



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

Corso di
Programmazione Logica e Funzionale

Anno accademico
2021-2022

Progetto realizzato da
Barzotti Cristian
290725
Kania Nicholas
291188

Corso tenuto dal professore
Bernardo Marco

Implementazione di DCT e DFT in Haskell e Prolog

Indice

1	Specifica del problema	4
2	Analisi del problema	5
2.1	Dati di ingresso del problema	5
2.2	Dati di uscita del problema	5
2.3	Relazioni intercorrenti tra i dati del problema	5
3	Progettazione dell'algoritmo	6
3.1	Scelte di progetto	6
3.2	Passi dell'algoritmo	6
4	Implementazione dell'algoritmo	7
5	Testing	8
5.1	Haskell	8
5.2	Prolog	8

Capitolo 1

Specifica del problema

Si propone di implementare le funzioni di elaborazione segnali e immagini denominate DCT (*Discrete Cosine Transform*) e DFT (*Discrete Fourier Transform*), e le relative funzioni inverse IDCT e IDFT.

Nota

Ci teniamo a precisare che la funzione DCT è suddivisa in diversi tipi; in questo progetto siamo andati ad implementare quelle che formalmente sono chiamate DCT tipo II e DCT tipo III¹. In questa documentazione ci riferiremo alle funzioni rispettivamente come DCT (per la funzione DCT tipo II) e IDCT (per la funzione DCT tipo III).

¹https://en.wikipedia.org/wiki/Discrete_cosine_transform

Capitolo 2

Analisi del problema

2.1 Dati di ingresso del problema

- un array di numeri reali per il calcolo di DCT e IDCT;
- un array di numeri complessi per il calcolo di DFT e IDFT.

2.2 Dati di uscita del problema

- un array di numeri reali calcolati con DCT;
- un array di numeri reali calcolati con IDCT;
- un array di numeri complessi calcolato con DFT;
- un array di numeri complessi calcolato con IDFT.

2.3 Relazioni intercorrenti tra i dati del problema

Capitolo 3

Progettazione dell'algoritmo

3.1 Scelte di progetto

3.2 Passi dell'algoritmo

I passi dell'algoritmo per effettuare le dovute trasformate sono i seguenti:

- Inserimento di un array di numeri da tastiera, per il calcolo della DCT;
-

Capitolo 4

Implementazione dell'algoritmo

Capitolo 5

Testing

5.1 Haskell

5.2 Prolog