

Progetto 1

Elaborazione di Immagine

Specifiche:

Sviluppo di una applicazione, lanciata da linea di comando, in grado di elaborare un'immagine a colori in formato BMP RGB non compresso (Versione 3, .bmp) e di salvare il risultato, eventualmente riscalato a nuove dimensioni.

L'applicazione viene lanciata da linea di comando con la seguente sintassi:

> elaboraimmagine <fileinput> <comando> <dimensioni_x> <dimensioni_y> <fileoutput>

dove:

<fileinput> e' il file .bmp di input

<comando> e' il tipo di elaborazione che si richiedi (vedere sotto)

<dimensioni_x>, un numero, e' la nuova dimensione x in pixel (riscalata) del file di output

<dimensioni_y> stessa cosa per la dimensione y (riscalata) del file di output

<fileoutput> nome del file di output che verra' creato.

Il <comando> e' una stringa di testo che puo' essere sia "blur" che "sharpen" e indica il tipo di filtro a convoluzione da applicare all'immagine. I kernel dei filtri possono essere semplici kernel 3x3.

Realizzazione:

Come prima cosa l'applicazione deve leggere correttamente un file in formato .bmp non compresso a colori RGB versione 3.

Un file di tale tipo e' composto da un header iniziale contenente tutta una serie di informazioni (ad esempio larghezza ed altezza dell'immagine), seguito dalla codifica RGB dei pixel dell'immagine stessa. Una buona idea puo' essere quella di riservare un'area di memoria in cui caricare inizialmente l'header del file, leggervi le informazioni importanti (in questo caso altezza e larghezza) per poi, dinamicamente, creare uno spazio di memoria pari a LxHx3 byte atto a contenere le LxH terne di byte RGB di ogni pixel e leggervi il resto del file. Per una descrizione dettagliata del formato BMP con esempi di codice scritto in C si puo' fare riferimento (ad esempio) a questo sito: <http://www.brescianet.com/appunti/vari/tracciati/bmpformat.htm> .

Sara' poi necessario creare un secondo buffer di output da "riempire" applicando l'opportuno kernel a convoluzione sull'immagine cosi' importata (si faccia riferimento agli esempi forniti e visti a lezione), costruendo la nuova immagine filtrata.

A questo punto si dovra' provvedere, nel caso in cui le dimensioni richieste per il file di output NON coincidano con quelle di partenza, a riscalare l'immagine (in un ulteriore, nuovo buffer) utilizzando un filtro di scala (si faccia riferimento, di nuovo, agli esempi forniti e visti a lezione).

In ultimo, si salvera' il risultato finale creando un nuovo file binario, scrivendo prima il "vecchio" header seguito poi dal nuovo buffer contenente l'immagine filtrata. Si faccia attenzione a modificare l'header, nelle parti relative alla dimensione dell'immagine, nel caso in cui l'output

finale abbia dimensioni differenti, affinché il file .bmp così costruito sia coerente e sia possibile visualizzarlo con qualsiasi altro programma.

Punti Facoltativi:

- si implementino altri possibili comandi/filtri (come quelli, ad esempio, visti a lezione).
- si implementi la possibilità di leggere/scrivere file in formato PNG
- (difficile) si implementi la possibilità di leggere/scrivere le immagini ANCHE in un formato proprietario compresso; si lascia completa libertà di decidere la strategia di compressione da applicare.