

---

---

# COMPRESSEIONE DI IMMAGINI TRAMITE LA DCT

METODI DEL CALCOLO SCIENTIFICO - PROGETTO II



Andrea Carubelli 803192  
Alessio Abondio 808752

# INTRODUZIONE

- Con l'avanzare della tecnologia, i dati sono sempre qualitativamente migliori ma anche «*più pesanti*»
- Questo ha portato ad un'inevitabile attenzione riguardo l'efficienza degli algoritmi di compressione
- Questi algoritmi sono, nella maggior parte dei casi, basati su trasformate che traslano il dominio da quello spaziale a quello delle frequenze in modo da poter operare direttamente sulle frequenze che compongono il segnale

# PARTE I - DCT 2

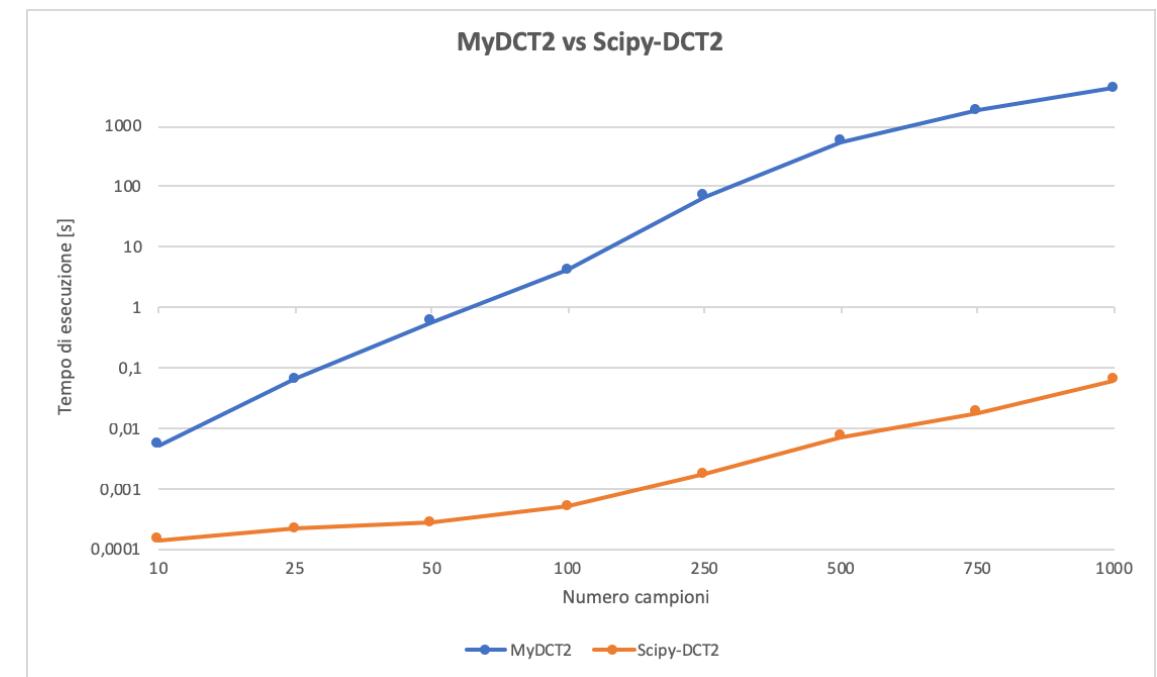
- Una delle trasformate più famose è la DCT (*Discrete Cosine Transform*) (DCT2 nel caso di input bidimensionali)
- La DCT è una trasformazione lineare che ci fa passare dai coefficienti “spaziali” a quelli in frequenza
- Lo scopo dell’applicazione di questa trasformata è quello di comprendere il contributo di ogni frequenza al fine di eliminare quelle con un contributo minimo, permettendo di ridurre la quantità di informazione necessaria a memorizzare il segnale, minimizzando la perdita di qualità

# OBIETTIVI

- L'obiettivo di questa parte è quello di implementare al DCT2 e di confrontarne le performance con quelle di una versione già presente in una specifica libreria open source. Abbiamo deciso di:
  - Implementare in Python l'algoritmo teorico della DCT2
  - Confrontare la nostra implementazione con la DCT della libreria in Python «Scipy.fftpack»
- Lo scopo principale è di valutare quanto la versione fornita dalla libreria migliori effettivamente il tempo di esecuzione rispetto alla versione originale

# CONFRONTO

CAMPIONI	MyDCT2[s]	Scipy-DCT2[s]
10	0,005466938	0,000145912
25	0,065145731	0,000221968
50	0,587640047	0,000275135
100	4,238348	0,000505209
250	69,7790442	0,001790047
500	542,493272	0,007333279
750	1.813,10	0,018507004
1000	4.165,84	0,0623399

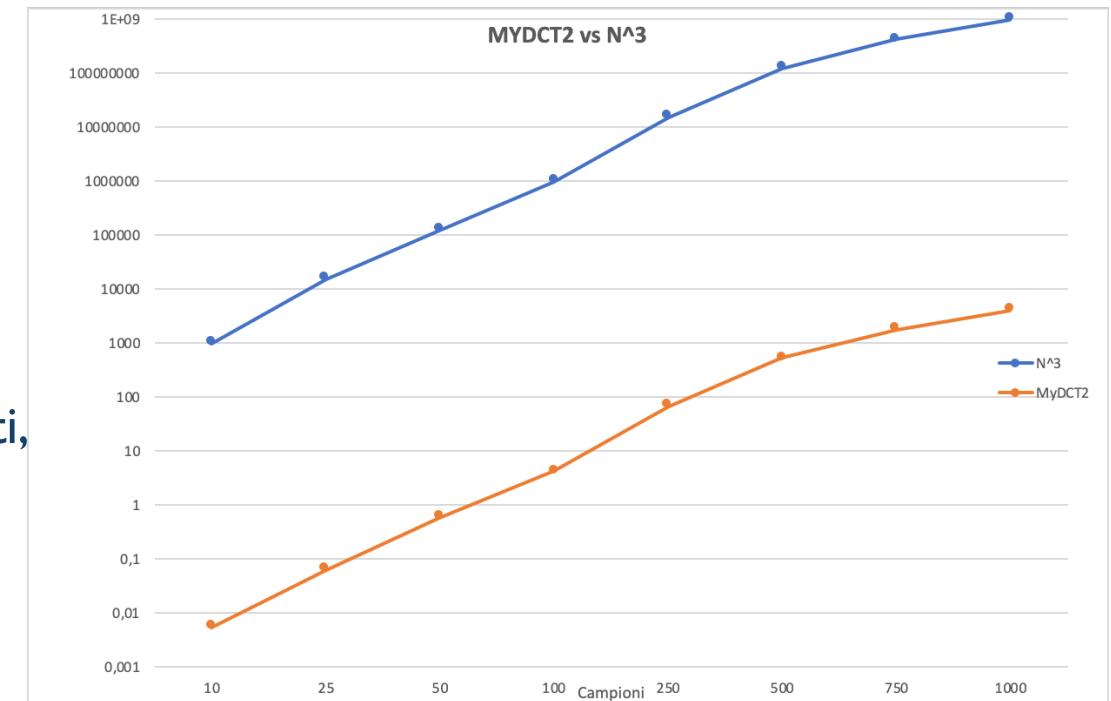


# RISULTATI - I

- MyDCT2 è stata implementata applicando la DCT monodimensionale, che richiede tempo  $O(n^2)$ , su righe e colonne ( $n$ ) dell'input per migliorare l'efficienza algoritmica da  $O(n^4)$  a  $O(n^3)$
- Lo stesso è stato fatto per la DCT2 di Scipy, in quanto la libreria conteneva solo la versione monodimensionale
- La principale differenza tra le due implementazioni risiede nel fatto che Scipy utilizza FFT (*Fast Fourier Transform*) per eseguire DCT, abbassando così il costo teorico da  $O(n^3)$  ad un costo approssimato  $O(n^2\log(n))$

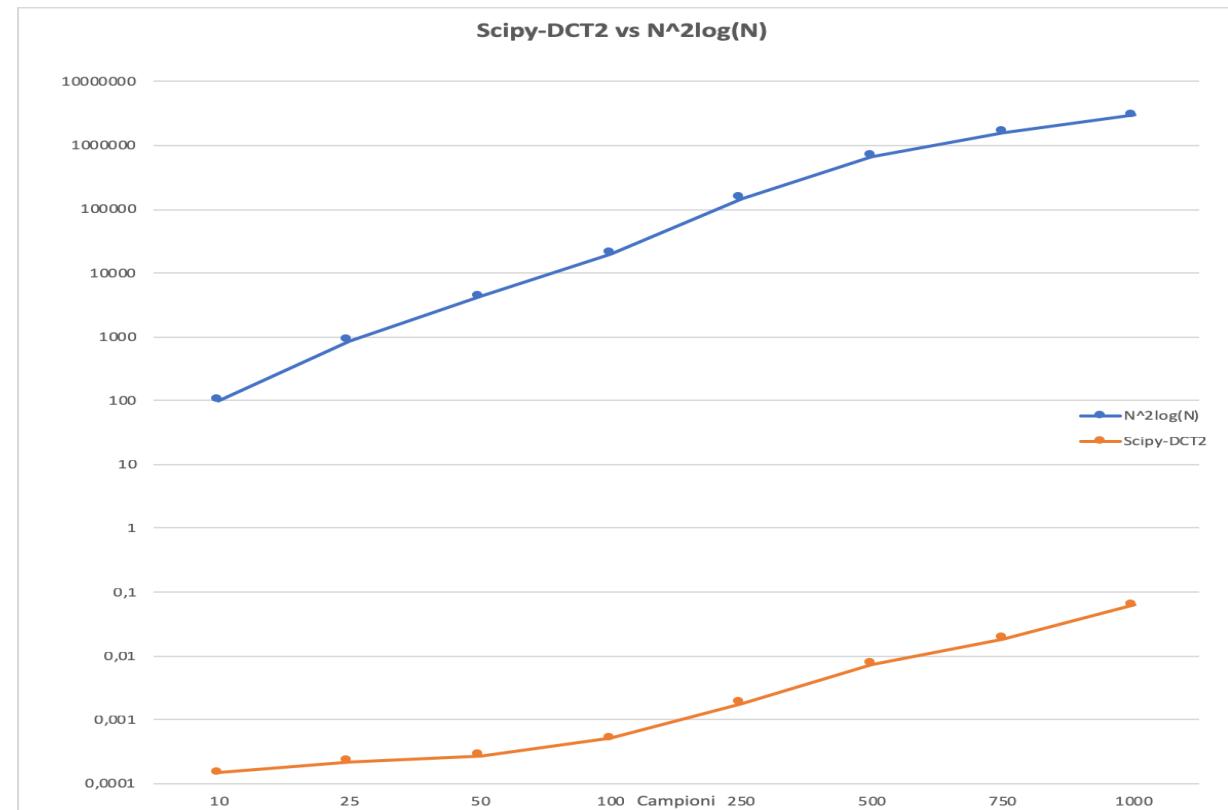
# RISULTATI - II

- Come possiamo notare dal grafico, i tempi raccolti nell'esecuzione della funzione *MyDCT2* seguono, lo stesso andamento di una curva  $y = n^3$ , coerentemente al suo andamento teorico  $O(n^3)$
- Calcolando la *cross-correlation* tra le due serie, infatti, essa risulta uguale a 0.9999152



# RISULTATI - III

- Come possiamo notare dal grafico, i tempi raccolti dall'esecuzione della funzione *Scipy-DCT2*, hanno lo stesso trend della funzione  $n^2\log(n)$
- Infatti, calcolando la *cross-correlation*, si ottiene un valore di 0.9705.



---

---

---

## PARTE II – MANIPOLAZIONI DELLE FREQUENZE



# IMMAGINI DIGITALI E FREQUENZE

- Un'immagine digitale è una matrice di pixel. Questa matrice deriva dalla digitalizzazione di un segnale analogico (l'insieme delle frequenze che compongono l'immagine)
- Attraverso l'utilizzo di trasformate (DCT) è possibile «effettuare il processo inverso», ovvero ottenere le frequenze (digitalizzate) che compongono l'immagine
- La modifica di queste frequenze ha ripercussioni sull'immagine che saranno visibili una volta effettuata la IDCT «*inverse DCT*»

# OBIETTIVI

- Lo scopo principale di questa fase è quello di progettare e implementare un software che permetta di manipolare le frequenze di un'immagine a piacere in toni di grigio, in formato .bmp
- Le modifiche possibili avvengono attraverso la scelta di due parametri:
  - un intero  $d$  compreso tra 0 e  $(2F - 2)$  che sarà la soglia di taglio delle frequenze.
    - Maggiore è il valore  $d$ , minore è il numero di frequenze selezionate
  - un intero  $F$  che sarà l'ampiezza della finestra in cui si effettuerà la DCT2
    - Maggiore è il valore  $F$ , più grande sarà la finestra.  $F$  sarà il parametro che deciderà in quante finestre quadrate  $F \times F$  sarà suddivisa l'immagine. Più piccolo sarà il valore di  $F$ , maggiore sarà il costo computazionale
    - Ad ogni blocco verrà applicata la DCT2

# PROCEDIMENTO

- L'utente potrà scegliere un'immagine, con estensione `.bmp`, dal proprio *file system*
- Se l'immagine non è quadrata, verrà ritagliata in forma quadrata, in base alla dimensione più piccola fra larghezza ed altezza
- L'utente a questo punto potrà scegliere a piacere i valori  $d$  e  $F$  che saranno rispettivamente il taglio delle frequenze  $C_{kl}$  con  $k + l \geq d$  e l'ampiezza della finestra
- L'immagine verrà suddivisa in blocchi  $f$  di pixel di dimensioni  $F \times F$
- Verrà applicata la DCT2 su ogni blocco  $f$  di dimensioni  $F \times F$
- Si eliminano le frequenze  $C_{kl}$  con  $k + l \geq d$
- Verrà applicata la IDCT2 (DCT2 inversa) su ogni blocco  $f$
- Si ricompone l'immagine e si mostra a schermo l'immagine originale affiancata dall'immagine elaborata

# INTERFACCIA GRAFICA

Immagine originale

Parametro F

Parametro d

Immagine elaborata

Tasto per la conversione

Original image

Elaborated image

GUI Interface - Frequencies Manipulation

d: 10

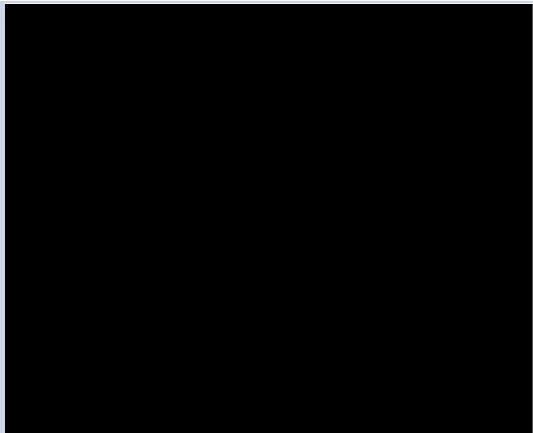
F: 100

Select input image

Convert

The screenshot displays a graphical user interface titled "GUI Interface - Frequencies Manipulation". At the top, there are three colored window control buttons (red, yellow, green). Below the title, there are two input fields: "d:" containing the value "10" and "F:" containing the value "100". A button labeled "Select input image" is positioned between these fields. On the left, a black and white image of a cathedral's interior is labeled "Original image". On the right, a processed version of the same image is labeled "Elaborated image". A large green rectangular callout box covers the central area of the interface. Inside this box, the text "Parametro d" is placed above the "d:" field, "Parametro F" is placed above the "F:" field, and "Tasto per la conversione" is placed below the "Convert" button. To the left of the "Original image", a green callout box contains the text "Immagine originale". To the right of the "Elaborated image", another green callout box contains the text "Immagine elaborata".

$d = 0$



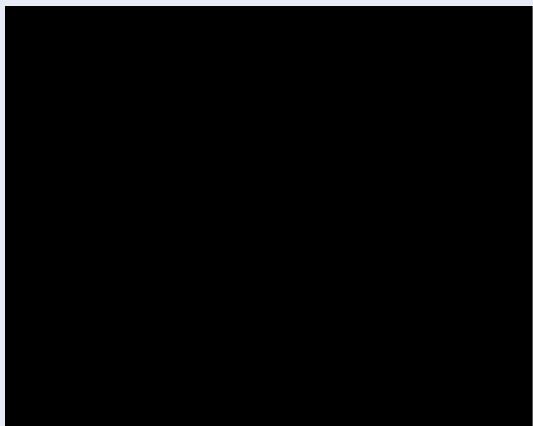
$d = 10$



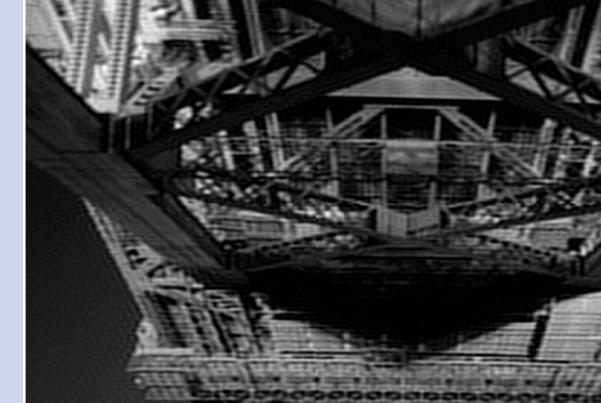
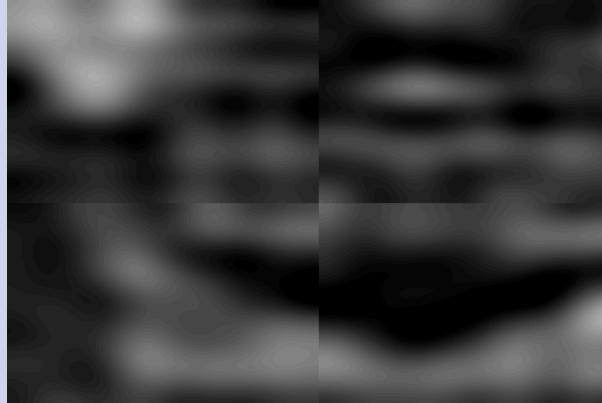
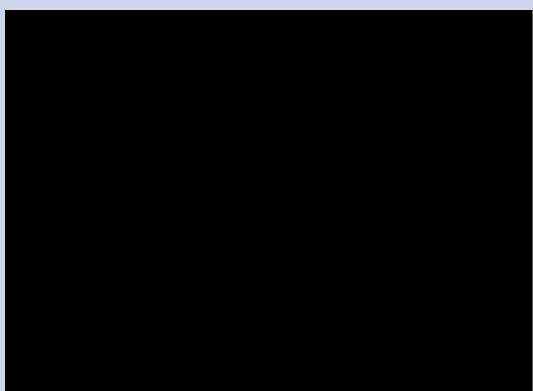
$D = 100$



$F = 100$



$F = 500$



$F = 1000$

---

---

---

**GRAZIE PER L'ATTENZIONE**

