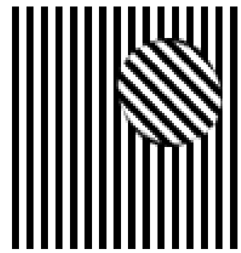
Esame del corso di
Tecniche Avanzate per il Trattamento delle Immagini

Data: 8 Settembre 2008

Es.1. Sia dato il seguente insieme di 8 elementi: **x**=[2,5,5,7,7,8,10,15], nell'ipotesi che i loro valori siano dovuti ad una distribuzione gaussiana, si **dimostri** (illustrando i vari passaggi) come vengono definiti *media* e *varianza* utilizzando gli stimatori a Massima Verosimiglianza e se ne calcoli il valore.

Es.2: Utilizzando il **criterio di segmentazione di tessiture tramite statistica Gaussiana** indicare come è possibile separare le due regioni in figura.
Indicare inoltre punti di forza e possibili errori di tale metodo.
Si assuma che lo spessore delle linee è pari a 1 pixel.

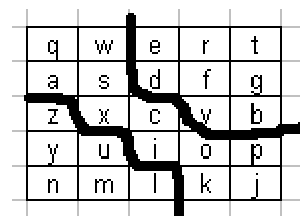


Es.3: Si consideri la matrice A, si applichi un filtro mediano 3x3 con riempimento degli elementi extra-bordo a 0. Alla matrice così ottenuta si applichi un filtro laplaciano; infine si disegni sulla matrice risultante dove, approssimativamente, si troveranno i contorni col metodo del laplaciano.

A=

5	4	3	2	0
1	2	4	5	2
3	0	5	5	4
2	4	4	2	0
4	2	1	4	1

Es. di matrice risultante:



a partire da quelli dell'immagine originale.

Es.4: Si consideri noto un algoritmo per l'estrazione di feature da adoperarsi in un sistema di riconoscimento supervisionato capace di discriminare immagini raffiguranti 2 possibili oggetti (A e B). Il sistema di riconoscimento accetta in input una sola feature per ogni immagine, mentre l'algoritmo estrae 2 feature da ogni immagine (e.s. l'altezza e la luminanza media dell'oggetto). Avendo a disposizione 2 insiemi di training (t_1 e t_2 di dimensioni: size(t_1,1) = # immagini con oggetto A; size(t_2,1) = # immagini con oggetto B; size(t_1,2) = size(t_2,2) = 2) si elenchino i passi necessari per rendere i dati idonei ad essere usati come input per il riconoscitore, cercando di sfruttare al massimo le informazioni disponibili. Si implementi ogni passo mediante codice Matlab.

Elenco di alcune funzioni Matlab:

```
mean
imerode
end
im2bw
figure
eig
repmat
rgb2gray
imread
title
sqrt
getimage
strel
diag
imdilate
imshow
rgb2ind
size
```

Es.5: Data un'immagine truecolor memorizzata in un file immagine.bmp ed avente i dati rappresentati ad 8bit per ogni piano di colore si implementino i seguenti punti mediante codice Matlab:
• Leggere, caricare nel workspace e visualizzare l'immagine.
• Convertire l'immagine truecolor riducendo il numero di colori rappresentabili ad 8 mediante quantizzazione a minima varianza e visualizzare il risultato.
• Descrivere una procedura tale che, se applicata durante la conversione di cui al passo precedente, sia in grado di produrre all'occhio umano l'illusione di una maggiore profondità di colore. Si indichino a parole i passi necessari da compiere e si concluda la spiegazione invocando in codice Matlab i comandi necessari per ottenerla e visualizzarne il risultato.
• Convertire l'immagine di partenza truecolor ad una rappresentazione a 256 toni di grigio e, successivamente, da una rappresentazione a 256 toni di grigio ad una a soli due valori (binaria).
• Applicare all'immagine binaria ottenuta un'operazione che elimini dettagli di dimensione irrilevante, immaginati come quadrati di dimensione 10x10, mantenendo inalterate le altre figure rappresentate, immaginate come quadrati di dimensione >= 30x30.

Soluzioni

1. Si vedano i lucidi delle lezioni. $\overline{x} = 7.375$ e $\sigma^2 = 15.125$
2. Utilizzando lo stimatore statistico Gaussiano:

$$T_{f_1}(p,q) = \sum_{(m,n)\in\Omega_1} [f_1(m,n) - f_1(m+p, n+q)]^2,$$

con (p,q) = (1,0) e/o con (p,q) = (1,1) è possibile individuare le due diverse regioni in quanto T(1,0) sarà sempre nullo nella regione con righe verticali e non nullo in quella con righe oblique e viceversa.

3. Il risultato del filtro mediano è pari a:

0	2	2	2	0
1	3	4	4	2
1	3	4	4	2
2	3	4	4	1
0	2	2	1	0

La successiva applicazione del filtraggio laplaciano dà:

6	-6	-1	-4	8
1	-7	-8	-12	-4
4	-2	-3	-7	-1
-7	-6	-9	-14	3
7	-5	-2	3	6

La successiva estrazione darà:

6	-6	-1	-4	8
1	-7	-8	-12	-4
4	-2	-3	-7	-1
-7	-6	-9	-14	3
7	-5	-2	3	6

4.

- Trasformazione applicata per ridurre la dimensionalità delle feature in modo supervisionato (conoscenza a priori delle classi di appartenenza dei dati)

$$y = A^T x$$

- Dove:
  - A (NxM) matrice di trasformazione le cui colonne sono gli autovettori corrisp. agli M autovalori più alti della matrice:

$$C = inv(S_w) * S_b$$

- S_w matrice della varianza complessiva intra-cluster

$$S_w = \sum_j p_j Cov_j$$

$p_j = \textbf{\textit{prob. a priori classe}}\ j$

$Cov_j = \textbf{\textit{covarianza dati classe}}\ j$

- S_b matrice della varianza complessiva inter-cluster

$$S_b = \sum_j (\mu_j - \mu) * (\mu_j - \mu)^T$$

$\mu_j = \textbf{\textit{centroide dati classe}}\ j$

$\mu = \textbf{\textit{centroide complessivo dati}}$

- Si calcolano i rispettivi centroidi e le matrici Sw e Sb

```
mu_1 = mean(t_1, 1);
mu_2 = mean(t_2, 1);
mu_3 = size(t_1,1)/(size(t_1,1)+size(t_2,2))*mu_1 +
size(t_2,1)/(size(t_1,1)+size(t_2,2))*mu_2;

Cov_1 = (t_1-repmat(mu_1, size(t_1, 1), 1))' * (t_1-repmat(mu_1, size(t_1, 1), 1));

Cov_2 = (t_2-repmat(mu_2, size(t_2, 1), 1))' * (t_2-repmat(mu_2, size(t_2, 1), 1));

S_w = size(t_1,1)/(size(t_1,1)+size(t_2,2))*Cov_1 +
size(t_2,1)/(size(t_1,1)+size(t_2,2))*Cov_2;
S_b = (mu_1 - mu_3)'*(mu_1 - mu_3) + (mu_2 - mu_3)'*(mu_2 - mu_3);
```

- Si calcola C, i suoi autovettori ed autovalori
  ```
  C = S_b/S_w;
  [V D] = eig(C);
  ```
- Si calcola la matrice di trasformazione composta dall'autovettore associato all'autovalore più alto
  ```
  P = diag(1./sqrt(D(end,end)))*V(:,end)';
  ```
- Si procede alla trasformazione dei dati delle singole classi
  ```
  t_r_1 = P*(t_1 - repmat(mu_3', 1, size(t_1,2)));
  t_r_2 = P*(t_2 - repmat(mu_3', 1, size(t_2,2)));
  ```

5.

- img_rgb = imread('immagine.bmp');
  figure; imshow(img_rgb);
  (oppure: figure
  imshow('immagine.bmp');
  img_rgb = getimage;)

- [img_ind map_ind] = rgb2ind(img_rgb, 8, 'nodither');
  figure; imshow(img_ind, map_ind); title('Conversione Truecolor --> Indexed (mappa ad 8 valori)');

-
  - Algoritmo di **Floyd-Steinberg** basato sulla dispersione dell'errore:
    - Visita il pixel (i,j)
      - Dato suo valore originale, assegna a (i,j) il valore più vicino (a minima distanza) tra quelli a disposizione (mappa di colori o valori di intensità)
      - Calcola la differenza tra il valore originale e quello assegnato
      - Distribuisci questo errore tra i pixel vicini secondo le proporzioni in tabella sommandolo ai loro valori

0	Pixel attuale	7/16
3/16	5/16	1/16

    - Ripeti la procedura visitando il pixel successivo (visita sx→dx (anche alternato), alto→basso)

  [img_ind map_ind] = rgb2ind(img_rgb, 8);
  figure; imshow(img_ind, map_ind); title('Conversione Truecolor --> Indexed (mappa ad 8 valori)');

- ```
  img_gray = rgb2gray(img_rgb);
  figure; imshow(img_gray); title('Conversione Truecolor --> Grayscale');
  % grayscale --> binary
  img_bin = im2bw(img_gray, 0.4);
  figure; imshow(img_bin); title('Conversione Grayscale --> Binary (soglia a 0.4)');
  ```

- ```
  B = strel('square', 30);
  A_er_B = imerode(img_bin,B);
  figure
  imshow(A_er_B); title('Immagine erosa con structuring element quadrato 30x30');
  A_er_dil_B = imdilate(A_er_B,B);
  figure
  imshow(A_er_dil_B); title('Immagine erosa e dilatata con structuring quadrato 30x30');
  ```