Filtrage spatial et morphologie mathématique

1 Introduction

Les notions d'imagerie abordées en cours sont illustrées dans ce td à l'aide de matlab, outil de calcul numérique. La table 1 montre un exemple de script matlab permettant de lire, manipuler et sauvegarder des images.

Le but de ce TD est de d'illustrer la notion de filtrage temporel et les principaux opérateurs de morphologie mathématique binaire.

2 Filtrage temporel

On rappelle la formule vue en cours qui applique un filtre linéaire à une image `I` :

$$\mathtt{I_f}(x,y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} W(i,j).\mathtt{I}(x+i, y+j) \tag{1}$$

La fonction Matlab donnée sur la table 3 montre comment appliquer un filtre à une image en utilisant la fonction `masqn` (table 4). Modifiez ce script pour visualiser l'effet des filtres suivants :

2.1 Débruitage

Les filtres spatiaux sont très utilisés afin de réduire l'effet du bruit dans les images.

1. Filtre moyenneur
   $W_1 = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
2. Filtre Gaussien
   La formule générale du filtre gaussien est :
   $$W_G(i,j) = C \exp\left[-\frac{i^2+j^2}{2.\sigma^2}\right]$$

- Pour un écart type du bruit 50, comparez un filtre moyenneur de taille 3×3 avec un filtre moyenneur de taille 11×11.
- Pour un écart type du bruit 50, comparez un filtre Gaussien d'écart type $\sigma = \sqrt{2}$ avec un filtre Gaussien d'écart type $\sigma = 5$. On utilisera un support de noyau de taille $2\sigma + 1$.

2.2 Détection de contours

La détection des contours d'une image passe par le calcul de la dérivée spatiale première ou seconde de l'image. Pour le calcul de la dérivée première de l'image, les filtres les plus connus sont les **filtres de Sobel**, définis par le noyau suivant : $W_{s1} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$ Pour les contours verticaux
$W_{s2} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$ pour les contours horizontaux.

1. Montrer en quoi un filtre de Sobel approxime la dérivée spatiale d'une image.
2. Afficher, pour une image non bruitée, la valeur absolue de la réponse d'un filtre de Sobel horizontal et vertical d'une image, que l'on appelle gradient en x et en y de l'image, et noté Gx et Gy.
3. Calculez et afficher une pseudo-image de la norme du gradient, obtenue par : $||G|| = \sqrt{Gx^2 + Gy^2}$.

La détection de contour peut aussi s'effectuer en recherchant le passage à zéro avec changement de signe de la dérivée seconde d'un signal. Cette opération est approximée par un filtre Laplacien de noyau : $W_4 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

2.3 Questions supplémentaires

1. Essayez de concevoir un filtre de sobel permettant de faire apparaitre les contours diagonaux ?
2. Que se passe-t-il lorsque l'on cherche à calculer la dérivée d'une image bruité (pour le test vous appliquerez une amplitude de bruit de 50). Comparer les deux stratégies suivantes :
   (a) Calcul du module du gradient par : $||G|| = \sqrt{Gx^2 + Gy^2}$ où Gx et Gy sont obtenus en appliquant un filtre de Sobel sur l'image de départ.
   (b) Calcul du module du gradient par : $||G|| = \sqrt{Gx^2 + Gy^2}$ où Gx et Gy sont obtenus en appliquant un filtre de Sobel sur l'image de départ à laquelle on a appliqué un pré-filtrage Gaussien d'écart type $\sigma = 3$.
3. Pour la deuxième stratégie, comparez l'influence du bruit sur la norme du gradient estimée par des opérateurs de Sobel avec la réponse d'un filtre Laplacien. Quel est le filtre le plus sensible ? pourquoi ?

3 Morphologie mathématique

Nous nous limiterons, dans le cadre de ce sujet, à appliquer des opérateurs morphologiques sur des images binaires. Il existe 2 opérations élémentaires en morphologie mathématiques : l'érosion et la dilatation. A partir de ces deux opérations élémentaires, on

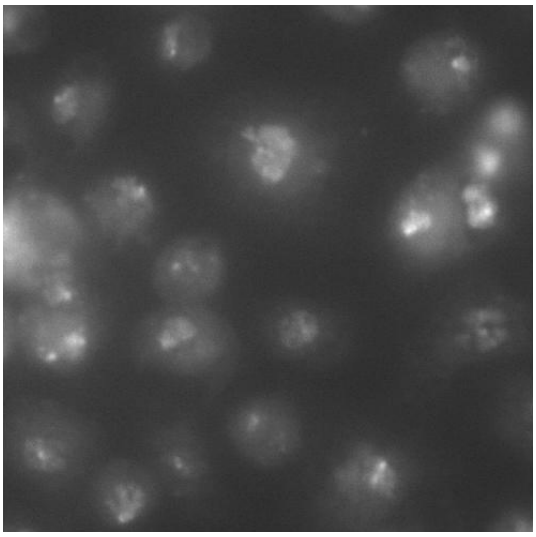

Figure 1 – Image microscopique de cellules vivantes

définit les opérateurs d'ouverture et de fermeture. On définit une image binaire `I`, un élément structurant `B` et un test d'intersection. La dilatation de `I` par `B` est l'ensemble des points P tels que `B` centré en P rencontre `I` :

$$\delta_{\mathtt{B}}(\mathtt{I}) = \{z : \mathtt{B}_z \cap \mathtt{I} \neq \emptyset\} \tag{2}$$

L'érosion de `I` par `B` est l'ensemble des points P tels que `B` centré en P est inclus dans `I` :

$$\varepsilon_{\mathtt{B}}(\mathtt{I}) = \{z : \mathtt{B}_z \subset \mathtt{I}\} \tag{3}$$

Editez le script `testmorpho.m` (table 5) et modifiez le afin d'analyser, pour les éléments structurants $ES_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$, $ES_2 = (1,1,1)$ et $ES_3 = (1,1,1)^T$ les points suivants :

1. le résultat de l'opération d'érosion (fonction `erode`),
2. le résultat de l'opération de dilatation (fonction `dilate`),
3. le résultat d'une ouverture (fonction `ouverture`),
4. le résultat d'une fermeture (fonction `fermeture`).
5. Que se passe-t-il lorsque l'on applique plusieurs fois la même érosion ou dilatation ?
6. Que se passe-t-il lorsque l'on applique plusieurs fois la même ouverture ou fermeture ? Pourquoi ?

4 Cas d'étude : segmentation de cellules

La figure 4 est une image de cellules vivantes, issue d'un microscope électronique.

On cherche à produire une image binaire, le moins bruité possible, où les cellules apparaissent en blanc. Pour cela, le schéma classique de traitement d'images est le suivants :

1. Normalisation de l'image. On applique un étirement d'histogramme (idéalement, il faudrait appliquer une égalisation d'histogramme),

2. Filtrage de l'image pour éliminer le bruit du à l'électronique du microscope,
3. Binarisation de l'image
4. Application d'opérations de morphologie mathématique pour débruiter l'image binarisée.

L'image à traiter s'appelle `cells.png`. Vous devez écrire une fonction Matlab produisant une image binaire des cellules vivantes.

```
%%%%%%%%%%%%%%%%%%%%%%%%
% script d'exemple matlab
%%%%%%%%%%%%%%%%%%%%%%%%
%
% lecture d'une image
[I,map]=imread('meadownb.jpg');
%%%%%%%%%%%%%%%%%%%%%%%%
% Affichage d'une image
image(I);
% pause il faut appuyer sur une
% touche pour reprendre
pause ;
colormap(gray(256));
%%%%%%%%%%%%%%%%%%%%%%%%
% Convertion de l'image en
% format double afin de la modifier
Id=double(I);
%%%%%%%%%%%%%%%%%%%%%%%%
% Extraction d'une partie de l'image
% des lignes 100 à 300 et
% des colonnes 200 à 400
I3=Id(100:300,200:400);
%%%%%%%%%%%%%%%%%%%%%%%%
% Sauvegarde d'une image
imwrite(uint8(I3),gray(256),'toto.jpg','JPEG');
```

Table 1 – Exemple de script matlab.

```
%%%%%%%%%%
% Exemple de fonction matlab fichier fo.m
% Les parametres d'entree sont a,b,c
% les parametres de sortie sont t1 et t2
%%%%%%%%%%
%%%%%%%%%%
% Prototype
function [t1,t2]=fo(a,b,c)
%
t1=a+b;
t2=b+c;
```

Table 2 – Exemple de fonction matlab.

```
function␣testmasque
%␣Blaise␣Pascal␣University
%␣T.␣Chateau
%␣testmasque
%␣This␣function␣loads␣an␣image,␣add␣some␣noise,
%␣applies␣a␣filter␣and
%␣display␣the␣resulting␣filtered␣image
%
%%%%%%%%%%
[x,map]=imread('Nenupharsb.jpg');
I=double(x(:,:,1));␣%␣extract␣red␣component
%␣apply␣random␣noise
[ty,tx]=size(I);
Amp=50;␣%␣Change␣here␣to␣modify␣the␣noise␣amplitude
Br=Amp.*randn(ty,tx)-Amp/2;
I=I+Br;
%␣Some␣classical␣kernels
%␣W=ones(11)./(11*11);␣%␣mean
%␣W=[-1␣-1␣-1␣;␣-1␣8␣-1␣;␣-1␣-1␣-1];␣%␣laplacian
W=␣gaus(7,2);␣%␣gaussian␣kernel␣(size␣ta␣and␣bandwidth␣sigma)
Im1=masqn(I,W);␣%␣Call␣masqn␣function
figure(1);colormap(gray);
subplot(1,2,1);␣%␣split␣the␣figure␣in␣1␣line␣and␣2␣columns
imagesc(I);
title('Initial␣image');
subplot(1,2,2);
imagesc(Im1);
title('Filtered␣image');

%␣End␣of␣main␣function

function␣W␣=␣gaus(ta,sigma)
%␣build␣a␣gaussian␣kernel
%␣ta:␣size␣of␣the␣kernel
%␣sigma:␣bandwidth
W=zeros(ta);
ta2=(ta-1)/2;
for␣i=␣-ta2:ta2
␣␣␣␣for␣j=␣-ta2:ta2
␣␣␣␣␣␣␣␣W(i+ta2+1,j+ta2+1)=exp(-(i^2+j^2)/(2*sigma^2));
␣␣␣␣end
end
W␣=␣W./sum(W(:));
```

Table 3 – Script testmasque.

```
%% ISIMA, Imagerie, T. Chateau
function [I2]=masqn(I,m)

% Calcul taille masque et image
[ty,tx]=size(m);
[Ity, Itx]=size(I);
I2=0.*I;
% agrandissement de I pour gerer les effets de bord
tmpx=round((tx-1)/2);
tmpy=round((ty-1)/2);
I=[I(1:tmpy+1,:) ; I ; I(end-tmpy:tmpy,:)];
I=[I(:,1:tmpx+1) I I(:,end-tmpx:tmpx)];
% double boucle sur le masque
for i=1:ty % lignes
    for j=1:tx % colonnes
        I2=I2+m(i,j).*I(i:end-ty+i,j:end-tx+j);
    end
end
```

Table 4 – fonction masqn.

```
% Blaise Pascal University
% T. Chateau
% testmorpho.m
% This function loads an image, translates it into a binary one
% and apply a morphological filter
%
%%%%%%%%%%
[x,map]=imread('sceneb.gif');
I=double(x(:,:,1));
% binary image
I=fix(I./255);
figure(1);colormap(gray);
% Strucuring element definition:
% It can be modified
W=ones(5);
% Call erode function
Im=erode(I,W);
%Im=dilate(Im,W);
% Display
subplot(1,2,1);
imagesc(I);
title('Initial image');
subplot(1,2,2);
imagesc(abs(Im));
title('Filtered image');
```

Table 5 – script testmorpho.