www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Capitolo 8
COMPRESSIONE DI IMMAGINI

La **compressione delle immagini** è il processo che riduce la quantità di dati necessari per rappresentare un'immagine, cioè una certa quantità di informazione. La distinzione tra dati e informazione è che i *dati* rappresentano *l'informazione*; visto che differenti quantità di dati possono rappresentare la stessa quantità di informazione, allora le rappresentazioni che contengono informazioni irrilevanti o ripetute contengono *dati ridondanti*.
Se supponiamo che *b* e *b'* denotano il numero di bit in due rappresentazioni della stessa informazione, la *ridondanza relativa dei dati R* della rappresentazione con *b* bit è:

$$R = 1 - \frac{1}{C_r}$$

dove *C* è il **rapporto di compressione**, o coefficiente di ridondanza, ed è definito come:

$$C = \frac{b}{b'}$$

dove *b* è il file originale, mentre *b'* è il file compresso.
La compressione si realizza quando la ridondanza viene ridotta o eliminata.

Solitamente le immagini risentono di tre tipi di ridondanza:

1- Ridondanza della codifica (*Coding Redundancy*):
i codici a 8 bit che vengono utilizzati per rappresentare le intensità delle immagini contengono più bit del necessario.
Più in dettaglio un codice è un sistema di simboli (lettere, numeri, bit) utilizzati per rappresentare una certa quantità di informazione. Ad ogni pezzo di informazione è assegnata una sequenza di *simboli codificati*, chiamati *codeword*. Il numero di simboli che costituisce ciascun codice è la sua *lunghezza*.

Come abbiamo già visto nel capitolo 3 se assumiamo che una variabile discreta casuale r_k rappresenti le intensità dell'immagine *MxN* e che $p_r(r_k)$ sia la probabilità di ciascuna r_k avremo:

$$p_r(r_k) = \frac{n_k}{MN} \quad k=0,1,2,\ldots,L-1$$

dove n_k è il numero di volte che l'intensità r_k appare nell'immagine.
Ora possiamo aggiungere che se il numero di bit utilizzati per rappresentare ciascun valore r_k è $l(r_k)$, il numero medio di bit richiesti per rappresentare ciascun pixel è:

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)\, p_r(r_k)$$

Quindi la lunghezza media dei codeword assegnati ai valori di intensità è ottenuta sommando i prodotti del numero di bit utilizzati per rappresentare ciascuna intensità per la probabilità che quella data intensità sia presente; il numero totale di bit richiesti per rappresentare un'immagine *MxN* è MNL_{avg}.
Questo tipo di ridondanza è quasi sempre presente quando le intensità di un'immagine sono rappresentate utilizzando un codice binario naturale, in quanto una codifica binaria assegna lo stesso numero di bit sia ai valori più probabili che a quelli meno probabili, introducendo una ridondanza nella codifica dei dati. Questo avviene perché alcune intensità sono molto più probabili di altre.

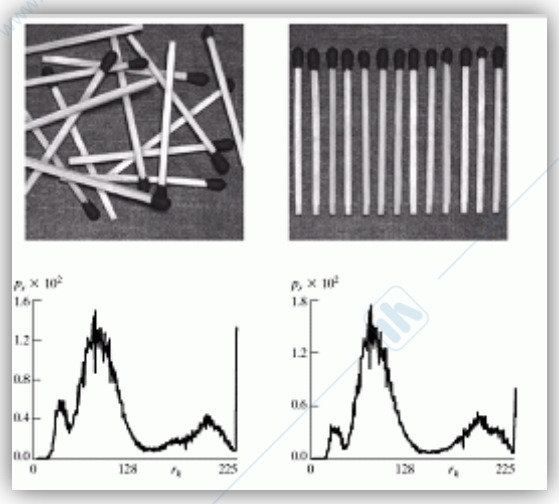
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

2- Ridondanza spaziale e temporale (*Interpixel Redundancy*):
nella maggior parte delle immagini i pixel sono *relazionati spazialmente* (ciascun pixel è simile ai pixel del suo intorno o dipende da esso), quindi l'informazione è replicata inutilmente nei pixel correlati e l'informazione contenuta in ogni singolo pixel è davvero piccola.
La ridondanza temporale invece è presente nei video.

La ridondanza spaziale può essere ridotta attraverso una rappresentazione più efficiente, ma non visiva, chiamata *mapping*. Il mapping può essere di due tipi: *reversibile* se i pixel originali possono essere ricostruiti senza errori a partire dall'insieme dei dati trasformati; *irreversibile* in caso contrario.
In dettaglio, nella rappresentazione dell'immagine sotto forma di sequenze di *coppie run-length* ciascuna coppia individua l'inizio di una nuova intensità e il numero dei pixel consecutivi che condividono quel valore. Quindi ciascuna linea di 256 pixel della rappresentazione originale è rimpiazzata in quella run-length da un singolo valore di intensità a 8 bit e da un valore di lunghezza pari a 256.

Vediamo un esempio:



Le due immagini hanno istogrammi identici e la codifica run-length può essere utilizzata per ridurre la ridondanza. La correlazione tra i pixel deriva dalle relazioni geometriche e strutturali degli oggetti presenti nell'immagine.
Il processo di codifica non altera il livello di correlazione tra i pixel all'interno delle immagini.

3- Informazione irrilevante (*Psychovisual Redundancy*):
è quel tipo di informazione che nelle immagini è meno importante perché viene ignorata dal sistema visivo umano e può essere eliminata in quanto non viene ridotta la percezione qualitativa dell'immagine. Questo avviene perché il sistema visivo umano opera una media delle intensità e *percepisce solo il valore medio* e ignora le piccole variazioni di intensità presenti.
Questo tipo di ridondanza si può eliminare perché l'informazione persa non è essenziale né per l'elaborazione visiva né per il fine ultimo dell'immagine; questo tipo di rimozione viene definito *quantizzazione*, cioè un'operazione non reversibile di mapping di un largo range di valori in input in un numero limitato di valori in output.

(a) Immagine originale con 256 possibili livelli di grigio
(b) Immagine ottenuta con una quantizzazione uniforme a 16 livelli. Notiamo che si introducono dei falsi contorni.
(c) Immagine ottenuta con la quantizzazione IGS improbe gray-scale) che sfrutta le peculiarità del sistema visivo umano.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Misurare l'informazione nelle immagini

La teoria dell'informazione ci aiuta a dire quanti bit sono realmente necessari per rappresentare un'immagine, cioè qual è la più piccola quantità di dati sufficiente a descrivere un'immagine senza che nessuna informazione vada persa.

Tutto parte dal fatto che la sorgente dell'informazione può essere modellata come un processo probabilistico e può essere quindi oggetto di misure. Un evento casuale *E* con probabilità *P(E)* contiene

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

unità di informazione.

Se *P(E) = 1* abbiamo un evento certo, quindi *I(E) = 0* cioè nessuna informazione è associata all'evento. Non c'è nessuna incertezza correlata all'evento, quindi nessuna informazione viene trasferita per "comunicare" che l'evento è accaduto; si verifica sempre.

La base del logaritmo determina l'unità utilizzata per misurare l'informazione; noi utilizziamo la base *2*, infatti l'unità di informazione è il *bit*.

Data una sorgente di eventi casuali e statisticamente indipendenti (quindi una sorgente senza memoria), a partire da un insieme discreto di eventi possibili $\{a_1, a_2, a_3, \ldots, a_j\}$ con probabilità $\{P(a_1), P(a_2), \ldots, P(a_j)\}$ *l'informazione media per la sorgente di output*, cioè l'**entropia** (di *primo ordine*), è data da:

$$H = -\sum_{j=1}^{J} P(a_j) \log P(a_j)$$

Le a_j sono i simboli di sorgente.

Se consideriamo un'immagine come output di una sorgente di intensità immaginaria senza memoria, possiamo utilizzare l'istogramma dell'immagine per stimare le probabilità dei simboli della sorgente. Quindi l'entropia diventa:

$$\tilde{H} = -\sum_{k=0}^{L-1} p_r(r_k) \log_2 p_r(r_k)$$

Otteniamo quindi l'informazione media della sorgente di intensità immaginaria in bit; importante ricordare che non è possibile codificare i valori di intensità dell'immagine con meno di $\tilde{H}$ bit/pixel. Dopo qualche esempio possiamo notare che il rapporto tra l'entropia e l'informazione in un'immagine è tutt'altro che intuitivo, infatti ci possono essere casi in cui sembra sia quasi assente informazione visiva, ma il valore dell'entropia è alto e viceversa, cioè valori bassi di entropia e molta informazione visiva.

Criteri di fedeltà

Per quantificare la natura e la quantità di informazione persa ci sono due criteri:

- O*ggettivo*:

misura l'errore tra l'immagine di input e quella di output. E' semplice e conveniente, ma si preferisce l'altro criterio per via del fatto che alla fine le immagini vengono viste dalle persone.

Due esempi di misure di fedeltà sono:

1- L'*errore quadratico medio* (root-mean-square) tra due immagini, che viene calcolato utilizzando le seguenti espressioni.

   L'errore *e* tra $f(x,y)$ e $\hat{f}(x,y)$:

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

   dove $\hat{f}(x,y)$ è un'approssimazione di $f(x,y)$, ottenuta dalla compressione e dalla successiva decompressione dell'input.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

L'errore *totale* tra le due immagini è dato da:

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y)-f(x,y)]$$

dove le immagini sono di dimensioni $M \times N$.
L'errore quadratico medio è dato dalla radice quadrata della media dell'errore al quadrato in una matrice $M \times N$:

$$e_{rms}=\left[\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y)-f(x,y)]^2\right]^{1/2}$$

2- *PSNR* che normalizza l'errore medio quadratico rispetto alla distorsione massima in maniera tale da poter meglio confrontare le prestazioni degli algoritmi di compressione su immagini differenti in maniera più uniforme. Maggiore è il valore del PSNR maggiore è la qualità registrata.

I metodi per la valutazione della qualità possono essere classificati in base alla presenza di una maggiore o minore *disponibilità del segnale di riferimento*, con il quale confrontare l'immagine distorta, nel seguente modo:

- ***FR (full reference)***: metrica di qualità nella quale si ha la completa disponibilità dell'immagine di riferimento e quindi dove si può realizzare un confronto completo tra l'immagine distorta, della quale si vuole conoscere la qualità, e l'immagine di riferimento.
- ***NR (no reference)***: metrica di qualità nella quale non si ha alcuna informazione sull'immagine di riferimento.
- ***RR (reduced reference)***: metrica di qualità nella quale l'immagine di riferimento non è interamente disponibile. Dall'immagine originale si estraggono solo alcune informazioni parziali che vengono elaborate per fornire una valutazione della qualità dell'immagine distorta presa in esame.

- S*oggettivo*:
è il criterio più appropriato e può essere fatto presentando un'immagine decompressa a un campione di osservatori e, successivamente, effettuando la media delle loro valutazioni.
Vediamo un esempio di una possibile scala di valutazione:

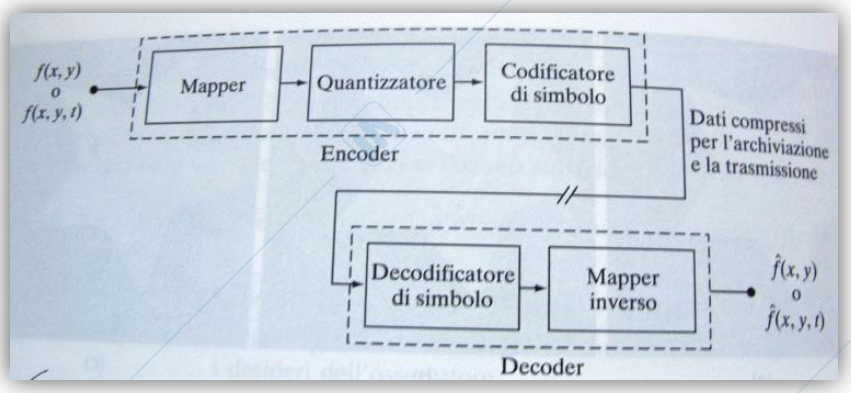
Valore	Valutazione	Descrizione
1	Eccellente	Un'immagine di estrema qualità che soddisfa al meglio i desideri dell'osservatore.
2	Buona	Un'immagine ad alta qualità, che soddisfa abbastanza l'osservatore. L'interferenza è trascurabile.
3	Passabile	Un'immagine di qualità accettabile. L'interferenza è trascurabile.
4	Marginale	Un'immagine di bassa qualità, che si spera di poter migliorare. L'interferenza comincia ad essere visibile.
5	Inferiore	Un'immagine di bassissima qualità, che si può ancora guardare. L'interferenza è chiaramente presente.
6	Inutilizzabile	Un'immagine pessima, impossibile da guardare.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Modelli di compressione di un'immagine

L'immagine mostra quali sono le componenti e le fasi principali del processo di compressione di un'immagine:



Quindi le due componenti principali che formano un sistema di compressione di un'immagine sono l'encoder e il decoder:

<u>*Processo di codifica o compressione*</u>

1. l'**encoder** (codificatore): esegue la compressione su di un'immagine di input *f(x,y)*.
   In dettaglio rimuove le ridondanze definite precedentemente attraverso tre operazioni indipendenti:
   a) ***Mapper***: trasforma il segnale in un formato che riduce la ridondanza spaziale e temporale; questa operazione è solitamente reversibile e può ridurre la quantità di dati richiesti per rappresentare l'immagine.
   b) ***Quantizzatore***: riduce l'accuratezza dell'output del mapper in accordo a criteri di fedeltà prestabiliti. In poche parole elimina l'informazione irrilevante dalla rappresentazione compressa.
   Questo processo è irreversibile, quindi in caso di compressione lossless di informazione il quantizzatore non può essere presente e deve essere omesso.
   c) ***Codificatore di simbolo*** (encoder di simbolo): genera un codice a lunghezza variabile o fissa per rappresentare l'output del quantizzatore ed esegue il mapping dell'output secondo il codice.
   I codeword più brevi vengono assegnati ai valori di output del quantizzatore che risultano essere più frequenti.
   Con questi tre processi abbiamo eliminato le tre ridondanze presentate precedentemente.

<u>*Processo di decodifica o di decompressione*</u>

2. il **decoder** (decodificatore): esegue la decompressione di un'immagine compressa dall'encoder, quindi genera un'immagine ricostruita $\hat{f}(x, y)$.
   Le componenti del decoder sono:
   a) ***decodificatore di simbolo***: esegue l'operazione inversa operata dal codificatore di simbolo.
   b) ***mapper inverso***: esegue l'operazione inversa del mapper dell'encoder.
   La *quantizzazione* non è presente, e non può essere inclusa in un modello di decodifica, in quanto ha come risultato una *perdita irreversibile di informazione*.

In base al risultato ottenuto alla fine del processo possiamo avere due tipi di sistemi:
- <u>*error free*</u> (o lossless): quando l'immagine ricostruita è uguale all'immagine *f(x,y)* di input. Questo tipo di codifica abbiamo già detto che è obbligatoria in alcuni ambiti come quello medico, per non compromettere l'accuratezza della diagnosi.
- <u>*lossy*</u>: quando l'immagine ricostruita in output è distorta.

Quel programma che è in grado di codificare e decodificare si chiama *codec*.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

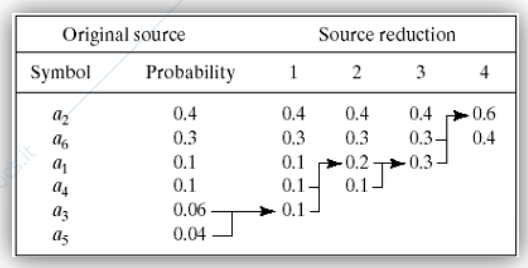
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Metodi di base per la compressione

Codifica di Huffman

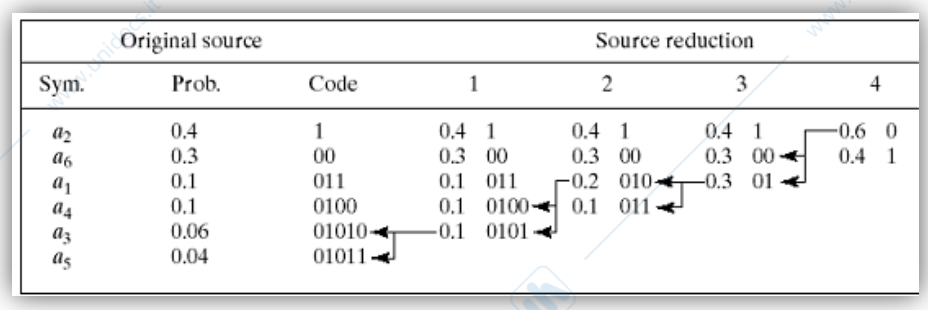
La codifica di Huffman è uno dei principali metodi di compressione. Quando si codificano i simboli di una sorgente di informazione uno alla volta, permette di ottenere un codice ottimale.
I passi da seguire per ottenere la codifica sono:

- creare una serie di riduzioni della sorgente *ordinando le probabilità dei simboli*, combinando i simboli con probabilità più bassa e sostituendoli con un singolo simbolo nella riduzione di sorgente successiva.
  All'estrema sinistra un insieme di sorgenti di simbolo e le loro probabilità sono ordinate dall'alto al basso *in ordine decrescente*.
  Per formare la prima riduzione, le due probabilità più in basso vengono combinate per *formare un simbolo composto*; il simbolo composto e la sua probabilità vengono posizionati nella prima colonna della riduzione di sorgente in modo che le probabilità della sorgente ridotta continuino ad essere ordinate in modo decrescente.
  Questo processo continua fino al raggiungimento di una sorgente ridotta che possiede solo due simboli.



- Codificare ogni sorgente ridotta cominciando dalla sorgente più piccola, procedendo fino alla sorgente originale.
  In una sorgente a due simboli il codice binario di lunghezza minima è costituito da 0 e 1.

Il codice ottenuto viene chiamato *codice a blocco* perché ciascuna sorgente di simboli viene mappata in una sequenza fissa di simboli. Una volta ottenuto il codice, la codifica/decodifica si attua utilizzando la tabella generata; infatti ogni sequenza di Huffman di simboli può essere decodificata esaminando i singoli simboli che compongono la sequenza da sinistra a destra. Il codice è istantaneamente e unicamente decodificabile. E' *istantaneo* perché ogni codeword in una sequenza di simboli può essere decodificato senza fare riferimento ai simboli successivi. E' *unicamente decodificabile* perché ogni sequenza di simboli può essere decodificata in un solo modo.



Procedura di Huffman per l'assegnazione di un codice

Le ultime considerazioni sono che il metodo di Huffman non è banale se abbiamo un elevato numero di simboli da codificare; inoltre quando le probabilità dei simboli della sorgente possono essere stimate a priori, si può ottenere un codice ottimale attraverso codici di Huffman già calcolati, come avviene con le codifiche JPEG e MPEG.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Codifica LZW (Lempel-Ziv-Welch)

Questo tipo di codifica assegna delle *codeword a lunghezza fissa* a sequenze di simboli a lunghezza variabile. La caratteristica chiave di questa codifica è che non è necessaria una conoscenza a priori della probabilità di occorrenza dei simboli che devono essere codificati.

I passi da seguire per ottenere la codifica sono:

- All'inizio della codifica si costruisce un *dizionario* (codebook) contenente i simboli della sorgente da codificare. Per le immagini monocromatiche a 8 bit alle 256 intensità (0,1,2…,255) vengono assegnate le prime 256 *parole* del dizionario.
  Quando l'encoder esamina sequenza dopo sequenza i pixel dell'immagine, le sequenze di intensità che non sono nel dizionario vengono man mano posizionate nella locazione successiva inutilizzata. La *dimensione del dizionario* è un parametro importante del sistema, infatti se è troppo piccolo il rilevamento delle sequenze dei livelli di intensità combacianti sarà meno probabile; se è troppo grande la dimensione delle codeword comprometterebbe le performance della compressione.
- L'immagine viene codificata attraverso la visita dei suoi pixel da sinistra verso destra e dall'alto verso il basso. Ciascun valore di intensità durante la visita è concatenato con una variabile chiamata "sequenza attualmente riconosciuta", che in principio è nulla o vuota. Il dizionario cerca ciascuna delle sequenze concatenate:
  - *se la trova*, la sostituisce con una nuova sequenza concatenata e riconosciuta.
  In questo caso nessun codice di output è stato generato e il dizionario non si è alterato.
  - *se non la trova*, l'indirizzo della sequenza attualmente riconosciuta è l'output del successivo valore codificato, inoltre una sequenza concatenata ma non riconosciuta viene aggiunta al dizionario e la sequenza attualmente riconosciuta viene inizializzata al valore corrente del pixel.

Dai passi necessari per effettuare la codifica emerge che la creazione del dizionario avviene contemporaneamente alla codifica dei dati. Un decoder LZW genera un dizionario di decompressione identico che decomprime simultaneamente lo stream di dati codificati.
Quando si verifica un *overflow del dizionario* (cioè si riempie completamente) è necessario iniziarne uno nuovo opportunamente inizializzato.
Questa codifica quindi permette di comprimere un'immagine andando a determinare le numerose *ripetizioni delle sequenze di intensità*.

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Esempio di codifica LZW

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Codifica a blocchi mediante trasformata

In questo tipo di codifica l'immagine viene divisa in piccoli *blocchi non sovrapposti di uguale dimensione* che vengono elaborati in modo indipendente utilizzando una trasformata 2-D. Ogni blocco (o sottoimmagine) viene mappato, tramite una trasformata lineare e reversibile (il nostro amico Fourier), in un insieme di *coefficienti della trasformata*, che vengono poi quantizzati e codificati. Nella maggior parte dei casi molti coefficienti hanno una piccola magnitudo, che può quindi essere scartata, introducendo una piccola distorsione nell'immagine.

L'**encoder** quindi esegue quattro operazioni:

- *costruzione delle sottoimmagini*: hanno dimensione *nxn* e vengono trasformate per generare MN/n^2 matrici di valori trasformati.
- *trasformazione*: serve per decorrelare i pixel di ciascuna sottoimmagine, raccogliendo quanta più informazione possibile in un ristretto numero di coefficienti della trasformata.
- *quantizzazione*: si quantizzano grossolanamente i coefficienti che contengono la minor quantità di informazione, perché questi coefficienti hanno un impatto minimo sulla qualità della sottoimmagine ricostruita.
- *codifica*: dei coefficienti quantizzati tramite un codice a lunghezza variabile.

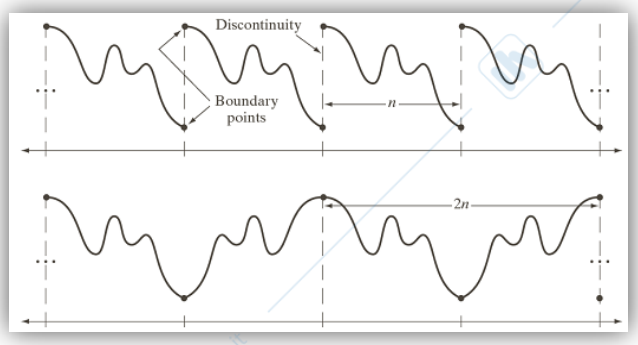
Possiamo avere due tipi di codifica:

1) *Codifica mediante trasformata adattativa*: quando i passi della codifica possono essere adattati al contenuto dell'immagine
2) *Codifica mediante trasformata non adattativa*: quando i passi della codifica rimangono fissi per tutte le sottoimmagini

Il **decoder**, come al solito, implementa la sequenza inversa di operazioni rispetto all'encoder ad eccezione della quantizzazione.
Da tutto ciò emerge che il processo di compressione non avviene durante la fase di trasformazione, ma avviene grazie alla *quantizzazione dei coefficienti trasformati*.

Scelta della trasformata

Come abbiamo visto prima questo tipo di codifica si basa sulle trasformate e la scelta di questa dipende dalle risorse computazionali disponibili e dall'entità di errore nella ricostruzione che può essere tollerata. Una delle più utilizzate è la *trasformata discreta del coseno DCT*, per via del fatto che raggiunge un buon compromesso tra la capacità di conservare l'informazione e la complessità di calcolo. Il vantaggio che la contraddistingue dalle altre è che può essere *implementata in un singolo circuito integrato*, inglobando una notevole quantità di informazione in un piccolo numero di coefficienti, riducendo l'*artefatto blocking* cioè l'aspetto "a blocchi" che si ottiene quando i contorni delle sottoimmagini rimangono visibili. Questo tipo di artefatto avviene perché i pixel di confine assumono il valore medio delle discontinuità formate in quei punti.



(a) Periodicità DFT
(b) Periodicità DCT

Notiamo che nella DCT non sono presenti le discontinuità che caratterizzano la DFT.

In generale comunque ci interessano le trasformate che raccolgono o ridistribuiscono la *maggior parte dell'informazione su pochi coefficienti*, in quanto forniscono migliori approssimazioni della sottoimmagine e quindi con minori errori di ricostruzione.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Importante ricordare che l'errore quadratico medio di un'immagine *MxN* è uguale all'errore quadratico medio di una singola sottoimmagine.

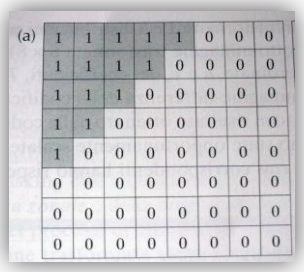
Scelta della dimensione della sottoimmagine

La dimensione della sottoimmagine va ad influenzare anch'essa l'errore di codifica e la complessità di calcolo. La scelta sulla dimensione della sottoimmagine solitamente è effettuata per fare in modo che la *ridondanza tra le sottoimmagini adiacenti sia ridotta a un livello accettabile* e meglio se è una potenza di 2. Le dimensioni più utilizzate sono *8x8* e *16x16*.

L'**allocazione dei bit** è l'insieme dei processi di eliminazione, quantizzazione e codifica dei coefficienti della sottoimmagine trasformata.

Nei sistemi di codifica mediante trasformate ci sono due modi per selezionare i coefficienti residui:

- *Codifica zonale*: si basa sul concetto che l'informazione viene descritta come un fenomeno incerto e viene implementata utilizzando una *singola maschera fissa per tutte le sottoimmagini*.
  In dettaglio, quando i coefficienti della trasformata hanno varianza massima trasportano con sé la maggior parte dell'informazione di un'immagine e vengono quindi maggiormente preservati nel processo di codifica. I coefficienti di massima varianza vengono posizionati attorno all'origine della trasformata dell'immagine.



- *Codifica threshold* (a soglia): codifica di tipo adattativa, cioè *la locazione dei coefficienti della trasformata da preservare varia per ogni sottoimmagine*.
  Il concetto principale è che, per ciascuna sottoimmagine, i coefficienti della trasformata di magnitudo maggiore danno un contributo più significativo alla qualità della sottoimmagine. Per effettuare il threshold sulle sottoimmagini trasformate possiamo utilizzare tre modi:
  1) una soglia singola e globale applicata a tutte le sottoimmagini.
     Il livello di compressione differisce da immagine a immagine, perché dipende dal numero di coefficienti che eccede la soglia globale.
  2) una soglia differente per ogni sottoimmagine (codifica N-largest).
     Per ogni sottoimmagine viene scartato lo stesso numero di coefficienti. Il tasso di codifica è costante e noto a priori.
  3) soglia variabile in funzione della locazione di ciascun coefficiente della sottoimmagine. Come nel primo caso il valore di codifica è variabile.

JPEG

E' uno degli standard di compressione più utilizzati per le immagini statiche e a tono continuo. Sono possibili tre differenti modalità di codifica:

a) *sistema sequenziale baseline*:
   è un sistema baseline per la *codifica lossy* che si basa sulla DCT.
   La compressione viene eseguita in tre passi: calcolo della DCT; quantizzazione e assegnazione del codice a lunghezza variabile.
   L'immagine viene partizionata in blocchi da 8x8, da sinistra verso destra e dall'alto verso il basso. Per ogni blocco (sottoimmagine) 8x8 i suoi 64 pixel sono traslati di livello sottraendo 2^{k-1}, dove 2^k è il numero massimo dei livelli di intensità. Viene poi calcolata la trasformata discreta del coseno 2-D di ciascun blocco, questa viene a sua volta quantizzata e riordinata utilizzando un modello a zigzag per formare una sequenza 1-D dei coefficienti quantizzati. Questo vettore viene costruito utilizzando uno schema incrementale rispetto alle frequenze spaziali, infatti prima ci sono le basse frequenze poi quelle alte.
   Questo tipo di codifica trae vantaggio delle lunghe *sequenze di zeri* che vengono normalmente create dal riordinamento.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

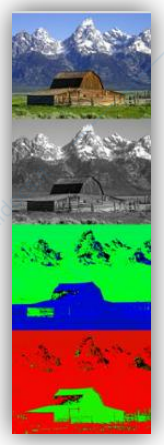
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

b) *Sistema di codifica esteso* per le applicazioni che richiedono un livello maggiore di compressione e accuratezza, oppure con ricostruzione progressiva.
c) *Sistema di codifica lossless* per la compressione reversibile.

In generale i passi da seguire per effettuare una compressione JPEG sono quattro:

1. Trasformare RGB in YIQ/YUV, separando l'intensità dal colore.
   Il modello **YUV** è più simile alla percezione umana dei colori, rispetto al modello RGB. La componente Y è la luminosità; U e V sono le componenti colore. YUV non è uno spazio colore assoluto, ma un modo per effettuare l'encoding dell'informazione RGB.

(a)
(b)
(c)



(d) Immagine RGB
(e) Componente Y
(f) Componete U
(g) Componente V

2. Identificare i dati ridondanti utilizzando la DCT.
   L'immagine è suddivisa in blocchi da 8x8 e ogni blocco è analizzato con la DCT.
3. Quantizzare i dati rimanenti.
   E' un procedimento lossy, nel quale i valori dei coefficienti della DCT sono divisi da un valore costante differente per ogni blocco e i risultati sono arrotondati al valore intero più vicino.
4. Effettuare l'encoding lossless sul risultato.

Vediamo infine le differenze principali tra la compressione JPEG e la compressione GIF:

Profondità di colore:

- JPEG salva in full color (24 bits/pixel)
- GIF salva in 8 bits/pixel

Preservazione dei contorni:

- JPEG tende a sfocare I bordi
- GIF è efficiente con le immagini disegnate

B&W:

- JPEG non è adatto per le immagini a due toni
- GIF per le immagini in toni di grigio è lossless

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

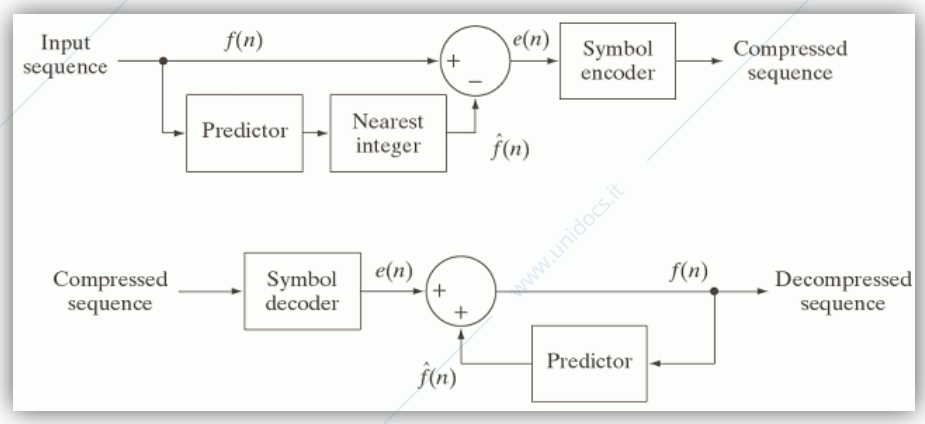
www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

Codifica predittiva

La codifica predittiva è un tipo di codifica che viene utilizzata in molti standard tra cui anche il JPEG, in quanto raggiunge ottimi risultati. Si basa sull'eliminazione delle ridondanze interpixel, spaziali e temporali, estraendo e codificando solo la *nuova informazione* di ogni pixel, cioè la differenza tra il valore attuale e il valore predetto di ogni pixel.

Può essere:

- *Codifica predittiva lossless*:
  L'immagine mostra le componenti di questo sistema



Notiamo che, come al solito, abbiamo due componenti principali che svolgono operazioni opposte: l'encoder (codificatore) e il decoder (decodificatore). In questo sistema però abbiamo una componente aggiuntiva: il *predittore*, sia per l'encoder che per il decoder.

I passi da seguire per effettuare la codifica/decodifica sono i seguenti:

- Quando i successivi campioni del segnale discreto di input del tempo vengono introdotti nell'encoder, il predittore genera un valore atteso basandosi su un certo numero di campioni precedenti.

- L'output $\hat{f}(n)$ del predittore viene arrotondato all'intero più vicino e viene calcolato l'errore di predizione tramite la seguente espressione:

$$e(n) = f(n) - \hat{f}(n)$$

  il valore ottenuto è codificato con un codice a lunghezza variabile per generare il successivo elemento nello stream di dati compressi

- Il decoder ricostruisce *e(n)* da una codeword a lunghezza variabile ed esegue l'operazione inversa

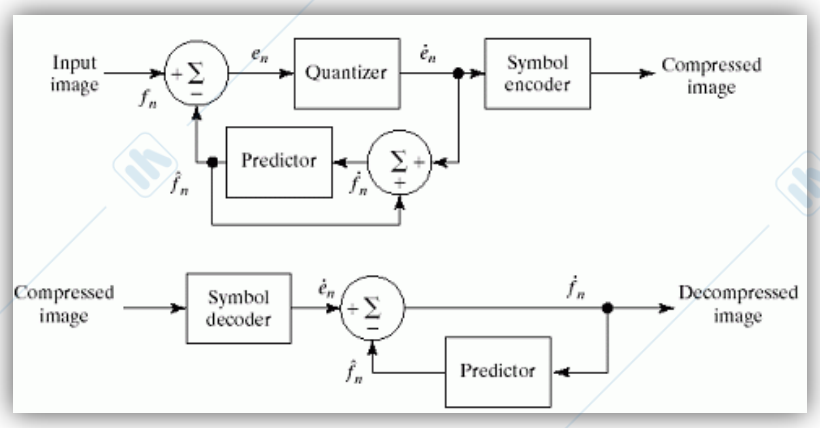
$$f(n) = e(n) + \hat{f}(n)$$

  per decomprimere o ricreare la sequenza di input originale.
  $\hat{f}(n)$ può essere generato tramite metodi globali, locali e adattativi.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

- *Codifica predittiva lossy:*
  L'immagine mostra le componenti di questo sistema



Dall'immagine notiamo che, come prima, abbiamo due componenti principali che svolgono operazioni opposte: l'encoder (codificatore) e il decoder (decodificatore). In questo sistema però oltre al predittore abbiamo anche un *quantizzatore* nella fase di codifica, introdotto per esaminare l'accuratezza nella ricostruzione e la performance della compressione nel contesto dei predittori spaziali. Il quantizzatore, che va a sostituire la funzione di arrotondamento all'intero più vicino presente nell'altro sistema, viene inserito tra l'encoder di simboli e il punto in cui si forma l'errore di predizione; questo viene fatto perché permette di mappare l'errore di predizione in un range di output limitato, denotato da $\dot{e}(n)$, che stabilisce la quantità di compressione e di distorsione del sistema. Per fare in modo che il processo di quantizzazione sia corretto è necessario modificare l'encoder in modo che le predizioni dell'encoder e del decoder siano equivalenti. Questa equivalenza viene ottenuta disponendo il predittore all'interno di un ciclo di feedback continuo, dove il suo input $\dot{f}(n)$ è generato in funzione delle predizioni precedenti e dei corrispondenti errori quantizzati, quindi:

$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n)$$

Dove $\hat{f}(n)$ denota l'output. Questo tipo di configurazione previene l'accumulo di errore nell'output del decoder.

In generale le distorsioni che si creano in questo tipo di codifiche dipendono da un insieme di interazioni tra la quantizzazione e i metodi di predizione impiegati.
Da notare infine che predittore e quantizzatore sono progettati indipendentemente l'uno dall'altro, in quanto il predittore è progettato assumendo che non ci sia alcun errore, mentre il quantizzatore minimizza il proprio errore.

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari