

2m11.2785.3

Université de Montréal

Graphes de flot et coupes en restauration bayésienne d'images

par

Martin Bonneville

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

Janvier 2000

© Martin Bonneville, 2000



QA

76

U54

2000

v.032



Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé

Graphes de flot et coupes en restauration bayésienne d'images

présenté par

Martin Bonneville

a été évalué par un jury composé des personnes suivantes:

Neil Stewart, président du jury
Jean Meunier, directeur de recherche
Sébastien Roy, codirecteur de recherche
Yoshua Bengio, membre du jury

Mémoire accepté le 21 avril 2000

Sommaire

Ce mémoire se consacre à l'étude et au développement de techniques d'optimisation, qui sont reliées au problème de maximum *a posteriori* en restauration bayésienne d'images. Suite à une revue des techniques existantes, nous concentrons nos efforts sur des techniques d'optimisation basées sur le calcul de la coupe de coût minimum dans des graphes de flot particuliers.

Dans un premier temps, nous démontrons qu'il est possible, pour une classe de modèle *a priori* restreinte à un potentiel de clique linéaire, d'obtenir de façon non-itérative et efficace l'optimum global du problème de maximum *a posteriori*.

En second lieu, nous proposons un algorithme itératif de recherche locale qui permet d'optimiser la probabilité *a posteriori* pour une classe moins restreinte de potentiel de clique.

Des résultats théoriques sont présentés sous forme de démonstrations mathématiques qui expliquent le fonctionnement et les limitations des algorithmes présentés et développés. Puis, c'est par l'expérimentation que nous validons la théorie en effectuant une étude comparative de nos algorithmes avec des algorithmes standards.

Nous avons appliqué les techniques analysées et développées aux problèmes de restauration d'images bruitées et de correspondance stéréoscopique. Dans chacune des applications, le maximum de la probabilité *a posteriori* trouvé par les algorithmes basés sur le calcul de la coupe de coût minimum était substantiellement plus élevé qu'avec les techniques standards.

Table des matières

Sommaire	iv
Table des figures	vii
Remerciements	xvi
1 Introduction	1
2 Graphes de flot	3
2.1 Graphe de flot	3
2.1.1 Flot	4
2.1.2 Coupe	5
2.1.3 Flot maximum et coupe minimum	6
2.2 Algorithmes de calcul du flot maximum	6
2.2.1 Notions supplémentaires	6
2.2.2 L’algorithme de Ford et Fulkerson	8
2.2.3 L’algorithme de Golberg et Tarjan	9
3 Modèle bayésien en restauration d’images	12
3.1 Restauration d’images	12
3.2 Le problème d’étiquetage	13
3.2.1 Champ aléatoire de Markov	15
3.2.2 Champs aléatoire de Gibbs	17

<i>TABLE DES MATIÈRES</i>	vi
3.2.3 Probabilité <i>a posteriori</i> et énergie associée	19
3.3 Modélisation des connaissances <i>a priori</i>	20
4 Optimisation et maximum <i>a posteriori</i>	25
4.1 Méthodes de calcul du MAP	25
4.1.1 Méthodes locales	25
4.1.2 Méthodes globales	27
5 Minimisation d'énergie et graphe de flot	34
5.1 Une généralisation de Roy et Cox [RC98]	35
5.1.1 Le graphe	35
5.1.2 Un contre-exemple	38
5.1.3 Un graphe modifié	38
5.1.4 La coupe minimum minimise l'énergie	41
6 Approximation par graphes de flot	48
6.1 Une approche approximative	48
6.2 Espace de réétiquetage	49
6.2.1 Réétiquetage $\alpha\beta$	50
6.2.2 Réétiquetage α	54
7 Résultats	64
7.1 Expérience sur une surface constante par morceaux	65
7.1.1 Utilisation d'un potentiel de clique linéaire	66
7.1.2 Utilisation d'un potentiel de clique quadratique tronqué . .	75
7.2 Expérience sur la segmentation de régions non-ordonnées	76
7.3 Expérience en vision stéréoscopique	77
7.3.1 La séquence du parcomètre	79
7.3.2 La séquence de l'arbre	84

<i>TABLE DES MATIÈRES</i>	vii
8 Discussion et conclusion	89
8.1 Discussion	89
8.1.1 Améliorations	90
8.2 Conclusion	91
Bibliographie	92

Table des figures

2.1	Illustration d'un graphe de flot avec deux noeuds terminaux : la source s et le drain t	4
2.2	Graphe de flot et coupe. La représentation de la coupe comme partition de l'ensemble des sommets est caractérisée par l'ensemble des sommets dans \mathcal{T} et dans $\overline{\mathcal{T}}$. La représentation équivalente de la coupe peut aussi se voir par les arêtes en pointillées qui vont d'un sommet de \mathcal{T} vers un sommet de $\overline{\mathcal{T}}$	5
2.3	Illustration du fonctionnement de l'algorithme de Golberg et Targan. En (a) nous avons une section d'un graphe de flot. En (b), nous avons le résultat d'une poussée de 100 unités de flot de la source s vers le premier noeud. L'excès contenu dans ce noeud est $e = 100$. (c) Effet d'une poussée de 30 unités de flot (à partir du premier noeud).	10
3.1	Illustration de la correspondance qu'engendre une configuration $f = \{f_0, \dots, f_{m-1}\}$ entre l'ensemble des sites \mathcal{S} et l'ensemble des étiquettes $\mathcal{L} = \{0, \dots, M - 1\}$	13
3.2	Systèmes de voisinage souvent utilisés en vision et traitement d'images. (a) système de premier ordre : 4-voisins. (b) système de second ordre : 8-voisins.	16

3.3	Les cliques associées aux systèmes de voisinages de premier et second ordre. Le système de second ordre regroupe l'ensemble de toutes les cliques présentes. Le système de premier ordre engendre les cliques ayant les formes suivantes : 1, 2 et 3. En pratique, les cliques constituées de paires de sites (formes 2, 3, 4 et 5) sont les plus utilisées.	17
3.4	Graphe du modèle de Potts.	21
3.5	Graphe du potentiel quadratique.	22
3.6	Graphe du potentiel linéaire.	23
3.7	Graphe du potentiel quadratique tronqué.	23
4.1	Graphe de la formulation de Greig <i>et al.</i> [GPS89].	30
4.2	Graphe de la formulation multi-terminal de Boykov <i>et al.</i> [BVZ98]. Les noeuds blancs correspondent aux sites (pixels) de \mathcal{S} , tandis que les noeuds noirs sont les terminaux (étiquettes en segmentation).	32
5.1	Représentation du graphe de flot correspondant à une image 1D avec 3 pixels $\{i'', i, i'\}$ (de la gauche vers la droite), lorsque le système de voisinage \mathcal{N} est de premier ordre (graphe de Roy et Cox [RC98]). La coupe minimum \mathcal{C} , formée des arêtes en pointillées, sépare le graphe en deux ensembles de noeuds, ceux connectés à la source s (noeuds blancs, $\in \mathcal{T}$) et les autres connectés au drain t (noeuds noirs, $\in \overline{\mathcal{T}}$). Les arêtes sortant de la source ont une capacité infinie.	36

5.2 Voici un graphe de flot correspondant à la formulation de Roy et Cox [RC98], qui démontre que cette formulation ne garantit pas toujours une solution qui correspond à un étiquetage valide selon l'équation (5.1). Les capacités montrées sont celles du graphe original (pas celles du graphe résiduel). La capacité des arêtes avants et inverses sont identiques (comme dans un graphe non-dirigé). La coupe correspond aux arêtes en pointillés. La coupe \mathcal{C} est de coût minimum, car toutes les autres coupes contiennent au moins une arête de coût 100. 39

5.3 Les capacités incidentes au noeud $(i, k + 1)$ sont : A, B, C et D. Pour que l'arête $((i, k), (i, k + 1))$ ne soit jamais saturée, il faut que sa capacité respecte : $K_{ik} > A + B + C$. La capacité D n'entre pas dans cette somme à cause de la propriété de conservation du flot. 41

5.4 Illustration d'une coupe (arêtes pointillées) possédant plus d'une arêtes de vraisemblance pour le site i . La région rehaussée en gris contient une arête inverse. Cette arête est coupée pour qu'il n'y ait pas de connection allant de l'ensemble \mathcal{T} (noeuds blancs) vers l'ensemble $\overline{\mathcal{T}}$ (noeuds noirs). 43

5.5 Illustration de l'apparence d'une coupe entre les sites i et i' . Les sommets blancs sont dans l'ensemble \mathcal{T} , tandis que les sommets noirs sont dans $\overline{\mathcal{T}}$. (a) Lorsque $f_i = 0 < f_{i'} = 3$, (b) lorsque $f_i = f_{i'} = 1$ 45

6.1 Pseudo-code de l'algorithme de réétiquetage $\alpha\beta$. Nous illustrons le fonctionnement de l'algorithme à la figure 6.2. 50

6.2 Illustration du fonctionnement de l'algorithme de réétiquetage $\alpha\beta$, lorsque α est le blanc et β est le gris. (a) la configuration avant le réétiquetage $\alpha\beta$ et (b) la configuration après le réétiquetage $\alpha\beta$. . 51

6.3	Aspect du graphe de l'algorithme de réétiquetage $\alpha\beta$. Les noeuds gris sont dans \mathcal{S}_α et les noeuds noirs sont dans \mathcal{S}_β	52
6.4	Pseudo-code de l'algorithme de réétiquetage α . Il y a plusieurs façons de générer aléatoirement une permutation. Nous utilisons la distribution (histogramme) des étiquettes. Nous illustrons le fonctionnement de l'algorithme à la figure 6.5.	55
6.5	Illustration du fonctionnement de l'algorithme de réétiquetage α (lorsque α est le blanc). (a) la configuration avant le réétiquetage α et (b) la configuration après le réétiquetage α	55
6.6	Aspect du graphe utilisé dans l'algorithme de réétiquetage α	56
6.7	Les arêtes connectées au drain t correspondent aux arêtes de vraisemblance associant l'étiquette α , tandis que les arêtes connectées à la source s sont associées à l'étiquette de l'itération précédente (par exemple au étiquettes a et b). Les coupes entraînant des discontinuités dans l'étiquetage f^c sont de la forme : (a),(b),(c),(d), (e) et (f), où les sites i et i' sont voisins, c'est-à-dire que $\{i, i'\} \in \mathcal{P}_N$. Pour (e) et (f), les coûts associés aux discontinuités sont incorporés dans la capacité des arêtes de vraisemblance (s, i) et (s, i') , qui associent les étiquettes a et b aux sites i et i' respectivement.	61
7.1	(a) Image originale. (b) Image bruitée avec un bruit gaussien de moyenne 0 et d'écart-type 25. (c) Restauration sans contexte (maximum de vraisemblance).	65
7.2	Résultat obtenu par le recuit simulé de Geman et Geman [GG84] après 10000 itérations de l'algorithme (1289 minutes). Une décroissance logarithmique de la température a été utilisée, de même qu'un potentiel de clique linéaire avec $\beta = 0.5$. L'énergie initiale est de 1541118 et l'énergie finale est de 157427.	66

7.3 Résultats du MAPE avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 114654, (b) 154137, (c) 216679, (d) 309969, (e) 336809 et (f) 102371. 67

7.4 Résultats de l'IMC avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 139935, (b) 214779, (c) 359972, (d) 653899, (e) 1597630 et (f) 29342090. 69

7.5 Résultats du réétiquetage α avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 111975, (b) 158153, (c) 218658, (d) 322866, (e) 280621 et (f) 102371. 70

7.6 Résultats du réétiquetage $\alpha\beta$ avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 111574, (b) 152636, (c) 217687, (d) 284025, (e) 276987 et (f) 103125. 72

7.7 Graphique de la décroissance de l'énergie dans le temps (en minutes) pour chacun des algorithmes, lorsque le potentiel de clique est linéaire. (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100.0$ 73

7.8	Graphique de la décroissance de l'énergie par itération des algorithmes lorsque le potentiel de clique est linéaire. (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100.0$	74
7.9	Résultats d'une restauration utilisant un potentiel de clique quadratique tronqué avec $\beta = 0.7$ et $\alpha = 1200$. (a) résultat obtenu avec l'IMC. (b) résultat obtenu par l'algorithme de réétiquetage α . (c) résultat obtenu par l'algorithme de réétiquetage $\alpha\beta$. L'énergie initiale est de 1334314. L'énergie finale est de : (a) 278586, (b) 170002 et (c) 137970.	75
7.10	(a) image originale composée de 4 niveaux de gris : 0, 10, 20 et 30, (b) image bruitée avec un bruit gaussien de moyenne 0 et d'écart-type 10.	76
7.11	Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme IMC sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403, (b) 76838 et (c) 104882 . L'énergie finale est : (a) 53313, (b) 56497 et (c) 59041	77
7.12	Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme de réétiquetage α sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403, (b) 76838 et (c) 104882 . L'énergie finale est : (a) 53194, (b) 55177 et (c) 56827	78
7.13	Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme de réétiquetage $\alpha\beta$ sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403 (b) 76838 (c) 104882. L'énergie finale est : (a)53193, (b) 55180 et (c) 56603.	78
7.14	Image de la scène du parcomètre	79

7.15 Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique linéaire, 24 disparités et $\beta = 2.0$. (a) MAPE, (b) IMC, (c) réétiquetage α et (d) réétiquetage $\alpha\beta$. L'énergie initiale est de 1012729. L'énergie finale est de : (a) 154702, (b) 228067, (c) 205988 et (d) 220853. 81

7.16 Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique de Potts, 24 disparités et $\beta = 10.0$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 850966. L'énergie finale est de : (a) 302393, (b) 244793 et (c) 248247. 82

7.17 Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique quadratique tronqué, 24 disparités, $\beta = 1.0$ et $\alpha = 50$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 2329670. L'énergie finale est de : (a) 220891, (b) 230540 et (c) 242365. 83

7.18 Image de la scène de l'arbre 84

7.19 Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique linéaire, 24 disparités et $\beta = 20.0$. (a) MAPE, (b) IMC, (c) réétiquetage α et (d) réétiquetage $\alpha\beta$. L'énergie initiale est de 9562952. L'énergie finale est de : (a) 1605728, (b) 2553917, (c) 2519824 et (d) 3057936. 86

7.20 Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique de Potts, 24 disparités et $\beta = 100$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 8481311. L'énergie finale est de : (a) 3760584, (b) 3166478 et (c) 3591949. 87

7.21 Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique quadratique tronqué, 24 disparités, $\beta = 20.0$ et $\alpha = 700$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 32100146. L'énergie finale est de : (a) 3301562, (b) 12208421 et (c) 9178847.

Remerciements

Je voudrais remercier mon directeur de recherche M. Jean Meunier pour son soutien, sa grande disponibilité et ainsi que pour ses commentaires qui m'ont aidé à guider et à réaliser ce travail.

J'aimerais aussi remercier mon codirecteur Sébastien Roy qui, par des discussions et travaux communs, a su cultiver mon intérêt pour la recherche. Sans lui, ce mémoire porterait probablement sur un tout autre sujet.

Je remercie M. Neil Stewart et M. Yoshua Bengio pour leurs commentaires pertinents sur le contenu de ce mémoire.

Enfin, je voudrais remercier tous les gens qui me sont chers et qui m'ont encouragé à finir ce mémoire, soit : Alexandre Schoeb, Israël Noël, Jacques Bonneville, Mireille Langis et Clotilde Gaudreau. Finalement, je remercie Marie-Mai Brideau (décédée au début de ma maîtrise), qui était comme une mère.

Chapitre 1

Introduction

En vision artificielle et en traitement d'images, nous sommes confrontés à des problèmes de restauration d'images lorsque que nous désirons extraire de l'information comme : l'intensité, la texture, l'élévation, la disparité, etc., à partir d'observations discrétisées et dégradées de la réalité. Une approche bayésienne permet une formulation probabiliste du problème de restauration sous forme d'étiquetage¹, en définissant un modèle de la dégradation (fonction de vraisemblance) et un modèle *a priori* sur le contexte (structure locale, cohérence spatiale, ...). Ainsi, dans la théorie de Bayes, le problème de restauration consiste à rechercher l'étiquetage le plus probable, c'est-à-dire celui qui maximise la probabilité *a posteriori* (MAP).

Notre objectif est de développer et d'analyser des techniques de calcul efficaces du MAP, qui sont basées sur la coupe de coût minimum dans des graphes de flot particuliers.

Le second chapitre initie le lecteur aux notions de graphe de flot. Plus particulièrement, nous expliquons l'équivalence entre le flot maximum et la coupe de coût minimum. Nous y décrivons les algorithmes de calcul du flot maximum qui sont utilisés, en pratique, pour calculer la coupe de coût minimum.

¹Une étiquette peut représenter une texture, une intensité, une disparité, etc.

Dans le troisième chapitre, nous présentons le problème de restauration d'images comme un problème d'étiquetage qui maximise la probabilité *a posteriori*. Nous expliquons le modèle probabiliste qui se compose d'un modèle de la dégradation (fonction de vraisemblance) et d'un modèle *a priori* sur l'étiquetage qui est modélisé par un champ de Markov avec interactions locales.

Le quatrième chapitre fait la revue des techniques d'optimisation les plus utilisées pour trouver le maximum de la probabilité *a posteriori*.

Le cinquième chapitre explique comment transposer un problème de recherche globale du maximum de la probabilité *a posteriori* en un problème de coupe de coût minimum dans un graphe de flot particulier. Premièrement, nous décrivons un modèle existant et nous expliquons pourquoi ce modèle ne permet pas de modéliser correctement le problème de maximum *a posteriori* lorsque le potentiel de clique utilisé par la distribution *a priori* a une forme linéaire générale. Ensuite, nous expliquons quelques solutions possibles à ce problème.

Le sixième chapitre présente deux techniques de recherche locale basées sur le calcul de la coupe de coût minimum dans un graphe de flot particulier. Ces méthodes permettent l'utilisation de modèles *a priori* plus généraux. Dans un premier temps, nous présentons l'algorithme de réétiquetage $\alpha\beta$ développé par Boykov *et al.* [BVZ99]. Puis, nous présentons la méthode de réétiquetage α que nous avons développée en parallèle à la leur.

Dans le chapitre sept, nous évaluons la performance des algorithmes développés, en comparant leurs performances à celles d'algorithmes existants. Premièrement, nous montrons des résultats obtenus sur des images synthétiques en restauration d'images bruitées. Puis, nous présentons une application réelle en vision stéréoscopique.

Le huitième chapitre conclut le mémoire par une discussion des résultats et des performances des algorithmes présentés. Finalement, nous terminons en donnant un aperçu des améliorations possibles et des travaux futurs.

Chapitre 2

Graphes de flot

Ce chapitre présente une base théorique [AMJB93, CLR90] sur les graphes de flot qui sera utile à la compréhension des prochains chapitres. Ainsi, après avoir défini ce qu'est un graphe de flot, nous discutons de différentes propriétés reliées à ceux-ci. Par la suite, nous définissons les problèmes de flot maximum et de coupe de coût minimum. Puis, nous décrivons quelques algorithmes permettant de les calculer.

2.1 Graphe de flot

Un graphe de flot est un graphe dirigé $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (voir la figure 2.1), dont l'ensemble des sommets \mathcal{V} contient deux noeuds distincts : une source s et un drain t . Les arêtes du graphe \mathcal{G} sont contenues dans l'ensemble \mathcal{E} . Elles sont dirigées et elles ont leurs sommets dans l'ensemble \mathcal{V} . La fonction de capacité des arêtes du graphe est définie sur l'ensemble des paires de sommets et elle prend des valeurs non-négatives :

$$c : \mathcal{V} \times \mathcal{V} \rightarrow \mathbf{R}^+.$$

Lorsque l'arête $(u, v) \notin \mathcal{E}$, la fonction de capacité prend la valeur $c(u, v) = 0$.

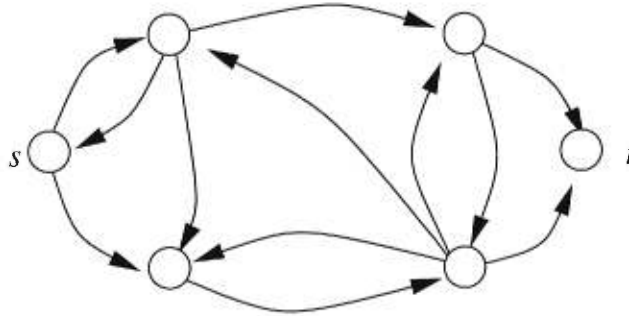


FIG. 2.1 – Illustration d'un graphe de flot avec deux noeuds terminaux : la source s et le drain t .

2.1.1 Flot

Un flot dans un graphe de flot \mathcal{G} est une fonction réelle $f : \mathcal{V} \times \mathcal{V} \rightarrow \mathbf{R}$ qui satisfait les propriétés suivantes :

Contrainte de capacité : pour tout $u, v \in \mathcal{V}$, $f(u, v) \leq c(u, v)$.

Anti-symétrie : pour tout $u, v \in \mathcal{V}$, $f(u, v) = -f(v, u)$.

Conservation du flot : pour tout $u \in \mathcal{V} - \{s, t\}$, $\sum_{v \in \mathcal{V}} f(u, v) = 0$.

La quantité $f(u, v)$ est soit positive, négative ou nulle et elle correspond à un flot entre les noeuds u et v . La valeur $(|\cdot|)$ d'un flot f se définit comme la quantité de flot arrivant au drain t :

$$|f| = \sum_{v \in \mathcal{V}} f(v, t).$$

Une expression équivalente, de la valeur du flot, est donnée par la quantité de flot émergeant de la source s . Ce qui se définit comme :

$$|f| = \sum_{v \in \mathcal{V}} f(s, v).$$

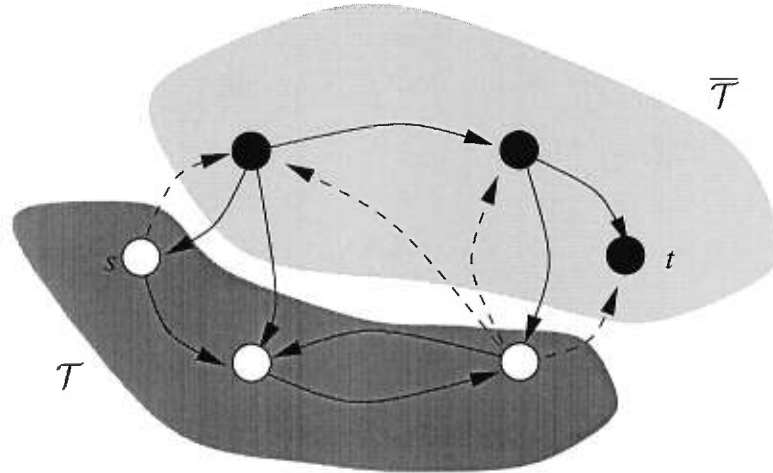


FIG. 2.2 – Graphe de flot et coupe. La représentation de la coupe comme partition de l'ensemble des sommets est caractérisée par l'ensemble des sommets dans \mathcal{T} et dans $\bar{\mathcal{T}}$. La représentation équivalente de la coupe peut aussi se voir par les arêtes en pointillées qui vont d'un sommet de \mathcal{T} vers un sommet de $\bar{\mathcal{T}}$.

2.1.2 Coupe

Une coupe dans un graphe de flot \mathcal{G} est une partition de l'ensemble des sommets \mathcal{V} en deux sous-ensembles distincts : \mathcal{T} et $\bar{\mathcal{T}} = \mathcal{V} - \mathcal{T}$, où $s \in \mathcal{T}$ et $t \in \bar{\mathcal{T}}$. Alternativement, nous pouvons définir une coupe comme étant l'ensemble des arêtes dirigées de \mathcal{T} vers $\bar{\mathcal{T}}$, ayant un sommet dans chacun des différents ensembles. Ainsi une arête $(u, v) \in \mathcal{E}$ fait partie de la coupe si $u \in \mathcal{T}$ et $v \in \bar{\mathcal{T}}$. La capacité (coût) d'une coupe correspond simplement à la somme des capacités des arêtes qu'elle contient (les arêtes de \mathcal{T} vers $\bar{\mathcal{T}}$). Ainsi,

$$c(\mathcal{T}, \bar{\mathcal{T}}) = \sum_{u \in \mathcal{T}, v \in \bar{\mathcal{T}}} c(u, v)$$

2.1.3 Flot maximum et coupe minimum

Flot maximum

Le problème de flot maximum dans un graphe de flot est simple à définir. Il survient lorsqu'on cherche à envoyer la plus grande quantité de flot possible entre les deux noeuds terminaux, c'est-à-dire entre la source s et le drain t , sans que le flot excède la capacité des arêtes (c'est-à-dire sans violer les contraintes de capacité, d'anti-symétrie et de conservation).

Théorème de flot maximum et coupe minimum

La valeur du flot maximum allant de la source s au drain t dans le graphe de flot G est égale à la capacité de la coupe de coût minimum parmi toutes les coupes possibles séparant la source du drain dans le graphe de flot G . Ainsi,

$$c(\mathcal{T}, \overline{\mathcal{T}}) = |f|,$$

où f est le flot maximum.

2.2 Algorithmes de calcul du flot maximum

Il existe beaucoup d'algorithmes de calcul du flot maximum, mais nous n'en ferons pas une revue exhaustive. Nous présentons plutôt le fonctionnement général de deux principaux types d'algorithmes. Le premier type d'algorithme se base sur la notion de chemins d'augmentation de flot de Ford et Fulkerson [FF62], alors que le second porte sur la notion de poussée de pré-flot [Kar74, GT88, GR97] pour calculer le flot maximum. De plus, ce dernier a l'avantage d'être le plus rapide.

2.2.1 Notions supplémentaires

Pour mieux comprendre les mécanismes internes de ces algorithmes, nous devons introduire quelques notions supplémentaires.

Graphe de flot résiduel

Le concept de graphe de flot résiduel \mathcal{G}_f , induit par un flot f sur un graphe \mathcal{G} , constitue la base des algorithmes de calcul du flot maximum. Étant donnée un flot f , la capacité résiduelle $c_f(u, v)$ d'une arête $(u, v) \in \mathcal{E}$ correspond à la quantité maximale de flot additionnel pouvant circuler entre les noeuds u et v par l'arête (u, v) . Ainsi, la capacité résiduelle d'une arête (u, v) est

$$c_f(u, v) = c(u, v) - f(u, v).$$

Il s'en suit que le graphe résiduel \mathcal{G}_f est constitué des noeuds de \mathcal{G} et des arêtes pouvant admettre une quantité additionnelle de flot. Ainsi, le graphe de flot résiduel se définit comme $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}_f)$, où \mathcal{E}_f est l'ensemble des arêtes résiduelles (c'est-à-dire, les arêtes pouvant encore admettre du flot) :

$$\mathcal{E}_f = \{(u, v) \in \mathcal{V} \times \mathcal{V} : c_f(u, v) > 0\}.$$

Chemin d'augmentation de flot

Étant donné le flot f dans le graphe \mathcal{G} , nous définissons un chemin d'augmentation de flot comme étant un chemin simple p allant de la source s au drain t dans le graphe résiduel \mathcal{G}_f .

Pré-flot

Un pré-flot est une fonction réelle $g : \mathcal{V} \times \mathcal{V} \rightarrow \mathbf{R}$ qui satisfait les propriétés suivantes :

$$\text{pour tout } u, v \in \mathcal{V}, g(u, v) \leq c(u, v),$$

$$\text{pour tout } u, v \in \mathcal{V}, g(u, v) = -g(v, u),$$

$$\text{pour tout } u \in \mathcal{V} - \{s, t\}, \sum_{v \in \mathcal{V}} g(u, v) \geq 0.$$

Les deux premières équations correspondent à la contrainte de capacité et d'antisymétrie d'un flot. Tandis que, la dernière équation est une relaxation de la propriété de conservation de flot.

Excès de flot

Pour un pré-flot g , la quantité de flot contenue dans noeud u s'appelle l'excès de flot $e(u)$. Il se définit comme :

$$e(u) = \begin{cases} \sum_{v \in \mathcal{V}} g(u, v), & \text{si } u \neq s \\ \infty, & \text{sinon.} \end{cases}$$

Ainsi, un noeud u est en excès de flot (par rapport au flot g), si $e(u) \geq 0$.

Étiquetage valide comme distance

Un étiquetage valide d'un pré-flot g , correspond à l'estimation de la distance d'un noeud u au drain t . Cette fonction à valeurs entières et non-négatives est définie sur l'ensemble des noeuds \mathcal{V} ,

$$d: \mathcal{V} \rightarrow \mathbf{Z}^+,$$

tel que $d(t) = 0$, $d(u) > 0$ pour tout $u \neq t$ et $d(v) \geq d(u) - 1$ pour toutes arêtes résiduelles (u, v) . Lorsqu'un étiquetage d est valide, l'estimation de la distance $d(u)$ correspond à une borne inférieure de la distance entre le noeud u et le drain t . Par conséquent, tous les chemins de u à t contiennent au moins $d(u)$ arêtes.

2.2.2 L'algorithme de Ford et Fulkerson

L'algorithme de Ford et Fulkerson [FF62] est basé sur la notion de chemins d'augmentation de flot. À chaque itération, l'algorithme cherche un chemin d'augmentation de flot, c'est-à-dire un chemin de la source vers le drain ayant une capacité non-nulle. Une fois ce chemin trouvé, l'algorithme augmente le flot f dans

le graphe de la quantité maximale de flot pouvant passer par le chemin d'augmentation choisi. L'algorithme continue ainsi jusqu'à ce qu'il n'existe plus de chemins d'augmentation de flot dans le graphe résiduel \mathcal{G}_f .

2.2.3 L'algorithme de Golberg et Tarjan

Golberg et Tarjan [GT88] ont proposé un algorithme basé sur la notion de poussée de pré-flot (notion introduite par Karzanov [Kar74]). De façon incrémentale, les pré-flots sont poussés dans le graphe, en saturant les arêtes, afin que certains noeuds aient un excès de flot. Puis, cet excès est ensuite poussé vers le drain ou retourné à la source. Il existe plusieurs algorithmes basés sur la poussée de pré-flot, mais l'algorithme de Golberg et Tarjan [GT88, GR97] est le plus rapide. Son efficacité est tirée de la combinaison des notions de pré-flot g et d'étiquetage valide d de g . Cette combinaison permet de pousser l'excès de flot de certains noeuds du graphe vers les noeuds qui sont estimés les plus près du drain dans le graphe résiduel. L'estimé de la distance au drain correspond à l'étiquetage d . Il est périodiquement mis-à-jour pour mieux correspondre au graphe résiduel courant.

Dans une première phase l'algorithme pousse du flot vers le drain t en saturant les arêtes et en conservant les excès de flot dans les noeuds (voir la figure 2.3). À la fin de cette première phase, l'excès de flot du drain t est égal à la valeur de la coupe minimale. De plus, l'ensemble des noeuds qui sont toujours connectés au drain dans le graphe résiduel engendre la coupe (ce qui forme la partition de \mathcal{V} en \mathcal{T} et $\overline{\mathcal{T}}$). Dans une seconde phase, le pré-flot est converti en flot, en retournant ce qui reste d'excès de flot à la source par les chemins estimés les plus courts. Si c'est seulement la coupe de coût minimum qui est désirée, cette dernière étape est inutile.

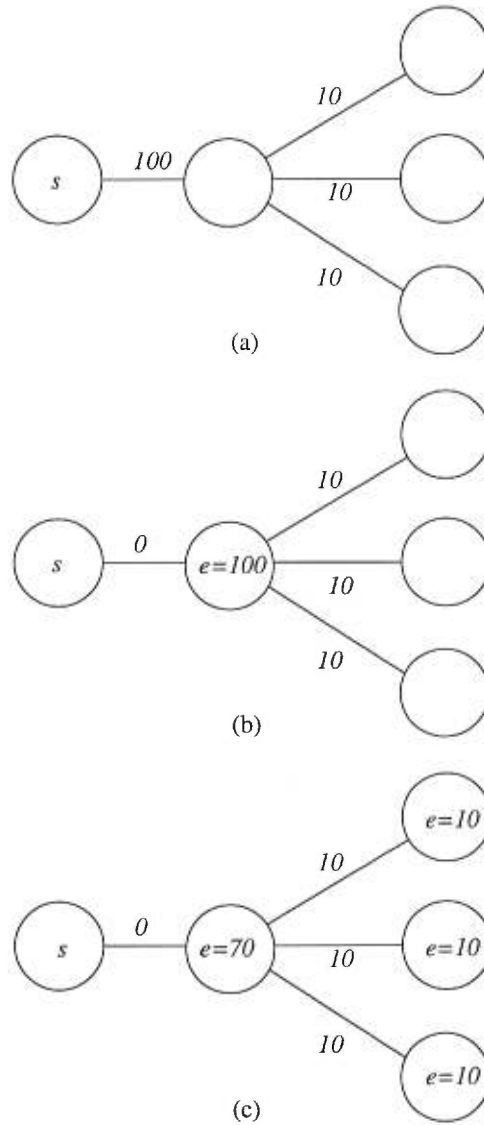


FIG. 2.3 – Illustration du fonctionnement de l’algorithme de Golberg et Tarjan. En (a) nous avons une section d’un graphe de flot. En (b), nous avons le résultat d’une poussée de 100 unités de flot de la source s vers le premier noeud. L’excès contenu dans ce noeud est $e = 100$. (c) Effet d’une poussée de 30 unités de flot (à partir du premier noeud).

Heuristique de réétiquetage global

Bien que la complexité en pire cas de l'algorithme soit $O(nm \log(n^2/m))$, où n est le nombre de noeuds et m est le nombre d'arêtes, il est possible d'obtenir de meilleures performances en moyenne. À cette fin, Golberg et Trajan [GT88] utilisent un heuristique de réétiquetage global, qui met à jour la fonction de distance en calculant la distance du plus court chemin allant de chacun des noeuds au drain dans le graphe résiduel. Cette heuristique doit être appliquée périodiquement. Selon Golberg et Rao [GR97], il semble qu'une petite constante fois le nombre de noeuds du graphe est raisonnable.

Chapitre 3

Modèle bayésien en restauration d'images

Ce chapitre introduit le problème de restauration d'images en s'inspirant des travaux de Besag [Bes86], de Boykov *et al.* [BVZ98], de Geman et Geman [GG84], de Greig *et al.* [GPS89] et de Li [Li95].

3.1 Restauration d'images

Le problème en restauration d'images est de retrouver l'étiquetage (par exemple, le niveau de gris ou la couleur) original d'une image inconnue, à partir d'une version dégradée de celle-ci. Pour y parvenir, les algorithmes de restauration sont conçus de manière à exploiter les caractéristiques du signal et de la dégradation. Une approche possible est la restauration bayésienne d'images. Elle est élégante et bien fondée mathématiquement et depuis les travaux de Geman et Geman [GG84] et de Besag [Bes86], elle a connu un essor considérable. L'intérêt des approches bayésiennes provient de la possibilité d'introduire explicitement des connaissances *a priori*, notamment sur la structure spatiale des images, dans la modélisation du mécanisme de dégradation.

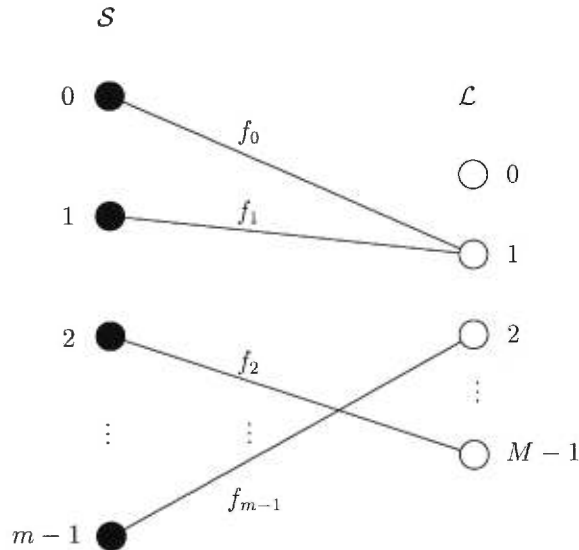


FIG. 3.1 – Illustration de la correspondance qu’engendre une configuration $f = \{f_0, \dots, f_{m-1}\}$ entre l’ensemble des sites \mathcal{S} et l’ensemble des étiquettes $\mathcal{L} = \{0, \dots, M-1\}$.

Ce chapitre fait une brève introduction aux champs de Markov, qui dans un cadre d’étiquetage bayésien, servent à modéliser les connaissances *a priori* sur la structure de l’image à retrouver. Nous discuterons par la suite des propriétés de différents modèles *a priori*.

3.2 Le problème d’étiquetage

La restauration d’images se traite comme un problème d’étiquetage discret. Ce dernier est typiquement décrit par un ensemble $\mathcal{S} = \{0, \dots, m-1\}$ de sites (pixels de l’image) et un ensemble d’étiquettes $\mathcal{L} = \{l_0, \dots, l_{M-1}\}$, où la valeur de l’étiquette l_i dépend de l’application (par exemple, cela pourrait être une valeur ordonnée comme le niveau de gris des pixels). Tel qu’illustré à la figure 3.1, un étiquetage est engendré par une configuration $f = \{f_0, \dots, f_{m-1}\}$, où $f_i \in \mathcal{L}$, qui établit une correspondance entre l’ensemble \mathcal{S} des sites et l’ensemble \mathcal{L} des

étiquettes.

Une observation x provient de la dégradation de l'étiquetage original f . Formellement, le processus de dégradation se décrit comme :

$$x = \phi(f) \odot e,$$

où le processus de dégradation est une combinaison de modifications géométriques et de flou introduit par la fonction d'étalement (réponse à l'impulsion) du système d'acquisition modélisé par $\phi(\cdot)$ et d'un bruit e issu d'un processus aléatoire qui peut être additif ou multiplicatif (défini par \odot).

Lorsque la distribution *a priori* et la fonction de vraisemblance des configurations sont connues exactement (ce qui n'est généralement pas le cas), le meilleur estimé de l'étiquetage est donné par l'estimateur maximisant le critère de Bayes. En pratique, l'estimateur choisi est celui qui maximise la probabilité *a posteriori*. Nous tentons de trouver la configuration $F = f$ la plus probable, donc celle qui maximise la probabilité *a posteriori* $Pr(F = f | X = x)$, étant donnée l'observation $X = x$. D'après le théorème de Bayes,

$$Pr(F = f | X = x) = \frac{Pr(F = f) Pr(X = x | F = f)}{Pr(X = x)},$$

où $Pr(F = f)$ est la probabilité *a priori* de l'étiquetage, $Pr(X = x | F = f)$ est la *fonction de vraisemblance* d'une observation x étant donnée une configuration $F = f$ et $Pr(X = x)$ est la probabilité d'une observation $X = x$. Cette dernière ne dépend pas de f . La configuration \hat{f} qui maximise la probabilité *a posteriori* est donnée par

$$\begin{aligned} \hat{f} &= \arg \max_f Pr(F = f | X = x), \\ &= \arg \max_f Pr(X = x | F = f) Pr(F = f), \end{aligned}$$

ce qui correspond au meilleur étiquetage bayésien possible.

Dans la théorie de Bayes, la fonction de vraisemblance et la distribution *a priori* d'une configuration sont supposées connues. En faisant des hypothèses dis-

tributionnelles, comme par exemple sur la nature du bruit, la fonction de vraisemblance permet de caractériser le processus de dégradation observé, tandis que la distribution *a priori* permet de modéliser les interactions spatiales et contextuelles d'une configuration f . Nous la modélisons par un champs aléatoire de Markov avec interdépendances locales.

3.2.1 Champ aléatoire de Markov

En statistique, les champs aléatoires de Markov (CAM) servent à analyser les dépendances spatiales et contextuelles d'un phénomène physique. Dans un problème d'étiquetage bayésien, ils servent à caractériser les distributions de probabilités des configurations possibles d'étiquettes. Avant d'entrer dans le vif du sujet et de formaliser la notion de CAM, nous allons introduire la notation utilisée.

Un champ aléatoire est une famille de variables aléatoires $F = \{F_0, \dots, F_{m-1}\}$ définie sur l'ensemble des sites \mathcal{S} . Une réalisation de ce champ est représentée par l'événement joint $F = f$, avec $f = \{f_0, \dots, f_{m-1}\}$. Pour rendre la notation moins lourde, nous dénotons par $Pr(f)$ la probabilité de l'événement joint $F = f$ et par $Pr(f_i)$ la probabilité marginale que la variable aléatoire F_i prenne la valeur f_i .

Système de voisinage

La relation spatiale entre les sites de \mathcal{S} est définie par un système de voisinage $\mathcal{N} = \{\mathcal{N}_i : i \in \mathcal{S}\}$, où \mathcal{N}_i est l'ensemble de tous les sites voisins du site i , tel que $i \notin \mathcal{N}_i$. En pratique, les systèmes de voisinage utilisés sont ceux de premier et second ordre (voir la figure 3.2). Le système de premier ordre définit l'ensemble des sites voisins comme

$$\mathcal{N}_i = \{i' \in \mathcal{S} : d^2(i, i') \leq 1, i \neq i'\},$$

tandis que dans un système de second ordre, cet ensemble est plutôt défini comme

$$\mathcal{N}_i = \{i' \in \mathcal{S} : d^2(i, i') \leq 2, i \neq i'\},$$

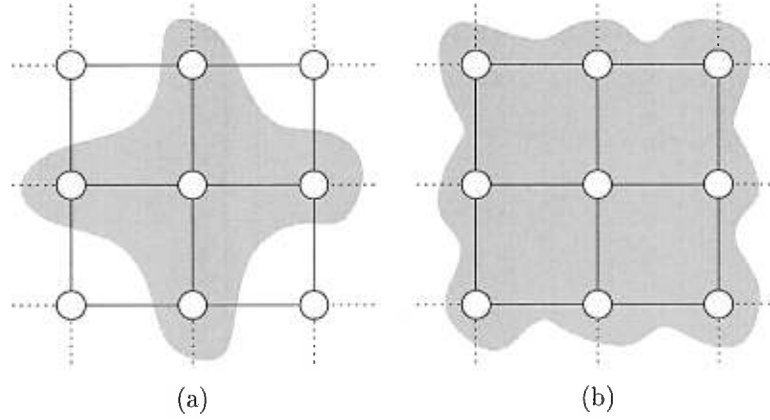


FIG. 3.2 – Systèmes de voisinage souvent utilisés en vision et traitement d’images. (a) système de premier ordre : 4-voisins. (b) système de second ordre : 8-voisins.

où $d(i, i')$ est la distance euclidienne.

Définition de champ aléatoire de Markov avec interdépendance locale

Un champ aléatoire F est dit un champ aléatoire de Markov avec interdépendance locale sur l’ensemble des sites \mathcal{S} , par rapport au système de voisinage \mathcal{N} si et seulement si les conditions suivantes sont satisfaites :

$$Pr(f_i | \{f_{i'} | i' \in \mathcal{S} - \{i\}\}) = Pr(f_i | \{f_{i'} | i' \in \mathcal{N}_i\}), \quad (3.1)$$

$$Pr(f) > 0 \text{ pour tout } f. \quad (3.2)$$

La condition (3.1) est connue comme la propriété de Markov. Elle implique que la probabilité conditionnelle que la variable aléatoire F_i prenne la valeur f_i sachant la configuration du champ F ailleurs, dépend uniquement de la valeur de F dans un voisinage du site i . La condition de positivité (3.2) implique que toutes les configurations sont possibles.

La probabilité jointe $Pr(f)$ d’un CAM est complexe. C’est pourquoi, il est difficile de définir les probabilités conditionnelles locales de façon à ce qu’elle soit consistante. Le théorème de Hammersley-Clifford règle ce problème, en établissant

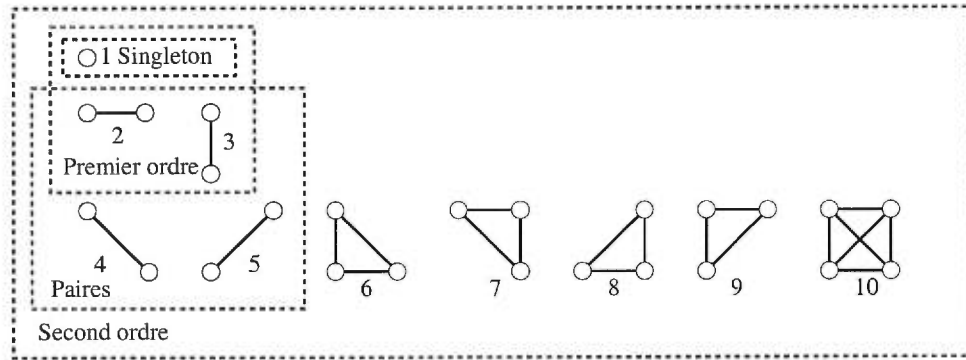


FIG. 3.3 – Les cliques associées aux systèmes de voisinages de premier et second ordre. Le système de second ordre regroupe l'ensemble de toutes les cliques présentes. Le système de premier ordre engendre les cliques ayant les formes suivantes : 1, 2 et 3. En pratique, les cliques constituées de paires de sites (formes 2, 3, 4 et 5) sont les plus utilisées.

la correspondance entre les CAM localement dépendant et les champs aléatoires de Gibbs (CAG).

3.2.2 Champs aléatoire de Gibbs

Les champs de Gibbs sont originaires de la physique statistique, où l'étude des propriétés globales d'un système de treillis était obtenue à partir de modèles locaux. L'approche servait à modéliser le comportement de matériaux ferromagnétiques en considérant seulement l'interaction de spins entre atomes voisins. Deux atomes voisins ont un potentiel positif, s'ils ont des spins opposés. L'état du treillis, qui est calculée comme la somme de tout les potentiels, est caractérisé par son énergie. Ainsi, une configuration d'atomes qui a une basse énergie est plus stable qu'une configuration de haute énergie.

Clique

Une clique c , pour un système de voisinage \mathcal{N} donné, correspond à n'importe quel sous-ensemble de sites qui sont voisins (incluant les singletons, voir la figure 3.3). L'ensemble de toutes les cliques possibles est noté \mathcal{C} .

Le regroupement des différents sites, tel que démontré à la figure 3.3 constitue le système de clique associé au système de voisinage de second ordre, alors que le système de premier ordre engendre des cliques ayant plutôt les formes 1, 2 et 3 de la figure 3.3.

Définition de champ aléatoire de Gibbs

Une famille de variables aléatoires est un CAG sur \mathcal{S} par rapport au système de voisinage \mathcal{N} si et seulement si la probabilité d'une configuration f suit une distribution de Gibbs :

$$Pr(f) = \frac{e^{-E(f)}}{Z},$$

où $Z = \sum_f e^{-E(f)}$ est une constante de normalisation appelée fonction de partition. Elle est obtenue en sommant les valeurs du numérateur sur toutes les configurations possibles f . En pratique, la complexité de cette fonction est très grande, ce qui rend son calcul impossible. Par exemple, pour un ensemble de sites \mathcal{S} contenant seulement 64×64 pixels et un ensemble \mathcal{L} contenant seulement deux étiquettes, il y a 2^{4096} configurations possibles. La fonction d'énergie $E(f)$ est définie en terme de potentiels de cliques $V_c(f)$ sur l'ensemble de toutes les cliques possibles $c \in \mathcal{C}$:

$$E(f) = \sum_{c \in \mathcal{C}} V_c(f). \quad (3.3)$$

Le Théorème de Hammersley-Clifford établit qu'un champ aléatoire F est un CAM par rapport au système de voisinage \mathcal{N} , si et seulement si F suit une distribution de Gibbs définie sur le potentiel des cliques de ce système de voisinage. Cet énoncé

d'équivalence entre CAM et CAG permet de définir la probabilité jointe d'une configuration d'un champ de Markov avec interdépendance locale.

3.2.3 Probabilité *a posteriori* et énergie associée

Pour pouvoir résoudre le problème de MAP, nous devons faire quelques hypothèses : étant donnée une configuration f , les variables aléatoires X, \dots, X_{n-1} sont conditionnellement indépendantes et les X_i ont tous la même fonction de densité conditionnelle $Pr(X_i = x_i | F_i = f_i)$. Ainsi, la fonction de vraisemblance se réduit à la forme suivante,

$$Pr(X = x | F = f) = \prod_{i \in \mathcal{S}} Pr(X_i = x_i | F_i = f_i).$$

Dans ce mémoire, nous utilisons uniquement des systèmes de voisinages \mathcal{N} , de premier et second ordre, constitués de paires de sites voisins. Toutefois, nous pourrions utiliser des systèmes plus généraux. L'ensemble des cliques constitué de paires non-ordonnées et distinctes de sites est défini comme

$$\mathcal{P}_{\mathcal{N}} = \{\{i, i'\} | i' \in \mathcal{N}_i, i, i' \in \mathcal{S}\}.$$

Puisque les cliques de $\mathcal{P}_{\mathcal{N}}$ sont non-ordonnées, nous avons que $V_{ii'}(f_i, f_{i'}) = V_{i'i}(f_{i'}, f_i)$. En faisant l'hypothèse que seules les cliques de $\mathcal{P}_{\mathcal{N}}$ ont un potentiel non-nul, nous pouvons réécrire l'énergie *a priori* $E(f)$ de l'équation (3.3) sous la forme suivante :

$$E(f) = \sum_{\{i, i'\} \in \mathcal{P}_{\mathcal{N}}} V_{ii'}(f_i, f_{i'}).$$

Maximisation de la probabilité *a posteriori*

Regroupant ce qui précède, le problème d'étiquetage s'écrit sous forme de maximisation de la probabilité *a posteriori*

$$\begin{aligned}\hat{f} &= \arg \max_f Pr(F = f | X = x), \\ &= \arg \max_f Pr(F = f) Pr(X = x | F = f), \\ &= \arg \max_f \frac{e^{-E(f)}}{Z} \prod_{i \in S} Pr(X_i = x_i | F_i = f_i),\end{aligned}$$

où \hat{f} est la configuration cherchée et $E(f) = \sum_{\{i,i'\} \in \mathcal{P}_N} V_{ii'}(f_i, f_{i'})$.

Minimisation de l'énergie associée

En pratique, nous transformons le problème de maximisation de la probabilité *a posteriori* en un problème de minimisation d'énergie, en appliquant le logarithme négatif à la probabilité *a posteriori*. Ainsi, nous obtenons :

$$\begin{aligned}\hat{f} &= \arg \min_f -\log \frac{e^{-E(f)}}{Z} \prod_{i \in S} Pr(X_i = x_i | F_i = f_i), \\ &= \arg \min_f \sum_{\{i,i'\} \in \mathcal{P}_N} V_{ii'}(f_i, f_{i'}) + \sum_{i \in S} -\log(Pr(X_i = x_i | F_i = f_i)), \\ &= \arg \min_f E(f) + E(x | f), \\ &= \arg \min_f E(f | x),\end{aligned}\tag{3.4}$$

où $E(f | x)$ est la fonction d'énergie associée à la probabilité *a posteriori* $Pr(F = f | X = x)$.

3.3 Modélisation des connaissances *a priori*

Dans un cadre bayésien, l'utilisation de connaissances *a priori* est fondamentale. Dans plusieurs applications, l'introduction d'information *a priori* permet d'améliorer substantiellement la solution ; typiquement en réduisant le bruit ou en éliminant les solutions non-désirées.

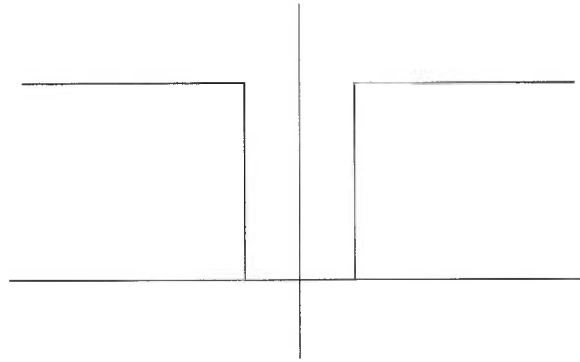


FIG. 3.4 – Graphe du modèle de Potts.

Dans cette section, nous allons expliquer les différences entre plusieurs modèles de potentiels de cliques. Nous distinguerons les différents types de restaurations, en mettant l'emphase sur leurs propriétés, leurs avantages et leurs inconvénients.

Par l'équivalence Markov-Gibbs, la conception de la fonction de distribution de la probabilité *a priori* se fait en spécifiant le potentiel des cliques V_c de la distribution de Gibbs correspondante. Ainsi, la fonction d'énergie *a priori* $E(f)$ est construite en modélisant le potentiel de cliques qui est défini sur le système de voisinage \mathcal{N} . La connaissance *a priori* la plus commune est de favoriser les solutions lisses, c'est-à-dire les solutions pour lesquels les sites voisins ont des étiquettes similaires (selon une certaine notion de similitude).

Segmentation et étiquettes non-ordonnées

Lorsque les étiquettes sont non-ordonnées (par exemple, en segmentation de texture), le problème de restauration en est un de segmentation en régions homogènes. Le modèle de Potts [Bes86, GG84, Li95] est souvent utilisé pour résoudre ce genre de problème. Dans ce modèle, les potentiels de cliques ont une valeur nulle seulement lorsque deux sites voisins sont identiquement étiquetés, autrement ils ont une valeur constante. Formellement, le modèle de Potts (illustré à la figure 3.4) est défini comme :

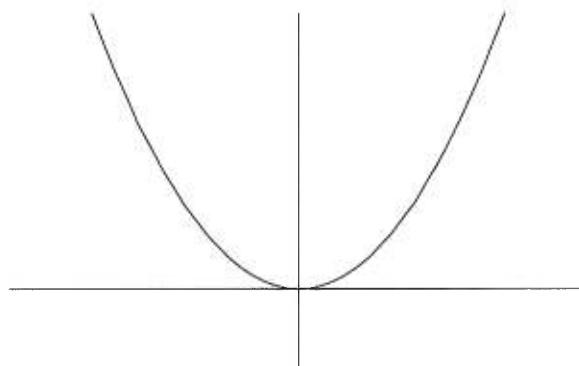


FIG. 3.5 – Graphe du potentiel quadratique.

$$V_{ii'}(f_i, f_{i'}) = \beta_{ii'}(1 - \delta(f_i - f_{i'})),$$

où $\beta_{ii'}$ est un paramètre contrôlant le degré de lissage de la restauration et $\delta(\cdot)$ est une fonction qui prend la valeur 1 lorsque son argument est nul et la valeur 0 autrement. Plus le paramètre $\beta_{ii'}$ est grand, plus l'homogénéité dans l'étiquetage est grande. Par exemple, lorsque $\beta_{ii'} = 0$, il n'y a pas d'interaction entre les sites voisins et lorsque $\beta_{ii'} = \infty$, l'interaction est maximale.

Régularisation

La restauration de surface est un exemple classique d'étiquetages ordonnés. Dans cette application, les étiquettes représentent l'élévation de la surface. Un potentiel de clique souvent utilisé est le potentiel quadratique (voir la figure 3.5) :

$$V_{ii'}(f_i, f_{i'}) = \beta_{ii'}(f_i - f_{i'})^2.$$

Ce potentiel favorise les plages uniformes et les changements lisses dans l'étiquetage. Un autre exemple, un peu plus robuste aux discontinuités, est le potentiel de clique linéaire (illustré à la figure 3.6), qui est défini comme :

$$V_{ii'}(f_i, f_{i'}) = \beta_{ii'}|f_i - f_{i'}|.$$

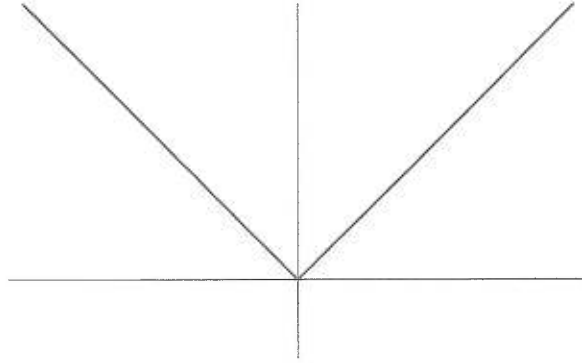


FIG. 3.6 – Graphe du potentiel linéaire.

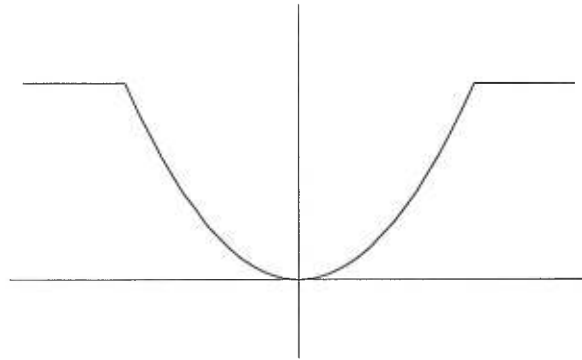


FIG. 3.7 – Graphe du potentiel quadratique tronqué.

Régularisation robuste aux discontinuités

L'utilisation de potentiel quadratique produit une pénalité proportionnelle à $(f_i - f_{i'})^2$. En présence de discontinuité, la pénalité tend à être trop grande, ce qui favorise une solution trop lisse et la disparition des discontinuités dans l'étiquetage. Il s'en suit que ce type de potentiel ne permet pas de modéliser correctement une distribution *a priori* d'une surface continue par morceaux. Une solution à ce problème consiste à saturer le potentiel de clique afin qu'il n'excède pas une certaine valeur, ce qui permet de préserver les discontinuités [BZ87]. Cette solution (voir la figure 3.7) utilise un potentiel de clique quadratique tronqué :

$$V_{ii'}(f_i, f_{i'}) = \min\{\beta_{ii'}(f_i - f_{i'})^2, \alpha_{ii'}\}.$$

Le choix de la constante $\alpha_{ii'}$ dépend des discontinuités présentes dans l'application.

Chapitre 4

Optimisation et maximum *a posteriori*

4.1 Méthodes de calcul du MAP

Comme mentionné précédemment, la solution \hat{f} du problème de MAP est définie par $\hat{f} = \arg \min_f E(f)$. En cherchant cette solution, nous nous heurtons à des problèmes combinatoires de grandes tailles ou à de la minimisation de fonctions non-convexes. Les diverses techniques disponibles se regroupent en deux grandes catégories, la recherche locale et la recherche globale. Ce chapitre présente une revue des techniques de recherche les plus communes ou qui servent de fondement aux méthodes développées dans ce mémoire. Il en existe plusieurs autres, mais nous n'en ferons pas la revue exhaustive. Dans ce qui suit nous dénotons par \mathbf{F} l'ensemble de toutes les configurations possibles.

4.1.1 Méthodes locales

Plusieurs algorithmes se basent sur un principe de recherche locale. À partir d'une configuration f de l'espace des configurations possibles \mathbf{F} , nous cherchons une solution f' (s'il en existe une) appartenant au voisinage de f , qui réduit

l'énergie, c'est-à-dire que $E(f') < E(f)$. Le processus continue ainsi jusqu'à convergence. Il existe plusieurs méthodes pour générer une configuration f' à partir de la configuration f . Ces méthodes procèdent par énumération dans le cas d'un étiquetage discret ou sont basées sur le gradient (ou autres directions de descente) dans le cas d'un étiquetage continu.

Itération des modes conditionnels

L'*itération des modes conditionnels* (IMC) est un algorithme qui a été proposé par Besag [Bes86]. Cet algorithme est itératif et produit une nouvelle configuration $f^{(n+1)}$ maximisant la probabilité conditionnelle $Pr(f_i | x, \{f_{i'}^{(n)} | i' \in \mathcal{S} - \{i\}\})$, où $f^{(n)}$ est la configuration de l'itération précédente. Deux hypothèses sont utilisées par Besag [Bes86]. Tout d'abord, il suppose l'indépendance conditionnelle des observations x_i étant données les étiquettes f_i . Ceci permet d'obtenir la forme suivante

$$Pr(x | f) = \prod_{i \in \mathcal{S}} Pr(x_i | f_i),$$

comme expression de la vraisemblance.

Ensuite, il suppose qu'une configuration f possède la propriété de Markov. Il s'en suit, par le théorème de Bayes, que

$$Pr(f_i | x, \{f_{i'}^{(n)} | i' \in \mathcal{S} - \{i\}\}) \propto Pr(x_i | f_i) Pr(f_i | \{f_{i'}^{(n)} | i' \in \mathcal{N}_i\}). \quad (4.1)$$

Soulignons que ce qui rend attrayante la méthode d'IMC, c'est qu'il est beaucoup plus simple de maximiser $Pr(f_i | x, \{f_{i'}^{(n)} | i' \in \mathcal{S} - \{i\}\})$, plutôt que de maximiser directement la probabilité *a posteriori* $Pr(f | x)$.

Lorsque le problème de maximisation de la probabilité *a posteriori* est sous forme de minimisation d'énergie, l'IMC procède plutôt selon la règle :

$$f_i^{(n+1)} \leftarrow \arg \min_{f_i} -\log(Pr(x_i | f_i)) + \sum_{i' \in \mathcal{N}_i} V_{i,i'}(f_i, f_{i'}^{(n)}),$$

ce qui permet de générer itérativement des configurations diminuant, à chacune des itérations, l'énergie *a posteriori*. En plus d'être effectuée pour chacun des sites $i \in \mathcal{S}$, cette étape est répétée jusqu'à convergence de l'algorithme. Notons que deux sites voisins ne peuvent être mis-à-jour simultanément (voir Besag [Bes86] pour plus de détails).

Les résultats obtenus avec l'IMC sont dépendants de la configuration initiale $f^{(0)}$. Bien qu'il n'existe pas de méthodes permettant de déterminer une configuration initiale garantissant la qualité du résultat final, elle est souvent obtenue par un maximum de vraisemblance, ce qui correspond à une solution ne tenant pas compte du contexte. Ainsi, une configuration initiale possible est donnée par

$$f^{(0)} = \arg \min_f E(x | f).$$

Lorsque le potentiel de clique $V_{ii'}(f_i, f_{i'}) = 0$ pour tout $f_i, f_{i'} \in \mathcal{L}$, le résultat de la convergence de l'algorithme d'IMC correspond à celui d'un étiquetage par un maximum de vraisemblance. Tandis qu'une valeur infinie du potentiel de clique, $V_{ii'}(f_i, f_{i'}) = \infty$ pour tout $f_i, f_{i'} \in \mathcal{L}$, entraîne la convergence de l'IMC vers un étiquetage équivalent à un "maximum de votes" dans le voisinage utilisé, ce qui ne correspond pas à l'optimum global de la probabilité *a posteriori*, car cet étiquetage devrait être constant et de valeur égale à la moyenne des étiquettes observées.

4.1.2 Méthodes globales

Pour minimiser globalement une fonction objective, il faut trouver tous les minima locaux et démontrer qu'il n'en existe pas d'autre. D'ordre général, pour le type de fonctions d'énergie considérées, il n'existe pas d'algorithmes efficaces et il faut énumérer les minima locaux de façon exhaustive. Ainsi, nous avons le choix de trouver l'optimum global exact à partir d'un algorithme de recherche prenant des temps de calculs intolérables ou nous pouvons chercher une solution localement optimale, ce qui prend en général, un temps de calcul acceptable.

Il existe plusieurs approches, mais les méthodes les plus souvent utilisées (en restauration bayésienne d'images) sont le recuit et la recherche aléatoire. Dans la recherche aléatoire, une nouvelle configuration est générée suivant une certaine distribution. Parfois, ces configurations augmentent l'énergie du système, ce qui permet d'échapper aux mauvais minima locaux.

Le recuit, quant à lui, est introduit dans la recherche locale afin d'éviter les mauvais minima locaux. Il existe deux grands courants de pensée sur les méthodes de recuit. Premièrement, il y a les approches stochastiques comme le recuit simulé de [Kir83] (Geman et Geman [GG84] l'ont adapté au problème de MAP en restauration d'images), qui génère une nouvelle configuration à partir d'une distribution donnée. Deuxièmement, il y a les méthodes déterministes comme la non-convexité graduée de Blake et Zisserman [BZ87], qui utilisent une recherche locale pour générer la prochaine configuration. Les algorithmes stochastiques ont l'avantage d'avoir une justification théorique, tandis que les méthodes déterministes demeurent des heuristiques et ils ne peuvent garantir la convergence vers l'optimum global.

Recuit simulé

Le recuit simulé [Kir83] est un algorithme s'inspirant de la physique statistique et fût adapté au problème de MAP par Geman et Geman [GG84]. Kirkpatrick *et al.* [Kir83] présentent le recuit simulé comme une approximation du processus de refroidissement d'une substance liquide vers un solide. L'objectif ultime est d'amener le système dans son état d'énergie minimale. Plutôt que d'utiliser une approche comme l'IMC ou toutes autres méthodes de recherches locales, une méthode de recherche aléatoire comme l'échantillonneur de Gibbs [GG84] est utilisée pour générer la prochaine configuration. La recherche aléatoire est contrôlée par une suite décroissante de températures T (voir plus loin dans le texte pour une explication). Une nouvelle configuration f ne réduit pas nécessairement l'énergie, car des accroissements sont parfois permis pour permettre d'échapper aux mi-

nima locaux. Ainsi, lorsque la température est élevée, de grands changements ou accroissements de l'énergie sont permis ; lorsque la température est basse, seul de petits changements et de faibles accroissements peuvent être acceptés et finalement lorsque la température est près de zéro, aucun accroissement n'est permis.

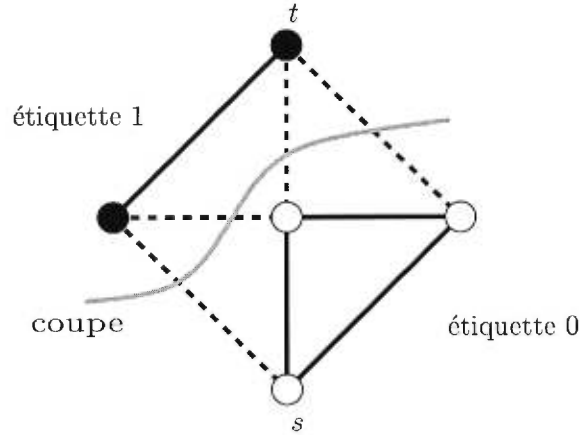
En résumé, l'algorithme de recuit simulé de Geman et Geman [GG84] débute à température élevée avec une configuration initiale $f^{(0)}$. Ensuite, pour chaque site i on remplace $f_i^{(n)}$ de la configuration courante, par $f_i^{(n+1)}$ générée de la distribution $Pr(x_i | f_i)Pr(f_i)$ avec une probabilité $(Pr(x_i | f_i)Pr(f_i))^{(1/T)}$. Ensuite, la température est réduite selon la suite décroissante choisie. Ces étapes sont répétées tant que la température n'est pas arrivée à zéro.

La convergence vers le minimum global de la fonction d'énergie *a posteriori* est théoriquement garantie, mais le nombre d'itérations requis, pour diminuer la température à zéro, est comparable au nombre de configurations possibles. En pratique, des heuristiques sont utilisés pour réduire la température. Geman et Geman [GG84] suggèrent la fonction logarithmique $T(n) = C / \log(n + 1)$, qui décroît à chaque itération. Le lecteur remarquera que la méthode d'IMC correspond au cas particulier où l'on congèle instantanément le système en posant $T = 0$ dès le départ.

Non-convexité graduée

La non-convexité graduée de Blake et Zisserman [BZ87] est un algorithme de recuit déterministe permettant d'approximer la solution globale lors de minimisation de fonction non-convexe. Cette algorithme permet d'éviter les mauvais minima locaux et de trouver une bonne approximation du minimum global en un temps de calcul raisonnable.

Dans la non-convexité graduée, un paramètre de convexité est introduit dans la fonction objective à minimiser, ainsi la fonction objective $E(f)$ devient la fonction d'énergie $E_\gamma(f)$, où γ est le paramètre de convexité. Nous ne donnons pas de

FIG. 4.1 – Graphe de la formulation de Greig *et al.* [GPS89].

détails sur la forme de $E_\gamma(f)$, car cela dépend de l'application. L'idée générale est de débiter la minimisation avec un paramètre de convexité suffisamment grand pour que la fonction devienne convexe, puis une recherche locale est effectuée jusqu'à l'obtention d'un minimum, possiblement global. Le paramètre de convexité est ensuite diminué et le processus recommence jusqu'à ce que le paramètre de convexité atteigne la valeur ciblée. Ainsi, l'algorithme donne une suite de minima qui converge vers une approximation du minimum global.

Coupe minimum et MAP

Greig *et al.* [GPS89] ont proposé une technique de calcul du MAP, pour la restauration d'images binaires (un modèle de Potts est utilisé comme potentiel de clique). Leur approche est basée sur le calcul du flot maximum et de la coupe minimum dans un graphe de flot particulier (voir la figure 4.1). Le graphe contient un noeud par site de \mathcal{S} (ces noeuds sont appelés noeuds sites), ainsi que deux noeuds terminaux : la source s et le drain t . Les arêtes qui engendrent des chemins entre les terminaux (de s vers t) sont définies de la source aux noeuds sites et des noeuds sites vers le drain. Ces connexions ont une capacité égale à la vraisemblance d'assigner l'étiquette 0 ou 1 respectivement. Les noeuds sites sont connectés entre

eux par des arêtes définies par le système de voisinage \mathcal{N} utilisé et dont la capacité est égale au coût d'une discontinuité (selon le modèle de Potts) dans l'étiquetage.

La configuration maximisant la probabilité *a posteriori* est obtenue de façon efficace en calculant la coupe de coût minimum dans le graphe. Bien que cette méthode soit restreinte aux images binaires, elle n'en est pas moins intéressante, car une recherche exhaustive de l'optimum global dans l'espace des configurations possibles (2^{4096} pour une petite image de 64×64 pixels) est prohibitive (coûteuse en temps de calcul).

La technique de Greig *et al.* [GPS89] est un cas particulier de la méthode de MAP exact (lorsque \mathcal{L} contient deux étiquettes) présentée au chapitre 5. Ainsi, pour avoir plus de renseignement sur la structure du graphe et sur la fonction de coût de ses arêtes, le lecteur peut se référer à ce chapitre.

Coupe multi-terminal et MAP

Boykov *et al.* [BVZ98] ont démontré que le problème de restauration avec étiquettes non-ordonnées, utilisant un modèle de Potts comme potentiel de clique, est équivalent à un problème de coupe multi-terminal dans un graphe particulier. Ce graphe (montré à la figure 4.2) contient un noeud pour chaque site de \mathcal{S} (noeuds "sites") et ainsi qu'un noeud "terminal" pour chacune des étiquettes de \mathcal{L} . Les noeuds sites sont connectés entre eux par une arête située entre deux noeuds sites voisins selon le système de voisinage \mathcal{N} . La capacité de ces arêtes correspond au coût d'une discontinuité (selon le modèle de Potts) dans l'étiquetage. Les connexions entre noeuds sites et noeuds terminaux engendrent une série de chemins entre les sommets terminaux. Ces arêtes ont une capacité modélisant la vraisemblance qu'un site soit étiqueté par l'étiquette associée au noeud terminal correspondant.

Boykov *et al.* [BVZ98] ont démontré que minimiser l'énergie est équivalent à trouver la coupe multi-terminal de coût minimum dans ce graphe. Dans le graphe

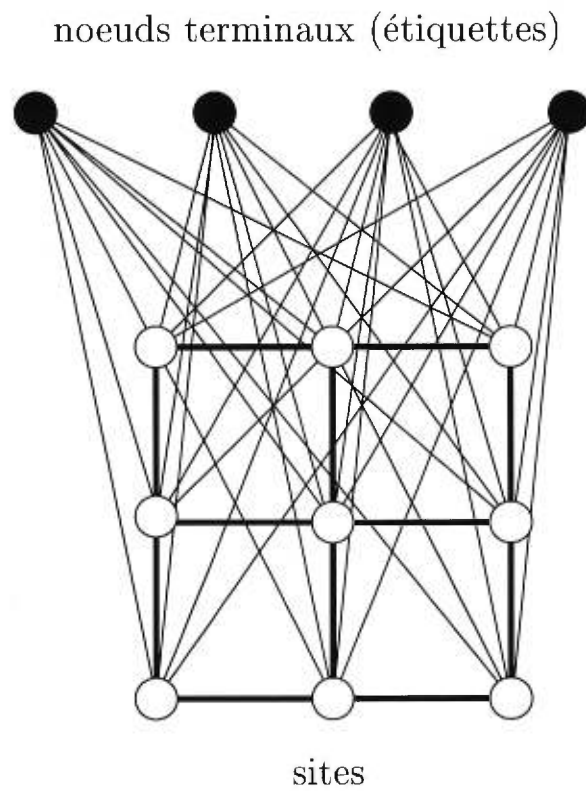


FIG. 4.2 – Graphe de la formulation multi-terminal de Boykov *et al.* [BVZ98]. Les noeuds blancs correspondent aux sites (pixels) de \mathcal{S} , tandis que les noeuds noirs sont les terminaux (étiquettes en segmentation).

induit par la soustraction des arêtes de la coupe de l'ensemble des arêtes du graphe, les noeuds terminaux ne sont plus connectés entre eux. Tandis que les noeuds sites sont chacun connectés à exactement un des noeuds terminaux (pas nécessairement distincts), ce qui engendre un étiquetage qui minimise l'énergie (maximise la probabilité *a posteriori*).

Le problème de coupe multi-terminal de coût minimum est un problème NP-complet [BVZ98] qui ne peut être solutionné efficacement. Boykov *et al.* [BVZ98] utilisent plutôt une méthode, approximative et itérative, ne garantissant pas la convergence vers l'optimum global, mais plutôt vers une approximation de celui-ci. Un avantage de cet algorithme de calcul du MAP est qu'à chaque itération, l'algorithme échange simultanément l'étiquette de plusieurs sites à la fois. Par contre, cette formulation, sous forme de coupe multi-terminal, contraint à l'utilisation du modèle de Potts comme potentiel de clique (voir la figure 3.4) :

$$V_{i'v}(f_i, f_{i'}) = \beta_{i'v}(1 - \delta(f_i - f_{i'})).$$

Par conséquent, la méthode de Boykov *et al.*[BVZ98] est restreinte à une classe de problèmes où les étiquettes sont non-ordonnées, tel que la segmentation.

Chapitre 5

Minimisation d'énergie et graphe de flot

Dans ce chapitre, nous démontrons qu'un problème de coupe minimum, dans un graphe de flot particulier, peut être utilisé pour optimiser globalement et efficacement la fonction d'énergie *a posteriori* suivante :

$$E(f|x) = \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}}} V_{ii'}(f_i, f_{i'}) + \sum_{i \in \mathcal{S}} -\log Pr(x_i|f_i), \quad (5.1)$$

lorsque le potentiel de clique a la forme linéaire suivante :

$$V_{ii'}(f_i, f_{i'}) = \sum_{\min\{f_i, f_{i'}\} < k \leq \max\{f_i, f_{i'}\}} \beta_{ii'k}, \quad (5.2)$$

où $\beta_{ii'k} \geq 0$ et f_i est une quantité ordonnée. Le cas où le potentiel de clique est défini comme :

$$V_{ii'}(f_i, f_{i'}) = \beta_{ii'} |f_i - f_{i'}|$$

est un cas particulier de l'équation (5.2), que nous obtenons en posant $\beta_{ii'k} = \beta_{ii'}$ pour tout k .

5.1 Une généralisation de Roy et Cox [RC98]

Tout abord, nous présentons un résultat de Roy et Cox [RC98] et nous expliquons comment généraliser leur technique en un problème de MAP. Dans leur article, Roy et Cox [RC98] utilisent, sans toutefois en faire la justification mathématique, une technique de coupe minimum dans un graphe de flot pour établir la correspondance stéréoscopique d'une série d'images. Nous proposons de définir la fonction d'énergie exacte qu'ils cherchent à minimiser, en traduisant leur approche, comme un problème de MAP avec potentiel de clique linéaire (voir Bonneville *et al.* [BRME98]).

Dans un premier temps, nous utilisons un graphe identique à celui de Roy et Cox [RC98]. Nous démontrons, par un contre-exemple, que leur approche ne permet pas toujours d'obtenir la solution optimale au problème de MAP avec potentiel de clique linéaire. Ensuite, nous effectuons une modification de la fonction de capacité du graphe et nous expliquons comment cela permet d'obtenir une solution optimale au problème de minimisation de l'énergie *a posteriori* avec potentiel de clique linéaire. La modification de la fonction de capacité que nous utilisons est un compromis entre la solution d'Ishikawa et Geiger [IG98b, IG98a] et un résultat de Boykov *et al.* [BVZ98]. Par abus de langage, nous appelons cet algorithme : l'algorithme de maximum *a posteriori* exact (MAPE).

5.1.1 Le graphe

La transposition du problème de minimisation de l'énergie se fait en construisant un graphe de flot $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ approprié. Nous structurons le graphe en définissant l'ensemble des noeuds \mathcal{V} , l'ensemble des arêtes \mathcal{E} et la fonction de capacité $c(a)$, où $a \in \mathcal{E}$.

L'ensemble des noeuds se définit comme :

$$\mathcal{V} = \mathcal{V}' \cup \{s, t\},$$

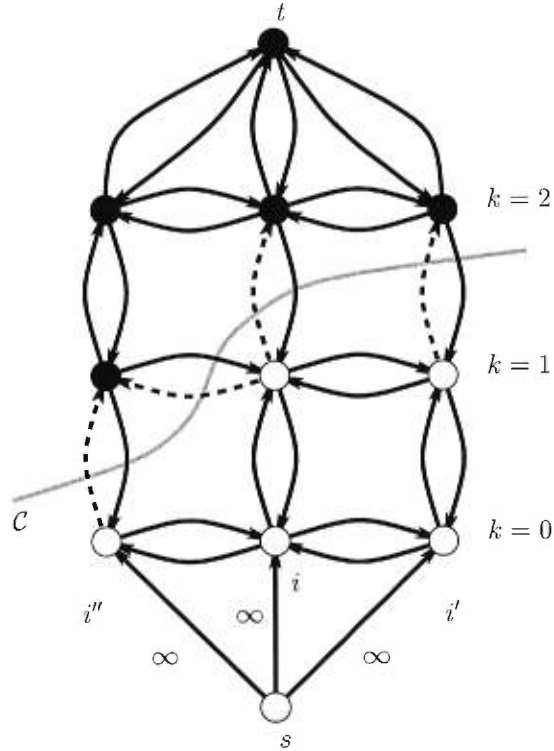


FIG. 5.1 – Représentation du graphe de flot correspondant à une image 1D avec 3 pixels $\{i'', i, i'\}$ (de la gauche vers la droite), lorsque le système de voisinage \mathcal{N} est de premier ordre (graphe de Roy et Cox [RC98]). La coupe minimum \mathcal{C} , formée des arêtes en pointillées, sépare le graphe en deux ensembles de noeuds, ceux connectés à la source s (noeuds blancs, $\in \mathcal{T}$) et les autres connectés au drain t (noeuds noirs, $\in \overline{\mathcal{T}}$). Les arêtes sortant de la source ont une capacité infinie.

où s et t sont la source et le drain et $\mathcal{V}' = \{(i, k) \mid i \in \mathcal{S}, k \in \mathcal{L}\}$ est un ensemble de noeuds associé à l'ensemble de toutes les configurations possibles d'étiquettes. La structure des noeuds et des connections dans le graphe \mathcal{G} est illustrée à la figure 5.1 pour un système de voisinage 2-voisins \mathcal{N} . Les arêtes de ce graphe sont dirigées. Toutefois, les arêtes et leurs inverses ont la même capacité.

Il y a deux ensembles disjoints d'arêtes qui composent \mathcal{E} :

$$\mathcal{E} = \mathcal{E}_v \cup \mathcal{E}_p,$$

où \mathcal{E}_v et \mathcal{E}_p représentent les ensembles d'arêtes de vraisemblance et de pénalité respectivement.

Les arêtes de vraisemblance (\mathcal{E}_v) engendrent des chemins allant de la source s au drain t de la forme :

$$s \rightarrow (i, 0) \rightarrow (i, 1) \rightarrow \dots \rightarrow (i, M - 1) \rightarrow t,$$

pour tous les sites $i \in \mathcal{S}$. Les arêtes directement connectées à la source ont une capacité infinie :

$$c(s, (i, 0)) = \infty.$$

Ainsi, elles ne sont jamais saturées par un flot de taille finie, ce qui implique qu'elles ne font jamais partie de la coupe de coût minimum. Les autres arêtes de vraisemblance $((i, k), (i, k + 1))$ et leurs inverses $((i, k + 1), (i, k))$ ont une capacité finie $c((i, k), (i, k + 1)) = c((i, k + 1), (i, k))$, tel que

$$c((i, k), (i, k + 1)) = -\log(\text{Pr}(x_i \mid f_i = k)).$$

Afin de simplifier la notation, nous supposons pour tout $i \in \mathcal{S}$, que les sommets (i, M) réfèrent tous au sommet t . Par conséquent, les arêtes $((i, M - 1), (i, M))$ correspondent aux arêtes $((i, M - 1), t)$ du graphe \mathcal{G} .

Les connections engendrées par l'ensemble des arêtes de pénalité \mathcal{E}_p servent à modéliser le potentiel de cliques. Dans un système de cliques, il existe une

interaction entre deux sites $i, i' \in \mathcal{S}$ seulement s'ils sont voisins dans \mathcal{N} . Comme les cliques, c'est par le système de voisinage utilisé que nous définissons les arêtes de pénalité. Par conséquent, une connection existe entre les noeuds (i, k) et (i', k) , seulement si $i' \in \mathcal{N}_i$ et ce pour tout $i, i' \in \mathcal{S}$ et $k \in \mathcal{L}$.

La fonction de capacité des arêtes de pénalité \mathcal{E}_p est définie comme :

$$c((i, k), (i', k)) = \beta_{i'k},$$

où $i' \in \mathcal{N}_i$.

5.1.2 Un contre-exemple

La coupe de coût minimum, du graphe de Roy et Cox, ne modélise pas correctement la fonction d'énergie (5.1). À la figure 5.2, nous illustrons un contre-exemple¹ qui démontre que la coupe de coût minimum du graphe de flot contient plusieurs arêtes de vraisemblance pour un même site. Par conséquent, le coût de cette coupe ne correspond à aucun étiquetage associé à la fonction d'énergie (5.1).

5.1.3 Un graphe modifié

Nous proposons de modifier la formulation de Roy et Cox [RC98], afin que la coupe de coût minimum corresponde à un étiquetage minimisant l'équation (5.1).

Quelques travaux récents [BVZ98, IG98b, IG98a], portant sur des graphes semblables, permettent de résoudre ce problème. Boykov *et al.* [BVZ98] utilisent aussi un graphe non-dirigé, mais ils ajoutent une grande constante positive à la fonction de capacité des arêtes de vraisemblance. En pratique, pour des raisons de convergence des algorithmes de calcul du flot maximum, les capacités du graphe sont normalisées pour être entières. Donc, l'ajout d'une constante arbitrairement grande peut occasionner des problèmes de normalisation des capacités du graphe. Ishikawa et Geiger [IG98b, IG98a] utilisent plutôt un graphe de flot possédant des

¹Ce contre-exemple est issu d'une discussion avec le professeur Neil Stewart

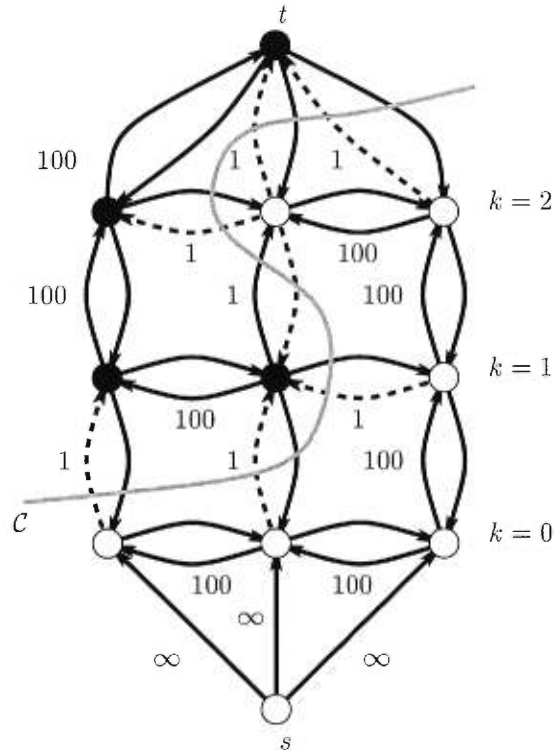


FIG. 5.2 – Voici un graphe de flot correspondant à la formulation de Roy et Cox [RC98], qui démontre que cette formulation ne garantit pas toujours une solution qui correspond à un étiquetage valide selon l'équation (5.1). Les capacités montrées sont celles du graphe original (pas celles du graphe résiduel). La capacité des arêtes avants et inverses sont identiques (comme dans un graphe non-dirigé). La coupe correspond aux arêtes en pointillés. La coupe \mathcal{C} est de coût minimum, car toutes les autres coupes contiennent au moins une arête de coût 100.

arêtes de vraisemblance dirigées, tel que les arêtes inverses possèdent une capacité infinie. Théoriquement, ceci résout le problème, bien qu'en pratique l'implémentation d'arêtes de coûts infinis en chacun des noeuds du graphe peut augmenter substantiellement le temps de calcul du flot maximum. Dans leur article Golberg et Tarjan [GT88] ont démontré que la complexité théorique dépend de la capacité maximale d'une arête, donc plus la capacité est grande, plus le temps de calcul est long.

Notre Solution

Notre solution utilise, au lieu de ces coûts infinis, des constantes positives judicieusement choisies. Le choix de ces constantes est fait, afin que les arêtes inverses ne soient jamais saturées par un flot, entre s et t , dans le graphe \mathcal{G} . Ainsi, les arêtes inverses ne font jamais partie de la coupe de coût minimum, ce qui garantit l'optimalité de la solution au problème de minimisation d'énergie de l'équation (5.1).

Le nouveau graphe \mathcal{G} a une structure identique au graphe de Roy et Cox [RC98], à l'exception de la fonction de capacité qui est différente. Les capacités des arêtes de pénalité demeurent les mêmes :

$$c((i, k), (i', k)) = \beta_{i'k},$$

où $\{i, i'\} \in \mathcal{P}_N$. Le coût d'une arête de vraisemblance avant (arête dirigée vers le drain) est identique :

$$c((i, k), (i, k + 1)) = -\log(\text{Pr}(x_i | f_i = k)).$$

Tandis que la capacité de son inverse est

$$c((i, k + 1), (i, k)) = K_{ik},$$

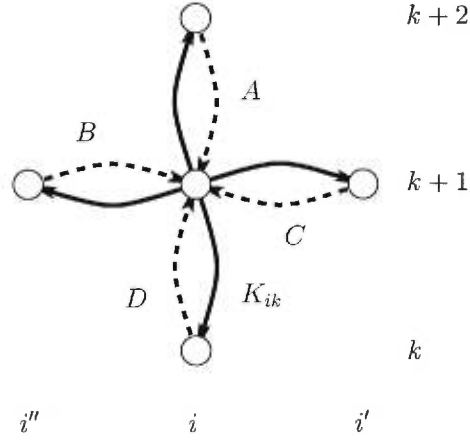


FIG. 5.3 – Les capacités incidentes au noeud $(i, k+1)$ sont : A, B, C et D. Pour que l'arête $((i, k), (i, k+1))$ ne soit jamais saturée, il faut que sa capacité respecte : $K_{ik} > A + B + C$. La capacité D n'entre pas dans cette somme à cause de la propriété de conservation du flot.

où la constante K_{ik} est définie récursivement comme :

$$\begin{aligned} K_{iM-1} &= 0, \\ K_{ik} &= \sum_{i' \in \mathcal{N}_i} c((i, k), (i', k)) + K_{ik+1} + \epsilon, \end{aligned} \quad (5.3)$$

tel que $0 \leq k \leq M - 2$ et $\epsilon > 0$.

5.1.4 La coupe minimum minimise l'énergie

Dans cette section, nous démontrons que la coupe de coût minimum du graphe \mathcal{G} décrit à la section 5.1.3 permet d'obtenir la configuration f qui minimise globalement la fonction d'énergie postérieure de l'équation (5.1).

Tout d'abord, nous définissons l'étiquetage $f^{\mathcal{C}}$ engendré par la coupe \mathcal{C} comme :

$$f_i^{\mathcal{C}} = k, \text{ si } ((i, k), (i, k+1)) \in \mathcal{C}. \quad (5.4)$$

Lemme 1 *Dans le graphe \mathcal{G} , si une arête de vraisemblance inverse $((i, k+1), (i, k))$ a une capacité $c((i, k+1), (i, k)) = K_{ik}$ alors elle ne sera jamais saturé par un flot.*

Preuve. La quantité de flot maximale pouvant arriver au noeud $(i, k+1)$ correspond à la somme des capacités des arêtes incidentes à ce noeud (voir la figure 5.3) :

$$c_{\text{incidente}} = c((i, k+2), (i, k+1)) + \sum_{i' \in \mathcal{N}_i} c((i, k+1), (i', k+1))$$

Par la propriété de conservation du flot et puisque $c_{\text{incidente}} < K_{ik}$, l'arête de vraisemblance inverse $((i, k+1), (i, k))$ ne peut être saturé par un flot dans le graphe \mathcal{G} . \square

Lemme 2 *Si pour un site $i \in \mathcal{S}$, il existe plus d'une arête de vraisemblance (avant ou inverse) dans une coupe \mathcal{C} du graphe \mathcal{G} , alors une de ces arêtes est une arête de vraisemblance inverse.*

Preuve. Soit le site $i \in \mathcal{S}$. Sans perte de généralité, choisissons deux arêtes de vraisemblance $a_{ik}, a_{ik'} \in \mathcal{C} \cap \mathcal{E}_v$ consécutives, c'est-à-dire que : pour tout $k'' \in \mathcal{L}$, avec $k < k'' < k'$ nous avons $a_{ik''} \in \mathcal{E}_v$ et $a_{ik''} \notin \mathcal{C}$.

Nous allons maintenant montrer qu'une de ces arêtes (a_{ik} ou $a_{ik'}$) est une arête inverse (voir la figure 5.4). Nous savons que

$$a_{ik} = ((i, k), (i, k+1)) \text{ ou } a_{ik} = ((i, k+1), (i, k))$$

et

$$a_{ik'} = ((i, k'), (i, k'+1)) \text{ ou } a_{ik'} = ((i, k'+1), (i, k')).$$

Supposons que le noeud $(i, k) \in \mathcal{T}$ (le cas où $(i, k) \in \overline{\mathcal{T}}$ est semblable). Ceci implique que $(i, k'') \in \overline{\mathcal{T}}$ pour tout $k'' \in \mathcal{L}$ tel que $k < k'' \leq k'$, car pour tout k'' tel que $k < k'' < k'$ nous avons que $a_{ik''} \notin \mathcal{C}$.

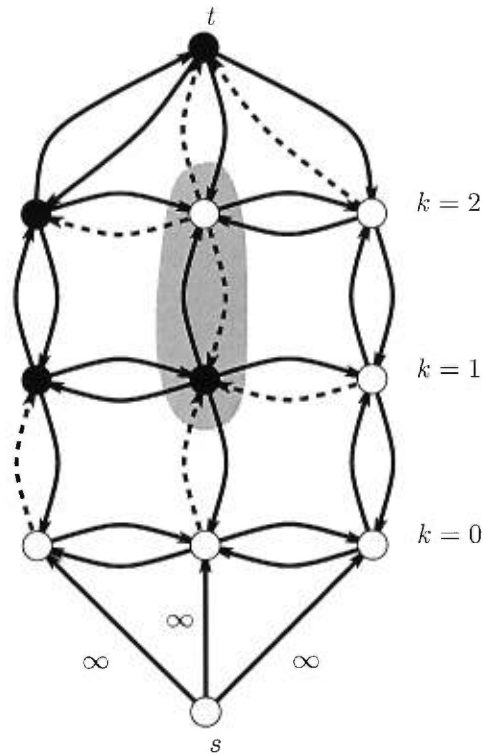


FIG. 5.4 – Illustration d'une coupe (arêtes pointillées) possédant plus d'une arêtes de vraisemblance pour le site i . La région rehaussée en gris contient une arête inverse. Cette arête est coupée pour qu'il n'y ait pas de connection allant de l'ensemble \mathcal{T} (noeuds blancs) vers l'ensemble $\bar{\mathcal{T}}$ (noeuds noirs).

Puisque le noeud $(i, k') \in \overline{\mathcal{T}}$ et que l'arête $a_{ik'} \in \mathcal{C}$, nous avons que

$$a_{ik'} = ((i, k' + 1), (i, k')) \in \mathcal{C},$$

car l'arête $a_{ik'}$ crée un chemin de \mathcal{T} vers $\overline{\mathcal{T}}$, c'est-à-dire qu'elle établit un chemin entre la source et le drain. Donc, une de ces arêtes est une arête de vraisemblance inverse. \square

Théorème 1 *Soit la coupe de coût minimum \mathcal{C} associée au flot maximum dans le graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, avec la fonction de capacité $c(\cdot)$. Pour tout site $i \in \mathcal{S}$, il existe un et un seul $k \in \mathcal{L}$ tel que l'arête de vraisemblance $a_{ik} \in \mathcal{C}$. De plus, cette arête est une arête avant, c'est-à-dire que $a_{ik} = ((i, k), (i, k + 1))$.*

Preuve. Premièrement, nous devons montrer qu'il existe au moins un $k \in \mathcal{L}$ tel que l'arête de vraisemblance $a_{ik} = ((i, k), (i, k + 1)) \in \mathcal{C}$, pour tout les sites $i \in \mathcal{S}$. Pour chaque $i \in \mathcal{S}$, il existe un chemin de s vers t dans \mathcal{G} de la forme :

$$s \rightarrow (i, 0) \rightarrow (i, 1) \rightarrow \cdots \rightarrow (i, M - 1) \rightarrow t.$$

Puisque toutes les coupes de \mathcal{G} doivent isoler le noeud source s du drain t , alors la coupe \mathcal{C} doit briser ce chemin. Par conséquent, elle doit contenir au moins une arête de vraisemblance de la forme $a_{ik} = ((i, k), (i, k + 1))$, pour $k \in \mathcal{L}$, car $(s, (i, 0))$ a une capacité infinie.

Maintenant, il faut prouver qu'il y a un seul $k \in \mathcal{L}$ tel que l'arête de vraisemblance $a_{ik} \in \mathcal{C}$, pour tout site $i \in \mathcal{S}$. Par contradiction, supposons qu'il existe plus d'une arête de vraisemblance pour un même site $i \in \mathcal{S}$. Par le lemme 2, une de ces arêtes est une arête inverse. Par le lemme 1, aucun flot ne peut saturer une arête inverse. Donc, aucune arête inverse ne fait partie de \mathcal{C} , ce qui est une contradiction. Donc, il existe une et une seule arête de vraisemblance par site $i \in \mathcal{S}$. De plus, c'est une arête avant. \square

Corollaire 1 *La configuration $f^{\mathcal{C}}$ issue de la coupe de coût minimum minimise l'énergie a posteriori $E(f^{\mathcal{C}}|x)$.*

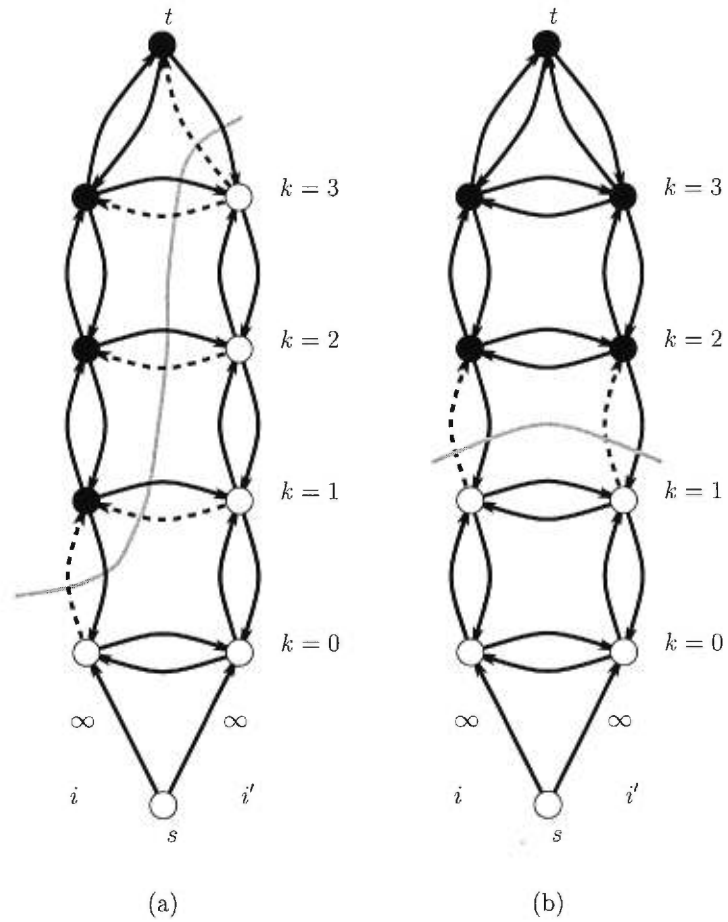


FIG. 5.5 – Illustration de l'apparence d'une coupe entre les sites i et i' . Les sommets blancs sont dans l'ensemble \mathcal{T} , tandis que les sommets noirs sont dans $\bar{\mathcal{T}}$. (a) Lorsque $f_i = 0 < f_{i'} = 3$, (b) lorsque $f_i = f_{i'} = 1$.

Preuve. Nous devons démontrer que le coût de la coupe de coût minimum correspond à l'énergie à minimiser. La capacité de la coupe \mathcal{C} se décompose sous la forme suivante :

$$|\mathcal{C}| = |\mathcal{C} \cap \mathcal{E}_v| + |\mathcal{C} \cap \mathcal{E}_p|. \quad (5.5)$$

Par le théorème 1 et l'équation (5.4), nous avons que la part de la capacité engendrée par les arêtes de vraisemblances est :

$$\begin{aligned} |\mathcal{C} \cap \mathcal{E}_v| &= \sum_{a \in \mathcal{C} \cap \mathcal{E}_v} c(a) \\ &= \sum_{i \in \mathcal{S}} -\log Pr(x_i | f_i^{\mathcal{C}}). \end{aligned} \quad (5.6)$$

Il reste à compiler la contribution des arêtes de pénalité. Par le théorème 1, il y a une unique arête de vraisemblance par site de \mathcal{S} , alors supposons que les sites $i, i' \in \mathcal{S}$ sont étiquetés $f_i = k$ et $f_{i'} = k'$ respectivement.

Si $k = k'$, il existe aucun $k'' \in \mathcal{L}$, tel que $(i, k'') \in \mathcal{T}$ et $(i', k'') \in \overline{\mathcal{T}}$, ou vice-versa. Donc, pour tout $k'' \in \mathcal{L}$, l'arête $((i, k''), (i', k''))$ ne fait pas partie de la coupe \mathcal{C} (voir la figure 5.5 (b)).

Dans le cas $k < k'$, par le théorème 1, nous avons pour $k < k'' < k'$, que les sommets (i, k'') et (i', k'') ne font pas partie du même ensemble, c'est-à-dire que $(i, k'') \in \mathcal{T}$ et $(i', k'') \in \overline{\mathcal{T}}$. Donc, l'arête $((i, k''), (i', k''))$ fait partie de la coupe \mathcal{C} et sa capacité est $c((i, k''), (i', k'')) = \beta_{ii'k}$ (voir la figure 5.5 (a)).

La contribution des arêtes de pénalité à la capacité de la coupe, entre les sites i et i' , est définie comme :

$$|\mathcal{C} \cap \{((i, k''), (i', k'')) \mid k < k'' \leq k'\}| = \begin{cases} 0, & \text{si } k = k', \\ \sum_{k < k'' \leq k'} \beta_{ii'k''), & \text{sinon.} \end{cases}$$

Or, cette équation correspond à un potentiel de clique linéaire entre une paire de sites voisins $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$.

$$\begin{aligned} |\mathcal{C} \cap \{((i, k''), (i', k'')) \mid k \leq k'' \leq k'\}| &= V_{ii'}(k, k'), \\ &= V_{ii'}(f_i^{\mathcal{C}}, f_{i'}^{\mathcal{C}}). \end{aligned}$$

En sommant sur toutes les paires distinctes de sites voisins, nous obtenons :

$$|\mathcal{C} \cap \mathcal{E}_p| = \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}}} V_{ii'}(f_i^{\mathcal{C}}, f_{i'}^{\mathcal{C}}). \quad (5.7)$$

Par les équations (5.5), (5.6) et (5.7), nous obtenons que le coût de la coupe de coût minimum est :

$$\begin{aligned} |\mathcal{C}| &= \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}}} V_{ii'}(f_i^{\mathcal{C}}, f_{i'}^{\mathcal{C}}) + \sum_{i \in \mathcal{S}} -\log Pr(x_i | f_i^{\mathcal{C}}), \\ &= E(f^{\mathcal{C}} | x). \end{aligned}$$

Donc, la configuration $f^{\mathcal{C}}$ issue de la coupe de coût minimum minimise l'énergie *a posteriori* $E(f | x)$. \square

Chapitre 6

Approximation par graphes de flot

6.1 Une approche approximative

Dans ce chapitre, nous présentons deux méthodes d'optimisation approximative basées sur le calcul du flot maximum et de la coupe de coût minimum dans des graphes. Ces algorithmes permettent, contrairement aux méthodes de Bonneville *et al.* [BRME98], Boykov *et al.*, Greig *et al.* [GPS89], Ishikawa et Geiger [IG98b, IG98a] et Roy et Cox [RC98], de minimiser une fonction d'énergie plus générale de la forme :

$$E(f|x) = \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}}} V_{i,i'}(f_i, f_{i'}) + \sum_{i \in \mathcal{S}} -\log Pr(x_i|f_i), \quad (6.1)$$

où le potentiel de clique $V_{i,i'}(f_i, f_{i'})$ a comme seule restriction d'être positif et d'être défini sur l'ensemble des paires de sites voisins $\mathcal{P}_{\mathcal{N}}$. Bien entendu, ce type de minimisation peut être non-convexe et peut donner lieu à des problèmes combinatoires qui sont NP -complets. À moins d'avoir démontré que $P = NP$, il faut se concentrer sur le développement d'algorithmes efficaces qui sont approximatifs, c'est-à-dire qu'ils ne garantissent pas la convergence vers un minimum global de la fonction d'énergie *a posteriori* de l'équation (6.1), mais plutôt vers une approximation (un minimum local) de celui-ci.

Contrairement aux méthodes de descente [Bes86, BZ87, GG84] généralement utilisées pour minimiser l'équation (6.1), les approches que nous présentons dans ce chapitre permettent, lors d'une itération, de modifier l'étiquette de plusieurs sites à la fois.

6.2 Espace de réétiquetage

Une configuration f engendre une partition de l'ensemble des sites \mathcal{S} en sous-ensembles de sites étiquetés par la même étiquette. Pour une étiquette $k \in \mathcal{L}$, le sous-ensemble des sites étiquetés k est défini comme :

$$\mathcal{S}_k = \{i \in \mathcal{S} \mid f_i = k\}.$$

Les algorithmes, que nous présentons, sont basés sur la constatation que l'étiquetage de l'ensemble des sites \mathcal{S} est équivalent à la partition de \mathcal{S} en sous-ensembles \mathcal{S}_k . Nous pouvons obtenir un nouvel étiquetage de \mathcal{S} en modifiant la partition courante, c'est-à-dire en effectuant des échanges de sites entre les différents sous-ensembles \mathcal{S}_k .

Dans ce qui suit, nous définissons la notion d'échange de sites entre sous-ensembles \mathcal{S}_k en terme d'espaces de réétiquetage et nous montrons qu'il est possible de réduire l'énergie *a posteriori* de l'équation (6.1), en utilisant des techniques particulières de réétiquetage.

D'abord, nous présentons un algorithme récemment publié par Boykov *et al.* [BVZ99] (algorithme de réétiquetage $\alpha\beta$), qui va servir d'introduction à la technique que nous avons développée en parallèle à la leur (algorithme de réétiquetage α). Dans leur article, Boykov *et al.* [BVZ99] présentent un second algorithme (expansion α) qui est limité au potentiel de clique satisfaisant l'inégalité du triangle (très limitatif). De plus, le graphe de flot utilisé pour cet algorithme est très complexe et aucun résultat intéressant n'est fourni (pas de validation).

Algorithme de réétiquetage $\alpha\beta$

1. $f :=$ une configuration initiale
2. Choisir deux étiquettes α et β
 Trouver la configuration f' à partir d'un réétiquetage $\alpha\beta$ de f
 Si $E(f'|x) < E(f|x)$, alors $f = f'$.
3. Répéter 2 pour chaque paires d'étiquettes possibles.
4. Répéter 2 et 3 jusqu'à ce que l'énergie ne décroisse plus.

FIG. 6.1 – Pseudo-code de l'algorithme de réétiquetage $\alpha\beta$. Nous illustrons le fonctionnement de l'algorithme à la figure 6.2.

6.2.1 Réétiquetage $\alpha\beta$

D'une configuration initiale f et de deux étiquettes $\alpha, \beta \in \mathcal{L}$, la configuration f' est issue d'un réétiquetage $\alpha\beta$ (l'algorithme est décrit à la figure 6.1), si

$$\mathcal{S}_k = \mathcal{S}'_k \text{ pour tout } k \neq \alpha, \beta,$$

où \mathcal{S}_k et \mathcal{S}'_k sont les sous-ensembles de sites étiquetés k pour les configurations f et f' respectivement. La différence entre les étiquetages f et f' de \mathcal{S} est que certains sites étiquetés α sont maintenant étiquetés β et vice-versa. Les échanges d'étiquettes s'effectuent uniquement entre les ensembles \mathcal{S}_α et \mathcal{S}_β .

Dans l'article de Boykov *et al.* [BVZ99], il est démontré que la configuration f' d'énergie minimale issue d'un réétiquetage $\alpha\beta$ est obtenue en calculant la coupe de coût minimum dans un graphe de flot particulier.

Le graphe

En se donnant deux étiquettes $\alpha, \beta \in \mathcal{L}$, la structure du graphe

$$\mathcal{G}_{\alpha\beta} = (\mathcal{V}_{\alpha\beta}, \mathcal{E}_{\alpha\beta})$$

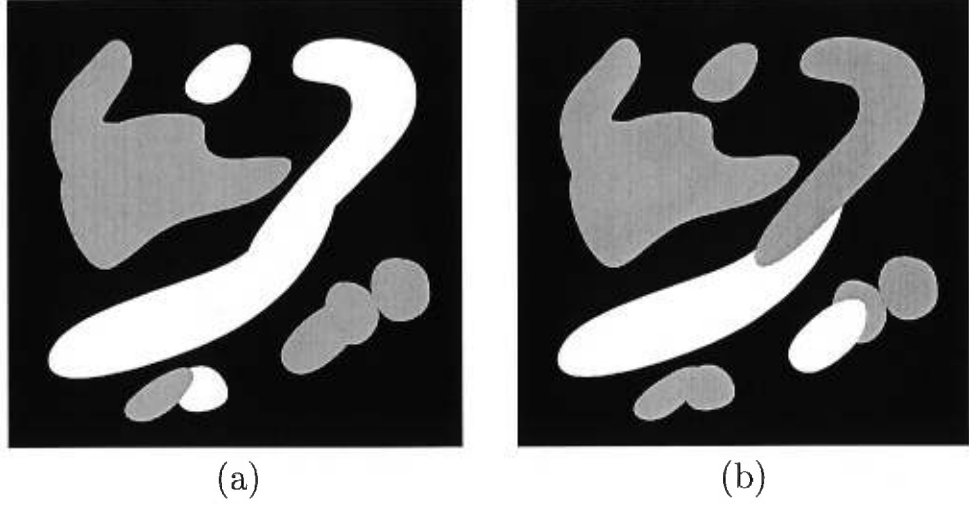


FIG. 6.2 – Illustration du fonctionnement de l'algorithme de réétiquetage $\alpha\beta$, lorsque α est le blanc et β est le gris. (a) la configuration avant le réétiquetage $\alpha\beta$ et (b) la configuration après le réétiquetage $\alpha\beta$.

est déterminée par la partition qu'engendre la configuration f (voir la figure 6.3). Plus spécifiquement, elle est déterminée par les sous-ensembles \mathcal{S}_α et \mathcal{S}_β de \mathcal{S} .

L'ensemble des sommets $\mathcal{V}_{\alpha\beta}$ est constitué de deux noeuds terminaux, la source s et le drain t , en plus de contenir un sommet pour chacun des sites de $\mathcal{S}_\alpha \cup \mathcal{S}_\beta$, c'est-à-dire qu'il y a un noeud pour chacun des sites étiquetés α ou β par la configuration f . L'ensemble des sommets est défini comme :

$$\mathcal{V}_{\alpha\beta} = \mathcal{S}_{\alpha\beta} \cup \{s, t\},$$

où $\mathcal{S}_{\alpha\beta} = \mathcal{S}_\alpha \cup \mathcal{S}_\beta$.

L'ensemble des arêtes du graphe est défini comme :

$$\mathcal{E}_{\alpha\beta} = \mathcal{E}_{v\alpha\beta} \cup \mathcal{E}_{p\alpha\beta},$$

où $\mathcal{E}_{v\alpha\beta}$ et $\mathcal{E}_{p\alpha\beta}$ sont les ensembles d'arêtes de vraisemblance et de pénalité respectivement.

Les sommets de $\mathcal{S}_{\alpha\beta}$ sont connectés entre eux par des arêtes de pénalité. Il y a une connexion entre les sommets $i, i' \in \mathcal{S}_{\alpha\beta}$ seulement si le site $i' \in \mathcal{S}$ est voisin

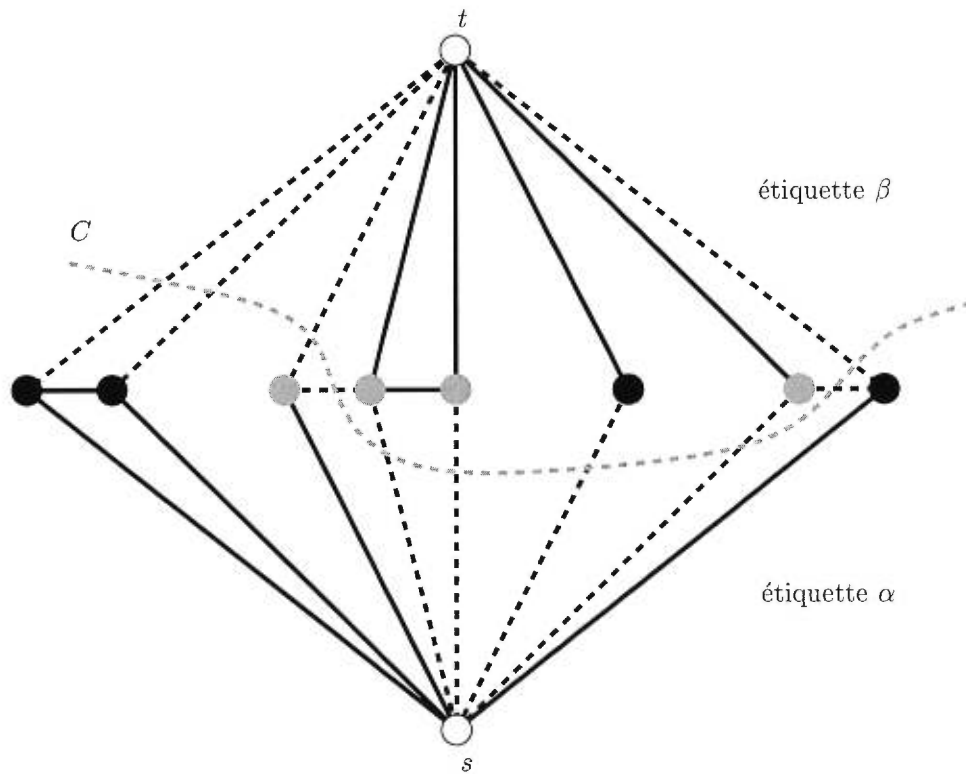


FIG. 6.3 – Aspect du graphe de l’algorithme de réétiquetage $\alpha\beta$. Les noeuds gris sont dans \mathcal{S}_α et les noeuds noirs sont dans \mathcal{S}_β .

du site i par rapport au système de voisinage \mathcal{N} utilisé. La capacité des arêtes de pénalité est :

$$c(i, i') = V_{ii'}(f_i, f_{i'}),$$

où les sites $i, i' \in \mathcal{S}_{\alpha\beta}$ et $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$.

Les chemins de la source s vers le drain t sont engendrés par les arêtes de vraisemblance contenues dans l'ensemble $\mathcal{E}_{v\alpha\beta}$. Pour tout les sites $i \in \mathcal{S}_{\alpha\beta}$, il y a une arête de la source vers le noeud i , et de ce noeud vers le drain, qui sont respectivement (s, i) , dont la capacité est

$$c(s, i) = -\log Pr(x_i | f_i = \alpha) + \sum_{i' \in \mathcal{N}_i, i' \notin \mathcal{S}_{\alpha\beta}} V_{i,i'}(\alpha, f_{i'})$$

et (i, t) , qui a plutôt la capacité suivante :

$$c(i, t) = -\log Pr(x_i | f_i = \beta) + \sum_{i' \in \mathcal{N}_i, i' \notin \mathcal{S}_{\alpha\beta}} V_{i,i'}(\beta, f_{i'}).$$

La coupe

Soit \mathcal{C} une coupe du graphe $\mathcal{G}_{\alpha\beta}$. Nous définissons la configuration $f^{\mathcal{C}}$ engendrée par cette coupe comme :

$$f_i^{\mathcal{C}} = \begin{cases} \alpha, & \text{si } i \in \mathcal{S}_{\alpha\beta} \text{ et } (s, i) \in \mathcal{C}, \\ \beta, & \text{si } i \in \mathcal{S}_{\alpha\beta} \text{ et } (i, t) \in \mathcal{C}, \\ f_i, & \text{si } i \notin \mathcal{S}_{\alpha\beta}. \end{cases}$$

Dans leurs travaux, Boykov *et al.* [BVZ99] ont démontré que la coupe de coût minimum \mathcal{C} du graphe $\mathcal{G}_{\alpha\beta}$ permet d'obtenir le réétiquetage $\alpha\beta$ de la configuration f qui minimise le plus l'énergie *a posteriori*. En effet, ils ont démontré que

$$|\mathcal{C}| = E(f^{\mathcal{C}} | x) - K,$$

où la constante K est la même pour toutes les coupes du graphe $\mathcal{G}_{\alpha\beta}$. De plus,

$$K = \sum_{\substack{\{i, i'\} \in \mathcal{P}_{\mathcal{N}} \\ i, i' \notin \mathcal{S}_{\alpha\beta}}} V_{ii'}(f_i, f_{i'}) + \sum_{i \notin \mathcal{S}_{\alpha\beta}} -\log Pr(x_i | f_i).$$

6.2.2 Réétiquetage α

Dans cette section, nous présentons un algorithme (l'algorithme est décrit à la figure 6.4) qui ressemble à l'algorithme de réétiquetage $\alpha\beta$, mais qui permet la modification des étiquettes d'un plus grand nombre de site à la fois. Plutôt que restreindre le réétiquetage à deux sous-ensembles de sites (\mathcal{S}_α et \mathcal{S}_β), nous permettons à tous les sites de \mathcal{S} d'échanger leur étiquette $l \in \mathcal{L} - \{\alpha\}$ pour α , à l'exception des sites déjà étiquetés α , et en respectant un ordre établi par une permutation de $\mathcal{L} - \{\alpha\}$.

Sélectionnons une étiquette $\alpha \in \mathcal{L}$. La permutation ρ de l'ensemble $\mathcal{L} - \{\alpha\}$ est identifiée par :

$$\rho = (l_0^p, \dots, l_{M-2}^p),$$

où $l_i^p \in \mathcal{L} - \{\alpha\}$ est la i ème étiquette permutée. Nous permettons à un site étiqueté l_i^p d'échanger son étiquette courante, seulement s'il n'est pas voisin d'un site étiqueté $l_{i'}^p$ ayant, dans l'ordre engendré par la permutation ρ ($l_i^p < l_{i'}^p$), la permission de changer pour l'étiquette α . Nous appelons ce type d'échange d'étiquettes un réétiquetage α . Ainsi, d'une configuration initiale f , d'une étiquette α et d'une permutation ρ , la configuration f' est issue d'un réétiquetage α , si $\mathcal{S}_\alpha \subset \mathcal{S}'_\alpha$ et $\mathcal{S}'_l \subset \mathcal{S}_l$ pour tout $l \neq \alpha$.

Dans ce qui suit, nous allons démontrer que le réétiquetage α , qui permet de minimiser l'équation (6.1), est obtenu en calculant la coupe de coût minimum dans un graphe de flot particulier.

Le graphe

En se donnant une étiquette α et une permutation ρ , la structure du graphe

$$\mathcal{G}_\alpha = (\mathcal{V}_\alpha, \mathcal{E}_\alpha)$$

est déterminée par la partition de l'ensemble des sites \mathcal{S} , qu'engendre l'étiquetage f (voir la figure 6.6).

Algorithme de réétiquetage α

1. $f :=$ une configuration initiale.
2. Choisir une étiquette α .
Générer une permutation ρ de $\mathcal{L} - \{\alpha\}$.
Trouver la configuration f' à partir d'un réétiquetage α de f .
Si $E(f'|x) < E(f|x)$, alors $f = f'$.
3. Répéter 2 pour chaque étiquettes $\alpha \in \mathcal{L}$ possibles.
4. Répéter 2 et 3 jusqu'à ce que l'énergie ne décroisse plus.

FIG. 6.4 – Pseudo-code de l'algorithme de réétiquetage α . Il y a plusieurs façons de générer aléatoirement une permutation. Nous utilisons la distribution (histogramme) des étiquettes. Nous illustrons le fonctionnement de l'algorithme à la figure 6.5.

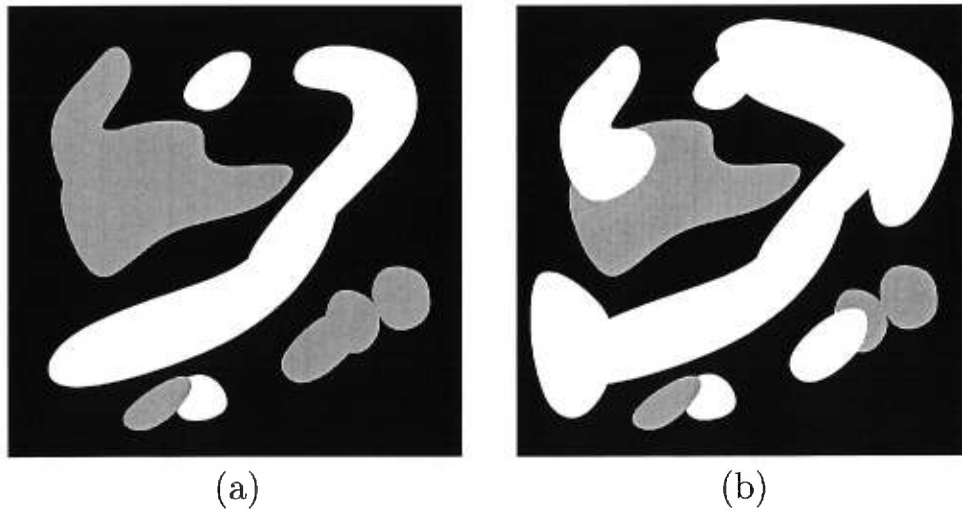


FIG. 6.5 – Illustration du fonctionnement de l'algorithme de réétiquetage α (lorsque α est le blanc). (a) la configuration avant le réétiquetage α et (b) la configuration après le réétiquetage α .

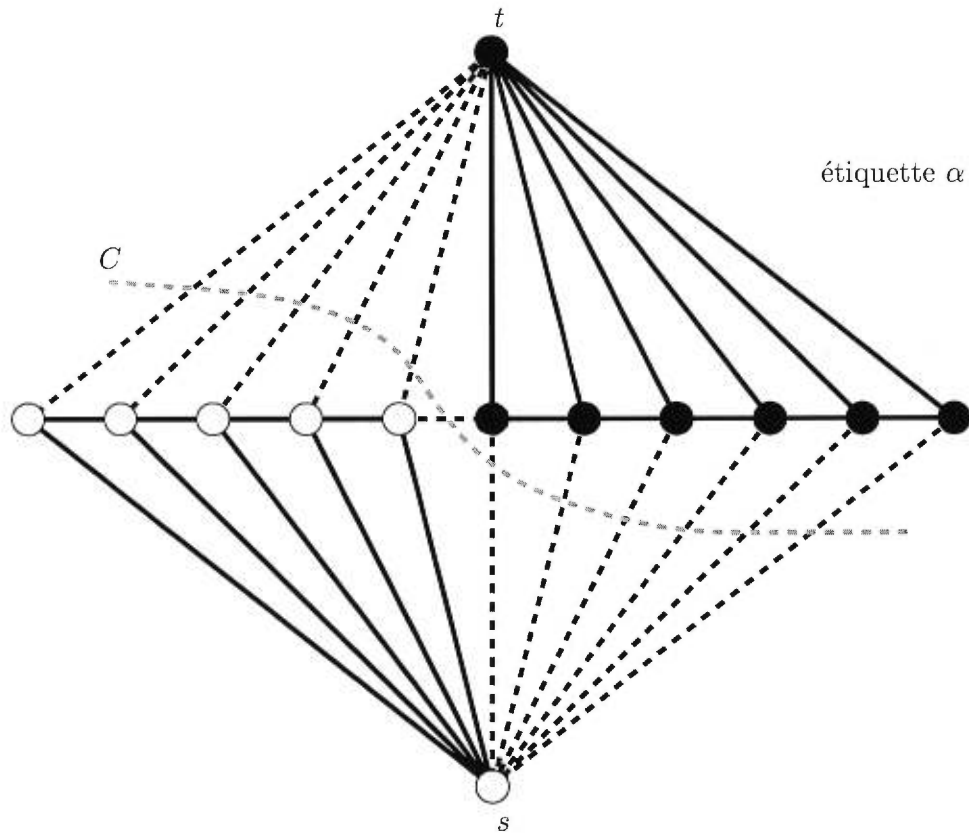


FIG. 6.6 – Aspect du graphe utilisé dans l’algorithme de réétiquetage α .

L'ensemble des sommets \mathcal{V}_α est constitué de deux noeuds spéciaux, une source s et un drain t , en plus de contenir un sommet pour chacun des sites de \mathcal{S} . Ainsi, nous définissons l'ensemble des sommets comme :

$$\mathcal{V}_\alpha = \mathcal{S} \cup \{s, t\}.$$

L'ensemble des arêtes du graphe se subdivise en un ensemble d'arêtes de vraisemblance $\mathcal{E}_{v\alpha}$ et un ensemble d'arêtes de pénalité $\mathcal{E}_{p\alpha}$. Ainsi, nous avons :

$$\mathcal{E}_\alpha = \mathcal{E}_{v\alpha} \cup \mathcal{E}_{p\alpha},$$

L'ensemble $\mathcal{E}_{v\alpha}$ génère des chemins de la source vers le drain. Il contient des arêtes de la forme : (s, i) ou (i, t) , pour tout $i \in \mathcal{S}$. L'ensemble des arêtes de pénalité connecte les sommets voisins $i, i' \in \mathcal{S}$ du graphe entre eux, seulement si la clique $\{i, i'\} \in \mathcal{P}_\mathcal{N}$.

Comment construire la fonction de capacité

Avant de définir la fonction de capacité du graphe, nous introduisons une variable d'état τ_i qui permet de déterminer si, à l'itération courante n , le site i peut troquer son étiquette $f_i^{(n)}$ (configuration à l'itération n) pour une étiquette α . Ainsi, nous posons :

$$\tau_i = \begin{cases} 0, & \text{s'il existe } i' \in \mathcal{N}_i \text{ tel que } \tau_{i'} = 1 \text{ et } f_{i'} < f_i \text{ dans } \rho, \\ 1, & \text{sinon.} \end{cases}$$

La capacité des arêtes de vraisemblance sortant de la source est définie com-

me :

$$c(s, i) = \begin{cases} -\log Pr(x_i | f_i = f_i^{(n)}) + \sum_{\substack{i' \in \mathcal{N}_i \\ i' \in \mathcal{S}_{\bar{\alpha}} \\ \tau_{i'} = 0}} \frac{V_{i,i'}(f_i, f_{i'})}{2}, & \text{si } i \in \mathcal{S}_{\bar{\alpha}} \text{ et } \tau_i = 0, \\ -\log Pr(x_i | f_i = f_i^{(n)}) + \sum_{\substack{i' \in \mathcal{N}_i \\ i' \in \mathcal{S}_{\bar{\alpha}} \\ \tau_{i'} = 0}} V_{i,i'}(f_i, f_{i'}), & \text{si } i \in \mathcal{S}_{\bar{\alpha}} \text{ et } \tau_i = 1, \\ \infty, & \text{si } i \in \mathcal{S}_{\alpha}, \end{cases}$$

où $\mathcal{S}_{\bar{\alpha}} = \mathcal{S} - \mathcal{S}_{\alpha}$.

Les arêtes allant des noeuds de \mathcal{S} vers le drain ont comme capacité :

$$c(i, t) = \begin{cases} -\log Pr(x_i | f_i = \alpha), & \text{si } i \in \mathcal{S}_{\alpha}, \\ -\log Pr(x_i | f_i = \alpha), & \text{si } i \in \mathcal{S}_{\bar{\alpha}} \text{ et } \tau_i = 1, \\ \infty, & \text{si } i \in \mathcal{S}_{\bar{\alpha}} \text{ et } \tau_i = 0. \end{cases}$$

Finalement, la capacité des arêtes de pénalité est définie comme :

$$c(i, i') = \begin{cases} V_{i,i'}(\alpha, f_{i'}), & \text{si } \tau_i = \tau_{i'} = 1, \\ V_{i,i'}(\alpha, f_{i'}), & \text{si } \tau_i = 1, \tau_{i'} = 0, i' \in \mathcal{S}_{\bar{\alpha}}, \\ V_{i,i'}(\alpha, f_{i'}), & \text{si } \tau_i = 0, i \in \mathcal{S}_{\alpha} \text{ et } i' \in \mathcal{S}_{\bar{\alpha}}, \\ 0, & \text{si } \tau_i = \tau_{i'} = 0 \text{ et } i, i' \in \mathcal{S}_{\bar{\alpha}} \text{ ou } i, i' \in \mathcal{S}_{\alpha}, \\ c(i', i), & \text{sinon,} \end{cases}$$

où $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$.

Les arêtes de vraisemblance qui ont une capacité infinie ne font jamais partie de la coupe de coût minimum, car elles ne peuvent être saturées par une quantité finie de flot. En pratique nous pouvons remplacer les capacités infinies par une capacité de valeur constante. Nous n'en ferons pas la démonstration, car un résultat semblable a été démontré au chapitre précédent. Dans le cas où $c(s, i) = \infty$, nous pouvons remplacer cette capacité, par une constante excédant la somme des capacités des arêtes sortant du noeud i :

$$K_{si} = c(i, t) + \sum_{i' \in \mathcal{N}_i} c(i', i) + \epsilon, \text{ pour } \epsilon > 0$$

et lorsque $c(i, t) = \infty$, la capacité peut être remplacée par un nombre plus grand que la somme des capacités des arêtes incidentes au noeud i :

$$K_{it} = c(s, i) + \sum_{i' \in \mathcal{N}_i} c(i, i') + \epsilon, \text{ pour } \epsilon > 0.$$

La coupe

Théorème 1 *Soit la coupe \mathcal{C} de coût minimum associée au flot maximum dans le graphe $\mathcal{G}_\alpha = (\mathcal{V}_\alpha, \mathcal{E}_\alpha)$. Elle contient une et une seule arête de vraisemblance : soit (s, i) ou soit (i, t) , pour chaque site $i \in \mathcal{S}$.*

Preuve. Premièrement, la coupe \mathcal{C} contient au moins une arête de vraisemblance (s, i) ou (i, t) , car il existe un chemin de la source s au drain t pour chaque site de \mathcal{S} de la forme,

$$s \rightarrow i \rightarrow t.$$

Il reste à démontrer qu'il y a une unique arête de vraisemblance pour chaque site $i \in \mathcal{S}$. C'est-à-dire,

$$\text{si } (s, i) \in \mathcal{C} \text{ alors } (i, t) \notin \mathcal{C}$$

ou

$$\text{si } (i, t) \in \mathcal{C} \text{ alors } (s, i) \notin \mathcal{C}.$$

Par contradiction, supposons que $(s, i), (i, t) \in \mathcal{C}$. Nous avons que la coupe \mathcal{C} engendre une partition de l'ensemble des sommets $\mathcal{V}_\alpha = \mathcal{T} \cup \overline{\mathcal{T}}$, tel que $\mathcal{T} \cap \overline{\mathcal{T}} = \emptyset$, $s \in \mathcal{T}$ et $t \in \overline{\mathcal{T}}$.

Puisque $(s, i) \in \mathcal{C}$ et que $s \in \mathcal{T}$, nous avons que $i \in \overline{\mathcal{T}}$. De plus, nous avons que $(i, t) \in \mathcal{C}$ et $t \in \overline{\mathcal{T}}$, donc $i \in \mathcal{T}$. Ces deux affirmations impliquent que $i \in \mathcal{T} \cap \overline{\mathcal{T}}$. Mais ceci contredit l'hypothèse $\mathcal{T} \cap \overline{\mathcal{T}} = \emptyset$. Donc, il y a une et une seule arête de vraisemblance contenue dans la coupe \mathcal{C} pour chaque site $i \in \mathcal{S}$. \square

Puisqu'il y a une seule arête de vraisemblance faisant partie de la coupe \mathcal{C} pour chacun des sites de \mathcal{S} , nous définissons l'étiquetage $f^{\mathcal{C}}$ engendré par la coupe \mathcal{C} comme :

$$f_i^{\mathcal{C}} = \begin{cases} f_i^{(n)}, & \text{si } (s, i) \in \mathcal{C}, \\ \alpha, & \text{si } (i, t) \in \mathcal{C}. \end{cases} \quad (6.2)$$

Il reste à démontrer que la capacité d'une coupe de \mathcal{G}_α correspond à l'énergie de l'étiquetage résultant $f^{\mathcal{C}}$ de \mathcal{S} , c'est-à-dire que

$$|\mathcal{C}| = E(f^{\mathcal{C}}|x).$$

Corollaire 1 *Si \mathcal{C} est une coupe du graphe \mathcal{G}_α et $f^{\mathcal{C}}$ est l'étiquetage correspondant, alors $|\mathcal{C}| = E(f^{\mathcal{C}}|x)$.*

Preuve. Afin de simplifier ce qui suit, nous allons définir quatre ensembles, qui correspondent aux ensembles d'arêtes de pénalité associées aux types de discontinuités illustrées à la figure 6.7 (a), (b), (c) et (d) respectivement. Ainsi,

$$\begin{aligned} \mathcal{P}_{\mathcal{N}_1} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 1, \tau_{i'} = 1\}, \\ \mathcal{P}_{\mathcal{N}_2} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 0, \tau_{i'} = 1, i \in \mathcal{S}_\alpha\}, \\ \mathcal{P}_{\mathcal{N}_3} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 0, \tau_{i'} = 0, i \in \mathcal{S}_\alpha, i' \in \mathcal{S}_{\bar{\alpha}}\}, \\ \mathcal{P}_{\mathcal{N}_4} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 1, \tau_{i'} = 0, i' \in \mathcal{S}_{\bar{\alpha}}, f_i^{\mathcal{C}} = \alpha\}. \end{aligned}$$

Les discontinuités dans l'étiquetage $f^{\mathcal{C}}$ qui sont illustrées à la figure 6.7 (e) et (f) correspondent aux ensembles suivant :

$$\begin{aligned} \mathcal{P}_{\mathcal{N}_5} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 0, \tau_{i'} = 0, i, i' \in \mathcal{S}_{\bar{\alpha}}\}, \\ \mathcal{P}_{\mathcal{N}_6} &= \{\{i, i'\} \in \mathcal{N} \mid \tau_i = 1, \tau_{i'} = 0, i, i' \in \mathcal{S}_{\bar{\alpha}}, f_i^{\mathcal{C}} \neq \alpha\}. \end{aligned}$$

Les $\mathcal{P}_{\mathcal{N}_i}$ sont disjoints deux à deux et nous avons :

$$\mathcal{P}_{\mathcal{N}} = \mathcal{P}_{\mathcal{N}_1} \cup \mathcal{P}_{\mathcal{N}_2} \cup \mathcal{P}_{\mathcal{N}_3} \cup \mathcal{P}_{\mathcal{N}_4} \cup \mathcal{P}_{\mathcal{N}_5} \cup \mathcal{P}_{\mathcal{N}_6} \cup \mathcal{P}_{\mathcal{N}'},$$

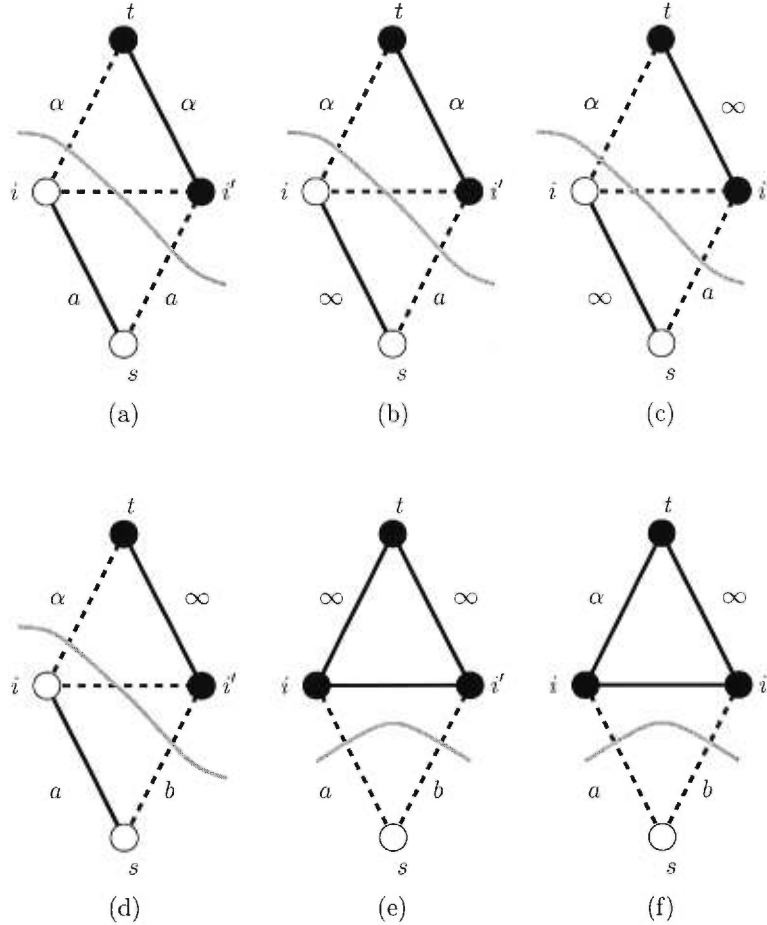


FIG. 6.7 – Les arêtes connectées au drain t correspondent aux arêtes de vraisemblance associant l'étiquette α , tandis que les arêtes connectées à la source s sont associées à l'étiquette de l'itération précédente (par exemple aux étiquettes a et b). Les coupes entraînant des discontinuités dans l'étiquetage f^c sont de la forme : (a),(b),(c),(d), (e) et (f), où les sites i et i' sont voisins, c'est-à-dire que $\{i, i'\} \in \mathcal{P}_N$. Pour (e) et (f), les coûts associés aux discontinuités sont incorporés dans la capacité des arêtes de vraisemblance (s, i) et (s, i') , qui associent les étiquettes a et b aux sites i et i' respectivement.

où $\mathcal{P}_{\mathcal{N}'}$ correspond à l'ensemble des sites voisins n'ayant pas de discontinuités dans l'étiquetage $f^{\mathcal{C}}$,

$$\mathcal{P}_{\mathcal{N}'} = \{\{i, i'\} \in \mathcal{P}_{\mathcal{N}} \mid f_i^{\mathcal{C}} = f_{i'}^{\mathcal{C}}\}.$$

Pour démontrer ce corollaire, nous devons montrer que la capacité d'une coupe $|\mathcal{C}|$ correspond à la valeur de l'énergie $E(f^{\mathcal{C}}|x)$. Décomposons la capacité de la coupe en capacités induites par les arêtes de vraisemblance et par les arêtes de pénalité,

$$|\mathcal{C}| = |\mathcal{C} \cap \mathcal{E}_{v\alpha}| + |\mathcal{C} \cap \mathcal{E}_{p\alpha}|.$$

Puisque

$$\sum_{\substack{i \in S_{\bar{\alpha}} \\ \tau_i = 0}} \sum_{\substack{i' \in \mathcal{N}_i \\ \tau_{i'} = 0 \\ i' \in S_{\bar{\alpha}}}} \frac{V_{i,i'}(f_i, f_{i'})}{2} = \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_5}} V_{i,i'}(f_i, f_{i'}),$$

nous avons que la capacité de la coupe induite par les arêtes de vraisemblance est

$$\begin{aligned} |\mathcal{C} \cap \mathcal{E}_{v\alpha}| &= \sum_{i \in S_{\bar{\alpha}}} H(f_i^{\mathcal{C}}|x) + \\ &\quad \left(\sum_{\substack{i \in S_{\bar{\alpha}}, \tau_i = 1 \\ f_i^{\mathcal{C}} \neq \alpha}} H(f_i^{\mathcal{C}}|x) + \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_5}} V_{i,i'}(f_i, f_{i'}) \right) + \\ &\quad \left(\sum_{i \in S_{\bar{\alpha}}, \tau_i = 0} H(f_i^{\mathcal{C}}|x) + \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_6}} V_{i,i'}(f_i, f_{i'}) \right) + \\ &\quad \sum_{\substack{i \in S_{\bar{\alpha}}, \tau_i = 1 \\ f_i^{\mathcal{C}} = \alpha}} H(f_i^{\mathcal{C}}|x) \\ &= \sum_{i \in S} H(f_i^{\mathcal{C}}|x) + \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_5} \cup \mathcal{P}_{\mathcal{N}_6}} V_{i,i'}(f_i, f_{i'}). \end{aligned}$$

Maintenant, il reste à dériver la capacité de la coupe \mathcal{C} engendrée par les arêtes

de pénalité :

$$\begin{aligned}
|\mathcal{C} \cup \mathcal{E}_{p\alpha}| &= \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_1}} V_{i,i'}(f_i, f_{i'}) + \\
&\sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_2}} V_{i,i'}(f_i, f_{i'}) + \\
&\sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_3}} V_{i,i'}(f_i, f_{i'}) + \\
&\sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_4}} V_{i,i'}(f_i, f_{i'})
\end{aligned}$$

Donc,

$$\begin{aligned}
|\mathcal{C}| &= \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}_1} \cup \mathcal{P}_{\mathcal{N}_2} \cup \mathcal{P}_{\mathcal{N}_3} \cup \mathcal{P}_{\mathcal{N}_4} \cup \mathcal{P}_{\mathcal{N}_5} \cup \mathcal{P}_{\mathcal{N}_6}} V_{i,i'}(f_i^{\mathcal{C}}, f_{i'}^{\mathcal{C}}) + \sum_{i \in \mathcal{S}} H(x|f^{\mathcal{C}}) \\
&= \sum_{\{i,i'\} \in \mathcal{P}_{\mathcal{N}}} V_{i,i'}(f_i^{\mathcal{C}}, f_{i'}^{\mathcal{C}}) + \sum_{i \in \mathcal{S}} H(x|f^{\mathcal{C}}) \\
&= E(f^{\mathcal{C}}|x),
\end{aligned}$$

ce qui démontre que la capacité d'une coupe \mathcal{C} du graphe G_α engendre un étiquetage ayant pour énergie $E(f^{\mathcal{C}}|x)$. \square

Par le corollaire 1, nous avons que la coupe de coût minimum permet de minimiser l'énergie *a posteriori* de l'équation (6.1).

Chapitre 7

Résultats

Le but de cette section est de comparer la performance des différents algorithmes présentés et développés. Nous n'effectuons pas de comparaisons des algorithmes en terme de distance (erreur) entre l'image restaurée et l'image originale, car un mauvais minimum local (haute énergie) peut avoir une distance plus petite que le minimum global, si la distribution *a priori* ne reflète pas correctement les propriétés locales de l'image originale. Plutôt, nous évaluons les divers algorithmes en comparant directement la qualité du minimum local, de la fonction d'énergie *a posteriori*, qu'ils trouvent.

Dans un premier temps, nous évaluons la performance des divers algorithmes pour la restauration d'images synthétiques. Puis, nous présentons les résultats d'expériences sur des images réelles en vision stéréoscopique.

Il est à noter que dans les expériences qui suivent, nous ne justifions pas le choix des distributions *a priori* utilisées, car la sélection et la conception de distributions *a priori* qui décrivent bien l'image originale est un problème distinct, qui dépasse le cadre de ce mémoire.

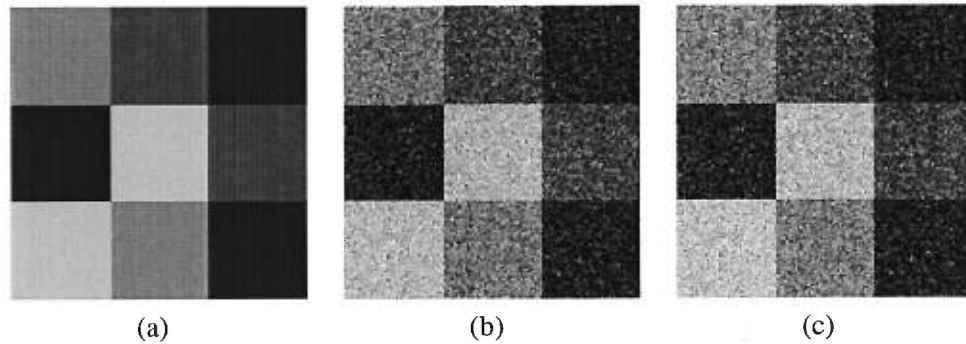


FIG. 7.1 – (a) Image originale. (b) Image bruitée avec un bruit gaussien de moyenne 0 et d'écart-type 25. (c) Restauration sans contexte (maximum de vraisemblance).

7.1 Expérience sur une surface constante par morceaux

L'image originale est un damier de 128×128 pixels qui est illustrée à la figure 7.1 (a). Cette image est une représentation en niveaux de gris d'une surface constante par morceaux. Les niveaux de gris dans cette image sont ordonnés, car ils représentent des élévations. Les différents plateaux du damier sont à 40, 80, 120 ou 180. À la figure 7.1 (b), nous observons l'image dégradée par un bruit gaussien additif de moyenne $\mu = 0$ et d'écart-type $\sigma = 25$. La force de ce bruit est raisonnable compte tenue qu'une discontinuité est au minimum de 40, ce qui est inférieure à 2σ .

Toutes les restaurations de l'image bruitée du damier utilisent un ensemble \mathcal{L} contenant 25 étiquettes uniformément distribuées entre 0 et 250. Le système de voisinage \mathcal{N} utilisé est un système de 8-voisins.

À la figure 7.1 (c), nous voyons le résultat, qui ne tient pas compte du contexte, d'une restauration par un maximum de vraisemblance. L'image restaurée est toujours bruitée, ce qui démontre la nécessité d'introduire de l'information sur la cohérence spatiale de l'image originale sous forme de distribution *a priori*.

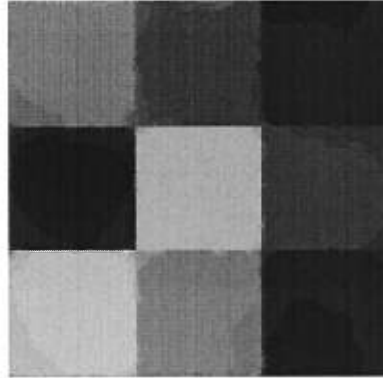


FIG. 7.2 – Résultat obtenu par le recuit simulé de Geman et Geman [GG84] après 10000 itérations de l’algorithme (1289 minutes). Une décroissance logarithmique de la température a été utilisée, de même qu’un potentiel de clique linéaire avec $\beta = 0.5$. L’énergie initiale est de 1541118 et l’énergie finale est de 157427.

7.1.1 Utilisation d’un potentiel de clique linéaire

Le potentiel de clique linéaire utilisé pour restaurer l’image bruitée du damier est :

$$V_{ii'}(f_i, f_{i'}) = \beta |f_i - f_{i'}|,$$

où $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$.

Résultats du recuit simulé

Le recuit simulé de Geman et Geman [GG84] est exécuté avec la décroissance logarithmique de température suivante :

$$T^{(n)} = \frac{3.0}{\log(n+1)}.$$

En théorie, si la décroissance de température est suffisamment lente, le recuit simulé permet d’obtenir l’optimum global de la fonction d’énergie *a posteriori*. Cette particularité fait du recuit un algorithme terriblement inefficace. Pour cette raison, nous l’avons exécuté uniquement pour $\beta = 0.5$ (voir la figure 7.2). En 1289

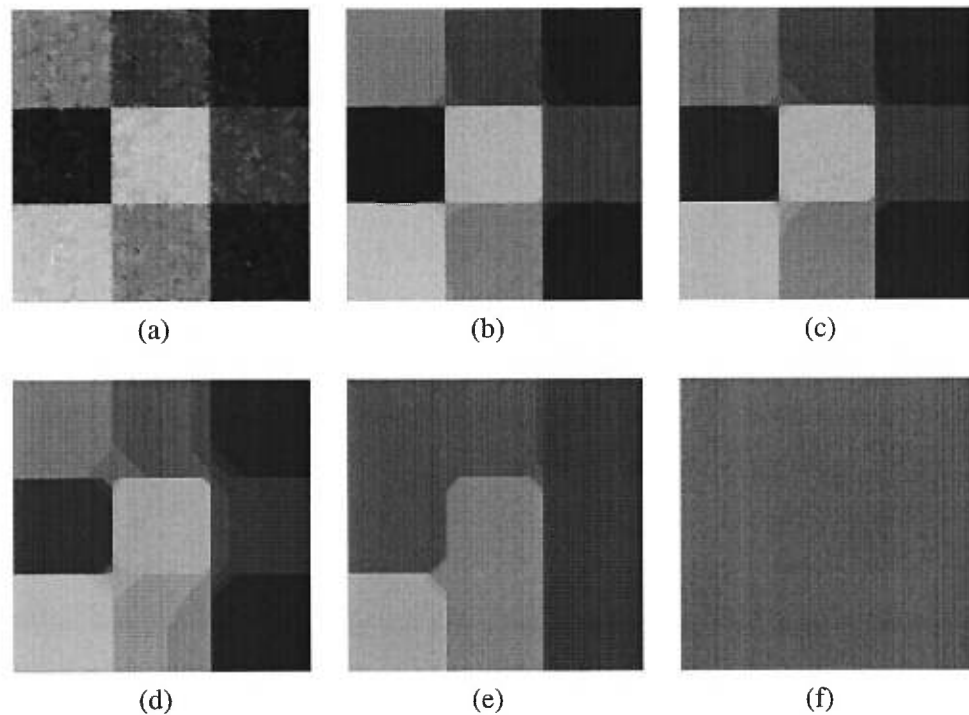


FIG. 7.3 – Résultats du MAPE avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 114654, (b) 154137, (c) 216679, (d) 309969, (e) 336809 et (f) 102371.

minutes, le recuit a trouvé une configuration ayant une énergie de 157427, ce qui est très près de la solution exacte obtenue par la méthode de MAPE (voir la figure 7.3 (b)).

Résultats du MAPE

Lorsque le potentiel de clique est linéaire, la méthode de MAPE permet d'obtenir la configuration qui minimise globalement la fonction d'énergie postérieure. Ainsi, nous pouvons évaluer la qualité des minima locaux obtenus par les autres algorithmes. Les résultats obtenus par l'algorithme de MAPE sont illustrés à la

figure 7.3 pour différentes valeurs du paramètre β .

L'optimum trouvé par la méthode de MAPE est sujet à des erreurs numériques. Elles sont introduites par l'utilisation de capacités entières dans le graphe de flot. Pour mettre en relief cette remarque, nous observons que la configuration trouvée par l'algorithme $\alpha\beta$ de la figure 7.6 (b) a une énergie plus basse (152636) que la configuration trouvée par la méthode de MAPE (figure 7.3 (b) avec une énergie de 154137).

Comme illustré à la figure 7.7, la méthode de MAPE offre un temps d'exécution comparable à celui de l'IMC. Les temps sont inférieurs aux temps du recuit simulé, du réétiquetage α et du réétiquetage $\alpha\beta$. De plus, la méthode de MAPE est non-itérative (voir figure 7.8). Donc, contrairement aux autres algorithmes, aucun critère d'arrêt ne doit être spécifié.

Résultats de l'IMC

Les résultats obtenus par l'IMC sont présentés à la figure 7.4. Nous remarquons que la performance de l'IMC se dégrade avec l'augmentation de la force du contexte (β). De plus, l'algorithme tend rapidement vers une solution correspondant à un maximum de votes dans le voisinage utilisé.

Pour toutes les valeurs du paramètre β utilisées, l'algorithme d'IMC ne réussit pas à atteindre un niveau d'énergie aussi bas que les autres algorithmes. Par contre, la décroissance de l'énergie dans le temps est beaucoup plus rapide que les méthodes de réétiquetage α et de réétiquetage $\alpha\beta$ (voir la figure 7.7). À la figure 7.8, nous remarquons que la plus grande décroissance d'énergie se produit dans les premières itérations de l'algorithme. Un résultat semblable est aussi observé pour les autres algorithmes.

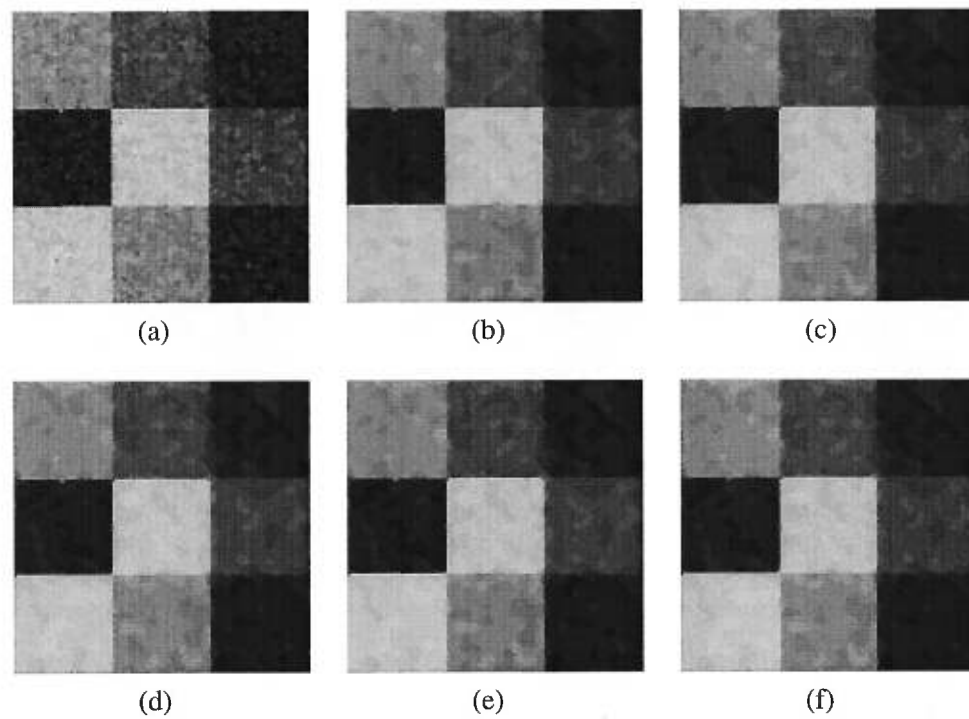


FIG. 7.4 – Résultats de l'IMC avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 139935, (b) 214779, (c) 359972, (d) 653899, (e) 1597630 et (f) 29342090.

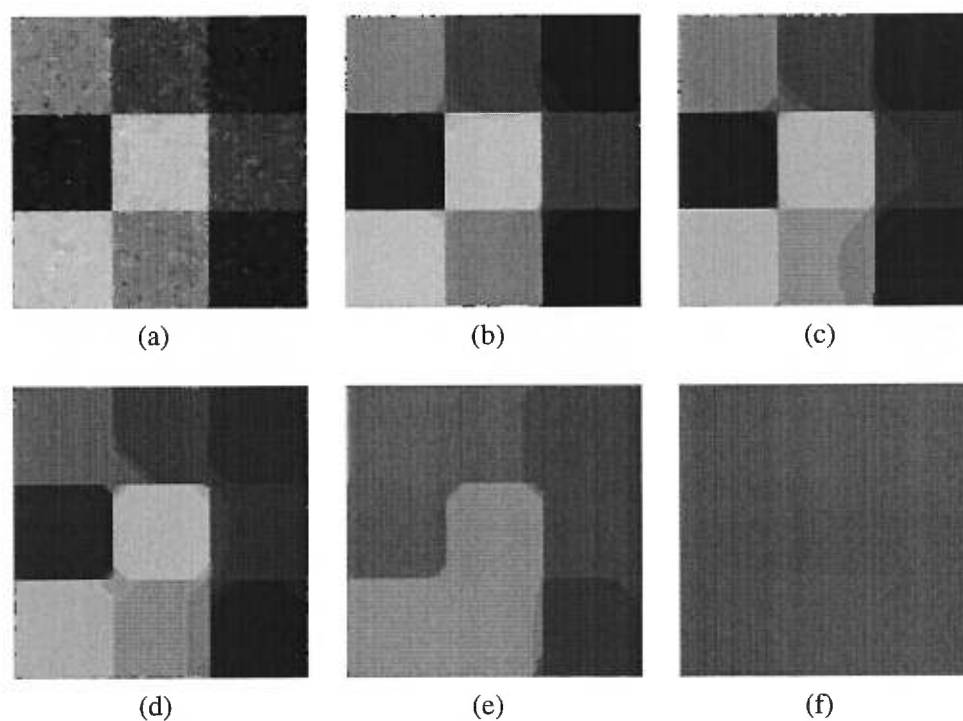


FIG. 7.5 – Résultats du réétiquetage α avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 111975, (b) 158153, (c) 218658, (d) 322866, (e) 280621 et (f) 102371.

Résultats du réétiquetage α

Les résultats obtenus par l'algorithme de réétiquetage α (voir la figure 7.5) sont comparables aux résultats du MAPE (figure 7.3) et du réétiquetage $\alpha\beta$ (figure 7.6).

À la figure 7.7, nous remarquons que la convergence de l'algorithme de réétiquetage α prend un peu moins de temps que l'algorithme de réétiquetage $\alpha\beta$. Toutefois, après quelques itérations (4 à 7 itérations), l'algorithme ne réduit presque plus l'énergie. Notez qu'une itération de l'algorithme de réétiquetage α prend une fraction du temps d'une itération de l'algorithme de réétiquetage $\alpha\beta$ (voir la figure 7.7). Ceci était prévisible, car chaque itération, de l'algorithme de réétiquetage α , est linéaire par rapport au nombre d'étiquette (nous considérons un cycle comme ayant un temps constant), tandis qu'une itération de l'algorithme de réétiquetage $\alpha\beta$ est quadratique.

Résultats du réétiquetage $\alpha\beta$

À la figure 7.6, nous présentons les résultats obtenus par l'algorithme de réétiquetage $\alpha\beta$. La plus grande diminution de l'énergie se produit lors de la première itération (voir la figure 7.8). Moins le lissage est important (β faible), plus rapide est la décroissance de l'énergie (figure 7.7) et moins le nombre d'itérations requis pour la convergence est grand (voir la figure 7.8).

Tout comme l'algorithme de réétiquetage α , un meilleur critère d'arrêt pourrait améliorer substantiellement le temps d'exécution. Toutefois, chaque itération est quadratique par rapport au nombre d'étiquettes.

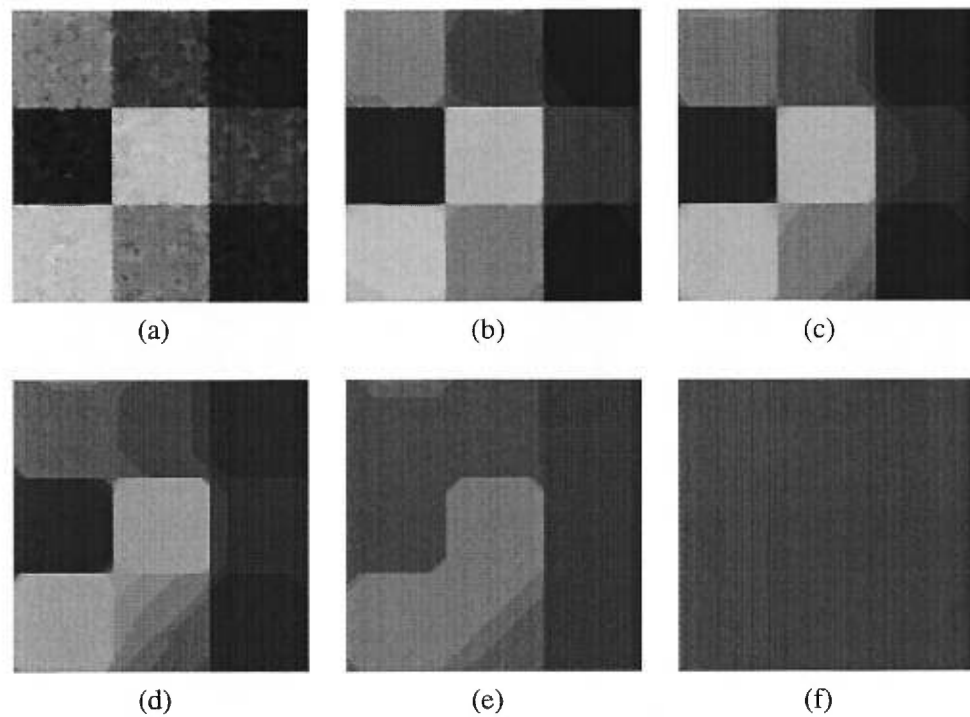


FIG. 7.6 – Résultats du réétiquetage $\alpha\beta$ avec potentiel de clique linéaire et (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100$. L'énergie initiale : (a) 357572, (b) 1541118, (c) 3020535, (d) 5979419, (e) 14855867 et (f) 295946688. L'énergie finale est de : (a) 111574, (b) 152636, (c) 217687, (d) 284025, (e) 276987 et (f) 103125.

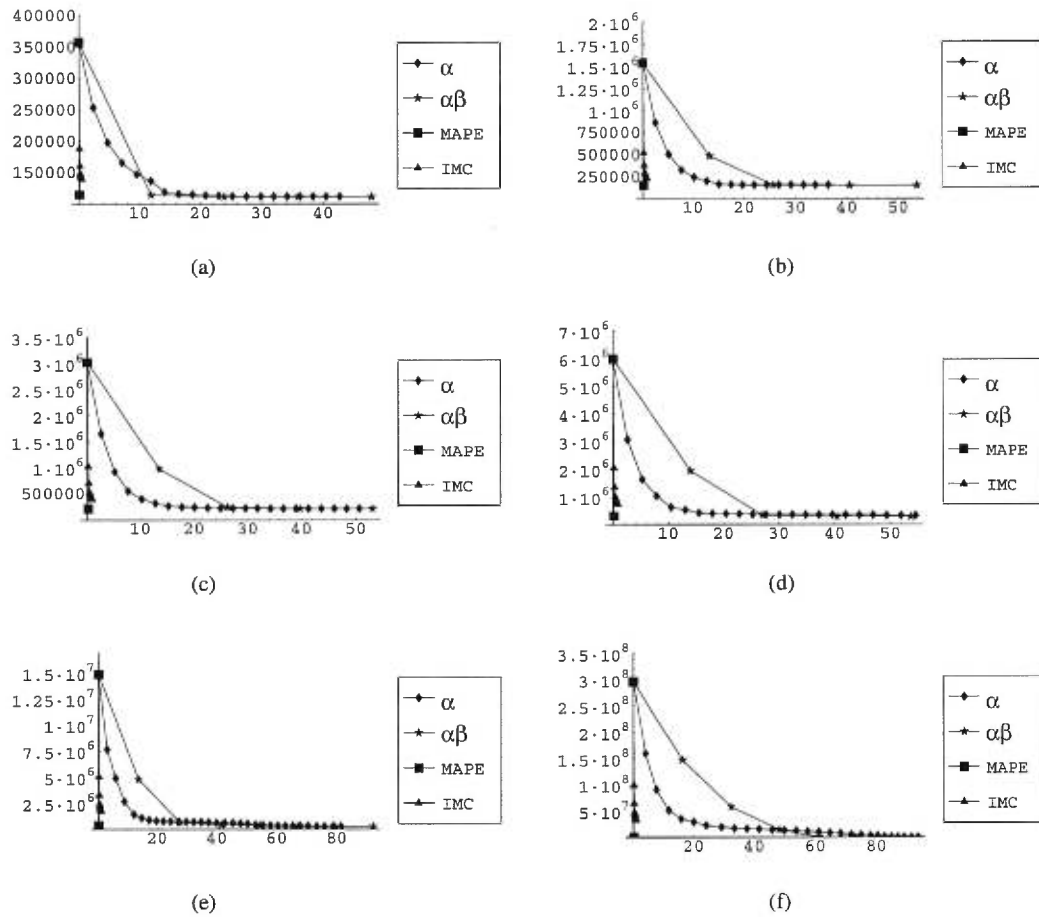


FIG. 7.7 – Graphique de la décroissance de l'énergie dans le temps (en minutes) pour chacun des algorithmes, lorsque le potentiel de clique est linéaire. (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100.0$.

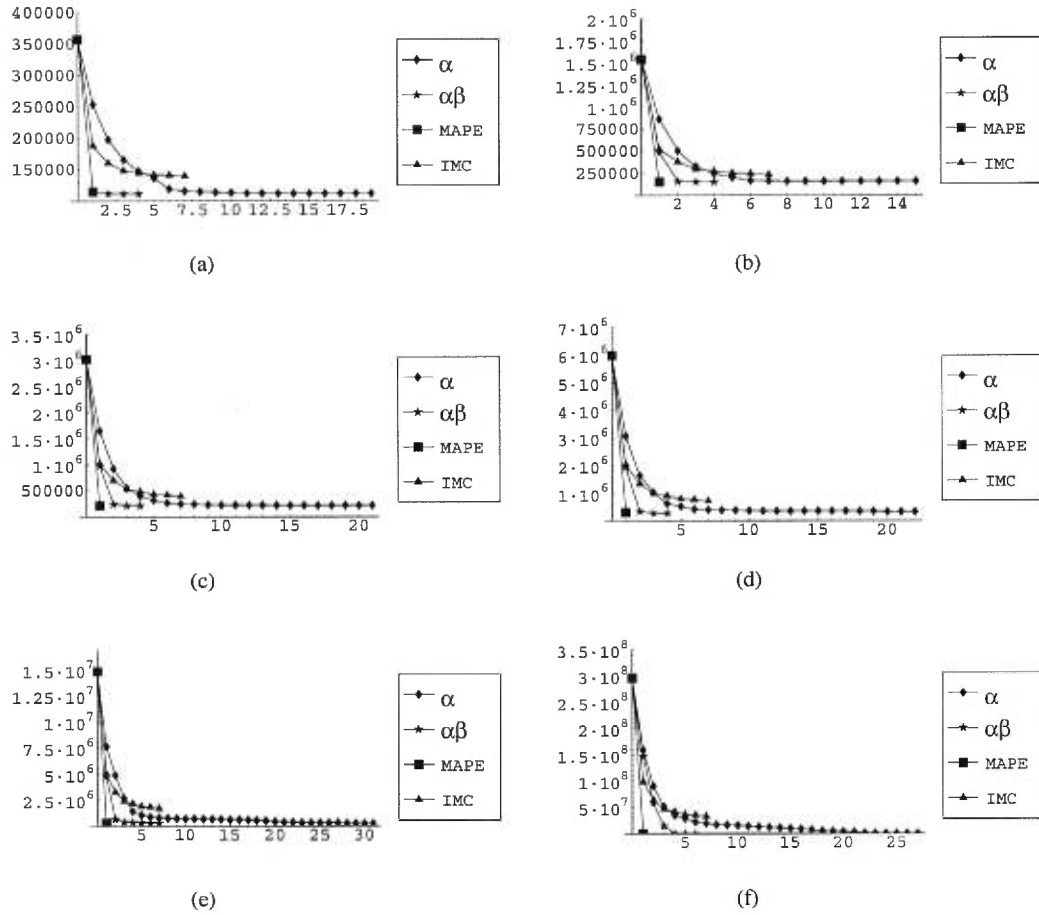


FIG. 7.8 – Graphique de la décroissance de l'énergie par itération des algorithmes lorsque le potentiel de clique est linéaire. (a) $\beta = 0.1$, (b) $\beta = 0.5$, (c) $\beta = 1.0$, (d) $\beta = 2.0$, (e) $\beta = 5.0$ et (f) $\beta = 100.0$.

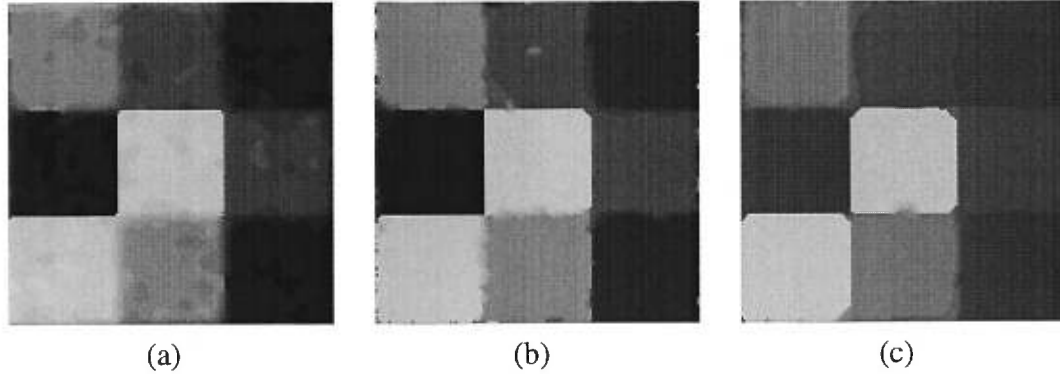


FIG. 7.9 – Résultats d’une restauration utilisant un potentiel de clique quadratique tronqué avec $\beta = 0.7$ et $\alpha = 1200$. (a) résultat obtenu avec l’IMC. (b) résultat obtenu par l’algorithme de réétiquetage α . (c) résultat obtenu par l’algorithme de réétiquetage $\alpha\beta$. L’énergie initiale est de 1334314. L’énergie finale est de : (a) 278586, (b) 170002 et (c) 137970.

7.1.2 Utilisation d’un potentiel de clique quadratique tronqué

L’utilisation d’un potentiel de clique quadratique tronqué permet la préservation des discontinuités. Le potentiel utilisé est :

$$V_{i,i'}(f_i, f_{i'}) = \min\{\beta(f_i - f_{i'})^2, \alpha\},$$

où $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$. La valeur de α est déterminée en fonction de la taille des discontinuités que nous désirons conserver.

Les résultats

Seules les méthodes d’IMC, de réétiquetage α et de réétiquetage $\alpha\beta$ sont utilisées, car la méthode de MAPE s’applique uniquement lorsque le potentiel de clique est linéaire. Les résultats de la restaurations pour $\beta = 0.7$ et $\alpha = 1200$ sont illustrés à la figure 7.9. Encore une fois, l’IMC a offert une piètre performance par rapport aux deux autres techniques, en obtenant une configuration ayant 278586

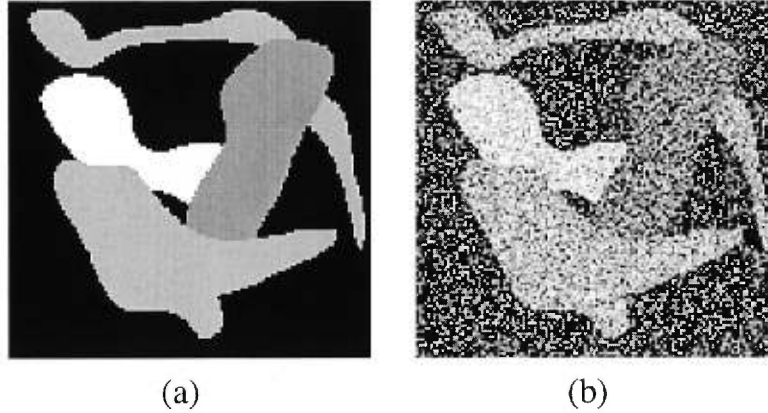


FIG. 7.10 – (a) image originale composée de 4 niveaux de gris : 0, 10, 20 et 30, (b) image bruitée avec un bruit gaussien de moyenne 0 et d'écart-type 10.

comme énergie, ce qui est 1.6 et 2.0 fois plus élevé qu'avec les algorithmes de réétiquetage α et $\alpha\beta$ respectivement.

7.2 Expérience sur la segmentation de régions non-ordonnées

L'image originale (figure 7.10 (a)) est un dessin abstrait de 128×128 pixels contenant des régions (classes) connexes étiquetées dans $\mathcal{L} = \{0, 1, 2, 3\}$. La valeur de ces étiquettes est 0, 10, 20 et 30 respectivement. L'image observée à la figure 7.10 (b) est le résultats de l'ajout d'un bruit gaussien de moyenne $\mu = 0$ et d'écart-type $\sigma = 10$. Le modèle de clique utilisé pour la restauration (segmentation) est un modèle de Potts de la forme :

$$V_{ii'}(f_i, f_{i'}) = \beta(1 - \delta(f_i - f_{i'})),$$

où $\{i, i'\} \in \mathcal{P}_{\mathcal{N}}$ et \mathcal{N} est le système de voisinage 8-voisins.

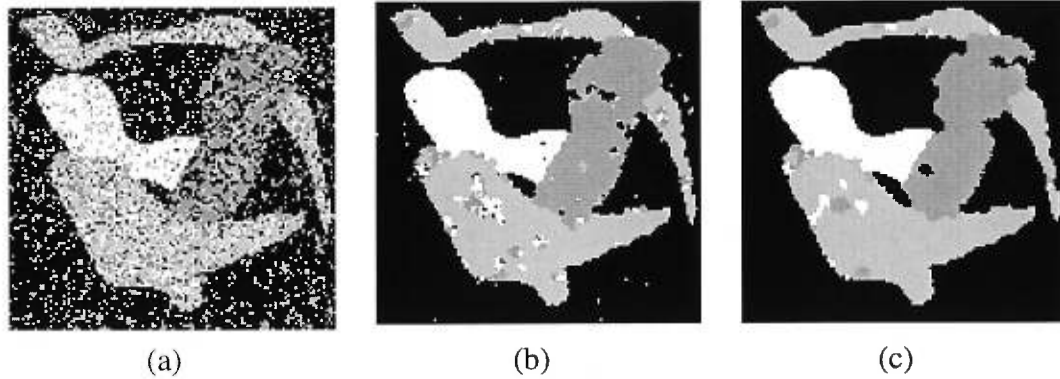


FIG. 7.11 – Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme IMC sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403, (b) 76838 et (c) 104882 . L'énergie finale est : (a) 53313, (b) 56497 et (c) 59041

Résultats

Les résultats obtenus par les algorithmes d'IMC, de réétiquetage α et $\alpha\beta$ sont illustrés aux figures 7.11, 7.12 et 7.13 respectivement. Chacun des algorithmes a été exécuté pour les valeurs : 0.1, 0.5 et 1.0 du paramètre de lissage β . Nous remarquons que plus le paramètre de lissage est fort, moins la qualité de la restauration obtenue par l'algorithme d'IMC est bonne. Tandis que les résultats obtenus par les algorithmes de réétiquetage α et $\alpha\beta$ sont comparables.

7.3 Expérience en vision stéréoscopique

La correspondance stéréoscopique se traite comme un problème de restauration bayésienne d'images. L'observation $X = x$ est une séquence stéréoscopique d'images et les étiquettes de \mathcal{L} sont des disparités (déplacements observés entre deux images de la séquence stéréoscopique). Soit x' et x'' deux images d'une séquence stéréoscopique, alors la disparité établit la correspondance entre le site i

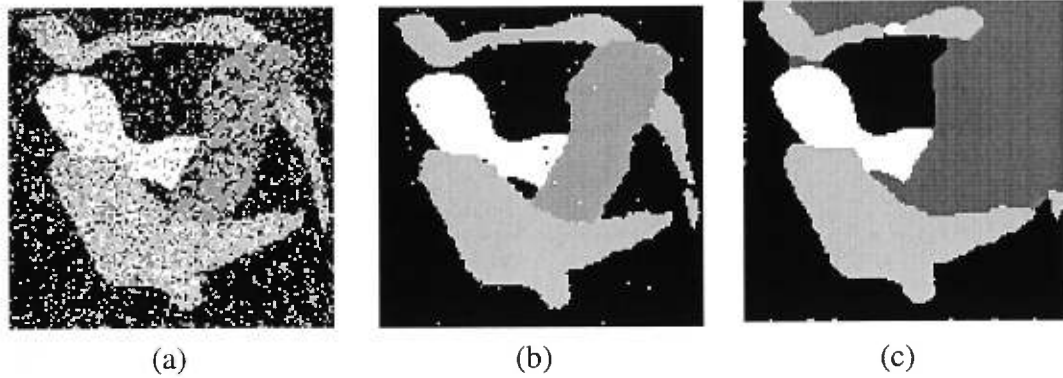


FIG. 7.12 – Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme de réétiquetage α sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403, (b) 76838 et (c) 104882 . L'énergie finale est : (a) 53194, (b) 55177 et (c) 56827

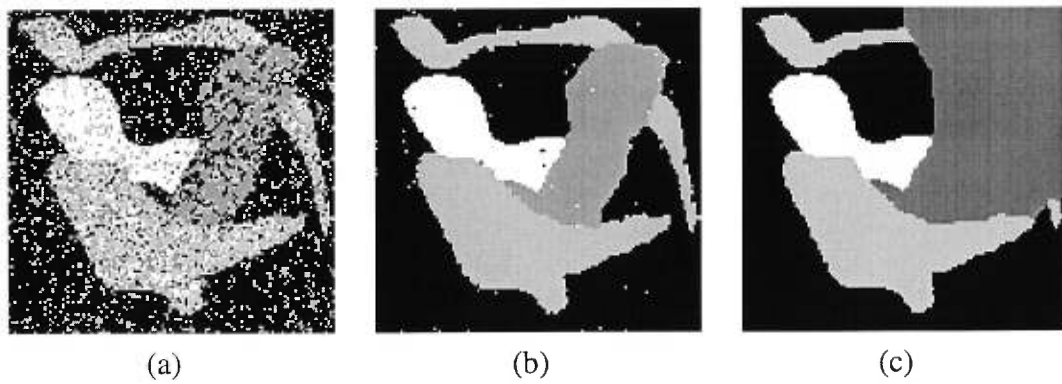


FIG. 7.13 – Résultats de la segmentation (modèle de Potts) obtenus par l'algorithme de réétiquetage $\alpha\beta$ sur l'image bruitée de la figure 7.10 (b). (a) $\beta = 0.1$, (b) $\beta = 0.5$ et (c) $\beta = 1.0$. L'énergie initiale est : (a) 54403 (b) 76838 (c) 104882. L'énergie finale est : (a) 53193, (b) 55180 et (c) 56603.

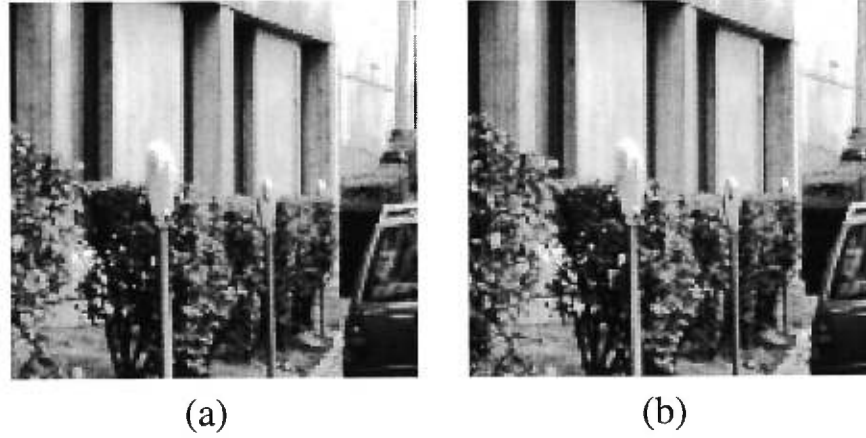


FIG. 7.14 – Image de la scène du parcomètre

de l'image x' et le site j de l'image x'' selon la relation suivante :

$$i = g(j, k_j),$$

où $k_j \in \mathcal{L}$ et $g(\cdot)$ est une fonction qui dépend de la géométrie de caméra.

7.3.1 La séquence du parcomètre

La séquence du parcomètre contient deux images (gauche et droite) de 128×128 pixels : $x = \{x^g, x^d\}$ (voir la figure 7.14 (a) et (b)). Le déplacement horizontal maximum entre deux points de cette séquence n'excède pas 6 pixels. L'ensemble \mathcal{L} contient 24 étiquettes (disparités) uniformément réparties entre 0 et 6. Nous définissons la fonction de vraisemblance comme :

$$\begin{aligned} Pr(x|f) &= \prod_{i \in \mathcal{S}} Pr(x_i | f_i) \\ &= \prod_{i \in \mathcal{S}} \frac{e^{-(x_i^g - x_{g(i, f_i)}^d)^2}}{K}, \end{aligned}$$

où K est une constante de normalisation et $g(i, f_i)$ est une fonction qui associe le site de \mathcal{S} qui est en correspondance avec le site i par la disparité f_i . Dans cette application, la géométrie de caméra est horizontale et pour tout $i \in \mathcal{S}$, nous avons $i = (i_x, i_y)$ et $g(i, f_i) = (i_x + f_i, i_y)$.

Résultats

Dans les figures 7.15, 7.16 et 7.17, nous illustrons les résultats de la correspondance stéréoscopique, lorsque les potentiels de clique sont linéaire, Potts et quadratique tronqué respectivement.

De façon générale, les algorithmes de MAPE, de réétiquetage α et $\alpha\beta$ ont permis d'obtenir un niveau d'énergie substantiellement plus bas qu'avec la méthode d'IMC. Toutefois, pour le potentiel de clique quadratique tronqué utilisé, l'algorithme d'IMC a offert une performance légèrement supérieure aux algorithmes de réétiquetage α et $\alpha\beta$.

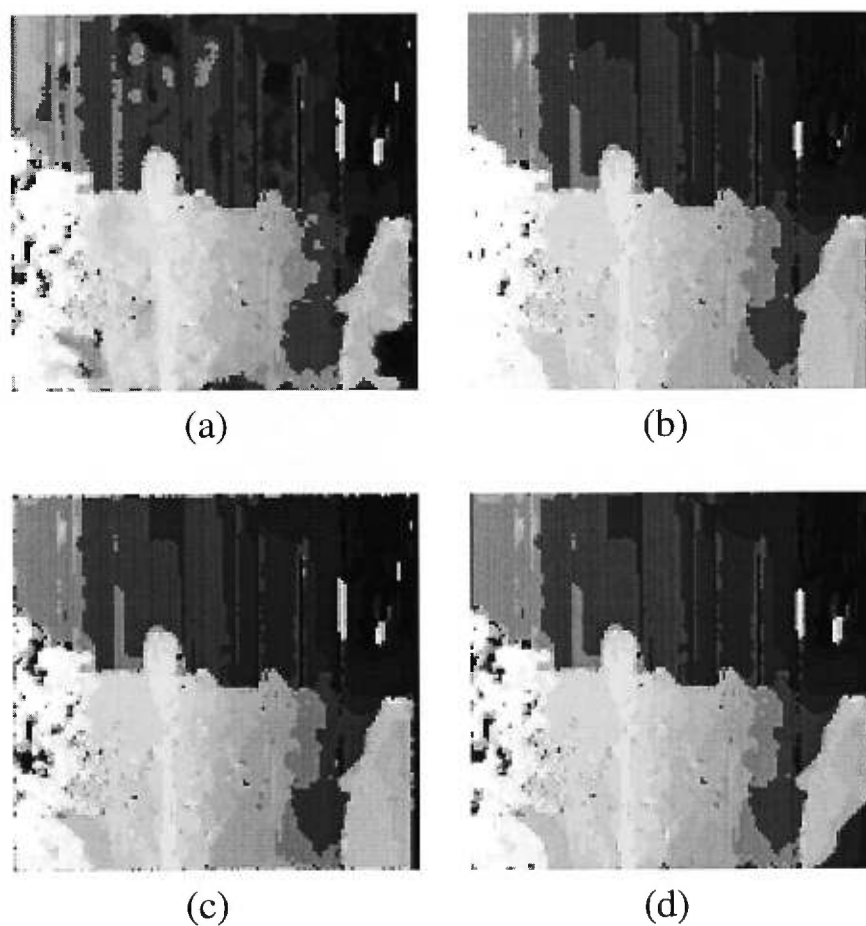


FIG. 7.15 – Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique linéaire, 24 disparités et $\beta = 2.0$. (a) MAPE, (b) IMC, (c) réétiquetage α et (d) réétiquetage $\alpha\beta$. L'énergie initiale est de 1012729. L'énergie finale est de : (a) 154702, (b) 228067, (c) 205988 et (d) 220853.

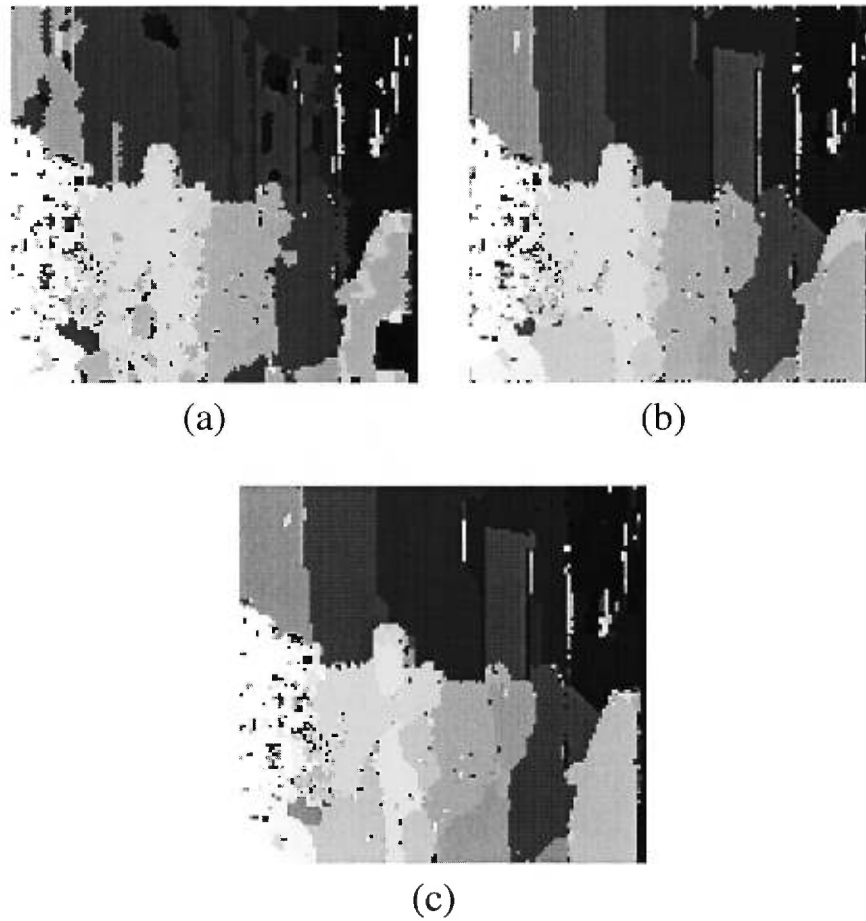


FIG. 7.16 – Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique de Potts, 24 disparités et $\beta = 10.0$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 850966. L'énergie finale est de : (a) 302393, (b) 244793 et (c) 248247.

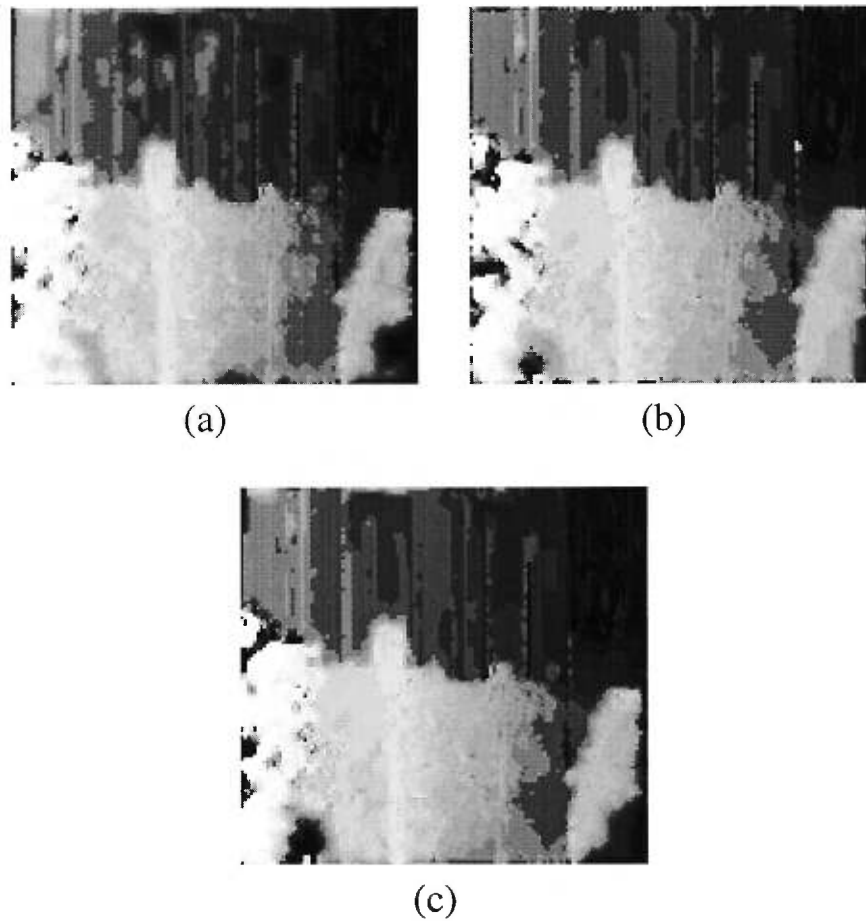


FIG. 7.17 – Résultats de la correspondance stéréoscopique obtenus sur la séquence du parcomètre en utilisant un potentiel de clique quadratique tronqué, 24 disparités, $\beta = 1.0$ et $\alpha = 50$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 2329670. L'énergie finale est de : (a) 220891, (b) 230540 et (c) 242365.

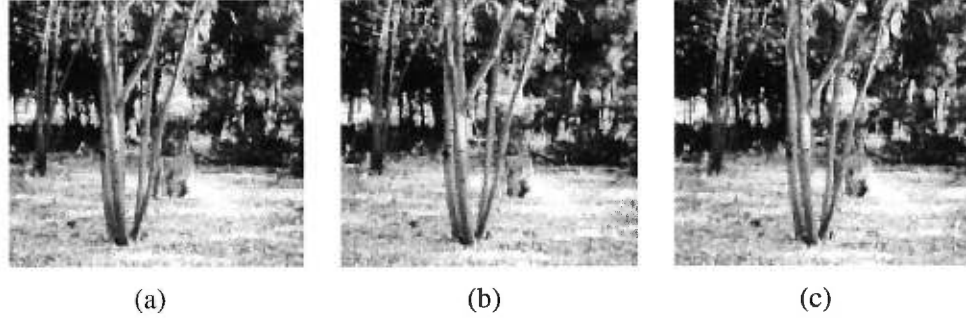


FIG. 7.18 – Image de la scène de l’arbre

7.3.2 La séquence de l’arbre

La séquence de l’arbre contient trois images (gauche, centre et droite) de 128×128 pixels : $x = \{x^g, x^c, x^d\}$ (voir la figure 7.18 (a) (b) et (c)). Le déplacement horizontal maximum entre deux points de cette séquence n’excède pas 4 pixels. Les 24 disparités de \mathcal{L} sont réparties uniformément entre 0 et 4. Nous définissons la fonction de vraisemblance comme :

$$\begin{aligned} Pr(x|f) &= \prod_{i \in \mathcal{S}} Pr(x_i|f_i) \\ &= \prod_{i \in \mathcal{S}} \frac{e^{-((x_i^c - x_{g(i, f_i)}^g)^2 + (x_i^c - x_{h(i, f_i)}^d)^2)}}{K}, \end{aligned}$$

où K est une constante de normalisation et les fonctions $g(i, f_i)$ et $h(i, f_i)$ dépendent de la géométrie de caméra. Dans cette application, la géométrie de caméra est horizontale. De plus, pour tout $i \in \mathcal{S}$ nous avons $i = (i_x, i_y)$, $g(i, f_i) = (i_x - f_i, i_y)$ et $h(i, f_i) = (i_x + f_i, i_y)$.

Résultats

Les résultats obtenus sur la séquence de l’arbre sont présentés aux figures 7.19, 7.20 et 7.21, pour les potentiels de clique : linéaire, Potts et quadratique tronqué respectivement.

Encore une fois, les algorithmes basés sur la coupe de coût minimum ont offert

une meilleure performance, à l'exception du résultat de la figure 7.21 obtenu pour un potentiel de clique quadratique tronqué.

Le résultat de la figure 7.21 est décevant pour les algorithmes de réétiquetage α et $\alpha\beta$. En effet, cela démontre bien les limites de l'optimisation locale et que peu importe l'heuristique utilisé, ça demeure un heuristique.

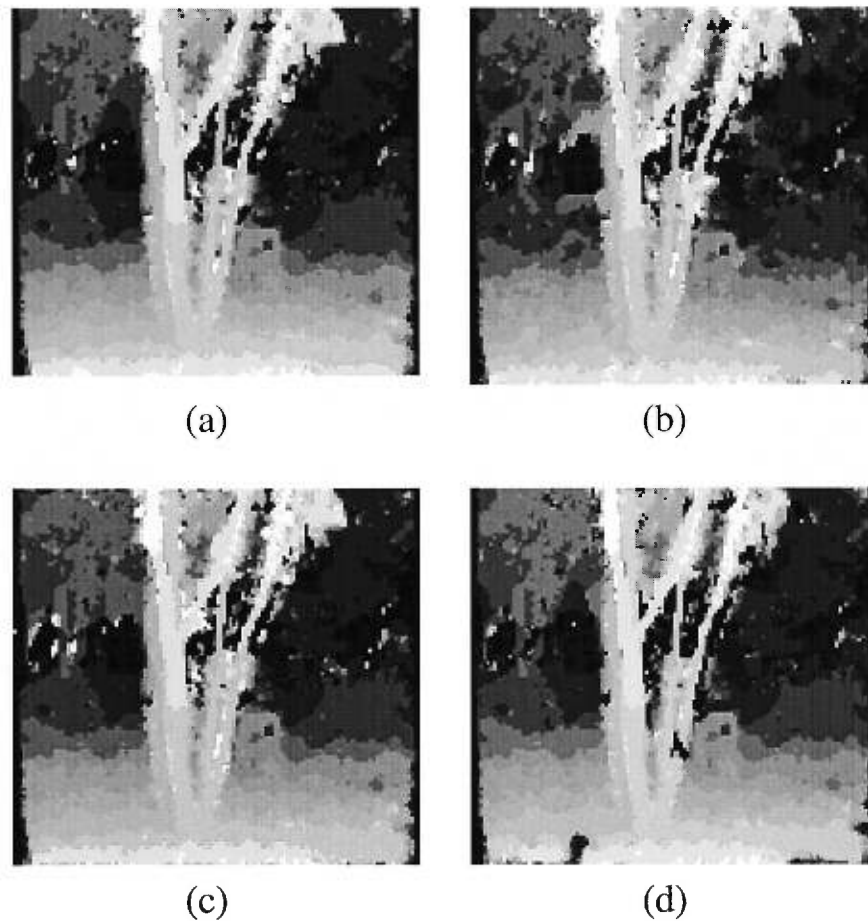


FIG. 7.19 – Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique linéaire, 24 disparités et $\beta = 20.0$. (a) MAPE, (b) IMC, (c) réétiquetage α et (d) réétiquetage $\alpha\beta$. L'énergie initiale est de 9562952. L'énergie finale est de : (a) 1605728, (b) 2553917 , (c) 2519824 et (d) 3057936.

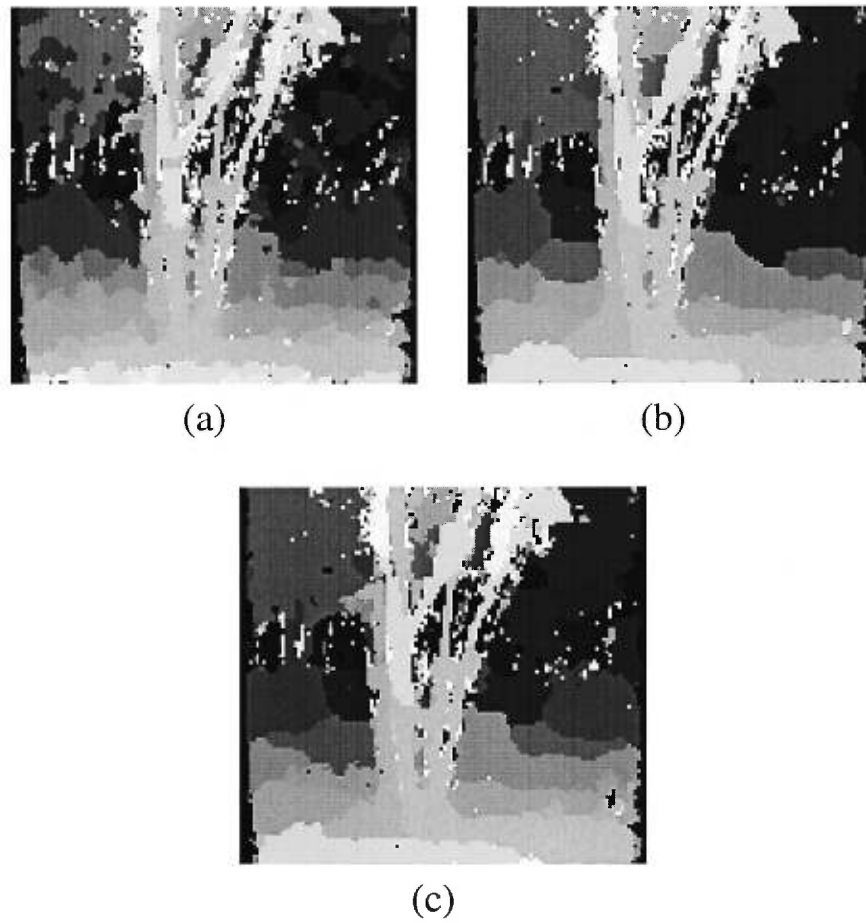


FIG. 7.20 – Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique de Potts, 24 disparités et $\beta = 100$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 8481311. L'énergie finale est de : (a) 3760584, (b) 3166478 et (c) 3591949.

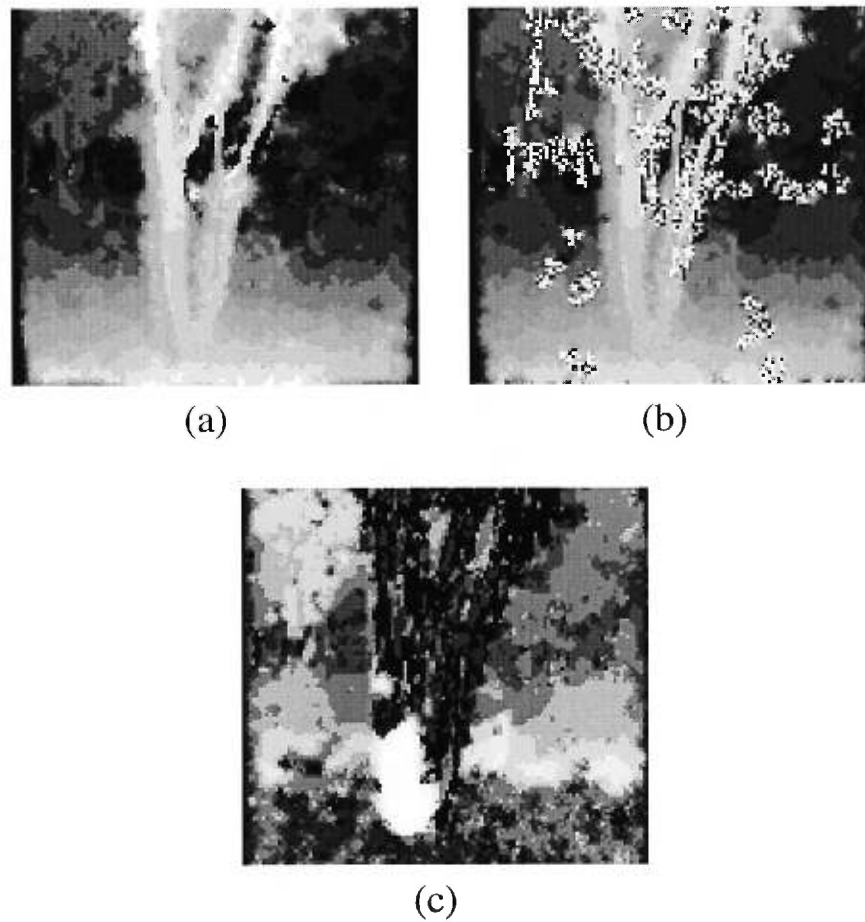


FIG. 7.21 – Résultats de la correspondance stéréoscopique obtenus sur la séquence de l'arbre en utilisant un potentiel de clique quadratique tronqué, 24 disparités, $\beta = 20.0$ et $\alpha = 700$. (a) IMC, (b) réétiquetage α et (c) réétiquetage $\alpha\beta$. L'énergie initiale est de 32100146. L'énergie finale est de : (a) 3301562, (b) 12208421 et (c) 9178847.

Chapitre 8

Discussion et conclusion

8.1 Discussion

Des algorithmes d'optimisation basés sur la coupe de coût minimum dans des graphes de flot ont été proposés et analysés. Dans un premier temps, nous avons obtenu des résultats théoriques sous forme de preuve mathématique qui démontre que les algorithmes de MAPE et de réétiquetage α permettent de minimiser la fonction d'énergie *a posteriori* du problème d'étiquetage bayésien.

Par l'expérimentation, nous avons comparé la performance de ces algorithmes à des techniques d'optimisation standards et récemment développés, dont le recuit simulé, l'IMC et le réétiquetage $\alpha\beta$.

Malgré sa limitation au potentiel de clique linéaire, l'algorithme MAPE a fourni des résultats intéressants autant sur des données synthétiques que sur des cas réels en correspondance stéréoscopique.

Pour un nombre d'étiquettes et une taille d'image raisonnable, le MAPE offre des temps d'exécution comparables à l'IMC, tout en garantissant d'obtenir l'optimum global de la fonction d'énergie *a posteriori*. Toutefois, sous la forme actuelle, le MAPE ne peut s'attaquer à des problèmes de très grandes tailles, car la taille du graphe de flot devient alors prohibitive.

Pour contourner la limitation du MAPE au potentiel de clique linéaire, nous avons laissé tomber la recherche globale en proposant l’algorithme de réétiquetage α (minimisation locale). Cette algorithme a offert une performance supérieure à l’IMC et comparable au réétiquetage $\alpha\beta$, tout en ayant une complexité inférieure à ce dernier.

Les algorithmes proposés et analysés, dont le MAPE, le réétiquetage α et $\alpha\beta$, permettent de modifier l’étiquetage de plusieurs sites à la fois. Cette propriété engendre des espaces de recherche beaucoup plus grands (toutes les configurations possibles dans le cas du MAPE) qu’avec l’IMC. Par conséquent, ils ont plus de chances de trouver un minimum locale ayant une énergie plus basse.

8.1.1 Améliorations

Les temps d’exécution des différents algorithmes pourraient être substantiellement améliorés. Plusieurs optimisations peuvent être apportées au stockage et à la représentation des graphes. De plus, l’algorithme de calcul du flot maximum de Ford et Fulkerson [FF62] est probablement plus approprié pour les algorithmes de réétiquetage α et $\alpha\beta$, car contrairement au MAPE, les graphes y sont beaucoup plus larges que profonds.

Pour rendre les techniques présentées applicables à des problèmes de plus grande taille, il faudra porter un effort particulier au développement d’algorithmes de graphe parallèle et “multi-échelle”. Ainsi, l’espace requis pour stocker les graphes sera réduit et la performance augmentée.

Les algorithmes de réétiquetage α et $\alpha\beta$ permettent de diminuer rapidement (en quelques itérations) l’énergie. Toutefois, après ces quelques itérations, nous n’obtenons plus de diminution substantielle de l’énergie. Nous pourrions instaurer un critère d’arrêt basé sur le taux de diminution de l’énergie à chaque itération ou encore, nous pourrions songer à utiliser des techniques d’optimisation hybrides qui combinent une des ces méthodes à une technique plus efficace en fin d’optimisation.

8.2 Conclusion

Finalement, ce mémoire a permis de développer et de faire la lumière sur les possibilités et les limitations de différents algorithmes d'optimisation utilisant la coupe de coût minimum dans un graphe de flot pour résoudre un problème associé à l'estimation du maximum *a posteriori* en étiquetage bayésien. Bien que nous nous sommes limités à des applications en restauration d'images et en correspondance stéréoscopique, nous croyons que l'utilisation de ces techniques en vision artificielle seront multiples. Nous songeons en particulier au calcul du flot optique et à divers problèmes de reconstruction d'images associés à l'imagerie médicale [BM99, BMR99, RG99].

Bibliographie

- [AMJB93] R. K. Ahuja, T. Magnanti et Orlin J. B. *Network flows : theory, algorithms and applications*. Prentice Hall, Englewood Cliffs, 1993.
- [Bes86] J. E. Besag. « On the statistical analysis of dirty pictures ». *Journal of the Royal Statistical Society, Series B*, volume 48, pages 259–302, 1986.
- [BM99] M. Bonneville et J. Meunier. « Finding the exact optical flow : a maximum flow formulation ». In *Vision Interface'99*, Trois-Rivière, Canada, 1999.
- [BMR99] M. Bonneville, J. Meunier et S. Roy. « Graph Cut : Application to Bayesian Emission Tomography Reconstruction ». In *IEEE International Conference on Image Analysis and Processing*, Venice, Italy, 1999.
- [BRME98] M. Bonneville, S. Roy, J. Meunier et A. C. Evans. « Using maximum flow for exact maximum a posteriori estimation in image restoration and segmentation ». In *Technical report NEC research*, 1998.
- [BVZ98] Y. Boykov, O. Veksle et R. Zabih. « Markov random fields with efficient approximations ». In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.
- [BVZ99] Y. Boykov, O. Veksle et R. Zabih. « Fast Approximate Energy Minimization via Graph Cuts ». In *International Conference on Computer*

- Vision*, Greece, 1999.
- [BZ87] A. Blake et A. Zisserman. *Visual reconstruction*. Cambridge, MA : MIT press, 1987.
- [CLR90] T. H. Cormen, C. E. Leiserson et R. L. Rivest. *Introduction to algorithms*. McGraw-Hill, New York, 1990.
- [FF62] L. R. Ford et D. R. Fulkerson. *Flows in networks*. Princeton University Press, Princeton, N.J., 1962.
- [GG84] S. Geman et D. Geman. « Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 6, pages 721–741, 1984.
- [GPS89] D. Greig, B. Porteous et A. Seheult. « Exact maximum a posteriori estimation for binary images ». *Journal of the Royal Statistical Society, Series B*, volume 51, numéro 2, pages 271–279, 1989.
- [GR97] A. V. Golberg et S. B. Rao. « Length function for flow computation ». Rapport technique 97-055, NEC Research Institute, Princeton NJ, 1997.
- [GT88] A. V. Golberg et R. E. Tarjan. « A new approach to the maximum flow problem ». *J. Assoc. Comput. Mach.*, volume 35, pages 921–940, 1988.
- [IG98a] H. Ishikawa et D. Geiger. « Occlusions, discontinuities, and epipolar lines in stereo ». In *Fifth European Conference on Computer Vision*, Freiburg, Germany, 1998.
- [IG98b] H. Ishikawa et D. Geiger. « Segmentation by grouping junctions ». In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.

- [Kar74] A. V. Karzanov. « Determining the maximal flow in a network by the method of preflows ». *Soviet Mathematics Doklady*, volume 15, pages 434–437, 1974.
- [Kir83] S. C. D. Kirkpatrick. « Optimization by simulated annealing ». *Science*, volume 220, pages 671–680, 1983.
- [Li95] S. Li. *Markov random field modeling in computer vision*. Springer-Verlag, 1995.
- [RC98] S. Roy et I. Cox. « A maximum-flow formulation of the n-camera stereo correspondence problem ». In *6th International Conference on Computer Vision*, Bombay, India, 1998.
- [RG99] S. Roy et V. Govindu. « MRF Solutions for probabilistic optical flow formulations ». In *IEEE Workshop on Graph Algorithms and Computer Vision*, Corfu, Greece, 1999.