



Zpracování obrazu (ZPO)

Program pro zobrazování vlastností obrazu

1 Popis varianty

Jedná se o vypracování programu, který je schopen zobrazit různé údaje o načteném obrazu, jako hodnota pixelu pod myším kursorom (B/W nebo RGB s čísly), jasový řez obrazu v místě kursoru apod. Prosím, modifikujte podle svých schopností a zálib. Výsledkem by měl být program s popisem funkcí.

1.1 Složení týmu

Na projektu pracuji individuálně.

1.2 Upřesnění varianty

Projekt nabízí poměrně vysokou volnost, co se týče výsledných vlastností (to se bohužel časem ukázalo jako jeho největší nevýhoda). Na základě mnou využívaných funkcí u podobného softwaru jsem se rozhodl pro následující vlastnosti/schopnosti:

- Jednoduché úpravy obrazu (rotace, zvětšení, převod do šedé)
- Zobrazení EXIF informací
- Získání hodnoty z pixelu na daných souřadnicích (zadání souřadnicemi i kliknutím myši) *Nakonec neimplementováno*
- Výpočet histogramu a detekce správné expozice
- Množství barev v obraze a jejich detekce
- Možnost spustit filtr s libovolným jádrem
- Prahování obrazu (i v RGB prostoru) s možností vykreslit hodnoty pouze pod/nad prahem
- Detekce a odstranění šumu

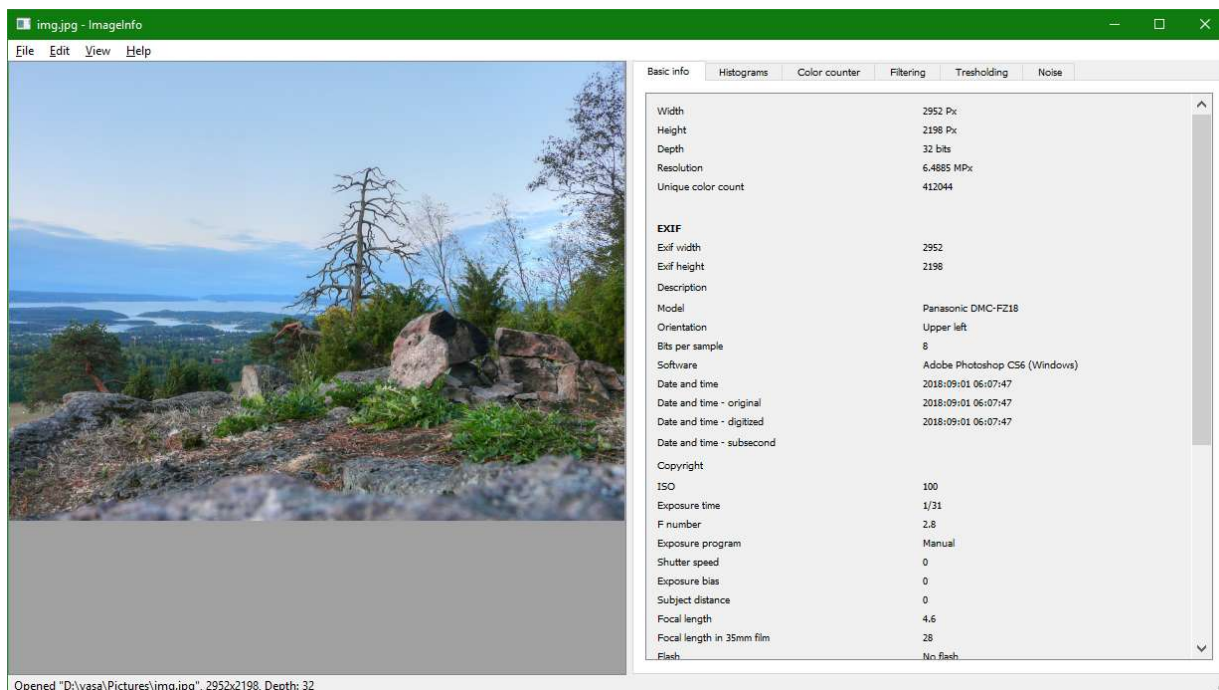
2 Návrh GUI

Hlavní okno programu je rozděleno na dvě části. V levém pod-okně je zobrazen aktuálně načtený obrázek, popř. výsledek spuštěné operace. V pravém jsou pak zobrazovány jednotlivé informace a ovládací prvky, které jsou členěny do záložek pro větší přehlednost.

2.1 Ovládání

Základní úpravy obrázku (zvětšení rotace, obnovení původních hodnot) je řešeno primárně pomocí klávesových zkratk. Jednotlivé ovládací prvky jsou umístěny v menu. Do budoucna by bylo vhodné je umístit na *toolbar*, ale to je spíše otázka GUI, což je mimo zaměření tohoto předmětu.

Spouštění diagnostických funkcí probíhá pomocí myši, přičemž ovládací prvky jsou umístěny na jednotlivých záložkách. I zde platí poznámka o ne zcela dokonalém GUI...



Obrázek 1: Ukázka základního okna

3 Technologie

Program je vytvořen pomocí frameworku Qt. Ten je využit nejen k vytvoření GUI, ale i k samotné práci s obrázkem (získání hodnoty pixelu a základní transformace). Pokročilejší úpravy obrazu, jakými jsou prahování či spouštění filtrů jsem implementoval sám. Pro zvýšení výkonu by bylo zřejmě vhodnější použít *OpenCV* knihovnu. Pro načtení EXIF informací je použita jednoduchá knihovna *easyexif*¹.

4 Studium

Toto konkrétní zadání nevyžaduje dodatečné studium nad rámec přednášek. Většina nastudovaných informací se týká samotného frameworku, přičemž zdrojem byla oficiální dokumentace.

Pro detekci správné expozice na základě histogramu jsem vycházel z vlastních zkušeností. Detekci šumu jsem nejdříve navrhl jako porovnání zašuměného a odšuměného obrázku (viz podkapitola ??). Tento předpoklad mi pak potvrdily i jednotlivé odpovědi na obdobnou otázku z internetových diskuzních fór.

5 Podrobnější popis řešení

Níže uvádím podrobnější popis řešení jednotlivých funkcí výsledného programu.

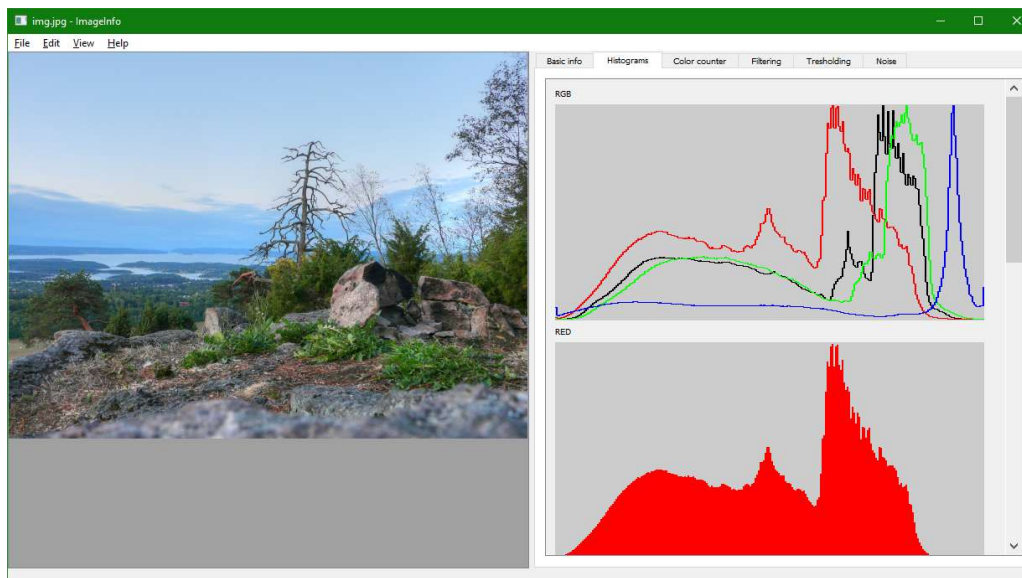
5.1 Základní informace

Zde asi není potřeba podrobnějšího popisu. Jsou zde vypsány základní informace jako rozlišení, počet barev, apod. Program dále otestuje přítomnost EXIF informací a poté vypíše všechny dostupné hodnoty.

¹<https://github.com/mayanklahiri/easyexif>

5.2 Histogramy

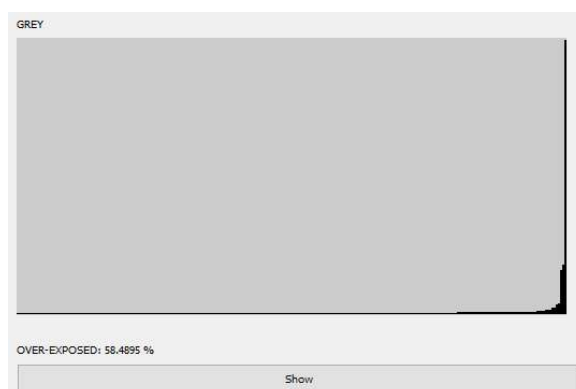
Po načtení obrázku jsou spočteny histogramy pro každou jeho barevnou složku. Posléze je obrázek převeden do odstínů šedi a spočte se histogram i pro jeho jas. Hodnoty těchto histogramů jsou vykresleny jako křivky do jednoho společného obrázku i samostatně.



Obrázek 2: Ukázka histogramů

Pod histogramy je umístěna informace, zda byl obrázek (respektive fotografie, u obrázků tato funkce postrádá smysl) správně exponován. Správně exponovaný obrázek má extrém histogramu okolo jeho středu (v závislosti na fotografované scéně). Např. na obrázku 2 je vrchol křivky situován více vpravo, což značí více světla v obraze (důvodem je v tomto konkrétním případě obloha), ale nedochází k tzv. *přepalům*².

Pokud by byl snímek podexponovaný, popř. přexponovaný, tak by extrém histogramu ležel na jeho okraji, jak můžeme vidět na obrázku 3. V rámci programu je krajní hranice široká 10 bodů (to znamená hodnoty 0-9, popř. 246-255), přičemž míra pro špatně exponovaný obrázek je 50 %, tedy pokud se 50 % všech pixelů na histogramu vyskytuje na jeho okraji, program daný snímek vyhodnotí jako špatně exponovaný. Volitelně si může uživatel vizualizovat tato místa.



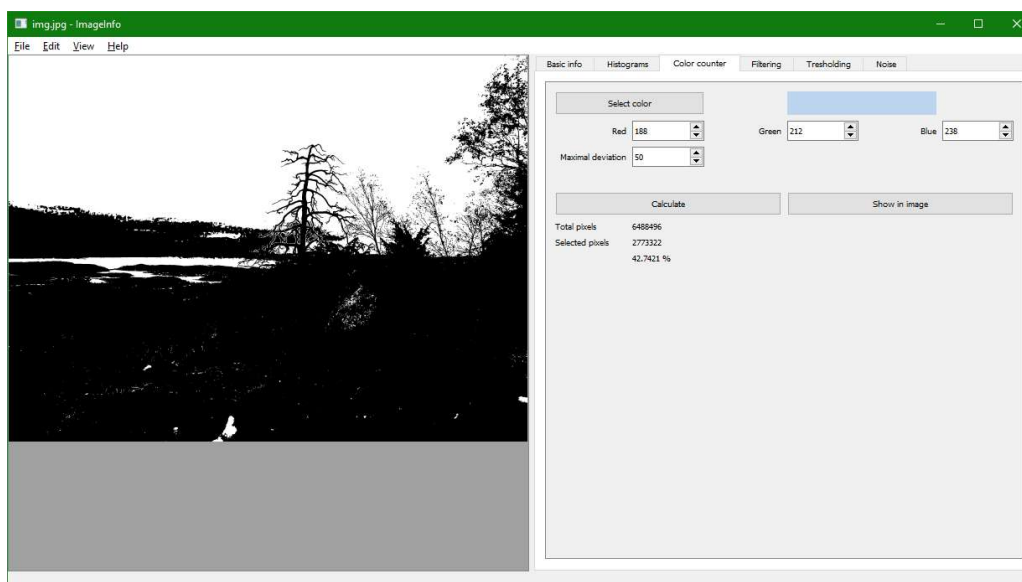
Obrázek 3: Přexponovaný snímek

²Přepalem se rozumí místo, na které dopadlo tak velké množství světla, že jej nebyl senzor schopný zpracovat. Výsledný obrázek proto v dané oblasti obsahuje hodnoty blízké maximálním.

5.3 Zastoupení barev v obraze

Tato záložka slouží k zobrazení procentuálního a absolutního zastoupení různých barev v obraze. Tuto funkci lze zkombinovat s jinými funkcemi. Např. s výše zmíněnou vizualizací přepalů, lze snadno zjistit, kolik procent obrazu bylo špatnou expozicí ztraceno.

Druhým možným využitím je statistické vyhodnocení obsahu. Na obrázku 4 je graficky znázorněna obloha (bílá plocha), která zde zabírá přes 40 % plochy snímku. Pokud by snímek zachycoval rozkvetlou louku, bylo by možné pomocí množství žluté na fotografii určit např. počet pampelišek na fotce...



Obrázek 4: Možnost prahování konkrétní barvy

5.4 Filtrování

Zde se nachází možnost spustit libovolný filtr nad obrazem. Tato schopnost s tématem sice přímo nesouvisí, ale filtrovací funkce byla implementována z jiných důvodů, proto by bylo škoda ji nevyužít.

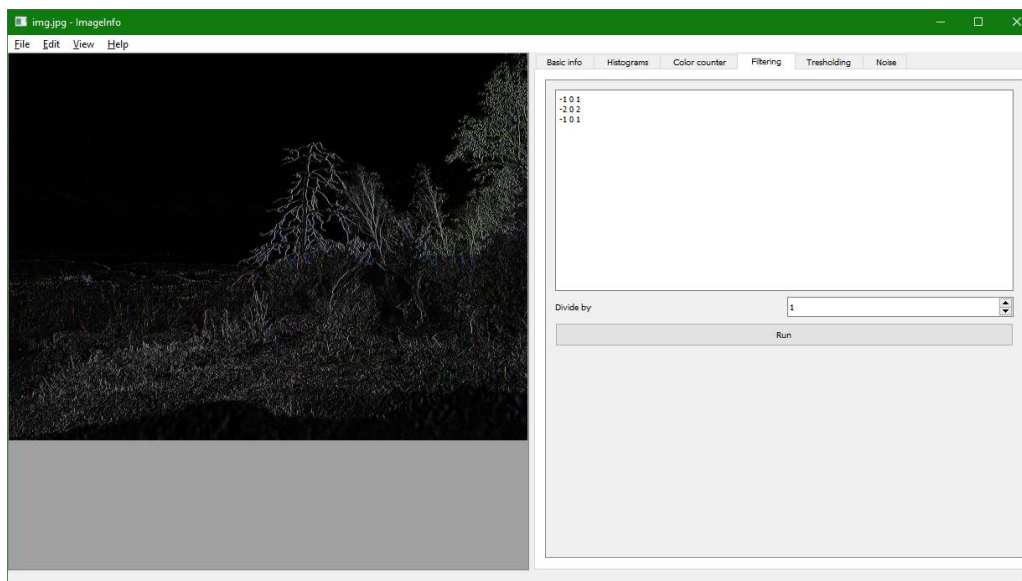
5.5 Prahování

Obdobně jako v případě filtrování se jedná spíše o možnost využít implementované funkce. Jedná se o podobnou schopnost jako v případě počítání barev, s tím rozdílem, že zde nejsou prahové hodnoty v daném rozmezí, nýbrž všechny hodnoty větší než zadaná. Ukázka použití se nachází na obrázku 6.

5.6 Šum

Poslední záložka obsahuje možnost odstranit šum z obrazu. K tomu je použit Gaussův filtr o velikosti 9 ($\sigma = 5$). Výsledný obrázek je porovnán s původním a na základě této hodnoty je určeno množství šumu v obraze. Ukázka použití na obrázku 7.

Myslím si, že se jedná o zajímavé téma, kterému by bylo vhodné věnovat více prostoru, bohužel mě tato možnost napadla až poslední týden před odevzdáním a tudíž jsem neměl dostatek času tuto funkci řádně otestovat a odladit.



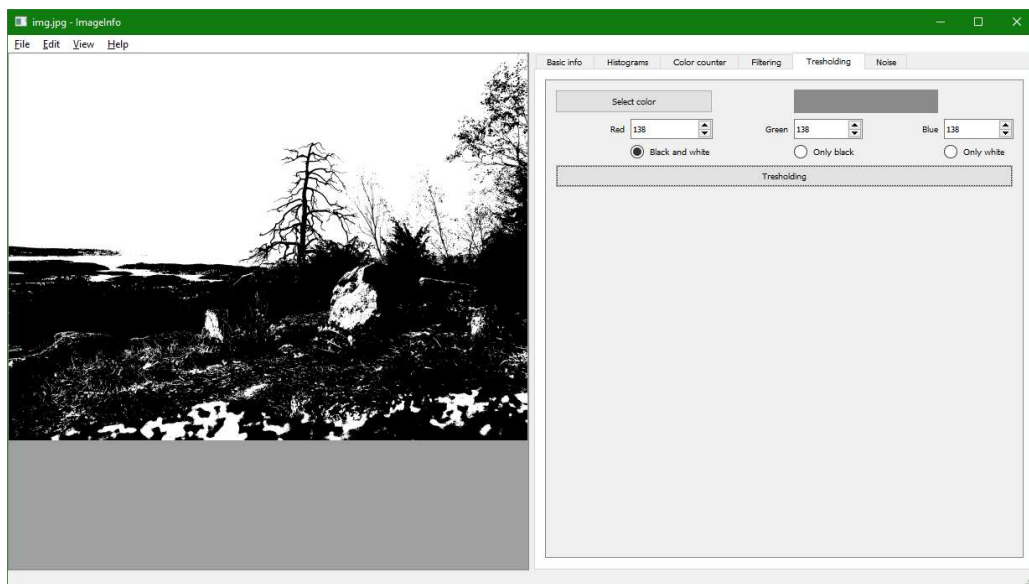
Obrázek 5: Ukázka při použití se Sobelovým operátorem

6 Testování

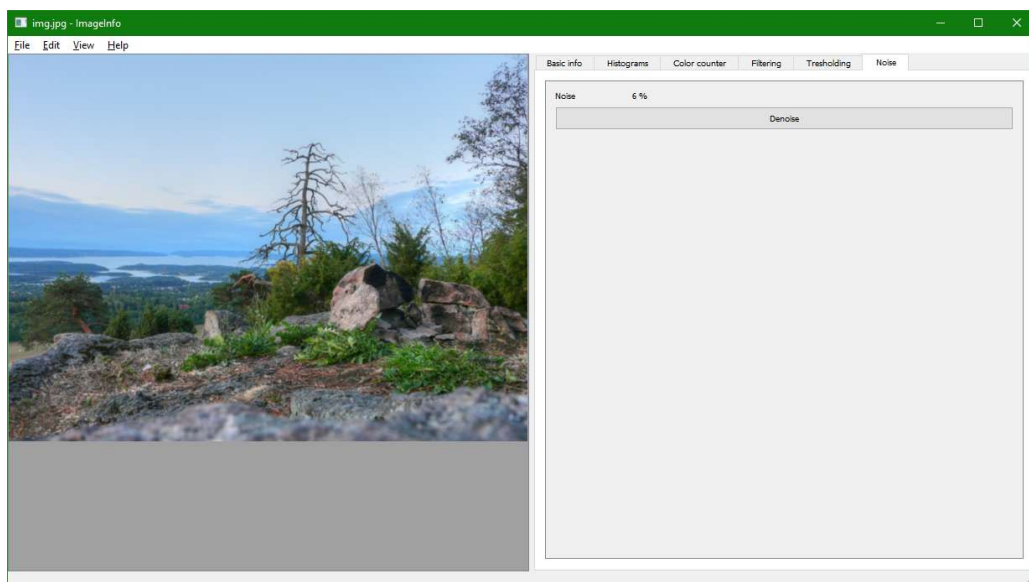
K testování jsem používal běžné fotky. Dále jsem vyfotil několik záměrně špatných fotografií (zašuměných, přepálených apod.) abych mohl otestovat všechny funkce. Výsledné hodnoty jsem porovnával vůči ostatním grafickým programům, popř. vůči očekávaní.

7 Závěr

Výsledkem je funkční prototyp programu, který je schopen o obrázku získat různé informace a dále s nimi pracovat. Mezi zajímavé funkce bych zařadil hodnocení expozice na základě histogramu či detekci množství šumu.



Obrázek 6: Ukázka jednoduchého prahování



Obrázek 7: Ukázka detekce a odstranění šumu