Tutorial 6 Image Segmentation

COMP 4421: Image Processing

March 14, 2016

Outline

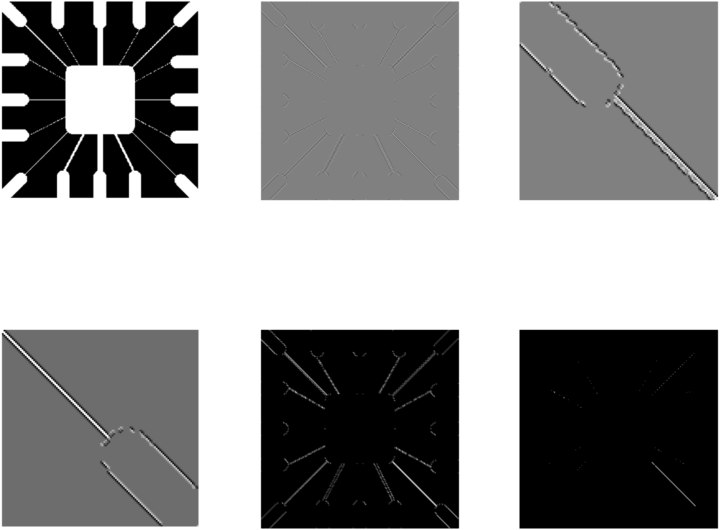
- Line Detection
- Hough Transform
- Thresholding
- Region Growing
- Watershed Segmentation
- Snakes: Active Contour Models

Line Detection

- Apply a mask and compute the image response
- Design of Mask determines what kind of line is detected

-1 -1 -1	-1 -1 2	-1 2 -1	2 -1 -1
2 2 2	-1 2 -1	-1 2 -1	-1 2 -1
-1 -1 -1	2 -1 -1	-1 2 -1	-1 -1 2
Horizontal	+45°	vertical	-45°

Line Detection



Line Detection

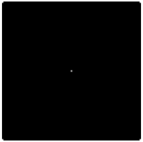
Matlab Code

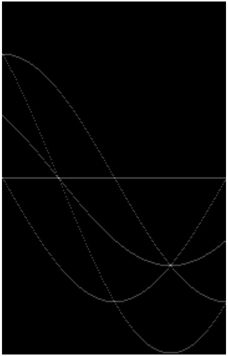
```
f = imread(’wirebond.tif’);
figure, subplot(2,3,1); imshow(f,[]);
w = [2 -1 -1; -1 2 -1; -1 -1 2];
g = imfilter(double(f),w);
subplot(2,3,2); imshow(g,[]);
gtop = g(1:120,1:120);
gtop = pixeldup(gtop, 4);
subplot(2,3,3); imshow(gtop, []);
gbot = g(end - 119:end, end-119:end);
gbot = pixeldup(gbot, 4);
subplot(2,3,4); imshow(gbot, []);
g = abs(g); subplot(2,3,5); imshow(g,[]);
T = max(g(:)); g = g >= T;
subplot(2,3,6); imshow(g,[]);
```

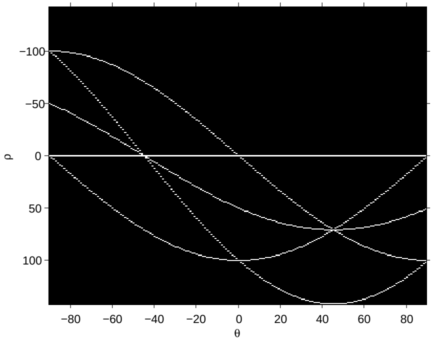
Hough Transform

- A method for line detection on binary images.
- Transform points in the x-y plane into the parameter space (ρ, θ).
- Detect local maximum on the parameter space (kind of voting), which corresponds to strong line signals.

Hough Transform







Hough Transform

Matlab Code

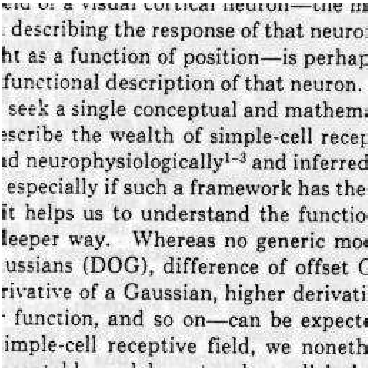
```
f = zeros(101,101);
f(1,1) = 1; f(101,1) = 1; f(1,101) = 1;
f(101,101) = 1; f(51,51) = 1;
figure,imshow(f,[]);
H = hough(f);
figure, imshow(H,[]);
[H, theta, rho] = hough(f);
figure, imshow(theta, rho, histeq(H), [], 'notruesize');
axis on; axis normal;
xlabel('\theta'),ylabel('\rho');
```

Thresholding

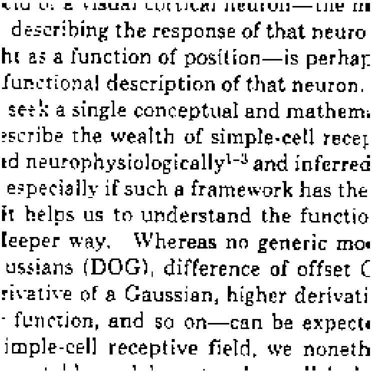
- Global Thresholding: threshold does not depend on image locations or local features/neighboring pixel properties.
- Local Thresholding: threshold depends on image locations or local features/neighboring pixel properties

Thresholding

- Global Thresholding



eld of a visual cortical neuron—the m
describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathem
escribe the wealth of simple-cell recep
d neurophysiologically$^{1-3}$ and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset G
rivative of a Gaussian, higher derivati
function, and so on—can be expect
imple-cell receptive field, we noneth



eld of a visual cortical neuron—the m
describing the response of that neuro
ht as a function of position—is perhap
functional description of that neuron.
seek a single conceptual and mathem
escribe the wealth of simple-cell recep
d neurophysiologically$^{1-3}$ and inferred
especially if such a framework has the
it helps us to understand the functio
leeper way. Whereas no generic mo
ussians (DOG), difference of offset G
rivative of a Gaussian, higher derivati
function, and so on—can be expect
imple-cell receptive field, we noneth

Thresholding

Matlab Code

```
f = imread(’text.jpg’);
figure,subplot(1,2,1),imshow(f);
f=double(f);
level = graythresh(f);    % help graythresh
O = bsxfun(@ge,f,level*255);    % help bsxfun
f= f.*O;
subplot(1,2,2),imshow(f);
```

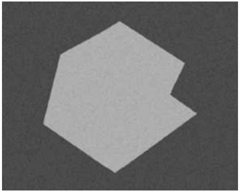
Thresholding

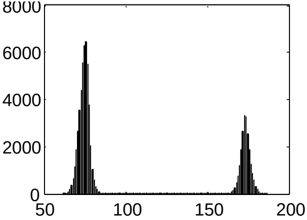
- Global Thresholding: threshold does not depend on image locations or local features/neighboring pixel properties.
- Local Thresholding: threshold depends on image locations or local features/neighboring pixel properties

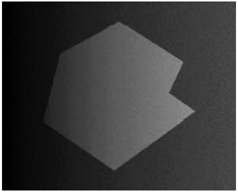
- The Global Thresholding may fail in many cases, eg. the effect of illumination

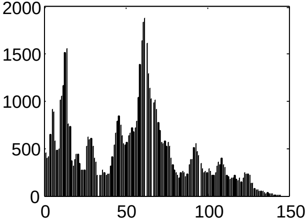
Thresholding

- Global Thresholding



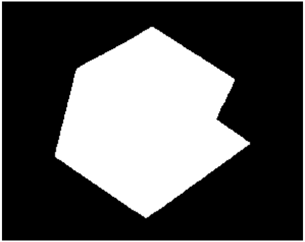


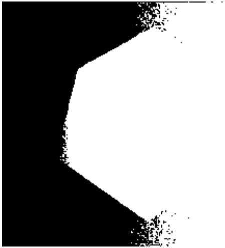




Thresholding

- Global Thresholding





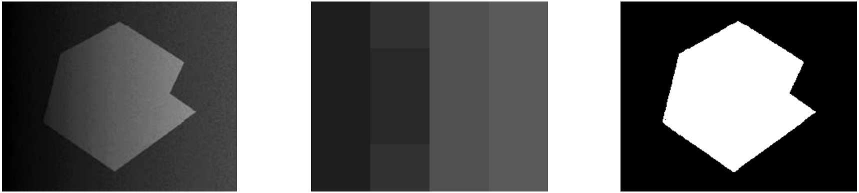
Thresholding

Matlab Code

```
% Read image and display histogram
f=imread(’image1.png’);
f=rgb2gray(f);
figure,subplot(2,2,1),imshow(f);
f=double(f); subplot(2,2,2),hist(f(:),500);
f1=imread(’image.png’);
f1=rgb2gray(f1); subplot(2,2,3),imshow(f1);
f1=double(f1); subplot(2,2,4),hist(f1(:),500);
% Apply global thresholding
O = bsxfun(@ge,f,120); f= f.*O;
figure,subplot(1,2,1),imshow(f);
O = bsxfun(@ge,f1,40); f1= f1.*O;
subplot(1,2,2),imshow(f1);
```

Thresholding

- Local Thresholding



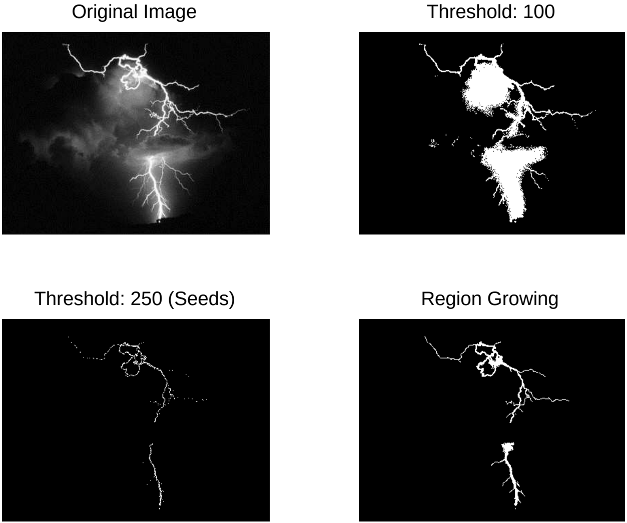
- Estimated Local Thresholds $t = \begin{bmatrix} 30 & 50 & 80 & 90 \\ 30 & 40 & 80 & 90 \\ 30 & 40 & 80 & 90 \\ 30 & 50 & 80 & 90 \end{bmatrix}$

Thresholding

Matlab Code

```
f1=imread(’image.png’); f1=rgb2gray(f1);
figure,subplot(1,3,1),imshow(f1);
f1=double(f1); T=f1;
Block=4; [x,y]=size(f1); sizeX=x/Block; sizeY=y/Block;
for i=1:Block
    for j=1:Block
        StartX=sizeX*(i-1)+1;
        StartY=sizeY*(j-1)+1;
        T(StartX:StartX+sizeX-1,StartY:StartY+sizeY-1)=t(i,j);
    end
end
subplot(1,3,2),imshow(uint8(T));
S=(f1>=T); subplot(1,3,3),imshow(S);
```

Region Growing



Original Image

Threshold: 100

Threshold: 250 (Seeds)

Region Growing

Region Growing

Matlab Code

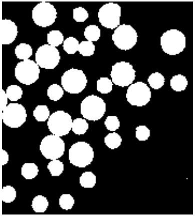
```
f=imread(’lightening.jpg’);
figure,subplot(2,2,1),imshow(f); title(’Original Image’);
f=rgb2gray(f);
f=double(f);
M = bsxfun(@ge,f,100);
S= f.*M;
subplot(2,2,2),imshow(S); title(’Threshold: 100’);
M = bsxfun(@ge,f,250);
S= f.*M;
subplot(2,2,3),imshow(S); title(’Threshold: 250 (Seeds)’);
[g, NR, SI, TI]=regiongrow(f, M, 50);
subplot(2,2,4),imshow(g); title(’Region Growing’);
```

Watershed Segmentation

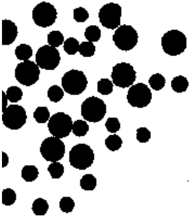
- Convert image to binary (im2bw)
- Distance Transform: the distance from every pixel to the nearest non-zero-valued pixel (bwdist)

Watershed Segmentation

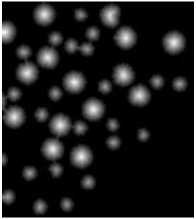
Binary Image



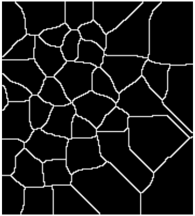
Complement Image



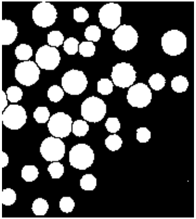
Distance Image



Ridge Pixels



Watershed Segmentation



Watershed Segmentation

Matlab Code

```
f=imread(’circle2.png’);f=rgb2gray(f);g=im2bw(f,graythresh(f));
figure,subplot(2,3,1),imshow(g); title(’Binary Image’);
gc = imcomplement(g); % or gc = ∼g;
subplot(2,3,2),imshow(gc);title(’Complement Image’);
D = bwdist(gc); % distant tranform
subplot(2,3,3),imshow(D,[]); title(’Distance Image’);
% L is a label matrix, whose positive integers corresponds to catchment
L = watershed(-D);
w = L ==0; % find the ridge pixels
subplot(2,3,4),imshow(w); title(’Ridge Pixels’);
% superimposed image of ridge lines and original binary image
g2 = g &  w; % note the oversegmentation
subplot(2,3,5),imshow(g2,[]); title(’Watershed Segmentation’);
```

Snakes: Active Contour Models

- Energy Function

$$E(\vec{V}) = E_{int}(\vec{V}) + E_{ext}(\vec{V})$$

- Internal Energy

$$E_{int}(\vec{V}) = \frac{1}{2}\int_0^1 \alpha(s)\left|\frac{d\vec{V}}{ds}\right|^2 ds + \frac{1}{2}\int_0^1 \beta(s)\left|\frac{d^2\vec{V}}{ds^2}\right|^2 ds$$

a. the first term controls the ‘tension‘ of the contour.
b. the second term controls the ‘rigidity‘ of the contour.

Snakes: Active Contour Models

- External Energy

$$E_{ext}(\vec{V}) = \int_0^1 E_{image}\left(\vec{V}(s)\right) ds$$

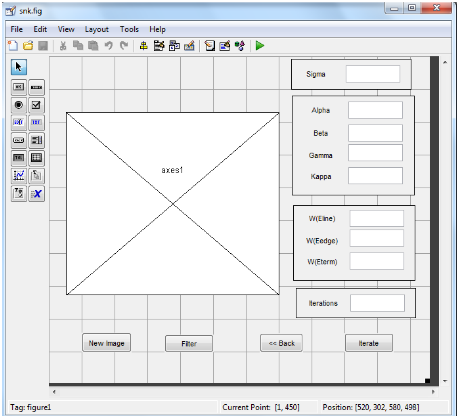
a. E_{image} represents the scalar potential (gradient) function defined on the image plane, e.g.

$$E_{image}\left(\vec{V}(s)\right) = -c\left|\triangledown\left[G_\sigma * I\left(\vec{V}(s)\right)\right]\right|$$

b. $c > 0$ is constant, $G_\sigma * I$ represents an image I convolved with a Gaussian smoothing filter with SD σ.

Snakes: Active Contour Models

- Matlab code for Snakes: Active Contour Models



Tag: figure1

Current Point: [1, 450]

Position: [520, 302, 580, 498]

Reference: http://www.mathworks.com/matlabcentral/fileexchange/28109-snakes-active-contour-models

Thank you!