

### 5.2.2 Trasformata discreta del coseno (DCT)

La codifica compressa di un'immagine gray-scale o a colori viene effettuata nella maggior parte dei casi per mezzo di un'operazione matematica nota con il termine di trasformata, la quale permette di passare dalla rappresentazione dei pixel nello spazio, è cioè da una matrice contenente i valori della luminosità di ciascun pixel (bitmap), a una rappresentazione equivalente espressa in termini di frequenze.

Questa descrizione è analoga a quella di un suono composto da sinusoidi (armoniche) di diversa ampiezza, la cui somma genera un'onda di forma qualsiasi. Nel caso dell'immagine si lavora su una funzione bidimensionale, che viene vista come somma di onde in direzione orizzontale e verticale.

Le onde a bassa frequenza rendono conto dell'andamento generale della luminosità dell'immagine, mentre quelle ad alta frequenza codificano i dettagli e il disturbo introdotto dall'elettronica dei dispositivi di acquisizione. Dal momento che le immagini fotografiche hanno generalmente un andamento continuo, la rappresentazione nel dominio delle frequenze è costituita da una matrice quadrata di valori, dei quali soltanto quelli corrispondenti alle frequenze più basse, raggruppati nell'angolo in alto a sinistra, sono significativi, mentre quelli corrispondenti alle frequenze più elevate hanno valore molto basso.

Dividendo poi ogni coefficiente della matrice trasformata per un certo valore specificato in una tabella è possibile attenuare ulteriormente le componenti meno significative, in modo da ottenere una matrice contenenti numerosi valori nulli.

Questa operazione è nota con il termine di *quantizzazione* e rappresenta lo stadio maggiormente responsabile della perdita di qualità e nello stesso tempo l'operazione che permette di ottenere la compressione più elevata.

Esistono vari tipi di trasformate: Fourier, Coseno, Hadamard, Karhunen-Loeve, ecc., ciascuna delle quali ha proprietà e complessità differenti. La trasformata di Fourier, ad esempio, riesce a concentrare molto bene tutta l'informazione dell'immagine trasformata in pochi coefficienti, ma comporta tempi di calcolo notevoli. La trasformata di Hadamard invece è molto veloce, ma poco efficace.

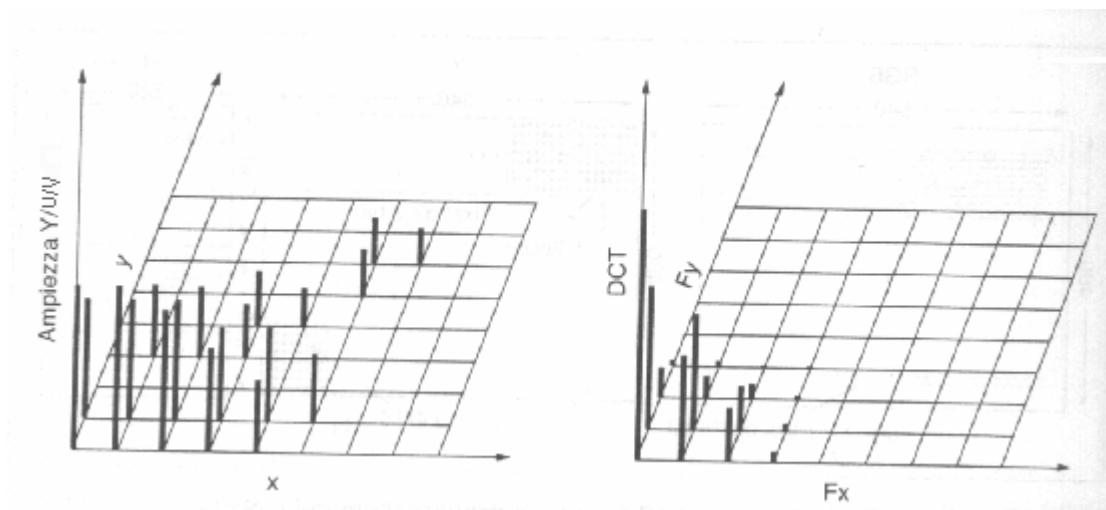
I gruppi di standardizzazione JPEG e MPEG hanno optato, allora, per la *Trasformata Discreta del Coseno* (DCT) che provvede alla compressione spaziale, essendo capace di rivelare le variazioni di informazione tra un'area e

quella contigua trascurando le ripetizioni, e inoltre, essa rappresenta un ottimo compromesso complessità-prestazioni.

La complessità dell'algoritmo di calcolo della DCT cresce tuttavia molto rapidamente con le dimensioni dell'immagine. Per questo motivo la DCT non viene applicata all'intera immagine, ma su piccoli blocchi di 8x8 pixel, nei quali viene scomposta ciascuna componente.

La DCT permette la trasformazione di un blocco di 8x8 pixel dell'immagine in un insieme di 8x8 numeri, detti coefficienti DCT, che descrivono il contenuto di frequenza (alte e basse) di tale blocco. L'elemento DCT (0,0) è il valore medio del blocco. Gli altri elementi dicono quanto potere spettrale è presente e ciascuna frequenza spaziale.

In teoria la trasformata DCT è fedele, ma in pratica poiché si utilizzano numeri in virgola mobile e funzioni trascendenti si introducono errori di rappresentazione che perdono un po' d'informazione. Di solito questi elementi decadono rapidamente allontanandosi dall'origine (0,0), come si vede in figura.



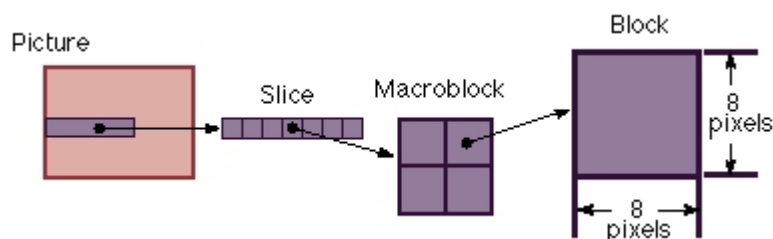
L'MPEG per comprimere i singoli fotogrammi suddivide l'immagine in blocchi di 16x16 pixel, che a causa del formato colore 4:2:0 corrispondono a 6 blocchetti 8x8 pixel, 4 blocchi di luminanza e 2 di crominanza:



dove ogni blocchetto è un insieme di 8x8 pixel contenenti i valori numerici (0-255) relativi a luminanza Y e crominanza U e V.

Ciascun insieme di 16x16 pixel prende il nome di **macroblocco**. Ogni frame viene suddiviso in un certo numero di strisce contigue di macroblocchi. La funzione delle *slice* è quella di fornire una protezione nei confronti degli errori di trasmissione: nel caso in cui parte dei dati all'interno di un frame risultino errati, il decodificatore è in grado di ripartire correttamente dalla slice successiva, riuscendo a recuperare il resto del frame. Le slice forniscono quindi il supporto per tutte le funzioni di sincronizzazione.

Ogni slice può contenere un numero variabile di macroblocchi e può partire da un punto qualsiasi del frame.



Un frame 352x288 contiene 396 macroblocchi di 16x16 pixel disposti come una matrice 22x18.

Macro1	Macro2	.....	Macro18
Macro19	Macro20	.....	Macro36
.....	.....	.....	.....
Macro379	.....	.....	Macro396

Considerato che ciascuno dei 396 macroblocchi sono in realtà costituiti da 6 blocchi di 8x8 pixel, avremo che l'encoder, per ciascuno dei 2376 blocchi (396 \* 6), esegue la DCT che trasforma gli 8x8 pixel nel loro contenuto nel campo delle frequenze. Si crea, così, una nuova matrice di 8x8 valori detti *coefficienti DCT* in cui in alto a sinistra ci sono le basse frequenze e in basso a destra le alte frequenze

(che dovranno essere il più possibile scartate). Bisogna dire che, prima di effettuare la DCT, sulle matrici Y, U, V viene eseguita una operazione di *shift a sinistra* di 128:

$$f(i,j) = f(i,j) - 128$$

in questo modo il range dei valori del frame non è più da 0 a 255 ma da -128 a 127. Questo shift facilita e ottimizza il calcolo della DCT.

Vediamo adesso come si calcolano i coefficienti DCT. Con  $f(i,j)$  si indica il valore in posizione  $(i,j)$  del blocco 8x8 di partenza. Con  $F(u,v)$  si indica invece il coefficiente DCT della matrice delle frequenze che è ottenuto applicando la seguente formula:

$$F(u, v) = \frac{1}{4} C(u)C(v) \sum_{u=0}^7 \sum_{v=0}^7 f(i, j) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right)$$

con  $u = 0..7$  e  $v = 0..7$  e  $C(x)$  che rappresenta un parametro così definito:

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{se } x = 0 \\ 1 & \text{altrimenti} \end{cases}$$

Per la decodifica, invece, si utilizza la seguente:

$$f(i, j) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right)$$

Con  $i = 0..7$  e  $j = 0..7$ . Questa trasformata restituisce il valore  $f(i,j)$  contenuto in un blocco 8x8 del frame decompresso.

Vediamo un esempio:

Frame (pixels)		Frequenze (coefficienti DCT)
32		94
31		62
32		33
32		7
32		0
32		0
32		0
32		0
32		0
32		72
32		21
32		15
29		0
32		0
32		0
32		0
32		0
32		19
32		4
32		2
32		0
32		0
32		0
32		0
32		0
32		3
32		0
31		1
32		0
30		0
32		0
32		0
31		0
32	DCT	0
32		0
32		0
32		0
32		0
28		1
32		0
32		0
32		0
32		0
32		0
32		2
32		0
32		0
32		0
32		0
32		0
31		0
32		0
32		0
32		0
33		0
32		0
31		0
32		0
32		0
32		0
33		0
32		0
32		0
32		0
30		0

<p>64 pixel con una luminosità simile tra loro (ad esempio la luminosità di un cielo scuro uniforme).</p> <p>0 = nero 255 = bianco</p>		<p>In alto a sinistra: basse frequenze In basso a destra: alte frequenze</p> <p>0 = presenza nulla di quella frequenza</p> <p>255 = presenza massima di quella frequenza</p>
--------------------------------------------------------------------------------------------------------------------------------------------	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Il coefficiente in alto a sinistra rappresenta il valore medio di luminosità di tutto il blocco e occorre rappresentarlo nel migliore dei modi.

Vediamo un esempio reale:



La striscia in alto contiene 10 blocchetti di 8x8 pixel di una immagine mentre quella in basso i 10 corrispondenti 8x8 coefficienti DCT. Le parti chiare della striscia superiore sono lo sfondo di un cielo, quelle viola sono fiori in primo piano mentre le parti in verde rappresentano la foresta sullo sfondo. Il tutto è rappresentato con un grosso zoom che permette la visualizzazione dei pixel.

E' evidente come il blocchetto 1 contiene uno sfondo pressoché uniforme e, nel campo delle frequenze, l'unico coefficiente fortemente diverso da zero (il puntino bianco) è quello in alto a sinistra; gli altri sono tutti uguali a zero (puntini neri).

Nei blocchetti 3, 4, 5 la presenza dei fiori (il viola) crea delle discontinuità e pertanto delle alte frequenze. Nelle relative DCT esistono dei valori sempre concentrati in alto a sinistra in cui ho le diverse gradazioni di grigio; osservo come anche in questo caso le altissime frequenze video sono pressoché nulle, infatti, i puntini in basso a destra sono tutti neri.

Quanto visto dimostra in maniera evidente come sono distribuiti i coefficienti della DCT di un blocco di 8x8 pixel di una immagine.

La DCT di per sé non crea nessuna compressione, infatti, se il decoder (il player MPEG) esegue l'operazione inversa, la IDCT, si riottiene il blocco 8x8 pixel senza aver guadagnato nulla. Per memorizzare la luminosità (valori nella prima colonna dell'esempio visto prima) degli 8x8 pixel mi occorrono 64 byte e lo stesso me ne servono per memorizzare i 64 coefficienti della DCT.

Il vantaggio della DCT è quello di aver “trasportato” i valori significativi in alto a sinistra e valori quasi tutti uguali a destra.