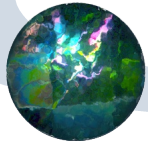
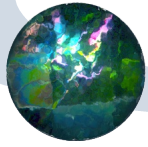


Cenni alla compressione delle immagini



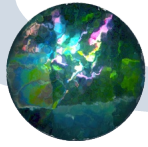
Introduzione

- Buona parte delle informazioni viene veicolata sotto forma di immagini
- Le immagini vengono gestite dai calcolatori come una matrice di digital picture elements, ossia pixels
- L'apparenza dell'immagine dipende da
 - il tipo di immagine
 - la sua risoluzione



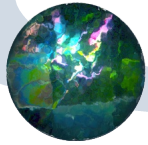
Tipi di immagini & risoluzioni

- a due livelli (bianco & nero)
 - es. i fax
- in scala di grigio
- a colori
- punti per pollice (dot per inches - dpi)
 - 600 x 600 – risoluzione attuale di una stampante laser di livello medio-basso
 - 1200 x 1200 – fotocompositrice economica
 - 4800 x 4800 – fotocompositrice ad alta risoluzione



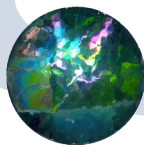
Immagini a due livelli: lo standard CCITT per i fax

- fax: facsimile
- **CCITT** Comité Consultatif International Téléphonique et Télégraphique, fa parte dell'**ITU** International Telecommunication Union, una delle agenzie specializzate delle Nazioni Unite
- Alla fine degli anni '70 il CCITT cominciò a pensare ad uno standard per la trasmissione di fax
- 1980 standard “CCITT Group 3”
 - group 1 & 2 sono tentativi precedenti, che utilizzavano un encoding semplificato e tecniche di modulazione molto semplici, ottenendo una trasmissione molto lenta



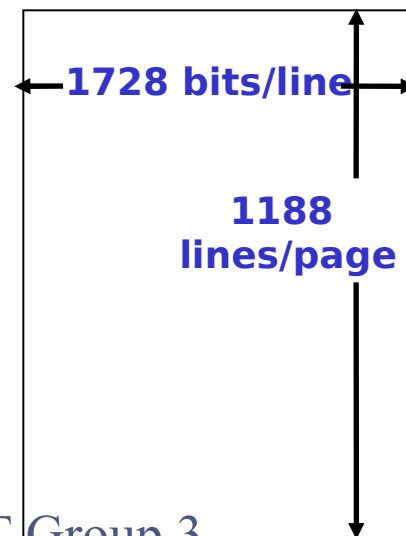
CCITT Group 3 - I

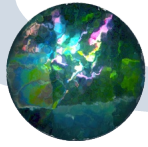
- E' lo standard più comune per la trasmissione dei fax
- E' accettato in tutto il mondo. Praticamente tutte le macchine fax lo supportano
- Utilizza degli algoritmi di compressione studiati per le immagini a due livelli



CCITT Group 3 - II

- Dimensione del foglio: A4 internazionale (non il formato paper comune negli US)
- risoluzione standard 204x98 dpi (200x100)
- alta risoluzione 204x196 dpi (200x200)
- immagini a due livelli, i.e. 1 bit/pixel
- dim immagine: 1728x1188 bit alla risoluzione standard
=> circa 2 Mbit
- Velocità trasmissione: 4.8 Kbit/s
- oggi è più alta, 14.4 – 33.6 Kbit/s
- A 4.8Kbit/s a risoluzione standard una pagina richiederebbe 430 sec, ma solo 1 minuto in media con gli algoritmi CCITT Group 3



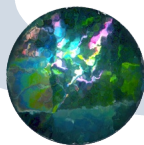


Codifica run-length

- Ogni linea è composta da sequenze di pixel dello stesso colore



- Si conta il numero di elementi di ciascun colore
- Esempio 3w 4b 9w 2b 2w 6b 5w 2b 5w...



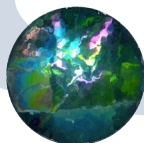
G3 1D

- La codifica Group 3 monodimensionale (G3 1D) è chiamata Modified Huffman (MH) dal momento che codifica le sequenze

con un codice di Huffman predefinito

- Per mantenere la sincronizzazione bianco/nero, ogni linea inizia con una sequenza di bianchi eventualmente lunghezza zero

<i>n</i>	white runs	black runs
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111



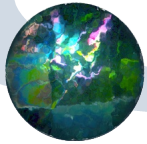
G3 1D



1000 011 10100 11 0111 0010 ...

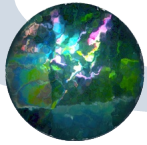
- le parole di codice di questo codice di Huffman predefinito sono stato calcolate dalle probabilità delle sequenze in documenti manoscritti

<i>n</i>	white runs	black runs
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111



G3 1D

- Dal momento che ogni linea ha 1728 bit, dobbiamo definire una parola di codice per ogni sequenza di bianchi e neri lunga fino a 1728
- Poiché le sequenze corte sono più probabili di quelle lunghe, le sequenze sono codificate in forma additiva
 - c'è una parola di codice di terminazione ed una di aggregazione (makeup)
 - Sequenze lunghe da 0 a 63 sono codificate con una sola parola di codice di terminazione
 - Sequenze più lunghe invece hanno una o più parole di codice di aggregazione ed una parola di codice di terminazione
- Ogni linea termina con un simbolo di EOL composto da undici 0 ed un 1

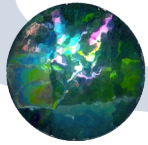


G3 2D

- La codifica Group 3 bidimensionale (G3 2D) è chiamata *Modified READ* (MR) dal momento che è una variante di un codice sviluppato in precedenza e chiamato READ (Relative Element Address Designate)
- Molte immagini hanno un alto livello di coerenza verticale fra linee consecutive

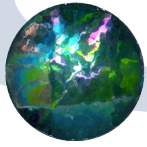


- I bit di *cambiamento* sono codificati rispetto alla posizione di cambiamento “vicina” e dello stesso colore nella linea precedente (linea di riferimento)

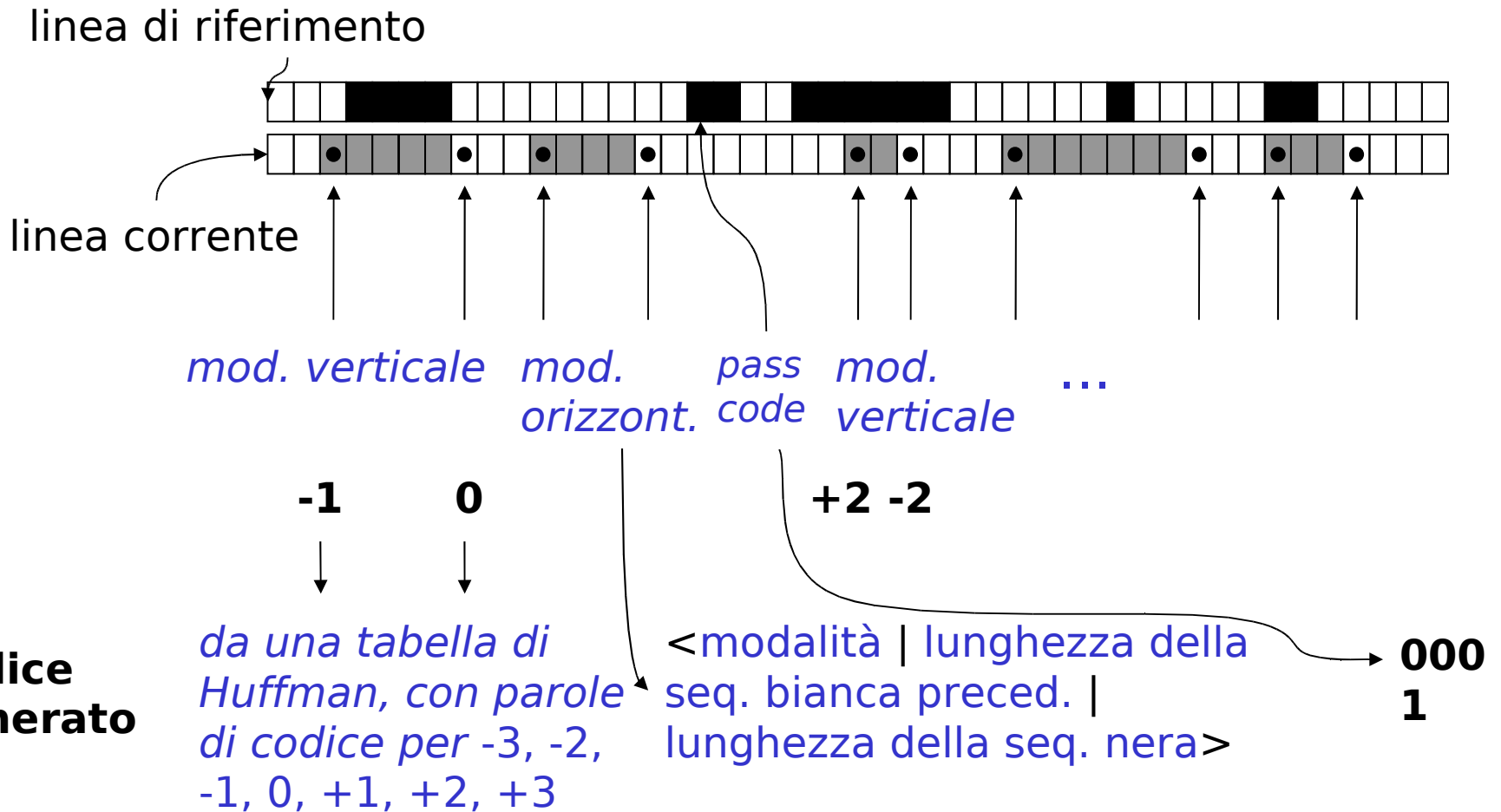


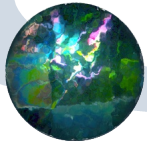
G3 2D

- “Vicina” significa in un intorno di 3 pixel
- Se ci sono elementi di cambiamento nella linea corrente che non hanno corrispondenti nella linea di riferimento => si torna alla modalità *orizzontale* (1D)
- Al contrario, se la linea di riferimento ha una sequenza che non ha controparti nella linea corrente => un codice speciale (pass code)



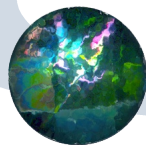
G3 2D





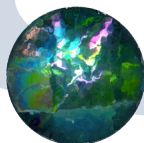
G3 2D

- La codifica bidimensionale è più fragile rispetto agli errori di trasmissione
 - Nella G3 1D un errore poteva causare problemi nell'intera linea, ma la sincronizzazione è ristabilita dall'EOL
 - Qui un errore nella linea di riferimento si propaga facilmente a tutte le altre linee
 - Per questo motivo, vi è una linea di riferimento ogni k linee (cioè $k-1$ linee sono codificate rispetto ad ogni linea di riferimento)
 - risoluzione standard $\Rightarrow k = 2$
 - alta risoluzione $\Rightarrow k = 4$



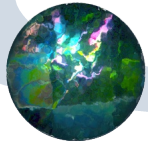
Le performance di compressione del CCITT fax standard

- Risoluzione standard ($\sim 200 \times 100$ dpi)
 - G3 1D \Rightarrow 0.13 bits/pixel 57s. per A4 a 4.8 Kbps
 - G3 2D (k=2) \Rightarrow 0.11 bits/pixel 47s. per A4 a 4.8 Kbps
- Alta risoluzione ($\sim 200 \times 200$ dpi)
 - G3 2D (k=4) \Rightarrow 0.09 bits/pixel 74s. per A4 a 4.8 Kbps
- La compressione è molto buona per le immagini da ufficio, dove le sequenze sono lunghe
- Sarebbe stata molto peggiore per immagini naturali a due livelli



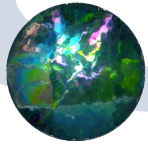
Immagini a molti colori: perché usare la compressione senza perdita?

- La compressione con perdita è spesso da preferire perché crea immagini significativamente più compresse, mantenendo una buona qualità
- Esistono comunque situazioni nelle quali utilizzare un'immagine approssimata non è accettabile
 - immagini mediche
 - documenti storici
 - immagini con valore legale



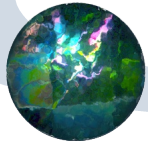
Immagini a molti colori: compressione senza perdita

- standard **GIF**
- standard **PNG**
- **JPEG-LS**
 - E' uno standard piuttosto recente. Lo standard JPEG includeva una modalità di compressione senza perdita, ma le sue performance non erano vicine allo stato dell'arte
 - stima del valore del pixel utilizzando un contesto piuttosto semplice: soluzione efficace ed a basso costo
 - www.hpl.hp.com/loco



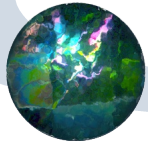
Il formato GIF - I

- Adottato da CompuServe per minimizzare il tempo richiesto per scaricare immagini da connessione via modem
- Il formato di compressione immagini senza perdita più usato fino al 1995
- Descrizione a 8 bit per pixel
- Immagini a 256 colori, ma con la possibilità di utilizzare una color map



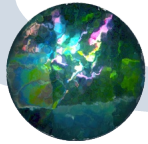
Il formato GIF - II

- La color map può essere specificata per ogni immagine o essere omessa
 - se specificata, viene inclusa come header nel file immagine, in forma non compressa
 - La color map è composta da 256 componenti a 24 bit, che specificano 256 colori RGB
- Lo schema di compressione usato è LZW
 - I simboli dell'alfabeto sono i 256 colori della color map più un codice di “clear” e un codice di “fine delle informazioni” code



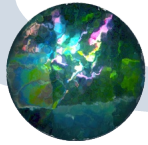
Il formato GIF - III

- Anche se questa feature non è molto usata, i file GIF possono contenere più di una immagine, ed è possibile condividere la color map
- L'informazione codificata LZW è raggruppata in blocchi preceduti da un contatore, in modo da poter saltare un'immagine senza decomprimerla
- Nel 1995 Unisys annunciò che ci sarebbero state delle royalties da pagare sulle implementazioni del formato GIF a causa di un vecchio brevetto che essi detenevano su LZW
- Questo fatto accelerò lo sviluppo di un nuovo formato di immagini compresso senza perdita, progettato per essere di pubblico dominio ed allo stato dell'arte



Il formato PNG - I

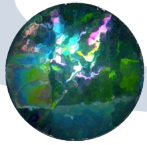
- **Portable Network Graphics** (pronuncia “ping”)
- Usa l'algoritmo **gzip**
- Attraverso alcune migliorie la compressione ottenuta è del 10-30% migliore rispetto al formato GIF
- Di default codifica i pixel nell'ordine di scansione raster, ma vari altri metodi sono disponibili
 - è possibile codificare la differenza orizzontale, cioè la differenza tra il valore del pixel corrente ed il precedente o la differenza verticale, cioè la differenza con il pixel superiore
 - la differenza dalla media, ossia la differenza rispetto alla media del pixel superiore e precedente
 - ...



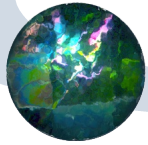
Il formato PNG - II

- E' possibile usare più di 256 colori, fino a 16 bit in scala di grigi and 48 bit a colori
- GIF usa uno speciale valore del pixel per indicare la trasparenza, PNG invece usa 256 differenti valori per pixel, consentendo di realizzare immagini che progressivamente si dissolvono nel background
- Sembra inevitabile che il formato PNG assuma gradatamente il ruolo di formato standard per la compressione senza perdita delle immagini, rimpiazzando il formato GIF

Immagini a molti colori: compressione con perdita

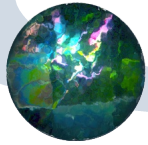


- Le immagini digitali sono esse stesse un approssimazione dei fenomeni reali che in realtà sono analogici
- le tecniche di compressione con perdita consentono di ottenere compressioni molto buone con una modesta perdita di dettagli
- Ciò è utile per memorizzare e trasmettere immagini



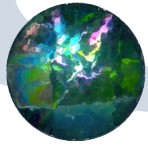
Continuous-tone images: lossy compression

- JPEG
- JPEG2000
 - un recente formato di codifica delle immagini che utilizza tecniche di compressione allo stato dell'arte basate sulle wavelet
 - estensione del file: jp2
 - In caso di immagini molto compresse, a parità di dimensione delle immagini, la qualità percepita delle immagini jpeg2000 è migliore rispetto a quella ottenibile con JPEG



Il formato JPEG - I

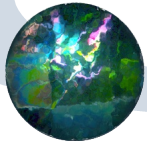
- JPEG è uno standard definito dal **Joint Photographic Experts Group** nel 1992
- E' stato concepito per trasmettere immagini a 64 Kbps
- Ha una modalità con perdita ed una senza perdita (non molto usata in realtà e oggi sostituita dal recente standard JPEG-LS)
- Con la compressione con perdita è possibile ottenere una buona qualità ad appena 1 bit/pixel
- La complessità dell'implementazione è ragionevole

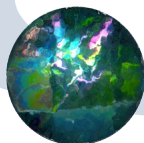


Il formato JPEG - II

- Può essere usato con immagini in scala di grigio e a colori
- Ogni canale dello spazio colore (RGB, YUV...) è trattato separatamente
- Consente un tipo di trasmissione progressiva che è molto più adatta della trasmissione raster per il Web

Raster vs. progressive transmission





La codifica JPEG - I



168	161	161	150	154	168	164	154
171	154	161	150	157	171	150	164
171	168	147	164	164	161	143	154
164	171	154	161	157	157	147	132
161	161	157	154	143	161	154	132
164	161	161	154	150	157	154	140
161	168	157	154	161	140	140	132
154	161	157	150	140	132	136	128



**Trasformata
Coseno
Discreta (DCT)**

214	48	-2	20	-10	0	1	-4
33	-7	10	11	4	-1	14	-9
-6	-3	7	-8	2	-2	4	10
8	-9	4	3	-14	10	5	5
-12	5	-1	-1	-14	8	-5	0
4	9	-7	2	4	-6	-13	5
1	-1	2	-1	1	2	-2	-4
0	0	0	1	3	-2	-5	-3



Quantizzazione



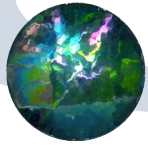
13	4	0	1	0	0	0	0
3	-2	1	1	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



**Codifica
binaria**

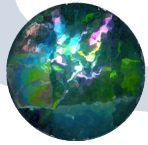


```
10111011001000111110110111001001
11001000111001111111110011111111
1001111111110011010XXXXXXXXXXXXXX
```



La codifica JPEG - II

- L'immagine è divisa in quadrati di 8x8 pixel
- Fase di preprocessing
- Viene applicata la Trasformata Coseno Discreta (DCT) su ogni quadrato
- Quantizzazione dei coefficienti
- Codifica dello stream di bit

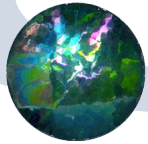


Preprocessing: trasformazione dello spazio colore & sottocampionamento

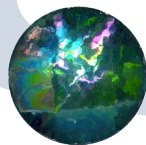
- da **RGB** a **YUV**

- La componente Y rappresenta la luminosità di un pixel, mentre le componenti U e V insieme rappresentano la tinta e la saturazione
- L'occhio umano può vedere un maggior numero di dettagli nella componente Y rispetto a quanto riesce a fare in U e V, che perciò possono essere compresse in modo più aggressiva
 - 4:4:4 nessun sottocampionamento
 - 4:2:2 sottocampionamento orizzontale di un fattore 2
 - 4:2:0 sottocampionamento sia orizzontale che verticale

Trasformata Coseno Discreta (DCT) - I



- La **trasformata coseno discreta** (DCT) è una trasformazione correlata a quella di Fourier, simile all'trasformata di Fourier discreta (DFT), ma utilizzando solo numeri interi
- Viene usata nel formato JPEG perché è abbastanza semplice da implementare efficientemente



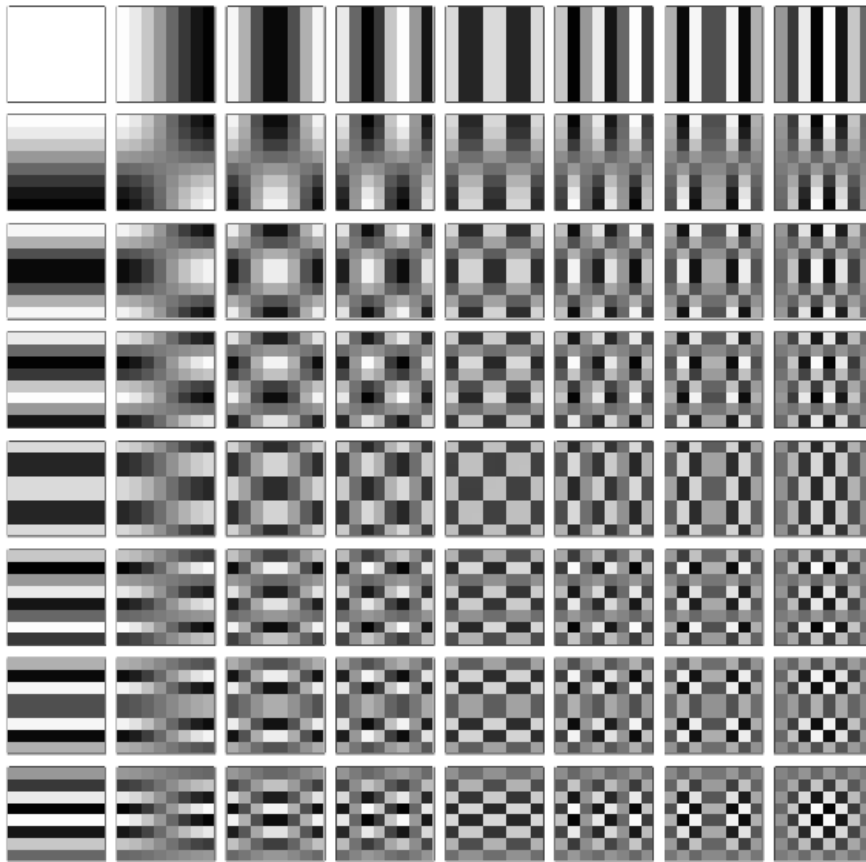
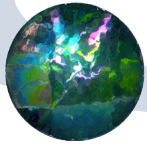
Trasformata Coseno Discreta (DCT) - II

$$B(k_1, k_2) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} 4 \cdot A(i, j) \cdot \cos \left[\frac{\pi \cdot k_1}{2 \cdot N_1} \cdot (2 \cdot i + 1) \right] \cdot \cos \left[\frac{\pi \cdot k_2}{2 \cdot N_2} \cdot (2 \cdot j + 1) \right]$$

dove

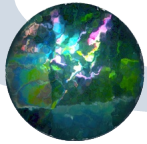
- il blocco è $N_1 \times N_2$ pixel (in JPEG, 8x8)
- $A(i, j)$ è il valore del pixel in posizione (i, j)
- $B(k_1, k_2)$ è il coefficiente DCT di posizione (k_1, k_2)
- bassi valori di k_1 corrispondono a basse frequenze verticali,
bassi valori di k_2 a basse frequenze orizzontali
- In generale le frequenze più alte hanno valori dei coefficienti molto piccoli

Trasformata Coseno Discreta *(DCT) - III*

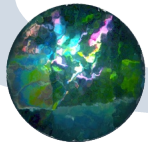


- La base delle funzioni della DCT
- ogni quadrato 8x8 si riduce a 64 coefficienti

Trasformata Coseno Discreta *(DCT) - IV*



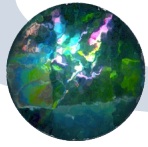
- Conoscendo con infinita precisione i 64 coefficienti DCT è possibile ricostruire esattamente i pixel del quadrato
- Ma
 - la precisione è finita
 - si ha quantizzazione dei coefficienti (sempre)
 - Alcuni coefficienti relativi alle alte frequenze non sono trasmessi. Questo consente di ottenere una compressione maggiore senza sacrificare molto la qualità dell'immagine, dato che l'occhio umano è meno sensibile alle alte frequenze



Quantizzazione - I

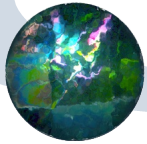
- La matrice DCT ottenuta viene scalata dividendo ciascuna componente per un fattore diverso
- Tale fattori sono stati decisi sulla base della sensibilità umana ai cambiamenti a ciascuna frequenza
- In pratica la matrice dei fattori è di solito la seguente

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

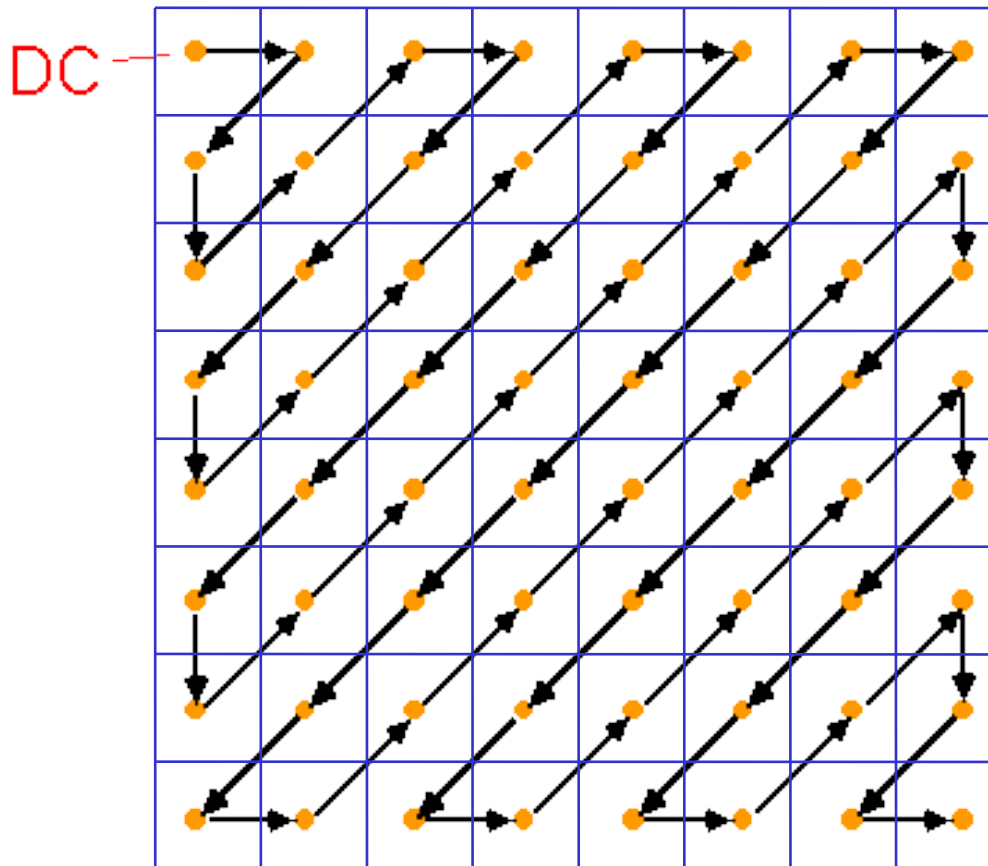


Quantizzazione - II

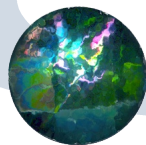
- Quindi tutti i valori sono arrotondati all'intero più prossimo
- Questo conduce ad avere un buon numero di zeri nelle zone delle alte frequenze, dove i fattori sono maggiori



Scansione a zig-zag



- I coefficienti delle basse frequenze sono trasmessi prima di quelli relativi alle alte frequenze
- Questo consente la visualizzazione progressiva dei blocchi 8x8



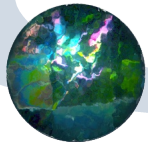
Trasmissione raster o progressiva

- **Trasmissione raster**

- Prima i coefficienti DCT del blocco in alto a sinistra, poi quelli della parte superiore dell'immagine e così via

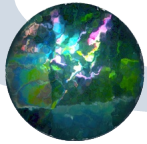
- **Trasmissione progressiva**

- prima tutti i coefficiente $(0,0)$, poi tutti gli $(0,1)$ e così via, seguendo la scansione a zig-zag in ogni blocco



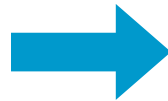
Codifica binaria

- DCT(0,0) ha di solito una variazione molto piccola da un blocco al successivo, dal momento che è il valor medio
 - Per questo motivo è conveniente codificare la differenza rispetto al valore nel blocco precedente
- Tipicamente lo stroma di bit è codificato con Huffman
 - E' possibile utilizzare la codifica aritmetica, guadagnando un po' di compressione al costo di una decodifica più lunga
- I codici di Huffman sono predefiniti, ma è possibile costruire il codice ottimo e inserirne la descrizione nello stream



La decodifica JPEG

10111011001000111110110111001001
11001000111001111111100111111111
1001111111110011010XXXXXXXXXXXXXX



Decodifica binaria



13	4	0	1	0	0	0	0
3	-2	1	1	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



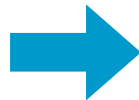
Dequantizzazione

Some values are lost!

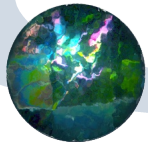
214	48	0	20	0	0	0	0
33	-7	10	11	0	0	0	0
0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
-12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



DCT inversa

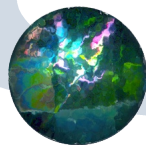


buona qualità, ma la ricostruzione non è esatta



JPEG: performance - I

Original size of color image (Kb)				313.076
Original size of B/W image (Kb)				104.437
Quality Factor	Color JPEG		B/W JPEG	
	File Size (Kb)	Comp. Ratio	File Size (Kb)	Comp. Ratio
75	23.039	13.59	21.02	4.97
20	8.457	37.02	7.599	13.74
5	4.009	78.09	3.257	32.07
3	3.268	95.80	2.522	41.41



JPEG performances - II

Originale



Fattore qualità 75



Fattore qualità 20



Fattore qualità 3

