

Exposure Fusion

Dario Loi (1940849), Flavio Gezzi (1958690)

Abstract—Exposure Fusion é una tecnica che permette di trasformare una sequenza di immagini con diversi valori di esposizione in un'unica immagine ad alto range dinamico (HDR), per questo progetto ci siamo concentrati sull'implementazione originale di questo metodo [1], ottenendo buoni risultati in termini qualitativi, abbiamo inoltre implementato una versione alternativa, che fa uso di alcuni step di pre-elaborazione, ottenendo risultati piú robusti e ovviando a una serie di limitazioni dovute a i mezzi a nostra disposizione.

1 INTRODUZIONE

Il metodo originale é dettagliato con precisione nel materiale originale [1], in questo documento ci limiteremo a riassumere i passaggi principali.

Per prima cosa, data una sequenza di immagini $I = \{I_0, \dots, I_N\}$, si calcolano tre metriche per ogni immagine I_i , il contrasto, la saturazione e l'esposizione:

$$C_i = L \circledast G_i \quad (1)$$

Dove L é un filtro di Laplace, \circledast é la convoluzione e G_i é l'immagine i -esima in grayscale.

$$S_{i,j,k} = \sum_{u=1}^3 \left(I_{i,j,k,u} - \frac{1}{3} \sum_{v=1}^3 I_{i,j,k,v} \right)^2 \quad (2)$$

$$E_{i,j,k} = \prod_{u=1}^3 \left(\exp\left(-\frac{(I_{i,j,k,u} - 0.5)^2}{2\sigma^2}\right) \right) \quad (3)$$

Il valore di σ in (3) é stato scelto empiricamente, e corrisponde a $\sigma = 0.2$.

Successivamente, si utilizzano queste tre metriche per calcolare una maschera di fusione W_i :

$$W_i = C_i \cdot S_i \cdot E_i \quad (4)$$

La maschera di fusione é normalizzata in modo che la somma di tutti i suoi valori sia 1, andando così a costituire una distribuzione di probabilitá.

$$\hat{W}_i = \frac{W_i}{\sum_{j=0}^N W_j} \quad (5)$$

Infine, per ottenere l'immagine finale in HDR, si costruiscono due piramidi, una gaussiana e una laplaciana, a partire rispettivamente dalle maschere di fusione e dalle immagini originali. Le piramidi vengono poi fuse, ottenendo una nuova piramide laplaciana, che viene fatta collassare in un'immagine finale in HDR.

2 IMPLEMENTAZIONE

Abbiamo scelto di fornire un implementazione in Python, disponibile in allegato, che permette di eseguire sia il metodo originale, sia la versione alternativa. La nostra implementazione é inoltre provvista di un'interfaccia grafica,

che permette all'utente di caricare le immagini, di scegliere il metodo da utilizzare e di salvare i risultati.

Per testare l'implementazione abbiamo inoltre fatto riferimento a un implementazione in Python già disponibile [2].

2.1 Il nostro metodo

A causa della mancanza di apparecchiature professionali, alcuni dei nostri test sono stati effettuati su scatti ottenuti da fotocamere mobile, di solito allineati in maniera pessima, per risolvere questo problema (e per migliorare la qualità delle immagini) abbiamo inserito uno step facoltativo di pre-elaborazione, che permette di allineare le immagini, il metodo utilizzato é il seguente:

- 1) Per ogni immagine si applicano le seguenti operazioni:
 - a) Si equalizza l'istogramma dell'immagine, per eliminare eventuali differenze di esposizione che porterebbero a falsi positivi.
 - b) Si applica un gaussian blur, per eliminare eventuali rumori.
 - c) Si converte l'immagine in grayscale.
- 2) Si sceglie l'immagine al centro della sequenza come immagine di riferimento.
- 3) Per ogni immagine, si utilizza un feature detector per trovare i punti di interesse, é stato empiricamente scelto di utilizzare ORB [3], sebbene ci siano diverse alternative. [4], [5]
- 4) Si costruisce una matrice di omografia tra l'immagine di riferimento e ogni altra immagine, utilizzando i punti di interesse trovati precedentemente, dopo aver opportunamente scartato gli outlier.
- 5) Si proiettano le immagini originali (non quelle pre-elaborate) allineandone rispetto all'immagine di riferimento.

Così facendo, si ottiene una sequenza di immagini i quali punti di interesse sono allineati, il metodo é funzionale qualora le immagini siano allineate in modo simile, ma non perfettamente, fallisce invece se le deformazioni tra le immagini sono troppo grandi, in questo caso l'oggetto di interesse dell'immagine risulta allineato ma si vengono a creare degli artefatti in corrispondenza dei bordi, come si può vedere nella figura 1.

Artefatti di allineamento



Figure 1: Possiamo notare nell’immagine originale (b) gli scatti sono troppo disallineati per permettere la corretta fusione dell’immagine, ciò risulta in un’immagine finale completamente sfocata, nell’immagine allineata (a) la fusione è effettuata correttamente nelle vicinanze del modellino, in ogni modo, è evidente la presenza di un artefatto di deformazione nell’angolo in alto a sinistra della foto che rende il risultato indesiderabile.

Exposure Fusion - Immagini Reflex

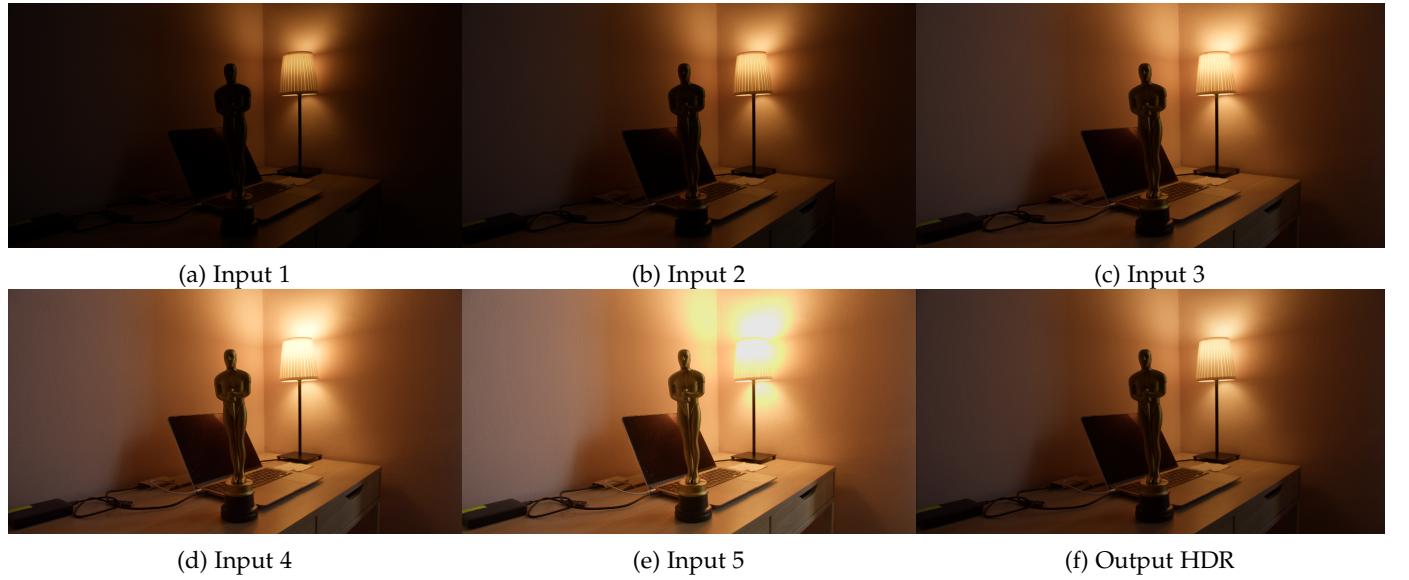


Figure 2: Abbiamo applicato il nostro metodo ad una serie di immagini (a, b, c, d, e) ottenute con una fotocamera reflex, l’immagine finale (f) non dimostra alcun tipo di deformazione da allineamento. L’illuminazione della lampada è molto più dolce rispetto alle immagini con forte esposizione, mentre sulla statuetta sono visibili un buon numero di dettagli. Possiamo inoltre notare che, in modo analogo a quanto visto per le immagini con flash nella paper originale [1], le immagini LDR ad alta esposizione sono fortemente illuminate dalla lampada, l’immagine HDR riesce a ridistribuire l’illuminazione in modo tale da consentire una maggior chiarezza delle aree in penombra.

3 RISULTATI

Il nostro metodo è stato testato su una serie di immagini ottenute con una fotocamera mobile, insieme a un numero limitato di immagini ottenute con una fotocamera reflex.

In seguito, verranno mostrati e commentati diversi risultati ottenuti con il nostro metodo, le immagini verranno rese disponibili in allegato in modo da rendere possibile la riproducibilità dei risultati.

Exposure Fusion - Ulteriori Esempi Reflex

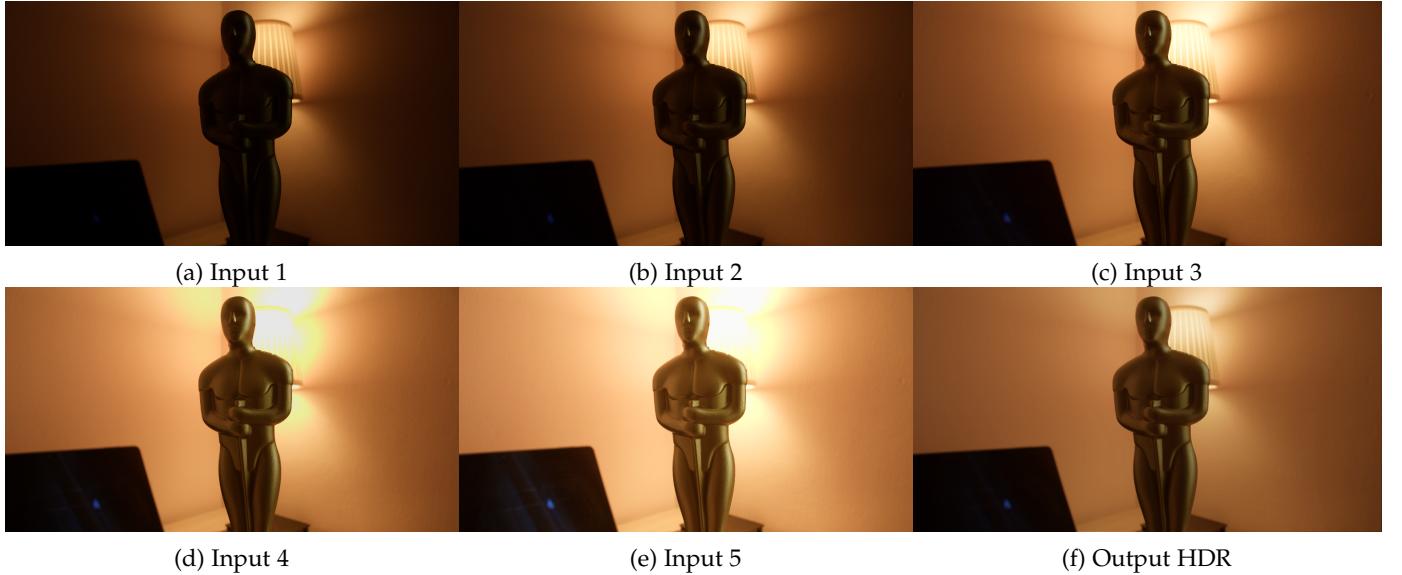


Figure 3: In seguito possiamo osservare un'altra applicazione del nostro metodo, anche qui abbiamo una sequenza di immagini (a, b, c, d, e) ottenute con una fotocamera reflex, l'immagine finale (f) non dimostra alcun tipo di deformazione e, analogamente al caso precedente.

Generalmente, il nostro metodo è in grado di ottenere risultati simili a quelli della paper originale, presentando allo stesso tempo le stesse limitazioni.

È infatti noto agli autori originali che il metodo presenta degli artefatti a bassa frequenza quando la profondità delle piramidi Gaussiane e Laplaciane è troppo bassa, questo problema è risolvibile aumentando la profondità, ciò però aumenta il tempo di computazione e può risultare in un secondo tipo di artefatto, in cui i valori di intensità non sono più presenti nell'intervallo (0, 1], questo problema è risolto da metodi più recenti [6], ma rimane nella nostra implementazione.

4 CONCLUSIONI

Crediamo di aver raggiunto risultati solidi, sviluppando un modulo che permette di implementare Exposure Fusion così come descritta originariamente.

Questo documento è accompagnato dalle seguenti risorse:

- Il codice sorgente del modulo, importabile come classe in qualsiasi file python.
- Un file Jupyter Notebook contenente esempi di utilizzo del modulo, atti a riprodurre i risultati mostrati in questo documento.
- Un'applicazione GUI realizzata con la libreria python TKinter, che permette l'interazione con exposure fusion attraverso un'interfaccia grafica.

REFERENCES

- [1] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion," 10 2007, pp. 382 – 390. [Online]. Available: <https://web.stanford.edu/class/cs231m/project-1/exposure-fusion.pdf>
- [2] arpesenti, "Exposure fusion," https://github.com/arpesenti/exposure_fusion, 2020.
- [3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," 11 2011, pp. 2564–2571.

- [4] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.
- [5] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.
- [6] C. Hessel, "Extended Exposure Fusion," *Image Processing On Line*, vol. 9, pp. 453–468, 2019, <https://doi.org/10.5201/ipol.2019.278>.