

Parallel histogram equalization

Macaluso Alberto

Matricola: 7115684

alberto.macaluso@edu.unifi.it

Abstract

Nell'elaborato è stato trattato un algoritmo in grado di effettuare l'histogram equalization di immagini in scala di grigi. Inizialmente verrà descritto l'algoritmo nella sua forma base (sequenziale) e in seguito saranno illustrate due diverse implementazioni parallele basate sull'architettura CUDA. Infine verranno mostrate un'analisi e un confronto delle varie versioni andando ad evidenziare le caratteristiche principali.

1. Introduzione

L'histogram equalization è una tecnica di elaborazione delle immagini utilizzata per migliorare il contrasto in un'immagine. Consiste nel riassegnare la distribuzione dei livelli di grigio in modo uniforme sull'intero intervallo disponibile, producendo un'immagine con un contrasto migliore e una maggiore vivacità dei dettagli. Questa tecnica è particolarmente utile quando l'istogramma dell'immagine originale è concentrato su una piccola gamma di valori di grigio.

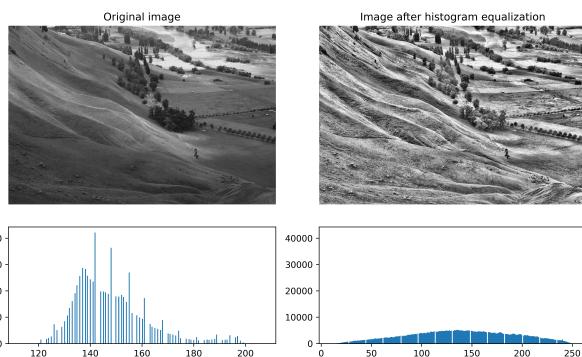


Figure 1: Esempio di histogram equalization

Ad ogni pixel dell'immagine originale viene applicata una trasformazione calcolata in modo tale da generare una nuova immagine con un istogramma "piatto".

L'algoritmo[1] può essere descritto nei seguenti passaggi:

1. Calcolare l'istogramma dell'immagine in ingresso, $H(i)$ dove $i = 0, 1, 2, \dots, L - 1$ e L è il numero di livelli di intensità (tipicamente 256 per immagini a 8 bit).
2. Normalizzare l'istogramma per ottenere la funzione di densità di probabilità (PDF), $P(i) = \frac{H(i)}{MN}$ dove M è il numero di righe e N è il numero di colonne dell'immagine.
3. Calcolare la funzione di distribuzione cumulativa (CDF), $C(i) = \sum_{j=0}^i P(j)$.
4. Mappare ogni intensità del pixel r in una nuova intensità s usando la formula:
$$s = (L - 1) \cdot C(r)$$
5. Infine, sostituire ogni intensità del pixel r nell'immagine originale con la sua corrispondente nuova intensità s .

Questi step assicurano che l'istogramma dell'immagine in uscita sia approssimativamente uniforme, portando ad un contrasto migliorato.

1.1. CLAHE

Nell'implementazione è stata utilizzata una versione più avanzata dell'histogram equalization base appena descritta. Contrast Limited Adaptive Histogram Equalization (CLAHE[2]) è una variante dell'histogram equalization che limita l'amplificazione del contrasto in aree dell'immagine con grande differenza di intensità. A differenza dell'histogram equalization standard, CLAHE divide l'immagine in regioni più piccole e applica l'equalizzazione dell'istogramma a ciascuna di queste regioni in modo adattivo.

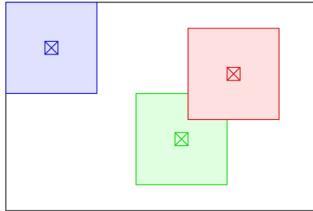


Figure 2: Applicazione dell'equalizzazione nelle regioni
(Fonte: Wikipedia[2])

Inoltre, per evitare l'aumento del rumore nelle aree uniformi, CLAHE utilizza un meccanismo di limitazione del contrasto. Questo consiste nel ridistribuire uniformemente i valori dell'istogramma che superano un certo limite (clip). Ciò impedisce che il contrasto venga amplificato eccessivamente, mantenendo una migliore qualità dell'immagine finale.

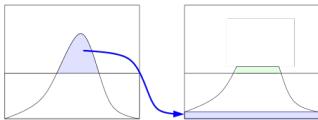


Figure 3: Redistribuzione dei valori oltre il limite (Fonte: Wikipedia[2])

1.2. Gestione dei bordi

Per poter applicare l'histogram equalization in modo adattivo è stato necessario gestire i pixel presenti nei bordi delle immagini. Per questo motivo è stata utilizzata una tecnica di border mirroring che consiste nello specchiare i valori dei pixel lungo i bordi delle immagini. L'offset per effettuare il mirroring è dato dal raggio delle regioni su cui verrà in seguito applicata l'equalizzazione dell'istogramma. In questo modo, per costruzione, ogni pixel avrà un intorno sufficiente a contenere tale finestra.

2. Implementazione in CUDA

spiegare come sono stati utilizzati i thread per effettuare l'histogram equalization, forse mettere immagine

2.1. Utilizzo della shared memory

spiegare la versione con shared memory, utilizzare schema

2.2. Profiling

aggiungere foto della profilazione fatta con nvidia Nsight

3. Risultati

descrivere le condizioni con cui sono stati effettuati tutti i benchmarks

3.1. CPU vs GPU

3.2. GPU vs GPU con shared memory

References

- [1] W. contributors, “histogram equalization formulation,” 2024. https://en.wikipedia.org/wiki/Histogram_equalization#Implementation.
- [2] W. contributors, “Adaptive histogram equalization,” 2022. https://en.wikipedia.org/w/index.php?title=Adaptive_histogram_equalization&oldid=1115555840.