DIRO
IFT 6150

TRAVAIL PRATIQUE N° 3

DÉCONVOLUTION & RESTAURATION

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle.
http : //www.iro.umontreal.ca/∼mignotte/ift6150
e-mail : mignotte@iro.umontreal.ca

INTRODUCTION

Si $f(x, y)$ représente l'image originale, $g(x, y)$ l'image convoluée (i.e., floue) et bruité, et $h(x, y)$[1] la fonction floue (ou PSF[3]), ce processus de dégradation peut s'exprimer par le modèle linéaire suivant :

$$g(x, y) = f(x, y) * h(x, y) + n(x, y), \tag{1}$$

où $n(x, y)$[2] représente le bruit additif gaussien et où $*$ désigne l'opérateur de convolution discret.

(**a**) (**b**)

FIG. 1 – (**a**) *Image originale* **photograph.pgm**. (**b**) *Image convoluée avec un flou uniforme 5*5 et dégradée par du bruit gaussien de variance* σ^2=20.0.

Retrouver $f(x, y)$ s'appelle ici une **restauration**. Dans le cas ou $n(x, y)$ est nul, le processus qui permet de retrouver l'image originale à partir de l'image floue (mais non bruitée) s'appelle une **déconvolution**.

[1]Le flou uniforme 2D est utilisé dans beaucoup de simulations de restauration d'images. Ce modèle de flou est celui, par exemple, qui approxime une erreur de mise au point d'appareils optiques (appareil photographique, caméra, télescope, etc.). La réponse impulsionnelle de cette dégradation peut s'exprimer par la relation suivante :

$$h(x, y) \;=\; \begin{cases} \frac{1}{L^2} & \text{si } -\frac{L}{2} \le x, y \le \frac{L}{2}, \\ 0 & \text{autrement.} \end{cases}$$

[2]Utiliser la fonction add_gaussian_noise.
[3]Point Spread Function.

Dans le cas où $n(x,y)$ est non nul mais que $h(x,y)=\delta$ (*i.e.,* $g(x,y)$ est bruitée mais non floue), retrouver $f(x,y)$ à partir de $g(x,y)$ s'appelle une opération de **débruitage** (denoising).

Le TP consiste à implémenter trois techniques de restauration/déconvolution/débruitage. Il s'agit de l'algorithme de Landweber [1], du seuillage des coefficients d'ondelettes et de la méthode récemment proposée par M. A. T. Figueiredo et Nowak [2].

A. DÉCONVOLUTION À L'AIDE DE L'ALGORITHME DE LANDWEBER

L'approche itérative proposée par Landweber *et al.* [1] permet de trouver la solution $\hat{f}(x,y)$ qui minimise la fonction de coût suivante[5] :

$$\hat{f}(x,y) = \arg\min_{f} \left\{ \| \, g(x,y) - h(x,y) * \hat{f}(x,y) \, \|^2 \right\}, \tag{2}$$

Cette technique conduit à l'approche itérative suivante :

$$\begin{aligned} & \hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) \; + \alpha \, h(-x,-y) * \Big(g(x,y) - h(x,y) * \hat{f}_k(x,y) \Big), \\ \text{avec} \quad & \hat{f}_0(x,y) = g(x,y). \end{aligned} \tag{3}$$

Cet algorithme, appelé aussi "one step gradient", conduit à trouver l'estimée $\hat{f}(x,y)$ itérativement par une technique de descente du gradient avec $g(x,y)$ comme solution initiale et α comme pas. Noter que dans le cadre de ce travail, $h(x,y)$ étant symétrique, on a $h(-x,-y)=h(x,y)$. Comme cet algorithme utilise l'opérateur de convolution $*$, celle-ci peut être réalisée efficacement dans le domaine de Fourier en utilisant une FFT.

(1) Réaliser une déconvolution de l'image **photograph.pgm** à laquelle vous aurez ajouté un flou à l'aide d'un filtre carré $h(x,y)$ dont la taille sera spécifiée par l'utilisateur. On prendra $\alpha=1$. Calculer pour chaque itération le ISNR obtenu. Pour votre bénifice, visualiser le résultat obtenu après $10, 20, 30, 40, 50, 100$ itérations.

$$\text{ISNR} = 10 \log_{10} \left(\frac{\sum_x \sum_y \big(f(x,y) - g(x,y) \big)^2}{\sum_x \sum_y \big(f(x,y) - \hat{f}(x,y) \big)^2} \right),$$

(2) L'algorithme de Landweber peut aussi servir à restaurer une image. Ainsi, ajouter un bruit gaussien à l'image convoluée et réaliser une restauration de l'image **photograph.pgm** floue et bruitée par la technique de Landweber. On prendra $\alpha=1$. Calculer pour chacune de ces itérations le ISNR obtenu (itérations comprises entre 0 et 50 par exemple).

TRANSFORMÉE EN ONDELETTES DE HAAR

Introduction

La transformée en ondelettes [3] est un outil qui appartient au domaine du traitement du signal. Son ancêtre est la transformée de Fourier, que vous connaissez déjà et qui permet, après décomposition du signal sur un ensemble de signaux de base (ensemble de cosinus pour la DCT ou exponentielle complexes pour la DFT formant une base orthogonale), d'explorer la composition fréquentielle du signal et par ses propriétés, de lui appliquer des opérateurs de filtrage.

---

[4] BSNR : Blurred Signal to Noise Ratio.

[5] En pratique, on doit préalablement estimer ce flou à partir de l'image originale avant de réaliser toute déconvolution.

Décomposition en ondelettes d'un signal 1D

De manière intuitive, la décomposition en ondelettes permet de séparer un signal en bandes de fréquences. A cette fin, on va créer une banque de filtres passe-bandes à partir d'un filtre passe-bas et d'un filtre passe-haut. La décomposition revient à représenter le signal d'une manière différente à chaque étape. Nous allons désigner un signal numérique 1D de longueur 2^n par le vecteur $\vec{s} = \{s_0, s_1, \ldots, s_{2^n-1}\}$. Durant la première étape de la décomposition, le signal $\vec{s}$ est transformé comme suit :

$$\vec{s} = \{s_0, s_1, \ldots, s_{2^n-2}, s_{2^n-1}\} \rightarrow \Big(\{m_0^1, \ldots, m_{2^{n-1}-1}^1\}, \{d_0^1, \ldots, d_{2^{n-1}-1}^1\}\Big) = (\vec{m}^1, \vec{d}^1),$$

où $\vec{m}^1$ est le vecteur des valeurs *moyennes*, $\vec{d}^1$ est le vecteur des valeurs *détails* (i.e., les coefficients d'ondelettes) et où $\rightarrow$ indique un niveau de décomposition. Dans la décomposition sur la base d'ondelettes de Haar, les détails et les moyennes sont reliés au signal par les formules suivantes :

$$\forall\ 0 \leq k \leq p, \qquad m_k^1 = \frac{s_{2k} + s_{2k+1}}{\sqrt{2}} \quad \text{et} \quad d_k^1 = \frac{s_{2k} - s_{2k+1}}{\sqrt{2}}.$$

Une fois la première transformation réalisée, le processus de réarrangement est réitéré sur les vecteurs des moyennes jusqu'à épuisement des termes : en considérant le vecteur initial des moyennes $\vec{s} = \vec{m}_0 = \{m_0^0, \ldots, m_{2^n-1}^0\}$:

$$\begin{array}{lll}
\vec{m}^0 & \rightarrow & \Big(\{m_0^1, m_1^1, \ldots, m_{2^{n-1}-1}^1\}, \{d_0^1, d_1^1, \ldots, d_{2^{n-1}-1}^1\}\Big) = (\vec{m}^1, \vec{d}^1) \\
\vec{m}^1 & \rightarrow & \Big(\{m_0^2, m_1^2, \ldots, m_{2^{n-2}-1}^2\}, \{d_0^2, d_1^2, \ldots, d_{2^{n-2}-1}^2\}\Big) = (\vec{m}^2, \vec{d}^2) \\
 & \ldots & \qquad\qquad \ldots \qquad\qquad \ldots \\
\vec{m}^{M-1} & \rightarrow & \Big(\{m_0^M, \ldots, m_{2^{n-M}-1}^M\}, \{d_0^M, \ldots, d_{2^{n-M}-1}^M\}\Big) = (\vec{m}^M, \vec{d}^M)
\end{array}$$

où M est le niveau de résolution ou de décomposition ($M = 3$ à la Fig. 2 (b)). En d'autres mots, après une étape de décomposition, le signal initial $\vec{s}$ est décomposé en deux vecteurs $\vec{m}^1$ et $\vec{d}^1$, tout deux de longueur 2^{n-1}. Par la suite, le vecteur $\vec{m}^1$ est décomposé à son tour en vecteurs $\vec{m}^2$ et $\vec{d}^2$, cette fois-ci de longueur 2^{n-2}. Cette opération est répété successivement M fois.

Lors des transformations successives, les vecteurs de moyennes $\vec{m}^i$ sont remplacés par leur transformation $(\vec{m}^{i+1}, \vec{d}^{i+1})$,

$$\vec{s} = \vec{m}^0 \rightarrow (\vec{m}^1, \vec{d}^1) \rightarrow ((\vec{m}^2, \vec{d}^2), \vec{d}^1) \rightarrow \ldots \rightarrow ((\ldots(\vec{m}^M, \vec{d}^M), \vec{d}^{M-1}), \ldots), \vec{d}^1)$$

A l'issue de la dernière transformation, le signal d'origine s est donc représenté par le terme final de moyenne m^M et tous les termes de détails,

$$\{d_0^M\}, \{d_0^{M-1}, d_1^{M-1}\}, \ldots \{d_0^2, d_1^2, \ldots, d_{2^{n-2}-1}^2\}, \{d_0^1, d_1^1, \ldots, d_{2^{n-1}-1}^1\}.$$

La reconstruction du signal se fera elle aussi de manière itérative en recomposant à partir de la dernière transformation, les moyennes du rang $p-1$ étant calculées à partir des moyennes et des détails du rang p. Dans le cas de l'ondelette de Haar, on a,

$$\begin{aligned}
m_{2k}^{p-1} &= \frac{m_k^p + d_k^p}{\sqrt{2}} \\
m_{2k+1}^{p-1} &= \frac{m_k^p - d_k^p}{\sqrt{2}}
\end{aligned}$$

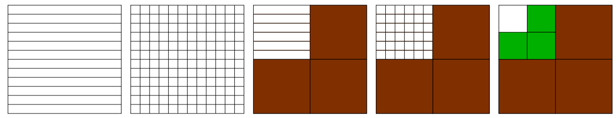

FIG. 2 – *Stratégie de décomposition en ondelettes pour une image.*

Décomposition en ondelettes d'une image

La décomposition en ondelettes d'une image (dont les dimensions sont des puissances de deux) revient à réaliser itérativement un niveau de décomposition selon les lignes puis un niveau de décomposition selon les colonnes puis recommencer sur l'image des valeurs moyennes et ainsi de suite (cf Figure 2). Le code servant à décomposer une image en ondelettes de Haar est disponible sur le site web du TP.

(**a**) (**b**)

FIG. 3 – *(**a**) Image originale* **photograph.pgm**. *(**b**) Décomposition de l'image* **photograph.pgm** *en ondelettes de Haar sur trois niveaux de résolution.(**c**) Ondelettes de l'image.*

B. DÉBRUITAGE PAR SEUILLAGE DES COEFFICIENTS D'ONDELETTES

Lorsqu'une image est décomposée en M niveaux d'ondelettes, chaque vecteur de coefficients $\vec{d}_i^k, \vec{d}_j^k, \vec{d}_{ij}^k$ (c.f. Figure 3 c) possède des informations dites *spatio fréquentielles* sur l'image d'entrée. En général, plus k est petit et plus $\vec{d}^k$ possède des informations de hautes fréquences. Ainsi, considérant que le bruit est généralement une information contenue dans les hautes fréquences, il est possible de *débruiter* une image en filtrant les coefficients d'ondelettes appartenant aux niveaux les plus bas.

(1) Vous devez réduire le bruit d'une image en seuillant ses coefficients d'ondelettes d'une manière similaire à celle présentée dans le manuel du cours [4, chap 7.5 p.391-92]. Pour ce faire, vous devez ajouter du bruit gaussien à la figure **photograph**, décomposer l'image bruitée en ondelettes de Haar, appliquer un seuil *flou* (*soft thresholding*) sur les coefficients des $\mathcal{N}$ niveaux les plus bas et reconvertir les ondelettes en une image. Vous devez conclure cette procédure en affichant la valeur de l'ISNR. La procédure de seuil flou doit être la suivante :

$$d_\beta^k(x,y) = \begin{cases} 0 & \text{si } |d_\beta^k(x,y)| < \tau \\ 255 \cdot \frac{d_\beta^k(x,y)}{|d_\beta^k(x,y)|} \cdot \frac{(|d_\beta^k(x,y)|-\tau)}{255-\tau} & \text{sinon} \end{cases} \tag{4}$$

où $\beta \in \{a,b,c\}$ et $k \in [0,\mathcal{N}[$. Noter que $\mathcal{N}$ et τ doivent être spécifiés par l'utilisateur.

C. RESTAURATION AVEC ALGORITHME DE FIGUEIREDO ET NOWAK

En supposant que $h(x,y)$ est connu, le problème de la restauration d'image est un problème que l'on sait être *mal posé*. Cette expression signifie que la solution de ce problème n'est pas unique et qu'une très faible variation sur l'image $g(x, y)$, un bruit par exemple, a pour effet de produire de grandes variations sur la solution estimée (i.e., l'image restaurée). Une solution pour rendre ce problème *bien posé* (i.e., avec une solution unique) consiste à exploiter dans la procédure de restauration une solution que l'on sait être *pas trop éloignée* du résultat désiré[6]. Une astuce complémentaire consiste à modéliser puis exploiter des informations *a priori* que l'on a sur l'image originale (*via* un terme de régularisation) et contraindre la solution de ce problème[7]. Dans ce sens, les techniques itératives ont été proposées. Ces dernières ont aussi l'avantage de ne pas réclamer l'implémentation d'une division comme dans le cas du filtre inverse de Wiener. De plus, le processus de restauration peut être contrôlé et visualisé à chaque itération.

L'algorithme proposé par Figueiredo et Nowak [2] consiste à alterner et répéter, jusqu'à convergence, les deux étapes suivantes :

1. Une étape de déconvolution réalisée une iteration de l'algorithme de Landweber exprimée par l'Eq. (3) (cf. Section précédente A).
2. La deuxième étape permet de contraindre le problème inverse mal posé de la restauration et utilise la transformée de l'image obtenue précédemment (i.e., déconvoluée a l'étape précédente) en ondelettes discrètes. Plus précisément, cette étape de débruitage est réalisée par un filtrage des coefficients d'ondelettes.

Les coefficients en ondelettes $\hat{\theta}$ de l'image déconvoluée sont filtrés en utilisant un seuil non linéaire. Si $\hat{\theta} = (\hat{\theta}_1,\, \hat{\theta}_2, \ldots, \hat{\theta}_N)$ représente le vecteur des N coefficients après filtrage et $\hat{w}$ le vecteur des coefficients avant filtrage, cette opération est de la forme :

$$\hat{\theta}_i = \frac{\Big((\hat{w}_i)^2 - 3\alpha^2 \Big)_+}{\hat{w}_i}$$

ou l'opérateur $(.)_+$ dénote l'opérateur de partie positive, définie par $(x)_+ = \arg\max\{x, 0\}$. D'autres variantes de seuil sont possibles [2].

(1) Ajouter un bruit gaussien à l'image convoluée par un filtre carré dont la taille aura été spécifiée par l'utilisateur. Réaliser une restauration de l'image **photograph.pgm** floue et bruitée par la technique de Figueiredo et Nowak (cf. Algo. 1). On prendra $\alpha^2 = \sigma^2$ et $M = 3$ niveaux de résolution pour la décomposition en ondelettes de Haar. On prendra comme configuration initiale, une estimée donnée par l'algorithme de Landweber (cf. Section B). Afficher pour chacune de ces itérations le ISNR obtenu (itérations comprises entre 0 et 50 par exemple).

Bibliographie

[1] L. Landweber. An iterative formula for fredholm integral equations of the first kind. *American Journal of Mathematic*, 73 :615–624, 1951.

[2] M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restauration. *IEEE Transaction on Image Processing*, 12(8) :906–916, 2003.

[3] G. Amara. An Introduction to Wavelets, IEEE Computational Science and Engineering, vol. 2, num. 2, IEEE Computer Society, 1995 : 383-390.

[4] Gonzalez R and Woods R. Digital Image Processing, 2nd Ed, Addison-Wesley Longman Publishing Co., Inc.,1992.

REMISE

[6] Le plus souvent, c'est l'image floue $g(x, y)$ elle même.

[7] Le plus souvent, on modélise le fait que l'image originale $f(x, y)$ n'a pas trop de hautes fréquences (dans le domaine de Fourier ou celui des ondelettes), que les voisins des pixels de cette image doivent avoir un niveaux de gris similaire, etc.

□ **ALGORITHME RESTAURATION EM DE FIGUEIREDO ET NOWAK**

$\hat{f}_0(x,y) = \hat{f}_{\text{Landweber}}(x,y)$: Configuration initiale

1. Déconvolution du type Landweber :

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y) + \frac{\alpha^2}{\sigma^2}\, h(-x,-y) * \Big(g(x,y) - h(x,y) * \hat{f}_k(x,y)\Big)$$

2. Filtrage dans le domaine des ondelettes :

   - Décomposition en ondelettes de $\hat{f}_{k+1}(x,y)$:
   $$\hat{w} \leftarrow \mathbf{DWT}\ [\hat{f}_{k+1}(x,y)]$$
   - threshold :
   $$\hat{\theta}_i = \frac{\Big(\big(\hat{w}_i\big)^2 - 3\alpha^2\Big)_+}{\hat{w}_i}$$
   - Décomposition en ondelettes inverse de $\hat{\theta}$:
   $$\hat{f}_{k+1}(x,y) \leftarrow \mathbf{DWT}^{-1}\ [\hat{\theta}]$$

3. Retour en 1. et $k \leftarrow k+1$ tant que :

$$\frac{\parallel \hat{f}_{k+1}(x,y) - \hat{f}_k(x,y) \parallel^2}{\parallel \hat{f}_k(x,y) \parallel^2} > 10^{-5}$$

TAB. 1 – Algorithme de Restauration de Figueiredo et Nowak.

DATE DE REMISE ET CONDITIONS

Vous devez rendre électroniquement les programmes fait en C (fichier .c et exécutable ainsi que la ligne de compilation ou le makefile) avant la date de remise spécifiée dans le fichier *barème* situé sur la page web du cours dans la rubrique *Introduction & Program.* Utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP<Numéro du Tp>. N'oubliez pas d'inscrire vos noms, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *TpIFT6150– <Numéro du Tp>-<Numéro de la question>.c.* Les programmes devront se compiler et s'exécuter sur Linux. Noter qu'il n'y a pas de rapport à remettre au terme de ce travail.