lt; (m * div)</math>))</p> $\alpha_{n+1}(i) = \frac{f_i(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)} i \in \Omega \quad (3.18)$ <p>14:    Initialisation : (<math>n = (m * div)</math>)</p> $\beta_n(i) = \frac{\sum_{j \in \Omega} A(i, j) f_j(y_n) \beta_{n+1}(j)}{\sum_{k \in \Omega} f_k(y_n) \sum_{j \in \Omega} \alpha_n(j) A(i, j)} i, j \in \Omega \quad (3.19)$ <p>15:    Induction : (<math>n &lt; (m * div)</math>)</p> $\beta_n(i) = \frac{\sum_{j \in \Omega} A(i, j) f_j(y_{n+1}) \beta_{n+1}(j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{j \in \Omega} \alpha_n(j) A(i, j)} i, j \in \Omega \quad (3.20)$ $\xi_n(i) = \alpha_n(i) \beta_n(i) i, j \in \Omega \quad (3.21)$ $\gamma_n(i, j) = \frac{\alpha_n(i) A(i, j) f_j(y_{n+1}) \beta_{n+1}(j)}{\sum_{k \in \Omega} f_k(y_{n+1}) \sum_{l \in \Omega} \alpha_n(l) A(l, k)} i, j \in \Omega \quad (3.22)$ <p>16:    Simuler le processus caché <math>X_m</math> du bloc <math>m</math> aléatoirement une seule fois ;</p> <p>17:    <math>div = m * div</math></p> <p>18:    <math>m++</math></p> <p>19:    <b>finsi</b></p> <p>20:    <b>fin tantque</b></p>
--	--

---

Après avoir calculé les probabilités  $\alpha$ ,  $\beta$ ,  $\xi$ ,  $\gamma$  et simulé aléatoirement les blocs de processus caché  $X = \cup_{m=1}^M X_m$  en utilisant l'algorithme 27, nous combinons ces probabilités pour calculer les paramètres globaux du modèle CMC-BI  $\theta^a = (\theta_x^a, \theta_y^a)$ . Nous testons la convergence de l'algorithme ECI, s'il atteint la convergence, nous prenons les paramètres obtenus et nous estimons la confi-

guration globale finale du processus  $X$  en utilisant l'algorithme MPM, sinon, nous recalculons les probabilités de Baum-Welch et les paramètres jusqu'à convergence. La procédure de la deuxième approche est exposée dans la figure 3.4

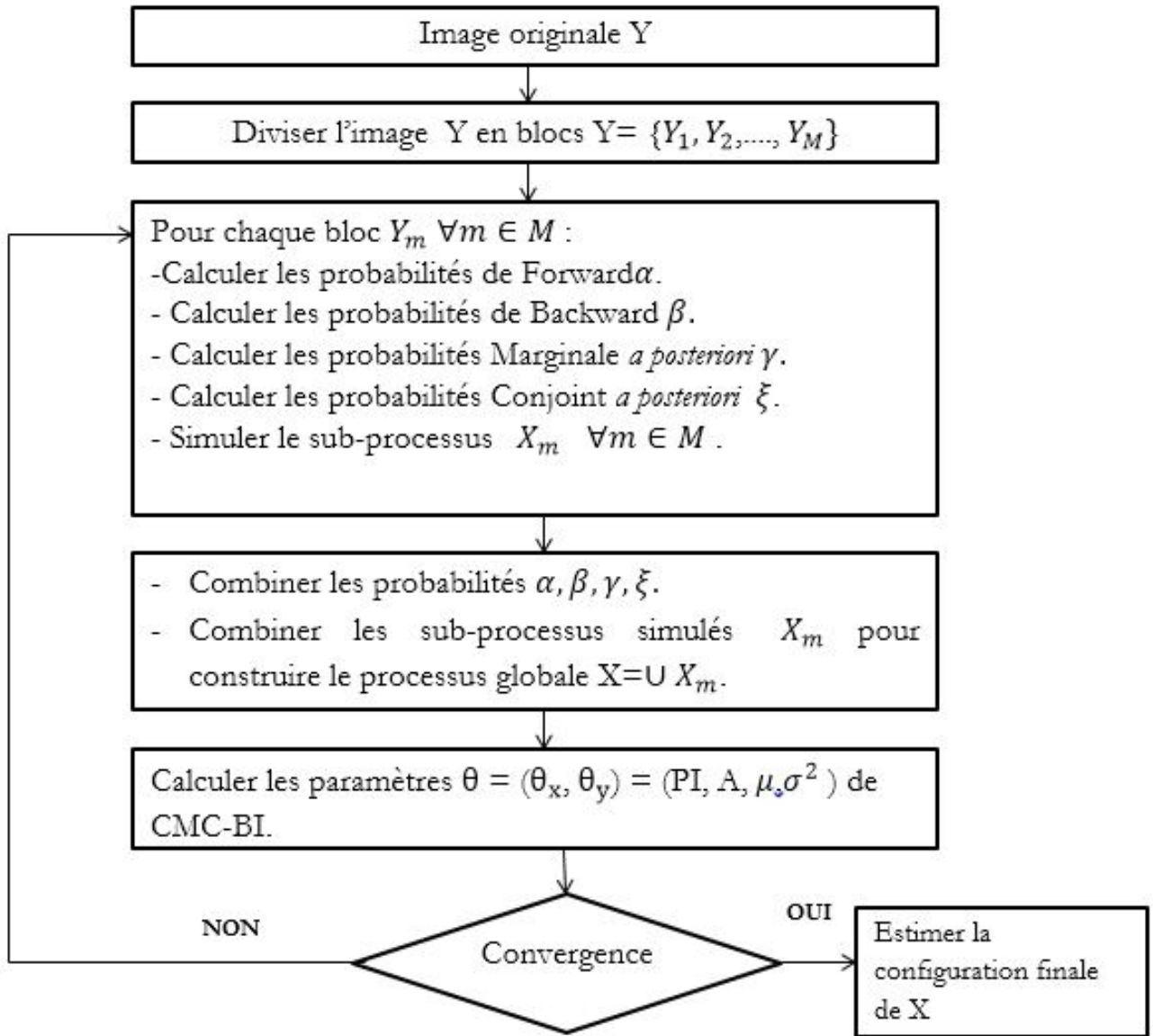


FIGURE 3.4 – Processus d'estimation dépendante : Approche 2

Également, deux variantes de l'approche dépendante (Approche 2-1, Approche 2-2) ont été proposées. Les sous-sections suivantes présentent en détail le processus de ces variantes.

### 3.3.1 Approche 2-1

L'approche 2-1 a la même procédure que l'approche 2, elle utilise l'algorithme 27 seulement pour calculer les probabilités  $\alpha$ ,  $\beta$ ,  $\xi$ ,  $\gamma$  de chaque bloc sans simuler le processus  $X_m$ . Ensuite, elle



combine ces probabilités pour simuler le processus global  $X$  et estimer les paramètres  $\theta^q = (\theta_x^q, \theta_y^q)$  du modèle CMC-BI. elle répète ces étapes, jusqu'à la convergence d'ECI. Enfin, elle estime la configuration globale du processus  $X$ . L'approche 2-1 se déroule comme suit :

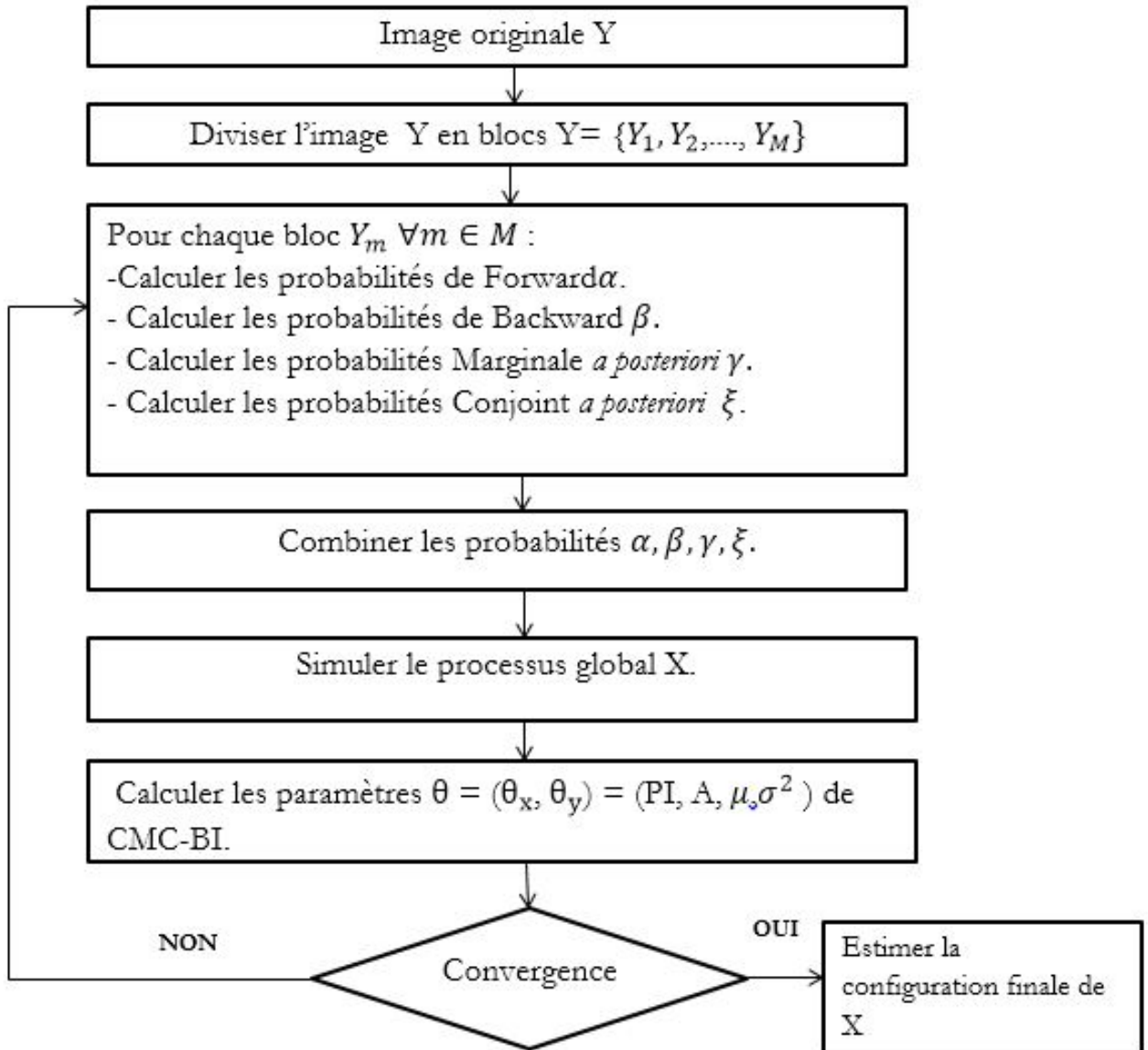


FIGURE 3.5 – Processus d'estimation dépendante : Approche 2-1

### 3.3.2 Approche 2-2

Dans l'approche 2-2, nous utilisons l'algorithme 27 pour calculer les probabilités de forward  $\alpha$ , et de backward  $\beta$  de chaque bloc. Ensuite, nous combinons ces probabilités pour calculer les probabilités marginales *a posteriori*  $\xi$  et les probabilités conjointes *a posteriori*  $\gamma$ , simuler le processus global  $X$ , et calculer les paramètres  $\theta^q = (\theta_x^q, \theta_y^q)$  du modèle CMC-BI. Nous répétons ces calculs jusqu'à la convergence d'ECI. Enfin, nous calculons les probabilités  $\alpha^Q$  et  $\beta^Q$  du processus

global, et nous estimons la configuration finale de processus  $X$ . La figure 3.6 montre la procédure de l'approche 2-2.

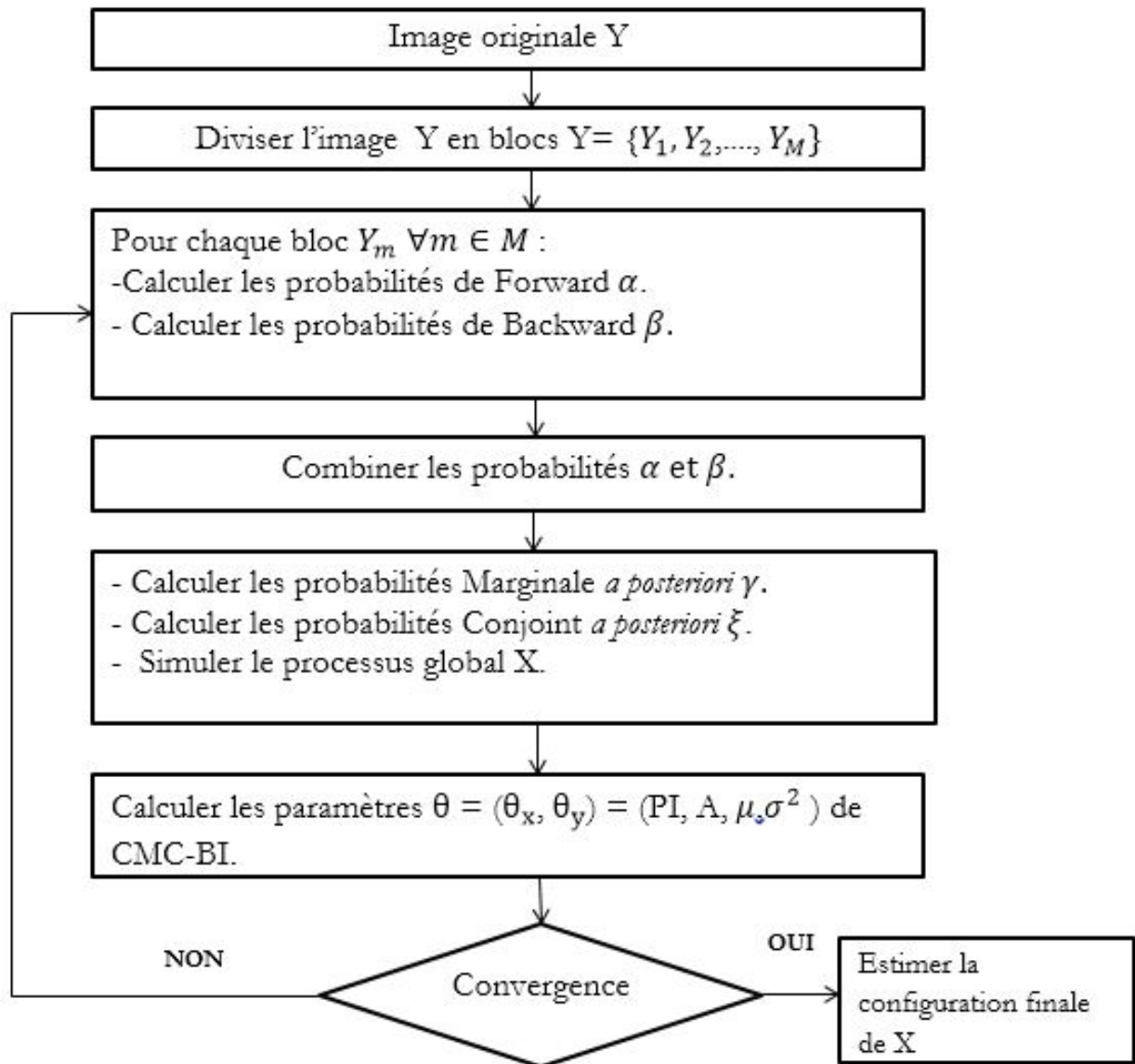


FIGURE 3.6 – Processus d'estimation dépendante : Approche 2-2

**Remarque 5.** - Notant que l'algorithme d'estimation dépendante 27 peut être appliqué avec n'importe quel estimateur comme SEM, MCEM. Ces estimateurs ont un processus d'estimation similaire à celui d'ECI. Pour l'algorithme EM, nous pouvons utiliser l'algorithme 27 sans simuler aléatoirement le processus  $X$  dans chaque itération.

- La différence entre ces approches réside dans la façon dont les probabilités de Baum Welch sont calculées, ainsi que la simulation du processus  $X$  avant ou après combinaison des probabilités.

## 3.4 Résultats et expérimentations

Dans cette section, nous évaluons l'efficacité des approches proposées par rapport à l'approche standard (CMC-BI avant décomposition). La comparaison est faite au niveau de la qualité de segmentation : l'indice de PSNR, l'indice de SSIM, le taux d'erreur, ainsi que, le temps d'exécution et le nombre des itérations pour atteindre la convergence. Nous avons réalisé quatre expériences de segmentation d' $n$  images, dans chaque expérience, nous avons défini le nombre de classes  $K$ , le nombre de blocs  $M$ . Pour toutes les expériences, nous avons suivi la même procédure d'initialisation des paramètres  $\theta^0 = (\theta_x^0, \theta_y^0)$  et du processus  $X^0$  que nous avons suivi dans les expériences présentées dans la section 2.2.3.

Nous avons segmenté, à titre expérimental, quatre images de tailles et de types différents. Dans l'expérience 1, nous avons une image médicale du col de l'utérus, dans l'expérience 2, nous avons une image couleur téléchargée à partir du data set PASCAL VOC2010 [113], dans l'expérience 3, nous disposons d'une image couleur téléchargée à partir du data set ADE 20k disponible dans [114], et dans l'expérience 4 nous avons une image satellitaire normalisée téléchargée à partir du data set ISPRS Potsdam [115].

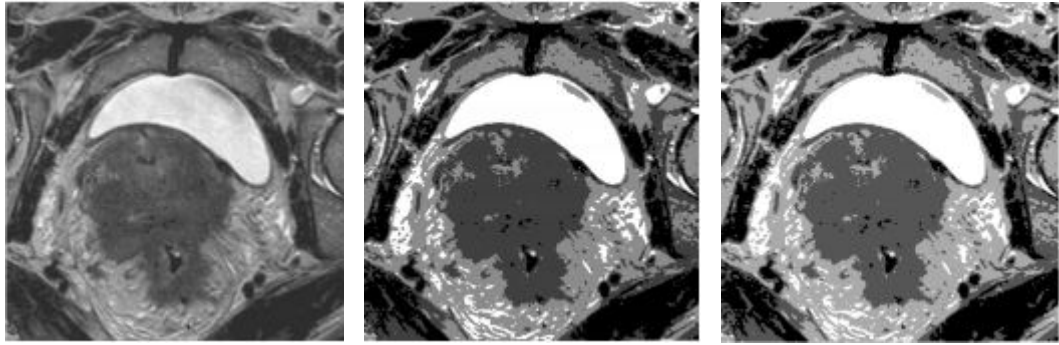
Le tableau 3.1 montre la taille des images, le nombre de classes  $K$  et le nombre de blocs  $M$  résultat de la division.

Expérience	Taille de l'image	Nombre de classes	Nombre de blocs
1	344*344	4	344
2	500*324	4	1620
3	2000*1500	10	30000
4	6000*6000	6	360000

TABLE 3.1 – Les conditions expérimentales des expériences

### 3.4.1 Résultats Visuels

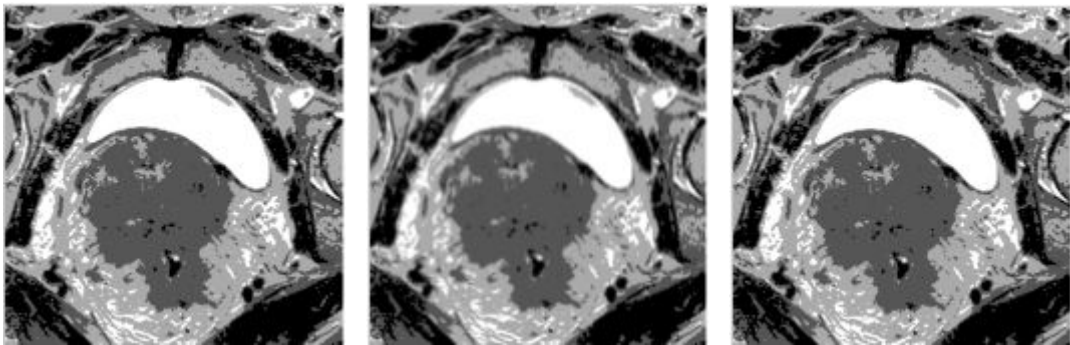
Les figures suivantes décrivent les différentes expériences menées, ainsi que, les résultats de la segmentation obtenus par approche proposée.



(a) Image originale  $Y$

(b) Configuration Initiale  $X^0$

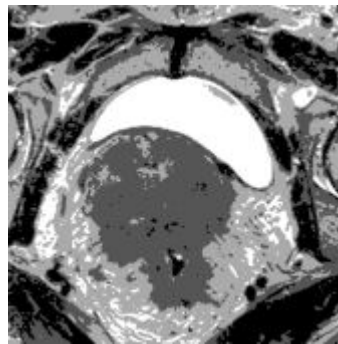
(c) Approche initiale



(d) Approche 1

(e) Approche 2

(f) Approche 2-1



(g) Approche 2-2

FIGURE 3.7 – Résultats de la segmentation de l'image médicale du col de l'utérus



(a) Image originale  $Y$



(b) Configuration Initiale  $X^0$



(c) Approche initiale



(d) Approche 1



(e) Approche 2



(f) Approche 2-1



(g) Approche 2-2

FIGURE 3.8 – Résultats de la segmentation de l'image couleur 1



(a) Image originale  $Y$



(b) Configuration Initiale  $X^0$



(c) Approche initiale



(d) Approche 1



(e) Approche 2

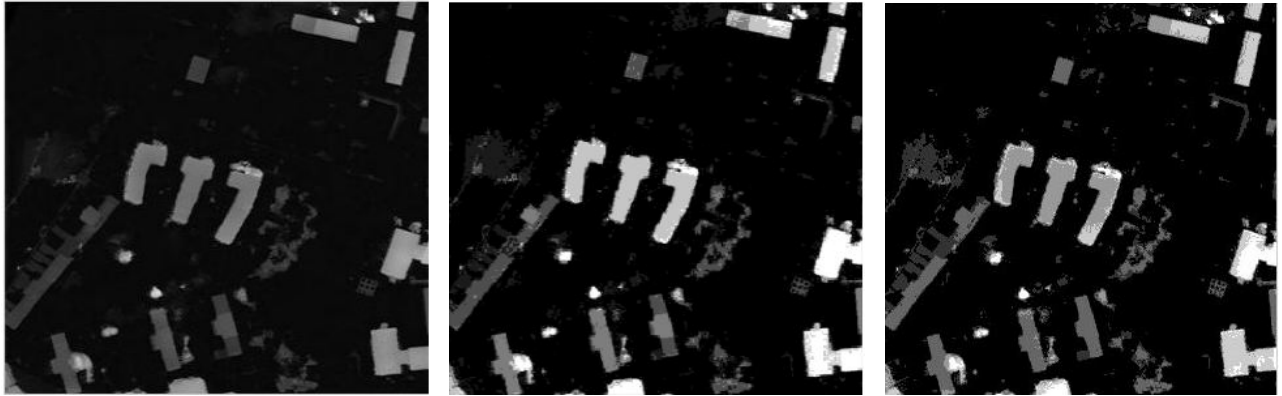


(f) Approche 2-1



(g) Approche 2-2

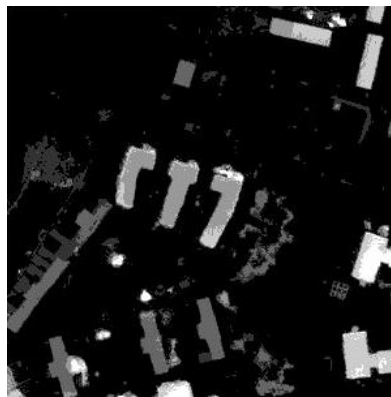
FIGURE 3.9 – Résultats de la segmentation de l'image couleur 2



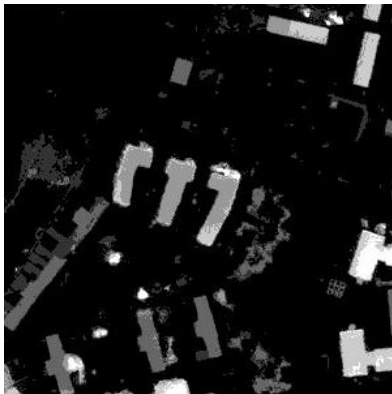
(a) Image originale  $Y$

(b) Configuration Initiale  $X^0$

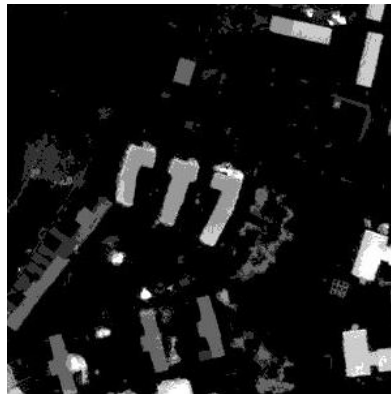
(c) Approche initiale



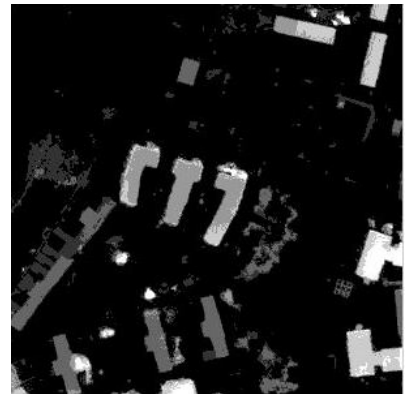
(d) Approche 1



(e) Approche 2



(f) Approche 2-1



(g) Approche 2-2

FIGURE 3.10 – Résultats de la segmentation de l'image satellitaire normalisée

Nous observons que, la qualité des images segmentées obtenues par les approches proposée est

similaire à celle de l’approche originale. Comme première conclusion, nos approches sont concurrentes à l’approche classique en terme de la qualité de segmentation.

### 3.4.2 Résultats Quantitatifs

Pour confirmer les résultats visuels obtenus, nous avons calculé quelques critères d’évaluation. La comparaison est faite au niveau de qualité de segmentation [102] : l’indice PSNR, l’indice SSIM et le taux d’erreur. Les résultats obtenus dans chaque expérience par approche sont résumés dans les tableaux suivants : le tableau 3.2 représente les valeurs de l’indice PSNR, le tableau 3.3 montre les valeurs de l’indice de similarité SSIM et le tableau 3.4 présente les valeurs de taux d’erreur.

Expérience	Approche initiale	Approche 1	Approche 2	Approche 2-1	Approche 2-2
1	29,6172	29,6172	29,6172	29,6172	29,6172
2	34,2024	34,2024	34,2024	34,2024	34,2024
3	35,7621	35,7621	35,7621	35,7621	35,7621
4	35,8622	35,8622	35,8622	35,8622	35,8622

TABLE 3.2 – Les valeurs de l’indice PSNR

Expérience	Approche initiale	Approche 1	Approche 2	Approche 2-1	Approche 2-2
1	0,8196	0,8196	0,8196	0,8196	0,8196
2	0,8934	0,8934	0,8934	0,8934	0,8934
3	0,9145	0,9145	0,9145	0,9145	0,9145
4	0,9350	0,9350	0,9350	0,9350	0,9350

TABLE 3.3 – Les valeurs de l’indice SSIM

Expérience	Approche initiale	Approche 1	Approche 2	Approche 2-1	Approche 2-2
1	15,6690	15,6690	15,6690	15,6690	15,6690
2	26,9111	26,9111	26,9111	26,9111	26,9111
3	16,323	16,323	16,323	16,323	16,323
4	23,7645	23,7645	23,7645	23,7645	23,7645

TABLE 3.4 – Les valeurs de taux d’erreur

Les valeurs des indices (PSNR, SSIM et taux d’erreur) illustrées dans les tableaux précédents sont presque identiques dans chaque expérience effectuée, ce qui confirme nos remarques concernant les résultats visuels obtenus. Nous pouvons l’expliquer par le fait que nous avons utilisé la même procédure de segmentation, ainsi que, les mêmes estimateurs et mêmes valeurs initiales des paramètres pour toutes les approches. De plus, nous avons calculé le temps d’exécution des approches proposées et nous l’avons comparé au temps d’exécution de l’approche initiale de CMC-BI. Le graphe de la



figure 3.11 représente le temps d'exécution en secondes de chaque approche proposée par expérience présentée.

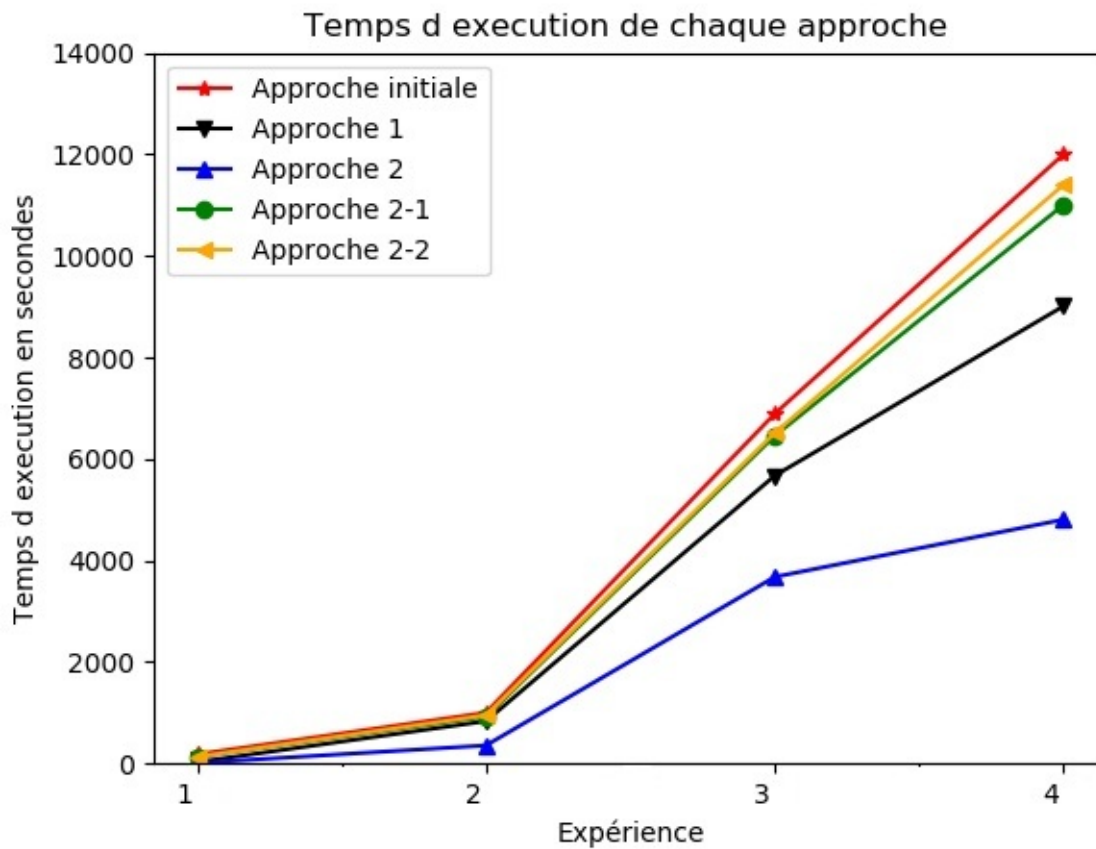


FIGURE 3.11 – Le temps d'exécution en secondes par approche proposée

Comme observation, nous observons que les approches proposées donnent de meilleurs résultats que l'approche initiale ; en fait, le temps d'exécution a considérablement diminué. L'approche d'estimation dépendante (approche 2) est plus performante que les autres, en effet, elle réduit le temps d'exécution de l'ordre de 70% par rapport à l'approche standard, 30% par rapport à l'approche indépendante (approche 1), et environ 50% par rapport aux variantes (approche 2-1 et approche 2-2). Nous notons que le calcul des probabilités de Baum Welch et la simulation de processus  $X$  sont les tâches les plus coûteuses dans le processus d'estimation.

Le temps d'exécution dépend de nombreux facteurs et conditions expérimentales tels que l'espace mémoire, l'unité d'exécution, la vitesse des microprocesseurs, la taille des données, le nombre de classes d'appartenance, le type d'images (gris, couleur), et la convergence des estimateurs de paramètres. A partir des résultats obtenus et paramètres évalués, nous constatons que l'approche standard fournit de meilleurs résultats de segmentation avec de nombreux types d'images filtrées (médicales, couleurs, texturées, satellites), mais elle nécessite un temps d'exécution important, sur-

tout, lorsque la taille des données à traiter est très large.

Également, nous avons étudié la rapidité de convergence des approches proposées. Pour cela, nous avons comparé le nombre d'itérations nécessaires pour atteindre la convergence pour les approches d'estimation dépendantes et l'approche initiale.

Il est difficile de calculer le nombre d'itérations de l'approche d'estimation indépendante, car chaque bloc converge indépendamment des autres et que le nombre de blocs est assez très important. La figure suivante montre le nombre d'itérations nécessaires pour atteindre la convergence de chaque approche proposée par expérience.

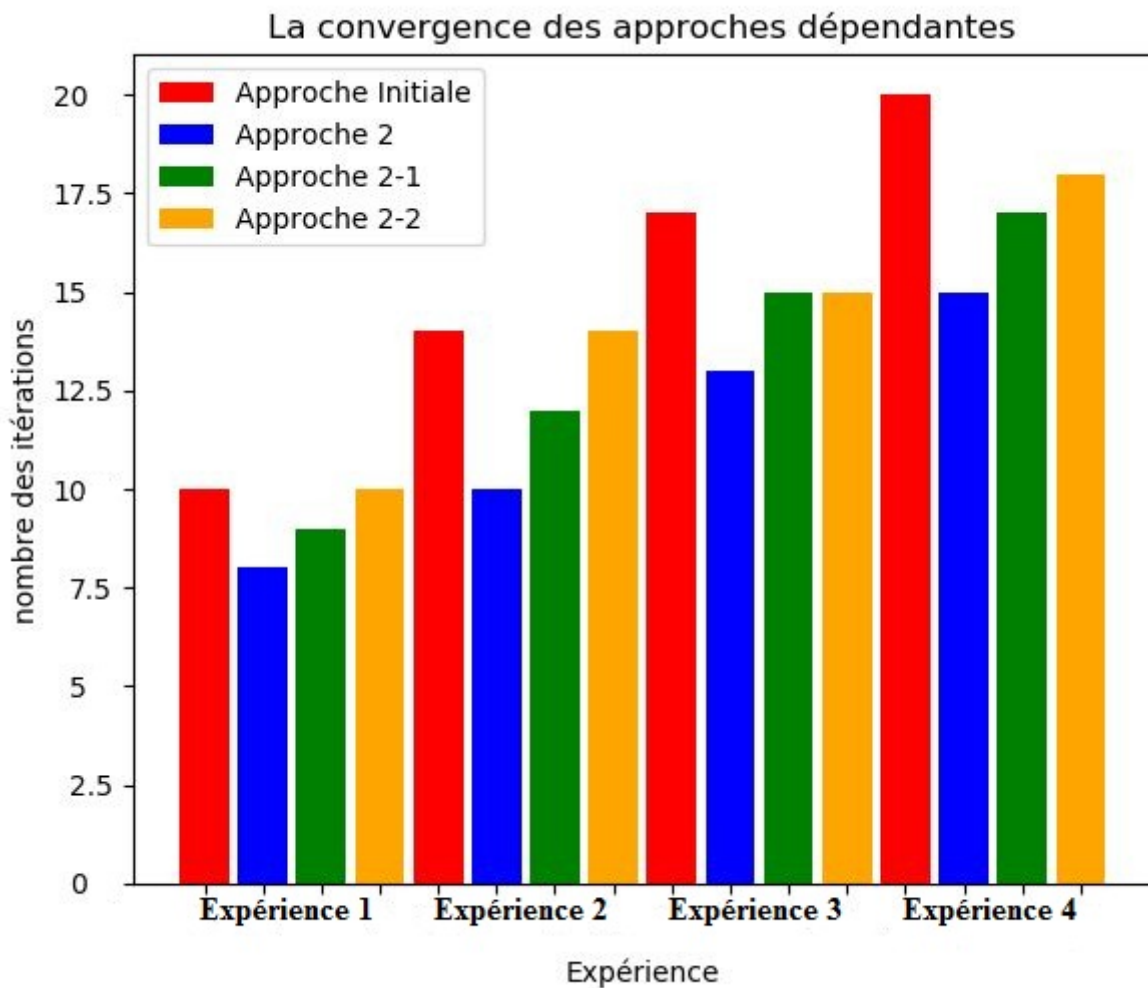


FIGURE 3.12 – Le nombre d'itérations pour atteindre la convergence par approche proposée

D'après la figure 3.12, nous remarquons que l'approche 2 est celle qui nécessite le moins d'itérations pour atteindre la convergence que les autres approches. Cela, confirme une autre fois de plus que cette approche est la plus rapide.

Enfin, nous pouvons conclure que toutes les approches proposées réduisent le temps d'exécution et

le nombre d'itérations nécessaires pour la convergence, et qu'elles conservent la même qualité de segmentation que l'approche standard.

L'approche 2 donne les meilleurs résultats en termes de la qualité de la segmentation, le temps d'exécution et la convergence.

### 3.5 Conclusion

Pour une segmentation d'images rapide et fiable, nous avons proposé quelques approches pour estimer les paramètres du modèle CMC-BI. La première approche est une méthode de division évidente qui divise l'image en un ensemble de blocs (sous-images) de même taille, chaque sous-image est segmentée indépendamment des autres (approche indépendante). La deuxième approche consiste à diviser l'image en blocs de même taille, chaque bloc est exécuté en se basant sur les résultats du bloc précédent (approche dépendante). L'approche 2 utilise l'algorithme 27 pour calculer les probabilités de Baum Welch, et simuler les sous-processus des blocs  $X_m$ . À partir de l'approche 2, nous avons construit deux variantes, la première variante (approche 2-1) utilise l'algorithme 27 de l'approche dépendante pour estimer les probabilités de Baum Welch de chaque bloc. La deuxième variante (approche 2-2) l'utilise uniquement pour calculer les probabilités de Forward Backward de chaque bloc. Pour démontrer les performances des approches proposées, nous les avons comparées avec l'approche initiale du modèle CMC-BI.

Visuellement, il n'y a pas de différence entre les images segmentées obtenues par toutes les approches proposées et l'approche initiale. Ces résultats sont confirmés par les paramètres évalués tels que l'indice PSNR, l'indice SSIM et le taux d'erreur qui sont sensiblement identiques dans toutes les expériences. Cependant, les approches proposées donnent des résultats encourageants en réduisant le temps d'exécution et le nombre d'itérations pour atteindre la convergence. En conclusion, l'approche 2 est meilleure en termes de la qualité de la segmentation, le temps d'exécution et le nombre des itérations pour atteindre la convergence.

Dans le chapitre suivant, notre étude se focalisera sur l'accélération de temps d'exécution du modèle récent chaîne de Markov triplet en appliquant l'approche 2 et les techniques de parallélisme. Nous proposerons aussi une approche pour extraire, avec précision, la région d'intérêt (tumeur) à partir des images cérébrales IRM.

# Chapitre 4

## Les chaines de Markov triplet hiérarchiques

### 4.1 Introduction :

L'imagerie médicale joue un rôle important en pathologie, l'aide au diagnostic des maladies et la détection de la nature des tumeurs (bénignes, pré-malignes ou malignes et ses positions), ainsi que le suivi des cas malades. Elle aide les médecins à choisir la stratégie à suivre pour le traitement et le rétablissement. Ils existent plusieurs techniques d'acquisition et de restitution d'images du corps humain, à savoir, les rayons X [116, 117], la résonance magnétique nucléaire IRM [118], la radioactivité [119]... En outre, les images médicales cérébrales IRM se caractérisent par la symétrie de ses régions qui sont divisées en deux régions symétriques par rapport à un axe verticale, horizontale ou anatomie. Nous pouvons représenter cette symétrie par deux stationnarités.

Dans ce chapitre, nous proposons un système de détection des tumeurs à partir des images cérébrales IRM en utilisant le modèle Chaîne de Markov Triplet(CMT) pour segmenter ces images. Les CMT sont parmi les modèles de Markov les plus robustes en segmentation d'images qui traitent les données non-stationnaires. Pour résoudre ce problème de non-stationnarité, les CMT introduisent un processus auxiliaire  $U$  qui représente la non-stationnarité des données à traiter.

Les CMT estiment deux processus, un caché  $X$  et l'autre auxiliaire  $U$ , et la taille des données à traiter est généralement large, le temps d'exécution de ces modèles est onéreux. Pour cela, nous proposons trois nouvelles approches, la première approche, appelée chaîne de Markov triplet hiérarchique, est fondée sur l'approche d'estimation dépendante présentée dans 3.3. La deuxième utilise les techniques de parallélisme pour estimer, en même temps, le processus caché  $X$  et le processus auxiliaire  $U$ . La dernière approche, appelée chaîne de Markov cachée hiérarchique parallèle, se base sur l'approche d'estimation dépendante et le parallélisme.

Dans les modèles triplet parallèles, nous estimons les deux processus cachés  $X$  et  $U$  en utilisant les techniques de parallélisme [120] : les multiprocessus [121] ou les threads [121]. Nous utilisons l'algorithme d'EM et l'algorithme MPM pour estimer les paramètres et les configurations finales. Après la segmentation des images cérébrales IRM, nous détectons la région d'intérêt en utilisant la technique de seuillage et la morphologie mathématique.

Le schéma suivant montre la démarche de la segmentation d'images cérébrales IRM suivie dans ce travail :

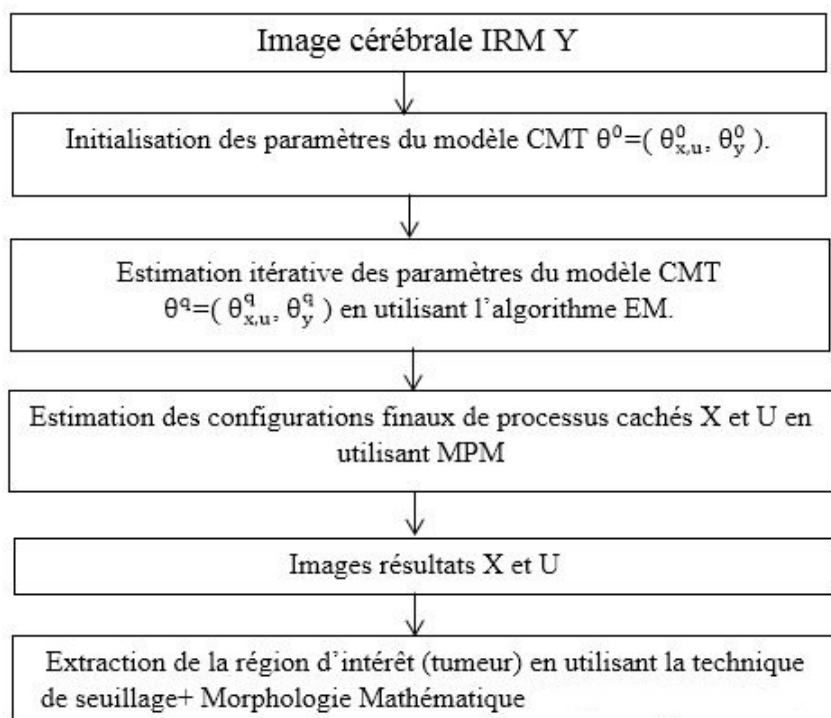


FIGURE 4.1 – Démarche de la segmentation d'images cérébrales IRM

## 4.2 Approche 1 : les chaînes de Markov triplet hiérarchiques

Le modèle Chaîne de Markov Triplet Hiérarchique(CMTH) est un modèle de Markov triplet qui a les mêmes propriétés que le modèle CMT abordé dans 2.5.1, il est basé sur l'approche d'estimation dépendante (l'approche 2 ...). Pour accélérer le temps d'estimation des paramètres  $\theta = (\theta_{x,u}, \theta_y)$ , le modèle CMTH divise l'image en blocs de même taille, le traitement du bloc courant dépend du bloc précédent, après le traitement de chaque bloc, nous combinons leurs solutions pour obtenir la solution finale. La démarche de segmentation d'images avec le modèle CMTH est présentée dans l'algorithme suivant :

---

**Algorithme 28** Les étapes de l'approche d'estimation dépendante adaptée au modèle CMTH

---

- 1: Initialiser le processus caché  $X^0$  en utilisant l'algorithme de K-means ;
  - 2: Initialiser les paramètres de chaque processus  $\theta^0 = (\theta_v^0, \theta_y^0)$  ;
  - 3: Transformer l'image  $Y$   $2D$  à une chaîne de Markov  $1D$  en utilisant le parcours ligne par ligne ;
  - 4: Diviser la chaîne obtenue en  $M$  blocs de même taille ;
  - 5: Calculer les probabilités de chaque bloc  $m$  à partir des probabilités du bloc précédent  $m - 1$ .  
Combiner ces probabilités, et estimer les paramètres de modèle CMT, répéter ces phases, jusqu'à convergence ;
  - 6: Estimer les configurations globales des processus cachés  $X$  et  $U$ .
- 

L'algorithme suivant montre la démarche de l'approche d'estimation dépendante adaptée au modèle CMTH :

---

**Algorithme 29** L'approche d'estimation dépendante adaptée au modèle CMTH

---

<b>ENTRÉES:</b> $M$ : est le nombre de blocs ;	10: <b>si</b> ( $m > 1$ ) <b>alors</b>
<b>ENTRÉES:</b> $m$ : est le bloc actuel ;	11: Initialisation : ( $n = div + 1$ )
<b>ENTRÉES:</b> $div$ : est la taille du bloc ;	
1: Initialisation : $m = 1$	$\alpha_n(v_n) = \sum_{x_{n-1}} \alpha_{div}(v_{n-1}) \cdot p(t_n   t_{n-1}) \quad (4.7)$
2: <b>tantque</b> ( $((m \leq M))$ ) <b>faire</b>	
3: <b>si</b> ( $m == 1$ ) <b>alors</b>	12: Induction : ( $(n > (div + 1))$ et ( $n < (m * div)$ ))
4: Initialisation : ( $n = 1$ )	$\alpha_n(v_n) = \sum_{x_{n-1}} \alpha_{n-1}(v_{n-1}) \cdot p(t_n   t_{n-1}) \quad (4.8)$
	$\alpha_1(v_1) = p(t_1) \quad (4.1)$
5: Induction : ( $(n > 1)$ et ( $n < div$ ))	
$\alpha_n(v_n) = \sum_{x_{n-1}} \alpha_{n-1}(v_{n-1}) \cdot p(t_n   t_{n-1}) \quad (4.2)$	13: Initialisation : ( $n = (m * div)$ )
6: Initialisation : ( $n = div$ )	$\beta_n(v_n) = \sum_{x_{n+1}} \beta_{n+1}(v_{n+1}) \cdot p(t_{n+1}   t_n) \quad (4.9)$
	$\beta_N(v_N) = 1 \quad (4.3)$
7: Induction : ( $n < div$ )	14: Induction : ( $n < (m * div)$ )
$\beta_n(v_n) = \sum_{x_{n+1}} \beta_{n+1}(v_{n+1}) \cdot p(t_{n+1}   t_n) \quad (4.4)$	$\beta_n(v_n) = \sum_{x_{n+1}} \beta_{n+1}(v_{n+1}) \cdot p(t_{n+1}   t_n) \quad (4.10)$
$\xi_n(v_n) = \alpha_n(v_n) \cdot \beta_n(v_n) \quad (4.5)$	$\xi_n(v_n) = \alpha_n(v_n) \cdot \beta_n(v_n) \quad (4.11)$
$\gamma_n(v_n, v_{n+1}) = \alpha_n(v_n) \cdot p(t_n   t_{n-1}) \cdot \beta_{n+1}(v_{n+1}) \quad (4.6)$	$\gamma_n(v_n, v_{n+1}) = \alpha_n(v_n) \cdot p(t_n   t_{n-1}) \cdot \beta_{n+1}(v_{n+1}) \quad (4.12)$
8: $m++$	15: $div = m * div$
9: <b>fin</b>	16: $m++$
	17: <b>fin</b>
	18: <b>fin tantque</b>

---

A chaque itération  $q$ , nous calculons ces probabilités en utilisant l'algorithme 29, nous combinons ces probabilités et nous calculons les paramètres  $V^q, \mu_i^q, (\sigma_i^q)^2$  en utilisant les equations 2.93, 2.94 et 2.95, nous répétons ces étapes jusqu'à convergence.

Après la convergence, nous estimons les configurations finales du processus  $X$  et  $U$  en utilisant les equations 2.96 et 2.97.

La figure ci-dessous résume la démarche de la première approche proposée :

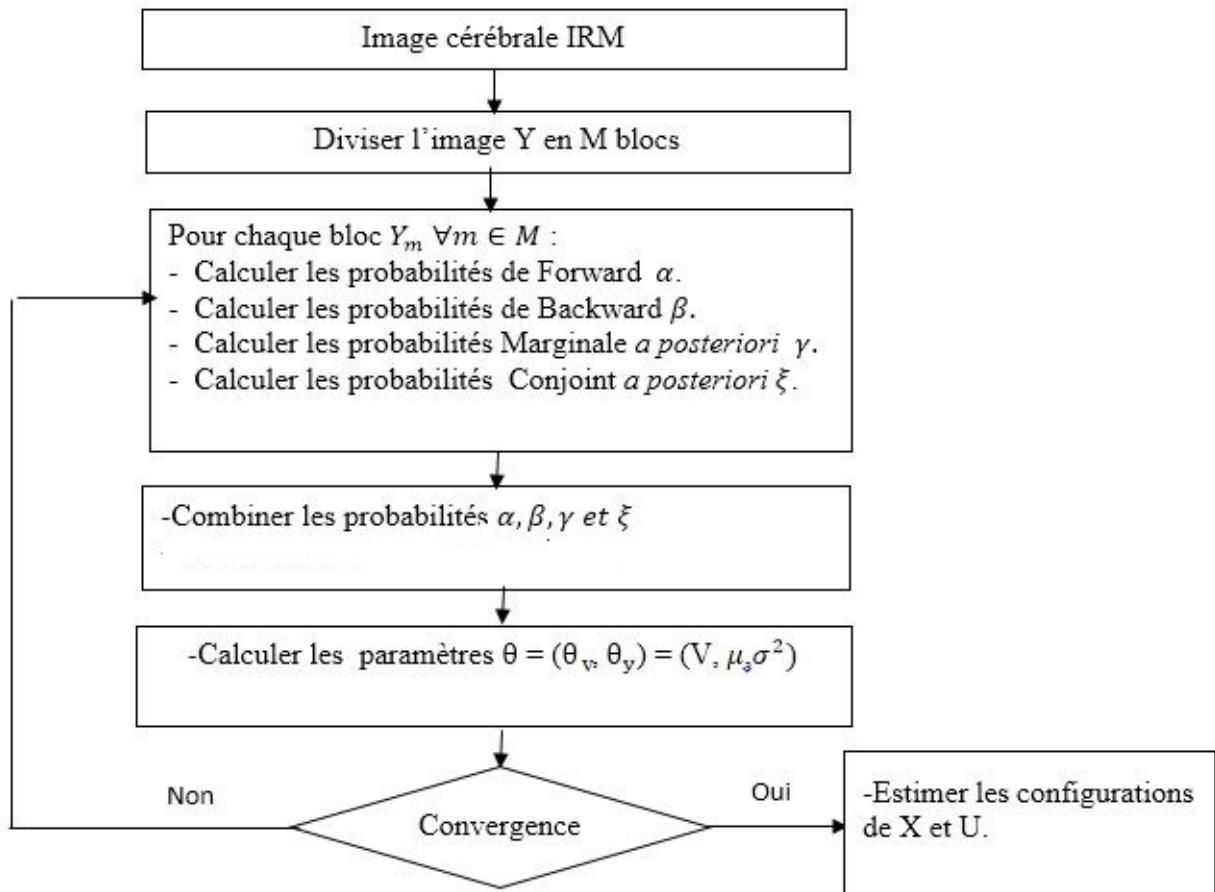


FIGURE 4.2 – Démarche de l'approche 1

### 4.3 Approche 2 : les chaînes de Markov triplet parallèles

La Chaîne de Markov Triplet Parallèle (CMTP) estime les processus cachés  $X$  et  $U$  en même temps en utilisant les techniques de parallélisme.

Nous utilisons deux mécanismes de parallélisme tels que les multiprocessus et les threads.

L'algorithme suivant montre la démarche de modèle CMTP :

---

**Algorithme 30** Les étapes de l'approche adaptée au modèle CMTP

---

- 1: Initialiser le processus caché  $X^0$  en utilisant la méthode de K-means ;
  - 2: Initialiser les paramètres de chaque processus  $\theta^0 = (\theta_v^0, \theta_y^0)$  ;
  - 3: Transformer l'image  $Y$  2D à une chaîne de Markov 1D en utilisant le parcours ligne par ligne ;
  - 4: Calculer les paramètres du modèle CMT en utilisant l'algorithme EM ;
  - 5: Estimer les configurations des processus cachés  $X$  et  $U$  en parallèle.
- 

Le parallélisme [122] est une technique qui sert à exécuter les parties de code d'un programme multitâche en même temps, afin d'accélérer le temps d'exécution. Il existe plusieurs techniques de



parallélisme à savoir, le parallélisme à mémoire partagée OpenMP (Open Multi-Processing) [123], le parallélisme à mémoire distribuée MPI(Message Passing Interface) [124], les threads, les multiprocessus...

Nous utilisons deux techniques de parallélisme plus simples tels que les multiprocessus et les threads. Un processus est un programme en cours d'exécution, il possède son propre espace mémoire : ses registres et ses piles, ses variables et son propre processus virtuel.

Un thread est une sous-tache interne du processus, il partage le même espace mémoire avec les autres unités d'exécution (threads) au sein d'un même processus, ainsi que, les même variables. Il occupe moins de ressources système qu'un processus

Ci-dessous, nous présentons les Pseudo-codes des multiprocessus et threads programmés avec python [125] pour estimer en parallèle les processus  $X$  et  $U$ . Les fonctions `calculerX` et `calculerU` servent à estimer  $X$  et  $U$  à partir du processus  $V$ . Nous utilisons le package OpenCV [126] pour traiter les images sous python, le package `multiprocessing` [127] a été utilisé pour le parallélisme avec les multiprocessus, pour le parallélisme avec les threads, nous importons le package `threading` [127]. nous utilisons deux processus ou deux threads, un pour estimer le processus  $X$  et l'autre pour estimer  $U$  qui s'exécutent en parallèle.

```
import cv2
import numpy as np
import time
import multiprocessing
t = time.time()
lock = multiprocessing.Lock()
p1 = multiprocessing.Process(target=CalculerX(V), args=(lock,V))
p2 = multiprocessing.Process(target=CalculerU(V), args=(lock,V))
p1.start()
p2.start()
p1.join()
p2.join()
print("done in", time.time() - t)
print("finish")
```

FIGURE 4.3 – Pseudo-code python de multiprocessus

```

import cv2
import numpy as np
import time
import threading
my_lock=threading.RLock()
t = time.time()
with my_lock:
    t1 = threading.Thread(target=calculerX(V), args=[V])
    t2 = threading.Thread(target=calculerU(V), args=[V])
    t1.start()
    t2.start()
    t1.join()
    t2.join()
print("done in", time.time() - t)
print("finish")

```

FIGURE 4.4 – Pseudo-code python de threads

Le schéma de la figure 4.5 montre la démarche de la deuxième approche proposée :

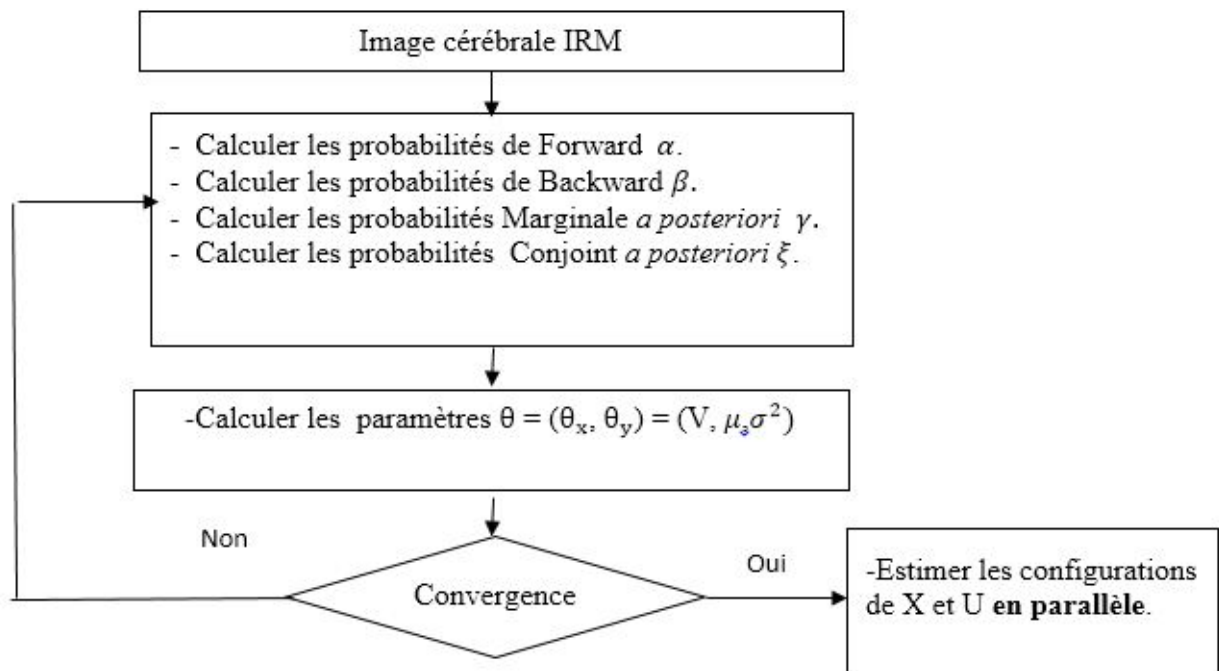


FIGURE 4.5 – Démarche de l'approche 2

## 4.4 Approche 3 : les chaines de Markov triplet hiérarchiques parallèles

Dans ce modèle, nous utilisons l'approche algorithmique "diviser pour régner" et les techniques de parallélisme. La Chaîne de Markov Triplet Hiérarchique Parallèle(CMTHP) utilise l'algorithme 29 pour estimer les paramètres, et les techniques de parallélisme : les multiprocessus ou les threads pour estimer les processus cachés  $X$  et  $U$  en exécutant 4.3 et 4.4.

---

**Algorithme 31** Les étapes de l'approche adaptée au modèle CMTHP

---

- 1: Initialiser le processus caché  $X^0$  en utilisant la méthode de K-means ;
  - 2: Initialiser les paramètres de chaque processus  $\theta^0 = (\theta_v^0, \theta_y^0)$  ;
  - 3: Transformer l'image  $Y$   $2D$  à une chaîne de Markov  $1D$  en utilisant le parcours ligne par ligne ;
  - 4: Diviser la chaîne obtenue en  $M$  blocs de même taille ;
  - 5: Calculer les probabilités de chaque bloc  $m$  à partir des résultats du bloc précédent  $m - 1$ .  
Combiner ces probabilités, et, estimer les paramètres du modèle CMT, répéter ces phases, jusqu'à convergence ;
  - 6: Estimer les configurations des processus cachés  $X$  et  $U$  en parallèle.
- 

La figure 4.4 présente la démarche de la deuxième approche :

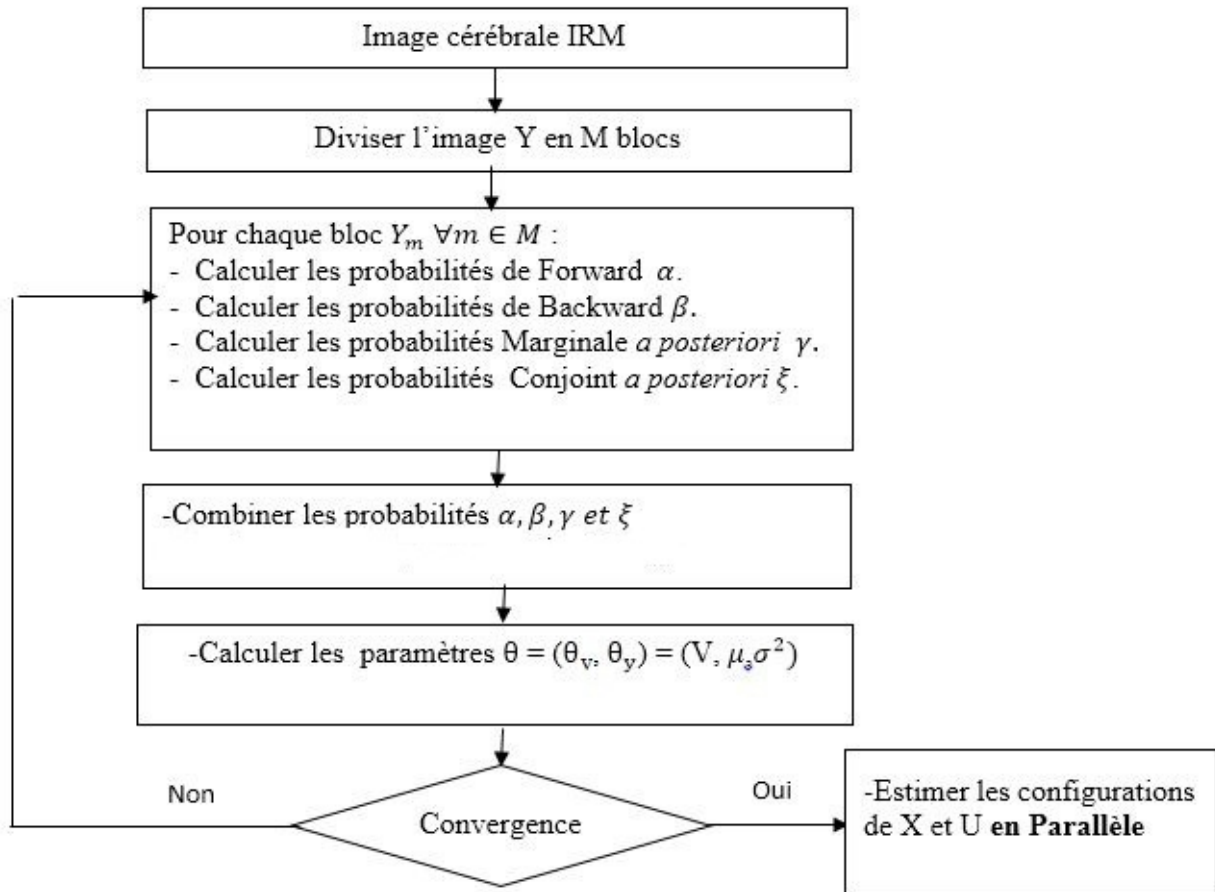


FIGURE 4.6 – Démarche de l'approche 3

## 4.5 Extraction de la région d'intérêt : Approche proposée

Comme nous l'avons déjà remarqué dans le chapitre 2, en utilisant la technique de seuillage la région d'intérêt n'est pas extraite avec précision, il y a des régions qui n'appartiennent pas aux régions d'intérêt qui apparaissent dans l'image résultat, ce qui représente un problème lors du diagnostic de la nature du tumeur. Pour cela, nous présentons dans ce paragraphe une approche d'extraction de la région d'intérêt qui utilise la technique de seuillage suivie par la Morphologie Mathématique(MM). Nous appliquons les quatre opérations de MM afin de découvrir la plus performante à détecter la tumeur.

### a. Technique de seuillage

Le seuillage d'images est une technique simple de segmentation d'images, c'est un moyen de créer une image binaire à partir d'une image niveau de gris ou couleur. Généralement, l'objectif est de séparer les pixels de premier-plan des pixels d'arrière-plan pour faciliter le traitement de

l'image. Le principe de seuillage est très simple, il faut définir un seuil  $\Upsilon$  choisi par l'utilisateur, parcourir les pixels de l'image  $X$ , comparer le pixel  $X(i, j)$  avec le seuil et affecter le pixel  $X(i, j)$  à une région selon le résultat de comparaison. L'algorithme suivant présente le seuillage utilisé dans ce travail. Les pixels qui appartiennent à la même région ont la même valeur.

---

**Algorithme 32** L'algorithme de seuillage(binarisation)

---

**ENTRÉES:**  $X$  : Image d'entrée niveau de gris

**ENTRÉES:**  $Np, Mp$  : Coordonnées des pixels

**ENTRÉES:**  $\Upsilon$  : Seuil

**SORTIES:**  $X$  : Image binaire

```

1: pour ( $i = 1; i = Np; i ++$ ) faire
2:   pour ( $j = 1; j = Mp; j ++$ ) faire
3:     si ( $X(i, j) \leq \Upsilon$ ) alors
4:        $X(i, j) = 1$ 
5:     sinon
6:        $X(i, j) = 0$ 
7:     finsi
8:   fin pour
9: fin pour

```

---

## b. Morphologie Mathématique

La morphologie mathématique a été introduite par Matheron [128] pour l'analyse de la structure géométrique du métal et des échantillons géologiques. Elle a été employée par Serra [129] pour l'analyse d'images. L'idée de base de la MM est d'étudier un ensemble d'éléments en utilisant un autre ensemble appelé Élément Structuré (SE) le rôle de l'élément structuré est de vérifier s'il est à la frontière ou s'il est inclut dans l'ensemble initial à chaque position, puis nous construisons l'ensemble résultat. La MM est basée sur quatre opérations morphologiques, à savoir, la dilatation, l'érosion, l'ouverture et la fermeture comme détaillé ci-dessous. Soit  $A(x, y)$  : une image bidimensionnelle niveau de gris et  $B$  un élément structuré

- La dilatation appliquée sur  $A(x, y)$  par l'élément structuré  $B(s, t)$  est définie par :

$$(A \oplus B)(x, y) = \max\{A(x - s, y - t) + B(s, t)\} \quad (4.13)$$

- L'érosion de  $A(x, y)$  par  $B(s, t)$  est définie par :

$$(A \ominus B)(x, y) = \max\{A(x + s, y + t) + B(s, t)\} \quad (4.14)$$

- Les opérateurs d'ouverture et de fermeture appliqués sur  $A(x, y)$  par  $B(s, t)$  sont définies par la différence entre l'image et son ouverture représentée respectivement comme suit :

$$(A \circ B)(x, y) = (A \ominus B) \oplus B \quad (4.15)$$

$$(A \cdot B)(x, y) = (A \oplus B) \ominus B \quad (4.16)$$

**Remarque 6.** — *L'ouverture est une érosion de  $A(x, y)$  suivit d'une dilatation appliquée sur le résultat d'érosion.*

- *La fermeture est une dilatation de  $A(x, y)$  suivit d'une érosion appliquée sur le résultat de dilatation.*
- *L'ouverture d'une image est l'ensemble des régions du premier-plan d'une image qui correspondent à un élément structuré particulier.*
- *La fermeture d'une image est l'ensemble des régions d'arrière-plan d'une image qui correspondent à un élément structuré particulier.*

## 4.6 Résultats et expérimentations

Pour montrer les performances de nos contributions, nous avons effectué une série d'expériences sur des images cérébrales IRM niveaux de gris de taille  $256 * 256$ . Après la segmentation des images par les approches proposées, nous avons appliqué notre méthode d'extraction de la région d'intérêt (tumeur) sur les images segmentées. Nous avons appliqué la technique de seuillage, ensuite les opérations de MM.

Nous avons, également, segmenté les images en trois classes  $K = 3$ , au niveau de tissu (Matière Grise(MG), Matière Blanche(MB), Liquide Céphalo-Rachidien(LCR)). Certes que les images cérébrales se caractérisent par la symétrie de ses régions, nous trouverons que les régions des images cérébrales sont symétriques par rapport à un axe verticale ou horizontale, les régions de ces images sont non-stationnaires, donc le nombre de stationnarité est  $M = 2$ .

Dans les modèles triplet hiérarchiques, nous avons divisé les images en 256 blocs chaque bloc est de taille 256. Dans toutes les expériences, nous avons suivi cette démarche pour initialiser les paramètres des modèles proposés.

$$\mathbf{V}^0 = \begin{pmatrix} 0.5 & 0.25 & 0.25 & 0 & 0 & 0 \\ 0.25 & 0.5 & 0.25 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \end{pmatrix}$$

Nous avons utilisé l'algorithme de K-means pour estimer le processus caché initial  $X^0$ . Nous avons calculé la moyenne et la variance à partir de cette initialisation.

Nous avons comparé le modèle CMT avec le modèle CMC-BI, pour initialiser les paramètres de ce dernier, nous avons suivi la même démarche de [2.2.3](#)

### 4.6.1 Résultats Visuels

Certes que les approches proposées donnent les même résultats de segmentation que l'approche initiale, nous avons présenté juste les résultats de l'approche initiale et nous avons comparé les résultats obtenus avec ceux du modèle CMC-BI .Les figures suivantes décrivent les différentes expériences effectuées, ainsi que les résultats de segmentation obtenus.

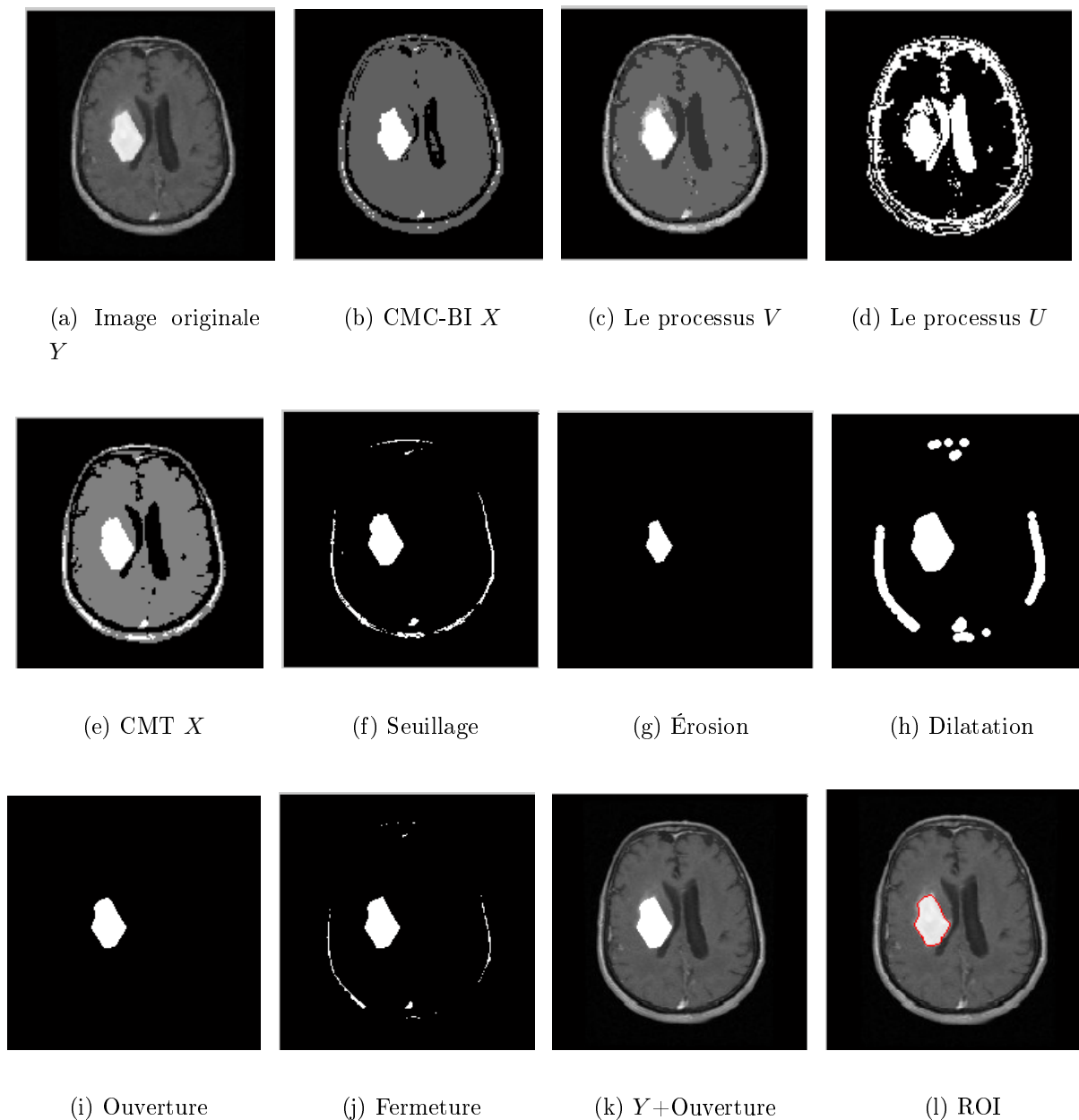
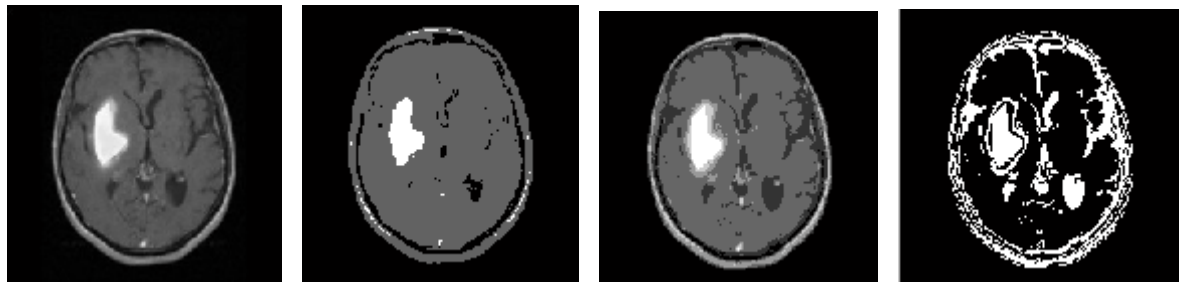


FIGURE 4.7 – Résultat de segmentation 1.

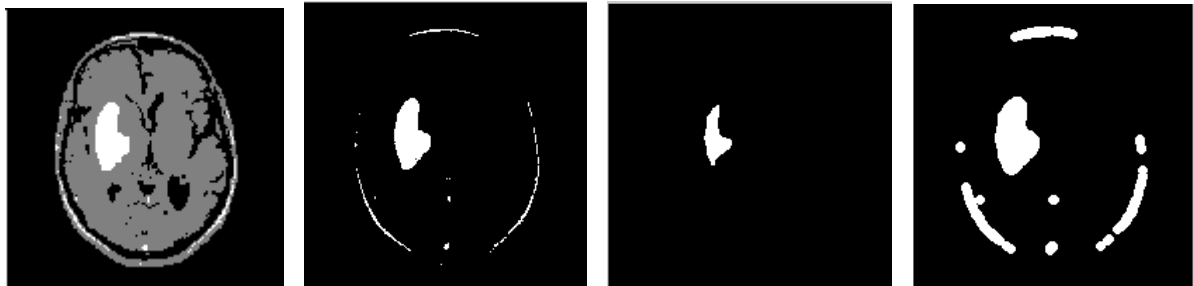


(a) Image originale  
 $Y$

(b) CMC-BI  $X$

(c) Le processus  $V$

(d) Le processus  $U$

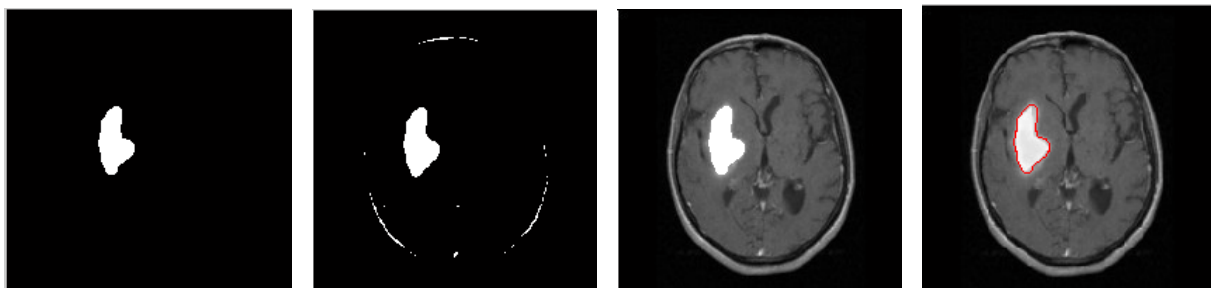


(e) CMT  $X$

(f) Seillage

(g) Érosion

(h) Dilatation



(i) Ouverture

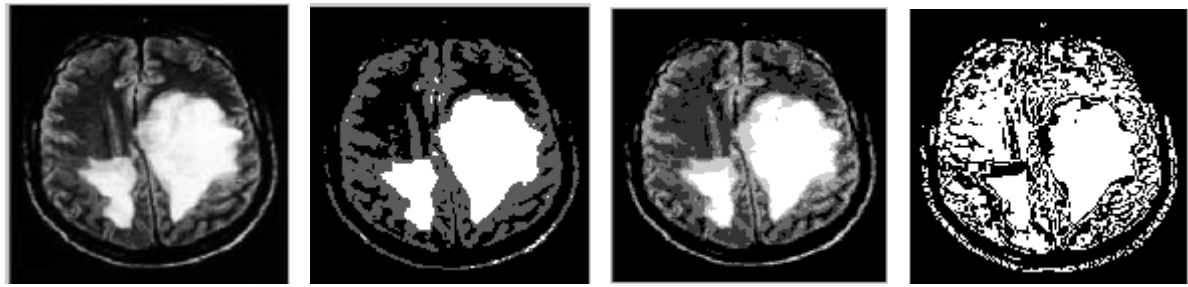
(j) Fermeture

(k)  $Y +$  Ouverture

(l) ROI

FIGURE 4.8 – Résultat de segmentation 2.





(a) Image originale  
 $Y$

(b) CMC-BI  $X$

(c) Le processus  $V$

(d) Le processus  $U$

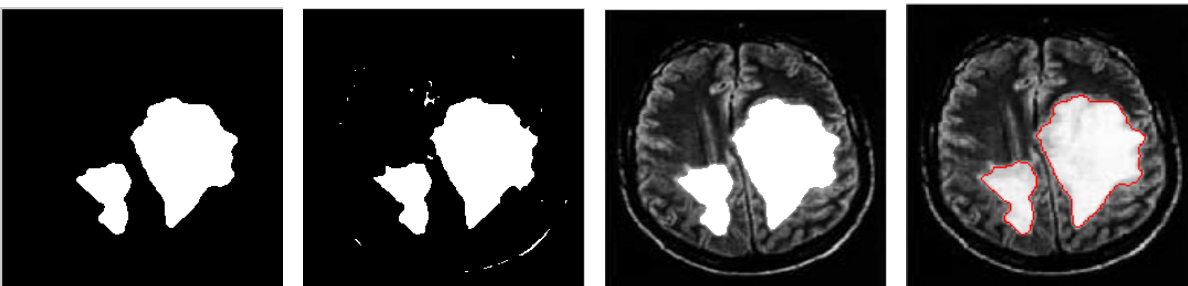


(e) CMT  $X$

(f) Seillage

(g) Érosion

(h) Dilatation



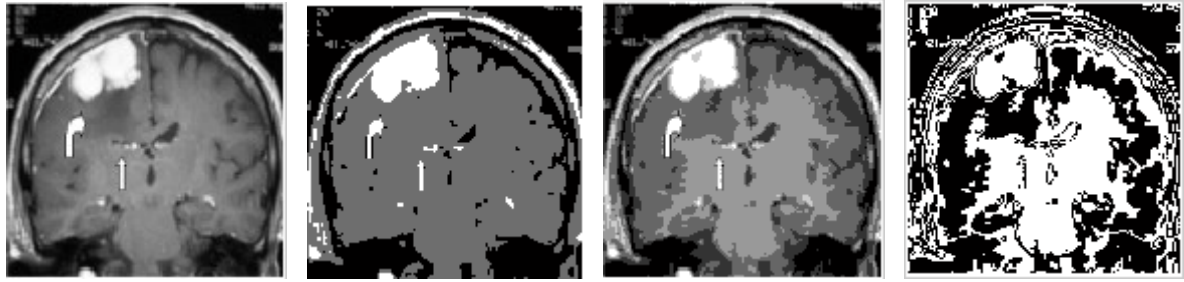
(i) Ouverture

(j) Fermeture

(k)  $Y+$  Ouverture

(l) ROI

FIGURE 4.9 – Résultat de segmentation 3.



(a) Image originale  
 $Y$

(b) CMC-BI  $X$

(c) Le processus  $V$

(d) Le processus  $U$

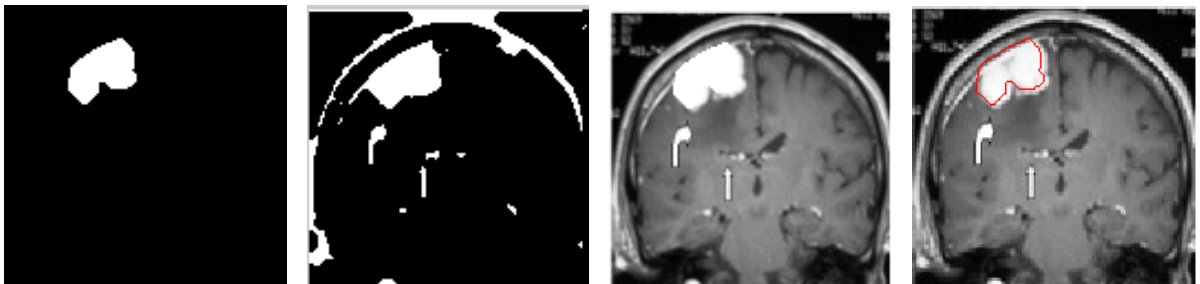


(e) CMT  $X$

(f) Seuillage

(g) Érosion

(h) Dilatation



(i) Ouverture

(j) Fermeture

(k)  $Y+$  Ouverture

(l) ROI

FIGURE 4.10 – Résultat de segmentation 4.

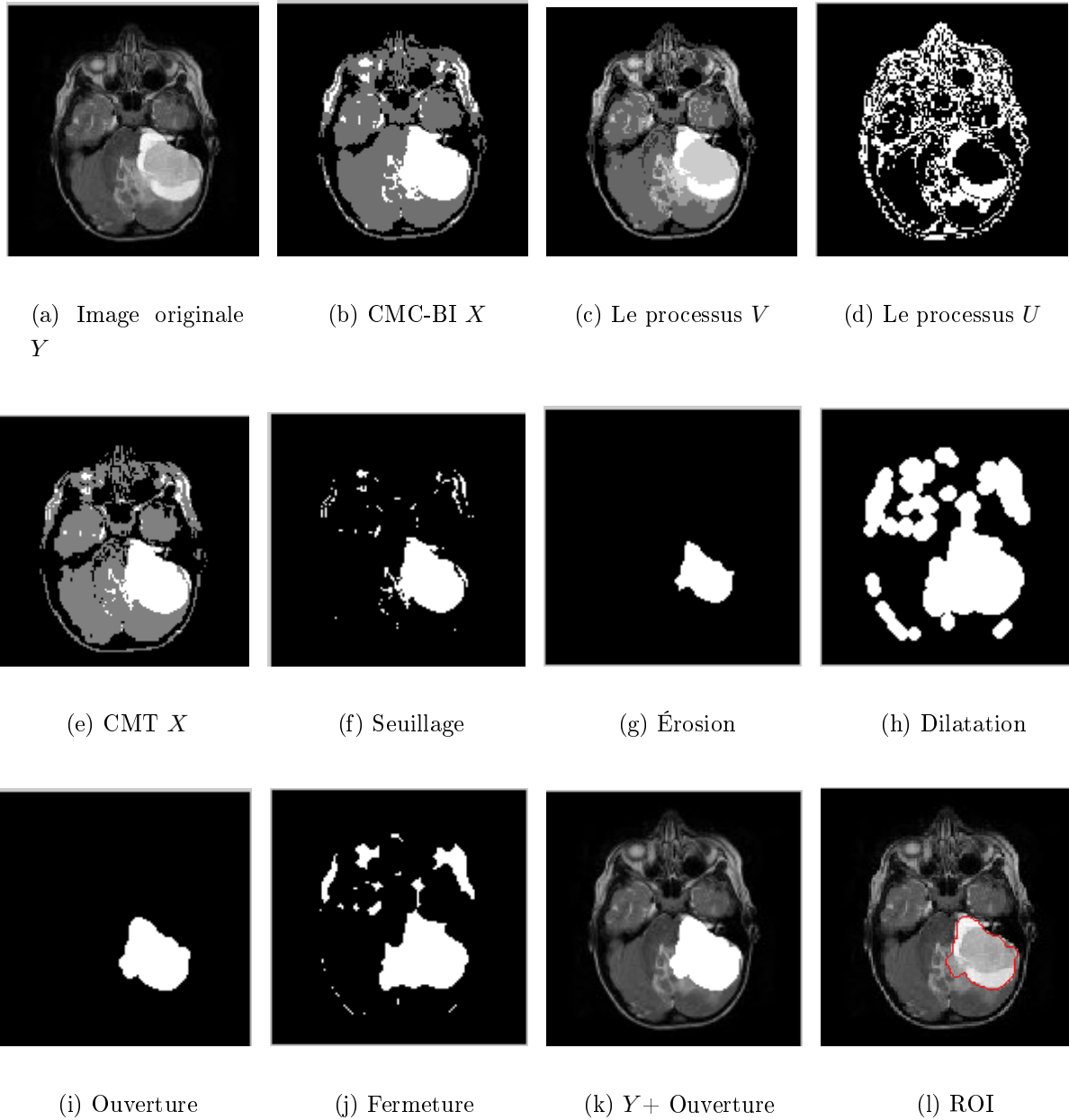


FIGURE 4.11 – Résultat de segmentation 5.

Nous remarquons que, la qualité des images segmentées obtenues par le modèle CMT est meilleure que celle du modèle CMC-BI. Comme première remarque, nos approches sont concurrentes à l'approche initiale avant division en terme de la qualité de segmentation. Une deuxième remarque concernant l'opération de MM la plus performante est l'opération de l'ouverture qui a détecté la région d'intérêt avec une précision de 97%.

## 4.6.2 Résultats Quantitatifs

Pour valider nos résultats visuels, nous calculons quelques critères d'évaluation. La comparaison est en termes de : l'indice de PSNR, l'indice de SSIM et le taux d'erreur. Les résultats obtenus dans chaque expérience par approche sont résumés dans les tableaux suivants : Le tableau 4.1 représente l'indice de PSNR, le tableau 4.2 montre l'indice de similarité SSIM et le tableau 4.3 présente le taux d'erreur.

Expérience	CMC-BI	CMT	Approche 1	Approche 2	Approche 3
Expérience 1	21,9500	31,9076	31,9076	31,9076	31,9076
Expérience 2	21,8058	33,7856	33,7856	33,7856	33,7856
Expérience 3	18,4713	32,2726	32,2726	32,2726	32,2726
Expérience 4	20,8248	30,8248	30,8248	30,8248	30,8248
Expérience 5	22,1529	34,5623	34,5623	34,5623	34,5623

TABLE 4.1 – les valeurs de l'indice PSNR

Expérience	CMC-BI	CMT	Approche 1	Approche 2	Approche 3
Expérience 1	0.5397	0.8395	0.8395	0.8395	0.8395
Expérience 2	0.5150	0.8645	0.8645	0.8645	0.8645
Expérience 3	0.4784	0.7894	0.7894	0.7894	0.7894
Expérience 4	0,4540	0,8259	0,8259	0,8259	0,8259
Expérience 5	0.4977	0,7986	0,7986	0,7986	0,7986

TABLE 4.2 – les valeurs de l'indice SSIM

Expérience	CMC-BI	CMT	Approche 1	Approche 2	Approche 3
Expérience 1	9,2127	2,8028	2,8028	2,8028	2,8028
Expérience 2	10,0450	3,9627	3,9627	3,9627	3,9627
Expérience 3	9.6075	1,7634	1,7634	1,7634	1,7634
Expérience 4	10,7432	2,1536	2,1536	2,1536	2,1536
Expérience 5	11,4270	2,8799	2,8799	2,8799	2,8799

TABLE 4.3 – Les valeurs de taux d'erreur

Les valeurs des indices PNSR, SSIM et taux d'erreur sont identiques pour chaque expérience, ce qui confirme les résultats visuels obtenus. Nous pouvons expliquer ça par : le fait que nous utilisons la même procédure de segmentation, ainsi que les mêmes estimateurs et les mêmes valeurs initiales des paramètres pour toutes les approches. Le modèle CMT est meilleur que le modèle CMC-BI. La CMT améliore la qualité de segmentation, elle considère que les observations sont non-stationnaires, pour modéliser cette non-stationnarité, elle introduit un troisième processus ce qui améliore la qualité de segmentation.

Aussi, nous avons calculé le temps d'exécution de chaque approche proposée, et nous le comparons avec le temps d'exécution de l'approche initiale du modèle CMT avant division. Le graphe de la figure 4.12 représente le temps d'exécution en secondes de chaque approche proposée par expérience.

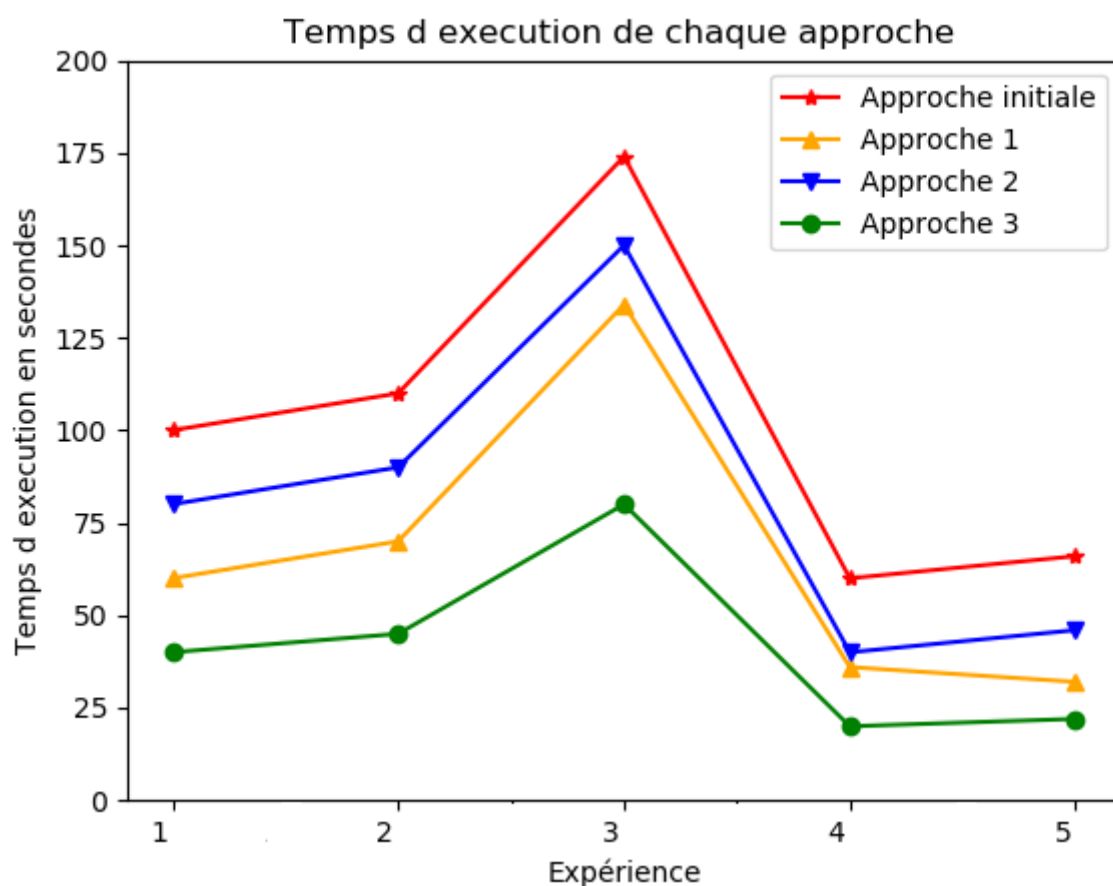


FIGURE 4.12 – Le temps d'exécution en secondes par approche proposée : étude comparative 1

Nous constatons que les approches proposées donnent de meilleurs résultats que l'approche initiale; en fait, le temps d'exécution a considérablement diminué. L'approche 3 du modèle CMTHP est très rapide que les autres approches, elle réduit le temps d'exécution par environ 60% par rapport à l'approche initiale, 30% par rapport à l'approche 1, et environ 45% par rapport à l'approche 2. L'approche 1 a diminué le temps d'estimation par environ 40% que l'initiale approche. et par

15% que l'approche 2, Aussi l'approche 2 a réduit le temps par 30% que l'approche initiale. Nos approches ont réduit le temps d'exécution du modèle CMT par un pourcentage important. Nous remarquons, la même remarque que nous avons remarquée dans les approches du modèle classique présentées dans 3 que la tâche de calcul des probabilités de Baum Welch est la tâche la plus coûteuse dans le processus d'estimation des paramètres. Ainsi que l'estimation des processus  $X$  et  $U$ . Le temps d'exécution d'une CMT dépend de plusieurs facteurs et conditions tels que : la taille des données, la taille des paramètres, le nombre de classes d'appartenance, le nombre de stationnarités, la convergence des estimateurs des paramètres, et le nombre des processus estimés. À partir des résultats obtenus, nous pouvons conclure que le modèle CMT fournit de meilleurs résultats de segmentation que les modèles classiques, mais, il nécessite un temps d'exécution important, surtout lorsque, la taille des données à traiter est très large.

Nous avons également comparé la rapidité des threads et des processeurs. Pour cela, nous calculons le temps d'exécution effectué par les threads et par les multiprocesseurs pour estimer les deux processus cachés en parallèle dans chaque expérience.

La figure suivante montre le temps d'exécution en secondes de chaque méthode de parallélisme pour les approches 2 et 3.

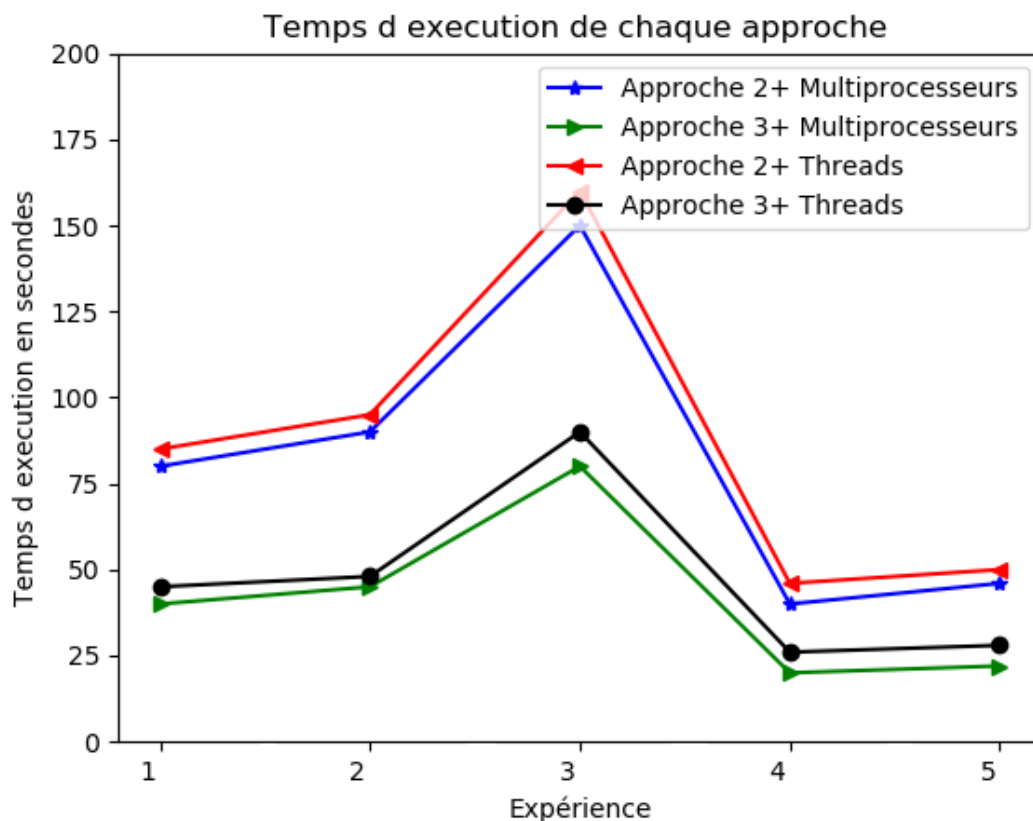


FIGURE 4.13 – Le temps d'exécution en secondes par approche proposée : étude comparative 2

D'après la figure 4.13, les approches qui utilisent les multiprocessus est plus rapide que les approches qui utilisent les threads par environ 15%. Cela confirme, que l'approche 3 est la plus rapide que les autres approches surtout lorsqu'elle utilise les multiprocessus. Les processeurs sont plus rapide que les threads. Chaque processeur a sa propre espace mémoire et unité d'exécution, contrairement aux threads qui sont des sous-processeurs qui partagent entre eux les mêmes ressources pour exécuter certaines taches en parallèle.

Aussi, nous avons étudié la rapidité de convergence des approches proposées. Nous avons calculons le nombre d'itérations nécessaires pour atteindre la convergence pour chaque approche proposée. Nous avons comparé les résultats obtenus avec l'approche initiale, le graphe suivant montre la convergence de chaque approche proposée dans chaque expérience. A partir des résultats obtenus, nous constatons que les approches qui utilisent l'estimation dépendante convergent rapidement que les approche qui utilisent l'approche d'estimation initiale.

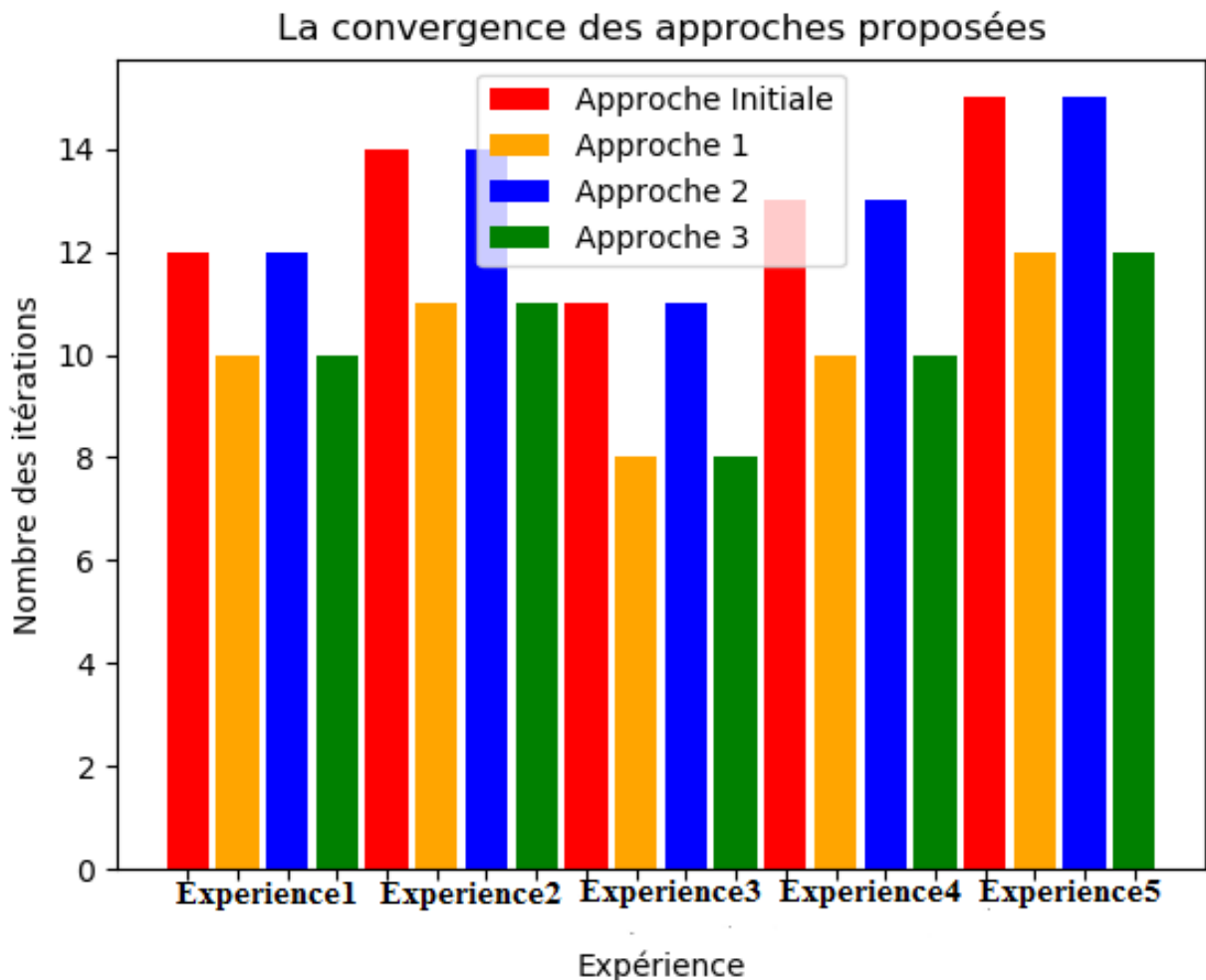


FIGURE 4.14 – La convergence de chaque approche proposée par expérience

Finalement, nous pouvons conclure que toutes les approches proposées réduisent le temps d'exé-

cution et la convergence conservant la même qualité de la segmentation que l'approche initiale. L'approche 3 + multiprocessus donne les meilleurs résultats en terme de temps d'exécution. Aussi, l'opération de l'ouverture du MM donne des résultats satisfaisants à détecter avec une grande précision la forme de la masse tumeur .

## 4.7 Conclusion

Dans ce chapitre, nous avons proposé trois approches pour accélérer le temps d'estimation du modèle chaîne de Markov triplet. La première approche, chaîne de Markov triplet hiérarchique, divise l'image en blocs de même taille. Le traitement de chaque bloc dépend du résultat du bloc précédent. Elle utilise l'approche d'estimation dépendante présentée dans 3.3 pour estimer les paramètres du modèle triplet. La deuxième approche, chaîne de Markov triplet parallèle, se base en phase de décision finale sur les techniques de parallélisme, les multiprocessus ou les threads, pour estimer, en même temps, les processus cachés  $X$  et  $U$  à partir du processus stationnaire  $V$ . La dernière approche, chaîne de Markov triplet hiérarchique parallèle, se concentre sur l'approche d'estimation dépendante présentée dans 3.3 pour estimer les paramètres du modèle triplet, et sur le parallélisme pour estimer les processus cachés  $X$  et  $U$ . Ces approches sont appliquées pour segmenter des images cérébrales IRM. Dans la phase de détection et d'extraction de la région d'intérêt, nous avons appliqué la technique de seuillage. Ensuite, nous avons appliqué sur l'image résultat de seuillage les quatre opérations de MM indépendamment, afin de trouver l'opération qui détecte la forme de la tumeur avec précision. A partir des résultats obtenus, nous remarquons que la troisième approche avec les multiprocessus est plus rapide que les deux autres, et que toutes ces approches ont diminué le temps d'exécution par un pourcentage remarquable par rapport à l'approche initiale. En outre, les approches proposées permettent une segmentation d'images avec la même qualité que l'approche initiale. Les processeurs sont plus rapides que les threads, car chaque processeur a ses propres ressources alors que les threads partagent les ressources du processeur père en synchronisant, par une file d'attente, les tâches effectuées par chaque thread. L'opération ouverture de la MM fournit des meilleurs résultats de détection de la forme de la tumeur par rapport aux autres opérations de la MM. Finalement, nos approches donnent des résultats encourageants en termes de qualité de segmentation et de temps d'exécution.



# Conclusion Générale

Le contexte général de cette étude était la segmentation markovienne des images. Le but est de proposer des approches pour accélérer le temps d'exécution du modèle chaîne de Markov cachée en se basant sur les techniques algorithmiques "diviser pour régner" et le parallélisme. Nous avons apporté dans ce cadre quelques contributions.

Tout d'abord, la première contribution est présentée sous forme des études comparatives : entre les estimateurs itératifs, entre les estimateurs Bayésiens, entre les chaînes de Markov cachées stationnaires et entre les chaînes de Markov cachées stationnaires et non-stationnaires.

Également, deux approches basées sur la technique "diviser pour régner" ont été proposées, la première à estimation indépendante, qui divise l'image à traiter en un ensemble des sous-blocs de même taille, chaque bloc est segmenté indépendamment des autres à l'aide du modèle chaîne de Markov cachée à bruit indépendant. Chaque bloc composant l'image est traité indépendamment d'une manière séquentielle comme une sous-image. La deuxième approche à estimation dépendante, divise l'image en un ensemble de blocs de même taille, le traitement de chaque bloc dépend du résultat de bloc précédent. A partir des résultats obtenus, nous constatons que l'approche d'estimation dépendante est plus rapide que toutes les autres approches proposées. Elle a réduit le temps d'estimation par environ 70% par rapport à l'approche initiale (avant division). Pour les résultats de segmentation visuels et quantitatifs toutes les approches proposées donnent les mêmes résultats que l'approche initiale.

Ensuite, l'approche à estimation dépendante a été réutilisée pour accélérer le temps d'exécution du modèle récent chaîne de Markov triplet, trois méthodes ont été exposées. La première estime d'une manière hiérarchique les paramètres du modèle triplet. La deuxième méthode garde le même processus d'estimation que la chaîne de Markov triplet. Les processus cachés  $X$  et  $U$  de ce modèle sont estimés en parallèle. La dernière méthode est basée sur les deux approches précédentes, elle utilise l'approche à estimation dépendante pour estimer les paramètres du modèle triplet, et, elle utilise le parallélisme pour estimer simultanément les processus cachés  $X$  et  $U$ . Ces trois approches ont été utilisées pour segmenter les images cérébrales IRM. A partir des résultats obtenus, nous pouvons conclure que les multiprocessus sont plus rapides à estimer les deux processus cachés  $X$  et  $U$  en parallèle par rapport aux threads avec environ 15%, et par 50% par rapport à l'approche initiale. La troisième méthode avec les multiprocessus est la plus performante, elle a réduit le temps

d'exécution par environ 60% du temps. Les résultats visuels et quantitatifs sont les mêmes que l'approche initiale de segmentation du modèle triplet.

Finalement, une dernière contribution fut de proposer une méthode pour extraire la région d'intérêt (la tumeur) à partir des images cérébrales IRM segmentées par le modèle triplet. Notre méthode est basée sur l'hybridation de la technique de seuillage avec la morphologie mathématique. Nous avons appliqué la technique de seuillage suivie par quelques opérations de la morphologie mathématique : l'érosion, la dilatation, l'ouverture et la fermeture. La comparaison de ces dernières opérations montre que l'opération d'ouverture est la meilleure, elle détecte la forme de la tumeur avec une précision de 97%.

Chaque modèle de Markov caché traite des problèmes précis, les modèles classiques sont utilisés pour traiter des problèmes basiques comme la segmentation d'images filtrées ; les modèles couples sont conçus pour résoudre les problèmes que les modèles classiques sont incapables de résoudre comme la segmentation d'images en tenant compte du bruit et des contours des objets ; les modèles triplet pour traiter les données non-stationnaires ; et enfin les modèles flous pour segmenter les images floues. Comme perspectives que nous les adresserons aux prochaines compétitions : un modèle de Markov caché robuste et rapide qui permet de segmenter les images en prenant compte, en même temps, de l'information bruit, le flou, la non-stationnarité, les contours, et concurrent aux méthodes de segmentation sémantiques à savoir : les réseaux de neurones convolutifs [25] et les coupes des graphes [22]. Et pour accélérer le temps de segmentation des modèles de Markov cachés : champs, arbres, couples et évidentiels, nous souhaiterons réutiliser nos approches ou d'autres techniques à savoir : le parallélisme à mémoire partagée OpenMP ou à mémoire distribuée MPI...

# Bibliographie

- [1] A. A. Aly, S. Bin Deris, N. Zaki, Research Review for Digital Image Segmentation Techniques, International Journal of Computer Science and Information Technology(IJCSIT) Vol 3, No 5, October 2011.
- [2] M. Bergouniou, Quelques Méthodes de Filtrage en Traitement d'Image, 2011.
- [3] F. Sultana, A. Sufian, P. Dutta, Advancements in Image Classification using Convolutional Neural Network, Fourth International Conference on Research in Computational Intelligence and Communication Networks proceeding, 2018, available at :arXiv :1905.03288v1 [cs.CV],8 May 2019.
- [4] L.A.D.S.A. Molligoda, P.G. Wijayarathna, Applicability of hidden Markov model approach for sinhala speech recognition ? a systematic Review, International Research Symposium on Engineering Advancements(RESA) 2015, 2015.
- [5] O. Augereau, Reconnaissance et classification d'images de documents, thèse de doctorat de l'université de Bordeaux 1, Soutenue le 14 février 2013.
- [6] N. Brunel, W. Pieczynski, Signal restoration using hidden Markov chains with copulas, Signal Processing, pp. 2304-2315, 2005.
- [7] M. Waseem Khan, A Survey : Image Segmentation Techniques, International Journal of Future Computer and Communication, vol. 3, no. 2, April 2014.
- [8] S. Saini, K. Arora, A Study Analysis on the Different Image Segmentation Techniques, International Journal of Information and Computation Technology, pp. 1445-1452, 2014.
- [9] S. S. Al-amri, N. V. Kalyankar, Khamitkar S.D, Image Segmentation by Using Threshold Techniques. 2014.
- [10] E. Roura, al, multispectral adaptive region growing algorithm for brain extraction on axial MRI, Computer methods and programs in biomedicine, pp.655-673, 2014.
- [11] N. Senthilkumaran, al, Edge detection techniques for image segmentation—a survey of soft computing approaches, International journal of recent trends in engineering, 2009.
- [12] K. Hadjar, O.Hitz, R.Ingold, Newspaper Page Decomposition Using a Split and Merge Approach,2001,pp.1186-1189, DOI :10.1109/ICDAR.2001.953972.
- [13] T.F Chan, al, Active contours without edges, Image processing IEEE Transactions, pp.266-277, 2001.

- [14] J. A. Canny, computational approach to edge detection, Pattern Analysis and Machine Intelligence IEEE Transactions , pp.679-698, 1986.
- [15] R. Haralick, Stanley Stenberg, Xinhua Zhuang, Image Analysis Using Mathematical Morphology, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.PAMI-9, no.4, July 1987.
- [16] P. Maragos, A Representation Theory for Morphological Image and Signal Processing, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.II, no.6, June 1989.
- [17] A.M.Raid, al, Image Restoration Based on Morphological Operations, International Journal of Computer Science, Engineering and Information Technology (IJCEIT), Vol. 4, No.3, June 2014.
- [18] C.Benson, al, Brain tumor extraction from MRI brain images using marker based watershed algorithm, pp.318-323, 2015.
- [19] S. tatiraju, al,Image segmentation using K-means clustering EM and normalized cuts, 2008.
- [20] Z.Wu, R. Leahy, An optimal graph theoretic approach to data clustering : Theory and its application to image segmentation,Pattern Analysis and Machine Intelligence, IEEE Transactions, pp.1101-1113, 1993.
- [21] Y.Boykov, G. Funka-Lea, Graph cuts and efficient ND image segmentation International journal of computer vision T.70, No.2, pp.109-131, 2006.
- [22] Y.Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, No9. pp. 1124-1137, 2004.
- [23] M. Sandler et al. Mobilenetv2 : Inverted residuals and linear bottlenecks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,pp. 4510-4520, 2018.
- [24] V.Vladimir Kniaz, Conditional GANs for semantic segmentation of multispectral satellite images,Proc. SPIE 10789, Image and Signal Processing for Remote Sensing XXIV, 107890R 9 October 2018, <https://doi.org/10.1117/12.2325601>.
- [25] V.Vladimir Kniaz, Deep learning for dense labeling of hydrographic regions in very high resolution imagery, Proc. SPIE 11155, Image and Signal Processing for Remote Sensing XXV, 111550W,7 October 2019, <https://doi.org/10.1117/12.2533161>.
- [26] SegNet Project : <http://mi.eng.cam.ac.uk/projects/segnet/>
- [27] V. Badrinarayanan , A. Kendall, R. Cipolla, Segnet : A deep convolutional encoder-coder architecture for image segmentation IEEE transactions on pattern analysis and machine intelligence, T.39, No.12, pp.2481-2495, 2017.
- [28] T. Xiao et al, Unified perceptual parsing for scene understanding Proceedings of the European Conference on Computer Vision (ECCV), pp. 418-434,2018.

- [29] K. Sun, al. High-Resolution Representations for Labeling Pixels and Regions, available at : arXiv preprint arXiv :1904.04514, 2019.
- [30] R. Van Hadel, Hidden Markov models, pp 51-64, july 28 2008.
- [31] N. Giordana , W. Pieczynski, Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation,IEEE Trans. Pattern Anal. Machine Intell., vol. 19, no. 5, pp. 465–475, May 1997.
- [32] S. Faisan, L. Thoraval, J. P. Armspach and F. Heitz, Hidden semi-Markov event sequence models : application to brain functional MRI sequence analysis, in IEEE Int. Conf. Image Processing , vol. 1, pp. 880–883, Rochester, New York, USA, September 22-25 2002.
- [33] El- H.Guerrou, al, Combination of Hidden Markov Random Field and Conjugate Gradient for Brain Image Segmentation,2017, available at : arXiv :1705.04823v2[cs.CV] 16 May 2017.
- [34] F. Girard-Ardhuin, al, Oil slick detection by SAR imagery : potential and limitation, in Oceans, pp. 22-26, San Diego, USA, september 2003.
- [35] P. Masson, W. Pieczynski, SEM algorithm and unsupervised statistical segmentation of satellite images, Transaction on Geosciences and Remote Sensing, vol.31, no.3, May 1993.
- [36] W. Pieczynski, D. Benboudjema, P. Lanchantin, Statistical image segmentation using triplet Markov fields, International Symposium on Remote Sensing, SPIEs, Crete, Greece, pp.22-27, 2002.
- [37] E. Monfrini,al, Image and Signal Restoration using Pairwise Markov Trees, IEEE Workshop on Statistical Signal Processing (SSP 2003), Saint Louis, Missouri, Sep-Oct 2003.
- [38] N. J. Rose, Hilbert-Type Space-Filling Curves,2000.
- [39] G. R. C. Marquez, H. J. Escalante, L. E. Sucar, Simplified Quadtree Image Segmentation for Image Annotation, AIAR2010 : Proceedings of the first Automatic Image Annotation and Retrieval Workshop, pp. 24-34, vol.1, issue.1, 2010.
- [40] W.Pieczynski, Pairwise Markov chains, IEEE Transactions Pattern Analysis Machine Intelligence,vol.25, no.5, May 2005.
- [41] S. Derrode, W. Pieczynski, Unsupervised data classification using pairwise Markov chains with automatic copulas selection, Computational Statistics and Data Analysis 63 pp.81-98, 2013.
- [42] P. Lanchantin , W. Pieczynski ,Unsupervised Restoration of Hidden non stationary Markov Chains Using Evidential Priors,IEEE Transactions on Signal Processing, vol. 53, no. 8, august 2005.
- [43] C. Carincotte, Unsupervised image segmentation based on a new fuzzy HMC model, ICASSP'04, Montreal, Canada, May 2004.
- [44] H. Caillol, A. Hillion, W. Pieczynski, Fuzzy random Markov field and insupervised image segmentation, IEEE transactions,Geosci, Remote sensing, 801-810,1993.

- [45] H. Hanzouli-Ben Salah, J. Lapuyade-Lahorgue, D. Benoit, J. Bert, P. Lambin, A. Van Baardwijk, E. Monfrini, W. Pieczynski, D. Visvikis, and M. Hatt, A framework based on hidden Markov trees for multimodal PET/CT image co-segmentation, *Medical Physics*, October 2017.
- [46] H. Li, S. Derrode, and W. Pieczynski, Adaptive on-line lower limb locomotion activity recognition using semi-Markov model and single wearable inertial sensor, *Sensors*, 2019.
- [47] H. Li, S. Derrode, and W. Pieczynski, An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model, *Neurocomputing*, pp. 94-105, 2019.
- [48] M.Ameur, N.Idrissi, C.Daoui, Markovian Segmentation of Color and Gray Level Images, 2016, pp.259-264, DOI :10.1109/CGiV.2016.57.
- [49] M.Ameur, N.Idrissi, C.Daoui, Triplet Markov Chain in Images Segmentation, 2018, pp. 1-8, DOI :10.1109/ISACV.2018.8354055.
- [50] M. Ameur, C. Daoui, N.Idrissi, Hierarchical Hidden Markov Models in Image Segmentation, *Scientific Visualization Journal*, Vol. 12, No. 1, March 2020, pp.22-47, DOI : 10.26583/sv.12.1.03, <http://sv-journal.org/2020-1/03/>.
- [51] Y. Deng, al, Color image segmentation, 2000.
- [52] M Alrashed, A.Moussa, R.Touani, Segmentation Markovienne non supervisée d'images couleurs, Juillet 2011.
- [53] théorème de Bayes : [https://www.ucd.ie/t4cms/Bayes Theorem.pdf](https://www.ucd.ie/t4cms/Bayes%20Theorem.pdf)
- [54] W. Pieczynski, J.-M. Cahen, Champs de Markov cachés flous et segmentation d'images, *Revue de statistique appliquée*, tome 42, no 3, p. 13-31, 1994.
- [55] S. Geman, D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp.721-741, 1984.
- [56] M.Ben Mabrouk, Modèles de Markov triplets en restauration des signaux, Thèse de doctorat Doctorat conjoint Télécom et Management SudParis et Université Pierre et Marie Curie, 2011.
- [57] A. Bonin, B. Chalmond, and B. Lavayssière. Monte-carlo simulation of industrial radiography images and experimental designs. *NDT et E. International*, pp.503-510, 2002.
- [58] B. Chalmond, *Eléments de modélisation pour l'analyse d'images*, Springer, 2000.
- [59] G. Winkler, *Image Analysis, random fields and Markov Chain Monte Carlo Methods : a mathematical introduction*, Springer, 2003.
- [60] F.Flitti, Technique de réduction de données et analyse d'images multispectrales astronomiques par arbres de Markov, thèse de doctorat de l'université Louis Pasteur, Strasbourg 1, 2005.
- [61] J. Besag, Spatial interaction and the statistical analysis of lattice systems. *J. of the Royal Statistical Society Series B*, pp.192-236, 1974.
- [62] C.P. Robert, The Metropolis-Hastings algorithm, arXiv :1504.01896v3 [stat.CO] 27 Jan 2016
- [63] <http://mribatet.perso.math.cnrs.fr/docs/DeveloppementLogiciel/Challenges/GradientStochastique/>

- [64] S.Rafi, Chaînes de Markov cachées et séparation non supervisée de sources, thèse de doctorat conjoint Télécom et Management SudParis -Université Pierre et Marie Curie,2012.
- [65] J.Dai, Hybrid Approach to Speech Recognition Using Hidden Markov Models and Markov Chains, IEE Proc. Vision, Image, and Signal Processing, Vol. 141, no. 5, pp. 273-279, 1994.
- [66] M. Hatt, F. Lamare, N. BouSSION, A. Turzo, C. Collet, F. Salzenstein, C. Roux, P. Jarritt, K. Carson, C. Cheze-Le Rest and D. Visvikis, Fuzzy hidden Markov chains segmentation for volume determination and quantitation in PET, Physics in Medicine and Biology, Vol. 52, No. 12, pp. 3467-3491, Jun. 2007.
- [67] M. Y. Boudaren, A. Belaïd, A New Scheme for Land Cover Classification in Aerial Images, Electronic Engineering and Computing Technology, Lecture Notes in Electrical Engineering, Chapter 40, Vol.60, pp. 471-482, Springer 2010.
- [68] K. Liang, X. Wang, D. Anastassiou, Bayesian Basecalling for DNA Sequence Analysis Using Hidden Markov Models, IEEE/ACM Trans. Computational Biology and Bioinformatics, Vol. 4, No. 3, pp. 430 – 440, Jul/Sep 2007.
- [69] F.Le Ber, M. Benoît, C. Schott, J.-F. Mari, C. Mignolet, Studying crop sequences with Carrot Age, a HMM based data mining software, Ecological Modelling, Vol.191, No.1, 2006.
- [70] R. Bhar, S. Hamori, Hidden Markov models : applications to financial economics, Springer, 2004.
- [71] B.Benmiloud, W.Pieczynski, Estimation des paramètres dans les chaînes de Markov cachées et segmentation d'images. Traitement du Signal 12, pp.433–454,1995
- [72] A. Hillion, W. Pieczynski, Taritement statistique des images, Cours Master 2, Université Pierre et Marie Curie, 2011.
- [73] L. E. Baum and T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, Ann. Math. Statist., vol. 37, no. 6, pp. 1554–1563, 1966.
- [74] C.Bouman, M. Shapiro, Multispectral image segmentation using a multiscale model, IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 3. IEEE, 1992.
- [75] P.Lanchantin, F.Salzenstein, Segmentation d'images multispectrales par arbre de Markov caché flou, 20 th Colloque sur le traitement du signal et des images, pp. 874-877,2005.
- [76] W.Pieczynski, Modèles de Markov en traitements d'images : Markov models in image processing, Traitement du Signal, Vol. 20, No. 3, pp. 255-278, 2003.
- [77] S. Derrode, W. Pieczynski, Segmentation non supervisée d'images par une chaine de Markov couple, 2004.
- [78] N.Brunel, Sur Quelques Extensions des chaines de Markov Cachées et Couples : Application à la Segmentation Non-supervisée de Signaux Radar, thèse de doctorat de l'université Paris 6, 2005.

- [79] M.E.Y Boudaren, Modèles Graphiques Evidentiels, thèse de doctorat conjoint Télécom et Management SudParis -Université Pierre et Marie Curie En cotutelle avec l'Université des Sciences et de la Technologie Houari Boumediene, 2014.
- [80] W.Pieczynski, Triplet Markov Chains, Comptes Rendus de l'académie des Sciences-Mathématique, série I,pp.275-278, 2002.
- [81] W. Pieczynski, Triplet Markov Chains and Evidence, Submitted to International Journal of Approximate Reasoning, September 2003.
- [82] M.E.Y.Boudaren, W. Pieczynski, Dempster–Shafer fusion of evidential pairwise Markov chains, IEEE Trans. Fuzzy Syst,2016, <http://dx.doi.org/10.1109/TFUZZ.2016.2543750>.
- [83] A.Dempster, al, Maximum likelihood from imcomplete data via the EM Algorithm ,Journal the Royal Statistic Society, 1977.
- [84] P. Lanchantin, J. Lapuyade-Lahorgue, W. Pieczynski, Unsupervised segmentation of triplet Markov chains hidden with long-memory noise, Signal Processing 88 pp.1134–1151, November 2007.
- [85] L. A. Zadeh, Fuzzy sets, Information and Control, vol. 8, pp. 338–353, 1965.
- [86] C.Carincotte, Segmentation Markovienne Floue d'Image : Application en Détection de Changements entre Images Radar, Thèse de doctorat de l'université Paul Cezanne, 2005.
- [87] M. Ameer, C. Daoui, N. Idrissi, Some Stationaries Hidden Markov Models , International Journal of Advanced Science and Technology, Vol. 29, No. 1, pp. 1043 –1057, January 2020.
- [88] M. Ameer, C. Daoui, N. Idrissi, Markovian Segmentation of Brain Tumor MRI Images ,pp.155-165, 2017.
- [89] N. Rechid, al, Segmentation non supervisée d'images basée sur les modèles de Markov cachés, Courrier de savoir, No.12, octobre 2011, pp 39-34.
- [90] A. Dempster, al, Maximum likelihood from imcomplete data via the EM Algorithm , Journal the Royal Statistic Society, serie B(Methodological), 1977.
- [91] G. Celeux, J.A .Diebolt, Stochastic approximation type EM algorithm for the mixture, Rapport de recherche INRIA, 1383, 1991.
- [92] Wei, Tanner, A Monte Carlo implementation Of the EM algorithm and the poor man's data augmentation algorithm, Journal of the American Statistical Association, 85,pp.699-704, 1987.
- [93] R. V. Handel, Hidden Markov models, July 2008.
- [94] J. C. Biscarat, G. Celeux and J. Diebolt, Stochastic versions of the EM algorithm, 1985.
- [95] G. J. McLachland and T. Krishnan, EM algorithm and exetensions, 1977.
- [96] P. Lanchatin, Chaines de Markov triplet et segmentation non supervisée de signaux, thèse de doctorat de l'Institut National des Telecommunucations,2006.



- [97] W. Pieczynski , Convergence of the iterative conditional estimation and application on the mixture proportion identification IEEE, Statistical Signal Workshop, SSP Madison,WI,USA, 26-29 August 2007.
- [98] P. Devijver, Baum's forward backward algorithm revisited, Pattern Recognition Letter.3,pp 369-373,1985.
- [99] W. Pieczynski , Sur la convergence de l'estimation Conditionnelle itérative, C .R.Acad.Sci, Paris, ser.1346,457-460, 2008.
- [100] G. D. Fornay , the Viterbi algorithm, Proceeding of IEEE, vol.61, No.3, pp 268-277,1970.
- [101] M. L. Corner and E.J.delp,The EM/ MPM Algorithm for segmentation of textured images, October, pp.1731-1744, 2000.
- [102] Y. Al-Najjar, D. Chen Soong, Comparison of Image Quality Assessment : PSNR, HVS, SSIM, UIQI. International Journal of Scientific and Engineering Research, Vol.3, Issue 8, August-2012.
- [103] La base d'images Wang : <http://wang.ist.psu.edu/docs/related/>
- [104] M. Ameer, N. Idrissi, C.Daoui, Color Images Segmentation using Pairwise Markov Chain, pp.1-6, 2018,DOI :10.1109/ICMCS.2018.8525988.
- [105] La base d'images VisTex : <http://vismod.media.mit.edu/pub/vistex/vistex.tar.gz.m>.
- [106] P.Lanchantin, W.Pieczynski, Unsupervised Non Stationary Image Segmentation Using Triplet Markov Chains, Advanced Concepts for Intelligent Vision Systems (ACVIS 04), Brussels, Belgium, 2004.
- [107] La base d'images Brodatz : [https://multibandtexture.recherche.usherbrooke.ca/original\\_brodatz.html](https://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html).
- [108] W. Pieczynski, EM and ICE in hidden and triplet Markov models, Stochastic Modeling Techniques and analysis ,International Conference, 2010.
- [109] L.A. Zadeh, Fuzzy Sets As a Basis For a Theory Of Possibility, Fuzzy Sets and Systems 1 Journal,pp. 3-28, 1978.
- [110] S. Lou, X. Jiang, J. Paul Scott, An efficient divide-and-conquer algorithm for morphological filters, 12th CIRP Conference on Computer Aided Tolerancing, Science Direct Procedia CIRP 10, pp. 142-147, 2013.
- [111] S. Paira, S. Chandra, S. Safikul Alam, and P. Sarthi Dey, Review Report on Divide and Conquer Sorting Algorithm, IEEE Kolkata Section.2014
- [112] M. Ameer, C. Daoui, N. Idrissi, Fast Markovian Images Segmentation, pp.1-6, 2020, DOI :10.1109/ICOA49421.2020.9094479.
- [113] La base d'images PASCAL VOC2010 dataset :<http://host.robots.ox.ac.uk/pascal/VOC/>
- [114] La base d'images ADE 20k dataset : <https://groups.csail.mit.edu/vision/datasets/ADE20K/>

- [115] La base d'images SPRS Potsdam dataset : [http://www2.isprs.org/commissions/comm3/wg4/semantic\\_labeling.html](http://www2.isprs.org/commissions/comm3/wg4/semantic_labeling.html)
- [116] [http://www.sfrnet.org/Data/upload/files/Le\\_20rayonnement20X.pdf](http://www.sfrnet.org/Data/upload/files/Le_20rayonnement20X.pdf)
- [117] <https://perso.telecom-paristech.fr/bloch/P6Image/RX-court.pdf>
- [118] <https://www.lachimie.fr/analytique/rmn/>
- [119] <https://www.laradioactivite.com/site/pages/phenomene.html>.
- [120] M.M.Hamayun, Simulation Native des Systèmes Multiprocesseurs sur Puce à l'aide de la Virtualisation Assistée par le Matériel, thèse de doctorat de l'université Grenoble,2013.
- [121] S.Thibault,Ordonnancement de processus légers sur architectures multiprocesseurs hiérarchiques : BubbleSched, une approche exploitant la structure du parallélisme des applications, thèse de doctorat de l'université Bourdeaux 1,2007.
- [122] M.A. Heroux, P.Raghavan, S.Horst, Parallel Processing for Scientific Computing, journal of Mathematics of Computation - Math. Comput. ER - 2006, DOI : 10.1137/1.9780898718133.
- [123] <https://www.openmp.org/>.
- [124] <https://computing.llnl.gov/tutorials/mpi/>.
- [125] <https://courspython.com>.
- [126] [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_tutorials.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html).
- [127] <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/2235545-faites-de-la-programmation-parallele-avec-threading>
- [128] S.Beucher, F.Meyer, The Morphological Approach to Segmentation : The Watershed Transformation, Mathematical Morphology in Image Processing journal, pp.433–481,1993, DOI :10.1201/9781482277234-12.
- [129] N.Hautière, D.Aubert, M.Jourlin, Measurement of local contrast in images, application to the measurement of visibility distance through use of an onboard camera, Traitement du Signal.2, 2006.