1 Matlab correction

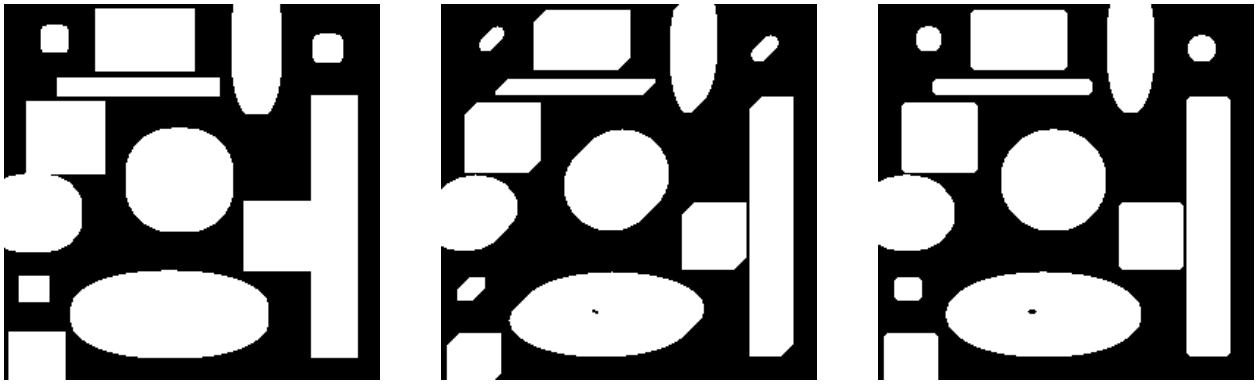
Images given as examples are stored as binary images. The structuring elements used, square, line and disk, are defined with the MATLAB® function strel, and some parameters (of size and shape).

```
% read image
B=imread('B.bmp');
% display image
figure;imshow(B,[]);title('Original image');

% Create structuring elements
se1 = strel('square',11) ;      % 15-by-15 square
se2 = strel('line',11,45);      % line, length 15, angle 45 degrees
se3 = strel('disk',5);          % disk, radius 15
```

1.1 Dilation

The dilation operation is illustrated in Fig.1.



(a) Square structuring element. (b) Line structuring element. (c) Disk structuring element.

Figure 1: Dilation with different structuring elements.

```
figure;
% dilation
dilateSquare = imdilate(B, se1);
dilateLine = imdilate(B, se2);
dilateDisk = imdilate(B, se3);
subplot(4,3,1); imshow(dilateSquare);title('dilation, square');
subplot(4,3,2); imshow(dilateLine);title('dilation, segment');
subplot(4,3,3); imshow(dilateDisk);title('dilation, disk');
```

1.2 Erosion

The dual operation of dilation is the erosion,illustrated in Fig.2.

(a) Square structuring element. (b) Line structuring element. (c) Disk structuring element.

Figure 2: Erosion with different structuring elements.

```
% erosion
erodeSquare = imerode(B, se1);
erodeLine = imerode(B, se2);
erodeDisk = imerode(B, se3);
```

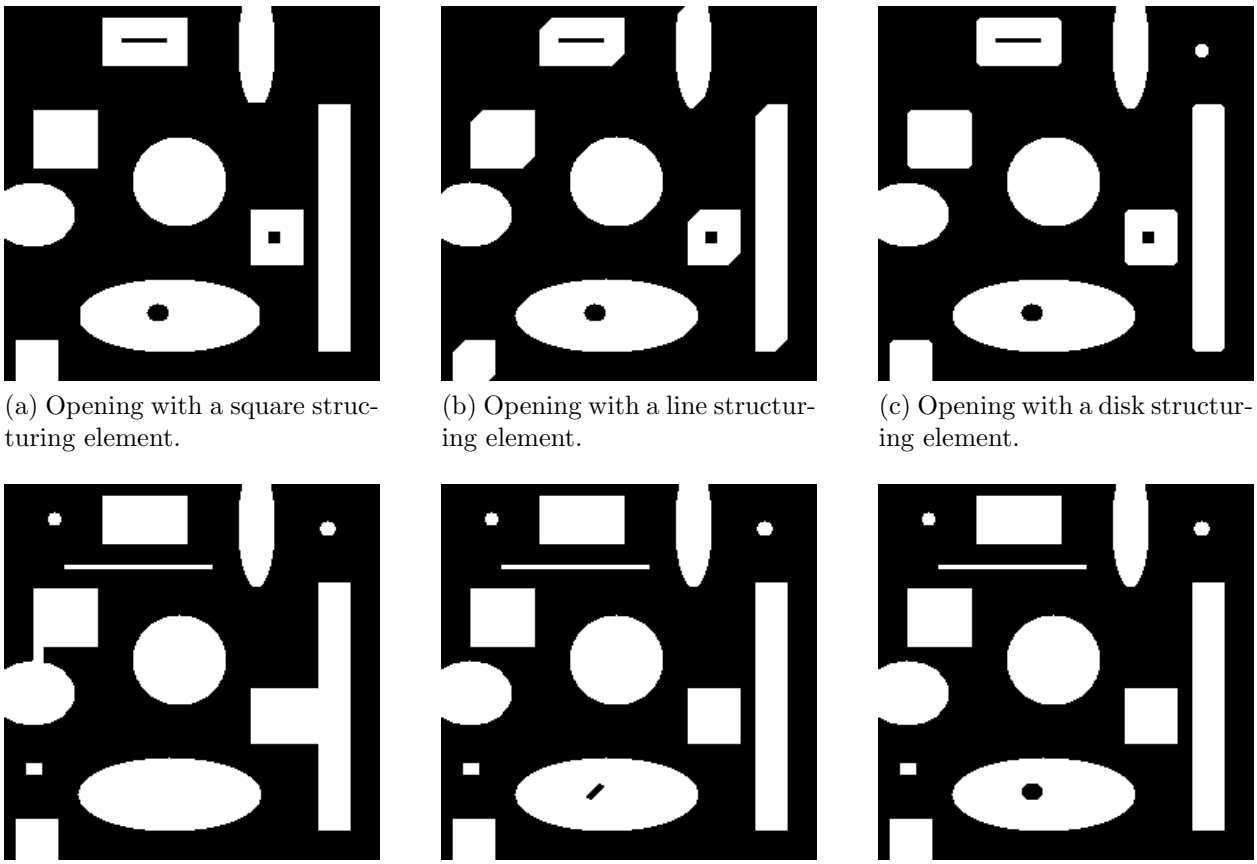
1.3 Opening and closing

Opening and closing are the combination of erosion and dilation in both orders, see illustration Fig.3.

```
% open
openSquare = imopen(B, se1);
openLine = imopen(B, se2);
openDisk = imopen(B, se3);
```

```
% close
closeSquare = imclose(B, se1);
closeLine = imclose(B, se2);
closeDisk = imclose(B, se3);
```

The following code is used to illustrate the impact of the size of the structuring element.



(a) Opening with a square structuring element.

(b) Opening with a line structuring element.

(c) Opening with a disk structuring element.

(d) Closing with a square structuring element.

(e) Closing with a line structuring element.

(f) Closing with a disk structuring element.

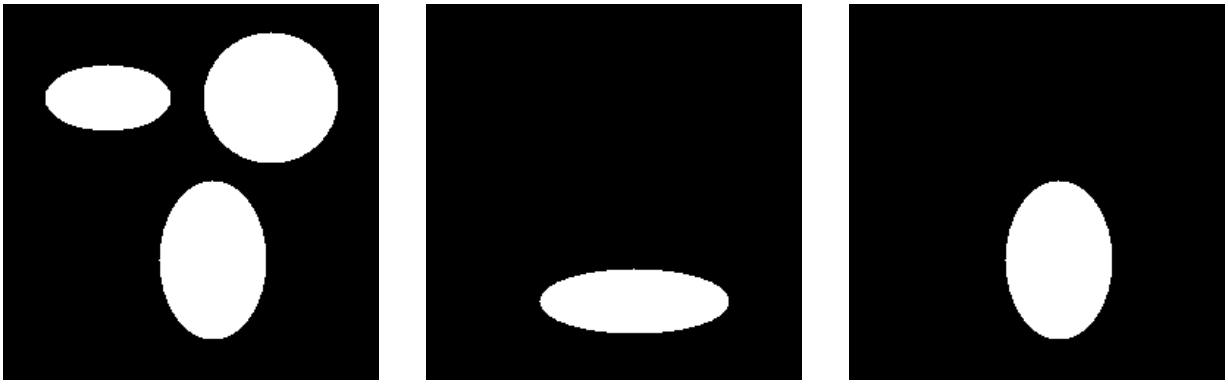
Figure 3: Opening and closing with different structuring elements.

```
%
% opening with varying size
figure;
for i=1:12
    sedisk(i)=strel('disk',i);
    openDisk = imopen(B, sedisk(i));
    subplot(4,3,i); imshow(openDisk);title(strcat('opening, ',int2str(i))
        ↪ );
end
```

1.4 Morphological reconstruction

The morphological reconstruction is employed in a lot of applications, like removing objects touching the borders of the image, of removing small objects. The intersection of two sets is coded as the minimum of binary arrays. The result is illustrated in Fig.4.

```
function B=reconstruct(A,M)
M=min(M,A);
r=bwarea(M);
s=0;
se=strel('disk',1);
while (r ~= s)
    s=r;
    M=min(A, imdilate(M, se));
    r=bwarea(M);
end
B=M;
```



(a) I_1 (b) M (c) $rec(I_1, M)$

Figure 4: Illustration of morphological reconstruction of I_1 by M.

1.4.1 Remove border objects

In order to remove the objects that may touch the border, a marker of the border is defined. In practice, the extreme pixels are marked, then the reconstruction of the objects by this marker is performed, then removed from the original image. Notice the definition of a border with value 255, as it can handle unsigned 8 bits images (thanks to the min function). See Fig.5. If $\mathcal{B}$ represents the border of the image (create an array of the same size as the image, with zeros everywhere and ones at the sides), then this operation is defined by:

$$\text{killBorders}(I) = I \setminus \rho_I(\mathcal{B})$$

```
function B=killBorders(A)
[m,n]=size(A);
M=zeros(m,n);
M(1,:)=255;
M(m,:)=255;
M(:,1)=255;
M(:,n)=255;
M=reconstruct(A,M);
B=A-M;
```

1.4.2 Remove small objects

The principal is first to make an erosion with a given structuring element se, in order to suppress the objects smaller than se. Then, a reconstruction is performed in order to get the original objects. See Fig.5.

It consists in an erosion followed by a reconstruction. The structuring element used in the erosion defines the objects considered as "small".

$$\text{killSmall}(I) = \rho_I(\varepsilon(I))$$

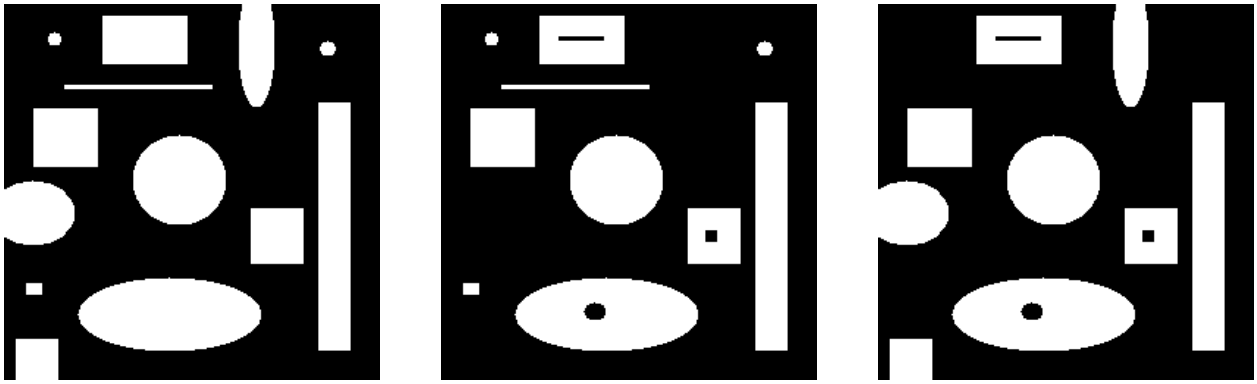
```
function B=killSmall(A,n)
se=strel('disk',n);
M=imerode(A,se);
B=reconstruct(A,M);
```

1.4.3 Close holes

In order to close holes, we work on the complementary set of A and try to isolate the background (that is supposed to touch the border of the image). See Fig.5. The operation is given by the following equation, with $\mathcal{B}$ the border of image I, and X^C denoting the complementary of set X:

$$\text{removeHoles(I)} = \{\rho_{I^C}(\mathcal{B})\}^C$$

```
function B=closeHoles(A)
Ac=imcomplement(A);
[m,n]=size(A);
M=zeros(m,n);
M(1,:)=255;
M(m,:)=255;
M(:,1)=255;
M(:,n)=255;
M=reconstruct(Ac,M);
B=imcomplement(M);
```



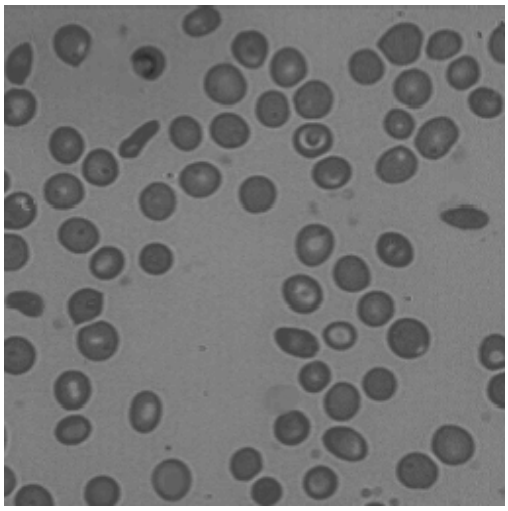
(a) Close holes in objects. (b) Kill border objects. (c) Kill small objects.

Figure 5: Illustration of the use of the morphological reconstruction.

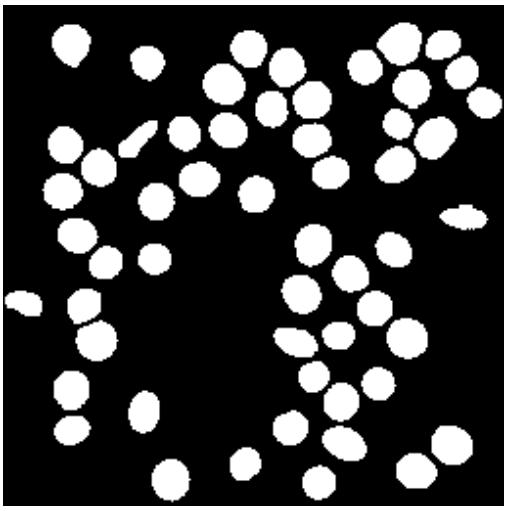
1.5 Application on cells image

The application on the cell image is quite trivial: after thresholding (binarizing) the original image (the threshold value is chosen experimentally), we can close the holes, remove the small objects and the ones touching the borders (see Fig.6).

```
A=imread('cells.bmp');
C=A<98;
B=closeHoles(C);
B=killBorders(B);
B=killSmall(B,5);
```



(a) Original image of cells.



(b) Segmented image.

Figure 6: Segmentation of the image of cells.