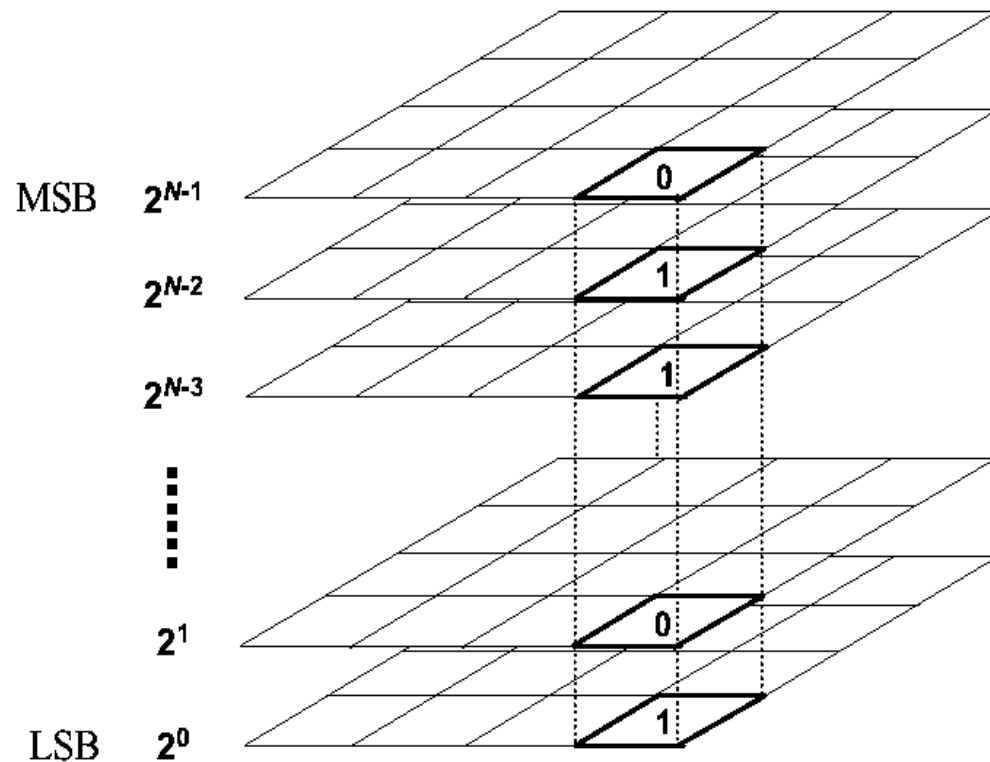


Bit Plane



Bit-plane

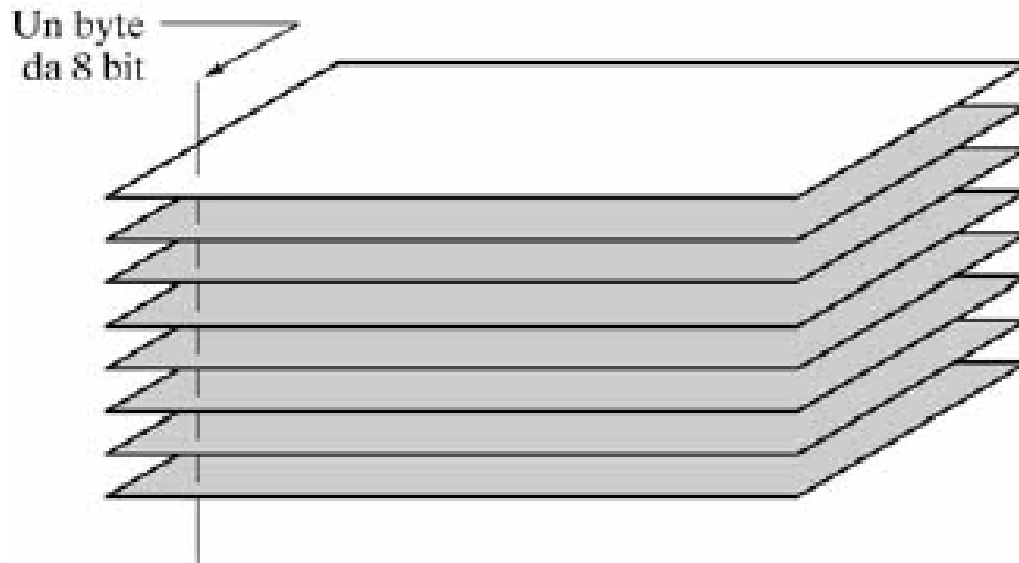
Un'immagine con una profondità colore di N bit può essere rappresentata da N piani di bit (bit-planes), ciascuno dei quali può essere vista come una singola immagine binaria. In particolare si può indurre un ordine che varia dal **Most Significant Bit (MSB)** fino al **Least Significant Bit (LSB)**.

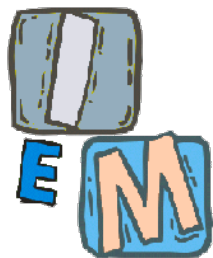




Bit-plane - Definizione

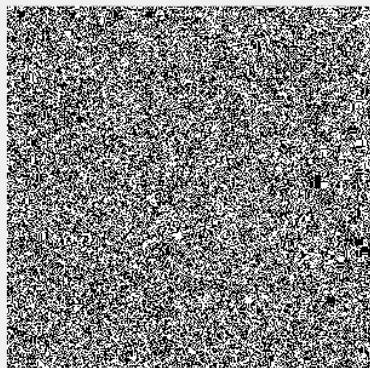
Il bit plane di un'immagine digitale a N bit, è un'insieme di N immagini binarie (piani), in cui l'immagine i -esima contiene i valori dell' i -esimo bit della codifica scelta.



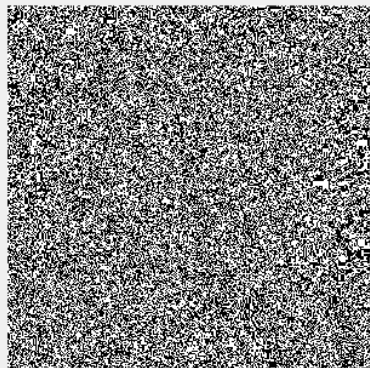


Bit plane di Lena

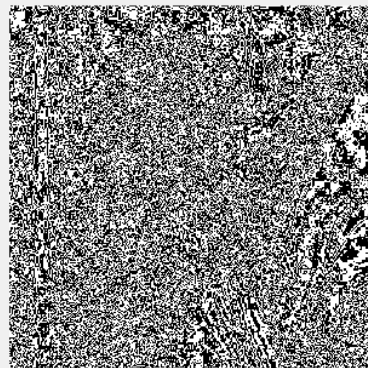
2^0



2^1



2^2



2^3



2^4



2^5



2^6

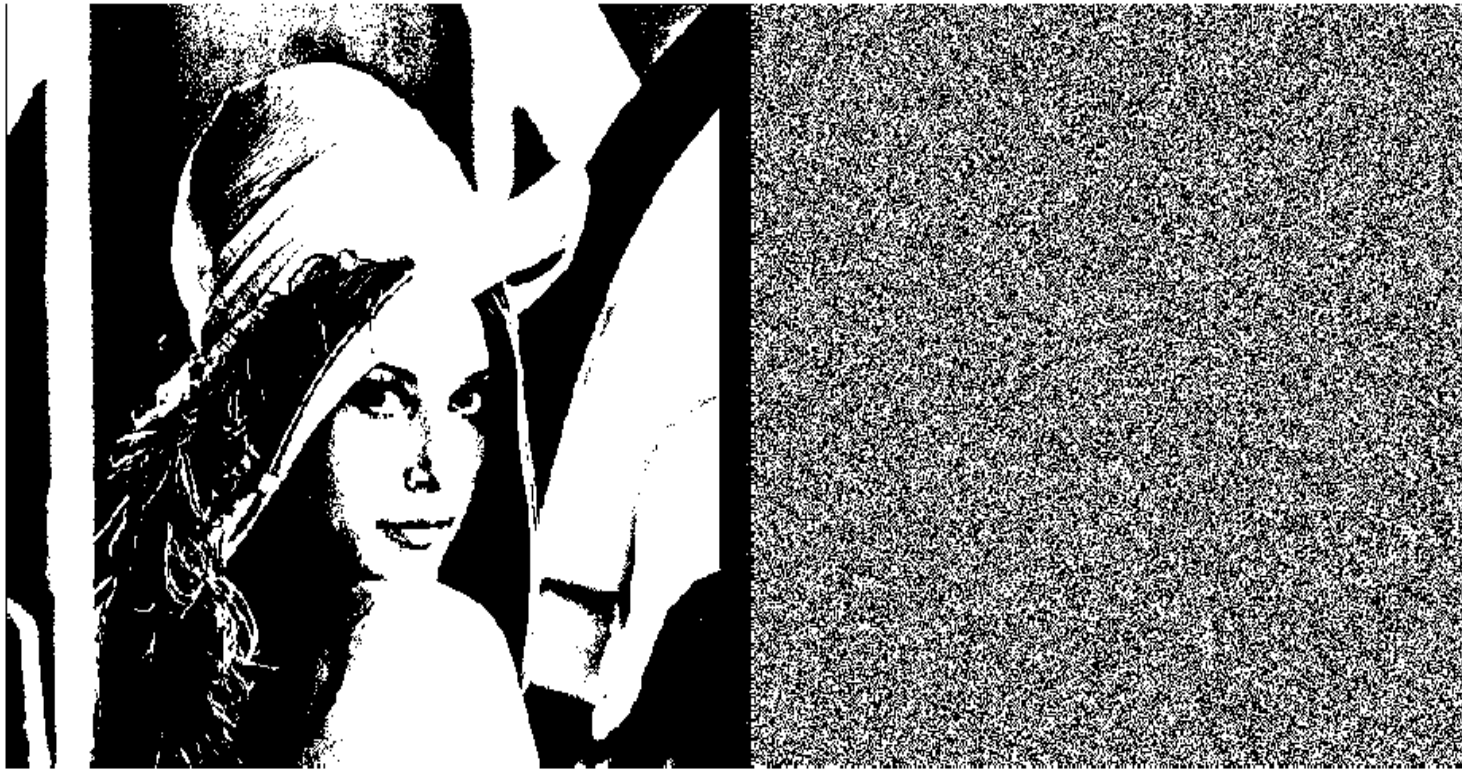


2^7





Bit-planes – Codifica binario puro

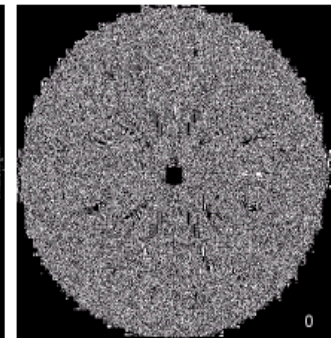
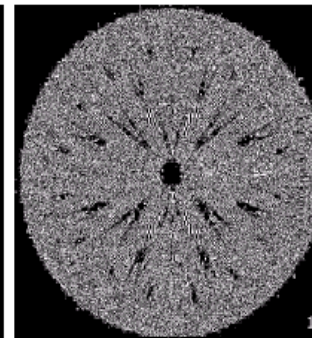
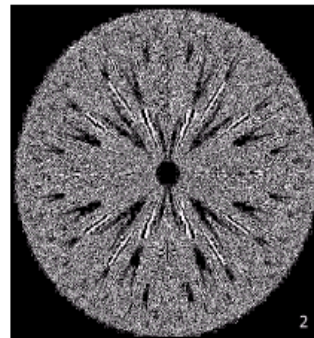
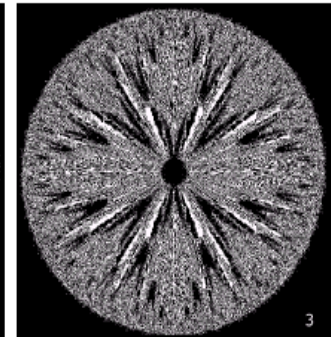
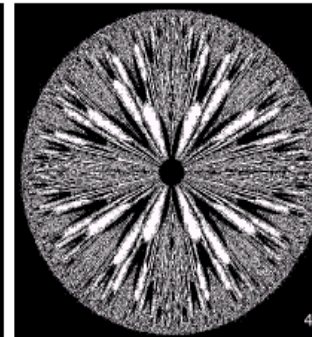
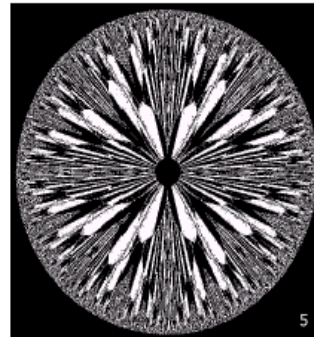
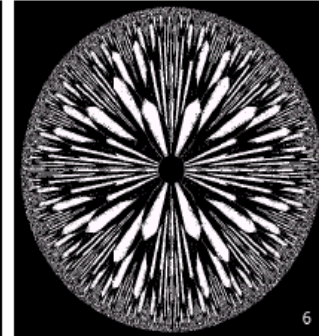
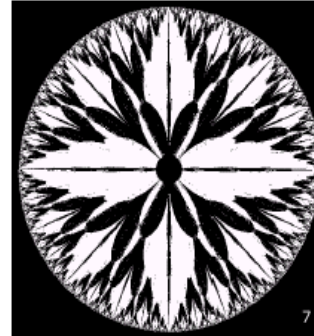
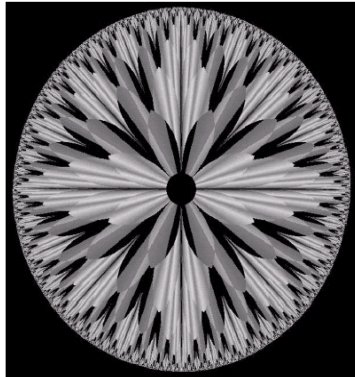


Most Significant bit (**MSB**)

Least Significant bit (**LSB**)



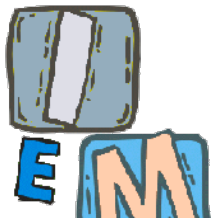
Bit-planes binario puro





Bit-planes binario puro: Osservazioni

- Se si usa la codifica in **binario puro** i piani di bit più significativi contengono informazioni sulla struttura dell'immagine, mentre quelli via via meno significativi forniscono i dettagli sempre più piccoli.
- Si noti che solo i piani dal 7 al 3 contengono dati significativi dal punto di vista visuale.
- Il rumore delle immagini e gli errori di acquisizione sono più evidenti nei piani bassi.



Bit-planes binario puro



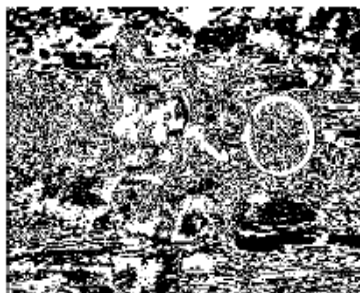
7



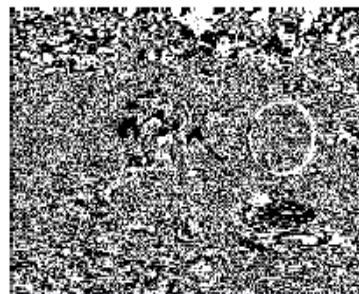
6



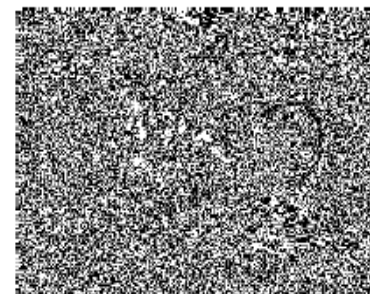
5



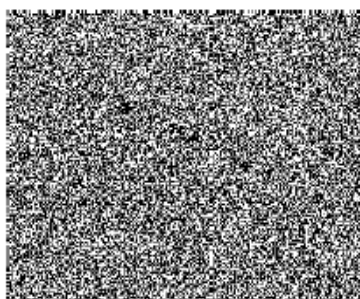
4



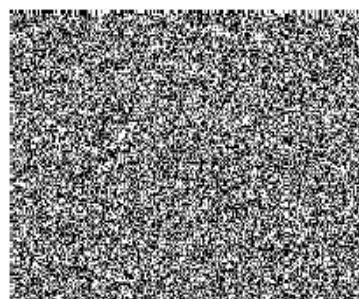
3



2



1



0

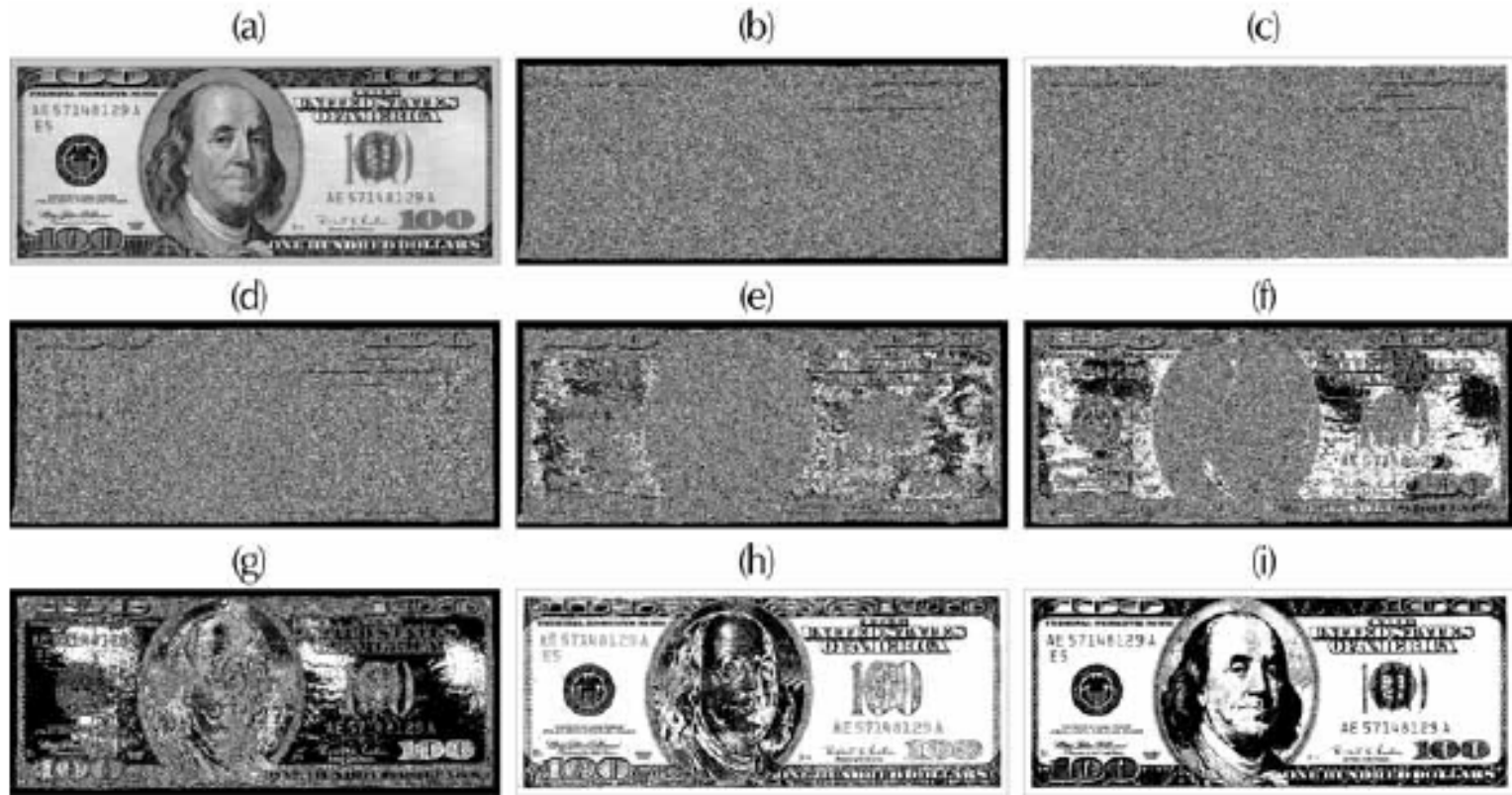


Usi bit-planes binario puro:

- Questo genere di scomposizione è molto utile per eliminare tutti i valori compresi in un certo range.
- Ad esempio, se si vogliono eliminare tutti i grigi compresi tra 32 e 63, è necessario porre a 0 il quinto bit, e quindi tutto il piano 5.
- Chiaramente, questa osservazione e le precedenti, sono valide se la codifica utilizzata è quella in **binario puro**.



Bit-plane binario puro - Esempio



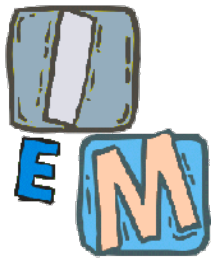
(a) Immagine a 8 bit in scala di grigio. Da (b) a (i) i piani di bit da 0 a 7. Ogni piano è una immagine binaria.



Bit-plane binario puro - Esempio



(a) Immagine ricostruita ci i piani di bit da 7 e 6. L'immagine assume solamente i valori 0, 64, 128, 192; (b) usando i piani di bit 7, 6, 5. L'immagine assume solamente, oltre lo zero, alcuni valori nel range 32 - 224; (c) i piani di bit 7, 6, 5, 4. L'immagine assume, oltre lo zero, alcuni valori nel range 16 - 240.



Ricostruzione senza un piano di bit

without 2^0



without 2^1



without 2^2



without 2^3



without 2^4



without 2^5



without 2^6



without 2^7





Bit-Plane - Problema

Se la codifica usata è quella in **binario puro**, allora risulta evidente uno svantaggio: **una piccola variazione può ripercuotersi su tutti i piani.**

Esempio: Se un pixel ha ad esempio intensità 127 (01111111) e il suo adiacente ha intensità 128 (10000000) allora la transizione tra 0 e 1 si ripercuote su tutti i piani di bit.

Serve un codice in cui valori molto vicini abbiano codifiche binarie molto simili!



Soluzione – Codice Gray

Il codice Gray a m bit $g_{m-1} \dots g_1 g_0$ che corrisponde al numero in binario puro $a_{m-1} \dots a_1 a_0$ può essere calcolato con la formula

$$\begin{aligned} g_i &= a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2 \\ g_{m-1} &= a_{m-1} \end{aligned}$$

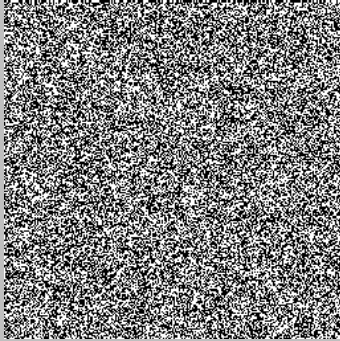
dove \oplus denota l'operatore XOR (OR esclusivo).

Il codice Gray gode della proprietà per cui ogni codeword differisce dalla precedente per un solo bit (distanza di Hamming = 1).

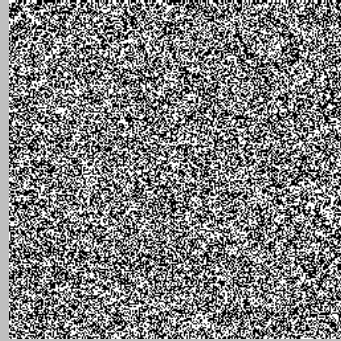


Binario puro VS Gray Code

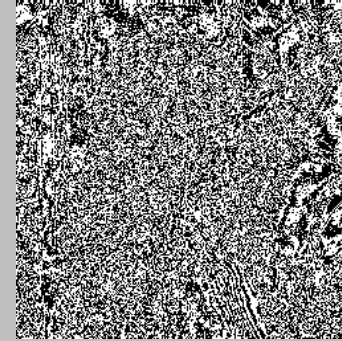
**BINARIO
PURO**



Piano 0



Piano 1

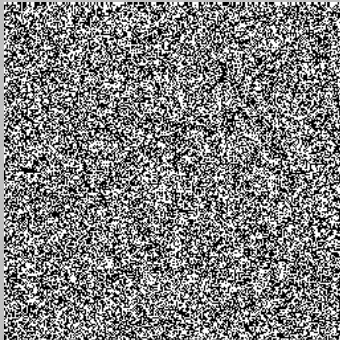


Piano 2

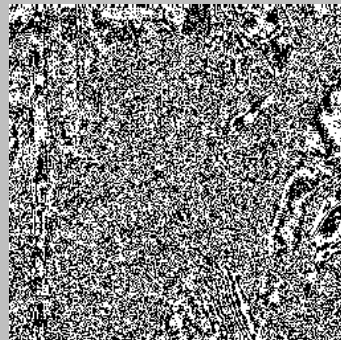


Piano 3

**CODICE
GRAY**



Piano 0



Piano 1



Piano 2



Piano 3



Binario puro VS Gray Code

**BINARIO
PURO**



Piano 4



Piano 5



Piano 6



Piano 7

**CODICE
GRAY**



Piano 4



Piano 5



Piano 6



Piano 7



Esempio con i numeri a 3 bit

Numero in decimale	Numero in binario puro	Numero in gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Si osservi il caso con maggiore variazione in binario, quello relativo ai numeri decimali 3 e 4. in questo caso a fronte di una grande variazione di bit nel binario puro si osserva la variazione di un solo bit nel gray code.

Inoltre, nel gray code ogni numero con il suo successivo differisce solo per la variazione di un bit.



- Tornando al nostro esempio dei numeri decimali 127 e 128.
- In binario puro abbiamo 01111111 per 127 e 10000000 per 128. I due numeri consecutivi hanno la variazione di tutti i bit.
- In gray code abbiamo 01000000 per 127 e 11000000 per 128, si noti la variazione di un solo bit.



Binario puro VS Gray Code

- Come si può osservare nell'esempio precedente, in base alla codifica i bit-plane presentano delle differenze.
- In particolare, i bit-plane delle immagini in codice Gray risultano tra loro più “**coerenti**” se confrontati con i rispettivi in binario puro. Se aumento l'intensità del pixel di 1 varierà infatti solo un bit (ossia solo un piano).
- Inoltre, il numero di **transizioni bianco-nero** nel singolo piano (complessità descrittiva) sono inferiori se si usa il codice Gray (es: confrontare i capelli di Lena tra i piani 6).
- Queste caratteristiche indicano una minore entropia (maggiore ridondanza) se si utilizza il codice Gray. Ciò significa che diventa più semplice comprimere a partire da immagini così codificate.



Binario puro VS Gray Code

ATTENZIONE!

- Dato che il significato associato ai bit è diverso tra le due codifiche, alcune proprietà di una non valgono per l'altra!
- Se si azzerano dei piani di bit in Gray code, si eliminano range di valori **diversi** (e meno significativi) rispetto a quelli in binario puro.
- Nonostante i dettagli e il rumore tenderanno a concentrarsi nei piani più bassi anche con il codice Gray, eliminare direttamente tali piani potrebbe introdurre artefatti indesiderati.