Compressione

Il termine compressione si utilizza per identificare la tecnica di elaborazione dati che tramite specifici algoritmi(eliminano le ridondanze di informazione dai dati) va a ridurre i bit necessari alla rappresentazione digitale di una informazione, questi algoritmi riducono lo spazio di archiviazione richiesto alla memorizzazione e riducono l’occupazione di banda necessaria in una trasmissione dati, i dati possono essere usati per rappresentare la stessa quantità di informazione, i dati irrilevanti o ripetuti vengono detti dati ridondanti.

Ridondanza della codifica

Una codifica è un sistema di simboli utili a rappresentare una certa quantità di informazioni, ad ogni “pezzo” di informazione e evento viene assegnata una codeword, caratterizzata da una lunghezza.

Ridondanza spaziale e temporale

Dal momento che la maggior parte degli array di intensità 2-D sono realizzati spazialmente(schermi), l’informazione è replicata inutilmente nei pixel correlati, in una sequenza video i pixel correlati temporalmente rappresentano un’informazione duplicata.

Classificazione dei metodi di compressione

Basata sul tipo di dati:

* 1. Compressione Generica;
  2. Compressione Audio;
  3. Compressione Immagini;
  4. Compressione Video;

1. Basata sul tipo di compressione:
   1. Compressione reversibile(lossless) (senza perdita di informazione);
   2. Compressione irreversibile(lossy) (con perdita di informazione);

Compressione LOSSLESS

Questa compressione trasforma i dati, in modo da risparmiare memoria, e poi li riesce a ricostruire senza errore e perdita di alcun bit di informazione, questa compressione è necessaria per ridurre lo spazio occupato da documenti, eseguibili e tutto quello che può occupare meno spazio, per raggiungere una compressione senza perdita si utilizza il primo teorema di Shannon che fa uso dell’Entropia delle sequenze, la quantità media di informazione associata alla singola generazione di un simbolo nella sequenza, più è grande l’incertezza della sequenza maggiore è l’entropia, il valore massimo si ha quando i simboli sono equiprobabili.

Teorema di Shannon

“per una sorgente discreta e a memoria zero, il bitrate minimo è pari all’entropia della sorgente.”

Quindi i dati possono essere rappresentati senza perdere informazione usando almeno un numero di bit pari a

N = numero di caratteri;

E = Entropia;

il teorema di Shannon fissa il numero minimo di bit, ma per trovarli occorre usare l’algoritmo di Huffman.

Codifica di Huffman

Questa codifica ha le seguenti proprietà:

1. Lunghezza Variabile associando ai simboli meno frequenti codici lunghi ed ai simboli più frequenti codici corti;
2. Nessuno codice è prefisso di altri;
3. Tende al limite imposta dal teorema di Shannon;

**Algoritmi LOSSLESS**

Codifica Run-length (RLE)

Le immagini che hanno delle ripetizioni di intensità lungo le righe o le colonne possono essere compresse rappresentando queste ripetizioni di intensità come coppie di run-length, dove ciascuna coppia individua l’inizio di una nuova intensità e il numero di pixel consecutivi.

Esempio:

00000111001011101110101111111

Può essere espresso come :

1. 5 volte 0, 3 volte 1, 2 volte 0, 1 volta 1, 1 volta 0, etc;

oppure possiamo segnare solo la lunghezza dei segmenti di simboli eguali:

1. 5, 3, 2, 1, 1, 3, 1, 3, 1, 1, 1, 7;

dopodiché dobbiamo trasformare questa codifica in binario:

Immagine che contiene testo, elettronico

Descrizione generata automaticamente

Codifica Differenziale

Se la sequenze dei valori varia lentamente è possibile ricordarsi solamente il primo valore, e per i successivi ricordarsi le differenze successive.

134, 137, 135, 128, 130, 134, 112

Possiamo rappresentarla come : memorizzo il primo valore : 134

-3, 2, 7, -2, -4, 22

Per le immagini la sequenza delle differenze ha un entropia minore di quella dei singoli valori, per tanto richiede meno bit per essere memorizzata.

**Compressione LOSSY**

Questo tipo di compressioni, si basano sul “se percettivamente non è importante buttalo via” trasformano i dati per risparmiare memoria, in maggior quantità rispetto alla compressione LOSSLESS, ma quando si decomprimono i dati ci sarà un perdita di informazione.

Per una buona compressione LOSSY è meglio rispettare delle regole:

1. Fissare la massima distorsione accettabile, l’algoritmo di compressione deve trovare la rappresentazione con il più basso numero di bit;
2. Cercare il miglior algoritmo che a parità di bit dia la minima distorsione;

**Requantization**

È una compressione che riduce il numero di livelli disponibile in modo da risparmiare bit per pixel:

1. Canale R : usa i 4 bit più significativi;
2. Canale G: usa i 6 bit più significativi;
3. Canale B : usa i 2 bit più significativi

Con questa compressione si risparmia il 50% di bit, inoltre se si utilizza in combinazione la compressione LZW o Huffman queste saranno più efficienti.

**Codifica JPEG/JPG**

Passi fondamentali per una codifica JPEG:

1. Pre-processing:
   1. Trasformazione da RGB a ;
   2. Sottocampionamento della crominanza;
   3. Suddivisione dell’immagine in sotto-immagini per canali;
2. Trasformazione:
   1. Trasformazione DCT(Trasformata Discreta del Coseno);
   2. Quantizzazione;
3. Codifica:
   1. Codifica dei Coefficienti DC;
   2. Ordinamento Zig-Zag per i coefficienti AC;
   3. Codifica dell’entropia

*Trasformazione da RGB* :

Y = Luminanza;

*Sottocampionamento della Crominanza*:

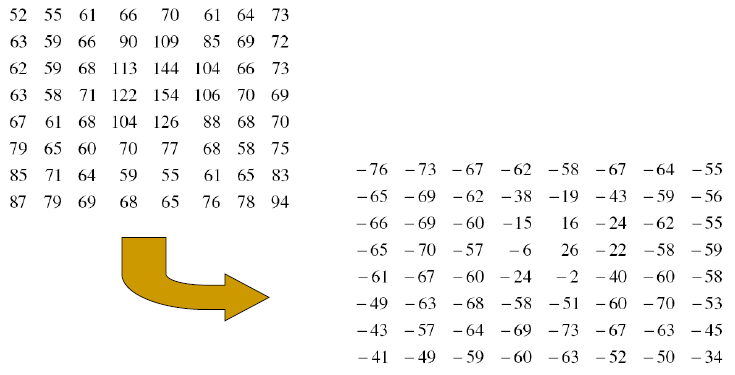
Y = viene presa tutta;

N.B. in questo passo la perdita di informazione sarà irreversibile.

*Partizione dell’immagine*:

per usufruire al meglio della ridondanza, JPEG divide l’immagine in quadrotti 8x8 di 64pixel non sovrapposti, ogni quadrotto avrà un elaborazione differente, per questo si ha il problema della quadrettatura visibile negli ingrandimenti o nelle stampe, maggiore sarà la compressione più sarà evidente la quadrettatura.

Prima di passare alla trasformazione DCT ai 64pixel di ogni quadrotto viene sottratta una quantità pari a dove è il numero massimo di livelli di grigio dell’immagine, in questo modo si fa uno shift dei livelli di grigio, portando il grigio medio(128) = 0.



*Trasformazione DCT*:

è una trasformata di Fourier, dove viene decorellato al massimo i dati permettendo un maggior rapporto di compressione in fase di codifica, un’immagine di 8x8 pixel può essere vista come un vettore di dimensione 64, per tanto ogni immagine è la somma pesata di 64 immagini impulsive, immagini tutte nere, tranne in un pixel che avrà valore 1, dove i pesi della somma sono dati dal livello di luminosità del pixel, queste immagini impulsive rappresentano la base impulsiva che non è l’unica base.

Con:

Altrimenti

Dopo l’applicazione della DCT abbiamo questo:

Immagine che contiene freccia

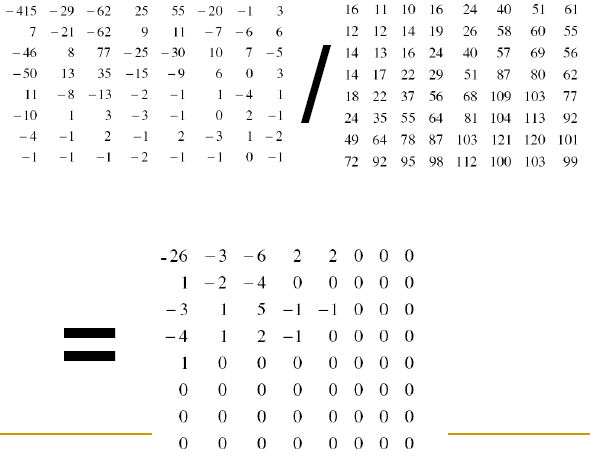
Descrizione generata automaticamente

Basta moltiplicare i singoli coefficienti alle basi della DCT e otteniamo il blocco precedente, il coefficiente DC è solo il primo in alto a sinistra, i restanti 63 sono coefficienti AC, ogni immagina 8x8 si ottiene moltiplicando ciascuna immagine per un coefficienti e sommando il tutto, questi sono i coefficienti DCT, mentre il primo in alto a sinistra è un valore proporzionale al valor medio della luminanza ed è detto coefficiente DC.

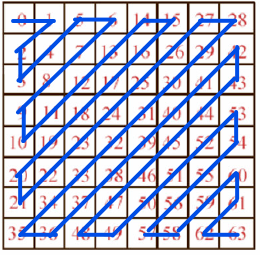
*Quantizzazione* :

non è conveniente usare un unico fattore di quantizzazione per tutti i 64 coefficienti DCT, per tanto per il coefficiente F(i, j) si utilizza il fattore di quantizzazione Q(i, j), questo fattore viene scelto dall’utente, che dovrà fornire la tabella assieme ai dati compressi, o preso da uno standard, la tabella dove ci sono tutti i fattori di quantizzazione si chiama tabella di quantizzazione, nella tabella l’utente dovrà fornire anche un Quality Factor che va da 1 a 100, maggiore sarà il QF, minore la quantizzazione e quindi minore la perdita di informazione.

Effetto della quantizzazione:



Tutti i coefficienti DCT vengono riordinati in un array di 64 unità seguendo l’ordinamento a serpentina per poi essere codificati in un altro stream.



Codifica coefficienti DC e AC:

i coefficienti della posizione (1, 1, alto a sinistra) vengono codificati tramite codifica differenziale, mentre i coefficienti AC tramite la codifica RLE