
Laboratorio di Immagini

Esercitazione 2:

Morfologia

Mauro Zucchelli
17/03/2016

Operatori morfologici

- Erosione
 - Dilatazione
 - Apertura
 - Chiusura
-

Erosione

```
function img_erode = my_erode(img,se)

[x,y] = size(img);
img_erode = zeros(x,y);
[sx,sy] = size(se);

lx = floor(sx/2);
ly = floor(sy/2);

for i=(lx+1):(x-lx)
    for j=(ly+1):(y-ly)
        roi = img(i-lx:i+lx, j-ly:j+ly);
        bool_roi = and(roi,se);
        if isequal(se,bool_roi)
            img_erode(i,j) = 1;
        end
    end
end
```

Erosione

- Caricare l'immagine pixels_wb.tiff
 - Fare una soglia a 100 per binarizzarla
 - `img = img > 100;`
 - Creare un elemento strutturale (es: crocetta)
 - `s = [0,1,0;1,1,1;0,1,0]`
 - Applicare l'operazione di erosione
 - `img_erode = my_erode(img,s);`
 - Visualizzare il prima il dopo e le differenze
-

Erosione

- Caricare l'immagine pixels_wb.tiff
 - Fare una soglia a 100 per binarizzarla
 - `img = img > 100;`
 - Creare un elemento strutturale (es: crocetta)
 - `s = [0,1,0;1,1,1;0,1,0]`
 - Applicare l'operazione di erosione
 - `img_erode = my_erode(img,s);`
 - Visualizzare il prima il dopo e le differenze
-

Erosione

```
img = imread('pixels_wb.tiff');

figure;
imshow(img)
img = img > 100;

s = [0,1,0;1,1,1;0,1,0]
|
img_erode = my_erode(img,s);

figure('Name','Erosion');
subplot(1,3,1)
imshow(img);
title('original')
subplot(1,3,2)
imshow(img_erode);
title('erosion')
subplot(1,3,3)
imshow(img-img_erode);
title('difference')
```

Dilatazione

- Provate a modificare il codice di “my_erode” per creare l’operazione di dilatazione
 - Create una nuova *function* e salvatela come my_dilate.m
-

MATLAB: erosione e dilatazione

- MATLAB ovviamente ha già tutto implementato
 - `strel`
 - `imerode`
 - `Imdilate`
 - Confrontate il risultato di `my_erode` con quello di `imerode` per essere sicuri di avere fatto giusto
 - `s= strel('diamond',1)`
 - `isequal(img_erode, img_erode2)`
-

MATLAB: erosione e dilatazione

```
%% PROVA DILATAZIONE
```

```
img_dilate = imdilate(img,s);
```

```
figure('Name','Dilatation');
```

```
subplot(1,3,1)
```

```
imshow(img);
```

```
title('original')
```

```
subplot(1,3,2)
```

```
imshow(img_dilate);
```

```
title('dilate')
```

```
subplot(1,3,3)
```

```
imshow(img_dilate-img);
```

```
title('difference')
```

MATLAB: apertura

- L'apertura è un'erosione seguita da una dilatazione...
 - Provate!
-

MATLAB: apertura

```
%% Prova apertura
img_erode= imerode(img,s);
img_open = imdilate(img_erode,s);

figure('Name','Apertura');
subplot(1,3,1)
imshow(img);
title('original')
subplot(1,3,2)
imshow(img_open);
title('open')
subplot(1,3,3)
imshow(img-img_open);
title('difference')
```

MATLAB: chiusura

- La chiusura è una dilatazione seguita da una erosione...
 - Provate!
-

MATLAB: chiusura

```
%% Prova chiusura
img_dilate = imdilate(img,s);
img_close= imclose(img_dilate,s);

figure('Name','Chiusura');
subplot(1,3,1)
imshow(img);
title('original')
subplot(1,3,2)
imshow(img_close);
title('close')
subplot(1,3,3)
imshow(img_close-img);
title('difference')
```

MATLAB: apertura e chiusura

- Ovviamente MATLAB ha già implementato tutto
 - `imopen`
 - `imclose`

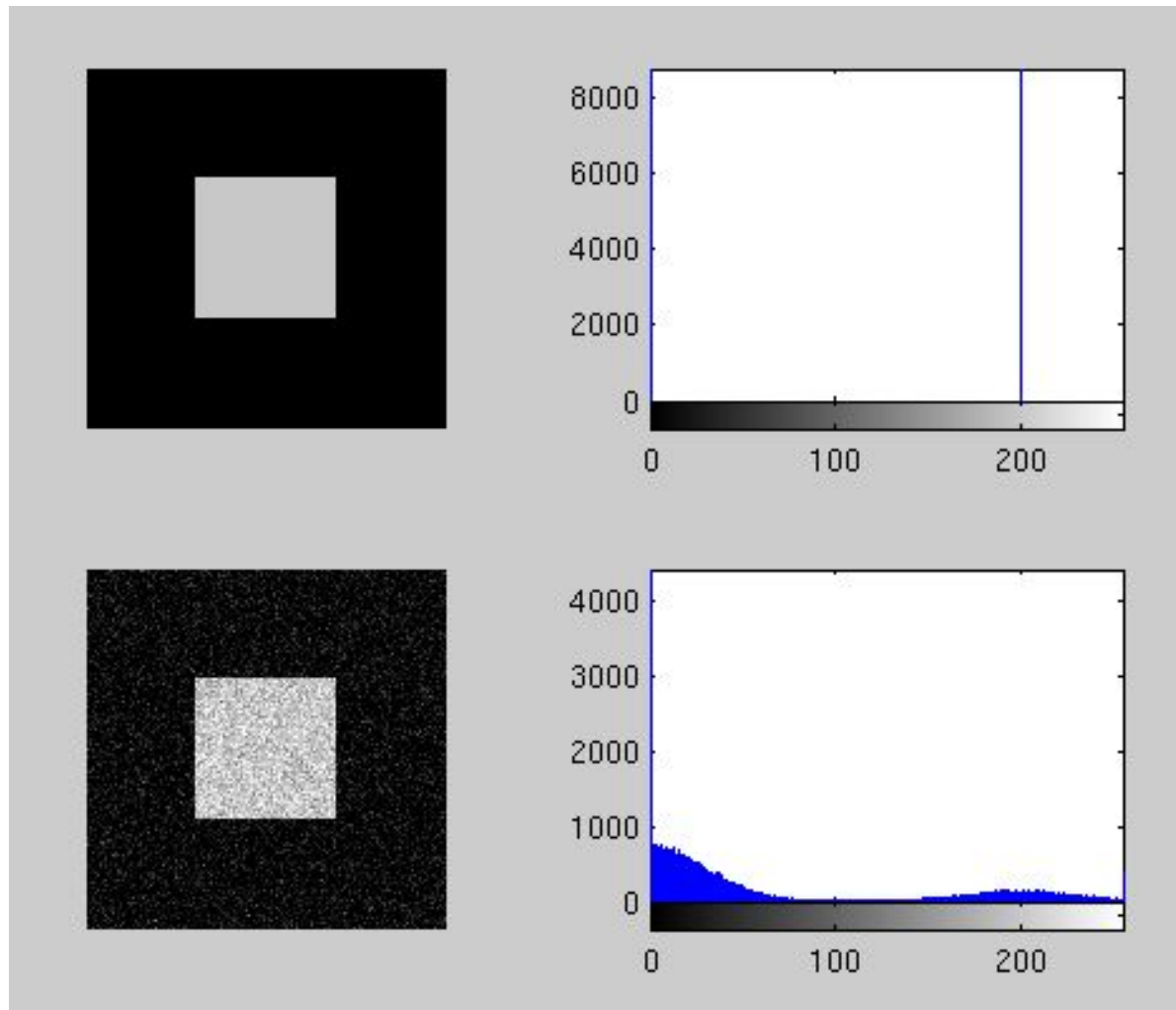
Applicazione NOISE REDUCTION

- Create un immagine 256x256 con dentro un quadrato bianco
 - Aggiungete rumore
 - Provate a sogliare l'immagine (aiutandovi con l'istogramma) al fine di ritrovare il quadrato
-

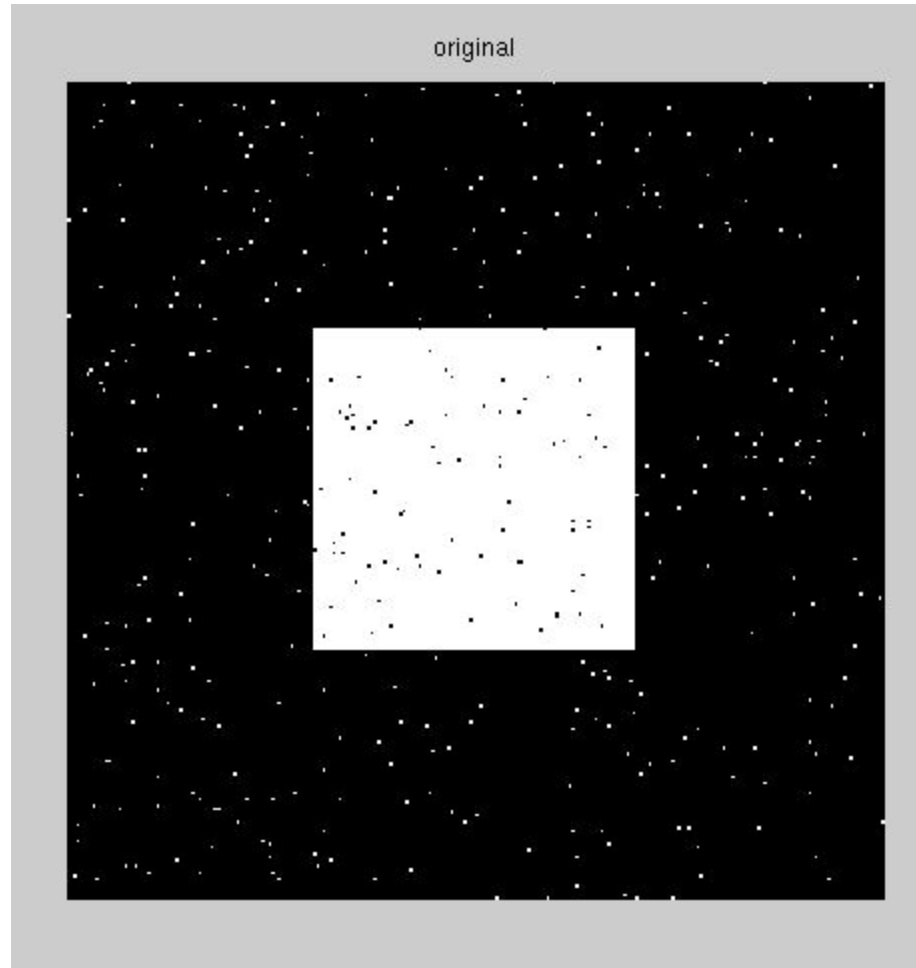
Applicazione NOISE REDUCTION

```
%% Application Noise reduction |
img = zeros(256,256);
img(128-50:128+50,128-50:128+50) =200;
img = uint8(img);
figure;
subplot(2,2,1)
imshow(img)
subplot(2,2,2)
imhist(img)
img_noise = double(img);
img_noise = img_noise + randn(256,256)*30;
img_noise(img_noise<0)=0;
img_noise = uint8(img_noise);
subplot(2,2,3)
imshow(img_noise)
subplot(2,2,4)
imhist(img_noise)
img_bw = img_noise>100;
figure;
subplot(1,3,1)
imshow(img_bw)
title('original')
```


Applicazione NOISE REDUCTION



Applicazione NOISE REDUCTION



Applicazione NOISE REDUCTION

- Trovate la sequenza di operatori morfologici che vi permetterà di “ripulire” il quadrato

Applicazione NOISE REDUCTION

```
img_bw = img_noise>100;
figure;
subplot(1,3,1)
imshow(img_bw)
title('original')

s = strel('square',3)
img_close = imclose(img_bw,s);
subplot(1,3,2)
imshow(img_close)
title('close')

img_open = imopen(img_close,s);
subplot(1,3,3)
imshow(img_open)
title('open')
```

Applicazione: immagini grayscale

- Caricate l'immagine "lena.jpg" (o la vostra preferita)
 - Convertitela in scala di grigi
 - Testate gli operatori morfologici
-
- Provate a cambiare l'elemento strutturale... Cosa cambia?
-

Applicazione: immagini grayscale

```
%% application to grayscale
img = rgb2gray(imread('lena.jpg'));
s = strel('square',11)
img_erode = imerode(img,s);
img_dilate = imdilate(img,s);
img_close= imclose(img,s);
img_open= imopen(img,s);
figure;
subplot(2,2,1)
imshow(img_erode)
title('erosion')
subplot(2,2,2)
imshow(img_dilate)
title('dilatation')
subplot(2,2,3)
imshow(img_open)
title('opening')
subplot(2,2,4)
imshow(img_close)
title('closing')
```

Applicazione: immagini grayscale

erosion



dilatation



opening



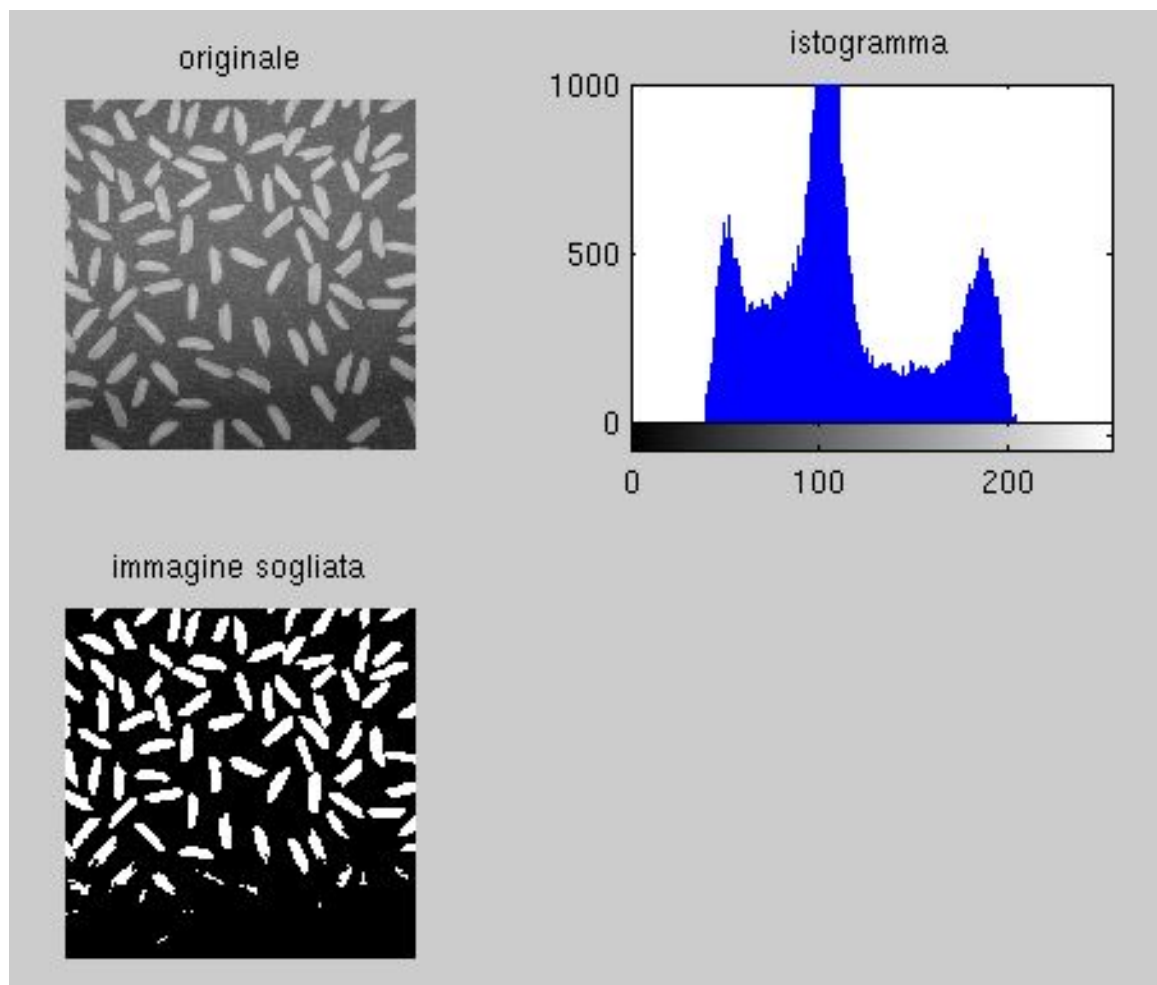
closing



Applicazione: rimozione background

- Caricate l'immagine "rice.png"
 - Cercate di trovare una soglia tale da evidenziare solo i chicchi di riso
 - Ovviamente aiutatevi con l'istogramma!
-

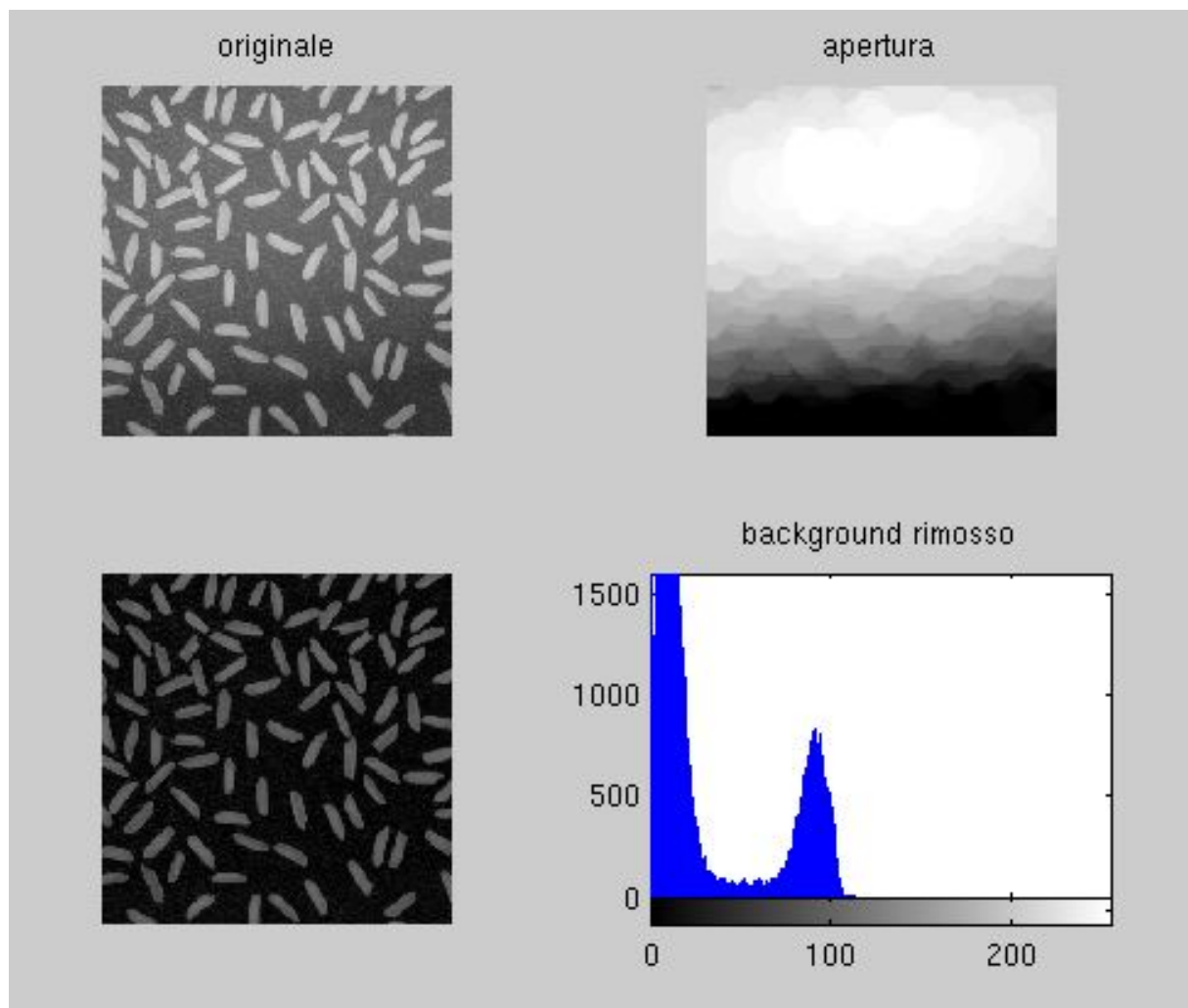
Applicazione: rimozione background



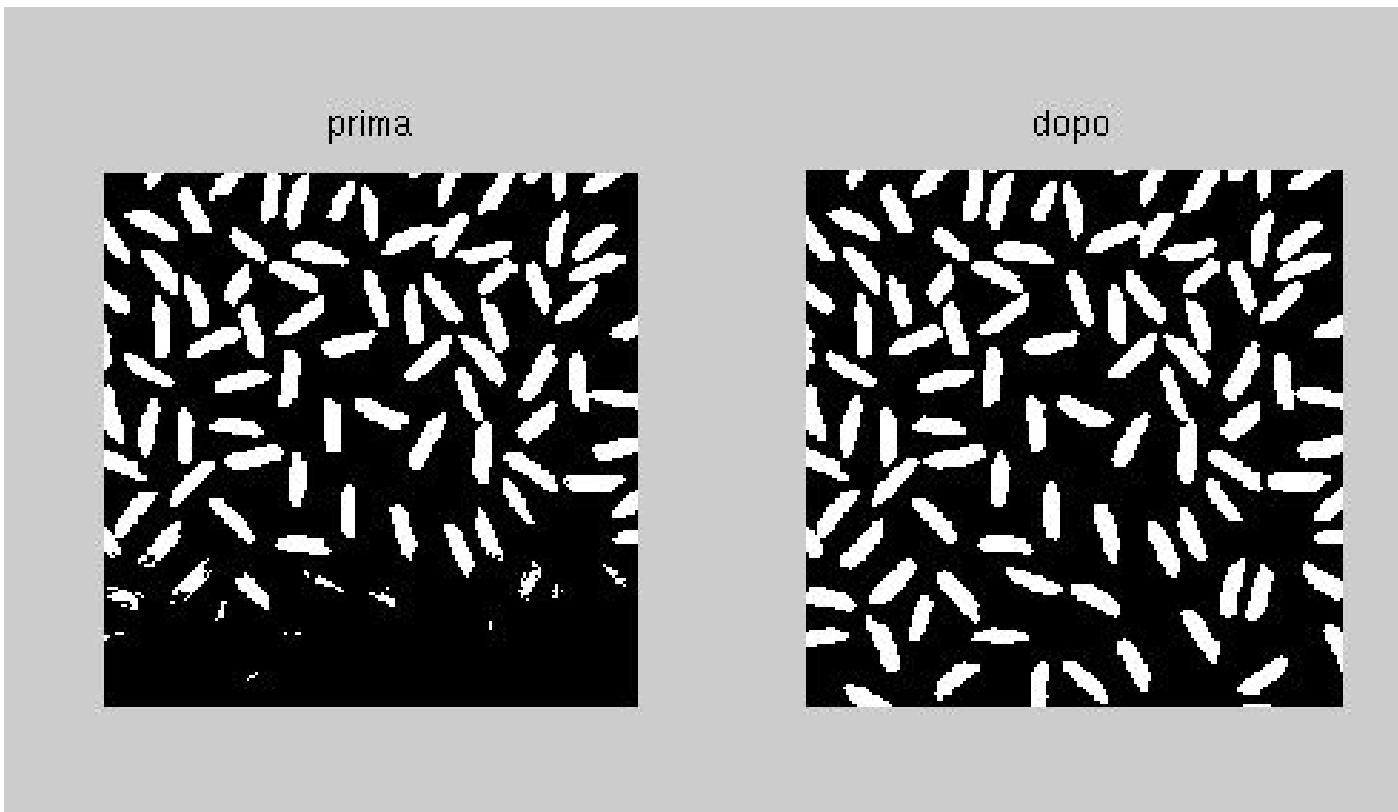
Applicazione: rimozione background

- A causa del diverso contrasto, non riuscite a prendere i chicchi in basso (in ombra)
 - Soluzione:
 - Definire un operatore morfologico più grande dei chicchi
 - “Aprire” l’immagine
 - Sottrarre l’immagine “aperta” a l’immagine originale
 - Provare a sogliare ora!
-

Applicazione: rimozione background



Applicazione: rimozione background



Applicazione: rimozione background

```
img = imread('rice.png');
figure
subplot(2,2,1)
imshow(img)
subplot(2,2,2)
imhist(img)

img_bw = img > 150;
subplot(2,2,3)
imshow(img_bw)
s = strel('disk',15);
background = imopen(img,s);
figure
subplot(2,2,1)
imshow(img)
subplot(2,2,2)
imshow(background,[])
subplot(2,2,3)
img_rem = img-background;
imshow(img_rem)
subplot(2,2,4)
imhist(img_rem)
```

FINE!
