MCS 2020

Algebra lineare numerica
Compressione di immagini tramite la DCT

Silva Edoardo 816560, Zhigui Bryan 816335, Marchetti Davide 815990

21/05/2020

1 Abstract

Si vuole presentare un software che implementa una compressione delle immagini in modo tale che esegua i task descritti nella traccia.
Il software permette all'utente:

1. Scegliere dal filesystem un'immagine .bmp in toni di grigio riportando un messaggio d'errore qual ora scegliesse un formato diverso oppure a colori.

2. Scegliere un valore intero F che determina il valore massimo che può scegliere del valore d che né indicherà la soglia di taglio delle frequenze.

3. Visualizzare sullo schermo affiancante: l'immagine originale e l'immagine ottenuta post compressione.

4. Ridimensionare l'immagine per riempire l'area mancante rispetto a quella originale.

Il software è stato scritto in python per lo sviluppo della GUI moderna con PyQT5.

2 Implementazione

Il software inizialmente presenta un'interfaccia dotate con le seguenti caratteristiche:

1. Un pannello diviso a metà dedicando una sezione all'immagine originale e una sezione all'immagine compressa;

2. Una sezione di parametri nella quale sono suddivise in due parti:

   A **Input:** permette di andare a reperire l'immagine e scalarla alle dimensioni disponibili con l'apposito check-box

   B **Parameters:** definisce i parametri sulla qualità di compressione dell'immagine.

Il programma in **back-end** va a connettersi ai singoli eventi che deve avviare mano a mano che l'utente decide di effettuare un'operazione settando o facendo sparire alcune informazioni non più rilevanti.

3 Libreria PyQT5

PyQT5 è una libreria che consente di usare il framework QT5 GUI che serve per creare GUI application nel linguaggio C++.
PyQT5 è un toolkit multipiattaforma che può essere eseguito su quasi tutti i sistemi operativi.
Usandolo con Python, è possibile creare applicazioni molto rapidamente senza perdere gran parte della velocità del C++
I moduli utilizzati in particolare sono:

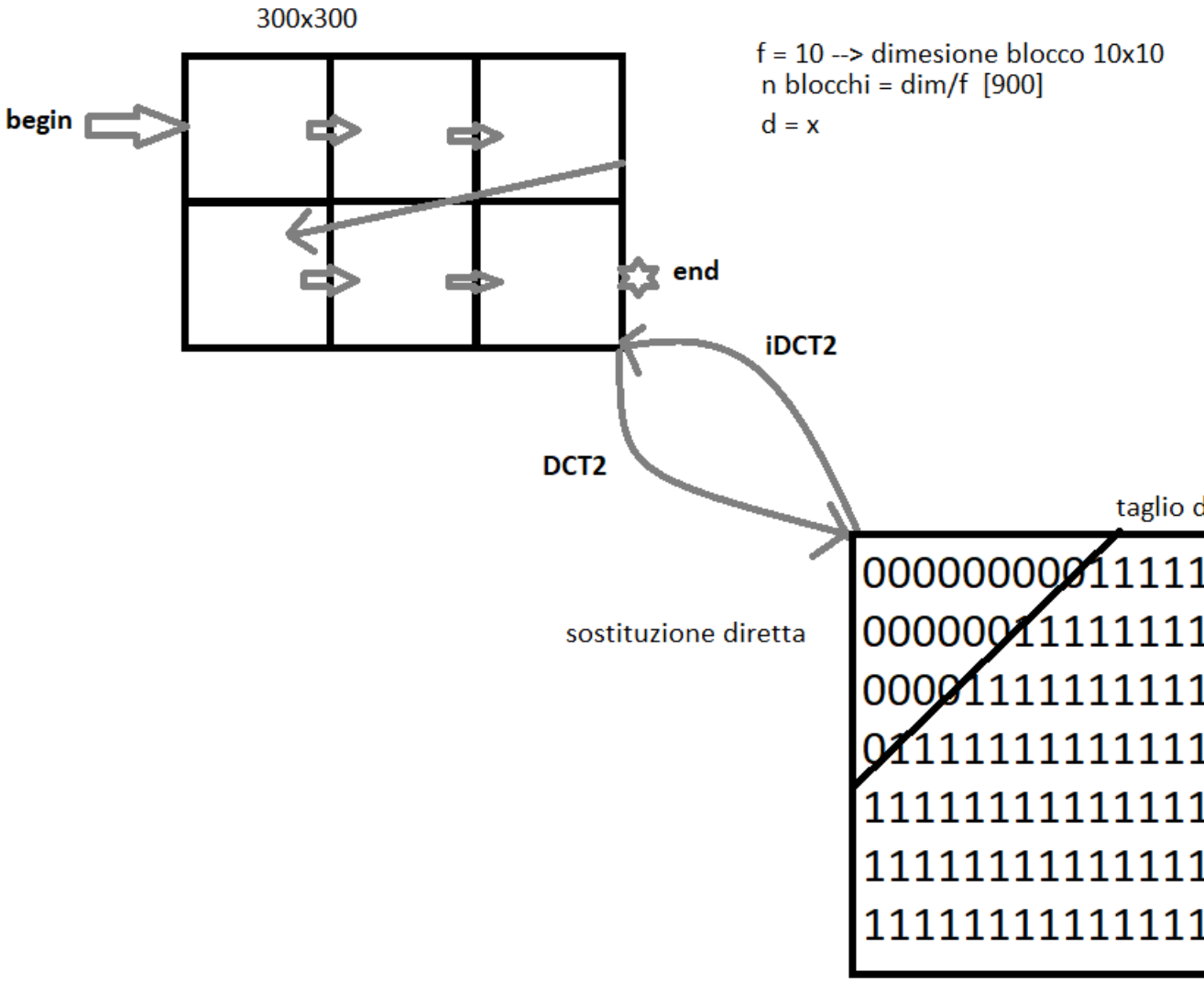
- **QtWidgets**
- **QtCore**
- **QtGUI**
- **QtWidgets**

4 Richiesta immagine

Si richiede all'utente di andare a reperire un'immagine cliccando sul bottone "open file" che presenta inizialmente la cartella dove si trova il progetto perché contiene una cartella apposita `..\folder\resources` dove può reperire immagini di esempio, ma può anche sceglierne una in locale del suo pc.
Una volta presa l'immagine va a verificare:

1. Se il formatto è .bmp, altrimenti presenta in messagge-box-error indicando che il formato dell'immagine è sbagliato;

2. Se l'immagine passata non sia a colori presentando un messagge-box-error indicando che l'immagine deve essere in toni di grigio.



L'immagine viene caricata nella prima sezione del pannello permettendo di scalarla come in figura ....

5 Compressione immagine

Una volta che viene caricata l'immagine l'utente dovrà definire i valori di F e d in modo tale da studiare la qualità della compressione con diversi esempi

che elencheremo alla fine.
Si spiega l'algoritmo nei seguenti passi:

1. L'immagine originale viene trasformata in formato pixmap;
2. Si definisce il numero totale di bit in rapporto con il numero di canali (RGBA= 4 canali);
3. Setta i valori di F e d immessi dall'utente;
   A Dato F si definisce l'ampiezza dei macro-blocchi;
4. Quindi l'immagine viene divisa in blocchi f di pixels di dimensione F x F, partendo dal primo blocco in alto a sinistra;
5. Per ogni blocco si effettuano le seguenti attività:
   A Si applica la dctn (DCT2 della libreria): c = DCT2(f);
   B Vengono eliminate le frequenze ckl con k+l >= d;
   C Si applica idctn sulla matrice c: ff = IDCT2(f)
6. Trasforma da bit a immagine;
7. Riporta l'immagine compressa sulla sezione dedicata.

6 Oservazioni

1. Le immagini richiedono di lavorare con i 4 tipi di canale: RGBA che in combinazione definiscono il colore.

La traccia richiede di lavorare con immagini in toni di grigio, dalle analisi fatte(come in figura in alto a destra) si nota che il valore di *un qualunque grigio* è uguale per tutti i canali.
Di conseguenza si è deciso di lavorare con un solo canale (in questo caso il primo, ovvero Red).

1. Quindi l'immagine viene divisa in blocchi f di pixels di dimensione F x F, partendo dal primo blocco in alto a sinistra;
2. Per ogni blocco si effettuano le seguenti attività:

A Si applica la dctn (DCT2 della libreria): c = DCT2(f);

B Vengono eliminate le frequenze ckl con k+l >= d;

C Si applica idctn sulla matrice c: ff = IDCT2(f)