COC202 Computer Vision
Lab 3 – Morphological image processing – Solutions

1.

```
myim = logical([ ...
    0 0 0 0 0 0 0 0; ...
    0 0 0 1 1 1 0 0; ...
    0 0 1 1 1 1 0 0; ...
    0 1 1 1 1 0 0 0; ...
    0 0 0 1 0 0 0 0; ...
    0 1 1 1 0 0 0 0; ...
    0 0 0 0 0 0 0 0; ...
    0 0 0 0 0 0 0 0]);

mystrel = logical([0 1 0; 1 1 1; 0 1 0]);
```

2.

```
function eim = erode(im, sel)
% morphological erosion
% for simplicity we assume that the SEL is of size 3x3

[dimy dimx] = size(im);
eim = logical(zeros(dimy, dimx));
tim = padarray(im, [1 1], 0, 'both'); % pad image

for i=2:dimy+1
    for j=2:dimx+1
        block = tim(i-1:i+1,j-1:j+1); % current window
        block = block & sel; % extract window pixels where SEL=1
        if (block == sel) % SEL fits block
            eim(i-1,j-1) = 1;
        else
            eim(i-1,j-1) = 0;
        end
    end
end
```

```
eim1 = erode(myim,mystrel)
eim2 = imerode(myim, mystrel)
```

3.

```
function dim = dilate(im, sel)
% morphological dilation
% for simplicity we assume that the SEL is of size 3x3

[dimy dimx] = size(im);
dim = logical(zeros(dimy, dimx));
tim = padarray(im, [1 1], 0, 'both'); % pad image

for i=2:dimy+1
    for j=2:dimx+1
        block = tim(i-1:i+1,j-1:j+1); % current window
        block = block & sel; % extract window pixels where SEL=1
        if (sum(sum(block)) > 0) % SEL hits block
            dim(i-1,j-1) = 1;
        else
            dim(i-1,j-1) = 0;
        end
    end
end
```

```
dim1 = dilate(myim, mystrel)
dim2 = imdilate(myim, mystrel)
```

4.

```
function oim = mopen(im, sel)
% morphological opening

oim = dilate(erode(im, sel),sel);
```

```
oim1 = mopen(myim, mystrel)
oim2 = imopen(myim, mystrel)
```

5.

```
function cim = mclose(im, sel)
% morphological closing

cim = erode(dilate(im, sel),sel);
```

```
cim1 = mclose(myim, mystrel)
cim2 = imclose(myim, mystrel)
```

6.

```
imshow(mclose(mopen(imread('fingerprint.tif'),ones(3)),ones(3)));
```

7.

```
function bim = boundary(im)
% morphological boundary extraction

sel = ones(3,3);
bim = im - erode(im,sel);
```

```
im = imread('head.tif');
bim = boundary(im);
imshow(bim);
```

8.

```
imb = imread('headboundary.tif');
imshow(imb);
[x,y] = getpts; % obtain point inside boundary from user
imf = logical(zeros(size(imb)));
imf(y,x) = 1;
imshow(imf);
imc = imcomplement(imb);
strel = [0 1 0; 1 1 1; 0 1 0];
newimf = (dilate(imf, strel)) & imc; % intersection of dilation and
complement
imshow(newimf)
while (any(any(imf~=newimf))) % as long as region is growing
    imf = newimf;
    imshow(imf); % show current result
    newimf = (dilate(imf, strel)) & imc;
end
```