

Daniele Gambera

Giorgio Leonardi

Sistemi multimediali

14 giugno 2020

Relazione progetto esame

Scopo del progetto, creare un programma JAVA che date in input delle immagini, simuli il comportamento del filtro HDR(High Dynamic Range) delle macchine fotografiche.

SVOLGIMENTO

Per lo scopo, sono state create due classi JAVA, una chiamata *HDRFusionAlpha* che si occupa di eseguire tutti i calcoli ed un main che ne richiama i metodi:

Main: Inizializza le variabili, dal array di *BufferedImage* che contiene tutte le foto da sovrapporre, a quello degli alfa che specifica quanto una determinata immagine incide sul risultato finale. Viene stampato a schermo un menù testuale che chiede all'utente quante foto vuole usare, queste dovranno essere rinominate con numeri naturali crescenti (es. 1.jpg, 2.jpg, ecc..). A questo punto è possibile scegliere tra quattro "effetti": personalizzata, si dà la possibilità di inserire valori di alfa per ogni singola foto; costante standard, i valori di alfa per tutte le foto è dato in maniera predeterminata ed ha valore (0,5) per tutte le immagini; costante personalizzata, si inserisce un valore di alfa che sarà costante a tutte le foto; gaussiana, si dà più importanza alla foto centrale e meno a quelle agli estremi. Ogni inserimento da tastiera viene controllato in modo che i dati non siano nel formato sbagliato. Successivamente chiama i metodi presenti nella

seconda classe JAVA che si occupano dei calcoli. Per concludere inserisce le due immagini, quella usata come base e quella processata, in due frame che vengono mostrate a schermo.

HDRFusionAlpha: Comprende quattro metodi, `setSourceData()` e `setWeights()` che si occupano di istanziare gli array di immagini e quello dei pesi(alfa). Inoltre crea un'immagine risultato con le stesse caratteristiche della prima immagine di quelle da elaborare.

Il metodo `calculate()` ha 4 cicli for annidati che selezionano pixel per pixel e banda per banda, tutte le immagini che si vogliono sovrapporre, calcola la somma di ogni componente con la seguente formula:

$$C(x,y) = (I_0(x,y) * \alpha_0 + I_1(x,y) * \alpha_1 + \dots + I_n(x,y) * \alpha_n) / (\alpha_0 + \alpha_1 + \dots + \alpha_n)$$

alfa qui rappresenta la trasparenza dell'immagine in questione, più il suo valore è vicino allo 0 meno sarà marcato quel pixel di quell'immagine nella sovrapposizione finale.

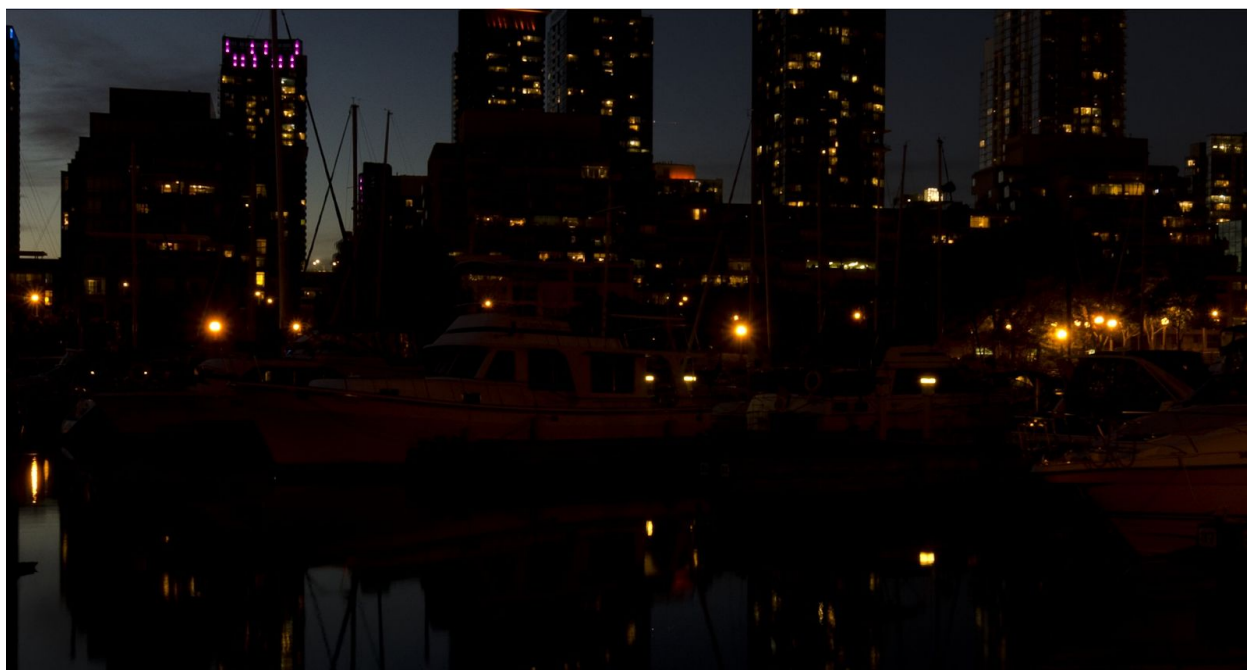
Per finire, la somma viene scritta nell'immagine di destinazione.

`getResult()` restituisce l'immagine dopo che è stata processata.

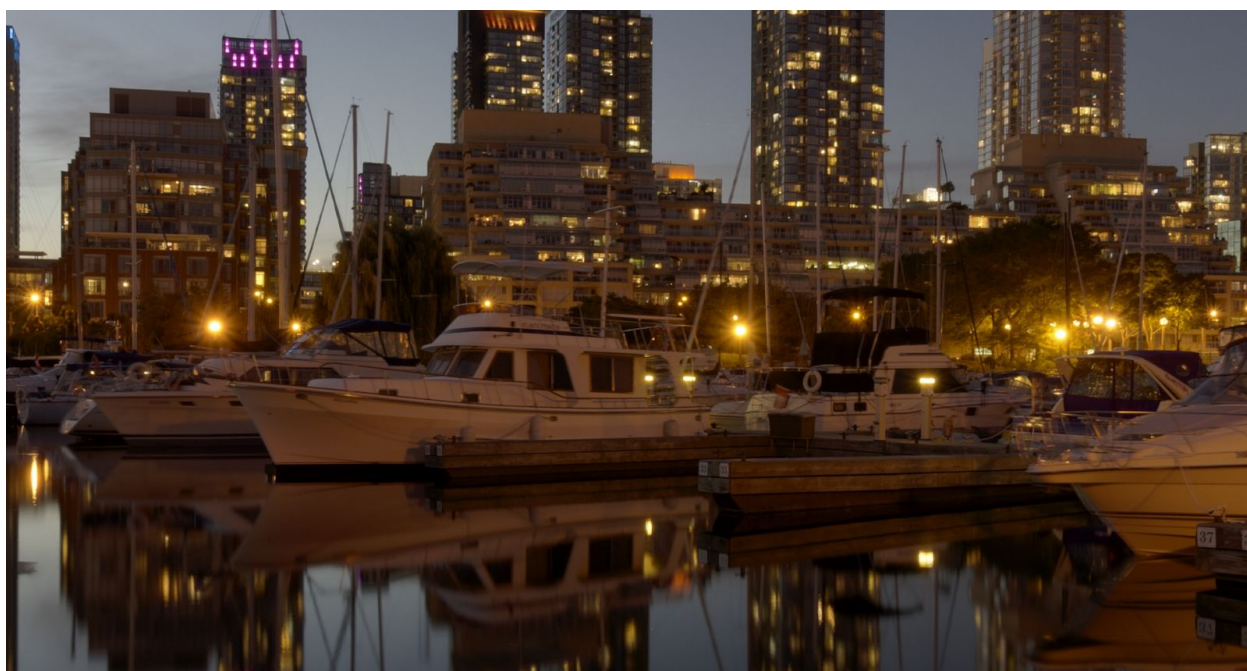
CONCLUSIONE

E' possibile inserire le proprie immagini, purché si rispettino le seguenti caratteristiche: tutte le foto devono essere dello stesso soggetto e della stessa grandezza.

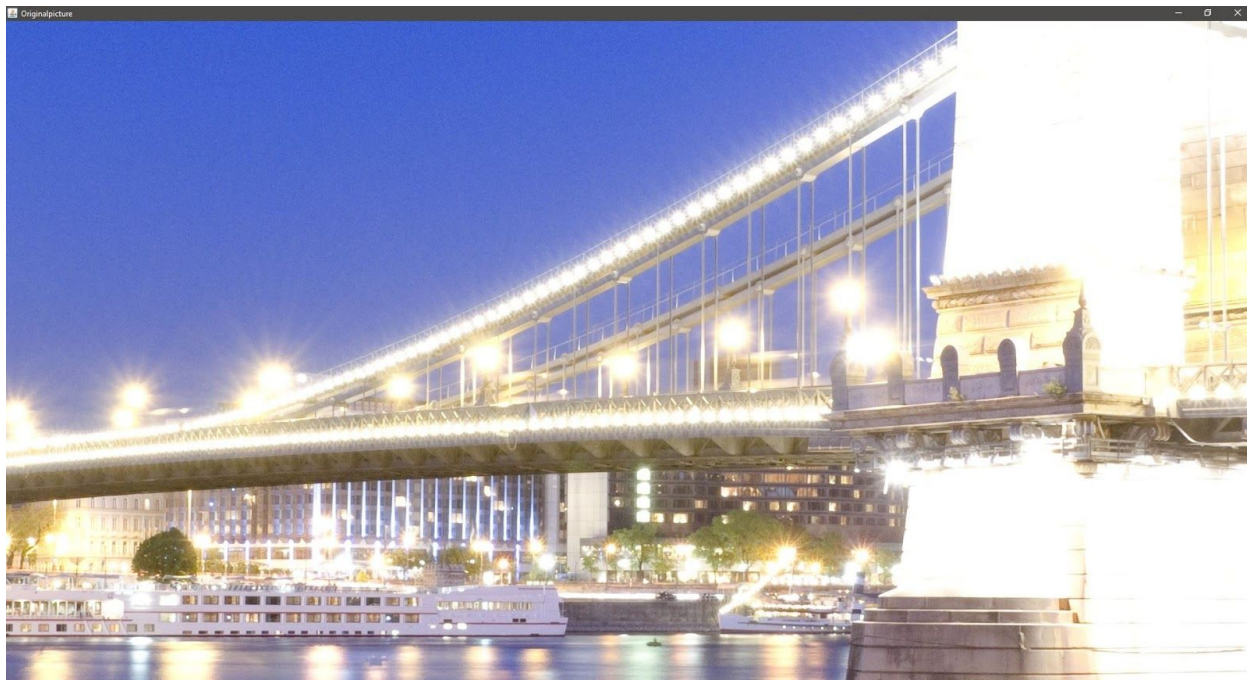
Usando le immagini che mi sono state date come esempio sono riuscito ad ottenere i seguenti risultati:



Sopra - Immagine di partenza



Sopra - Immagine con 4 sovrapposizioni e valori di alfa: 0,5 0,7 0,1 0,3



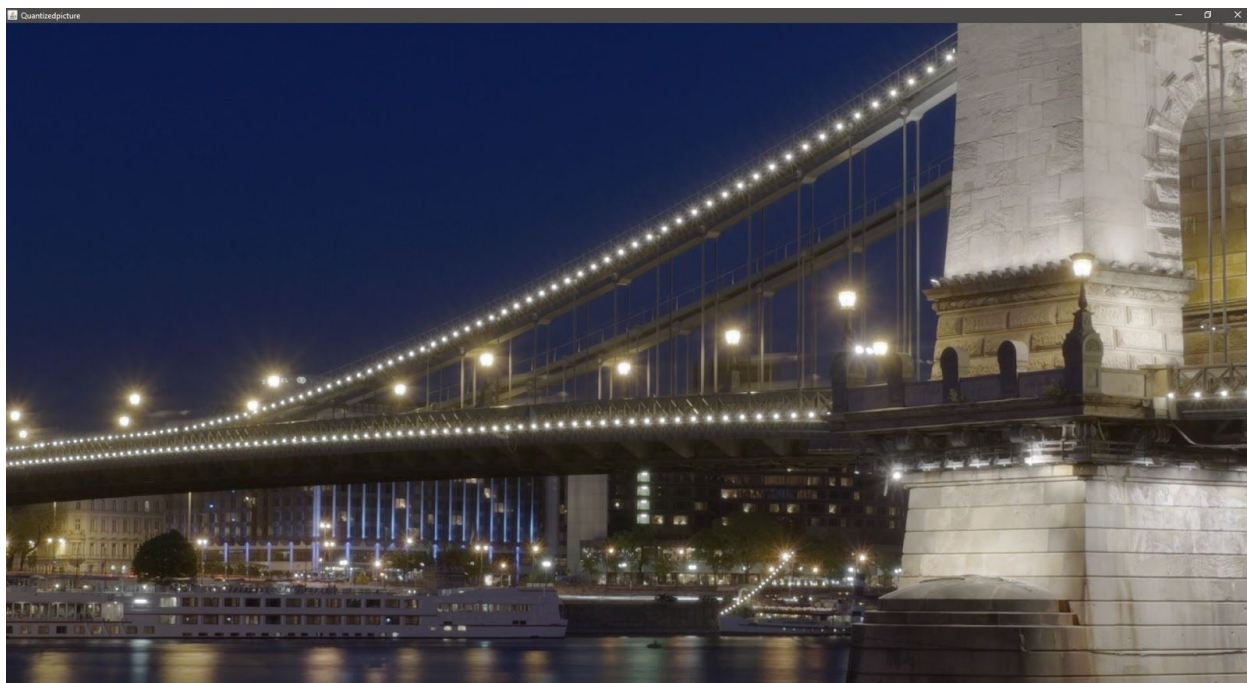
Sopra - Immagine di partenza



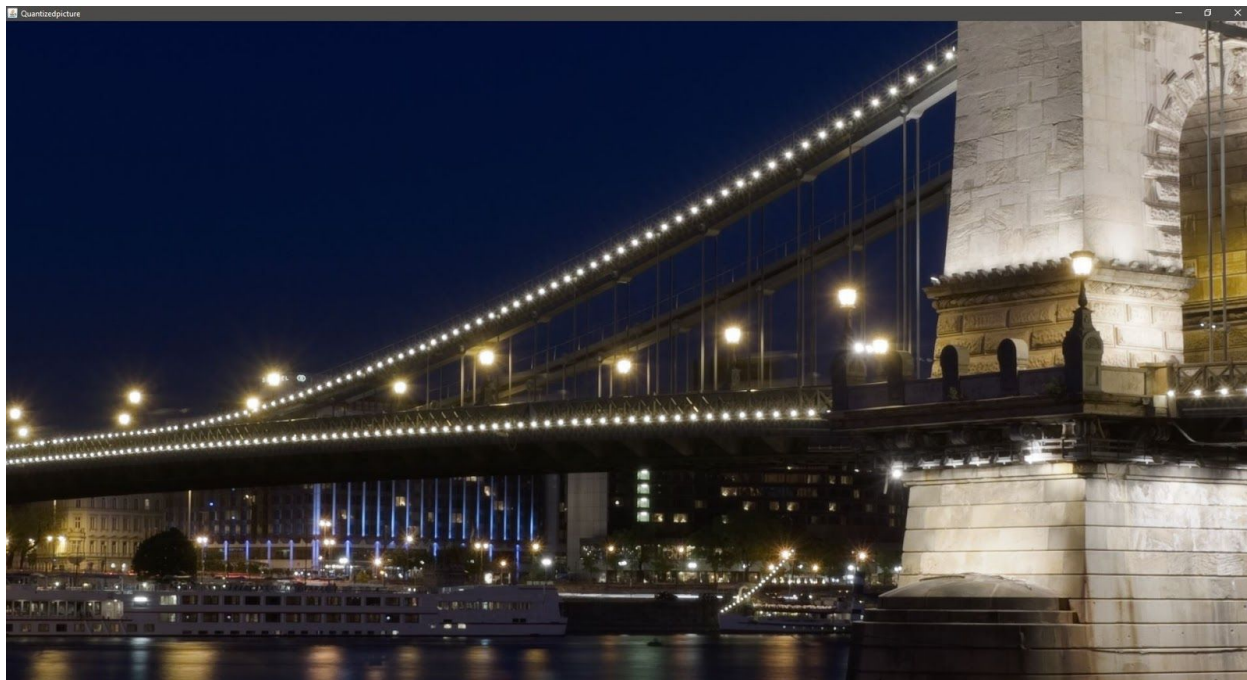
Sopra - Immagine con 5 sovrapposizioni e valori di alfa: 0,8 0,1 0,6 0,2 0,7



Sopra - Immagine con 5 sovrapposizioni e valori di alfa costante standard: 0,5



Sopra - Immagine con 5 sovrapposizioni e valori di alfa costante personalizzata: 0,8



Sopra - Immagine con 5 sovrapposizioni e valori di alfa gaussiani con valore centrale: 0,6