Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Progetto 2

Progetto del corso di Metodi del Calcolo Scientifico

Davide Costantini 856114

David Gargaro 845738

Anno Accademico 2023-2024

Sommario

Progetto 2

Lo scopo di questo progetto è di implementare la trasformata del coseno (DCT) e di studiarne gli effetti, confrontandola con l'implementazione offerta da una libreria open source.

Il progetto è diviso in due parti:

- Parte 1: l'obiettivo è quello di implementare la DCT bidimensionale (DCT2) confrontare i tempi di esecuzione con quelli ottenuti utilizzando la implementata da una libreria open source scelto. L'analisi prevede l'uso di array quadrati di dimensioni N×N con N crescente.
- Parte 2: l'obiettivo è quello di realizzare un programma con interfaccia grafica che permetta di applicare la DCT2 ad una immagine e dei parametri (ampiezza dei blocchi e soglia di taglio delle frequenze) specificati dall'utente, e mostrarne il risultato.

La Discrete Cosine Transform è una variante discreta della trasformata di Fourier, che permette di rappresentare una generica funzione $f: R \rightarrow R$ di di periodo p come una combinazione di funzioni seno e coseno. Nella sua versione 1D si basa sul calcolo dei coefficienti con la seguente equazione:

$$\alpha_k = \frac{\sum_{i=1}^{N} \cos\left(\pi k \frac{2i+1}{2N}\right) v_i}{w^k * w^k}$$

Per gli scopi della prima parte del progetto, la versione bidimensionale è stata implementata applicando la formula 1D prima sulle righe e poi sulle colonne della matrice.

Il codice del progetto è disponibile in questa repository:
https://github.com/Deivmercer/Metodi-del-Calcolo-Scientifico-2

Parte 1

Per la realizzazione della prima parte abbiamo suddiviso il problema in sottoproblemi più piccoli:

- procurarsi array quadrati N×N con N crescente
- implementazione della nostra DCT2
    - gestione della matrice
    - calcolo dei coefficienti ak
- risultati
    - DCT2
    - DCT2 della libreria OpenCV

Procurarsi array quadrati N×N con N crescente

Abbiamo generato array quadrati di dimensioni N×N con N variabile. I valori di N scelti sono stati da 100 a 2000 passando di 100 in 100.

```
for (int size = 100; size <= 2000; size += 100)

    {/Implementazione DCT2/}
```

Implementazione della nostra DCT2

Prima di tutto abbiamo effettuato la gestione della matrice per applicare correttamente la DCT2. Per calcolarla faremo la DCT rispetto alle righe e poi rispetto le colonne per cui è stato necessario scorrere sulla matrice per righe, trasporre la matrice e applicare la funzione nuovamente.

Così facendo verrà eseguita due volte la DCT sia per righe che per colonne.

Fatta la gestione della matrice siamo passati all'effettivo calcolo dei coefficienti applicando la formula della DCT per ogni vettore della nostra matrice.

Risultati

Al termine dell'implementazione abbiamo testato entrambe le DCT2, sia quella implementata da noi che quella di OpenCV.

Per verificare la correttezza della procedura abbiamo usato il caso test indicato in Fig.1.

231	32	233	161	24	71	140	245
247	40	248	245	124	204	36	107
234	202	245	167	9	217	239	173
193	190	100	167	43	180	8	70
11	24	210	177	81	243	8	112
97	195	203	47	125	114	165	181
193	70	174	167	41	30	127	245
87	149	57	192	65	129	178	228

venga trasformato in questo modo dalla DCT2:

1.11e+03	4.40e+01	7.59e+01	-1.38e+02	3.50e+00	1.22e+02	1.95e+02	-1.01e+02
7.71e+01	1.14e+02	-2.18e+01	4.13e+01	8.77e+00	9.90e+01	1.38e+02	1.09e+01
4.48e+01	-6.27e+01	1.11e+02	-7.63e+01	1.24e+02	9.55e+01	-3.98e+01	5.85e+01
-6.99e+01	-4.02e+01	-2.34e+01	-7.67e+01	2.66e+01	-3.68e+01	6.61e+01	1.25e+02
-1.09e+02	-4.33e+01	-5.55e+01	8.17e+00	3.02e+01	-2.86e+01	2.44e+00	-9.41e+01
-5.38e+00	5.66e+01	1.73e+02	-3.54e+01	3.23e+01	3.34e+01	-5.81e+01	1.90e+01
7.88e+01	-6.45e+01	1.18e+02	-1.50e+01	-1.37e+02	-3.06e+01	-1.05e+02	3.98e+01
1.97e+01	-7.81e+01	9.72e-01	-7.23e+01	-2.15e+01	8.13e+01	6.37e+01	5.90e+00

Dovete inoltre controllare che la prima riga del blocchetto 8×8 sopra:

231	32	233	161	24	71	140	245

venga trasformata dalla vostra DCT monodimensionale in

4.01e+02	6.60e+00	1.09e+02	-1.12e+02	6.54e+01	1.21e+02	1.16e+02	2.88e+01

Fig.1

Verificata la correttezza abbiamo applicato la nostra implementazione della DCT2 e quella di OpenCV sulle stesse matrici verificando i tempi di esecuzione.

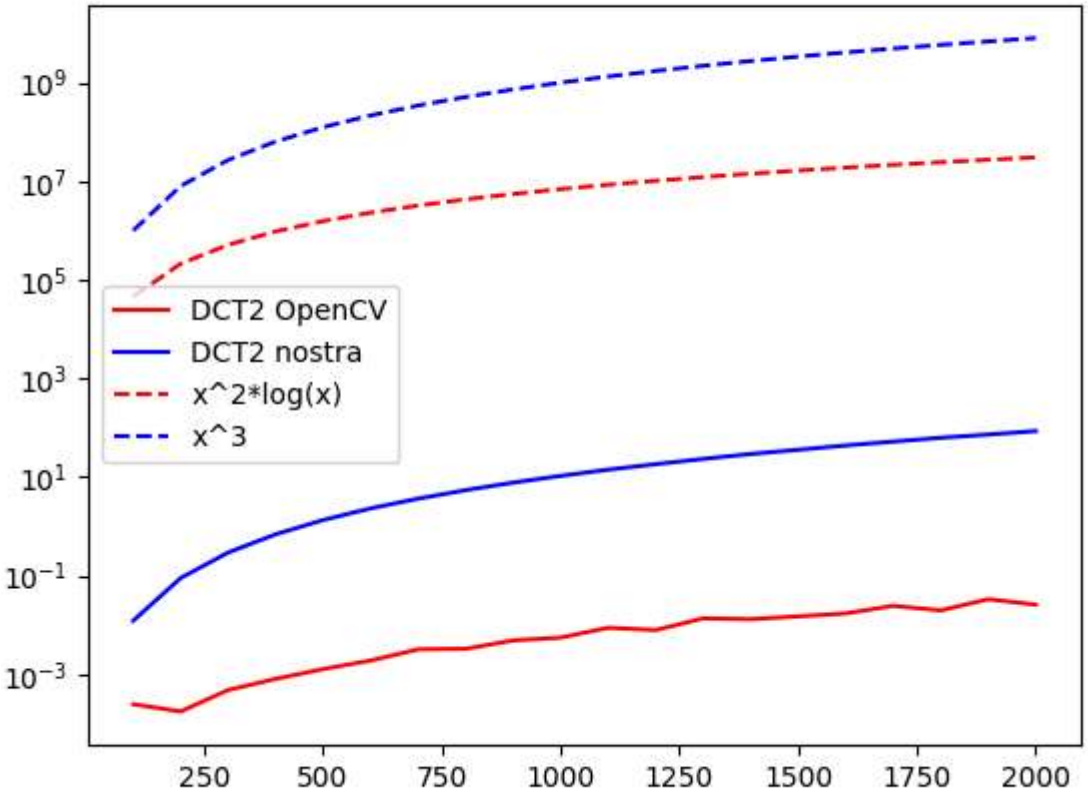

Fig.2

Come è possibile notare dal grafico in Fig.2, l'implementazione della DCT2 di OpenCV risulta molto più rapida della nostra, in quanto implementa la versione fast.

Parte 2

La seconda parte del progetto consiste nell'implementare un programma con interfaccia grafica, che l'utente potrà utilizzare per:

- Selezionare una immagine, in formato Windows Bitmap (.bmp), su cui effettuare al DCT2;
- Impostare un parametro F, maggiore di 0, che rappresenta l'ampiezza dei blocchi in cui sarà suddivisa l'immagine;
- Impostare un parametro d, compreso tra 0 e 2F-2, che rappresenta la soglia di taglio delle frequenze dell'immagine risultante dall'applicazione della DCT2.

Per realizzare questo programma abbiamo scelto il linguaggio C++ e la libreria OpenCV, una libreria open source di computer vision che offre, tra le altre cose, una implementazione della DCT. Inoltre, per realizzare l'interfaccia grafica, abbiamo scelto il framework Qt.

Il progetto si compone di tre parti:

- `main`: l'entry point del programma, che mostra l'interfaccia grafica;
- `MainWindow`: la classe che si occupa di gestire l'interfaccia grafica e l'interazione con l'utente;
- `DCT2`: la classe che applica la Discrete Cosine Transform fornita da OpenCV all'immagine selezionata dall'utente.

L'interfaccia grafica del programma, visibile in figura 3, è composta da:

- Immagine di sinistra: l'anteprima dell'immagine originale. L'immagine viene mostrata dopo che viene selezionata usando il pulsante "Select image", oppure dopo che viene modificato il contenuto del campo testuale contenente il percorso dell'immagine (se il percorso è corretto).
- Immagine di destra: l'anteprima del risultato dell'applicazione della DCT2 sull'immagine scelta, utilizzando i parametri impostati.
- Pulsante "Select image": permette di selezionare l'immagine utilizzando il selettore dei file del sistema.
- Campo testuale per il percorso dell'immagine: contiene il percorso dell'immagine selezionata. Il campo è modificabile dall'utente e in seguito alla sua modifica, se il campo contiene il percorso ad una immagine valida, verrà mostrata la sua anteprima.
- Campo testuale "Block size": il campo in cui può essere inserito il parametro F.
- Campo testuale "Threshold": il campo in cui può essere inserito il parametro d.
- Pulsante "Run": permette di avviare la procedura che applica la DCT2 all'immagine. Per funzionare, deve essere stata selezionata una immagine valida e devono essere stati inseriti dei parametri corretti. Se non si verifica una di questa due condizioni, l'interfaccia mostrerà un messaggio di errore sul campo contenente il valore non valido.

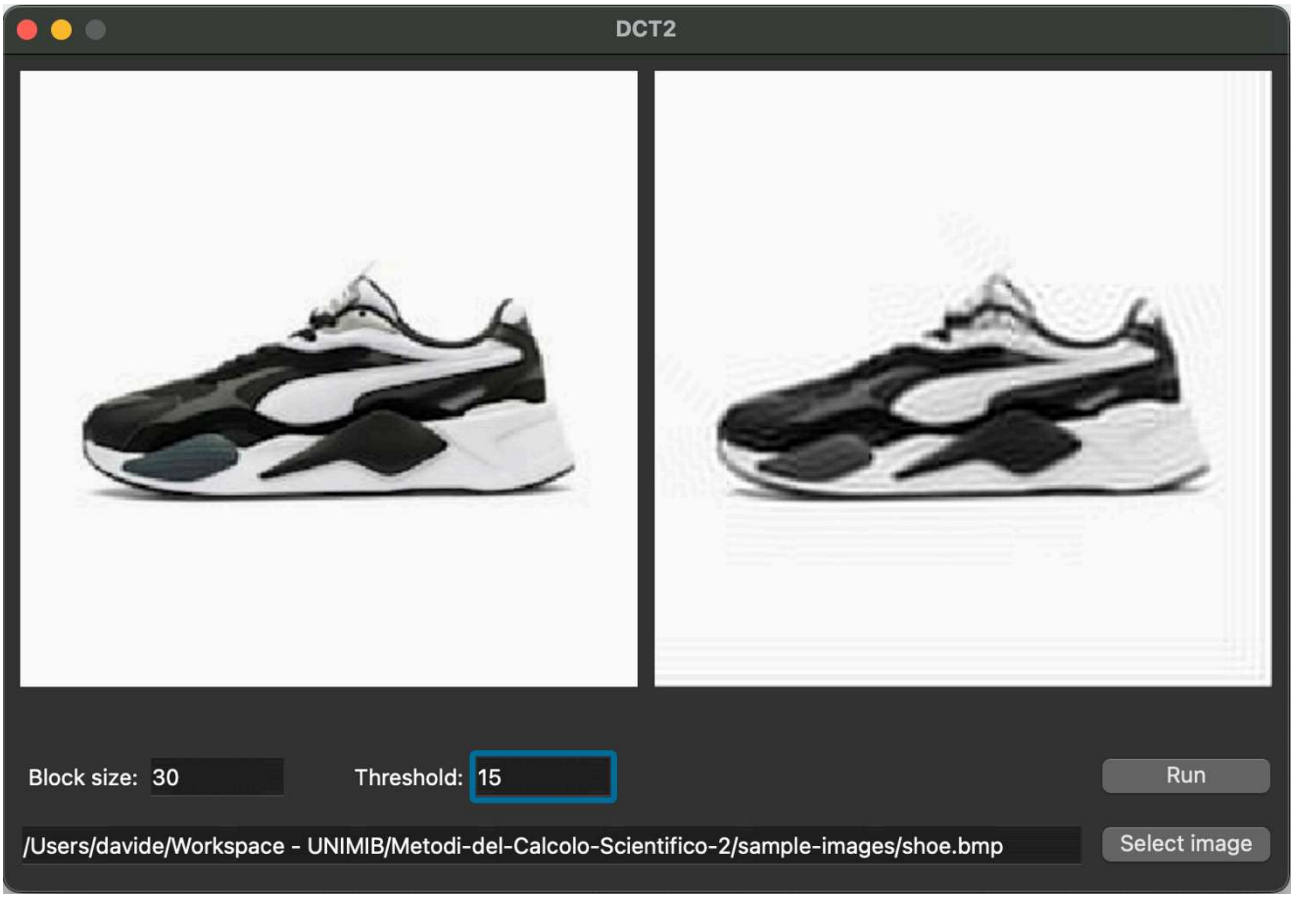

Fig. 3

Esempi di utilizzo

Nell'esempio che segue, è stata utilizzata l'immagine presente nella figura 4a con F = 20 e d = 15. Nell'immagine risultante, visibile nella figura 4b, è possibile notare alcuni artefatti. Questi sono causati dal fenomeno di Gibbs, che si verifica quando la funzione che vogliamo approssimare presenta dei punti di discontinuità.

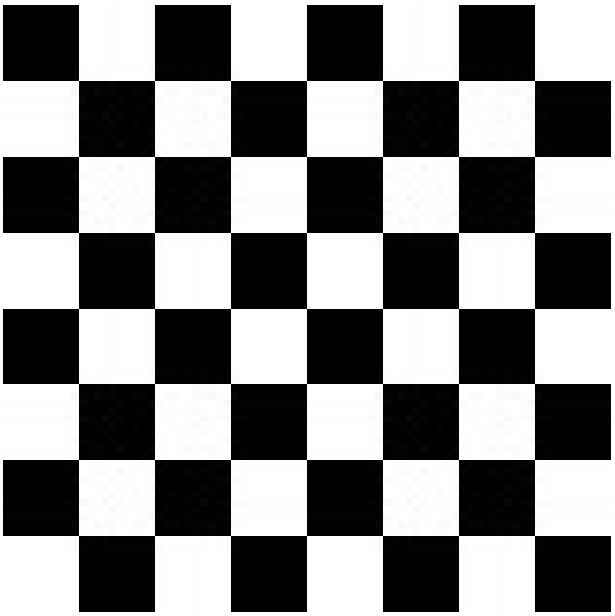

Fig. 4a

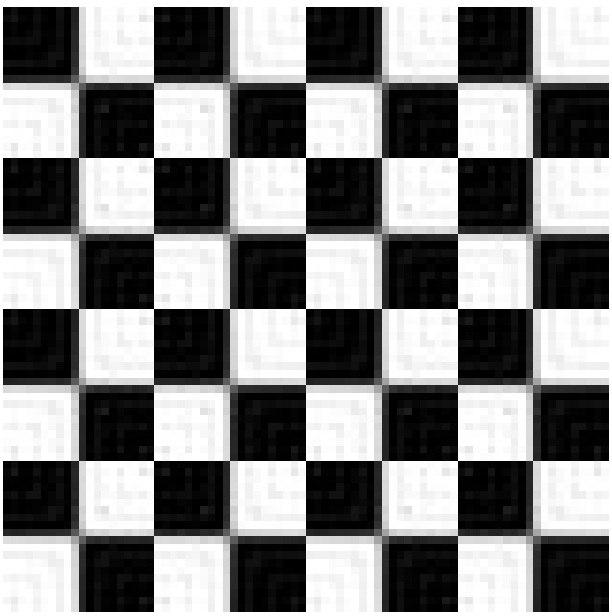

Fig. 4b

Aumentando la soglia di taglio delle frequenze, è possibile assistere ad una riduzione di questo fenomeno. Anche l'immagine in figura 5 è stata ottenuta partendo da quella presente nella figura 4a, utlizzando lo stesso parametro F, ma con d = 25:

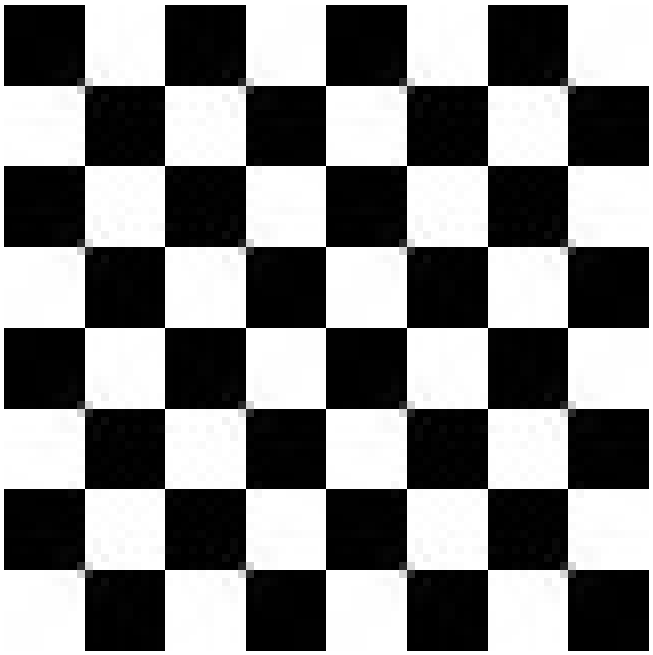

Fig. 5

Modificando il parametro F in modo che la dimensione dell'immagine non sia un suo multiplo, potrebbero verificarsi dei tagli nel lato destro o in basso dell'immagine. L'immagine presente in figura 6 è stata ottenuta ancora partendo dall'immagine in figura 4a (di dimensione 80x80), con d = 25 ma con F = 15. È possibile notare come i quadrati nell'ultima colonna a destra e nell'ultima riga in basso siano stati tagliati.

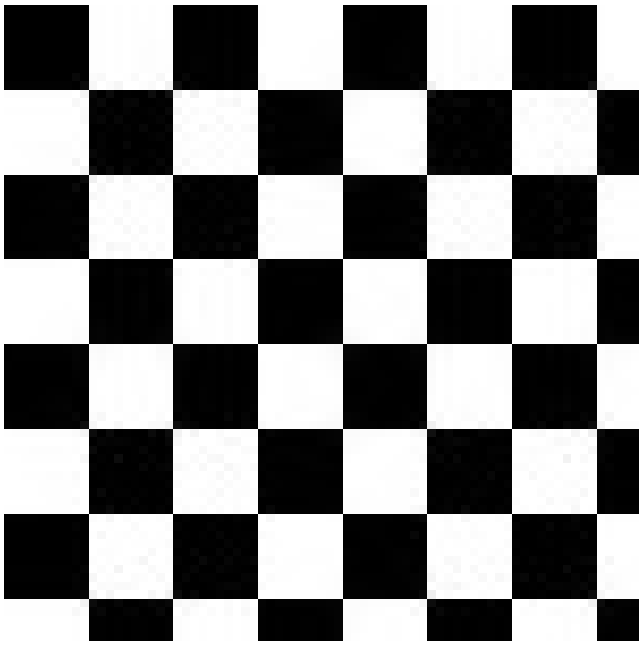

Fig. 6

La procedura è ovviamente applicabile anche ad immagini non quadrate. In questo caso è più probabile incorrere in tagli nell'immagine. Nel seguente esempio, l'immagine sorgente utilizzata, visibile nella figura 7a, ha dimensione 1000x600. I parametri utilizzati sono F = 215 e d = 180. Il risultato è visibile nella figura 7b.

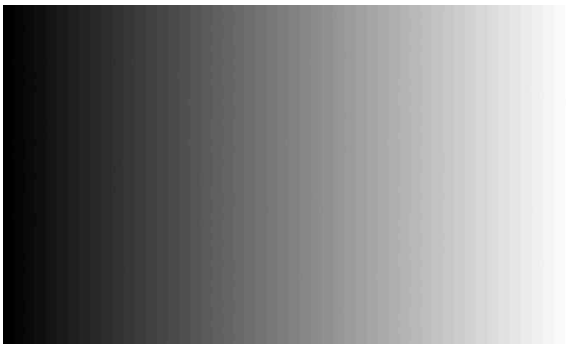

Fig. 7a

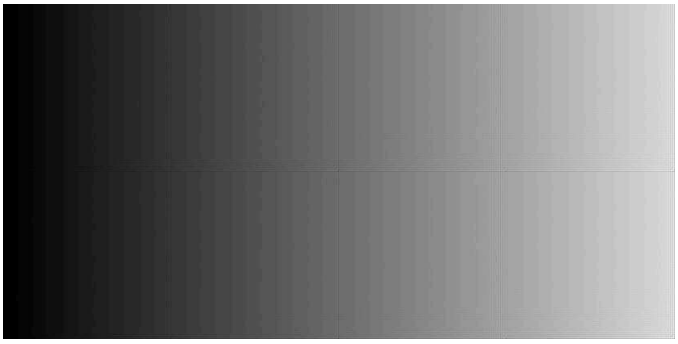

Fig. 7b