

Université de Montréal

Le rendu en demi-ton avec sensibilité à la structure

par
Jianghao Chang

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

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

juillet, 2011

© Jianghao Chang, 2011.

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

Le rendu en demi-ton avec sensibilité à la structure

présenté par:

Jianghao Chang

a été évalué par un jury composé des personnes suivantes:

| | |
|----------------------|------------------------|
| Jean Meunier, | président-rapporteur |
| Victor Ostromoukhov, | directeur de recherche |
| Pierre Poulin, | membre du jury |

Mémoire accepté le:

RÉSUMÉ

Dans ce mémoire nous allons présenter une méthode de diffusion d'erreur originale qui peut reconstruire des images en demi-ton qui plaisent à l'œil. Cette méthode préserve des détails fins et des structures visuellement identifiables présentes dans l'image originale.

Nous allons tout d'abord présenter et analyser quelques travaux précédents afin de montrer certains problèmes principaux du rendu en demi-ton, et nous allons expliquer pourquoi nous avons décidé d'utiliser un algorithme de diffusion d'erreur pour résoudre ces problèmes. Puis nous allons présenter la méthode proposée qui est conceptuellement simple et efficace. L'image originale est analysée, et son contenu fréquentiel est détecté. Les composantes principales du contenu fréquentiel (la fréquence, l'orientation et le contraste) sont utilisées comme des indices dans un tableau de recherche afin de modifier la méthode de diffusion d'erreur standard. Le tableau de recherche est établi dans un étape de pré-calcul et la modification est composée par la modulation de seuil et la variation des coefficients de diffusion.

Ensuite le système en entier est calibré de façon à ce que ces images reconstruites soient visuellement proches d'images originales (des aplats d'intensité constante, des aplats contenant des ondes sinusoïdales avec des fréquences, des orientations et des contrastes différents).

Finalement nous allons comparer et analyser des résultats obtenus par la méthode proposée et des travaux précédents, et démontrer que la méthode proposée est capable de reconstruire des images en demi-ton de haute qualité (qui préservent des structures) avec un traitement de temps très faible.

Mots clés: rendu en demi-ton, diffusion d'erreur, sensibilité aux structures d'une image, seuil, coefficients de diffusion, calibration, analyse, contenu fréquentiel.

ABSTRACT

In this work we present an original error-diffusion method which produces visually pleasant halftone images while preserving fine details and visually identifiable structures present in original images.

We first present and analyze the previous work to show the major problems in halftoning, and explain why we decided to use an error diffusion algorithm to solve the problems. Then we present our method which is conceptually simple and computationally efficient. The source image is analyzed, and its local frequency content is detected. The main components of the frequency content (main frequency, orientation, and contrast) serve as lookup table indices in a pre-computed database of modifications to a standard error diffusion. The modifications comprise threshold modulation and variation of error-diffusion coefficients.

The whole system is calibrated in such a way that the produced halftone images are visually close to original images (patches of constant intensity, patches containing sinusoidal waves of different frequencies/orientations/contrasts, as well as natural images of different origins).

Finally, we compare and analyze the results obtained by our method and previous work, and show that our method can produce high-quality halftone image (which is structure aware) within very short time.

Keywords: halftoning, error diffusion, structure aware, threshold, diffusion coefficients, calibration, analysis, frequency content.

TABLE DES MATIÈRES

| | |
|---|-------------|
| RÉSUMÉ | iii |
| ABSTRACT | iv |
| TABLE DES MATIÈRES | v |
| LISTE DES TABLEAUX | vii |
| LISTE DES FIGURES | viii |
| REMERCIEMENTS | x |
| CHAPITRE 1 : INTRODUCTION | 1 |
| 1.1 Introduction sur le rendu en demi-ton | 1 |
| 1.2 L'objectif | 1 |
| 1.3 Classification des algorithmes de rendu en demi-ton | 2 |
| CHAPITRE 2 : L'ÉTAT DE L'ART | 4 |
| 2.1 Rendu en demi-ton par point | 4 |
| 2.1.1 Méthodes de base | 4 |
| 2.1.2 Méthodes avec des propriétés de bruit bleu | 8 |
| 2.2 Rendu en demi-ton par voisinage | 13 |
| 2.2.1 Diffusion d'erreur | 13 |
| 2.2.2 Amélioration de l'ordre de parcours | 15 |
| 2.2.3 Amélioration du seuillage | 19 |
| 2.2.4 Amélioration des coefficients de diffusion | 25 |
| 2.3 Rendu en demi-ton par optimisation globale | 30 |
| 2.3.1 Recherche binaire directe | 30 |
| 2.3.2 <i>Structure-aware halftoning</i> | 32 |

| | |
|--|-----------|
| CHAPITRE 3 : LE RENDU EN DEMI-TON AVEC SENSIBILITÉ À LA STRUCTURE | 35 |
| 3.1 Analyse de fréquences locales | 36 |
| 3.1.1 Orientation locale | 36 |
| 3.1.2 Fréquence locale et contraste local | 38 |
| 3.1.3 Seuillage | 41 |
| 3.2 Coefficients de diffusion | 42 |
| 3.3 Calibration | 43 |
| CHAPITRE 4 : RESULTAT | 45 |
| 4.1 Evaluation visuelle | 45 |
| 4.2 Evaluation numérique | 70 |
| 4.2.1 PSNR | 70 |
| 4.2.2 MSSIM | 73 |
| 4.2.3 Temps de traitement | 78 |
| CHAPITRE 5 : CONCLUSION | 80 |
| 5.1 Limitations | 80 |
| 5.2 Travaux futurs | 83 |
| BIBLIOGRAPHIE | 84 |

LISTE DES TABLEAUX

| | | |
|-------|---|----|
| 2.I | Comparaison des coefficients de diffusion | 27 |
| 4.I | Comparaison de la métrique PSNR | 71 |
| 4.II | Comparasion de la Metrique MSSIM | 77 |
| 4.III | Comparaison des temps de traitement | 79 |

LISTE DES FIGURES

| | | |
|------|---|----|
| 1.1 | Rendu en demi-ton | 2 |
| 2.1 | La fenêtre de seuil de la méthode tramage de points regroupes . . . | 5 |
| 2.2 | La fenêtre de seuil de la méthode tramage de points disperses . . . | 5 |
| 2.3 | Le résultat de tramage de points regroupés | 6 |
| 2.4 | Le résultat de tramage de points dispersés | 7 |
| 2.5 | La fenêtre de seuil de <i>Void-and-Clusters</i> pour la génération d'un tableau de <i>dither</i> | 9 |
| 2.6 | La fenêtre de seuil de <i>Blue Noise Mask</i> | 10 |
| 2.7 | Résultat du <i>Blue Noise Mask</i> | 11 |
| 2.8 | Modèle de la méthode de Floyd et Steinberg de diffusion d'erreur | 14 |
| 2.9 | Ordre de parcours " <i>raster</i> " | 17 |
| 2.10 | Ordre de parcours en serpentins | 17 |
| 2.11 | Ordre de parcours courbe de remplissage | 18 |
| 2.12 | La méthode de Eschbach et Knox | 20 |
| 2.13 | Méthode proposée par Hwang et al. | 21 |
| 2.14 | La comparaison des méthodes d'amélioration du seuillage | 24 |
| 2.15 | La comparaison de la méthode de Floyd et Steinberg et de Ostro- moukhov | 28 |
| 2.16 | La comparaison des méthodes de Floyd et Steinberg, de Ostro- moukhov et de Shiau et Fan | 29 |
| 2.17 | La comparaison entre la méthode de Floyd et Steinberg et de la recherche binaire directe | 31 |
| 3.1 | Schéma de la méthode proposée | 35 |
| 3.2 | Orientations locales | 38 |
| 3.3 | Fréquences locales | 40 |
| 3.4 | Contrastes locaux | 40 |
| 3.5 | Calibration des paramètres dans la méthode proposée | 44 |

| | | |
|------|---|----|
| 4.1 | Bras | 47 |
| 4.2 | Osier | 48 |
| 4.3 | Chauve-souris | 49 |
| 4.4 | Bateau | 50 |
| 4.5 | Chat (original) | 51 |
| 4.5 | Chat (Ostromoukhov) | 51 |
| 4.5 | Chat (rehaussement de bordures : fort) | 52 |
| 4.5 | Chat (rehaussement de bordure : faible) | 52 |
| 4.5 | Chat (Pang et al.) | 53 |
| 4.5 | Chat (méthode proposée) | 53 |
| 4.6 | Désert | 54 |
| 4.7 | Genou | 55 |
| 4.8 | Lion | 56 |
| 4.9 | Oreille interne | 57 |
| 4.10 | Marmotte | 58 |
| 4.11 | Vieil homme | 59 |
| 4.12 | Plage (original) | 60 |
| 4.12 | Plage (Ostromoukhov) | 61 |
| 4.12 | Plage (rehaussement de bordures : fort) | 62 |
| 4.12 | Plage (rehaussement de bordures : faible) | 63 |
| 4.12 | Plage (Pang et al.) | 64 |
| 4.12 | Plage (méthode proposée) | 65 |
| 4.13 | Route | 66 |
| 4.14 | Arbre | 67 |
| 4.15 | Autoportraits de Van Gogh | 68 |
| 4.16 | Zèbres | 69 |
| 5.1 | Limitation de la méthode proposée | 82 |

REMERCIEMENTS

Je tiens tout d'abord à remercier mon directeur de recherche, Victor Ostromoukhov pour m'avoir guidé dans ma démarche de recherche et m'avoir mené jusqu'à une solide publication au cours de ma maîtrise. Egalement, merci d'avoir lu et évalué mon mémoire

Mes remerciements vont également à Benoît Alain, deuxième responsable de la rédaction de notre article, qui fut aussi un excellent collègue de recherche.

Ensuite, je remercie Pierre Poulin, pour avoir lu et aidé à évaluer mon mémoire. Egalement, merci de m'avoir aidé à corriger mon mémoire au niveau du français et m'avoir donné des conseils pour la mise en page.

J'aimerais remercier Jean Meunier, pour avoir lu et évalué mon mémoire.

Par ailleurs, j'aimerais remercier François Duranleau, Pierre Poulin, Neil Stewart et Jean Vaucher pour l'aide précieuse qu'ils ont fourni dans la préparation de notre article ainsi que notre présentation à SIGGRAPH Asia 2009.

Finalement, j'aimerais remercier Jingxue Yu, pour m'avoir aidé à corriger la mise en page de mon mémoire.

CHAPITRE 1

INTRODUCTION

1.1 Introduction sur le rendu en demi-ton

La rendu en demi-ton est une technique qui reconstruit l'image d'un ton continu en une image d'un ton binaire selon des propriétés de l'image originale. Quand l'image originale contient un variété infinie de couleurs ou de niveaux de gris, une méthode de rendu en demi-ton réduit sa variété infinie de couleurs et la convertit en format binaire en utilisant moins de couleurs, grâce à un mécanisme de l'œil humain. L'image reconstruite est une approximation de l'image originale quand on la voit en distance.

Le terme rendu en demi-ton est un nom général qu'on donne aux méthodes qui convertissent de l'information visuelle du domaine continu au domaine discret. Ce rendu peut être utilisé dans beaucoup de domaines où il faut réduire le nombre de couleurs d'une image, par exemple pour la compression d'images. Mais le domaine où cette méthode est le plus souvent utilisé est pour imprimer des images, c'est-à-dire, pour convertir l'image originale en une image qui contient seulement la couleur de l'encre (noir pour imprimer l'image en niveaux de gris, et cyan, magenta, jaune et noir pour imprimer l'image en couleur). La figure 1.1 est un exemple typique de rendu en demi-ton à partir d'une image en niveaux de gris.

Dans ce mémoire, nous nous concentrons sur le rendu en demi-ton digital et sur le rendu en demi-ton pour seulement des images en niveaux de gris.

1.2 L'objectif

En raison de la réduction de couleurs sur l'image reconstruite par le rendu en demi-ton, il y a certaines informations de l'image originale sont perdues. Quand la méthode de rendu en demi-ton reconstruit l'image originale, elle va "ajouter" du bruit de hautes fréquences sur l'image, parce qu'elle utilise seulement les couleurs noir et blanc (du bruit de hautes fréquences vient de la bordure entre des pixels noir et blanc). Ce bruit de hautes

fréquences va parfois couvrir des signaux de hautes fréquences existants dans l'image originale. Comme résultat, de fins détails et certaines textures seront moins visibles sur l'image reconstruite. Ainsi une bonne méthode de rendu en demi-ton doit minimiser la perte d'information et conserver le plus de détails et de fines textures dans l'image reconstruite.

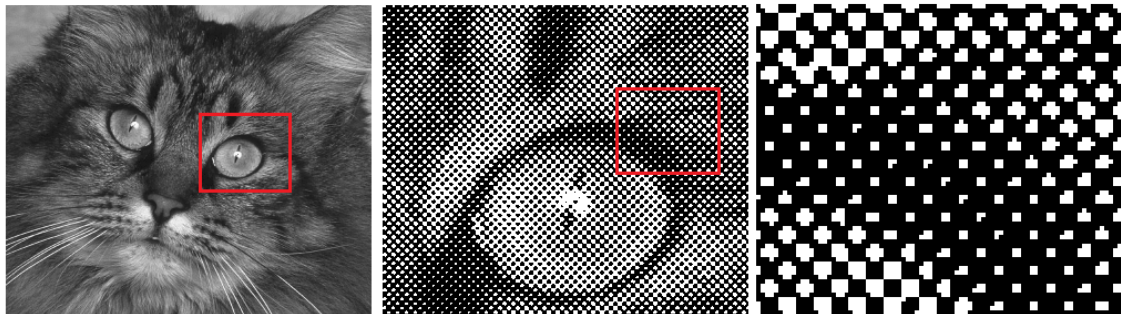


Figure 1.1 – Rendu en demi-ton : un exemple de rendu en demi-ton sur une image naturelle. De gauche à droite : l'image reconstruite à trois échelles différentes.

1.3 Classification des algorithmes de rendu en demi-ton

Il existe plusieurs façons de classifier des algorithmes de rendu en demi-ton. Dans ce mémoire nous utilisons la classification introduite par [15, 22], qui est une classification claire selon le temps traitement et selon la qualité de l'image reconstruite. La métrique utilisée dans cette classification est le nombre de pixels à traiter pour définir la couleur d'un pixel, c'est-à-dire, l'information considérée pour définir la couleur d'un pixel. Selon [22] et [15], nous pouvons classifier les méthodes de rendu en demi-ton en trois familles :

- Rendu en demi-ton par point : pour définir la couleur d'un pixel dans l'image reconstruite, la méthode considère seulement l'intensité de ce pixel dans l'image originale.
- Rendu en demi-ton par voisinage : pour définir la couleur d'un pixel dans l'image reconstruite, la méthode considère l'intensité de ce pixel et les intensités de ses voisins dans l'image originale.

- Rendu en demi-ton par optimisation globale : la couleur d'un pixel dans l'image reconstruite est définie par l'évaluation d'une fonction critère qui considère l'image originale en entier.

Dans ce mémoire, nous allons tout d'abord présenter une sélection des méthodes de rendu en demi-ton selon la classification, et comparer et discuter leurs résultats. Puis nous allons présenter la méthode proposée "*Structure Aware Halftoning*". Au chapitre 4, nous allons comparer la qualité des images reconstruites, les temps de traitement et des évaluations numériques entre la méthode proposée et des méthodes récentes au domaine. A la fin, nous allons discuter les limitation et suggérer des travaux futurs.

CHAPITRE 2

L'ÉTAT DE L'ART

Dans ce chapitre, trois familles des méthodes de rendu en demi-ton sont présentées. Des avantages et des inconvénients pour chacune des trois familles sont discutés à la fin de chaque section. Le but de ces explications est de donner une idée claire sur l'histoire des méthodes de rendu en demi-ton pour mieux situer la méthode proposée. Pour cette raison, la famille "par voisinage" est présentée plus en profondeur que les deux autres familles, et la famille "par point" n'est présentée que très brièvement parce qu'elle produit de mauvais résultats et qu'elle est de moins en moins utilisée aujourd'hui.

2.1 Rendu en demi-ton par point

Cette famille de rendu en demi-ton est aussi appelée *dithering* [3], parce que la couleur de chaque pixel dans l'image reconstruite $I_c(x,y)$ est obtenue à partir seulement de l'intensité du pixel aux mêmes coordonnées dans l'image originale $I_i(x,y)$ et d'une valeur seuil s . Le pixel est rendu en noir si $I_i(x,y) < s$, et en blanc si $I_i(x,y) \geq s$.

Ici le seuil s joue un rôle très important pour la qualité de l'image reconstruite. s peut être une fonction prise dans un tableau de données, une simple constante ou une fonction quelconque. Cependant, les deux derniers types de seuil sont rarement utilisés. Le seuil constant donne peu de contrôle et son résultat n'est pas satisfaisant à l'œil. Avec une fonction qui évalue le seuil en ligne, le temps de traitement peut être très élevé. Ainsi la plupart des méthodes de rendu en demi-ton dans cette famille construisent un tableau de seuil et prennent le seuil correspondant selon les coordonnées du pixel. Le tableau de seuil s'appelle aussi la "fenêtre" de *dithering*.

2.1.1 Méthodes de base

Il y a deux anciennes méthodes de base dans la famille de rendu en demi-ton de type *dithering* : tramage de points regroupés et tramage de points dispersés. La fenêtre

de seuil est créée manuellement, au lieu d'être créée par pré-calcul comme les méthodes plus récentes. Les figures 2.1 et 2.2 montrent des fenêtres de seuil utilisées dans ces deux méthodes.

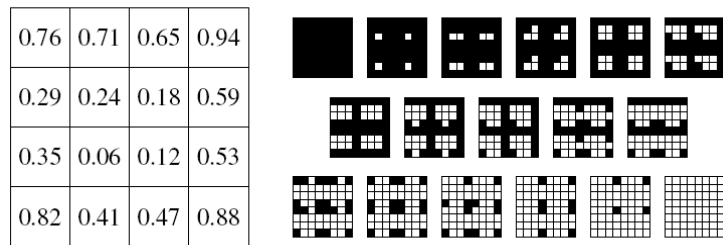


Figure 2.1 – La fenêtre de seuil pour une région carrée avec la méthode de tramage de points regroupés. Ici, les points sont rendus en blanc en ordre de proximité avec le centre, selon un patron de spirale anti-horaire. Des aplats de 8×8 pour chacun des 17 tons distinguables sont montrés à droite.

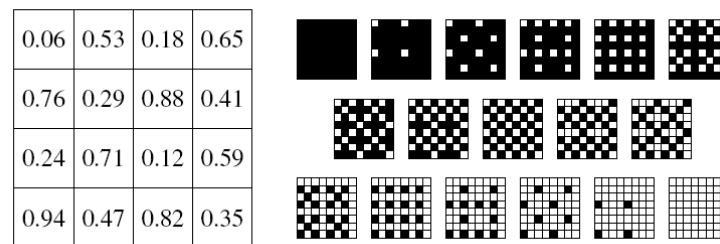


Figure 2.2 – La fenêtre de seuil pour une région carrée avec la méthode tramage de points dispersés. Ici, les points sont rendus en blanc selon le populaire patron de Bayer, de façon à minimiser les basses fréquences dans toute l'image quantifiée. Des aplats de 8×8 pour chacun des 17 tons distinguables sont montrés à droite.

Les figures 2.3 et 2.4 montrent des images reconstruites par ces deux méthodes. Parce que les fenêtres de seuil utilisées par ces deux méthodes sont normalement assez petites, typiquement 8×8 ou même 4×4 , les images reconstruites contiennent beaucoup de patrons réguliers. Ces patrons vont couvrir des signaux de hautes fréquences de l'image originale, et causer la perte d'information. Cette perte d'information cause de très mauvais résultats pour la méthode de tramage de points regroupés. Il y est même difficile d'identifier la personne dans l'image.

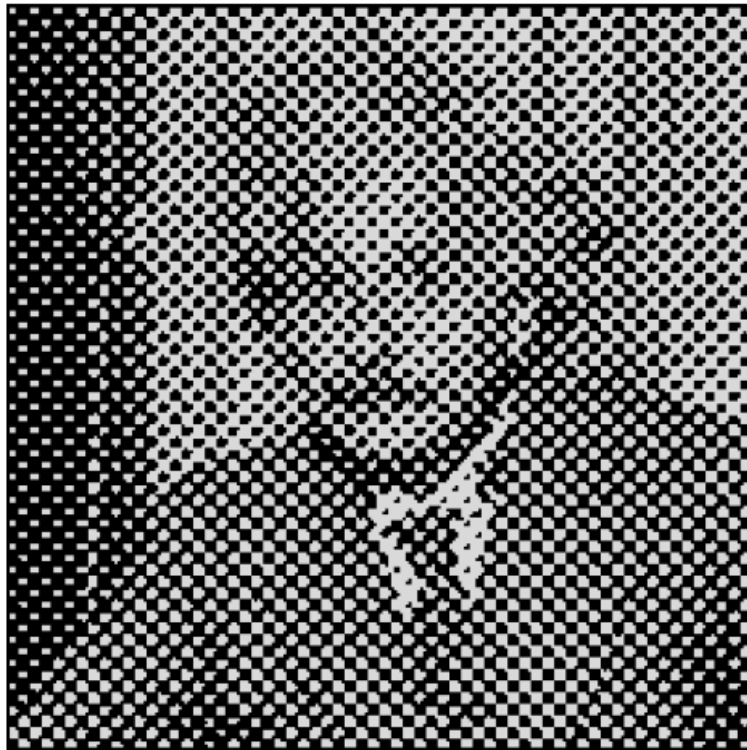


Figure 2.3 – Le résultat de la méthode de tramage de points regroupés. Notez que beaucoup de détails sont perdus parce que la fenêtre de seuil contient des signaux de basses fréquences.

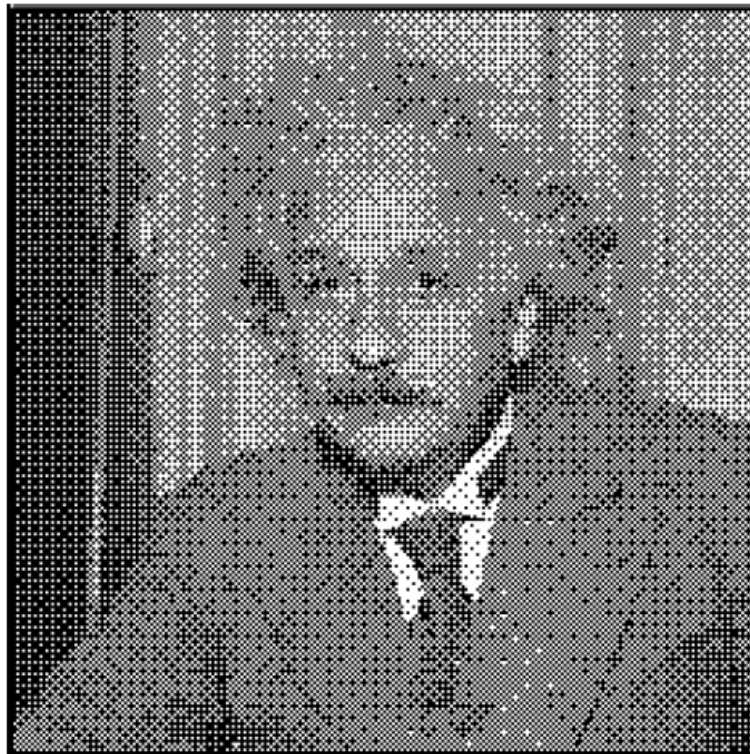


Figure 2.4 – Le résultat de la méthode de tramage de points dispersés. Notez qu'il y a moins de détails perdus sur cette image grâce à la fenêtre de seuil qui contient des signaux de hautes fréquences.

2.1.2 Méthodes avec des propriétés de bruit bleu

Les méthodes de tramage de points regroupés/dispersés vont créer des patrons réguliers sur l'image reconstruite. Ces patrons réguliers réduisent beaucoup la qualité de l'image parce que des détails sont détruits par ces patrons. Beaucoup de recherches se sont concentrées pour dériver des méthodes qui enlèvent ces patrons réguliers.

Dans [30], Ulichney a trouvé que des patrons réguliers proviennent des signaux de basses fréquences dans la fenêtre de seuil. Si la fenêtre de seuil ne contient que des signaux de hautes fréquences, alors le processus en demi-ton va ajouter du bruit de hautes fréquences dans l'image reconstruite. Ce bruit de hautes fréquences sera alors très difficile à visualiser par l'œil humain. Une fenêtre de seuil avec des propriétés de bruit bleu serait préférable. Ulichney a présenté une méthode pour construire une fenêtre de bruit bleu : la méthode *Void-and-Clusters*. Cette méthode commence par une image toute noire et la remplit en ajoutant des pixels blancs séquentiellement. Pour chaque pixel blanc ajouté, elle essaie de l'ajouter à l'endroit le plus loin possible des pixels blancs déjà ajoutés dans l'image. L'algorithme du disque de poisson est utilisé pour trouver cet endroit. Quand l'image est remplie avec des pixels blancs, la méthode utilise l'ordre dans lequel les pixels ont été ajoutés pour construire la fenêtre de seuil. Elle définit le seuil très proche de zéro pour le premier pixel ajouté, et le seuil très proche de un pour le dernier pixel ajouté. Voir la figure 2.5 pour la fenêtre de seuil de cette méthode.

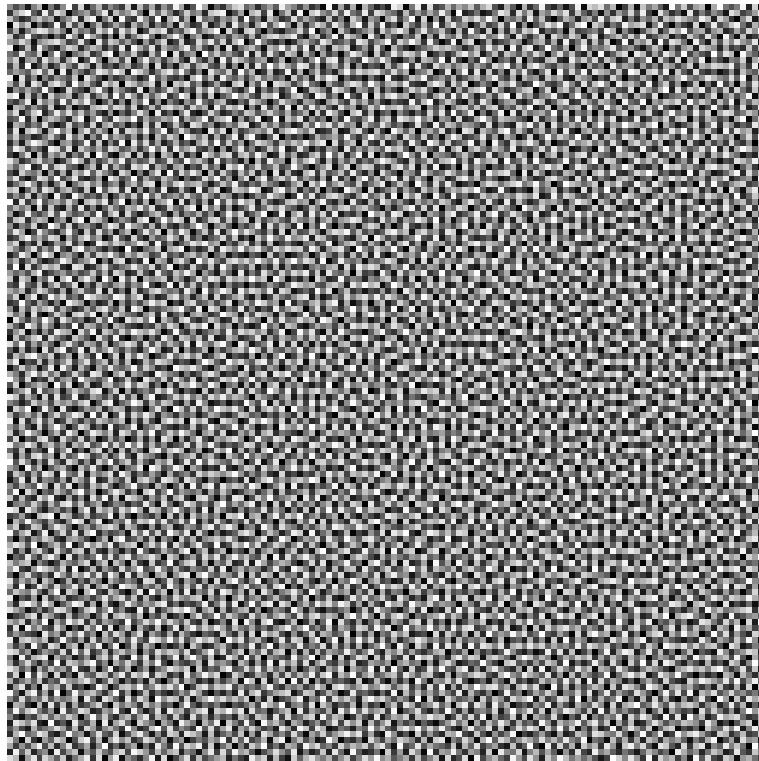


Figure 2.5 – La fenêtre de seuil de la méthode *Void-and-Clusters* pour la génération d'un tableau de *dither* [30].

Une autre méthode, *Blue Noise Mask*, présentée par Mitsa et Parker dans [18], construit la fenêtre de bruit bleu. Au lieu de construire l'image de distribution à partir d'aucun pixel blanc, cette méthode commence par une distribution de pixels blancs remplie à la moitié de l'image. Cette distribution de pixels blancs est obtenue par un autre algorithme, par exemple par le disque de poisson pour assurer qu'elle ne contient que du bruit bleu. Pour obtenir des autres distributions de pixels blancs, cette méthode essaie d'ajouter ou d'enlever des pixels blancs de l'image et compare le spectre de fourier avant et après avoir ajouté/enlevé ces pixels. Si le spectre conserve les propriétés de bruit bleu, ces nouveaux pixels blancs sont gardés et cette nouvelle distribution est utilisée pour trouver d'autres distributions de pixels blancs. La figure 2.6 montre la fenêtre de seuil de cette méthode.

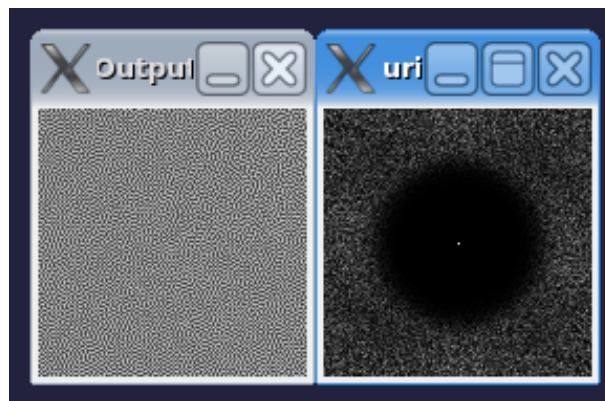


Figure 2.6 – La fenêtre de seuil de la méthode *Blue Noise Mask* [18] et son spectre de fourier.

La figure 2.7 montre une image reconstruite par cette méthode. Nous pouvons voir qu'il n'y a pas de patrons réguliers, parce que le bruit ajouté est de type bruit bleu. Cependant, la qualité de l'image reconstruite n'est pas assez satisfaisante à l'œil. Beaucoup de fins détails ne sont pas assez visibles par rapport à l'image originale. Parce que cette famille de rendus en demi-ton ne considère que le pixel courant, des informations entre le pixel courant et ses voisins, tel le gradient, ne peuvent pas être utilisées par cette famille de méthodes.

En conclusion, le temps de traitement des méthodes de cette famille est très rapide,

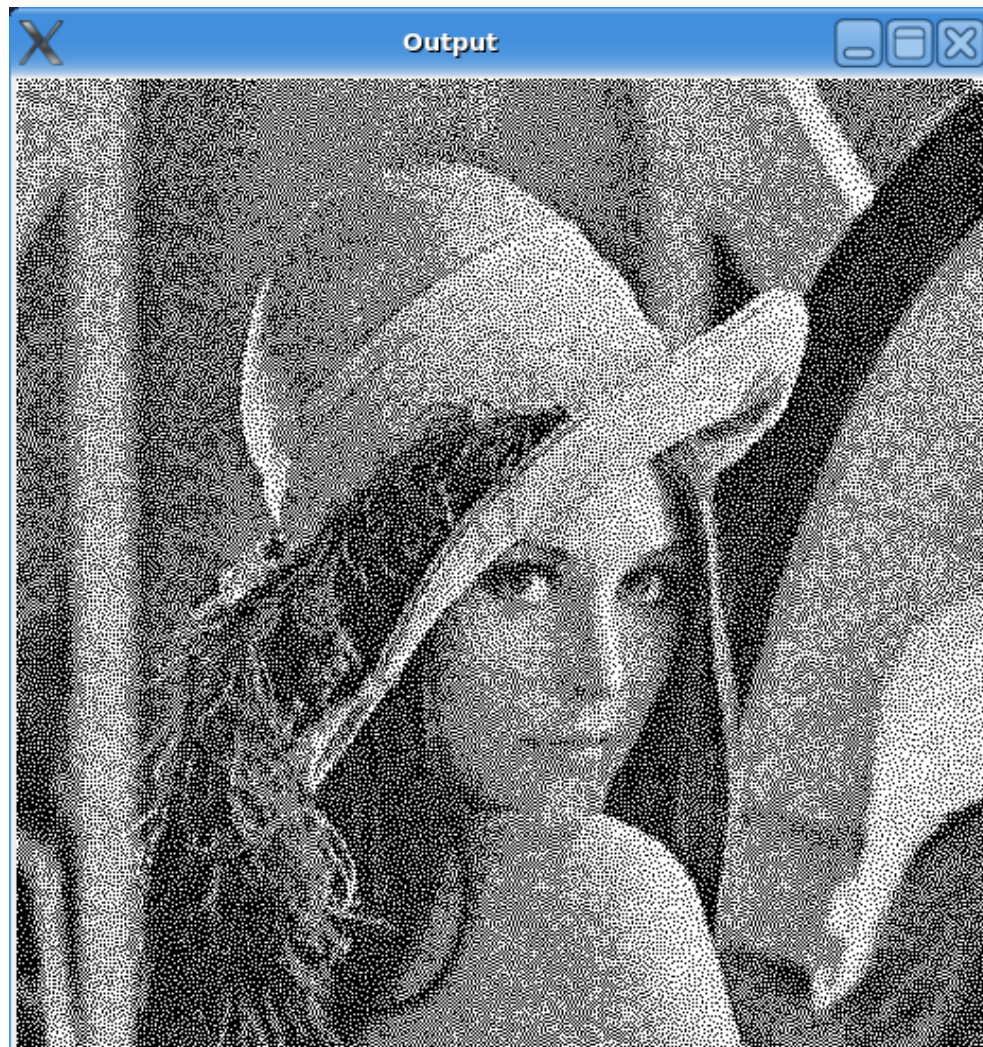


Figure 2.7 – Résultat du *Blue Noise Mask* [18]. Notez que la distribution des pixels suit un bruit bleu, donc sans patrons réguliers.

parce qu'elles traitent seulement le pixel courant. Cependant elles produisent des résultats de qualité, plutôt mauvaise.

2.2 Rendu en demi-ton par voisinage

L'avantage le plus intéressant des méthodes de la famille de rendus en demi-ton par point est sa vitesse. Mais en même temps, la qualité des images reconstruites par ces méthodes est mauvaise. Voici ses deux plus grands défauts :

- Des pertes d'information, parce que les méthodes de cette famille considèrent seulement la couleur du pixel courant. Sur l'image reconstruite, les couleurs des pixels sont définies séparément. La distribution de ces pixels (points blancs) ne montre donc pas des informations locales de l'image originale, comme de fins détails ou de fines textures. En plus, comme le processus de rendu en demi-ton réduit le nombre de couleurs de l'image reconstruite, il va ajouter du bruit de hautes fréquences sur l'image reconstruite. Parce que les méthodes de cette famille ne peuvent pas reconstruire l'image avec des propriétés de bruit bleu (certaines méthodes essaient de reconstruire l'image avec du bruit bleu, par exemple la méthode *Blue Noise Mask*, mais leurs effets sont très faibles par rapport aux deux autres familles de rendus en demi-ton), des détails de l'image originale sont recouverts par du bruit de hautes fréquences créé par le processus de rendu en demi-ton.
- Des patrons réguliers ou du bruit indésirable. Ce défaut peut être réduit par l'augmentation de la taille de la fenêtre de seuil, mais ils persistent toujours quand la taille de la fenêtre de seuil est plus petite que la taille de l'image originale.

Ces deux défauts apparaissent sur toutes les images dans la section précédente. Ces deux défauts sont beaucoup réduits par des méthodes de la famille de rendu en demi-ton par voisinage. Cette famille de rendu en demi-ton repose sur la méthode diffusion d'erreur et l'ensemble de ses améliorations.

2.2.1 Diffusion d'erreur

La diffusion d'erreur est une méthode qui utilise l'erreur d'un pixel pour définir les couleurs au voisinage de ce pixel. Supposons que l'image en entrée est I_o , et l'image en

sortie est I_c . $I_o(i, j)$ et $I_c(i, j)$ définissent le pixel de coordonnées i, j dans l'image originale et l'image reconstruite. Le processus de la méthode diffusion d'erreur est composé de trois étapes :

- (a) la couleur du pixel $I_c(i, j)$ est définie par une fonction de seuil $s(i, j)$ qui prend en argument la sommation des intensités du pixel $I_o(i, j)$ et des erreurs diffusées par les pixels déjà parcourus en entrée. Cette fonction retourne une couleur, noir ou blanc, en sortie.
- (b) le calcul de l'erreur e est la différence entre l'intensité du pixel $I_o(i, j)$ et l'intensité du pixel $I_c(i, j)$.
- (c) la diffusion de l'erreur e s'effectue vers le voisinage du pixel, c'est-à-dire les pixels qui n'ont pas encore été parcourus.

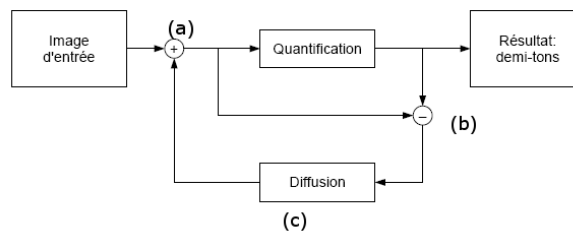


Figure 2.8 – Modèle de la méthode de Floyd et Steinberg de diffusion d'erreur [8].

La figure 2.8 montre le schéma de l'algorithme en détail. Cette méthode a été présentée par Floyd et Steinberg en 1975 [8]. A l'époque, ils ont utilisé une fonction de seuillage très simple, soit une valeur constante de 0.5, et ils ont utilisé les huit plus proches voisins pour diffuser l'erreur du pixel courant.

Les résultats à la section 4.1 montrent des résultats de la diffusion d'erreur de Floyd et Steinberg. Nous pouvons voir que la diffusion d'erreur produit un résultat incomparablement meilleur que l'ensemble des méthodes de la famille par point. Les détails et les fines textures sont préservés, et les points noirs et blancs sont assez irréguliers pour ne pas créer de motifs. Mais en même temps, la diffusion d'erreur a trois défauts majeurs :

- Dans les zones très sombres ou très claires (l'intensité proche de zéro ou de un), cette méthode reproduit des structures en serpent.
- Les détails fins sont préservés par rapport aux méthodes de la famille par point. mais il existe toujours des informations de l'image originale qui sont perdues.
- Pour certaines intensités ($\frac{1}{2}$, $\frac{1}{3}$, etc.), cette méthode produit des structures répétitives. Ces structures répétitives sont parfois très visibles et forment des "lignes virtuelles" sur l'image reconstruite, de telle sorte que l'image reconstruite n'est pas pareille à l'image originale. On appelle ces intensités des intensités de clé. Des aplats qui contiennent seulement une de ces intensités sont très difficiles à rendre par cette méthode.

Parce que la méthode diffusion d'erreur est une méthode de la famille par voisinage, sa vitesse dépend du nombre de voisins diffusés. Normalement, elle est un peu plus lente qu'une méthode de la famille par point.

Beaucoup de recherches se sont concentrées sur l'amélioration des trois défauts majeurs de cette méthode. Dans les prochaines sections, nous allons les présenter selon trois classes : l'amélioration de l'ordre de parcours, l'amélioration du seuillage, et l'amélioration des coefficients de diffusion.

2.2.2 Amélioration de l'ordre de parcours

La diffusion d'erreur de Floyd-Steinberg suit un ordre *raster* pour parcourir les pixels. Pour chaque ligne, l'algorithme traite toujours le pixel le plus gauche du pixel le plus droite (figure 2.9). Cet ordre de parcours va créer des très fortes structures en serpent (*worm-like*) si l'image originale contient une grande zone de même intensité à cause de sa nature asymétrique. Pour réduire cet effet, plusieurs recherches ont essayé d'autres ordres de parcours :

2.2.2.1 L'ordre de parcours en serpent

L'ordre de parcours en serpent est peut-être le premier ordre de parcours non standard, introduit en 1982 par Witten et Neal [32]. Dans cette configuration de l'ordre de parcours, les pixels sont traités de gauche à droite pour une ligne impaire, et de droite à gauche pour les pixels d'une ligne paire (voir figure 2.10). L'avantage de cet ordre de parcours est qu'il réduit les structures en serpent et qu'il n'utilise pas de mémoire supplémentaire pour enregistrer le chemin. Cependant il faut utiliser un autre ensemble de coefficients de diffusion pour les lignes paires, qui est l'ensemble miroir de coefficients de diffusion. En utilisant cet ordre de parcours, les structures en serpent sont beaucoup réduites. Nous pouvons malgré tout voir encore des motifs verticaux puisque l'ordre de parcours n'est pas symétrique verticalement.

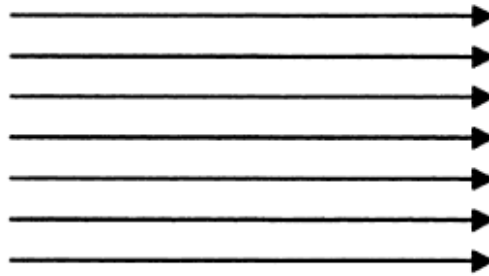


Figure 2.9 – Ordre de parcours "raster" utilisé par Floyd et Steinberg [8].

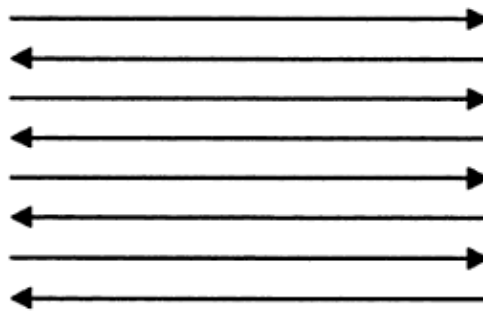


Figure 2.10 – Ordre de parcours en serpentin proposé par Witten et Neal [32].

2.2.2.2 Courbe de remplissage

En 1991, Velho et Gomez ont présenté une nouvelle famille d'ordres de parcours pour rendre l'image en demi-ton. *Space Filling Curves* est un ordre de parcours beaucoup plus compliqué que *raster* et en serpent. Il s'agit d'une courbe continue qui traverse l'espace unité $I^2 = [0, 1] \times [0, 1]$. Soit t un nombre réel dans l'intervalle I . Le courbe de remplissage permet de visiter tous les pixels dans I^2 avec t entre 0 et 1.

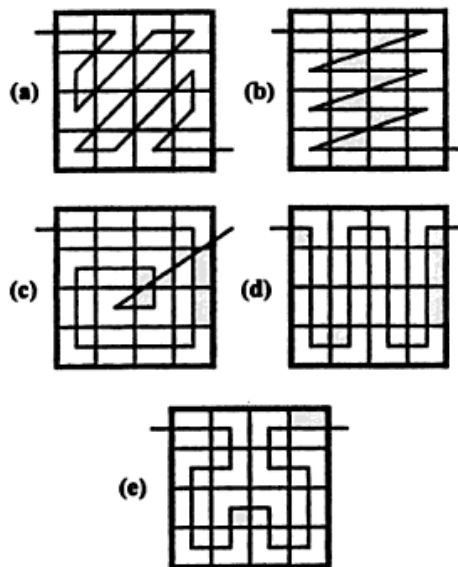


Figure 2.11 – Ordre de parcours courbe de remplissage avec quelques exemples de chemins parcourus présentés dans [11].

La figure 2.11 montre quelques exemples de courbes de remplissage. Cet ordre de parcours n'est pas très populaire parce qu'il faut une grande quantité de mémoire pour enregistrer le chemin de parcours, qui est normalement de la taille de l'image construite. En raison de la complexité de la fonction de la courbe de remplissage, cette méthode va introduire des structures internes qui vont réduire la netteté de l'image et briser des détails fins.

2.2.3 Amélioration du seuillage

L'algorithme de diffusion d'erreur de Floyd et Steinberg utilise un seuil global et constant pour calculer l'erreur, soit en pratique une valeur de 0.5. Beaucoup de recherches se sont concentrées pour trouver un meilleur seuil selon des propriétés de l'image originale.

2.2.3.1 Rehaussement de bordures [6]

Dans cette méthode, l'inversion de l'image originale I_i est utilisée comme une carte de seuil. L'idée principale est quand un pixel va être rendu comme un pixel blanc sur l'image reconstruite, si on prend un seuil plus petit que 0.5, il va apparaître plus "blanc" sur l'image reconstruite. Pour un pixel qui va être rendu noir sur l'image reconstruite, il va apparaître plus "noir" si on prend un seuil plus grand que 0.5. Donc, si on prend l'image inversée comme carte de seuil, des détails sur l'image reconstruite vont être plus visibles parce que des signaux de hautes fréquences de l'image originale seront augmentés. Eschbach et Knox ont aussi défini un coefficient global k qui détermine le poids à utiliser pour l'image inversée. La figure 2.12 montre un résultat de cette méthode avec valeurs de k différentes.

Cette méthode ouvre la voie pour modifier des seuils de la diffusion d'erreur afin d'augmenter la visibilité de détails fins dans l'image originale. Cependant, cette méthode a deux défauts majeurs. Premièrement, elle a seulement un coefficient global. Elle peut donc globalement contrôler la visibilité des détails de l'image mais n'offre pas de contrôle local. Deuxièmement, certaines images sont plus difficiles à rendre que les autres. Il faut donc prendre une valeur de k plus grande pour ces images. Cependant cette méthode ne donne pas la possibilité de choisir la valeur de k automatiquement pour des images différentes.

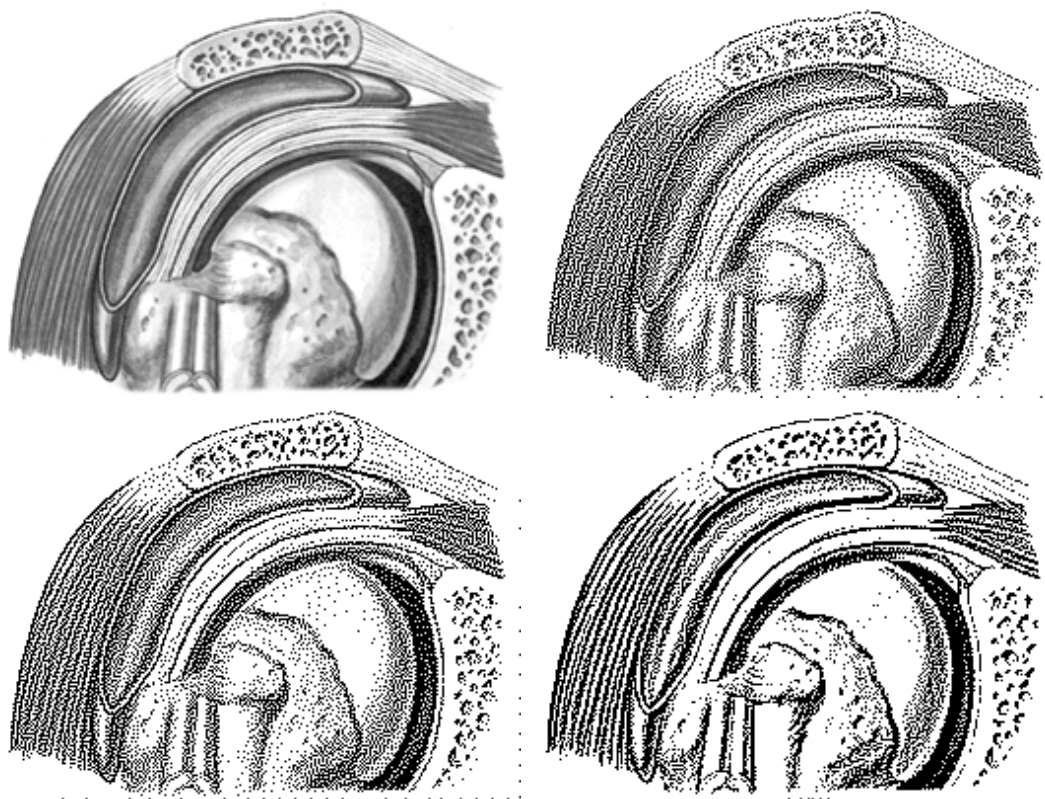


Figure 2.12 – Méthode proposée par Eschbach et Knox [6]. (a) Image originale. (b) $k = 0$. (c) $k = -2$. (d) $k = -5$. Cette image est aussi rendue avec d'autres méthodes à la figure 4.7.

2.2.3.2 Rehaussement de bordures amélioré selon un *first-order gradient shaping filter*

En 2004, Hwang et al. [9] ont publié une autre amélioration de la diffusion d'erreur basée sur le seuil. Leur méthode mesure les gradients des pixels de l'image originale et crée une image modifiée à partir de ces gradients. Cette image modifiée est ensuite utilisée comme l'image de seuil. Cette méthode peut être considérée comme une amélioration de la méthode de Eschbach et Knox. La figure 2.13 montre le schéma de l'algorithme de la méthode.

Hwang et al. ont utilisé une mesure pour améliorer les détails fins dans l'image originale : le gradient local. Ils ajoutent le gradient local dans la production du filtre FGSOI (*first-order gradient shaping of original image*) afin d'obtenir une meilleure distribution de points le long des textures de l'image originale. La figure 2.14 montre des résultats de cette méthode.

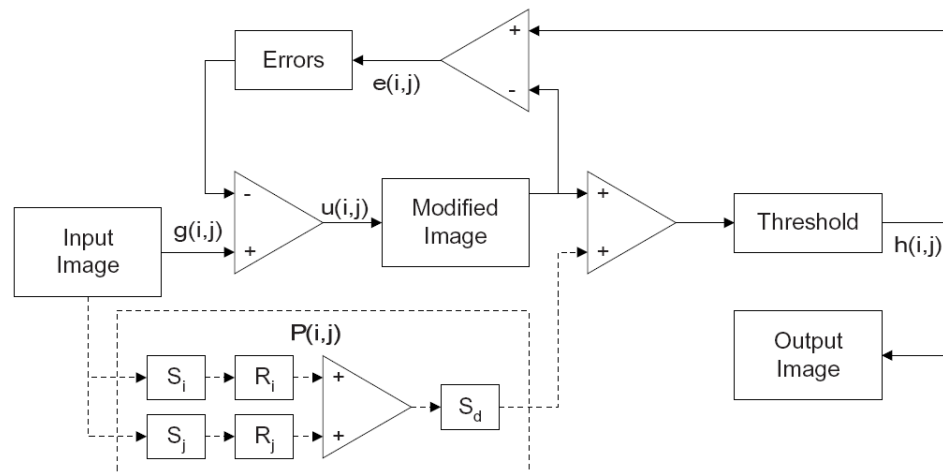


Figure 2.13 – Méthode proposée par Hwang et al. [9].

En conclusion, cette méthode considère à la fois l'intensité locale et la variation d'intensités locale. Cependant, cette amélioration de la diffusion d'erreur contient un défaut : la visibilité des détails est contrôlée par un seul paramètre global. Par conséquent, cette méthode est sensible à une sous-zone particulière dans l'image originale, expliqué dans [21]. Cette méthode va échouer quand l'image originale est composée de beaucoup de

contenus fréquences/contrastes.

2.2.3.3 Rehaussement de bordures selon de propriétés de la vision humaine

En 2006, Kwak et al. ont publié un article [13] afin d'améliorer la méthode de Hwang et al. [9]. Dans cette méthode, on calcule une mesure locale qui est la différence entre l'intensité du pixel et l'application d'une fenêtre 3×3 de moyennage de l'intensité du voisinage pondérées par leurs positions spatiales. La méthode utilise aussi le IEE (*information of edge enhancement*), qui est calculé par la mesure des activités spatiales normalisées multipliées par l'intensité locale. Le système visuel humain est aussi considéré dans la méthode. Elle réduit certains coefficients du filtre parce que l'œil humain est plus sensible aux axes horizontal et vertical.

La figure 2.14 compare des résultats de cette méthode et des méthodes [8] et [9]. Nous pouvons constater que cette méthode préserve plus de détails que les deux autres méthodes. Des pixels noirs et blancs sont distribués selon l'orientation des textures et des bordures. Cependant, cette méthode a un autre défaut : en présence de structures répétitives dans l'image originale, elle va introduire des moirés indésirables. Ces moirés sont clairement visibles dans les figures 4.12, et cela peu importe que l'amplitude des coefficients soit forte ou faible. Selon nos tests, les méthodes [6] et [9] ont aussi ce défaut.



Figure 2.14 – La comparaison des méthodes d’amélioration du seuillage. (a) Image originale. (b) L’image reconstruite par Eschbach et Knox [6]. (c) L’image reconstruite par Hwang et al. [9]. (d) L’image reconstruite par Kwak et al. [13].

2.2.4 Amélioration des coefficients de diffusion

Les coefficients de diffusion explorés par Floyd et Steinberg dans [8] ont été utilisés dans les méthodes de diffusion d'erreur pendant très longtemps, parce qu'ils reproduisent des résultats assez bons. Cependant, à cause des problèmes décrits dans la section 2.2.1, beaucoup de recherches se sont concentrées sur l'amélioration de ces coefficients afin de trouver une méthode pour éviter d'introduire des structures en serpentins et pour reproduire une distribution de points plus irrégulière pour les intensités de clé.

2.2.4.1 *Digital Halftoning*

Dans cette recherche, Ulichney [29] a essayé d'améliorer les coefficients de diffusion de plusieurs façons : changer le nombre de coefficients, mettre un ou plusieurs coefficients aléatoires, etc. Il a conclu que :

- Avec un seul coefficient, c'est difficile d'avoir une distribution de points irrégulière si la position du coefficient est fixée. Si la position du coefficient peut être changée, le résultat devient de plus en plus proche à la méthode de Floyd et Steinberg [8] si on augmente la taille du filtre (nombre de voisins).
- Avec deux coefficients, en utilisant le serpentins comme ordre de parcours, les patrons directionnels sont beaucoup réduits, et ces patrons réguliers peuvent être encore beaucoup plus réduits en utilisant 100% d'aléatoire à ces deux coefficients.
- Aucune combinaison de quatre coefficients ne peut produire des résultats meilleurs que la méthode de Floyd et Steinberg [8].

Dans sa conclusion, les coefficients de Floyd et Steinberg avec 50% d'aléatoire et l'ordre de parcours en serpentins donne les meilleurs résultats.

2.2.4.2 **Modification statique des coefficients de diffusion**

Beaucoup de recherches sont basées sur la modification de la valeur des coefficients de Floyd et Steinberg afin de réduire les structures en serpentins et les patrons réguliers sur

les intensités de clé. Certaines méthodes essaient d'ajouter des coefficients supplémentaires à l'endroit plus gauche du filtre [25, 26], d'autres essaient d'augmenter la taille du filtre pour produire un résultat plus lisse [16, 24]. Le tableau 2.I montre les coefficients de ces méthodes.

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| | | | X | $\frac{8}{16}$ |
| $\frac{1}{16}$ | $\frac{1}{16}$ | $\frac{2}{16}$ | $\frac{4}{16}$ | |

(a)

| | | | |
|--------|--------|-------|-------|
| | | X | 0.292 |
| 0.0248 | 0.3772 | 0.275 | |

(b)

| | | | | |
|----------------|----------------|----------------|------------------|------------------|
| | | X | $\frac{7}{24}$ | $\frac{2}{24}$ |
| $\frac{1}{24}$ | $\frac{3}{24}$ | $\frac{5}{24}$ | $\frac{1}{24}$ | $\frac{0.5}{24}$ |
| $\frac{1}{24}$ | $\frac{1}{24}$ | $\frac{2}{24}$ | $\frac{0.5}{24}$ | |

(c)

| | | | |
|--------|--------|--------|--------|
| | | X | 0.2911 |
| 0.1373 | 0.3457 | 0.2258 | |

(d)

| | | | |
|----------------|----------------|----------------|----------------|
| | | X | $\frac{7}{16}$ |
| $\frac{1}{16}$ | $\frac{3}{16}$ | $\frac{5}{16}$ | |

(e)

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| | | X | $\frac{8}{42}$ | $\frac{4}{42}$ |
| $\frac{2}{42}$ | $\frac{4}{42}$ | $\frac{8}{42}$ | $\frac{4}{42}$ | $\frac{2}{42}$ |
| $\frac{1}{42}$ | $\frac{2}{42}$ | $\frac{4}{42}$ | $\frac{2}{42}$ | $\frac{1}{42}$ |

(f)

Tableau 2.I – Comparaison des coefficients de diffusion : (a) [25], (b) [24], (c) [11], (d) [33], (e) [7], (f) [28]. La lettre X montre la position du pixel courant.

Ces méthodes améliorent la qualité de l'image reconstruite, mais elles ne peuvent pas complètement régler les problèmes principaux de l'algorithme de diffusion d'erreur.

2.2.4.3 Un algorithme simple et efficace de diffusion d'erreur

En 2001, Ostromoukhov a publié une autre méthode [20] pour améliorer les coefficients de la diffusion d'erreur standard. Au lieu d'utiliser un seul groupe de coefficients de diffusion sur toute l'image, cette méthode utilise une groupe de diffusion particulier pour chacune des intensités (0 à 255). Son objectif est de bien sélectionner les coefficients de diffusion pour chacune des intensités afin d'obtenir de meilleures distributions de points pour des aplats pour n'importe quelle intensité, et qui ne créent pas des structures répétitives sur les intensités de clé. La figure 2.15 montre une comparaison visuelle et son spectre de fourier entre cette méthode et la méthode de Floyd et Steinberg sur un aplat.

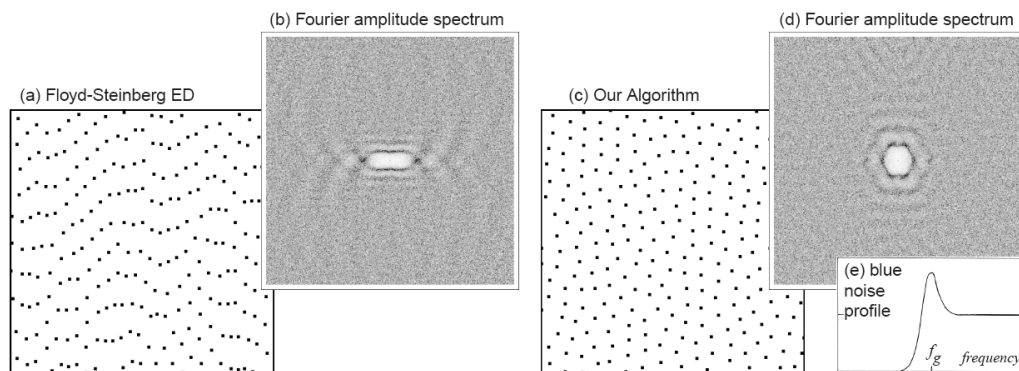


Figure 2.15 – La comparaison de la méthode de Floyd et Steinberg [8] et de Ostromoukhov [20] sur un aplat d'intensité constante. Gauche : la méthode de Floyd et Steinberg et son spectre de fourier. Droite : la méthode de Ostromoukhov et son spectre de fourier.

Le tableau de groupes de coefficients est créé en précalcul. Les coefficients des intensités plus difficiles à rendre (les intensités de clé) sont trouvés tout d'abord. Puis, pour avoir une belle progression de la distribution des points entre les intensités de clé, et pour ne pas créer de "lignes virtuelles" entre les intensités, des coefficients pour les intensités intermédiaires sont calculés par interpolation. Le critère utilisé pour décider

si un groupe de coefficients est bon, correspond à un compromis subjectif entre l'absence d'artefacts visuellement apparents et la qualité du profil de bruit bleu du spectre de fourier des distributions.

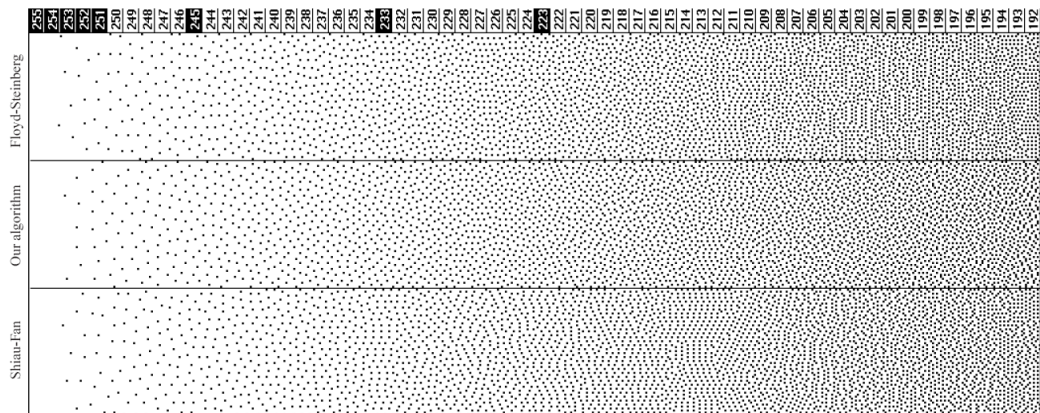


Figure 2.16 – La comparaison des méthodes de Floyd et Steinberg [8], de Ostromoukhov [20] et de Shiau et Fan [25]. La distribution de points est supérieure pour la méthode de Ostromoukhov, et elle contient moins de structures répétitives. Les valeurs d'intensité en noir (en haut de la figure) présentent les intensités de clé utilisées pour la construction des coefficients dans la méthode de Ostromoukhov.

Sur la figure 2.16, nous pouvons voir que cette méthode donne une meilleure distribution de points : les structures répétitives sur les intensités de clé sont beaucoup moins visibles par rapport aux autres méthodes. Grâce à l'utilisation de l'ordre de parcours en serpent, elle ne produit presque pas de structures en serpent.

En 2003, une amélioration de cette méthode a été publiée par Zhou et al. [35], où ils ont ajouté un peu de bruit blanc sur le seuil pour chaque intensité afin de réduire encore plus les structures répétitives sur les intensités de clé. Ils ont aussi introduit une façon automatique pour trouver les coefficients de chaque intensité.

En conclusion, ces deux méthodes ont réussi à résoudre deux problèmes principaux de la diffusion d'erreur : des structures en serpent, et des structures répétitives sur les intensités de clé. Cependant, des détails fins ne sont toujours pas assez visibles dans l'image reconstruite par rapport à ceux dans l'image originale.

2.3 Rendu en demi-ton par optimisation globale

Dans cette section, deux méthodes de la famille de rendus en demi-ton par optimisation globale sont présentées. Ces deux méthodes définissent l'intensité de chaque pixel selon des propriétés de l'image originale en entier.

2.3.1 Recherche binaire directe

Cette méthode est basée sur une optimisation sensible aux tons, développée par trois groupes de recherche au cours de la même année [2, 19, 23]. La méthode commence par prendre une image initiale qui contient le même nombre de pixels blancs que l'image originale. Ainsi cette image respecte le ton global de l'image originale. Ensuite elle change un ou deux pixels à la fois seulement lorsque ce changement améliore la qualité de l'image reconstruite selon un métrique qui décrit la similarité entre l'image en demi-ton et l'image originale.

$$\mu_{I_o}(x, y) = \sum_i \sum_j I_o[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (2.1)$$

$$\mu_{I_r}(x, y) = \sum_i \sum_j I_r[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (2.2)$$

$$E = \int \int \|\mu_{I_o} - \mu_{I_r}\|^2 dx dy \quad (2.3)$$

L'équation ci-dessus décrit la métrique utilisée par cette méthode, qui mesure la somme des différences du ton local entre l'image originale et l'image reconstruite, où I_o et I_r sont les images originale et reconstruite respectivement, ψ est un filtre gaussien utilisé pour pondérer l'équation, τ_x et τ_y correspondent à l'espacement le long des axes X et Y de la grille digitale de pixels.

La méthode parcourt l'image pixel par pixel, séquentiellement. Pour chaque pixel, elle essaie de changer son intensité avec un de ses huit voisins les plus proches, seulement si ce changement réduit la valeur de la métrique. Si ce changement est accepté, cette nouvelle distribution des pixels est conservée. La méthode évalue l'image recons-

truite plusieurs fois jusqu'à ce que aucun changement ne puisse être fait sur l'image.

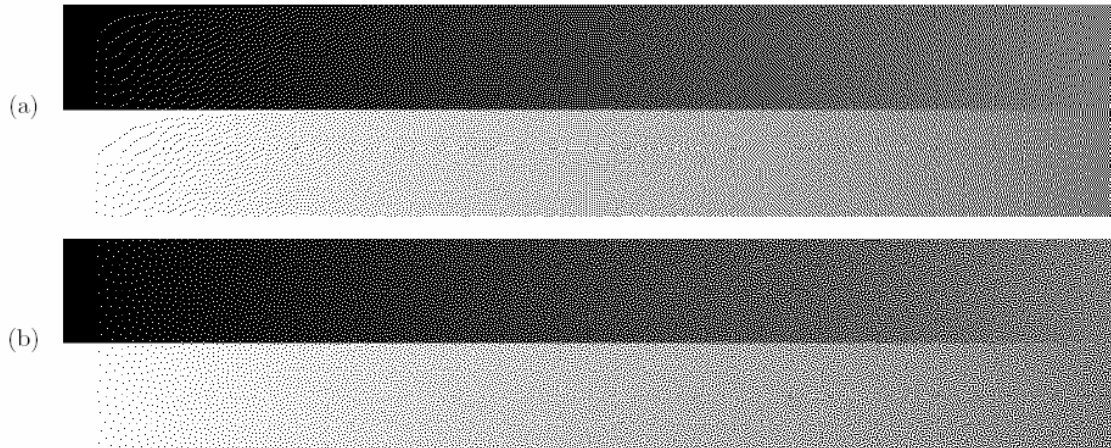


Figure 2.17 – La comparaison entre la méthode de Floyd et Steinberg et de la recherche binaire directe sur une plage d'intensités variant entre zéro et un. (a) Image reconstruite par Floyd et Steinberg [8]. (b) Image reconstruite par Analoui et Allebach [2].

La figure 2.17 montre une comparaison de cette méthode avec la méthode de Floyd et Steinberg sur une plage d'intensités variant entre zéro et un. Nous pouvons voir que la méthode de recherche binaire directe ne produit pas de structures en serpentins sur les zones claires et foncées. En plus, elle ne crée pas des patrons réguliers sur des intensités de clé.

Mais elle a aussi quelques problèmes. Premièrement, le choix de l'image initiale joue un rôle très important pour obtenir un bon résultat final. Si elle contient beaucoup de bruit blanc, la méthode converge très lentement. Par contre, si elle se trouve proche d'un minimum local de la métrique, il est impossible de trouver le minimum global parce qu'on change seulement une paire de pixels à la fois. Deuxièmement, parce qu'il faut recalculer la métrique chaque fois que la méthode change une paire de pixels, le temps de traitement est plus long par rapport aux méthodes des familles par point et par voisinage. Finalement, la métrique utilisée dans cette méthode ne prend pas en compte l'information de structure. Des détails fins ne sont pas conservés.

2.3.2 Structure-aware halftoning

La méthode de recherche binaire directe a résolu deux des trois défauts majeurs de la méthode de Floyd et Steinberg : les structures en serpentins et les patrons réguliers sur les intensités de clé. Cependant, des détails fins ne sont toujours pas préservés dans l'image reconstruite. En 2008, Pang et al. ont publié un nouvel article [21] dans la famille par optimisation globale, qui prend en compte non seulement le ton local dans sa métrique, mais aussi des informations structurelles locales.

Leur métrique est composée de deux termes : l'un sensible au ton local, comme la métrique utilisée dans la recherche binaire directe, et l'autre est une métrique appelée *Mean Structural Similarity Index Measure (MSSIM)* introduite par Wang et al. [31], qui mesure la similarité de structure entre deux images.

Le premier terme correspond à la moyenne des carrés des différences entre l'image originale et l'image en demi-ton filtrée par une fonction gaussienne. Idéalement, une gaussienne avec une valeur de σ très petite est utilisée, mais parce que l'image en demi-ton contient beaucoup de signaux de hautes fréquences, un σ de valeur moyenne est utilisée dans cette étape. Dans [21], une valeur $\sigma = 1.5$ est utilisée.

Le deuxième terme est une composition de trois sous-termes : l'intensité, le contraste et la structure locale. Soient I_o et I_r les images originale et en demi-ton respectivement. Alors :

$$\mu_o(x, y) = \sum_i \sum_j I_o[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (2.4)$$

$$\mu_r(x, y) = \sum_i \sum_j I_r[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (2.5)$$

$$\sigma_o^2(x, y) = \sum_i \sum_j (I_o[i, j] - \mu_o[i, j])^2 \psi(x - i\tau_x, y - j\tau_y) \quad (2.6)$$

$$\sigma_r^2(x, y) = \sum_i \sum_j (I_r[i, j] - \mu_r[i, j])^2 \psi(x - i\tau_x, y - j\tau_y) \quad (2.7)$$

$$\sigma_{or}^2(x,y) = \sum_i \sum_j (I_o[i,j] - \mu_o[i,j])(I_r[i,j] - \mu_r[i,j])\psi(x - i\tau_x, y - j\tau_y) \quad (2.8)$$

où ψ est un filtre gaussien de variance égale à 1.67, et τ_x et τ_y correspondent à l'espacement le long des axes X et Y de la grille de pixels. Dans l'article, une fenêtre de taille 11×11 est utilisée.

Selon la loi de Weber, le système visuel humain est plus sensible à la luminance relative plutôt que la luminance absolue. La composante luminance est obtenue par :

$$l(x,y) = \frac{2\mu_o(x,y)\mu_r(x,y) + k_1}{\mu_o^2(x,y) + \mu_r^2(x,y) + k_1} \quad (2.9)$$

La composante contraste est obtenue par une formulation similaire à la composante luminance :

$$c(x,y) = \frac{2\sigma_o(x,y)\sigma_r(x,y) + k_2}{\sigma_o^2(x,y) + \sigma_r^2(x,y) + k_2} \quad (2.10)$$

La composante structure est calculée avec l'aide de la corrélation entre l'image originale et l'image reconstruite :

$$s(x,y) = \frac{\sigma_{or}(x,y) + k_3}{\sigma_o(x,y)\sigma_r(x,y) + k_3} \quad (2.11)$$

La métrique M est calculée avec deux termes :

$$M = \omega E + (1 - \omega) \int \int l(x,y)c(x,y)s(x,y)dx dy \quad (2.12)$$

où E est le terme de sensibilité au ton local, et ω est un poids qui définit l'importance du premier terme par rapport à la sensibilité aux structures. Dans l'article, une valeur $\omega = 0.5$ est utilisée. Les trois constantes k_1, k_2, k_3 servent à éviter les singularités ; elles peuvent prendre n'importe quelles valeurs qui sont beaucoup plus petites que μ et σ .

Comme pour la méthode de recherche binaire directe, la méthode débute avec une image en demi-ton initiale qui a l'intensité globale égale à celle de l'image originale.

Puis la méthode essaie de changer quelques paires de pixels et évalue la métrique. Si le changement réduit la métrique, les paires de changements sont conservées. Au lieu de traiter les pixels séquentiellement, des paires aléatoires sont choisies afin d'éviter des structures symétriques. Pour être capable de converger vers le minimum global, un algorithme de recuit-simulé est utilisé [14].

Des images à la section 4.1 montrent la comparaison de la méthode avec la méthode de Ostromoukhov [20]. Dans l'image reconstruite par cette méthode, les trois défauts majeurs de la méthode de Floyd et Steinberg sont complètement réglés : il n'y a plus de structures en serpentins ; il n'y a pas de patrons réguliers sur les intensités de clé ; les détails fins sont préservés. L'image reconstruite par cette méthode peut être considérée comme le "*ground truth*" de l'image en demi-ton.

Cependant, il y a un défaut important avec cette méthode. Due à la complexité de calcul de la métrique et l'utilisation d'un algorithme de recuit-simulé, son temps de traitement est beaucoup plus élevé que les méthodes des deux autres familles. Au tableau 4.III, nous pouvons voir que son temps de traitement est une centaine de fois plus élevé qu'une méthode de la famille par voisinage. Même si cette méthode produit de très belles images, elle ne s'adapte pas aux conditions dans le contexte de l'impression.

CHAPITRE 3

LE RENDU EN DEMI-TON AVEC SENSIBILITÉ À LA STRUCTURE

Dans ce chapitre nous allons présenter la méthode proposée : *Structure-Aware Halftoning*. Elle est parue dans un article [5].

Dans la méthode proposée, la valeur de seuil constante sur toute l'image est remplacée par une image de seuils construite en appliquant un ensemble de filtres de Gabor en fonction du contenu fréquentiel local de l'image. Nous utilisons des filtres gaussiens pour trouver des coefficients de diffusion adaptés aux structures. Un nouveau paramètre de calibration ω contrôle la progression entre des coefficients de diffusion standard et des coefficients gaussiens.

Des paramètres de calibration sont définis dans une étape de précalcul pour chaque combinaison possible de contenu fréquentiel. Nous avons choisi six valeurs pour chaque composant (l'orientation, la fréquence et le contraste) du contenu fréquentiel, c'est-à-dire 216 combinaisons possibles. Un tableau de recherche est créé en fonction de ces trois composants. Durant la reconstruction de l'image, le contenu fréquentiel de chaque pixel est utilisé comme indice du tableau de recherche de calibration. Les valeurs du tableau sont utilisées pour déterminer le bon seuil et de bons coefficients de diffusion. La figure 3.1 montre un schéma de l'algorithme de la méthode proposée.

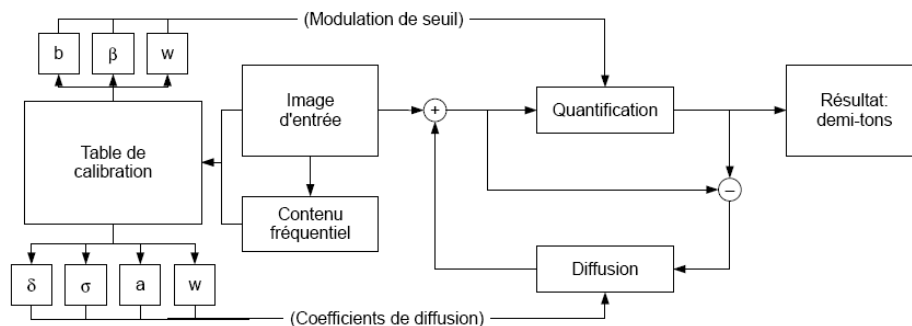


Figure 3.1 – Schéma de la méthode proposée pour la reconstruction de l'image.

3.1 Analyse de fréquences locales

Dans cette section, nous allons présenter la méthode utilisée pour trouver les contenus fréquentiels locaux. La méthode proposée va utiliser ces contenus pour choisir de bons coefficients de diffusion et pour décider l'amplitude de la modulation du seuil.

Les contenus doivent être trouvés pour chaque pixel dans l'image originale. Ainsi le processus pour trouver ces contenus doit être assez rapide pour ne pas trop ralentir le processus du rendu en demi-ton. Ce contenu doit aussi être assez précis afin d'augmenter la qualité du signal dans l'image reconstruite.

L'analyse de fourier classique [4] représente une image comme la somme de plusieurs ondes sinusoïdales avec des orientations, des périodes, des phases et des amplitudes différentes. Normalement, la Transformée de Fourier Discrète (DFT : *Discrete Fourier Transform*) [4] est utilisée sur l'image composée par des intensités discrètes. Dans cette représentation, la fréquence décrit la nature essentielle de l'image : les hautes fréquences correspondent aux détails fins et les basses fréquences correspondent aux variations à grande échelle (variations douces). Les images naturelles peuvent avoir des fréquences très variées.

La DFT est largement utilisée dans les analyses globales. Cependant, beaucoup d'applications en traitement d'images demande une analyse locale. Plusieurs recherches se sont concentrées sur ce but [4, 27]. Notre méthode utilise les contenus locaux des signaux (fréquences/orientations/contrastes principaux dans une petite fenêtre) afin d'avoir une bonne réponse aux structures présentes dans l'image. Dans le cas où il y a un contenu fréquentiel complexe, notre méthode ne considère que le contenu fréquentiel le plus important.

3.1.1 Orientation locale

L'analyse spectrale locale ne fait pas partie de nos contributions. N'importe quel outil d'analyse peut être utilisé, s'il détecte le contenu fréquentiel principal. Nous avons choisi une méthode présentée au chapitre 13 de [10] pour trouver l'orientation locale basée sur la variance du gradient local. Soit un filtre de flou isotropique B et soient les

filtres de dérivées partielles suivants :

$$D_x = (D_y)^t \quad D_y = \begin{bmatrix} 0.00459622 \\ -0.0239629 \\ 0.0727275 \\ -0.173894 \\ 0.378736 \\ -0.934465 \\ 0 \\ 0.934465 \\ -0.378736 \\ 0.173894 \\ -0.0727275 \\ 0.0239629 \\ -0.00459622 \end{bmatrix} \quad (3.1)$$

On définit $J_{pq} = B(D_p \dot{D}_q)$ une opération qui applique les filtres D_p et D_q à l'image originale pour chaque pixel, puis multiplie les deux images obtenues pixel par pixel, et applique finalement le filtre B au produit.

Trouver l'orientation locale s'effectue en solutionnant le problème de valeurs propres qui correspond à trouver la rotation à appliquer à la matrice $\begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix}$ pour la ramener à son système de coordonnées d'axes principaux [1] :

$$\begin{bmatrix} J_x & 0 \\ 0 & J_y \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix} \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \quad (3.2)$$

L'angle correspondant à l'orientation locale ϕ peut être trouvé par :

$$\tan 2\phi = \frac{2J_{xy}}{J_{xx} - J_{yy}} \quad (3.3)$$

L'angle ϕ représente l'orientation la plus importante qu'on trouve localement. La figure 3.2 montre un résultat visuel pour les valeurs de ϕ . Dans le cas où il n'y a pas d'information "orientée" dans l'image originale, ϕ peut correspondre à une valeur quelconque. Par exemple sur un aplat d'intensité constante, les valeurs apparaissent comme du bruit à la figure 3.2 (des petits cercles ou points isolés). Dans la méthode proposée, nous utilisons non seulement l'orientation locale, mais aussi la fréquence locale et le contraste local. Ainsi ce bruit sur l'orientation ne cause que peu de problèmes parce que sur les zones sans orientation évidente, le contraste local et la fréquence locale sont aussi très faibles, ce qui cause les coefficients et le seuil utilisés sur ces zones à être très proches du résultat de la diffusion d'erreur standard.

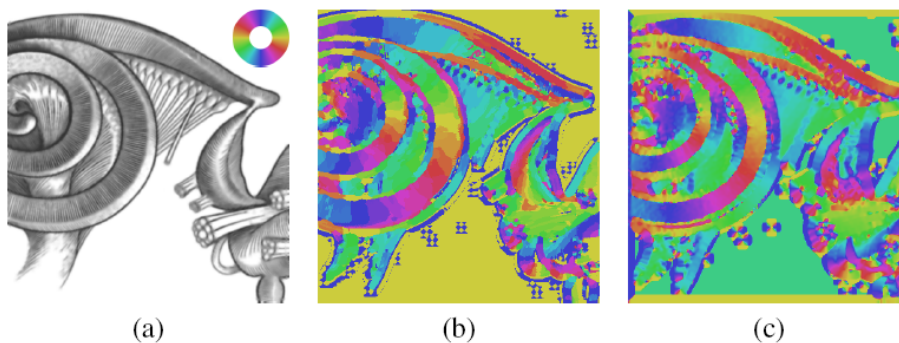


Figure 3.2 – L'image des orientations locales colorées. Nous montrons aussi les orientations trouvées par une autre méthode implémentée : la transformée de fourier discrète locale, qui consiste à trouver le signal maximum sur le spectre local. Nous n'avons pas utilisé cette méthode dans notre implémentation parce que son résultat est un peu trop bruité et pas assez lisse par rapport à la méthode préconisée [10]. (a) Image originale avec légende d'orientation en haut à droite. (b) Orientation selon la transformée de fourier discrète locale. (c) Orientation selon le calcul du gradient.

3.1.2 Fréquence locale et contraste local

Dans [10], Jahne donne aussi une façon de trouver des fréquences locales et des contrastes locaux. Cependant, nous avons utilisé la transformée de fourier discrète locale pour trouver ces valeurs parce qu'on a observé que cette dernière méthode nous donne un meilleur résultat sur les images reconstruites, grâce à sa stabilité sur différentes images.

En même temps, elle donne des fréquences et des contrastes plus lisses sur les pixels proches, ce qui aide à reconstruire l'image en demi-ton avec une distribution de points de meilleure qualité.

Une fois l'orientation locale ϕ trouvée, la fréquence locale peut être trouvée en cherchant la position de la valeur maximum dans le spectre de fourier le long de la direction de ϕ . Ce calcul est équivalent à trouver le maximum sur un tableau à une dimension. Cependant, on doit d'abord filtrer le spectre local avec un filtre non-linéaire pour diminuer les valeurs de basses fréquences sur le spectre, parce que ce sont des signaux de hautes et moyennes fréquences qui représentent des textures locales, mais pas des signaux de basses fréquences. On diminue un peu le poids du filtre sur les basses fréquences et on augmente un peu le poids du filtre sur les hautes fréquences.

Avec la fréquence locale et l'orientation locale, le contraste local est trouvé en retirant la valeur du point dans le spectre avec une orientation locale et fréquence locale.

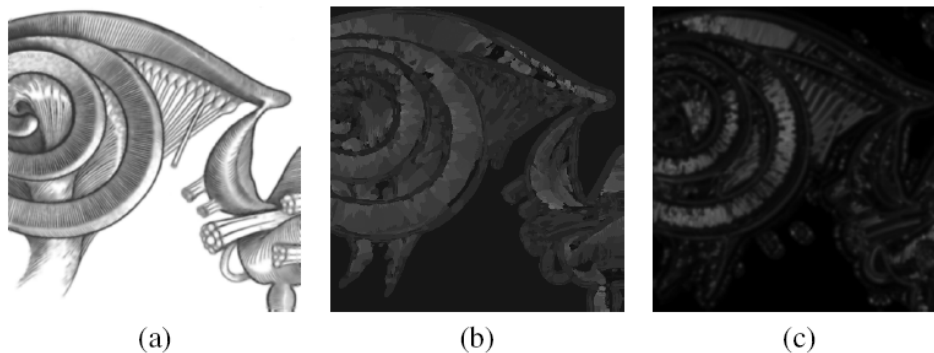


Figure 3.3 – L'image des fréquences locales colorées. L'intensité de chaque pixel représente sa fréquence locale. (a) Image originale. (b) Fréquences locales selon la transformée de fourier discrète locale. (c) Fréquences locales selon le calcul du gradient.

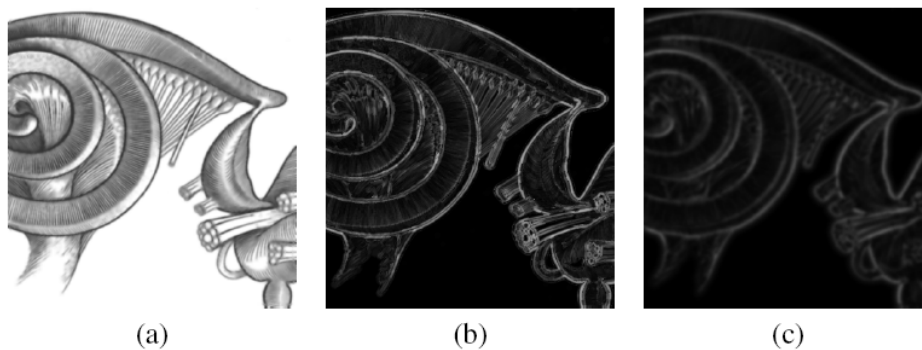


Figure 3.4 – L'image des contrastes locaux colorés. L'intensité de chaque pixel représente son contraste local. (a) Image originale. (b) Contrastes locaux selon la transformée de fourier discrète locale. (c) Contrastes locaux selon le calcul du gradient.

3.1.3 Seuillage

Comme nous avons vu au chapitre précédent, la modulation du seuil est un outil puissant et souvent utilisé dans les méthodes de rendu en demi-ton par voisinage afin de minimiser la perte d'information de l'image originale [9, 12, 13]. Cependant, ces méthodes sont incapables de préserver l'information de texture locale, soit parce qu'elles manquent de contrôle local, soit parce qu'elles ne considèrent pas l'information sur l'orientation.

Dans la méthode proposée, pour s'adapter aux structures de l'image originale, nous avons utilisé un filtre anisotrope pour construire l'image de seuil. L'orientation de ce filtre anisotrope est égale à l'orientation locale du pixel, la taille du filtre à la fréquence locale du pixel, et l'amplitude du filtre au contraste local. Pour avoir la flexibilité de ces contrôles dans le filtre, le filtre de Gabor est utilisé dans la méthode proposée, et il est défini par :

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \\ G_{\theta, \omega}(x, y) &= \exp\left(-\frac{x'^2 + y'^2}{2\sigma_G^2}\right) \cos \omega x' \end{aligned} \quad (3.4)$$

Comme les contenus fréquentiels sont variés d'un endroit à l'autre de l'image, l'orientation et la fréquence du filtre de Gabor peuvent être différentes d'un pixel à l'autre. Nous utilisons la fréquence et l'orientation trouvées dans la section précédente pour construire un filtre de Gabor pour chaque pixel :

$$\begin{aligned} \theta &= [x, y] \\ \omega &= w[x, y] \\ S(x, y) &= \beta \sum_{i=-l}^l \sum_{j=-l}^l c(x-i, y-j) G_{\theta, \omega}(i, j) \end{aligned} \quad (3.5)$$

Ici $2l$ est la taille de fenêtre filtrée par le filtre de Gabor. Une valeur $\sigma_G = 1.6$ est utilisée dans la méthode proposée.

La valeur de β contrôle l'amplitude du filtre pour chaque composition de contenu fréquentiel, c'est-à-dire, pour chaque composition possible de la fréquence locale et du contraste local, nous définissons une valeur de β parce que nous avons observé que

certaines compositions sont plus difficiles à rendre que d'autres. Par exemple, un pixel avec une fréquence locale très élevée et un contraste assez bas est beaucoup plus difficile à rendre qu'un pixel avec une basse fréquence et un fort contraste. Nous allons présenter comment définir et calibrer les valeurs de β dans la section 3.3.

3.2 Coefficients de diffusion

Les méthodes proposées par Ostromoukhov en 2001 [20] et Wang et al. en 2003 [35] utilisent un ensemble de trois coefficients de diffusion d'erreur pour chaque ton, afin d'obtenir une distribution de points belle et lisse dans l'image reconstruite. Cependant, ces deux méthodes conservent seulement des signaux de basses fréquences et elles peuvent brouiller des signaux de hautes fréquences comme des détails fins, comme montrés par les images au chapitre prochain. Ulichney a mentionné dans son livre [29] qu'un filtre plus étendu en taille va augmenter la capacité à rendre l'image plus visible, c'est-à-dire augmenter la visibilité de détails fins. Marcu et Abe [17] ont exploré l'utilisation de coefficients analytiques, en particulier de coefficients créés par un noyau gaussien.

Selon notre hypothèse, des coefficients de diffusion qui suivent le contenu fréquentiel local doivent conserver mieux les détails fins que la méthode de diffusion d'erreur standard. Nous avons testé cette hypothèse et nous avons trouvé que le meilleur résultat est obtenu avec un ensemble de 12 coefficients (les valeurs de ces coefficients sont obtenus par un filtre gaussien). Ce filtre gaussien a deux paramètres configurables pour chaque combinaison d'orientation locale, de fréquence locale et de contraste local, soient la variance σ^2 et l'anisotropie α .

Ces deux paramètres sont contrôlés par une calibration en précalcul, ce qui sera présenté à la prochaine section. En absence de contenu fréquentiel local, des coefficients standard vont être utilisés pour produire une belle transition. La transition des coefficients standard et des coefficients du filtre gaussien est contrôlée par une interpolation linéaire avec un paramètre de calibration ω .

3.3 Calibration

Dans la méthode proposée, il y a quatre paramètres à calibrer pour chaque combinaison possible du contenu fréquentiel local. β est l'amplitude du filtre de Gabor de la modulation de seuil ; σ et α sont les paramètres du filtre gaussien pour les coefficients de diffusion ; ω est le facteur d'interpolation entre les coefficients standard et les coefficients gaussiens. Parce que le nombre de ces combinaisons est très grand, il faut utiliser un processus pour simplifier la calibration.

Notre méthode s'appuie sur une hypothèse fondamentale : si nous pouvons trouver un résultat optimal dans une image artificielle qui contient un seul contenu fréquentiel, nous pouvons alors obtenir un résultat proche d'optimal dans une région d'une image naturelle qui a le même contenu fréquentiel.

Une méthode de la calibration des coefficients de diffusion automatique est utilisée dans [20] et [35] pour des aplats d'intensités constantes qui peuvent aussi être utilisées dans la méthode proposée. Cependant, nous avons choisi une méthode basée sur la comparaison côte-à-côte qui est fréquemment utilisée dans le domaine de la correspondance de couleurs par exemple [34]. Dans notre approche, un aplat de référence en ton continu est affiché côte-à-côte à un aplat en demi-ton qui montre le résultat de la méthode proposée avec des paramètres de calibration donnés. Les valeurs des paramètres de calibration sont modifiées par l'observateur jusqu'à ce qu'il observe que les deux côtés sont le plus similaire possible.

La figure 3.5 montre l'interface de la calibration. Pour calibrer des paramètres d'un contenu fréquentiel donné, nous avons créé un aplat carré avec la fréquence et l'orientation données, et qui contient deux signaux avec des contrastes différents : la partie centrale montrée dans le cercle rouge est le signal avec le contraste à calibrer ; la partie autour du cercle montre le signal avec le contraste déjà calibré.

D'abord, l'observateur modifie la distance entre l'écran et ses yeux jusqu'à ce que les ondes sinusoïdales dans la partie autour du cercle deviennent juste invisibles. A cette distance, les ondes sinusoïdales dans la partie centrale du côté référence doivent être sur le seuil de la visibilité. Ensuite, l'observateur déplace les curseurs de l'outil à droite

pour ajuster les paramètres de calibration jusqu'à les ondes sinusoïdales dans la partie centrale du côté demi-ton deviennent au seuil de visibilité.

Une fois la calibration finie, un tableau de recherche de trois dimensions est rempli par interpolation des paramètres de calibration pour chaque combinaison du contenu fréquentiel. Ce tableau est utilisé dans l'étape de reconstruction de l'image en demi-ton pour obtenir les bons coefficients et le bon seuil pour chaque pixel à rendre. Notez que cette calibration doit être refait pour chaque type d'affichage. Parce que la courbe de reproduction peut être différente d'un type d'affichage à l'autre.

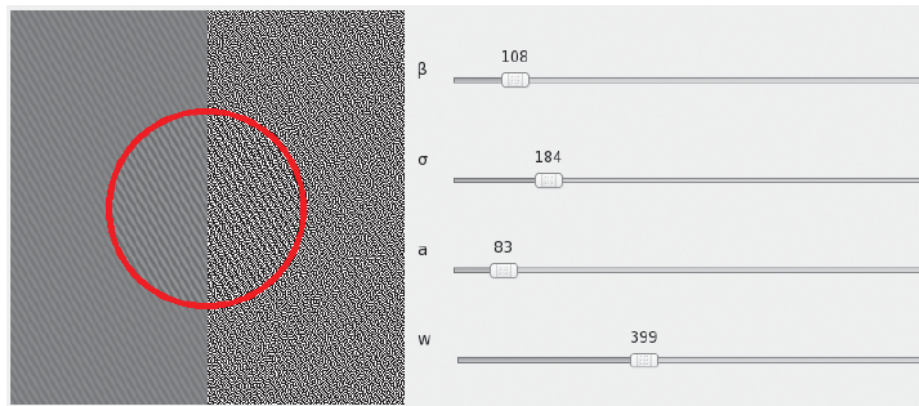


Figure 3.5 – L'outil proposé pour calibrer les paramètres dans la méthode proposée.

CHAPITRE 4

RESULTAT

Dans ce chapitre, les images en demi-ton reproduites par la méthode proposée et par les méthodes les plus récentes sont présentées. Nous allons évaluer ces résultats selon deux métriques MSSIM et le PSNR.

4.1 Evaluation visuelle

Pour visuellement comparer des images reconstruites, nous utilisons l'ensemble des images de test de [21]. Cet ensemble contient plusieurs caractéristiques importantes : des détails fins de contrastes variés, des zones de tons lisses, des structures visuellement identifiables (feuilles, cheveux, tissu), etc.

Nous comparons les images produites par la méthode proposée et par certaines méthodes récentes [13, 20, 21, 35]. Les images présentées sont en ordre (de gauche à droite, de haut en bas) : (a) l'image originale, (b) celle reconstruite par Zhou et Fang [35], (c) celle reconstruite par Kwak et al. [13] avec fort rehaussement de bordures, (d) celle reconstruite par Kwak et al. [13] avec faible rehaussement de bordures, (e) celle reconstruite par Pang et al. [21], et (f) celle reconstruite par la méthode proposée [5].

Sur les images, nous pouvons voir que la qualité des images reconstruites par la méthode proposée est meilleure que celle par l'ensemble de méthodes des diffusion d'erreur : les détails fins sont mieux préservés, nos images apparaissent plus nettes et plus jolies globalement. La comparaison visuelle de notre méthode avec celle de Pang et al. [21] montre que toutes les deux méthodes produisent des résultats de bonne qualité.

Nous ajoutons aussi quelques comparaisons avec des images artificielles : une plage d'intensités 0.5 à 1, avec des aplats fréquentiels et des contrastes variés. La qualité visuelle de notre méthode pour la plage n'est pas une surprise : en absence de contenu fréquentiel fort, notre diffusion d'erreur devient une diffusion d'erreur standard. Les aplats fréquentiels montrent les avantages de la méthode proposée comparés avec ceux

des autres méthodes de diffusion d'erreur. Comparée à la diffusion d'erreur standard, notre méthode est meilleure pour les plages de gris et les aplats fréquentiels. Notre méthode produit de meilleurs résultats que la famille de méthodes de modulation de seuil, grâce à son seuil et ses coefficients variés en fonction du contenu fréquentiel. Notez que dans l'image des aplats fréquentiels, sur les zones où la fréquence est très élevée, les méthodes de modulation de seuil vont "créer" des faux signaux fréquentiels, qui seront plus apparents que le signal original. Ces fausses fréquences sont beaucoup moins perceptibles dans l'image produite par notre méthode.

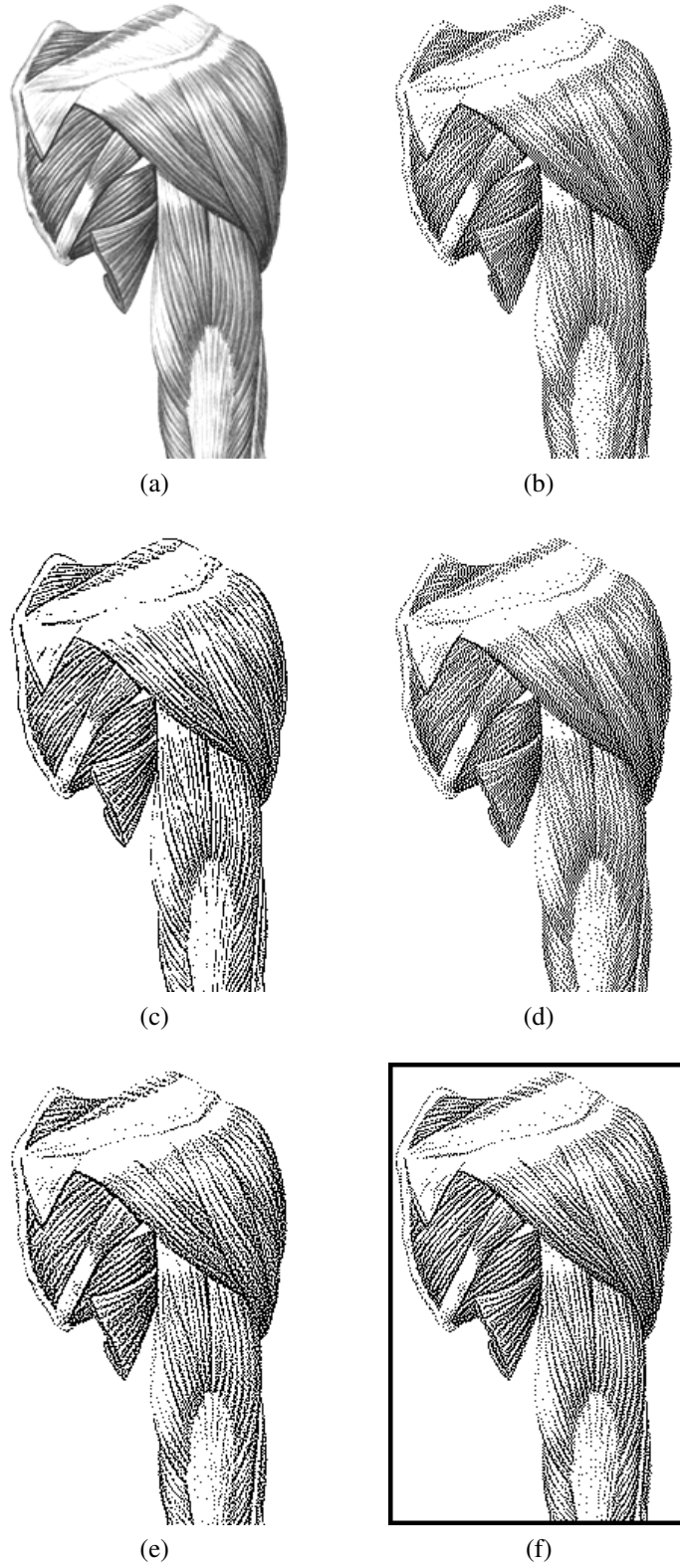
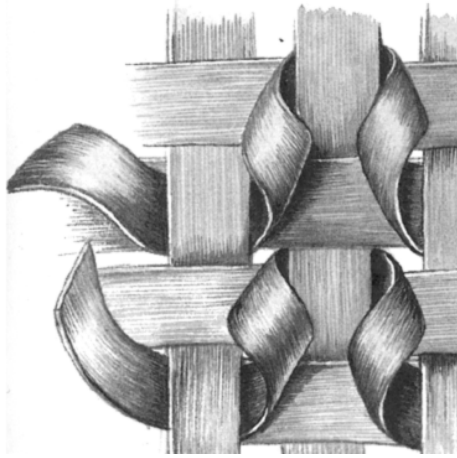
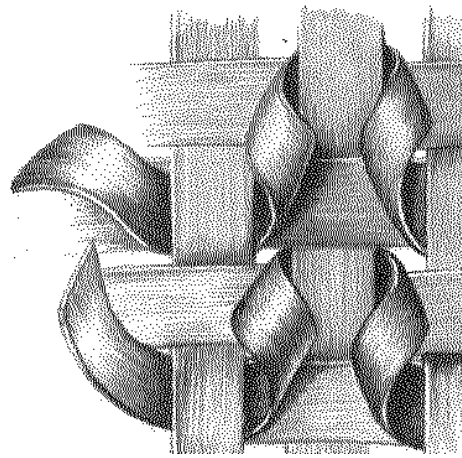


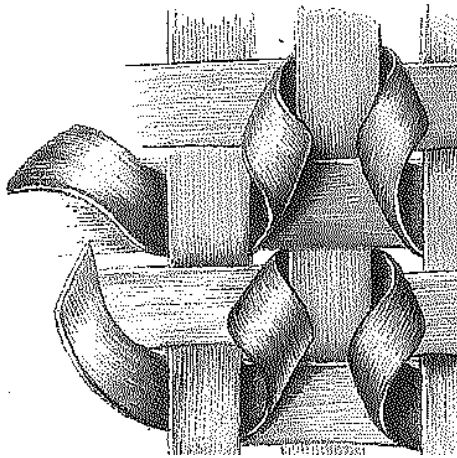
Figure 4.1 – Bras



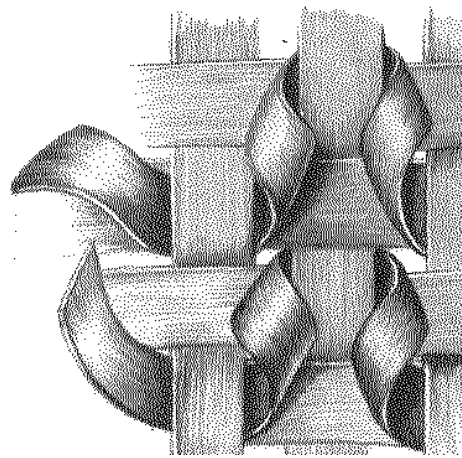
(a)



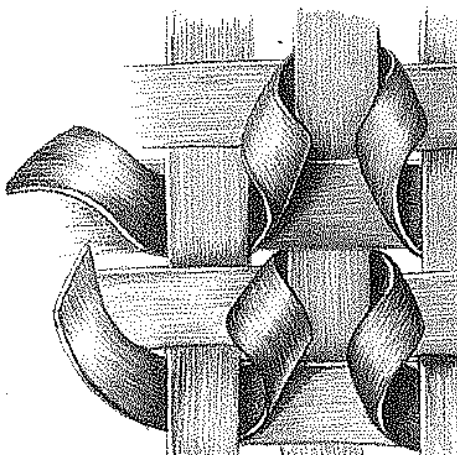
(b)



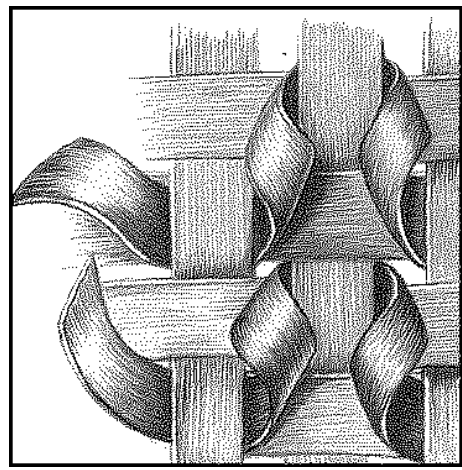
(c)



(d)

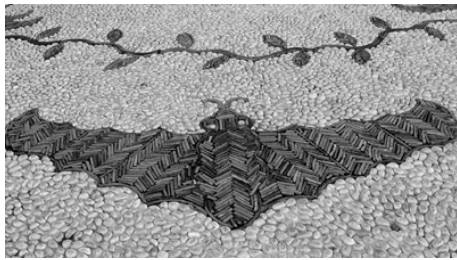


(e)

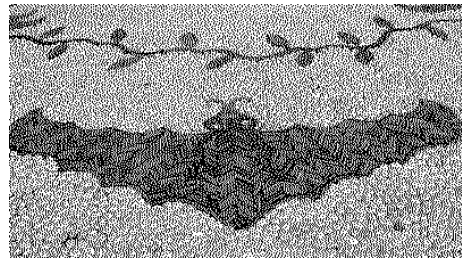


(f)

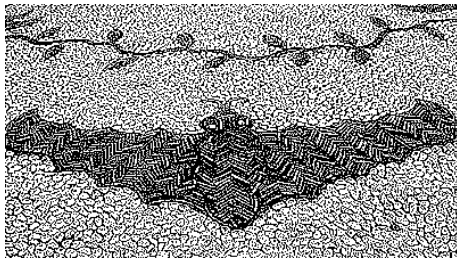
Figure 4.2 – Osier



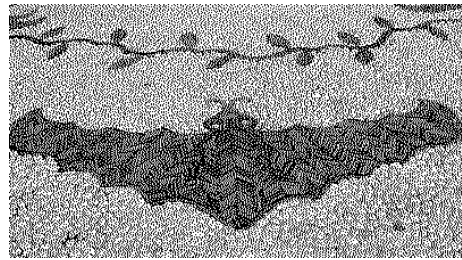
(a)



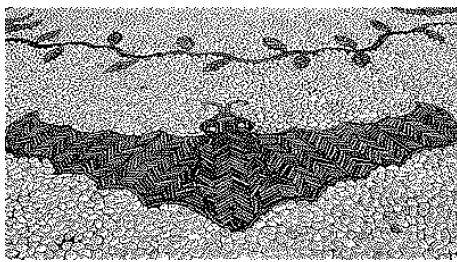
(b)



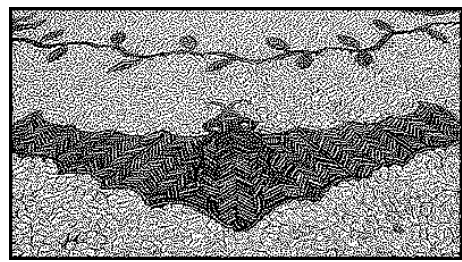
(c)



(d)



(e)

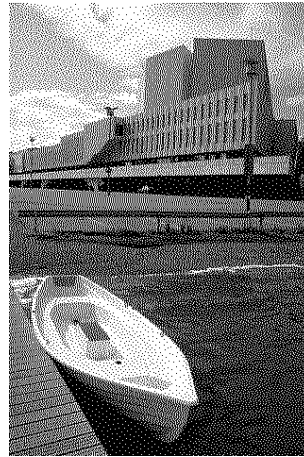


(f)

Figure 4.3 – Chauve-souris



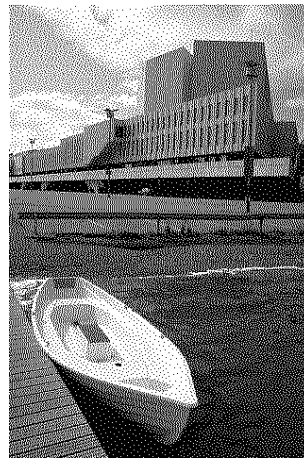
(a)



(b)



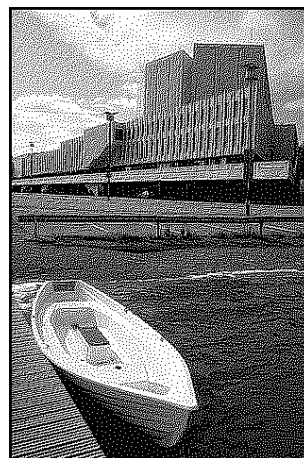
(c)



(d)



(e)



(f)

Figure 4.4 – Bateau



Figure 4.5 – Chat (original)

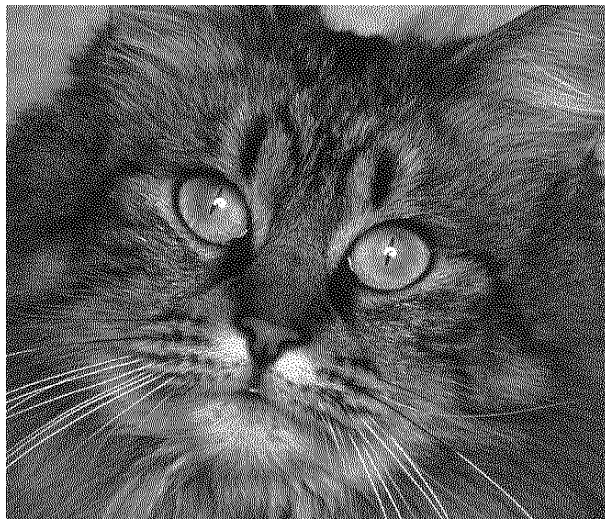


Figure 4.5 – Chat (Ostromoukhov)



Figure 4.5 – Chat (rehaussement de bordures : fort)

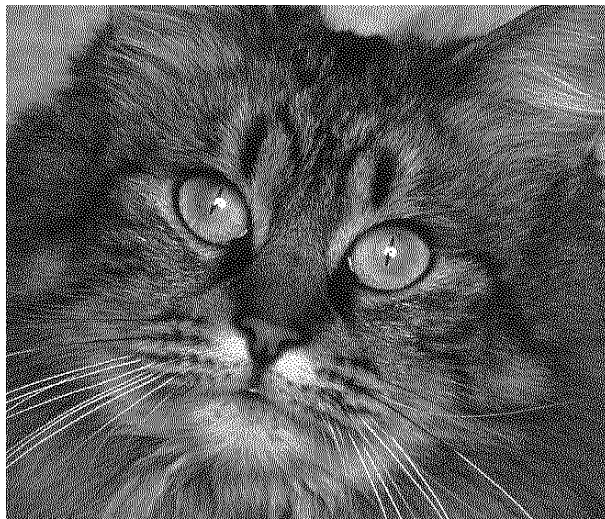


Figure 4.5 – Chat (rehaussement de bordure : faible)

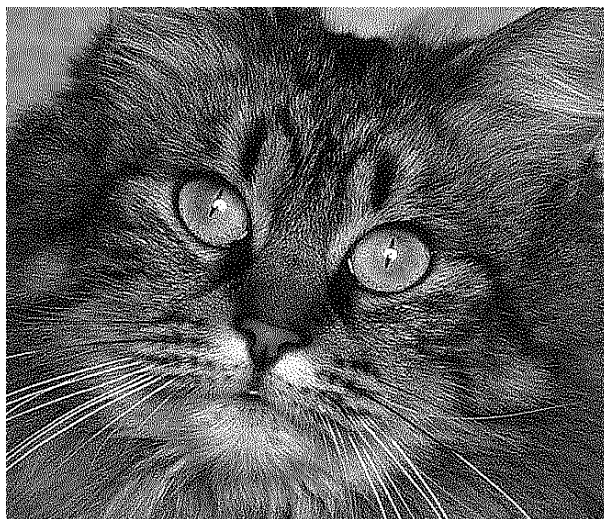


Figure 4.5 – Chat (Pang et al.)

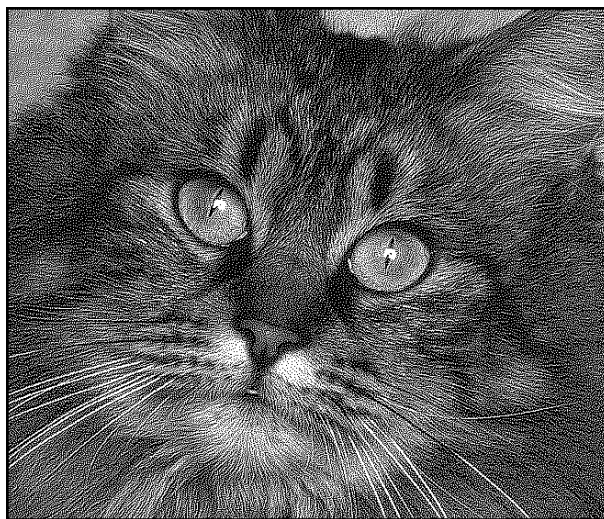


Figure 4.5 – Chat (méthode proposée)

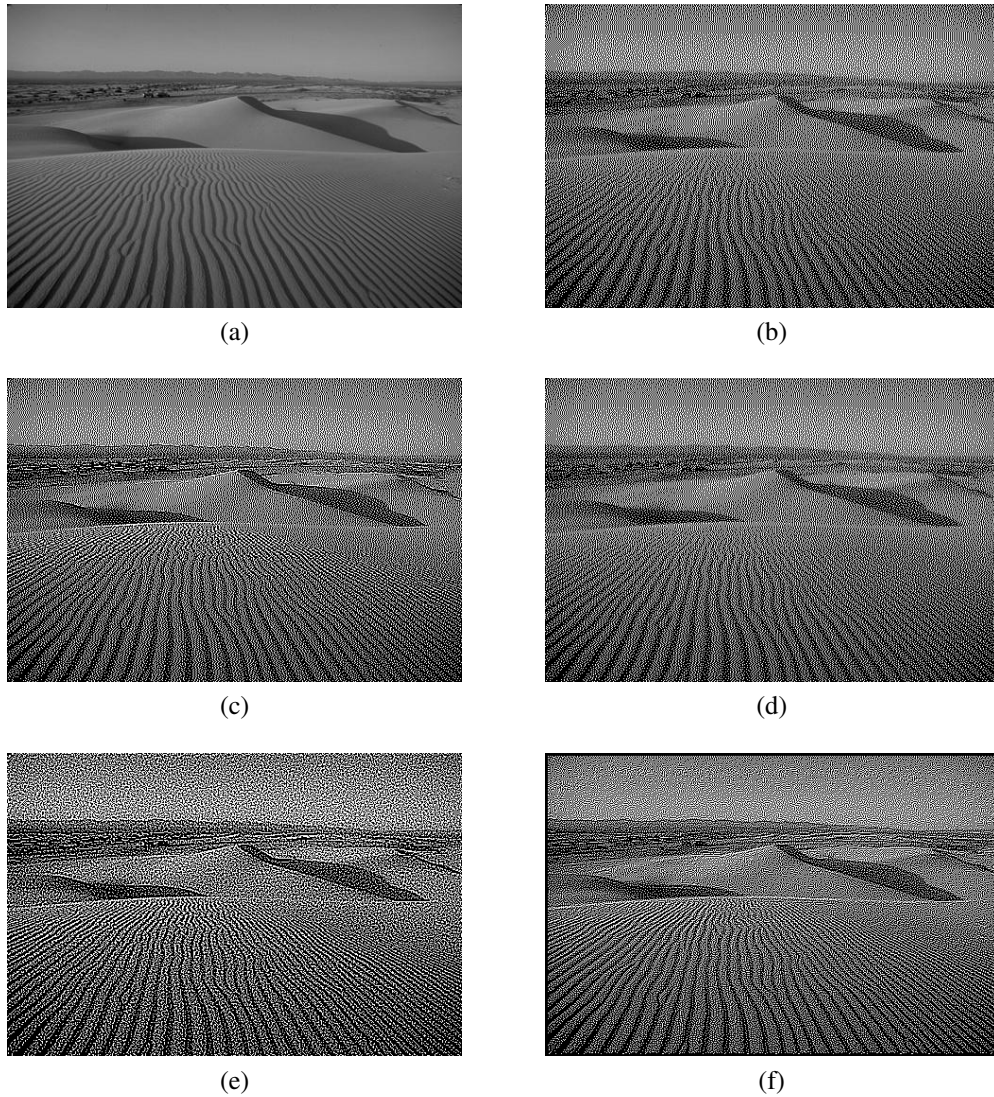


Figure 4.6 – Desert : Notez la "ligne" sur le ciel dans résultats de [20][35][13]. Parce que le ciel dans cette image est présenté par une zone gradient assez large. Et cette zone gradient est très difficile a rendre par les algorithmes error-diffusion standard a cause de leur défaut : introduire des structure réguliers.

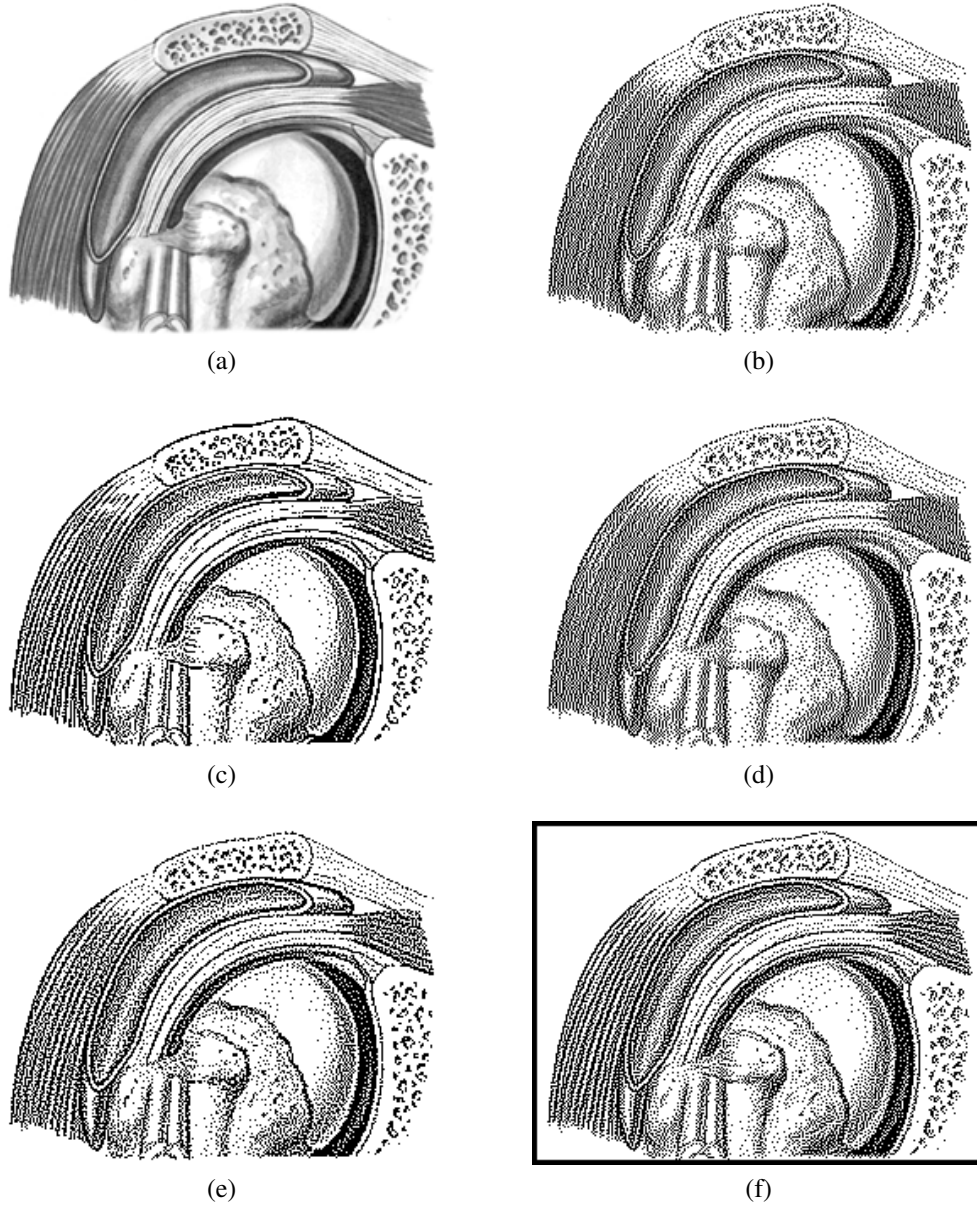
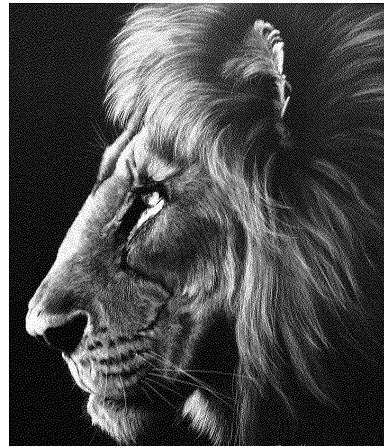


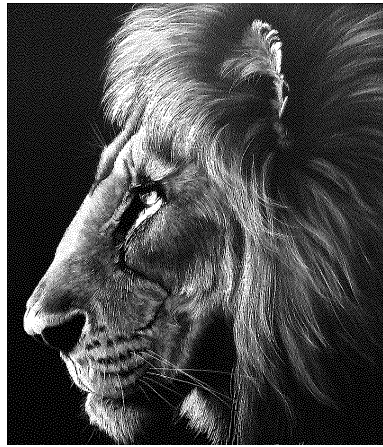
Figure 4.7 – Genou



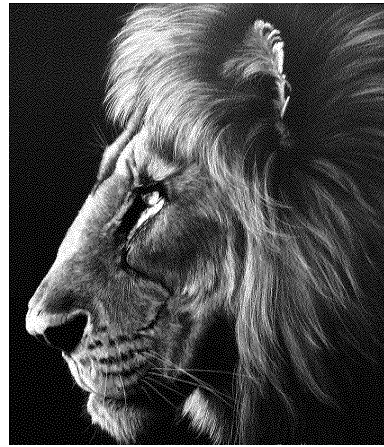
(a)



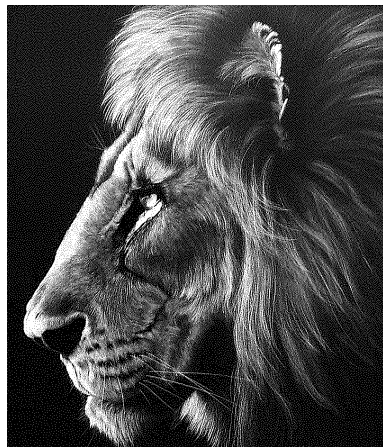
(b)



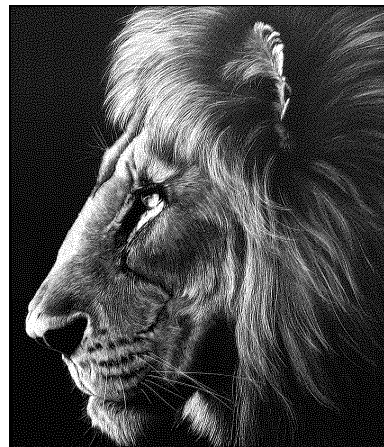
(c)



(d)



(e)



(f)

Figure 4.8 – Lion

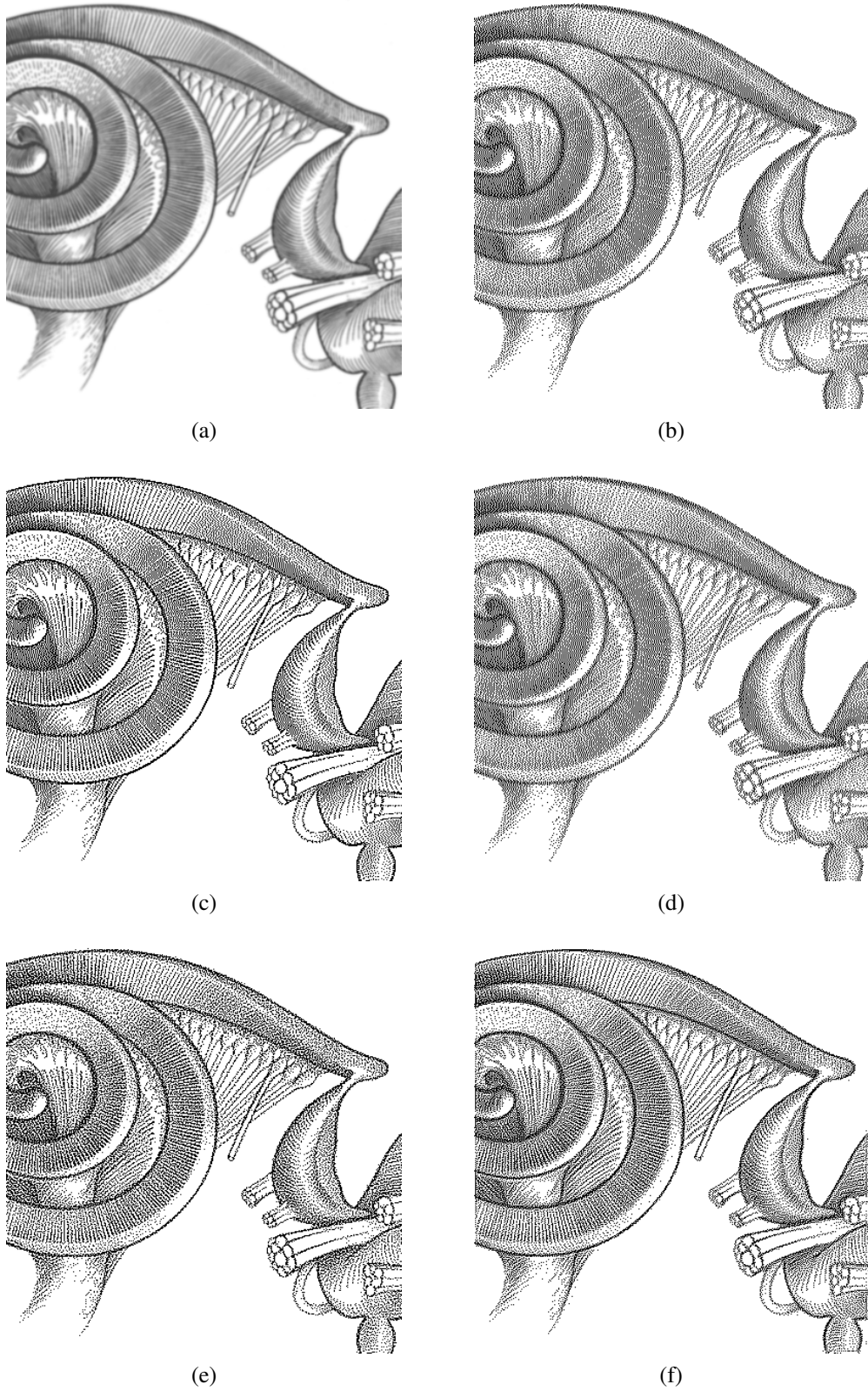


Figure 4.9 – Oreille interne



(a)



(b)



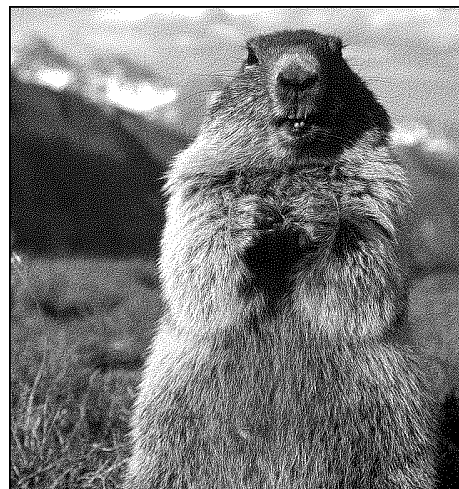
(c)



(d)

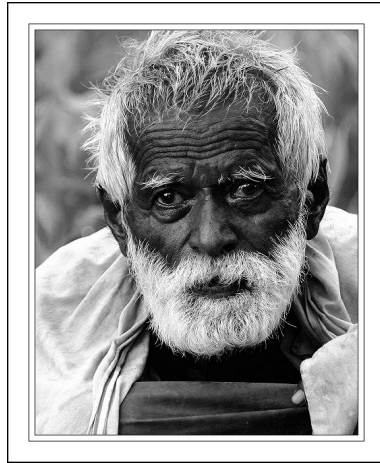


(e)

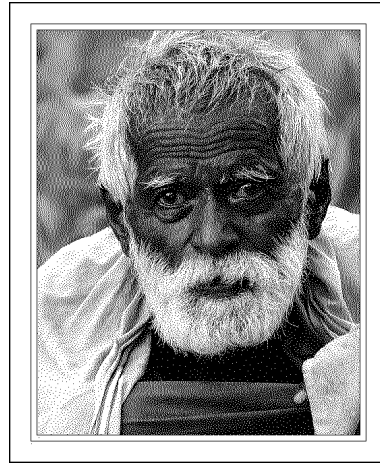


(f)

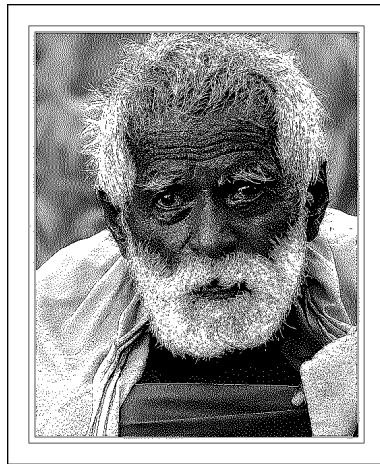
Figure 4.10 – Marmotte



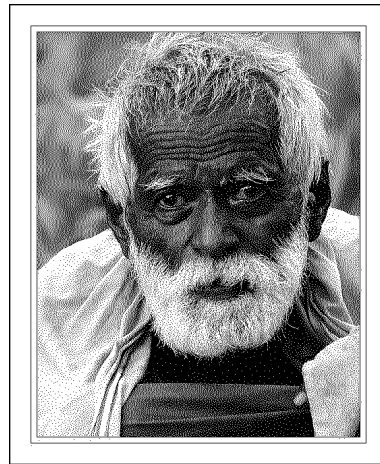
(a)



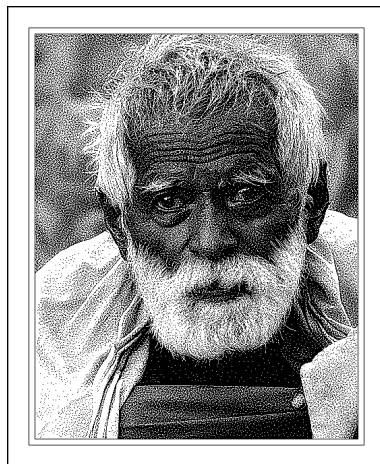
(b)



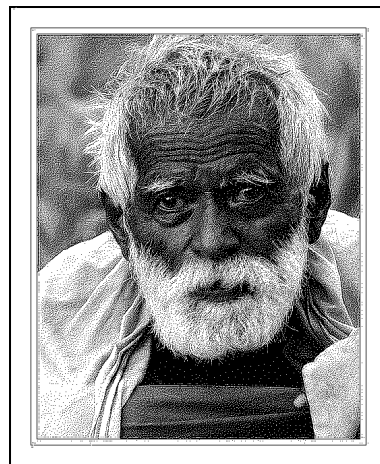
(c)



(d)



(e)



(f)

Figure 4.11 – Vieil homme

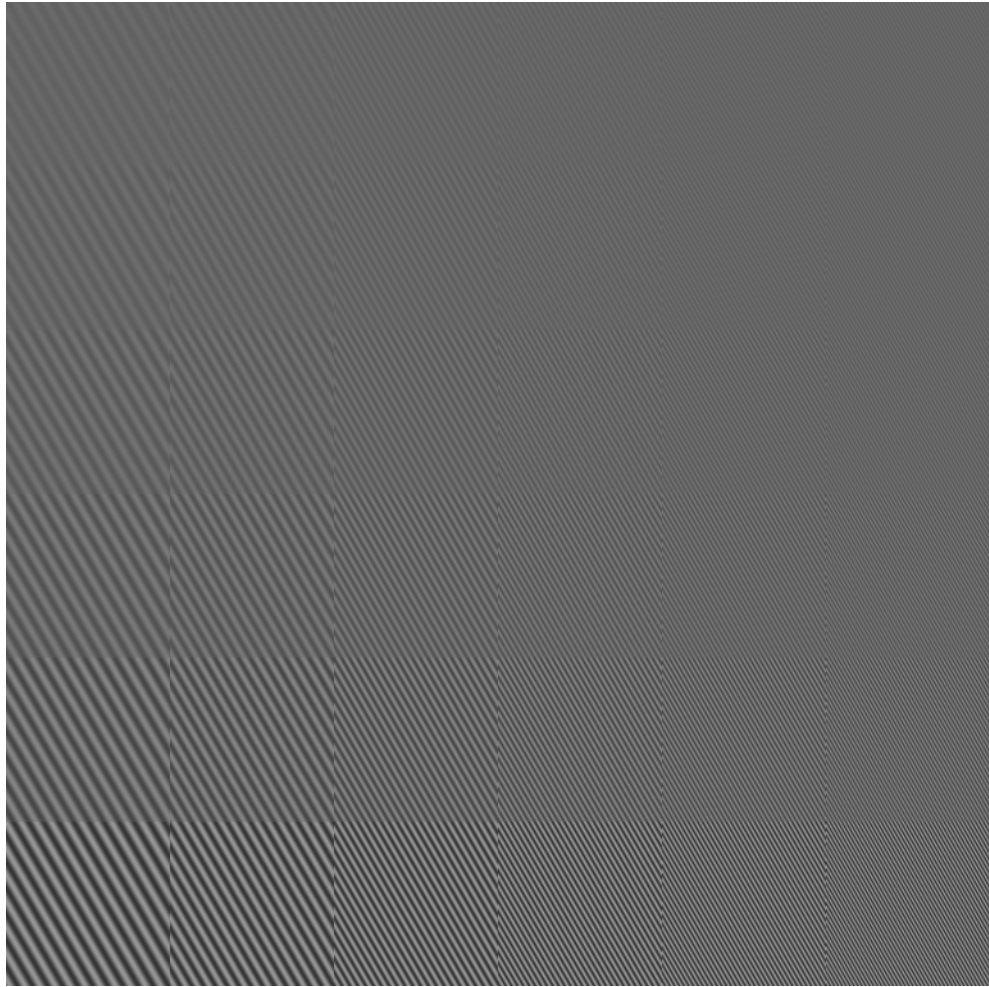


Figure 4.12 – Plage (original)

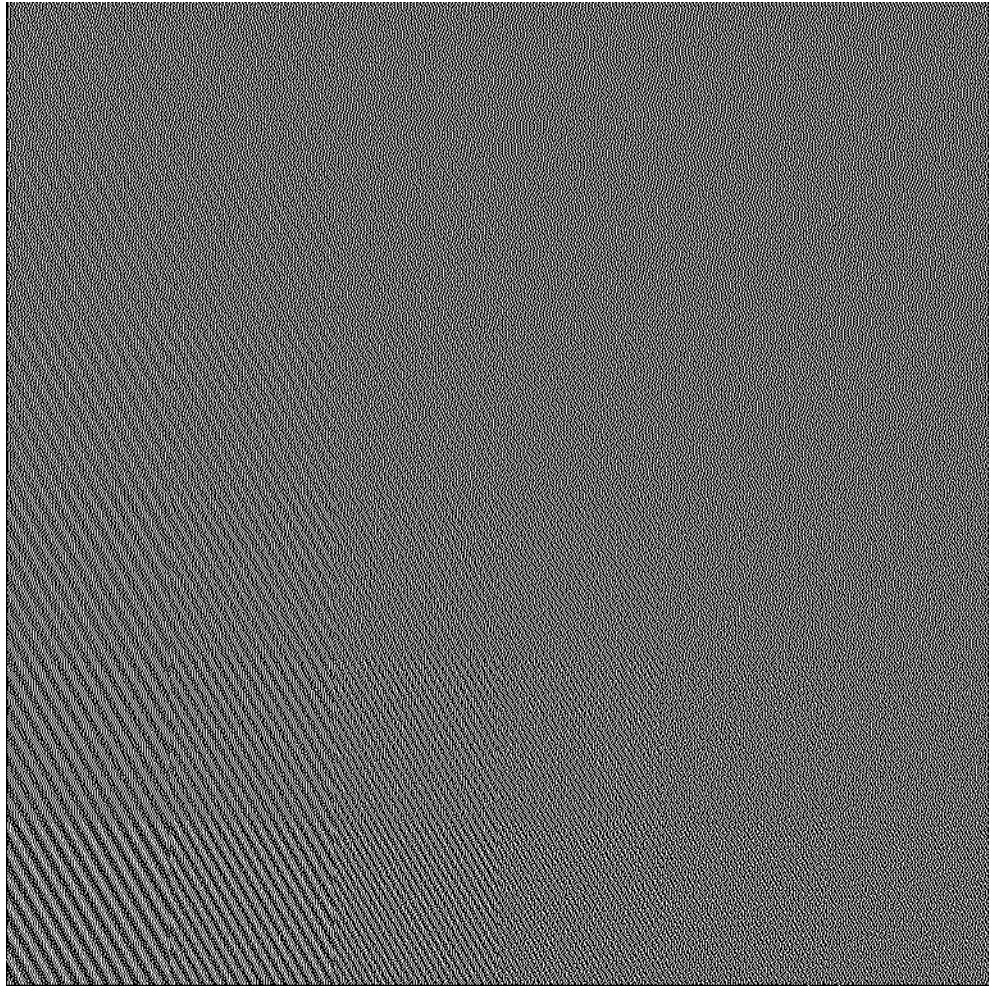


Figure 4.12 – Plage (Ostromoukhov)

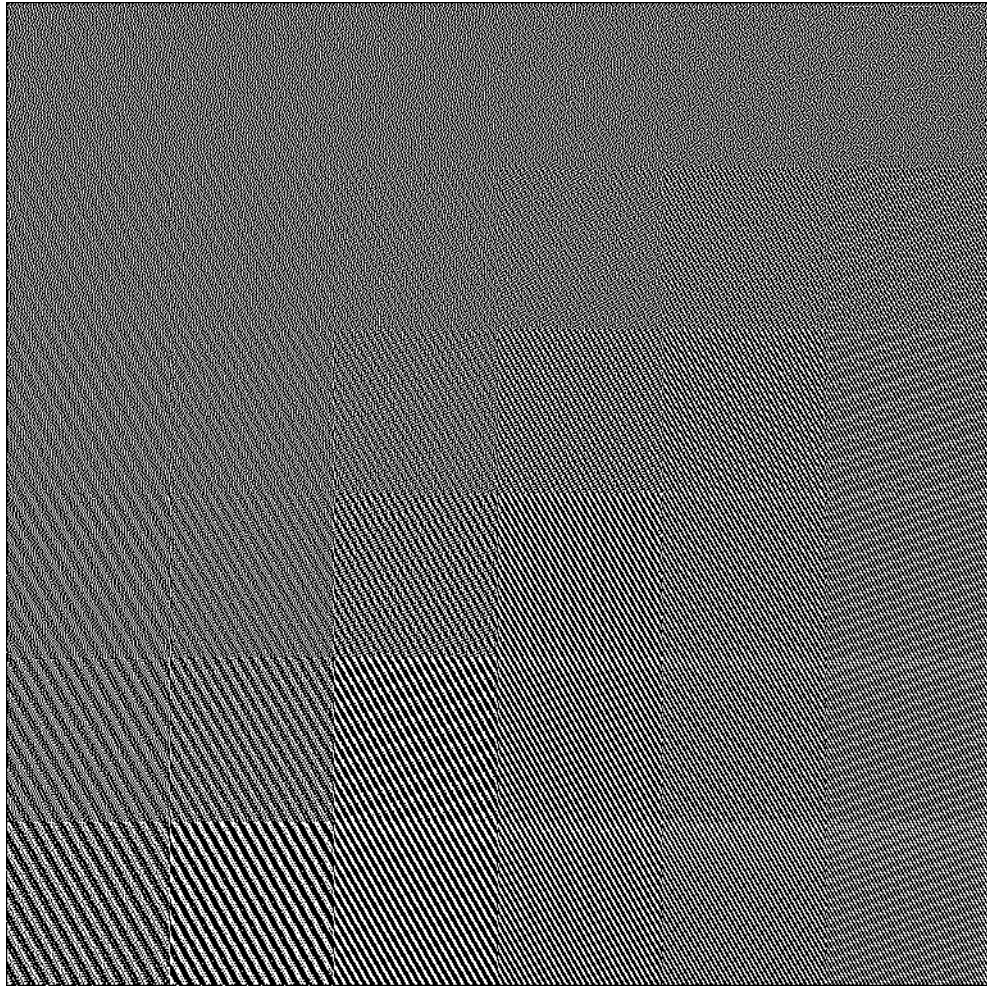


Figure 4.12 – Plage (rehaussement de bordures : fort)

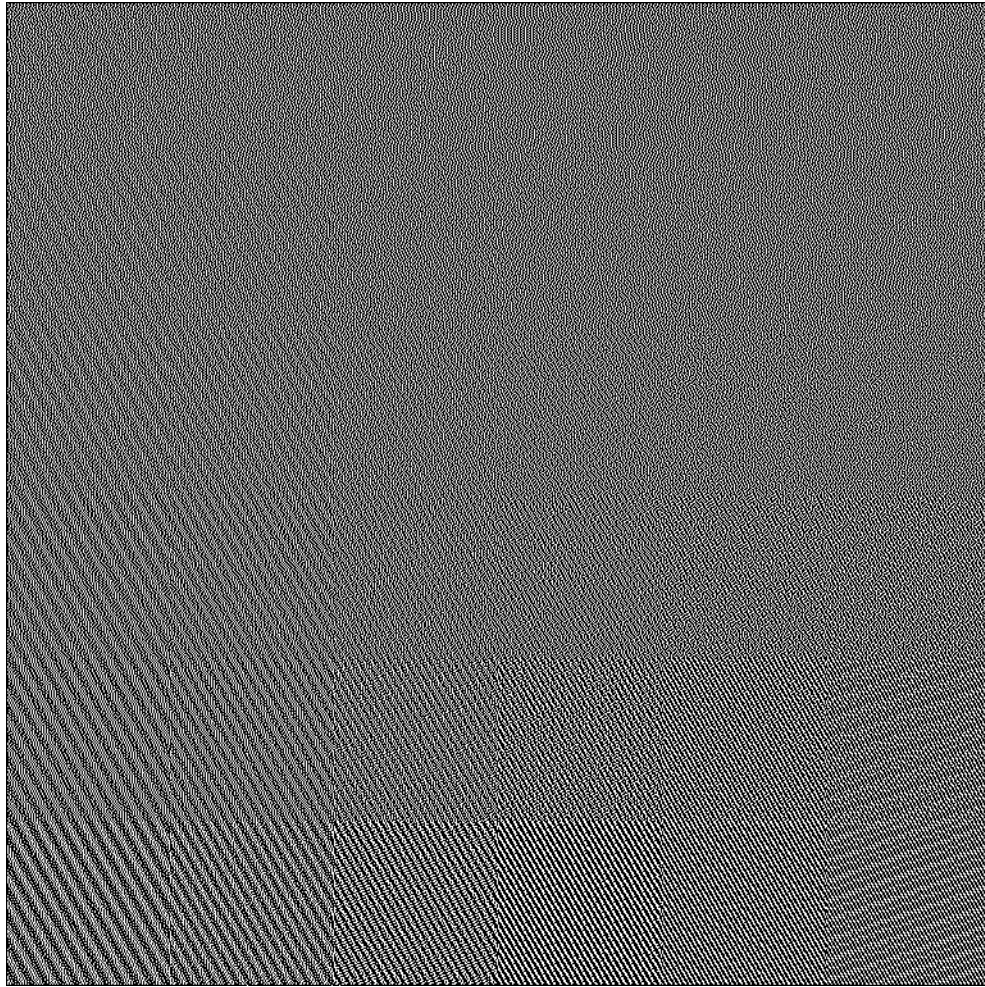


Figure 4.12 – Plage (rehaussement de bordures : faible)

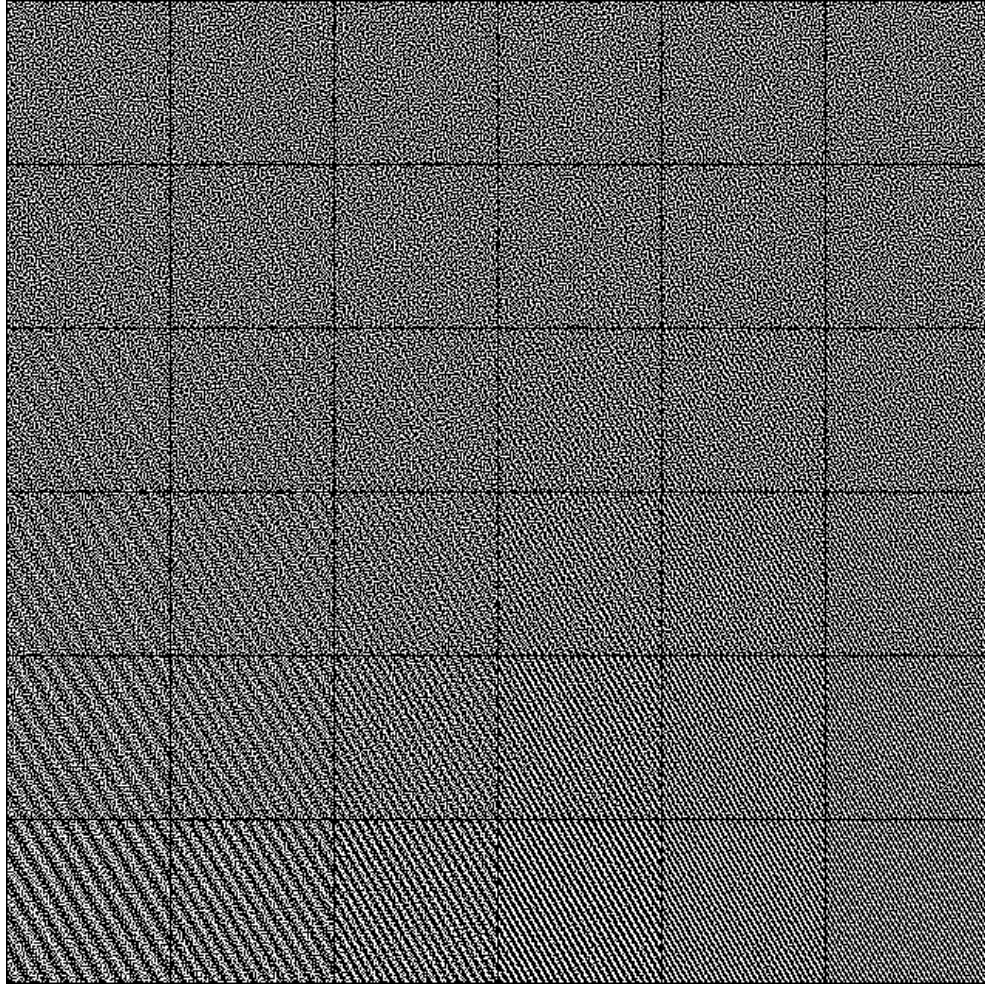


Figure 4.12 – Plage (Pang et al.)

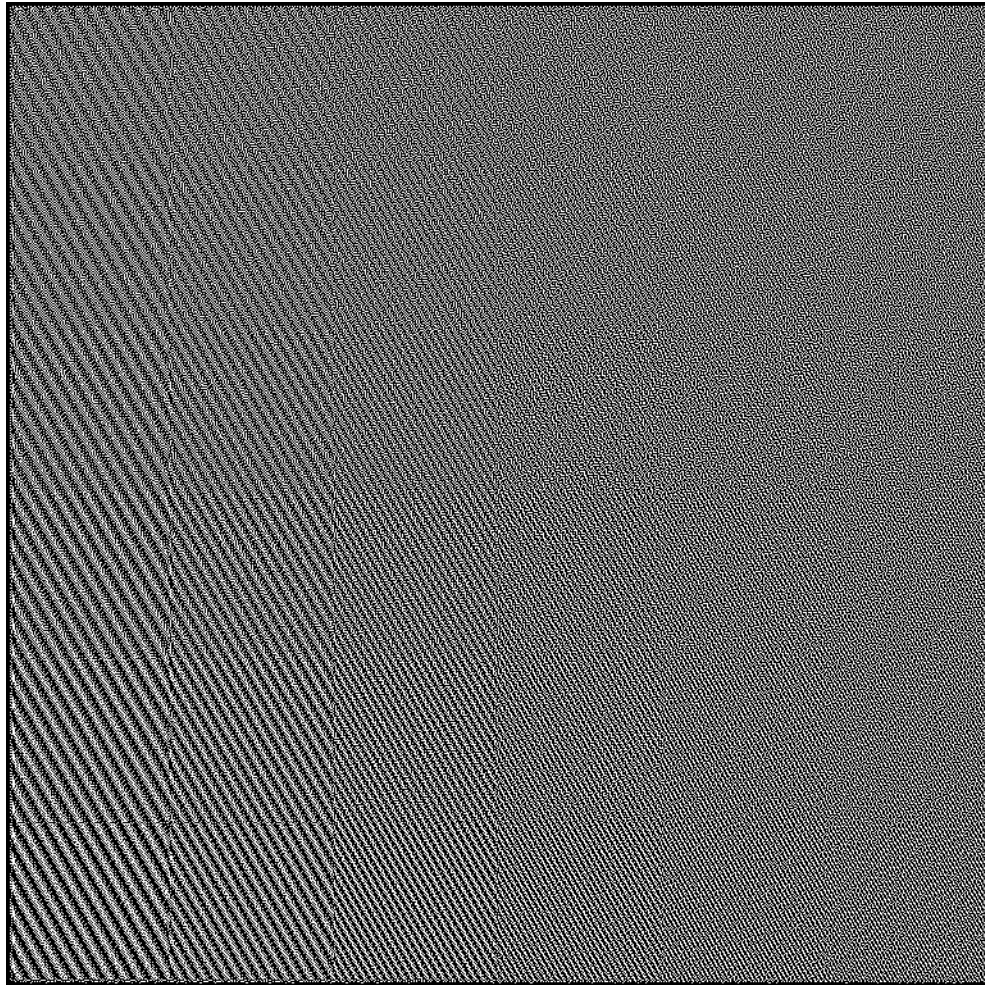
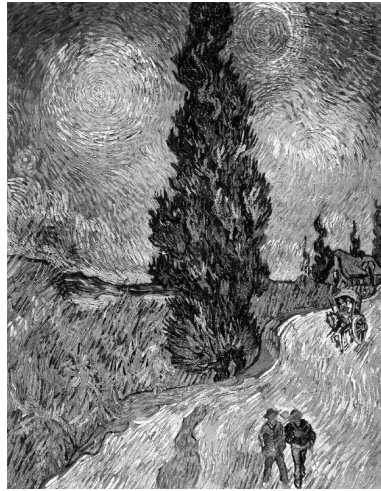


Figure 4.12 – Plage (méthode proposée)



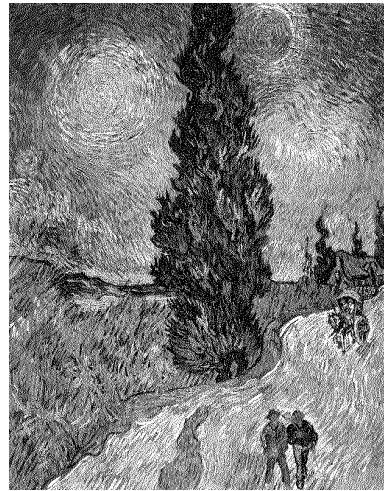
(a)



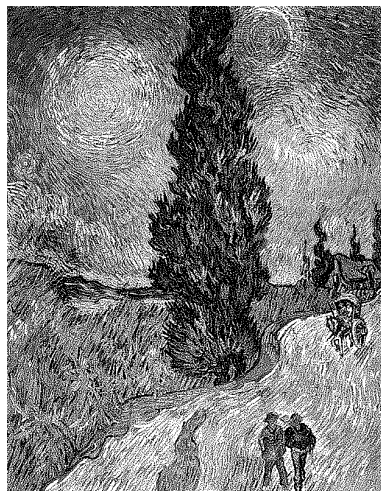
(b)



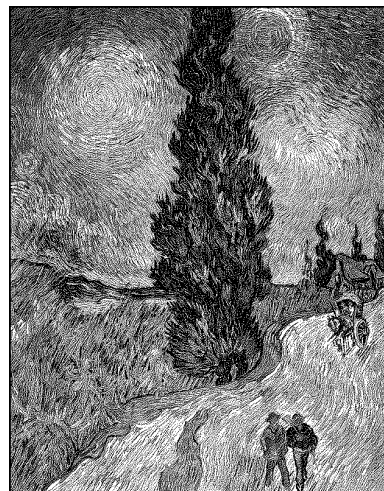
(c)



(d)



(e)



(f)

Figure 4.13 – Route



(a)



(b)



(c)



(d)

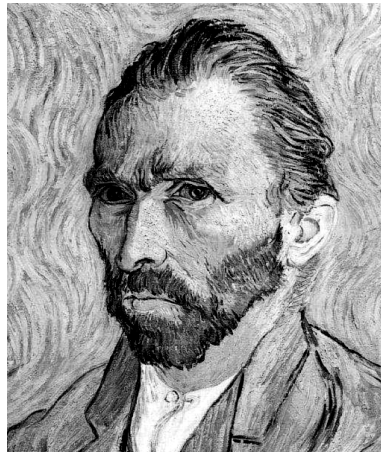


(e)

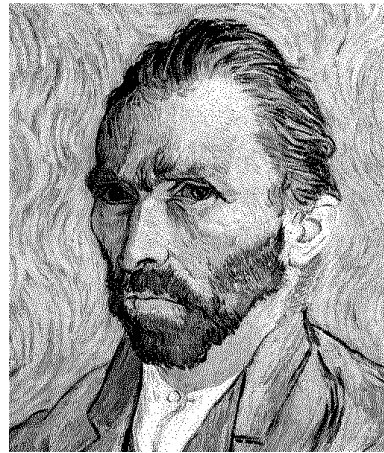


(f)

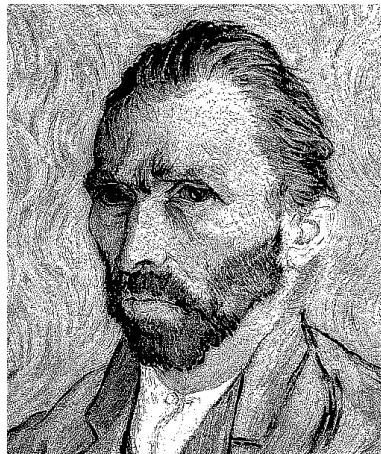
Figure 4.14 – Arbre



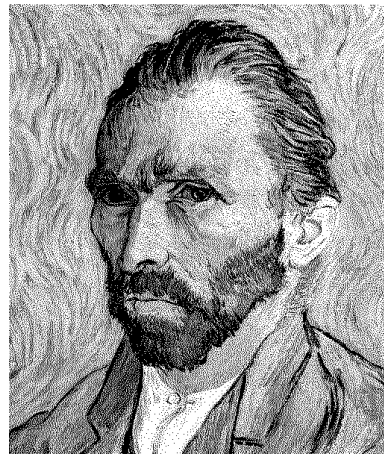
(a)



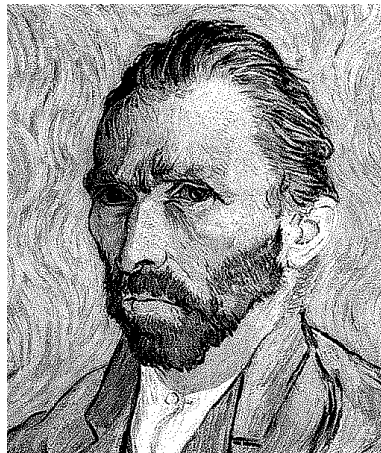
(b)



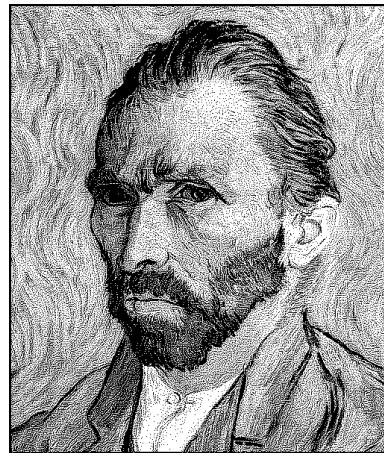
(c)



(d)



(e)

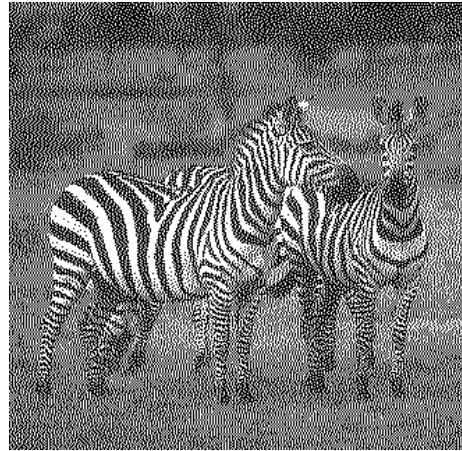


(f)

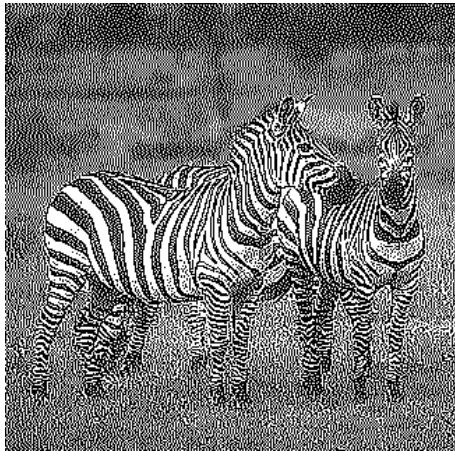
Figure 4.15 – Autoportraits de Van Gogh



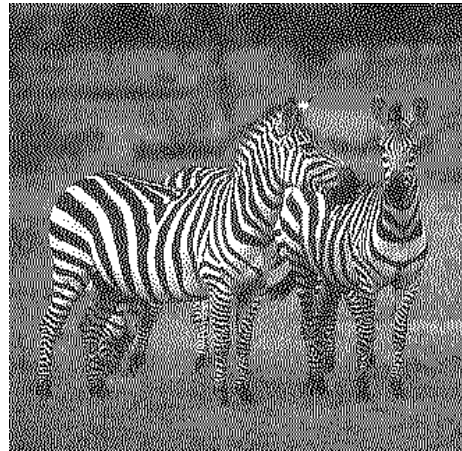
(a)



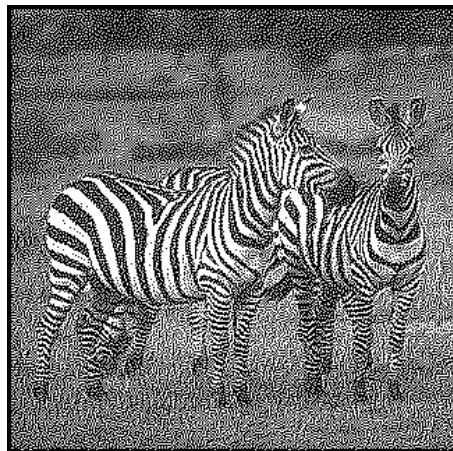
(b)



(c)



(d)



(f)

Figure 4.16 – Zèbres

4.2 Evaluation numérique

Pour confirmer que la méthode proposée est capable de préserver le ton local et des informations structurelles, deux métriques sont appliquées sur l'ensemble des images de test.

4.2.1 PSNR

La métrique "*peak-to-signal noise ratio*" (PSNR) est une métrique standard au domaine de la compression d'images avec perte. Cette métrique est basée sur l'erreur quadratique moyenne entre l'image originale I_o et l'image reconstruite I_r .

$$EQM = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I_o(i, j) - I_r(i, j)\|^2 \quad (4.1)$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{d^2}{EQM} \right) \quad (4.2)$$

où d est la valeur maximale d'intensité attribuable aux pixels.

| PSNR | | | | | |
|----------|----------------------|----------------------|---------------------|---------------------|---------------------|
| Image | Ostromoukhov 2001 | Zhou et Fang 2003 | Kwak et al. 2006 | Pang et al. 2008 | Méthode proposée |
| Bras | 41.333 | 30.618 | 27.796 | 35.467 | 29.715 |
| Buisson | 40.356 | 37.383 | 31.942 | 34.649 | 36.092 |
| Chat | 26.309 | 26.233 | 25.621 | 31.364 | 25.867 |
| Chemin | 38.832 | 34.050 | 27.685 | 32.365 | 31.458 |
| Genou | 39.210 | 29.238 | 27.129 | 33.489 | 27.746 |
| Homme | 39.977 | 38.288 | 28.741 | 32.013 | 34.747 |
| Lion | 39.794 | 38.225 | 34.391 | 35.682 | 33.999 |
| Oiseau | 42.560 | 34.370 | 29.296 | 36.769 | 33.316 |
| Osier | 40.549 | 30.386 | 28.041 | 29.117 | 29.099 |
| Portrait | 38.852 | 31.381 | 28.068 | 33.106 | 30.262 |
| Spirale | 40.703 | 34.131 | 33.477 | 34.766 | 30.509 |
| Taupe | 40.378 | 37.953 | 33.886 | 35.365 | 34.199 |
| Zèbres | 40.004 | 36.504 | 31.459 | 32.877 | 34.472 |
| Moyenne | 39.143 | 33.746 | 29.810 | 33.618 | 31.652 |

Tableau 4.I – Comparaison de la métrique PSNR entre la méthode proposée et des méthodes récentes. La méthode proposée se concentre mieux sur préserver des structures. Ainsi son PSNR est plus faible que la méthode [20], mais son PSNR est proche de celle de la méthode [21].

Nous avons utilisé un filtre passe-bas de taille 11×11 et de $\sigma = 2$. Voici le pseudo-code pour calculer la métrique PSNR :

$\{I_0$ et I_r sont des tableaux de taille $N \times M\}$

$EQM \leftarrow 0$

for $x = 0$ to $N - 1$, $y = 0$ to $M - 1$ **do**

$v_o \leftarrow 0$

$v_r \leftarrow 0$

$p \leftarrow 0$

for $i = -5$ to 5 , $j = -5$ to 5 **do**

$g \leftarrow \exp\left(\frac{i^2+j^2}{2\sigma^2}\right)$

$v_o \leftarrow v_o + g \cdot I_o[x+i, y+j]$

$v_r \leftarrow v_r + g \cdot I_r[x+i, y+j]$

$p \leftarrow p + g$

end for

$v_o \leftarrow v_o/p$

$v_r \leftarrow v_r/p$

$EQM \leftarrow EQM + (v_o - v_r)^2$

end for

$EQM \leftarrow EQM/(M \cdot N)$

$PSNR \leftarrow -10 \log_{10} EQM$

return PSNR

Cet algorithme est utilisé pour calculer le PSNR des résultats de la méthode proposée et des méthodes récentes sur l'ensemble des images de test. Le tableau 4.I montre la comparaison. Notre méthode donne des résultats très proches de ceux de la méthode [21], ce qui confirme que notre méthode préserve le ton local, même si la méthode a été créée pour conserver des structures dans l'image. La méthode [20] donne un PSNR plus élevé que toutes les autres méthodes parce qu'elle ne prend pas en compte des informations de structures.

4.2.2 MSSIM

La métrique "*mean structure similarity measure*" (MSSIM) est une métrique sur la qualité visuelle, introduite par Wang et al. [31] et utilisée dans [21] comme une source de critique principale. Cette métrique mesure la dissemblance entre l'image originale et l'image reconstruite selon trois valeurs calculées : l'intensité, le contraste et la structure. Ces trois valeurs sont calculées à partir des moyennes locales, des variances locales et des covariances locales.

Soient I_o et I_r les images d'entrée et de sortie, alors :

$$\mu_o(x, y) = \sum_i \sum_j I_o[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (4.3)$$

$$\mu_r(x, y) = \sum_i \sum_j I_r[i, j] \psi(x - i\tau_x, y - j\tau_y) \quad (4.4)$$

$$\sigma_o^2(x, y) = \sum_i \sum_j (I_o[i, j] - \mu_o[i, j])^2 \psi(x - i\tau_x, y - j\tau_y) \quad (4.5)$$

$$\sigma_r^2(x, y) = \sum_i \sum_j (I_r[i, j] - \mu_r[i, j])^2 \psi(x - i\tau_x, y - j\tau_y) \quad (4.6)$$

$$\sigma_{or}^2(x, y) = \sum_i \sum_j (I_o[i, j] - \mu_o[i, j])(I_r[i, j] - \mu_r[i, j]) \psi(x - i\tau_x, y - j\tau_y) \quad (4.7)$$

où ψ est un filtre gaussien pondéré de variance σ , et τ_x et τ_y correspondent à l'espace-ment le long des axes X et Y de la grille digitale de pixels.

Selon la loi de Weber pour le système visuel humain, l'œil est plus sensible à la luminance relative plutôt que la luminance absolue. Ainsi la composante luminance est obtenue par l'équation (2.9). La composante contraste est obtenue par l'équation (2.10). La composante structure est calculée par l'équation (2.11). La valeur MSSIM est la moyenne du produit de ces trois composantes :

$$MSSIM = \frac{1}{MN} \int \int l(x,y)c(x,y)s(x,y)dxdy \quad (4.8)$$

où M et N sont la taille de l'image sur l'axe X et Y respectivement.

Comme dans le calcul du PSNR, un filtre de taille 11×11 est utilisé pour calculer MSSIM. Voici le pseudocode pour calculer MSSIM :

$\{I_o$ et I_r sont des tableaux de taille $N \times M$

$SSIM \leftarrow 0$

for $x = 0$ to $N - 1$, $y = 0$ to $M - 1$ **do**

$\mu_o \leftarrow 0, \mu_r \leftarrow 0$

$p \leftarrow 0$

for $i = -5$ to 5 , $j = -5$ to 5 **do**

$g \leftarrow \exp\left(\frac{i^2+j^2}{2\sigma^2}\right)$

$\mu_o \leftarrow \mu_o + g \cdot I_o[x+i, y+j]$

$\mu_r \leftarrow \mu_r + g \cdot I_r[x+i, y+j]$

$p \leftarrow p + g$

end for

$\mu_o \leftarrow \mu_o/p$

$\mu_r \leftarrow \mu_r/p$

$\sigma_{oo} \leftarrow 0, \sigma_{rr} \leftarrow 0, \sigma_{or} \leftarrow 0$

$p \leftarrow 0$

for $i = -5$ to 5 , $j = -5$ to 5 **do**

$g \leftarrow \exp\left(\frac{i^2+j^2}{2\sigma^2}\right)$

$\sigma_{oo} \leftarrow g \cdot (I_o[x+i, y+j] - \mu_o)^2$

$\sigma_{rr} \leftarrow g \cdot (I_r[x+i, y+j] - \mu_r)^2$

$\sigma_{or} \leftarrow g \cdot (I_o[x+i, y+j] - \mu_o)(I_r[x+i, y+j] - \mu_r)$

$p \leftarrow p + g$

end for

$\sigma_{oo} \leftarrow \sigma_{oo}/p$

$\sigma_{rr} \leftarrow \sigma_{rr}/p$

$\sigma_{ro} \leftarrow \sigma_{ro}/p$

$SSIM \leftarrow SSIM + \frac{(2\mu_o\mu_r+k_1)(2\sigma_{or}+k_2)}{(\mu_o^2+\mu_r^2+k_1)(\sigma_{oo}+\sigma_{rr}+k_2)}$

end for

$SSIM \leftarrow SSIM/(M \times N)$

return $SSIM \times 100$

La comparaison au tableau 4.II montre que la méthode proposée donne une valeur MSSIM très proche de la méthode [21], qui est plus élevée que celles des méthodes [20] et [35]. Notez que la méthode [13] donne une valeur MSSIM beaucoup plus élevée, mais cette méthode ne conserve pas le ton local (elle donne une valeur PSNR très basse). A cause de cela, les images reconstruites par cette méthode sont très structurées, et elles ne sont pas satisfaisantes pour l'œil (voir les images dans la section 4.1). La méthode proposée et la méthode [21] offrent un meilleur compromis entre PSNR et MSSIM : des détails fins sur l'image originale sont préservés, et en même temps, des tons locaux sont aussi préservés, ce qui confirme l'observation visuelle.

| MSSIM | | | | | |
|----------|----------------------|----------------------|---------------------|---------------------|---------------------|
| Image | Ostromoukhov 2001 | Zhou et Fang 2003 | Kwak et al. 2006 | Pang et al. 2008 | Méthode proposée |
| Bras | 48.563 | 50.361 | 56.726 | 54.666 | 55.056 |
| Buisson | 5.603 | 6.478 | 15.189 | 11.841 | 10.941 |
| Chat | 6.488 | 5.754 | 12.807 | 11.714 | 10.827 |
| Chemin | 18.345 | 18.274 | 31.935 | 28.588 | 26.367 |
| Genou | 38.560 | 40.530 | 47.950 | 46.145 | 44.709 |
| Homme | 38.540 | 37.808 | 43.840 | 45.738 | 39.964 |
| Lion | 4.500 | 4.485 | 7.505 | 7.148 | 6.375 |
| Oiseau | 67.788 | 68.955 | 72.500 | 71.900 | 71.074 |
| Osier | 25.634 | 26.709 | 31.741 | 25.621 | 29.510 |
| Portrait | 17.673 | 17.684 | 30.502 | 26.450 | 24.292 |
| Spirale | 38.518 | 39.512 | 43.855 | 43.048 | 40.712 |
| Taupe | 5.165 | 5.318 | 10.141 | 9.015 | 7.756 |
| Zèbres | 12.094 | 12.083 | 17.459 | 19.224 | 16.046 |
| Moyenne | 25.190 | 25.688 | 32.473 | 30.854 | 29.510 |

Tableau 4.II – Comparaison de la métrique MSSIM entre la méthode proposée et des méthodes récentes. Le MSSIM de la méthode proposée est très proche du MSSIM de [21], et il est plus élevé que ceux des méthodes de diffusion d'erreur standard [13, 20, 35], parce qu'elle se concentre sur la préservation des structures.

4.2.3 Temps de traitement

Selon les sections précédentes, nous avons vu que la méthode proposée est capable de reconstruire des images très proches visuellement et numériquement de la méthode de Pang et al. [21]. Un autre critère très important à considérer est le temps de traitement sur l'image originale qui est un des défauts de la méthode [21], surtout parce qu'une méthode de rendu en demi-ton est normalement utilisée dans le contexte de l'imprimerie. Un temps de traitement supérieur à quelques secondes sur une image moyenne (typiquement 512×512) n'est pas très souhaitable pour les usagers.

Le tableau 4.III compare les temps de traitement sur la méthode proposée et des méthodes récentes. Les temps de traitement de la méthode proposée sont indépendants des temps de pré-traitement (calibration). Ce temps est un peu plus élevé que pour les méthodes de diffusion d'erreur standard (environ trois fois plus que la méthode [20][35] et une et demi plus que la méthode [13]). Cependant il est beaucoup plus faible que la méthode [21] (une centaine de fois plus rapide).

Pour la méthode proposée, nous avons calculé les fréquences locales en tâches parallèle, en utilisant un GPU GeForce 9800 GTX 512MB. Une fenêtre de taille 16×16 a été utilisée. Nous avons obtenu une vitesse de traitement d'environ 3s/Mpixel (0.75s pour une image de 512×512), qui est à peu près une à deux fois le temps de rendu en demi-ton. Cette partie du temps peut être réduit par la réduction de la taille de la fenêtre de traitement (ce qui va légèrement affecter la qualité de l'image reconstruite), ou par l'utilisation d'une autre technique pour trouver l'orientation locale, le contraste local et la fréquence locale.

| Temps de traitement | | | | | |
|---------------------|----------------------|----------------------|---------------------|---------------------|---------------------|
| Image | Ostromoukhov 2001 | Zhou et Fang 2003 | Kwak et al. 2006 | Pang et al. 2008 | Méthode proposée |
| Bras | 0.045 | 0.084 | 0.106 | 87.735 | 0.140 |
| Buisson | 0.123 | 0.246 | 0.294 | 231.053 | 0.368 |
| Chat | 0.266 | 0.520 | 0.650 | 493.648 | 0.831 |
| Chemin | 0.363 | 0.706 | 0.869 | 711.352 | 1.190 |
| Genou | 0.039 | 0.070 | 0.092 | 77.771 | 0.120 |
| Homme | 0.277 | 0.535 | 0.679 | 588.774 | 0.896 |
| Lion | 0.562 | 1.082 | 1.342 | 1113.166 | 1.730 |
| Oiseau | 0.671 | 1.373 | 1.636 | 1467.180 | 2.090 |
| Osier | 0.097 | 0.183 | 0.239 | 194.444 | 0.300 |
| Portrait | 0.221 | 0.428 | 0.605 | 441.512 | 0.686 |
| Spirale | 0.110 | 0.212 | 0.269 | 253.292 | 0.335 |
| Taupe | 0.406 | 0.810 | 0.997 | 822.101 | 1.266 |
| Zèbres | 0.074 | 0.140 | 0.181 | 141.232 | 0.233 |
| Moyenne | 0.250 | 0.491 | 0.612 | 509.481 | 0.783 |

Tableau 4.III – Comparaison des temps de traitement entre la méthode proposée et des méthodes récentes. Parce que la méthode proposée fait partie des méthodes de la famille de diffusion d’erreur, son temps de traitement est beaucoup plus faible que la méthode [21], et il est très proche de ceux des méthodes de diffusion d’erreur standard [20, 35].

CHAPITRE 5

CONCLUSION

Dans ce mémoire, nous avons présenté plusieurs méthodes de rendu en demi-ton. La méthode proposée, sensible à la structure, combine des avantages de plusieurs méthodes récentes, tels que présents dans la méthode de Ostromoukhov [20] et de celle de Zhou et Fang [35]. Elle est basée sur la méthode relativement rapide de diffusion d'erreur. En même temps elle a la capacité de reconstruire l'image de demi-ton avec des détails fins. Plusieurs résultats et des évaluations numériques nous ont montré que la méthode proposée peut reproduire des images avec une qualité similaire à la méthode de Pang et al. [21], selon de nombreux de tests sur des images naturelles et artificielles. Nous avons publié la méthode proposée et nos résultats sous forme d'un article [5] dans la conférence *SIGGRAPH Asia 2009*.

5.1 Limitations

Bien que la méthode proposée puisse reproduire une grande variété d'images en très peu de temps, comme nous avons vu dans le chapitre précédent, elle souffre aussi de quelques défauts. Un défaut que nous avons observé dans la méthode proposée est qu'elle n'est pas capable de bien rendre une image qui contient plusieurs contenus fréquents principaux dans un seul pixel, parce que la méthode proposée se base sur l'hypothèse suivant : si nous pouvons reconstruire l'image artificielle qui contient seulement le signal sinusoïdal du pixel courant, la méthode peut reconstruire une image en demi-ton parfaitement. Mais dans une image complexe, il peut arriver qu'un pixel contienne plus d'un signal principal (figure 5.1). Dans cette situation, la méthode proposée va juste choisir un des signaux principaux et essayer d'augmenter l'amplitude de ce signal afin qu'il puisse être bien visualisé sur l'image reconstruite. Cependant l'autre signal sera complètement ignoré.

Pour répondre à ce problème, il faut utiliser une technique plus complexe, capable

de trouver plusieurs contenus fréquentiels principaux pour chaque pixel. Dans le processus de diffusion d'erreurs, il faut alors combiner l'information de ces contenus pour construire des seuils et des coefficients pour de multiples signaux en même temps.

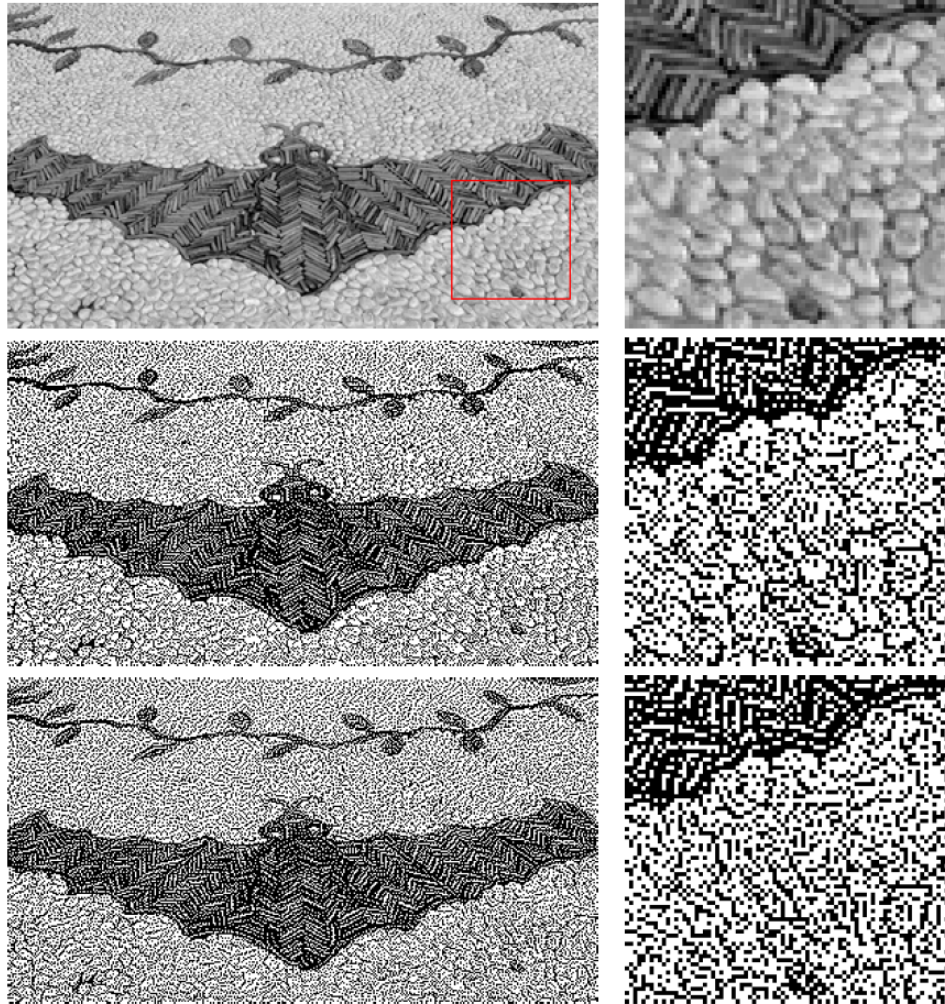


Figure 5.1 – Limitation de la méthode proposée. La méthode proposée n'est pas capable de reconstruire cette image parce qu'elle contient deux fréquences dominantes qu'il faut amplifier. (a) Image originale. (b) L'image reconstruite par la méthode [21]. (c) L'image reconstruite par la méthode proposée. Notez que la fréquence à amplifier est aléatoirement choisie, ce qui cause des structures qui ne sont pas assez apparentes en comparaison avec la méthode de Pang et al. [21]

5.2 Travaux futurs

Il existe plusieurs avenues pour améliorer la méthode proposée : une meilleure technique pour trouver les multiples contenus fréquentiels comme nous avons expliqué dans la section précédente ; une autre façon de générer des coefficients de diffusion au lieu d'utiliser simplement un filtre gaussien, etc. Mon collaborateur Benoît Alain a décrit une méthode qui génère les coefficients en utilisant la fonction Bessel, et qui donne un spectre de fourier bien meilleur que le filtre gaussien [1]. Pour simplifier les travaux, nous avons juste testé la méthode proposée avec des images en niveaux de gris, mais pas en couleur. Le résultat peut être différent pour des images en couleur parce qu'elles contiennent plus de canaux et il est possible que certains problèmes surgissent si on les traite simplement indépendamment.

BIBLIOGRAPHIE

- [1] Benoît Alain. Rendu d'images en demi-tons par diffusion d'erreur sensible à la structure. M.sc. thesis, Département d'Informatique et Recherche Opérationnelle, Université de Montréal, Décembre 2009.
- [2] M. Analoui et J. P. Allebach. Model based halftoning using direct binary search. *SPIE*, 1666 :96–108, 1992.
- [3] B. E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. *IEEE Intl. Conf. on Communications*, 1 :2611–2615, 1973.
- [4] Ronald N. Bracewell. *Fourier Transform and Its Applications*. McGraw-Hill Education, 1980.
- [5] Jianghao Chang, Benoît Alain, et Victor Ostromoukhov. Structure-Aware Error Diffusion. *ACM Transactions on Graphics*, 28(5) :162 :1–162 :8, Décembre 2009. Proceedings of SIGGRAPH-ASIA 2009.
- [6] R. Eschbach et K. T. Knox. Error-diffusion algorithm with edge enhancement. *JOSA (A)*, 8(12) :1844–1850, 1991.
- [7] Z. Fan. A simple modification of error diffusion weights. *IS&T 46th Annual Conf.*, pages 113–115, 1993.
- [8] R. W. Floyd et L. Steinberg. An adaptive algorithm for spatial grey scale. *Proc. Soc. Inf. Display*, 17 :75–77, 1976.
- [9] Byong-Won Hwang, Tae-Ha Kang, et Tae-Seung Lee. Improved edge enhanced error diffusion based on first-order gradient shaping filter. Dans *IEA/AIE'2004 : Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, pages 473–482. Springer Springer Verlag Inc, 2004.
- [10] Bernd Jahne. *Practical Handbook on Image Processing for Scientific and Technical Applications, Second Edition*. CRC Press, Inc., Boca Raton, FL, USA, 2004.

- [11] H.R. Kang. *Digital Color Halftoning*. SPIE Press, 1999.
- [12] K. T. Knox. Edge enhancement in error diffusion. Dans *Advance Printing of Paper Summaries, SPSE's 42nd Annual Conference*, pages 310–313, Boston, MA, Mai 1989.
- [13] Nae-Joung Kwak, Soung-Pil Ryu, et Jae-Hyeong Ahn. Edge-enhanced error diffusion halftoning using human visual properties. Dans *ICHIT '06 : Proceedings of the 2006 International Conference on Hybrid Information Technology*, pages 499–504, Washington, DC, USA, 2006. IEEE Computer Society.
- [14] Peter J. M. van Laarhoven, Emile H. L. Aarts, et Jan Karel Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1) :113–125, 1992.
- [15] Changhyung Lee et Jan P. Allebach. The hybrid screen : improving the breed. volume 6493, pages 435–450. SPIE, 2007.
- [16] G. Marcu. An error diffusion algorithm with output position constraints for homogeneous highlights and shadow dot distribution. *Journal of Electronic Imaging*, 9(1) :46–51, 2000.
- [17] G. Marcu et S. Abe. Halftoning by back error compensation. *IS&T NIP*, 12 :132–135, 1996.
- [18] T. Mitsa et K. J. Parker. Digital halftoning using a blue-noise mask. *Journal of the Optical Society of America A*, 9(11) :1920–1929, 1992.
- [19] J. B. Mulligan et A. J. Ahumada. Principled halftoning based on human vision models. 1992.
- [20] Victor Ostromoukhov. A simple and efficient error-diffusion algorithm. Dans *Proceedings of ACM SIGGRAPH 2001*, pages 567–572, Août 2001.
- [21] Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, et Pheng-Ann Heng. Structure-aware halftoning. *ACM Transactions on Graphics (SIGGRAPH 2008 issue)*, 27(3) :89 :1–89 :8, 2008.

- [22] Thrasyvoulos N. Pappas, Jan P. Allebach, et David L. Neuhoff. Model-Based Digital Halftoning. *IEEE Signal Processing Magazine*, 20(4) :14–27, Juillet 2003.
- [23] Thrasyvoulos N. Pappas et David L. Neuhoff. Least-squares model-based halftoning. volume 1666, pages 165–176. SPIE, 1992.
- [24] Seung-Ho Park, Ki-Min Kang, et Choon-Woo Kim. Estimation of error diffusion kernel using genetic algorithm. *SPIE 3300*, pages 330, 340.
- [25] J. Shiau et Z. Fan. A set of easily implementable coefficients in error diffusion with reduced worm artifacts. *SPIE*, 2658 :222–225, 1996.
- [26] J. Shu. Adaptive filtering for error diffusion quality improvement. *SID Dig. Tech. Papers*, pages 833–836, 1995.
- [27] E. P. Simoncelli et W. T. Freeman. The steerable pyramid : a flexible architecture for multi-scale derivative computation. Dans *ICIP '95 : Proceedings of the 1995 International Conference on Image Processing (Vol. 3)*, Washington, DC, USA, 1995. IEEE Computer Society.
- [28] P. Stucki. Mecca-a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction, 1981. Research Report RZ1060, IBM Res. Lab.
- [29] R. Ulichney. *Digital Halftoning*. MIT Press, 1987.
- [30] R. Ulichney. The void-and-cluster method for dither array generation. *SPIE*, 1913 :332–343, 1993.
- [31] Zhou Wang, A.C. Bovik, H.R. Sheikh, et E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4) :600–612, April 2004.
- [32] I. H. Witten et R. M. Neal. Using peano curves for bilevel display of continuous-tone images. *IEEE Computer Graphics & Appl.*, 2 :47–52, 1982.

- [33] Ping W. Wong et Jan P. Allebach. Optimum error-diffusion kernel design. volume 3018, pages 236–242. SPIE, 1997.
- [34] Günther Wyszecki et W. S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulae (Wiley Series in Pure and Applied Optics)*. Wiley-Interscience, 2e édition, Août 2000.
- [35] Bingfeng Zhou et Xifeng Fang. Improving mid-tone quality of variable-coefficient error diffusion using threshold modulation. *ACM Transactions on Graphics*, 22(3) :437–444, Juillet 2003.