Università degli Studi di Milano - Bicocca
Scuola di Scienze
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Magistrale in Informatica

Metodi del Calcolo Scientifico - Progetto 2

Compressione di immagini tramite la DCT

Alberici Federico - 808058
Bettini Ivo Junior - 806878
Cocca Umberto - 807191
Traversa Silvia - 816435

Anno Accademico 2019 - 2020

Indice

Introduzione

In questa relazione vengono presentate e discusse le modalita di implementazione della DCT (dall'inglese Discrete Cosine Transform), ovvero la più diffusa funzione che provvede alla compressione spaziale.
Nella prima parte viene confrontata la versione "Nativa" della DCT con alcune varianti conosciute, studiandone il costo computazionale.
Nella seconda parte viene documentato un semplice tool per applicare su immagini in toni di grigio, tramite un approccio di compressione tipo jpeg (senza utilizzare una matrice di quantizzazione), la funzione DCT2 implemetata.

Analisi DCT

1.1 Discrete Cosine Transform

Una DCT esprime una sequenza finita di punti in termini di una somma di funzioni coseno oscillanti a diverse frequenze. Ad oggi è una delle tecniche di trasformazione piu utlizzate nella Teoria dei segnali e nella compressione dei dati, in particolare nei media digitali (audio, video, radio ecc..).
In queste applicazioni infatti la maggior parte delle informazioni significative tendono ad essere concentrate in poche componenti a bassa frequenza della DCT. Questo permette di comprimere a piacere il dato scartando le componenti ad alta frequenza (compressione lossy).

1.2 DCT-II e DCT-III

La DCT-II è probabilmente la forma più utilizzata, infatti viene indicata come "la DCT".

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

La sua inversa è la DCT-III e per questo viene indicata come "l'inversa della DCT" o "IDCT".

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N}n\left(k+\frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

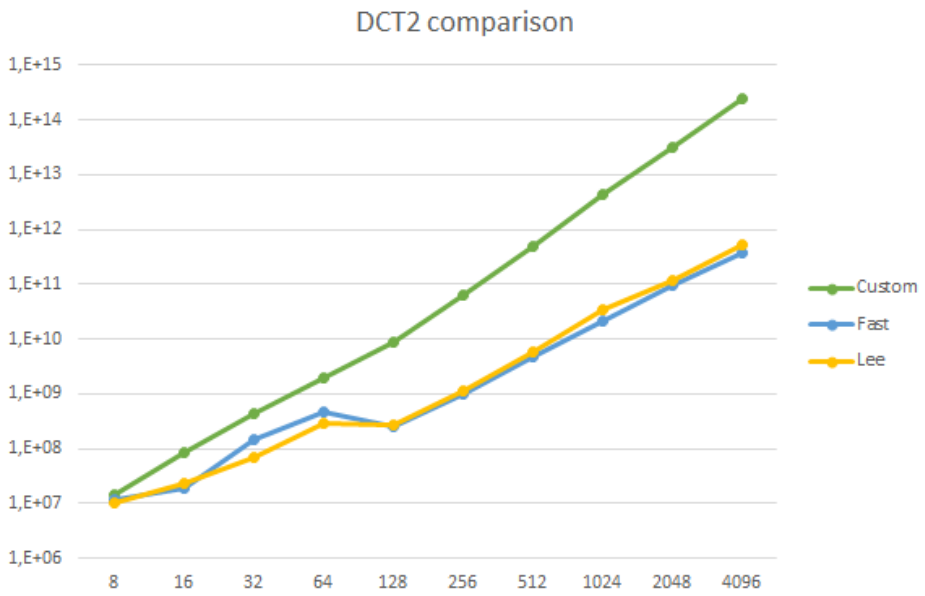
Entrambe le funzioni effettuano N somme per calcolare la k-esima componente di un vettore di N componenti, determinando un costo computazionale $O(N^2)$.

Implementazione

1.3 DCT tipo I - IV

1.4 DCT di Lee

1.5 Confronto



Test con immagini

Il programma è stato scritto tramite QT, una libreria multipiattaforma per lo sviluppo di programmi che utilizzano un'interfaccia grafica (attraverso l'uso di widget) che utlizza il linguaggio C++ (motivo per il quale abbiamo deciso di utilizzarlo).
Nella seconda parte del progetto, il programma utilizza le seguenti classi scritte da noi:

- main.cpp, la quale si occupa di eseguire l'intero corpo del programma;
- compress.cpp, la quale data un' immagine x esegue delle iterazioni per creare delle sottomatrici che verrano utiliazzate poi dalla libreria dct2 (presa da noi online, da sistemare). Inoltre, una volta che la libreria dct2 esegue il calcolo, questa classe si occupa di ricomporre l'immagine finale;
- mainwindow.cpp, classe che gestisce tutti gli aspetti e i trigger della interfaccia grafica;

Inoltre, viene utilizzata la libreria DCTFast nella quale ricaviamo la DCT2 operando prima per righe e poi per colonne.
Una volta avviato il programma, è possibile caricare l'immagine .bmp tramite un apposito tasto ed inserire i valori di f e d. Una volta inseriti i parametri, tramite il pulsante process viene chiamato il metodo on_parameters_clicked(), che dopo aver controllato che f e d rispettino tutti i vincoli trasmorma l'immagine (tramite la funzione pixmapToMatrix()) in una matrice e la invia alla funzione DCTcompress. Questa funzione divide l'immagine in blocchi, applica la DCT2 e poi restituisce una matrice che viene passata alla funzione matrixToPixmap() per poter essere visualizzata in output nell'iterfaccia grafica.
La funzione compress prende in input una matrice di interi e i parametri f e d restituendo in output una matrice di interi. Al suo interno, la matrice di input viene trasformata

nel formato double e viene eseguito un troncamento per poter scartare gli "avanzi". Iterativamente, per ogn i blocco quadrato f x f applichiamo la dct2 e poi ritorniamo la matrice nel formato int (arrottondando o suoi valori double all'intero più vicino, mettendo a zero i valori negativi e a 255 quelli maggiori di 255).

2.1 Tool di testing

2.2 Risultati