

1. Zistíte krivku odozvy kamery z LDR obrázkov



Svetlosť môžeme modelovať pomocou nameranej hodnoty i -teho pixela v j -tej expozícii ako

$$Z[i, j] = f(E[i] \cdot B[j]) \quad (1)$$

kde $E[i]$ je aktuálne žiarenie i -teho pixelu, $B[j]$ je expozičný čas j -teho obrázka, a f je krivka odozvy kamery ktorá mapuje expozičné hodnoty na hodnoty intenzity pixla (v rozptati 0-255). Ak definujeme $g(x) = \ln f^{-1}(x)$, môžeme napísať

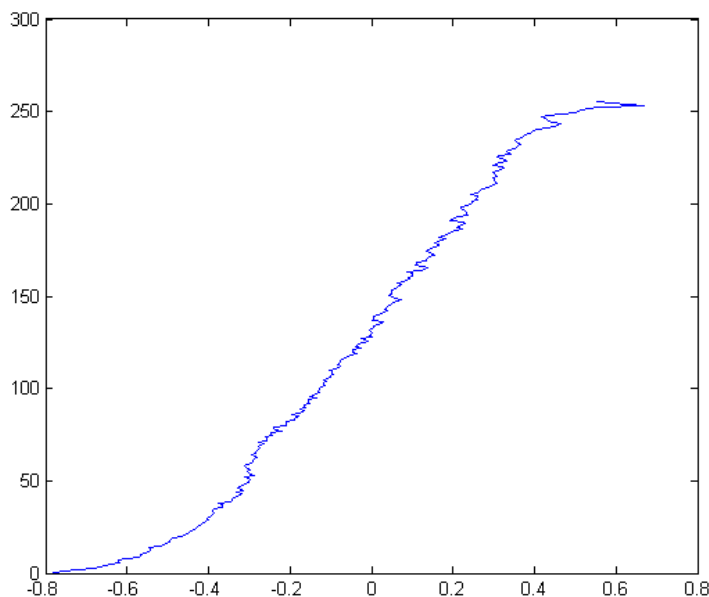
$$g(Z[i, j]) = \ln(E[i]) + \ln(B[j]) \quad (2)$$

Použijeme metódu Debevec et al. [1] na zistenie krivky $g(Z)$ pre hodnoty $Z \in [0, 255]$. Stiahnite si subor ([gsolve.m](https://github.com/gyrfalcon/Debevec)).

Rady:

1. Zvoľte si regularizačný parameter $\lambda \in [0.1, 5]$. Sledujte ako sa mení krivka zmenou parametra.
2. g vypočítajte pre všetky tri farebné kanály samostatne

4. Na výpočet nepoužívajte všetky pixeli!! Vyberte nahodne 1500 pixelov (Pozor! rovnakých v každom obrázku). Výpočet pre všetky body by trval dlho.
 5. Na výstupe dostanete dve hodnoty g a $\log$ žiarenia I/E pre každý pixel na vstupe.
 6. Zobrazte si funkciu g .
- ```
d=0:255;
plot(g,d)
```



## 2. Vytvorte mapu osvetlenia sceny

Ak máme vypočítanú krivku odozvy, môžeme vytvoriť mapu osvetlenia scény pomocou rovnice:

$$\ln(E[i]) = \frac{1}{P} \sum_{j=1}^P (g(Z[i,j]) - \ln(B[j])) \quad (3)$$

kde  $P$  je počet obrázkov s rôznymi expozíciami ktorý sme použili v prvej časti.

1. Implementujte pre každý farebný kanál samostatne
2. Na výstupe by ste mali dostať hdr obrázok (teda pole výšky a šírky obrázka krát 3 farebné kanály)
3. Pozor na to ze chcete hodnoty  $E[i]$  a nie  $\ln(E[i])$

### 3. Implementujte tone mapping

Aby sme vedeli zobrazit' obrázok na bežnom monitore musíme z neho spať vytvorit' LDR obrázok.

1. Vyskúšajte najskôr najjednoduchšiu metódu, zoškálovanie rozsahu pre každý farebný kanál, nasledovne

$$E_{norm}[i] = \frac{E[i] - E_{min}}{E_{max} - E_{min}} \quad (4)$$

kde  $E_{max}$  a  $E_{min}$  su maximum a minimum zo všetkých farebných kanálov. Ukážka:



2. Vytvorte LDR obrázok pomocou Durand02

"Fast Bilateral Filtering for the Display of High-Dynamic-Range Images"

Fredo Durand and Julie Dorsey, ACM SIGGRAPH 2002

(Pozri prednasku 7 slide 25)

pomôcky:

RGB -> LAB

použite bilaterálny filter

<http://www.mathworks.de/matlabcentral/fileexchange/12191-bilateral-filtering/content/Bilateral%20Filtering/bfilter2.m>

$C = L;$

Cbase je L filtrovane bilaterálnym filtrom.

vyskusajte

$C_{detail} = C - C_{base};$

$C_{detail} = C ./ C_{base};$

na zníženie kontrastu použite napr. Reinharda:

$lumOriginal = C_{base};$

$\epsilon = 0.00001;$

$logImg = \log(lumOriginal + \epsilon);$

$logAvgLum = \exp(\text{mean}(\text{mean}(logImg)));$

$lumKeyMapped = (0.18 ./ logAvgLum) .* lumOriginal;$

$C_{base} = lumKeyMapped ./ (1.0 + lumKeyMapped);$

LAB->RGB

Odovzdávanie:

Vytvorte GUI, kde sa budú zadávať LDR obrázky. Hodnoty expozície sa môžu zadávať/alebo získavať z obrázkou automaticky.

Vypočíta a vykreslí sa krivka odozvy, vytvorí sa HDR obrazok. Potom sa pomocou tone mapping vytvorí 2 LDR obrázky (škálovanie, Durand) a zobrazia sa.

Odfodíte si aspoň jednu scénu s aspoň 3 rôznymi expozíciami a priložte ich k riešeniu.