

Travail pratique n° 4

Analyse multirésolution

Instructions

- Les travaux pratiques peuvent être effectués seul ou par équipe de deux *maximum*. La composition de chaque équipe doit être indiquée au professeur au plus tard lors de la séance.
- Le compte rendu doit comporter une réponse concise mais complète à chacune des questions, accompagnée au besoin des courbes, figures et images appropriées.
- Le compte rendu peut prendre deux formes : soit un fichier pdf accompagné des scripts et fonctions Matlab que vous avez développés, soit un document (pdf ou html) généré à l'aide des fonctionnalités de publication de Matlab (menu “File / Publish” de l'éditeur Matlab), accompagné de son fichier source Matlab. Dans tous les cas, l'ensemble des fichiers doit être placé dans une unique archive **zip**.
- Le compte rendu doit être remis au plus tard 7 jours après la séance en utilisant l'outil approprié disponible sur le site web du cours.
- Le travail doit être remis par un seul des membres du groupe. Si tel n'est pas le cas, la version la plus récente du travail remis est prise en compte.

1 Introduction

Ce travail pratique est consacré à l'un des principaux outils utilisés en analyse multirésolution de signaux ou d'images numériques : la transformée en ondelettes rapide. Les tâches demandées vous permettront de vous familiariser avec cet outil et de l'utiliser dans un but précis : la correction de défauts *localisés* dans une image.

Les fonctions Matlab et les données à utiliser pour effectuer ce travail pratique se trouvent dans l'archive **TP4.zip** disponible sur le site web du cours.

2 Transformée en ondelettes rapide avec Matlab

On rappelle que prendre la transformée en ondelettes rapide (TOR) 2D d'une image consiste à la décomposer en quatre éléments dont la taille est sensiblement le quart de celle de l'image originale. Trois de ces éléments sont les *coefficients de détail* (respectivement horizontal, diagonal et

vertical) et le quatrième élément est *l'approximation* de l'image originale à l'échelle immédiatement supérieure. Cette décomposition est équivalente à un codage en sous-bandes 2D, les coefficients de détail correspondant aux fréquences hautes et l'approximation aux fréquences basses. Le processus de décomposition peut être itéré *sur l'approximation* pour obtenir une décomposition multi-échelles ou espace-échelle. Pour effectuer une telle décomposition, les paramètres à spécifier sont l'ondelette (ou la fonction d'échelle) utilisée et le nombre de niveaux de décomposition (ou le nombre d'échelles).

La boîte à outils « Ondelettes » de Matlab fournit plusieurs fonctions permettant de calculer et de manipuler la TOR d'un signal ou d'une image. Dans le cas 2D, les éléments de la transformée en ondelettes 2D sont stockés dans une paire de variables (**C**, **S**) où **C** contient l'ensemble des coefficients de la transformée et où **S** est un tableau de contrôle qui permet d'accéder à chaque élément de la transformée sans ambiguïté. Des détails sur la structure de ces variables sont disponibles dans l'aide en ligne de Matlab. Dans le cadre de ce travail pratique, les trois fonctions suivantes présentent un intérêt particulier :

wavedec2	Calcul de la TOR 2D
waverec2	Calcul de la TOR 2D inverse
wfilters	Calcul des filtres de codage-décodage en sous-bandes correspondant à une ondelette donnée

De plus, l'archive **TP4.zip** fournit plusieurs fonctions permettant de manipuler et d'afficher les éléments d'une TOR 2D. Le tableau suivant en donne une description succincte :

wavecopy	Extraction d'un élément d'une structure TOR 2D
wavepaste	Insertion d'un élément dans une structure TOR 2D
wave2gray	Affichage pyramidal d'une structure TOR 2D
wavecut	Mise à zéro d'un élément d'une structure TOR 2D
wavework	Édition d'un élément d'une structure TOR 2D

Une description plus complète des fonctions **wavecopy**, **wavepaste**, **wave2gray**, **wavecut** et **wavework**, qui ne disposent pas d'aide en ligne, est donnée en annexe. Pour effectuer le travail demandé aux questions 3 et 4, il est recommandé de se limiter à l'utilisation des fonctions **wavecopy**, **wavepaste** et **wave2gray**.

Par ailleurs, la boîte à outils « Ondelettes » fournit un catalogue d'ondelettes. Dans ce travail pratique, on n'utilisera que deux types d'ondelettes : les ondelettes de Daubechies (**dbx**) et les ondelettes symétriques (**symx**). Dans la désignation des ondelettes, **x** représente leur taille, qui est égale à la moitié de la longueur du support des filtres de codage-décodage en sous-bandes. De plus, l'ondelette **db1** représente l'ondelette de Haar.

3 Transformée en ondelettes rapide d'une image (8 points)

3.1 Transformée directe et transformée inverse (4 points)

Dans un premier temps, on utilise l'ondelette de Haar. Représentez la réponse impulsionnelle des filtres de codage-décodage en sous-bandes relatifs à cette ondelette. Effectuez et représentez la

TOR de l'image `Lenna.tif` en utilisant trois niveaux d'échelle. Effectuez ensuite la TOR inverse et vérifiez que l'image reconstruite est identique à l'image originale aux erreurs numériques près.

Effectuez les mêmes opérations avec l'ondelette `db4`. La TOR révèle-t-elle des caractéristiques différentes selon l'ondelette choisie ? Expliquez.

3.2 Utilisation pour la compression (4 points)

Une approche simple de la compression d'image consiste à mettre à zéro les coefficients de détail dont la valeur absolue est inférieure à un seuil donné. Effectuez cette opération avec l'image et les ondelettes utilisées au paragraphe précédent et effectuez la TOR inverse des coefficients ainsi modifiés ; évaluez l'erreur quadratique normalisée entre l'image de départ et l'image reconstruite, ainsi qu'un « taux de compression approché » défini comme le rapport entre nombre de coefficients mis à zéro et le nombre total de coefficients de la TOR. Commentez brièvement vos résultats ainsi que l'allure des images compressées.

Remarque Pour effectuer l'opération de seuillage des coefficients, vous pourrez vous inspirer des programmes de démonstration Matlab utilisés au cours n° 8.

4 Correction de défauts localisés (12 points)

Les coefficients de la transformée en ondelettes d'une image sont liés aux caractéristiques de l'image à une position donnée et à une échelle donnée. Ces propriétés peuvent être utilisées pour corriger des défauts localisés tels que les rayures qui peuvent affecter les photographies argentiques. Le but de cette partie est de corriger des défauts de ce type, qui seront représentés par des segments de droite d'intensité maximale placés dans une position connue de l'image.

4.1 Traitement des coefficients de la décomposition (6 points)

Pour améliorer l'image, on se propose de mettre à zéro les coefficients de la décomposition que l'on peut associer à la présence d'un défaut. Pour cela, écrivez une procédure Matlab permettant de mettre à zéro les éléments d'un tableau situés à une distance inférieure à une valeur δ d'un segment de droite AB , les coordonnées des points A et B étant spécifiées en proportion des dimensions du tableau auquel ils appartiennent.

Remarque On rappelle que la distance d d'un point M de coordonnées (x, y) à la droite passant par les points A et B de coordonnées respectives (x_1, y_1) et (x_2, y_2) a pour expression :

$$d = \frac{|x(y_2 - y_1) - y(x_2 - x_1) - (x_1 y_2 - x_2 y_1)|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$$

Calculez et représentez la TOR de l'image `Lenna_r.tif` avec l'ondelette `db4`. À l'aide de la procédure Matlab que vous venez de développer, mettez à zéro les coefficients de détail que vous pouvez associer à la présence du défaut et reconstruisez l'image à partir des coefficients ainsi modifiés. Ajustez les paramètres à votre disposition (type des coefficients de détail, nombre d'échelles affectées,

étendue de la zone mise à zéro) de manière à obtenir un résultat visuellement acceptable. On précise que les extrémités du défaut affectant l'image `Lenna_r.tif` ont les coordonnées suivantes :

$$A : (0, 1, 0, 5) \quad B : (0, 9, 0, 5)$$

4.2 Effet du type d'ondelette (2 points)

Effectuez le même traitement qu'au paragraphe précédent, mais en utilisant l'ondelette de Haar, puis l'ondelette `sym6`. Dans chaque cas, ajustez les paramètres à votre disposition. Commentez les résultats obtenus.

4.3 Effet de l'orientation du défaut (2 points)

En utilisant l'ondelette `sym6`, traitez l'image `Lenna_s.tif` en ajustant de nouveau les paramètres à votre disposition. Comparez les résultats à ceux obtenus avec l'image `Lenna_r.tif`. Commentez. On précise que les extrémités du défaut affectant l'image `Lenna_s.tif` ont les coordonnées suivantes :

$$A : (0, 1, 0, 4) \quad B : (0, 9, 0, 6)$$

4.4 Effet du type d'image (2 points)

Toujours avec l'ondelette `sym6`, traitez l'image `phare_s.tif`, dont le défaut a les mêmes caractéristiques que celui présent dans l'image `Lenna_s.tif`. Que constatez-vous ? Interprétez votre résultat en vous appuyant différences entre les caractéristiques locales des images `Lenna_s.tif` et `phare_s.tif`.

Annexes

A Précisions sur les fonctions `wavecopy`, `wavepaste` et `wave2gray`

On reproduit ci-après l'entête de chacune de ces trois fonctions, qui fournit l'équivalent de l'aide en ligne normalement disponible sous Matlab.

A.1 Fonction `wavecopy`

```
function y = wavecopy(type, C, S, n)
% WAVECOPY Fetches coefficients of a wavelet decomposition structure.
%   y = wavecopy(type, C, S, n) returns a coefficient array based on
%       variables type and n
%
% INPUTS
%   type    Coefficient category
%   -----
%   'a'     Approximation coefficients
%   'h'     Horizontal detail coefficients
%   'v'     Vertical detail coefficients
```

```

%      'd'      Diagonal detail coefficients
%
%      C, S      Wavelet data structure, according to the Wavelet Toolbox
%      n          Decomposition level (scale level). Ignored if type='a'
%
%  OUTPUT
%      y          Array containing the coefficients that were copied
%

```

A.2 Fonction wavepaste

```

function NC = wavepaste(type, C, S, n, x)
% WAVEPASTE Puts coefficients into a wavelet decomposition structure.
% NC = wavepaste(type, C, S, n, x) returns the new wavelet coefficients
%      after pasting x into it based on variables type and n
%
%  INPUTS
%      type      Coefficient category
%      -----
%      'a'        Approximation coefficients
%      'h'        Horizontal detail coefficients
%      'v'        Vertical detail coefficients
%      'd'        Diagonal detail coefficients
%
%      C, S      Wavelet data structure, according to the Wavelet Toolbox
%      n          Decomposition level (scale level). Ignored if type='a'
%      x          Two-dimensional detail or approximation coefficient
%                  matrix whose dimensions are appropriate for
%                  decomposition level n
%
%  OUTPUT
%      NC          Array containing the whole set new wavelet coefficients
%

```

A.3 Fonction wave2gray

```

function W = wave2gray(C, S, scale, border)
% WAVE2GRAY Displays wavelet decomposition coefficients.
% W = wave2gray(C, S, scale, border) displays and returns a wavelet
%      coefficient image in the usual "pyramid" form
%
%  INPUTS
%      C, S      Wavelet data structure, according to the Wavelet Toolbox
%
%      scale      Detail coefficient scaling
%      -----

```

```

%      0 or 1      Maximum range (default)
%      2, 3...    Magnify default by the scale factor
%      -1, -2...  Magnify absolute values  by abs(scale)
%
%      border      Gap between wavelet decomposition elements
%      -----
%      'absorb'    Gap replaces image (default)
%      'append'    Gap increases size of image
%
%  OUTPUT
%      W           Image containing the wavelet decomposition image in
%                  "pyramid" form
%

```

A.4 Fonction wavecut

```

function [NC, y] = wavecut(type, C, S, n)
%WAVECUT Zeroes coefficients in a wavelet decomposition structure.
% [NC, y] = wavecut(type, C, S, n) returns a new decomposition
% vector whose detail or approximation coefficients (based on type
% and n) have been zeroed. The coefficients that were zeroed are
% returned in y.
%
% INPUTS:
%      type      Coefficient category
%      -----
%      'a'       Approximation coefficients
%      'h'       Horizontal details
%      'v'       Vertical details
%      'd'       Diagonal details
%
%      C, S      Wavelet data structure, according to the Wavelet Toolbox
%      n         Decomposition level (scale level). Ignored if type='a'
%
%  OUTPUTS
%      NC        Array containing the whole set new wavelet coefficients
%      y         Array containing the coefficients that were set to zero
%

```

A.5 Fonction wavework

```

function [varargout] = wavework(opcode, type, C, S, n, x)
%WAVEWORK is used to edit wavelet decomposition structures.
% [varargout] = wavework(opcode, type, C, S, n, x) gets the
% coefficients specified by type and n for access or modification
% based on opcode.

```

```

%
% INPUTS:
%   opcode      Operation to perform
%   -----
%   'copy'      [varargout] = Y = requested (via TYPE and N)
%               coefficient matrix
%   'cut'       [varargout] = [NC, Y] = New decomposition vector
%               (with requested coefficient matrix zeroed) AND
%               requested coefficient matrix
%   'paste'     [varargout] = [NC] = new decomposition vector with
%               coefficient matrix replaced by X
%
%   type        Coefficient category
%   -----
%   'a'         Approximation coefficients
%   'h'         Horizontal details
%   'v'         Vertical details
%   'd'         Diagonal details
%
%   C, S        Wavelet data structure, according to the Wavelet Toolbox
%   n           Decomposition level (scale level). Ignored if type='a'
%   x           Two-dimensional coefficient matrix for pasting.
%

```