

Semestrální projekt předmětu Výpočetní fotograbie – závěrečná zpráva
Tone Mapping: implementace algoritmu Khan20

Milan Tichavský, xticha09@fit.vut.cz

1 Úvod

Zařízení běžně používaná k zobrazování digitálního obsahu, jako jsou monitory, televizory a tiskárny, nejsou schopna zobrazit celý dynamický rozsah světla přítomného v reálném světě. Z tohoto důvodu existují metody pro převod obrazu s vysokým dynamickým rozsahem (HDR) na standardní dynamický rozsah (SDR), aby bylo možné jej správně vizualizovat na dostupných zařízeních. Tento proces se nazývá "tone mapping".

Cílem této práce je implementace a analýza jednoho z moderních tone-mapping algoritmů, konkrétně metody Khan20 [1], jako pluginu do Tone Mapping Studio (TMS) [2].

2 Teoretické základy

Pro pochopení tone-mappingu je nutné zmínit vlastnosti lidského vidění. Lidské oko dokáže adaptivně vnímat velký rozsah jasových hodnot díky nelineárnímu vnímání jasu a kontrastu. Algoritmy tone-mappingu se často inspirují těmito vlastnostmi a snaží se přizpůsobit zobrazení tak, aby výsledný obraz působil přirozeně.

3 Popis algoritmu Khan20

Algoritmus Khan20 je globální tone-mapping operátor (TMO), to znamená že pro transformaci je využita monotónní neklesající mapovací funkce. Algoritmus je založený na transformaci obrazu pomocí funkce *Perceptual Quantizer (PQ)* a histogramu jasu. Schéma algoritmu je vidět na obrázku 1.

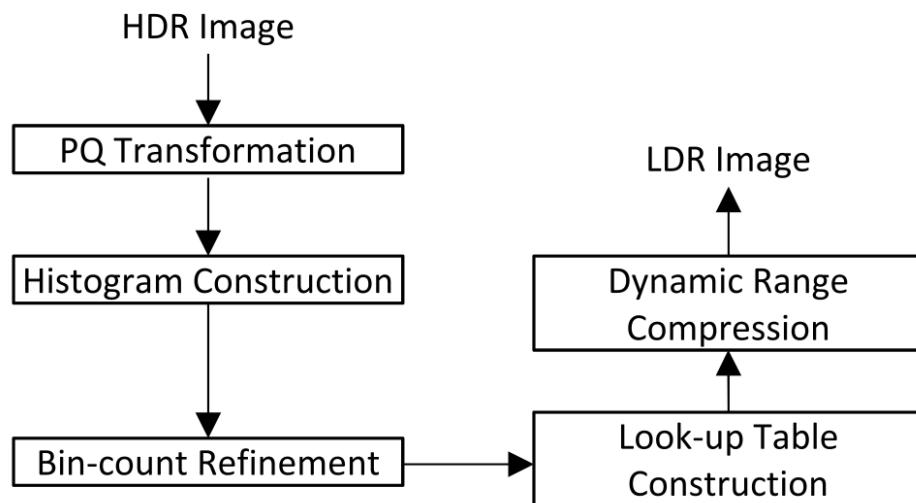
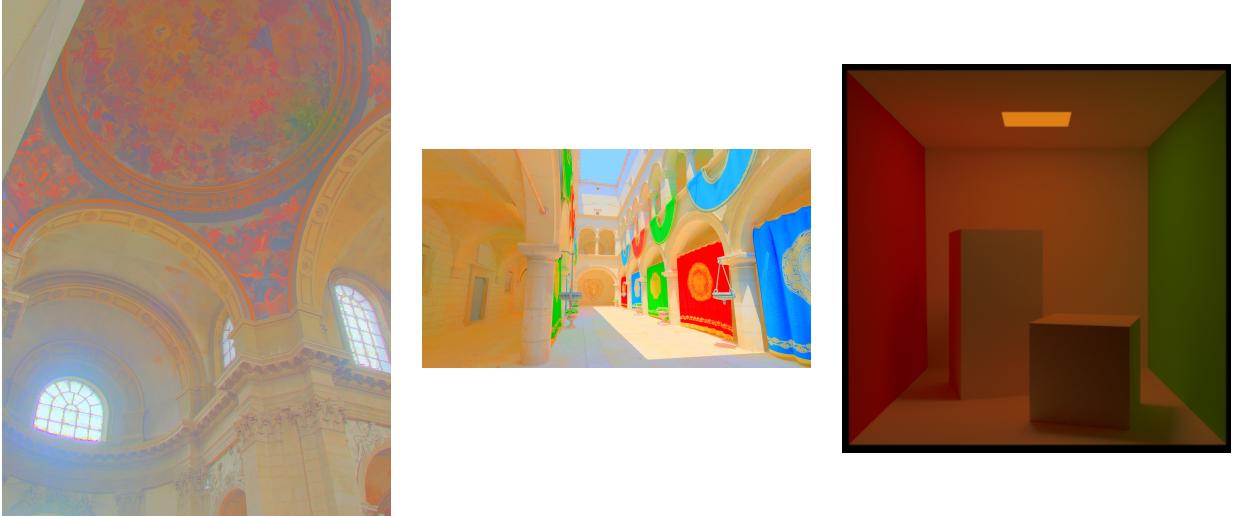


Figure 1: Schéma algoritmu převzato z článku Khan20 [1].

Funkce PQ vychází se snaží simulovat vnímaní lidského oka a transformuje vstupní jas v rozsahu $[0, 10000] \text{ cd/m}^2$ na normalizovaný rozsah $[0, 1]$, přičemž nižší jasové hodnoty mají k dispozici větší relativní rozlišení.

Jak je vidět v tabulce 1, z takto transformovaného obrázku lze jasně vidět celou scénu, která ale působí vybledle a má relativně nízký kontrast.

Table 1: Obrázky po transformaci funkcí *Perceptual Quantizer*.



Z toho důvodu je nutné zkonstruovat histogram jasů, ze které je vytvořena vyhledávací tabulka, podle které se provádí mapování každého vstupního pixelu. Počet sloupců je dán parametrem N (v příkazové řádce TMS lze zadat přepínačem `-Bins`), přičemž implicitní hodnota je 256.

Už dříve bylo ukázáno, že použití klasického histogramu je problematické, protože se může stát že malým oblastem bude přiřazeno až moc málo výsledného rozsahu. Například pokud máme čistě modrou oblohu zabírající velkou část snímku, nechceme, aby zabírala půlku výsledného rozsahu jasu. Proto je nutné omezit počet pixelů v každém *binu* pomocí parametru k , kde $k/N * pixel_count$ je maximální počet pixelů v daném sloupci. S pixelama které se do daného počtu nevlezou se nic neděje, akorát pro další výpočty je počet pixelů v daném sloupci podhodnocen. V TMS lze nastavit přepínačem `-Truncation` s implicitní hodnotou 5.

V článku samotném není přestě uvedeno, jakým způsobem se má obraz pomocí takto transformovaných úrovní jasu převést do výsledného souboru. Proto jsem vstupní obrázek převedl do Yxy barevného prostoru, nahradil jas (luminance) nově vypočtenými hodnotami a tento obrázek uložil jako výsledek tone-mappingu.

4 Výsledky

V tabulce 2 je vidět porovnání různých metod. Zvolil jsem ty metody, které ve výchozím nastavení byly schopny rozumně zobrazit všechny tři obrázky.

As it can be seen, on all testing images, the scene and all details are very clearly visible. I think the colors are sometimes bit too light and for more subjectively better looking images I'd stick with Drago03, as I like the darker and maybe a bit more saturated look of them. On the other hand, in terms of showing the details in darker regions I think this method is better. Also, I really like I don't have to tune parameters of the algorithm for each scene, because the defaults work across different scenes just fine.

Výpočetní složitost implementace je relativně nízká, program projde celkem 4x přes všechny pixely obrázku a provede relativně jednoduché výpočty v každém kroku. Počet průchodů by šel snížit, nicméně takto je kód podle mého názoru přehlednější, kdy se každá funkce mapuje na jednu fázi algoritmu.

4.1 Uživatelská studie

Poznatky získané testováním.

5 Závěr

References

- [1] KHAN, I. R., AZIZ, W., AND SHIM, S.-O. Tone-mapping using perceptual-quantizer and image histogram. *IEEE Access* 8 (2020), 31350–31358.
- [2] ČADÍK, M. Tone mapping studio. GitHub repository, 2025. Accessed: 2025-03-24.

Table 2: Srovnání různých metod; ve sloupcích zleva do prava jsou Khan20, Drago03 a Ward94. Všechny s výchozíma hodnotama parametrů v Tone Mapping Studiu.

