

Homework Multimedia

Lorenzo Mancini 2007951

1 Maggio 2023

Sommario

Report Homework 1: Una semplice tecnica di codifica d'immagini basata sulla DCT

1 Introduzione

L'obiettivo di questo Homework è quello di produrre un codice che effettui una semplice tecnica di codifica basata sulla DCT.

I passi principali del funzionamento del programma sono così riassunti:

1. Compressione dell'immagine:
 - trasformazione dell'immagine dal campo colori RGB al quello YCBCR
 - partizionare ogni componente YCBCR in blocchi di dimensione $N \times N$
 - applicazione della codifica DCT su ciascuna componente YCBCR
2. Immagine compressa:
 - sogliaggio delle componenti YCBCR
 - applicazione della codifica IDCT su ciascuna componente YCBCR
 - calcolo MSE , MSE_P e $PSNR$
3. il tutto ripetuto per ogni valore del parametro $N = \{8, 16, 64\}$;
 - variazione dei valori di R da 10 a 100 a passi di 10
 - visualizzazione della curva del PSNR in funzione di R
4. confronto delle tre curve ottenute

2 Scelte di progetto del codice

Come linguaggio di programmazione è stato scelto MatLab, perché linguaggio utilizzato a lezione e a cui il professore ha fatto riferimento durante le sue spiegazioni teoriche e sperimentali. Durante lo sviluppo del codice sono stati installati i seguenti Add-On:

- Image Processing Toolbox
- Simulink

necessari per poter eseguire il programma correttamente.

Non sono state utilizzate ulteriori librerie esterne all'ambiente di sviluppo descritto.

Accorgimento tecnico al codice del programma

Per poter visualizzare al meglio il grafico di ogni curva e infine il grafico comparativo di ciascuna di esse, sono stati creati due array e una matrice, accettati come argomenti dalla funzione `plot(x,y)`.

Nei casi del grafico di curva singola, x coincide con un array di valori del parametro R , chiamato `x_r`

e scala dell'asse x del grafico, mentre y coincide con l'array chiamato y_psnr contenente i valori del PSNR, scala dell'asse y.

Nell'ultimo caso, il parametro x rimane l'array di valori di $R \times r$, invece il secondo argomento y diventerà una matrice 10×3 contenente tutti i valori del PSNR calcolati per ogni curva parametrizzata da N.

In questo modo vengono visualizzate in un unico grafico tutte le tre curve studiate, per poter eseguire un confronto adeguato.

3 Istruzioni di esecuzione del codice

Lo sviluppo del progetto è stato effettuato in ambiente MatLab, per cui per poter eseguire il programma bisogna prima importare il progetto nella directory del workspace dell'elaboratore sulla quale si intende operare. Successivamente si procede all'apertura del file e nella sezione "EDITOR" avviare l'esecuzione tramite il tasto "Run".

4 Esempi di codifica

Per un'immagine fissata, tracciare le curve tasso distorsione per $N=8$, $N=16$ e $N=64$.
Immagine di riferimento utilizzata durante lo sviluppo del progetto:



Figura 1: Immagine di partenza in RGB

4.1 Curva parametrizzata per $N=8$

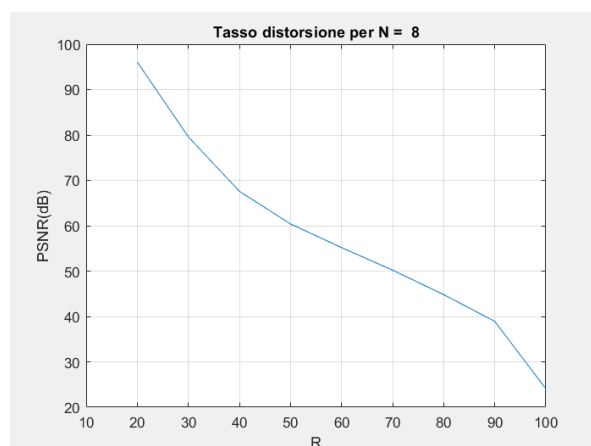


Figura 2: Grafico $N=8$

4.2 Curva parametrizzata per $N=16$

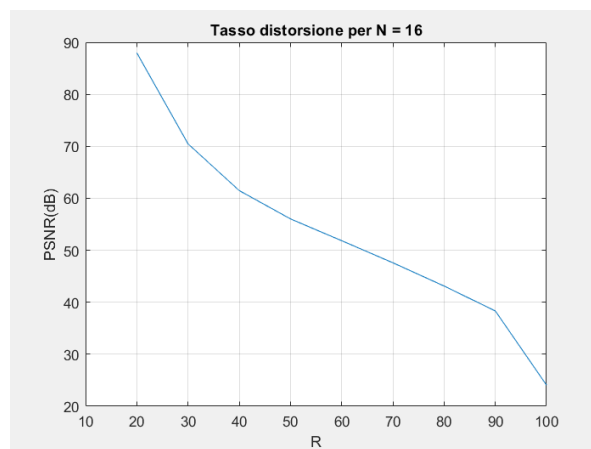


Figura 3: Grafico $N=16$

4.3 Curva parametrizzata per $N=64$

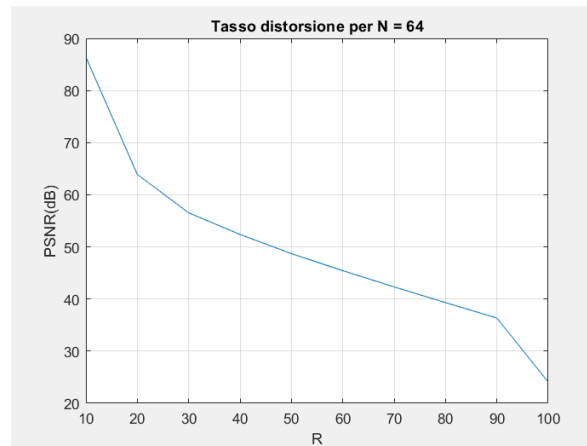


Figura 4: Grafico $N=64$

5 Commento dei risultati

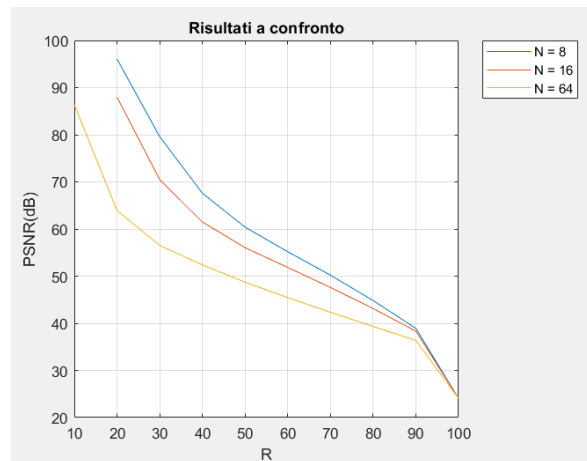


Figura 5: Confronto delle curve

Come si può osservare, le tre curve si differenziano maggiormente per bassi valori di R , ad indicare che la qualità della compressione è molto più alta a valori di R bassi e dipendente dalla grandezza dei blocchi scelta, ovvero dipendente da N .

Al crescere di R questa differenza tra le tre curve si assopisce, diventando trascurabile e anche indipendente da N .

Tuttavia si può notare che maggiori sono le dimensioni dei blocchi e più aumenta la probabilità che siano presenti brusche discontinuità che, nel dominio trasformato della codifica DCT, vengono disperse su un elevato numero di coefficienti di valore prossimo a zero.

Nel grafico soprastante, la curva gialla parametrizzata per $N=64$ conferma ciò che è stato appena dichiarato.

È necessario quindi operare su blocchi piccoli per garantire comunque un'elevata efficienza di codifica. La scelta ottimale da eseguire per le condizioni del problema porta a partizionare l'immagine in blocchi di dimensione $N=8$ per poter usufruire di una qualità più alta.