

Elaborazione di Segnali Multimediali  
a.a. 2017/2018

## Segmentazione Soluzioni

### 1 Tecniche edge-based

1. *Point e line detection*. Per quanto riguarda la rilevazione di un punto isolato:

```
clear all; close all; clc;
h = double(imread('turbina.jpg'));
figure(1); imshow(x, []);
h = [-1 -1 -1; -1 8 -1; -1 -1 -1];
y = imfilter(x,h);
figure(2); imshow(y, []);
soglia = 0.9*max(abs(y(:)));
z = abs(Y) > soglia;
figure(3); imshow(~z, []);
```

Per l'individuazione delle linee oblique (di spessore pari a 1 pixel):

```
clear all; close all; clc;
x = double(imread('quadrato.jpg'));
figure(1); imshow(x, []);
h = [2 -1 -1; -1 2 -1; -1 -1 2];
y = imfilter(x,h);
figure(2); imshow(y, []);
y = abs(y);
figure(3); imshow(-y, []);
soglia = max(y(:));
z = y >= soglia;
figure(4); imshow(~z, []);
```

Infine per ottenere una mappa con linee in qualunque direzione:

```
clear all; close all; clc;
x = double(imread('casa.tif'));
figure(1); imshow(x,[]);
h1 = [2 -1 -1; -1 2 -1; -1 -1 2];
y(:, :, 1) = imfilter(x, h1);
h2 = [-1 -1 2; -1 2 -1; 2 -1 -1];
y(:, :, 2) = imfilter(x, h2);
h3 = [-1 -1 -1; 2 2 2; -1 -1 -1];
y(:, :, 3) = imfilter(x, h3);
h4 = [-1 2 -1; -1 2 -1; -1 2 -1];
y(:, :, 4) = imfilter(x, h4);

m = max(y, [], 3);
m = abs(m);
figure(3); imshow(-m, []); % mappa con linee in tutte le direzioni

soglia = 0.9*max(m(:));
z = m >= soglia;
figure(4); imshow(~z, []); % risposte più forti per linee spesse 1 pixel
```

2. *Edge detection.* Edge detection con filtro di Sobel per un'immagine memorizzata nelle variabile x:

```
h1 = [-1 -2 -1; 0 0 0; 1 2 1];
h2 = h1';

d1 = imfilter(x, h1, 'replicate');
d2 = imfilter(x, h2, 'replicate');
d = sqrt(d1.^2 + d2.^2);
figure; imshow(d, []);
title('Gradiente con filtro di Sobel');

soglia = 1.5*mean2(d);
m = d > soglia;
figure; imshow(~m, []);
title('Mappa binaria dopo il thresholding');

%m = bwmorph(m, 'thin', 5);
%figure; imshow(~m, []);
%title('Mappa binaria dopo il thinning');
```

### 3. Confronto strategie.

```
x = double(imread('van.tif'));
figure; imshow(x, []);
[Nr, Nc] = size(x);
title('Immagine da cui estrarre i contorni');

% calcolo del modulo e della fase del gradiente
h1 = [-1 -2 -1; 0 0 0; 1 2 1];
h2 = [-1 0 1; -2 0 2; -1 0 1];

z1 = imfilter(x,h1,'replicate');
z2 = imfilter(x,h2,'replicate');
z = sqrt(z1.^2 + z2.^2);

soglia = 0.3*max(z(:));
M1 = z > soglia;
%Mb = bwmorph(M1, 'thin',5);
figure; imshow(~M1, []); title('mappa dei contorni con Sobel');

% Laplaciano di una gaussiana
M2 = edge(x,'log',0.7,3);
figure; imshow(~M2, []); title('mappa dei contorni con LoG');
```

### 4. Segmentazione oggetti chiari.

```
x = double(imread('dowels.tif'));
figure; imshow(x, []); title('Immagine originale');
M1 = edge(x,'canny',0.4,0.8);
figure; imshow(~M1, []); title('mappa dei contorni con Canny');
```

## 2 Tecniche class-based

### 1. Thresholding locale.

```
x = double(imread('yeast.tif'));
figure; imshow(x, []); title('immagine originale');

% k-means
[M N] = size(x);
xv = x(:);
idx = kmeans(xv,3);
mappa = reshape(idx,M,N);
figure; imshow(mappa, []); title('k-means');
```

```
% thresholding adattativo
DEV = colfilt(x,[3 3],'sliding',@std);
mask = x>30*DEV & x>1.5*mean2(x);
mask2 = DEV > 4;
mask = mask + mask2;
figure; imshow(mask,[]); title('thresholding adattativo');
```

## 2. Metodo di Otsu.

```
x = double(imread('molecole.tif'));
figure; imshow(x,[]); title('immagine originale');
[M N] = size(x);

% k-means
temp = x(:);
idx = kmeans(temp,2);
y = reshape(idx,M,N);
figure; imshow(y,[]); title('k-means');

% metodo di Otsu
f = hist(x(:), [0:255])/length(x(:)); % istogramma normalizzato
p = cumsum(f); % calcolo somma cumulativa
m(1) = 0; % calcolo medie cumulative
for i = 1:255
    m(i+1) = m(i)+i*f(i+1);
end;
mG = mean2(x);
sG = (std(x(:),1)).^2;
% la formula di seguito va calcolata solo per
% i valori di di p diversi da 0 e 1
ip = find(p==0);
jp = find(p==max(p));
pmod = p(max(ip)+1:min(jp)-1);
mmod = m(max(ip)+1:min(jp)-1);
sigma2B = ((mG*pmod-mmod).^2)./(pmod.*(1-pmod));
kstar = find(sigma2B == max(sigma2B)) + max(ip) -1;
mappa = x>kstar;
mappa = mappa+1;
figure; imshow(mappa,[]); title('otsu');
```