

Esame del corso di Tecniche Avanzate per il Trattamento delle Immagini



Data: 16 Maggio 2009

Es.1.[6 Pt] Data un'immagine 100x100, ad essa si applichi la seguente matrice di trasformazione (dopo aver reso omogenee le coordinate). Indicare come sarà l'immagine risultante e come

poter gestire eventuali interpolazioni necessarie.

$$\begin{bmatrix} 10\frac{\sqrt{2}}{2} & -10\frac{\sqrt{2}}{2} & -5 \\ 10\frac{\sqrt{2}}{2} & 10\frac{\sqrt{2}}{2} & 5 \end{bmatrix}$$

Es.2. [2pt] Indicare,utilizzando la notazione matematica adeguata, cosa si intende per "filtrare"

un'immagine a toni di grigio con la seguente matrice:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

[1 pt]A quale operazione matematica può essere assimilata (si distingua il comportamento in verticale da quello in orizzontale)?

[1 pt]Quale sarà l'effetto sull'immagine finale?

[2 pt]Indicare qualitativamente l'effetto che si produce sulla trasformata di Fourier 2D Bidimensionale dell'immagine iniziale.

[2 pt] Indicare un possibile codice per applicare tale filtro ad un'immagine in MATLAB

Es.3. [5 pt] Data l'immagine di 4 pixel costituita dai seguenti valori di grigio $\begin{bmatrix} 10 & 20 \\ 0 & 10 \end{bmatrix}$ ricavare i 4

coefficienti della wavelet di Haar per la sua decomposizione e verificare numericamente come tramite la trasformata inversa sia possibile riottenere l'immagine originale.

Es.4 [6 Pt] 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 l'area 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.

Es.5 [6 Pt] 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 a 16 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, realizzarne il negativo.

Elenco di alcune
funzioni Matlab:

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

Soluzioni

- 1) La matrice corrisponde ad una trasformazione affine con una variazione di scala pari a 10 (ingrandimento) una rotazione di 45° in senso antiorario ed una traslazione di -5 lungo le ascisse e 5 lungo le ordinate (le trasformazioni sono eseguite nel suddetto ordine):

$$T = t \cdot R \cdot S = \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & 5 \end{bmatrix} \cdot \begin{bmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} & 0 \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 10\frac{\sqrt{2}}{2} & -10\frac{\sqrt{2}}{2} & -5 \\ 10\frac{\sqrt{2}}{2} & 10\frac{\sqrt{2}}{2} & 5 \end{bmatrix}$$

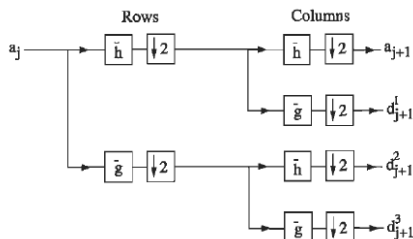
- 2) Si intende effettuare una convoluzione bidimensionale tra la matrice che rappresenta l'immagine e la matrice 3x3 indicata. In particolare tale matrice corrisponde al criterio di Sobel per l'estrazione di edge orizzontali. Tale filtro è costituito da un andamento passa-basso orizzontale e passa alto verticale. L'effetto sulla trasformata di Fourier 2D sarà quello di attenuare le componenti alle alte frequenze orizzontali e enfatizzare le componenti alle alte

frequenze verticali; in particolare, il filtro risulta essere separabile: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \cdot [1 \ 2 \ 1]$. Per quanto

riguarda l'implementazione in Matlab si faccia riferimento alla funzione *imfilter* ed al materiale delle esercitazioni presenti sul sito.

- 3) Utilizzando il solito schema di decomposizione si ottiene al primo passaggio $\begin{matrix} 5 & 15 \\ -5 & -5 \end{matrix}$ ed al

secondo passaggio, secondo lo schema sottoriportato, $\begin{matrix} 10 & 5 \\ -5 & 0 \end{matrix}$



- 4) La trasformazione applicata per ridurre la dimensionalità delle feature in modo supervisionato (conoscenza a priori delle classi di appartenenza dei dati) sarà:

$$y = A^T x$$

Dove:

A ($N \times M$) matrice di trasformazione le cui colonne sono gli autovettori corrisp. agli M autovalori più alti della matrice:

$$C = \text{inv}(S_w) * S_b$$

- S_w matrice della varianza complessiva intra-cluster

$$S_w = \sum_j p_j \text{Cov}_j$$

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

$$\text{Cov}_j = \text{covarianza dati classe } j$$

- S_b matrice della varianza complessiva inter-cluster

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

$$\mu_j = \text{centroide dati classe } j$$

$\mu = \text{centroide complessivo dati}$

■ Si calcolano i rispettivi centroidi e le matrici S_w e S_b

```
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, 16, 'nodither');`
`figure; imshow(img_ind, map_ind); title('Conversione Truecolor --> Indexed (mappa ad 8 valori)');`
- `imgray= rgb2gray(img_rgb);`
- `negativo=255- imgray;`
`figure; imshow(negativo); title('Conversione Truecolor --> Gray-scale -> negativo');`