Codifica di Immagini Fisse

Fondamenti di Telecomunicazioni

Anno Accademico 2009/2010

Contenuti

- Caratteristiche delle immagini
- Codifica JPEG

Perché comprimere…

Si supponga di voler trasmettere 30 fotografie in bianco e nero, digitalizzate e quantizzate con 8 bit/pixel di dimensione CIF (352x288) nel caso si utilizzi una codifica naturale senza compressione (es. BMP o TIFF).

Ogni fotografia risulta composta da 352x288=101376 pixel. Ogni foto viene poi digitalizzata e quantizzata con 8 bit per pixel, ottenendo un totale di bit per ogni foto pari a 8*101376=811008 bit.

Per 30 foto il conto totale in bit risulta pari a 811008*30=24330240 bit (ovvero più di 3MB di informazione).

Supponendo di utilizzare un canale GSM a 9.6 kbit/s, il tempo necessario alla trasmissione risulta pari a circa 42.2 minuti, con un modem a 64 kbit/s si riduce a 6.3 minuti.

Standard JPEG

JPEG è uno standard di compressione per immagini, definito nel 1991 da un gruppo di esperti (*Joint Photographic Experts Group*). JPEG è stato progettato per comprimere sia immagini a colori che immagini a gradazioni di grigio.

Si distinguono due tipi di compressione:

• compressione ***lossy***: scarta delle informazioni delle immagini poco visibile all'occhio umano e comprime le rimanenti informazioni. L'immagine compressa è una buona *approssimazione* dell'immagine originale;

• compressione ***lossless***: comprime tutte le informazioni di un'immagine, in modo tale che l'immagine prima compressa e poi decompressa, sia *identica* bit a bit a quella originale, senza alcuna perdita di informazioni e quindi di qualità.

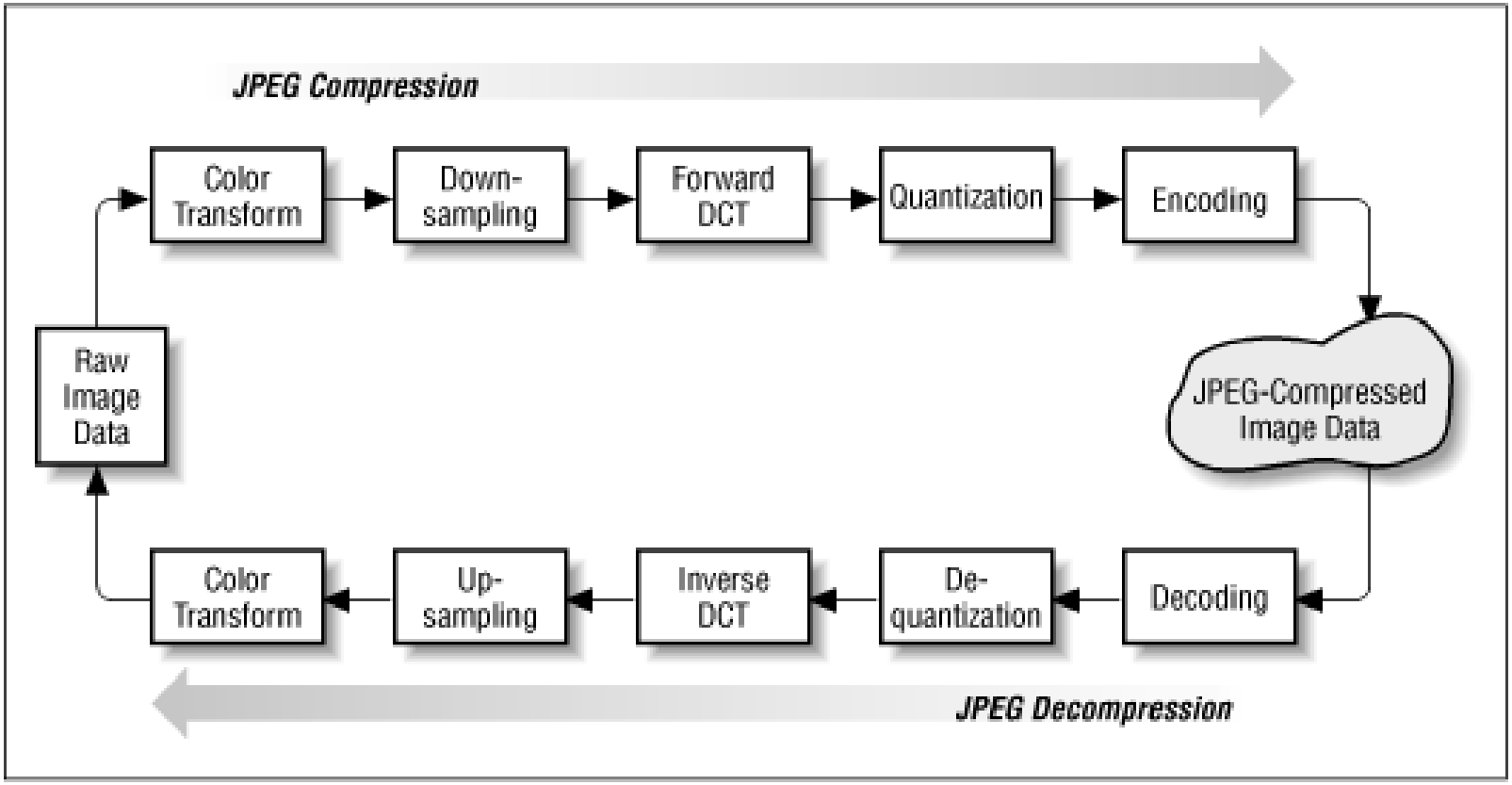
JPEG utilizza un algoritmo di compressione di tipo lossy.

Standard JPEG

Lo standard JPEG è stato sviluppato in modo da soddisfare le seguenti indicazioni:

- deve essere in grado di ottenere elevati rapporti di compressione;
- deve poter ricostruire l'immagine in modo totalmente reversibile, oppure in modo irreversibile, ma con una elevata fedeltà;
- deve poter essere parametrizzato in modo da lasciare all'applicazione la possibilità di definire al meglio il rapporto di compressione e la fedeltà dell'immagine ricostruita;
- l'algoritmo non deve dipendere né dalla complessità della scena rappresentata né, possibilmente, dalle caratteristiche intrinseche dell'immagine, quali dimensione dello spazio dei colori, rapporto bit/pixel, etc.;
- la sua complessità computazionale deve permettere sia un'implementazione software, sia un'implementazione hardware, che rimanga contenuta in termini di costo.

Algoritmo JPEG



Lettura file sorgente

I dati letti dal file immagine sorgente, sono organizzati come tre matrici, nello Spazio Colore **RGB**, di generiche dimensioni M x N.(es. CIF = 352 x 288).

Ogni matrice sarà successivamente divisa in blocchi di dimensione 8x8, ovvero ogni blocco conterrà 64 pixel dell'immagine originale. Quindi se M o N non risultano multipli di 8, sono aggiunte delle copie dell'ultima riga o dell'ultima colonna alla matrice dell'immagine originale, sino a che la dimensione della nuova matrice non diventi un multiplo di 8.

Nella decodifica si terrà conto di questi pixel di riempimento aggiunti, che verranno quindi eliminati

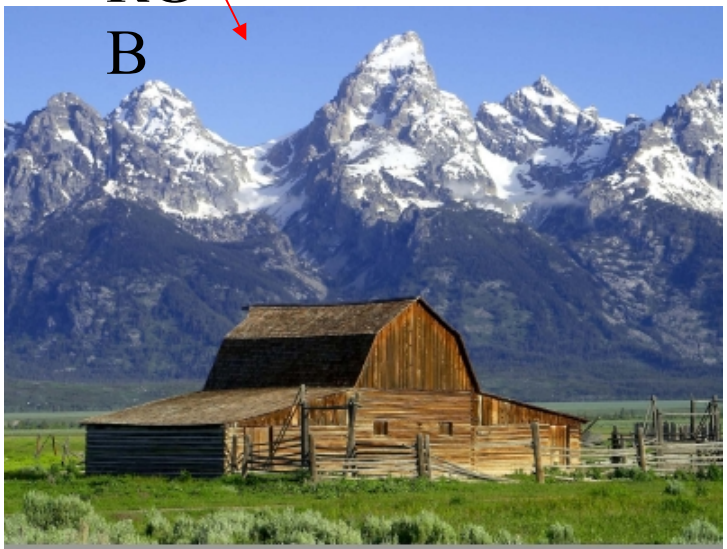
Trasformazione Spazio Colore

L'immagine viene trasformata dallo Spazio Colore RGB a quello Yuv (o YCbCr).

Dopo aver calcolato Y, u e v è possibile ridurre l'informazione riguardante le componenti di crominanza tramite sottocampionamento.

La componente di luminanza non deve invece essere decimata, altrimenti si avrebbe una perdita elevata di informazione a livello di qualità percettiva. Ciò deriva dal fatto che l'occhio umano è meno sensibile alle variazioni cromatiche e non a quelle di luminosità.

RGB -> Yuv



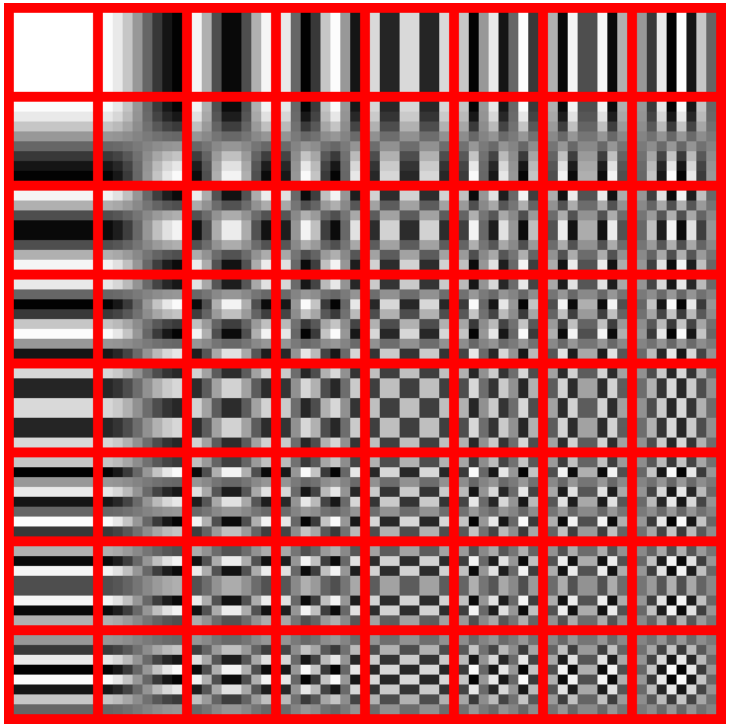
Trasformata (DCT)

Ogni matrice è divisa in blocchi di dimensione 8x8. Ad ogni blocco è applicata la trasformata DCT bidimensionale, che restituisce ancora un blocco di dimensioni 8x8.

Con l'applicazione della DCT si passa da una rappresentazione nel dominio spaziale a una nel dominio delle frequenze.

Una caratteristica saliente del blocco restituito dalla DCT è che nell'angolo in alto a sinistra si trovano i dati relativi alle basse frequenze ed energeticamente più rilevanti, mentre nell'angolo in basso a destra i coefficienti rappresentanti le alte frequente ed energeticamente meno importanti (si ricordi che l'occhio si comporta come un filtro passa-basso).

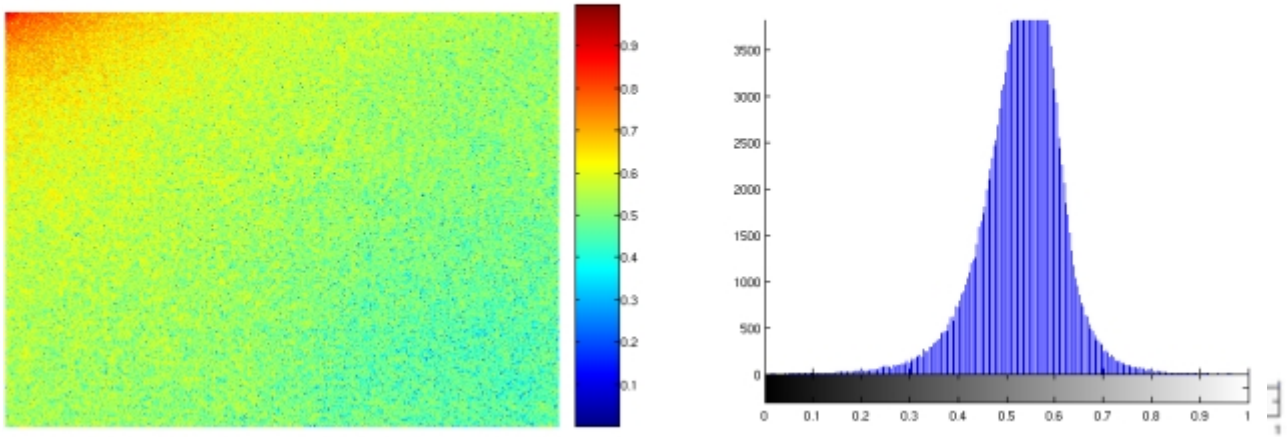
Frequenze (DCT)



L'utilità della DCT è duplice: da un lato si sfruttano le note proprietà di compattazione dell'energia, dall'altro si riescono a trattare i coefficienti DCT in modo indipendente senza perdere di efficienza nella compressione. La DCT è invertibile senza perdita d'informazione, a parte eventuali errori di arrotondamento.

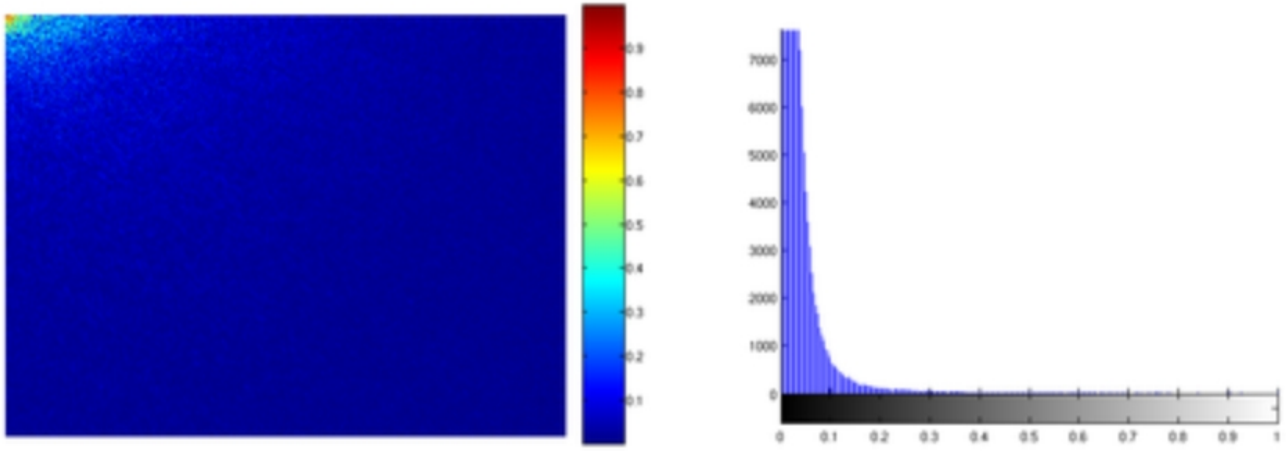
Perché DCT e non DFT?

DFT



Trasformata (DCT)

DCT



Quantizzazione

Escludendo la trasformazione nel formato Yuv, questa è la prima operazione lossy effettuata dall'algoritmo in esame.

Dopo aver ricevuto la matrice (8x8) delle frequenze dalla DCT, l'algoritmo applica delle divisioni e arrotondamenti (queste due operazioni vengono appunto chiamate quantizzazione), in modo da rendere più valori possibili uguali a 0 (che vengono compressi in maniera estremamente semplice tramite RLE o codifica Huffman).

In questa fase ogni elemento di ciascun blocco viene diviso per un coefficiente presente nella tabella di quantizzazione sempre di dimensione 8x8. L'effetto dell'operazione di divisione e' quello di arrotondare i valori prodotti dalla DCT, portando a zero quelli prossimi allo zero, smorzare le alte frequenze e arrotondare con maggiore precisione le frequenze basse, utilizzando una *Tabella di quantizzazione* per la luminanza e una opportuna per le due di crominanza.

Quantizzazione

in Codifica: $$X_q(k_1,k_2) = NearestInteger\left(\frac{X(k_1,k_2)}{Q_i(k_1,k_2)}\right)$$

in Decodifica: $$X(k_1,k_2) \cong Q_i(k_1,k_2)\cdot X_q(k_1,k_2)$$

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

Matrice di Quantizzazione $Q_L(k_1,k_2)$ per la luminanza (Y)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Matrice di Quantizzazione $Q_C(k_1,k_2)$ per le crominanze (U e V)

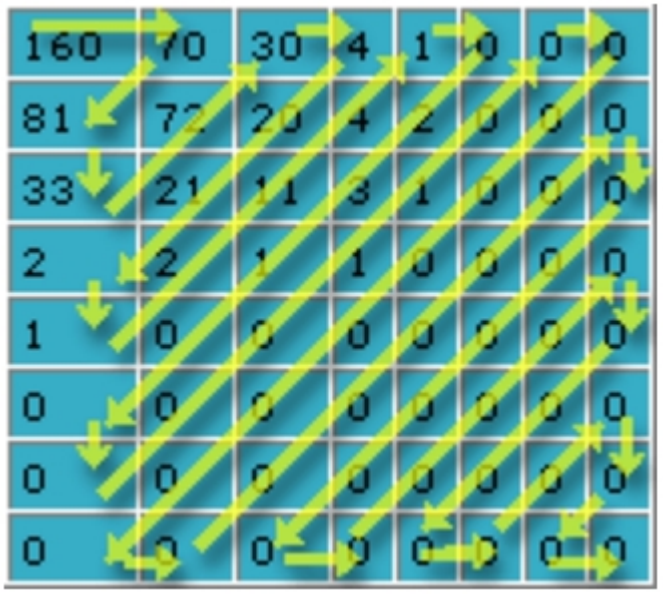
Errore di Quantizzazione

Le operazioni di divisione e di arrotondamento del risultato per alti coefficienti nella Tabella di quantizzazione definiscono una tabella DCT quantizzata con valori notevolmente meno accurati nella parte bassa a destra. E' evidente che l'errore aumenta all'aumentare del coefficiente di quantizzazione a causa degli arrotondamenti.

Le Tabelle di Quantizzazione non sono standard ma differiscono dal produttore di software, i gradi di compressione poi dipendono da essa (di solito vengono usate quelle definite dagli standard H.263 o MPEG).

Qualitativamente, se si ricostruisse l'immagine considerando le basse frequenze, essa risulterebbe priva di dettagli, con zone d'ombra e di luminosità notevolmente sfuocate. Diversamente considerando quelle alte si evidenzierebbero i contorni e i dettagli, mentre le parti sfumate verrebbero eliminate (quindi con zone tutte di una stessa tonalità). Queste matrici sono di solito mutuate da studi psicovisuali (dobbiamo ringraziare i limiti dei nostri occhi per questo!).

Riordino coefficienti



La fase finale del processo JPEG è la codifica dell'immagine quantizzata. La codifica JPEG consiste in tre differenti passi per comprimere l'immagine.

Se i valori del blocco vengono visti come un'onda acustica (con la posizione nel blocco in ascissa e il valore in ordinata), l'elemento del blocco in posizione (0,0) detto coefficiente DC rappresenta la componente continua dell'onda (da qui il nome DC, *Direct Current* o corrente continua); gli altri valori del blocco (coefficienti AC) sono relativi ad esso. Ai fini del calcolo, il coefficiente DC è il valore massimo in ampiezza presente nel blocco.

Codifica RLE

Run Length Encode (RLE) è una semplice tecnica di compressione applicata ai componenti AC: il vettore 1x64 risultante dalla lettura a zig-zag contiene diversi valori nulli in sequenza, per questo si rappresenta il vettore tramite coppie (skip, value), dove *skip* è il numero di valori uguali a zero e *value* è il successivo valore diverso da zero. La coppia (0,0) viene considerata come segnale di fine sequenza.

Sul valore DC di ciascun blocco viene invece applicata una tecnica detta DPCM. Questo sistema modifica il coefficiente DC dal valore assoluto ad un valore relativo al coefficiente DC del blocco precedente. Ad esempio, se i coefficienti DC di due blocchi adiacenti sono 227 e 205, il secondo diventerà -22. I blocchi adiacenti solitamente hanno un elevato grado di correlazione, così la codifica del coefficiente DC come differenza dal coefficiente precedente tipicamente produce un piccolissimo numero; gli altri elementi, di valori di per sé già contenuti, non vengono modificati.

Codifica Huffman

L'ultima codifica *entropica* applicata ai dati è la classica codifica a lunghezza di codice variabile.

In pratica i dati vengono suddivisi in 'parole' (stringhe di bit), viene analizzata la frequenza statistica di ciascuna parola e ognuna viene ricodificata con un codice a lunghezza variabile in funzione della frequenza di apparizione. Un codice corto per le parole che appaiono frequentemente e via via codici più lunghi per quelle meno frequenti.

Complessivamente il numero di bit necessari per rappresentare i dati si riduce consistentemente.

Esempi

Immagine originale in RGB, occupazione su disco in TIFF = 579KB

Immagine sulla componente Y

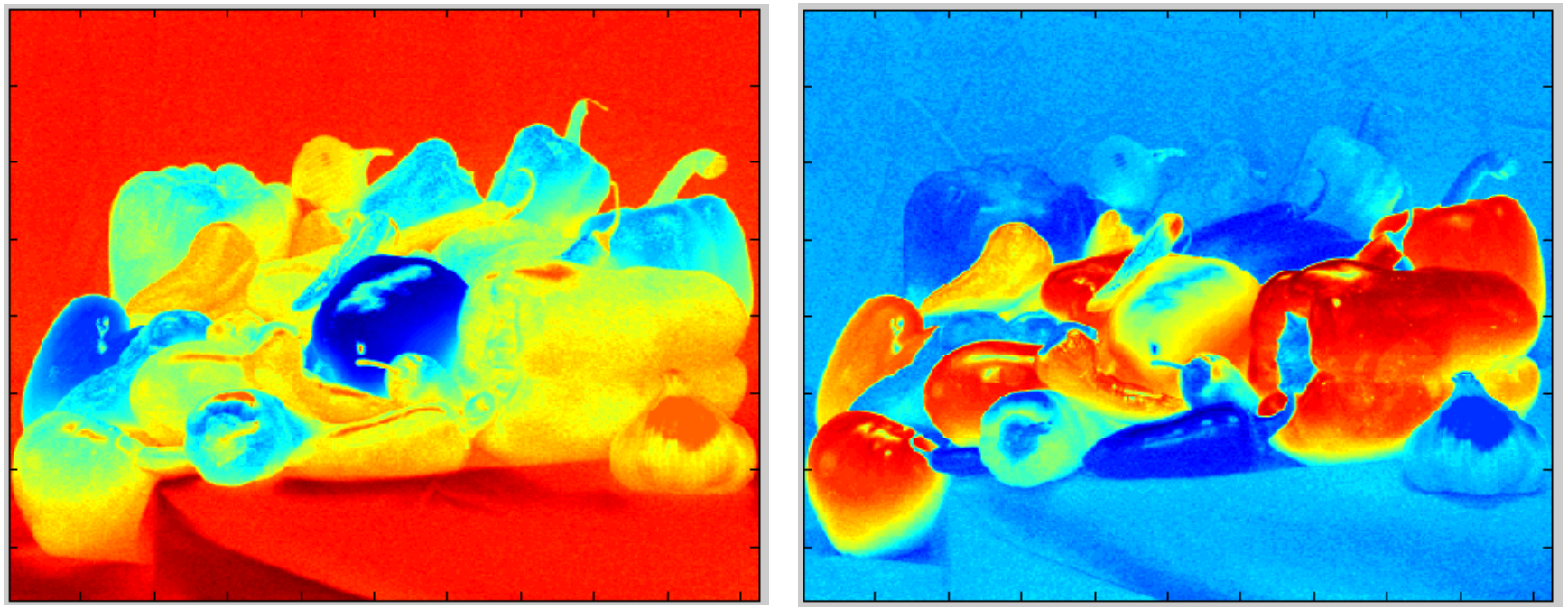


Immagine sulle componenti U, V

Immagine originale in RGB ricostruita senza la componente V

Esempi di compressione

Immagine compressa JPG con qualità 5% occupazione su disco = 4.89KB

Esempi di compressione

Immagine compressa JPG con qualità 15% occupazione su disco = 7.87KB

Esempi di compressione

Immagine compressa JPG con qualità 90% occupazione su disco = 40.3KB

Esempi di compressione

Immagine compressa JPG con qualità 100% occupazione su disco = 140KB