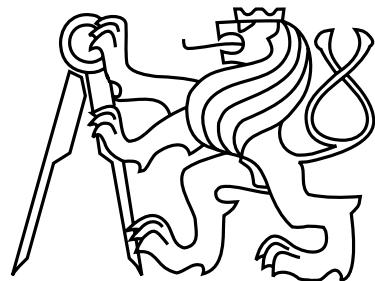


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce
Galerie fotografií s exportem na webové stránky

Pavel Macenauer

Vedoucí práce: Doc. Ing. Jan Janoušek Ph.D.

Studijní program: Elektrotechnika a informatika, dobíhající, Bakalářský
Obor: Výpočetní technika

4. ledna 2012

Poděkování

Chtěl bych poděkovat vedoucímu práce doc. Ing. Janu Janouškovi Ph.D. za podporu a možnost pracovat na bakalářské práci s tématikou zpracování obrazu, která mne velmi zajímá.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 4. 1. 2012

.....

Abstract

The subject of this thesis is GalleryManager, simple image editor and photo-organizer. This cross-platform graphical application organizes pictures into folders called galleries including their full description. Graphic editor allows to carry out simple image editing: rotation-resizing, changes of brightness, contrast and saturation and conversion of coloured images into black and white. Individual galleries and photos can be documented not only by adding names and descriptions, but also with embedded image metadata. Output from the program is a web presentation.

Focus was given on speed of image processing and possibility to expand the application with additional functionalities. Program supports images in JPEG format and 8-bit colour depth.

Abstrakt

Předmětem práce je vytvoření programu GalleryManager pro správu, úpravu a export fotografií. Jedná se o multiplatformní grafickou aplikaci, která slouží k organizaci fotografií do složek označovaných jako galerie a jejich správě. Obsahuje jednoduchý grafický editor umožňující základní obrazové úpravy. Jmenovitě rotaci, zvětšování/zmenšování, barevné úpravy jasu, kontrastu, sytosti a převod barevného obrazu na černobílý. Jednotlivé galerie a fotografie lze dokumentovat a to nejen za pomoci jmenovek a popisků, ale načítají se i obrazová metadata. Výstupem z programu je webová prezentace se zpracovanými fotogaleriemi.

Důraz při vývoji aplikace byl kladen na rychlosť zpracování obrazu a možnosť rozšiřovať program o další funkce. Zpracování fotografií probíhá z formátu JPEG v 8-bitové barevné hloubce.

Obsah

1	Úvod	1
2	Popis problému, specifikace cíle	3
2.1	Cíle práce	3
2.2	Podoba práce vzhledem k vytyčeným cílům	3
2.3	Existující implementace	4
2.3.1	Komerční software	4
2.3.2	Nekomerční software	5
3	Analýza a návrh řešení	7
3.1	Volba programovacích jazyků a knihoven	7
3.1.1	Grafické uživatelské rozhraní	7
3.1.2	Zpracování obrazu	7
3.1.3	Čtení EXIF údajů	8
3.1.4	Výstup z programu	8
3.2	Prostředí pro vývoj aplikace	9
3.3	Správa a záloha kódu	9
3.4	Struktura ukládaných informací	9
3.5	Zobrazování obsahu galerií a návrh GUI	9
3.6	Dostupné grafické úpravy a práce s pamětí	10
4	Realizace	13
4.1	Vnitřní informační struktura	13
4.2	Grafické uživatelské rozhraní	13
4.3	EXIF údaje fotografií	14
4.4	Úprava fotografií	14
4.4.1	Kontrast	15
4.4.2	Jas	15
4.4.3	Sytost a převod do černobílé	16
4.4.4	Rotace a transformace	17
5	Testování	19
5.1	Porovnání úprav s programem Adobe Photoshop	19
5.1.1	Převod do černobílé a vykreslování obrazu	19
5.1.2	Úprava kontrastu	20
5.1.3	Úprava sytosti	21

5.2	Testování exportu webové prezentace	22
6	Závěr	23
6.1	Zhodnocení splnění cílů	23
6.2	Návrhy na zlepšení a cíle do budoucnosti	24
Literatura		25
A	Seznam použitých zkratek	27
B	Instalační a uživatelská příručka	29
C	Obsah přiloženého CD	31

Seznam obrázků

3.1	Grafické uživatelské rozhraní při zobrazení fotografie a při zobrazení galerie	10
4.1	Hodnoty HS barevného prostoru HSV	16
5.1	Převod do černobílé a vykreslování obrazu	19
5.2	Úprava kontrastu	20
5.3	Úprava sytosti	21

Seznam tabulek

4.1 Struktura EXIF dat	14
------------------------	----

Kapitola 1

Úvod

V současné době je na trhu velké množství grafických editorů, programů pro správu fotografií a těch, které kombinují obojí. Z komerčních produktů jsou nejznámější rozsáhlé programové balíky Photoshop a Lightroom od Adobe Systems Inc. V České republice se velmi dobře prosadil i mnohem levnější Zoner Photo Studio. Bezplatně je možné využívat služeb Google Picassa nebo IrfanView. Komerční produkty mají za sebou již mnoho let vývoje a týmy vývojářů.

Aplikaci jsem pojmenoval GalleryManager. Jedná se o jednoduchý grafický editor a správce fotografií s možností tvorby popisů. Aplikaci do budoucna plánuji integrovat do správy vlastních webových stránek, zaměřených na cestovatelskou fotografii PhotographicAtlas.com a FotoAtlas.cz. Není zde cílem implementovat pokročilé algoritmy pro úpravu fotografií, které již nabízejí existující softwary, zaměřené na práci s velkými soubory a tisk. Naopak je kladen důraz na rychlosť a efektivitu pro potřeby webové prezentace.

Dokumentární nebo reportážní fotografování, pro které je GalleryManager určen, nesprávují své fotografie jednotlivě, ale jako celé série. Cílem je umožnit rychlou práci s více fotografiemi současně a zpřístupnit i neobrazové informace vkládané fotoaparáty do fotografie (například údaje o nastavení fotoaparátu, místě a času pořízení snímku a další).

Aktuálně je aplikace navržena tak, aby výstupem byla nezávislá webová stránka, kterou lze použít pro vlastní prezentaci na internetu.

Tato práce je implementační, nikoliv rešeršní, proto dokumentace bude popsána především z programátorského hlediska, popisující postup při tvorbě aplikace a možnosti použitých knihoven. Vzhledem k tomu, že bych se i do budoucna chtěl věnovat úpravám obrazu a nadále rozšiřovat aplikaci o další možnosti, budou popsány jednotlivé algoritmy pro úpravu fotografií, jejich výhody, nevýhody a srovnání s jinými obecně známými.

Kapitola 2

Popis problému, specifikace cíle

2.1 Cíle práce

Hlavním cílem této práce je vytvořit grafický editor, který umožní organizovat, popisovat a nakonec exportovat fotografie do podoby webové prezentace. Nejedná se o práci, která bude prezentována jako hotový program, ale spíše jako framework pro správu a úpravy fotografií, který půjde dále rozšiřovat.

Zpracování obrazu a informací spojených s fotografií je natolik rozsáhlé téma, že přesahuje rozsah bakalářské práce, časové možnosti i znalosti jediného člověka. Cílem aplikace GalleryManager je vytvořit:

- grafické uživatelské rozhraní
- stromovou strukturu podsložek (galerií) a jednotlivých fotografií
- možnost každou galerii a fotografií popisovat, do budoucna by měla být struktura přizpůsobena, aby šla rozšiřovat
- automatické vkládání informací z EXIF údajů fotografií
- základní grafické úpravy - jas, kontrast, sytost, transformace obrazu, ...
- export projektu do standardních formátů - XML, HTML/CSS a možnost ho ukládat a opět načítat
- multiplatformní podporu aplikace

2.2 Podoba práce vzhledem k vytyčeným cílům

Jedná se o grafickou aplikaci psanou v jazyce C++. Průběh práce s programem, při zpracování a organizaci vlastních fotografií vypadá takto:

1. po spuštění programu GalleryManager se spustí prázdný projekt bez jakýchkoliv podsložek

2. uživatel má možnost nahrát projekt, který vytvořil v minulosti, včetně všech podsložek a fotografií
3. v rámci projektu lze vytvářet nové podsložky (galerie) nebo upravovat již existující
4. do existujících nebo nově vytvořených podsložek nainstaluje uživatel nové fotografie
 - během importu fotografií se načtou a zobrazí EXIF informace
 - je možné přidávat další popisky
 - při rozkliknutí složky se zobrazí její obsah - podsložky jako obálky a fotografie jako malé náhledy
 - při rozkliknutí fotografie se zobrazí fotografie v plném rozlišení v prostředí pro úpravu
5. projekt lze ukládat ve formátu XML
6. projekt lze exportovat ve formátu HTML do podoby webové prezentace, přičemž vzhled se určí pomocí vložené CSS šablony
7. CSS lze libovolně vytvářet na základě popsáne HTML struktury, a tak vkládat vlastní design

2.3 Existující implementace

Rešerši existujících produktů jsem prováděl v průběhu listopadu a prosince 2011. Lze předpokládat, že se objeví další aplikace a existující produkty budou i nadále rozšiřovány do podoby dalších verzí.

Programy pro práci s fotografiemi lze rozdělit na aplikace pro správu fotografií a grafické editory. Společností, která vládne v komerční oblasti, je Adobe Systems Inc. Jako multifunkční grafický editor nabízí tato firma Photoshop, jeho odlehčenou verzi Photoshop Elements a balík programů Lightroom, určený téměř výhradně pro fotografy. V České Republice do cenově nižší kategorie zapadá Zoner Photo Studio. Na rozdíl od komerčních aplikací, které se zabývají především pokročilými úpravami obrazu, se nekomerční soustřeďují spíše na organizaci a prohlížení fotografií. K nim lze zařadit Google Picassa, IrfanView a software, přiložený ke správě uživatelského účtu na Rajce.cz. Žádný z těchto programů neposkytuje své zdrojové kódy, proto jediným zdrojem informací jsou uživatelské zkušenosti se samotnými aplikacemi. Zároveň neexistuje ani žádné API, které by umožnilo používat jejich funkce pro vlastní účely.

2.3.1 Komerční software

Dnes nejrozsáhlejší softwarový balík Photoshop od společnosti Adobe Systems Inc., prošel za posledních deset let dlouhým vývojem. Umožňuje provádět složité grafické úpravy pomocí práce s vrstvami obrazu, maskami a to nejen výběrovými, ale i kanálovými¹. Na začátku

¹Obraz ve Photoshopu si lze představit jako sendvič složený z jednotlivých obrazových vrstev, které se navzájem prolínají s různou intenzitou, kterou si volí uživatel. Masky, na druhou stranu, jsou uložené výběry, se kterými lze pracovat. Lze takto maskovat nejen plochy, ale i celé barevné kanály

vývoje zpracovával soubory pouze v 8-bitové barevné hloubce. Dnes, hlavně na základě požadavků z oblasti videa, umožňuje práci se soubory ve 32-bitové barevné hloubce (především pro technologii HDR²). Mimo jiné existuje i jeho odlehčená varianta Photoshop Elements, která neobsahuje tak mnoho specializovaných funkcí pro úpravu obrazu.

Adobe Lightroom je zaměřen na hromadné úpravy fotografických souborů, a to především ve formátech RAW³. Jeho jedinečnost spočívá ve vytváření celých sad grafických úprav, tedy maker, nad více fotografiemi současně. Výhodou je, že se uloží pouze jednotlivá makra úprav a nemění se tak originální soubor. Ten se změní až v okamžiku exportu dat mimo prostředí Lightroom. Umožňuje pouze základní grafické úpravy, v případě složitějších je nuten uživatel přejít do Photoshopu. Nevýhodou je, že jej nelze integrovat do vlastních projektů a napojit ho na vlastní webový server, k čemuž je GalleryManager do budoucna zamýšlen.

Jako most mezi aplikacemi Adobe slouží Bridge. Dovoluje zobrazovat nejrůznější grafické formáty, včetně jejich organizace. Tento software umožňuje velmi rychle generovat náhledy snímků, někdy na úkor velkého využití místa na disku. Spolupracuje nejen s aplikacemi pro práci s obrazem, tedy s Photoshopem a Lightroomem, ale i s dalšími balíky určenými pro zpracování videa (Premiere, AfterEffects) nebo vektorové grafiky (InDesign, Illustrator).

Především v České republice je oblíbený Zoner Photo Studio, který se soustředí na to samé jako Adobe Photoshop, tedy grafickou úpravu jednotlivých fotografií. Zdaleka však neobsahuje tolik funkcí. Pro uživatele je atraktivní svou cenou, jednoduchostí a zázemím v České republice.

2.3.2 Nekomerční software

Software nabízený zdarma slouží obvykle jako podpora nebo nadstavba služeb poskytovaných webovými servery. Česká webová stránka rajce.cz umožňuje uživatelům sdílet své fotografie v podobě galerií. Poskytuje software, pomocí kterého si uživatel vytvoří galerii, pojmenuje ji, popíše a na svém počítači vybere fotografie, které do ní chce vložit. Zmenšení na náhledy, pro potřebu webové prezentace na www.rajce.cz, provede program automaticky při následném nahrávání. Z grafických úprav lze fotografie otočit o 90 stupňů a pojmenovat. Zatím není možné je jakkoliv upravovat nebo číst jejich EXIF údaje v širším měřítku.

IrfanView je naopak spíše jednoduchým prohlížečem fotografií. Umožňuje, oproti komerčnímu softwaru, pouze základní grafické úpravy. Jeho hlavní předností a záměrem je zobrazovat fotografie, proto podporuje většinu dnes používaných grafických formátů a umožňuje spustit automatické prohlížení v podobě prezentace.

Svou koncepcí je GalleryManager asi nejpodobnější produktu Google Picasa, který je se spuštěním sociální sítě Google+ využíván pro synchronizaci fotografií s vlastním uživatelským účtem, ale lze jej používat i samostatně pro grafické úpravy. Galerie se zde vytvářejí přímo v podobě systémových složek, které si Picasa neustále za svého běhu indexuje. Uživatel tak má přehled o všech svých fotografiích na pevném disku. Zajímavé je, že každou chvíli

²High Dynamic Range je technologie umožňující skládat snímky s rozdílnými hodnotami expozice. Jejím praktickým uplatněním je například noční fotografie, kdy chceme zachytit jak kresbu ve světlech, tak i ve tmě.

³Bezkompresní obrazový formát, kterého využívají moderní digitální zrcadlovky k ukládání fotografií v maximálním kvalitě. Je specifický pro každého výrobce.

si určitou část využívané paměti uvolní a nezabírá tak velké množství paměti, jaké by člověk očekával od programu, který si neustále vytváří náhledy.

Sociální sítě dnes slouží především k masovému sdílení vlastních informací. Stejným způsobem pohlíží i Picasa na fotografie. Je přizpůsobena běžnému uživateli, kterému stačí jednoduché úpravy fotografií. Soubory lze synchronizovat s Google+ a prohlížet téměř každý dnes používaný formát. Neumožňuje však načítat všechny informace EXIF. A to především takové, které jsou pro fotografy důležité a dokumentovat tak dostatečně své galerie.

Kapitola 3

Analýza a návrh řešení

3.1 Volba programovacích jazyků a knihoven

GalleryManager je prohlížeč fotografií s jednoduchým grafickým editorem. Je navržen tak, aby mohl upravovat a obsluhovat velké množství fotografií paralelně. Potřebujeme tak co nejrychlejší jazyk s velkým množstvím dostupných knihoven, především pro tvorbu GUI, úpravu obrazu a čtení EXIF dat. Zároveň musí být objektově orientovaný a mapovat tak koncept reálného světa. Pro uživatele je důležité vytvářet stromovou strukturu galerií a fotografií o neomezené hloubce, kde každá galerie může v sobě obsahovat opět další galerie a fotografie.

Zvolen byl programovací jazyk C++, který všechny tyto požadavky splňuje. Jeho nevýhodou je, že oproti výše-úrovňovým jazykům jako Java, má zde programátor na starosti i správu paměti a musí velké množství funkcí, které by jinak byly implementovány v základních knihovnách, psát sám.

Jednotlivé použité knihovny je třeba analyzovat po částech aplikace, kde každá by měla být na druhé co nejméně závislá. Nechceme, aby docházelo k tomu, že při změně části kódu bude třeba přepisovat zbytek aplikace.

3.1.1 Grafické uživatelské rozhraní

K tvorbě grafického uživatelského rozhraní je použit široce rozšířený framework Qt od Nokie. K jeho výběru vedlo hned několik výhod. Je volně ke stažení pod licencí LGPL v2.1, umožňující volné používání pro nekomerční účely. Z programátorského hlediska má přehlednou dokumentaci, je multiplatformní, tudíž GUI a signály nejsou vázány na operační systém a umožňuje velké množství dalších funkcí spojených nejen s úpravou obrazu a GUI. Jeden z problémů, který musíme řešit, je vykreslování obrazu po provedení úpravy. Qt umožňuje vytvořit prázdný obraz a vykreslovat do něho pixel po pixelu, kterému se zadají pouze barevné informace.

3.1.2 Zpracování obrazu

Knihovny Qt umožňují pouze jednoduché možnosti zpracování obrazu. Jeden ze záměrů je možnost rozšířit možnosti GalleryManageru jakožto grafického editoru. Především tak

pracovat v budoucnosti i ve více bitových barevných hloubkách, kromě 8-bitových i 16-bitových a v jiných barevných prostorech, nejen RGB, ale i LAB¹ či CMYK. Tak, aby bylo možné zpracovávat fotografie stažené z profesionálních digitálních zrcadlovek. K načítání obrazových je použita opensource knihovna Boost::GIL od Adobe, která se využívá i v jejich komerčních produktech. To umožňuje daleko větší režii nad úpravou obrazu, než-li by nám poskytlo samotné Qt. Úpravy jsou pak z důvodu vzdělávacích účelů a požadavků na efektivitu implementovány pomocí adekvátních algoritmů za pomoci Boost::GIL a standardního C++.

3.1.3 Čtení EXIF údajů

Většina digitálních fotoaparátů nebo chytrých telefonů dnes ukládá metadata o pořízeném snímku, jakými jsou čas, clonové číslo, použitý fotoaparát, objektiv i jeho ohnisko, přímo do fotografie ve formátu EXIF. Kromě technických parametrů mezi nimi nemusí chybět ani GPS informace, má-li dané zařízení možnost určovat vlastní polohu. Uživatel ani programátor se tak nemusí starat o ruční vkládání dat, ale vše je získáváno přímo z fotografie.

K implementaci v prostředí C++ existují knihovny libexif a exiv2. Exiv2 je oproti libexif napsána v C++ a umožňuje číst i další standardy k uchovávání metadat - XMP² a IPTC³. Její nevýhodou je, že obsahuje závislosti na jiných knihovnách, jmenovitě zlib, gettext, libiconv a Expat, čímž značně přesahuje měřítko potřebné pro podporu metadat a zvětšuje i velikost celé práce.

Druhou knihovnou, napsanou v jazyce C, je libexif, kterou používáme i v této práci. Při integraci do moderního C++ prostředí toto vytváří úskalí především v konceptu C++, které už není zdaleka tak nízkoúrovňové jako C. Její výhodou je, že nepotřebuje žádné další závislosti. U GalleryManageru je požadavkem pouze zobrazení všech EXIF informací o fotografii, které by mohli uživatele zajímat. Knihovna je tak pro naše účely zcela dostačující.

3.1.4 Výstup z programu

Výstup z programu je ve dvou podobách. První z nich je ukládání samotného projektu do formátu, ze kterého půjde opět lehce načítat. K tomu je dnes hojně rozšířen jazyk XML. Výhodou využití frameworku Qt je, že v sobě zahrnuje i knihovny pro generování a čtení XML, pomocí kterých můžeme vygenerovat soubor obsahující všechny potřebné informace k uložení a zpětnému načtení celého projektu.

Druhou formou výstupu je vygenerování webové prezentace. HTML je velmi podobné XML, proto je použit opět stejný generátor, jako pro ukládání projektu, tentokrát pouze s HTML výstupem. Design prezentace je řešen už předem napsanými CSS styly, které se přiloží k webové stránce. CSS je zde zvoleno především z důvodu jeho rozšířenosti a nezávislosti na HTML kódu. Aplikace tak poskytne pouze kostru webové stránky, které se design vytvoří dodatečně. Součástí práce je jeden výchozí styl, další si mají možnost uživatelé vytvořit sami.

¹LAB je barevný prostor, kde L znamená lightness a AB jsou barevné složky obrazu. Jeho hlavní výhodou je možnost pracovat přímo se světlostí fotografie.

²Extensible Metadata Platform - standart od Adobe Systems Inc. k ukládání informací především v PDF, fotografiích, grafice

³Information Interchange Model od International Press Telecommunications Council je další ze standardů na ukládání dodatečných informací o souborech.

3.2 Prostředí pro vývoj aplikace

Ke tvorbě programu je použito volně dostupné vývojové prostředí Qt Creator, běžící na platformě Windows, které plně podporuje značení C++ syntaxe a čtení obsahu jednotlivých tříd. Navíc obsahuje i integrovanou dokumentaci frameworku Qt. K jeho volbě vedla nejen již zmíněná dokumentace frameworku Qt[8], ale i rychlost a jednoduchost používání, na rozdíl od běžně využívaného Microsoft Visual Studio, obsahujícího velké množství funkcí, které programátor nevyužije. Jako C++ kompilátor využívá Qt Creator systém MinGW.

3.3 Správa a záloha kódu

Při tvorbě větších softwarových projektů je třeba spravovat vlastní kód. Především z důvodu, aby bylo vidět, jaké dílčí úkoly už jsou hotové a jaké je třeba stále dodělat, se založil veřejný repositář. Z důvodu jednoduché instalace, kvalitního zázemí internetového serveru github.com a možnosti založit si zdarma účet, byl zvolen revizní systém Git[13] s grafickou nadstavbou pro Microsoft Windows GitTortoise⁴. Zdrojové kódy jsou volně k dispozici, aby nenarušovaly licenční podmínky používaných knihoven. Zároveň slouží repositář i k záloze dat v případě, že by došlo k jejich ztrátě v průběhu vypracovávání bakalářské práce.

3.4 Struktura ukládaných informací

Základní strukturou celého programu je projekt. Ten v sobě následně obsahuje libovolné množství podsložek, tj. galerií, které v sobě mohou obsahovat ještě další podsložky nebo přímo fotografie. Byla tak vytvořena stromová struktura galerií a fotografií o neomezené hloubce. Podmínkou je, že každá fotografia musí být součástí nějaké galerie.

Každá galerie nebo i fotografia o sobě obsahuje popisné informace, které se získávají ze dvou zdrojů. Prvním je pojmenování a popsání každé fotografie či galerie při jejím vytvoření nebo importu. Dalším zdrojem informací jsou EXIF data, ze kterých se zobrazí clonové číslo, čas, ISO, model fotoaparátu, objektiv a jeho ohnisková vzdálenost, tedy všechny potřebné údaje, aby si uživatel udělal představu o parametrech nastavených při pořízení fotografie.

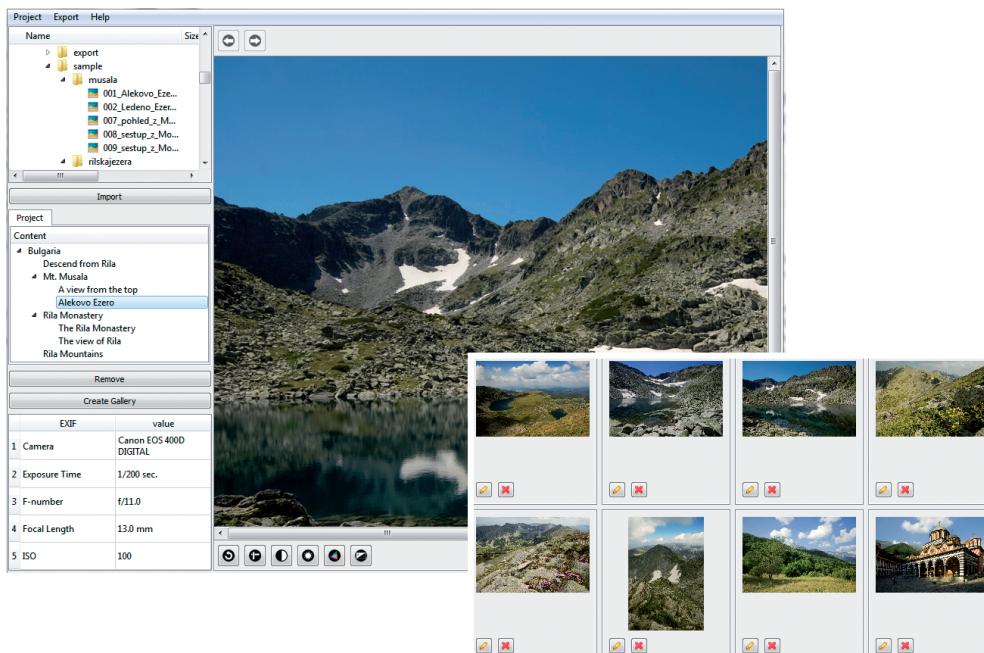
3.5 Zobrazování obsahu galerií a návrh GUI

Grafické uživatelské rozhraní je rozděleno na dvě části.

Levá část slouží pro správu projektu. Je zde zobrazena jeho struktura, detailní informace jednotlivých složek a fotografií. V ní nalezneme prvky pro správu projektu a zobrazování informací. Zároveň i ovládací tlačítka k editaci, mazání, importu či vytváření nových galerií.

Pravá část slouží k zobrazování a editaci. V ní se pak při kliknutí na danou galerii nebo fotografiю (v levé části), zobrazí buď obsah galerie nebo prostředí pro editaci fotografie. Lépe vše popíše obrázek 3.1.

⁴repositář je k nalezení na adrese <https://github.com/macicek/MGalleryManager-v2>



Obrázek 3.1: Grafické uživatelské rozhraní při zobrazení fotografie a při zobrazení galerie

V horní části aplikace se nachází menu, které obsahuje možnost ukládat a načítat projekt, export do podoby webové prezentace nebo pomoc odkazující na dokument v případě, že by si uživatel nevěděl rady.

3.6 Dostupné grafické úpravy a práce s pamětí

Úprav, které se mohou aplikovat na obraz, je neomezené množství. Zvoleny jsou takové, které běžný uživatel spíše použije:

- jas
- kontrast
- sytost
- převod do černobílé
- rotace
- transformace

Často to nejsou to pouze barvy, které chceme upravovat, ale jedná se i o změnu velikosti, aby fotografie nezabírala tolik prostoru nebo rotaci, v případě že je například horizont nakřivo. Ke každé úpravě je zároveň vhodné zachovávat její historii, aby ji šlo vrátit do původního stavu v případě, že se úprava nebude líbit.

Při zpracování obrazu se pracuje s obrovským množstvím dat. Proto nastává problém v případě, že chceme upravovat velké množství fotografií současně a zároveň udržovat informace o jejich úpravách. Některé úpravy můžou být destruktivní, tudíž už není možno zrekonstruovat původní snímek zpětným použitím zvolené úpravy. Minimálním množstvím, které tak musíme ukládat, je obraz každé úpravy tvořený všemi pixely. S každou další úpravou je méně pravděpodobné, že se uživatel vrátí zpět a využití paměti narůstá, proto je omezen počet uložených kroků na předem danou konstantu. Zároveň už není potřeba GUI prvků pro úpravu obrazu v případě, že se pracuje s jiným, proto se po změně zobrazovaného objektu uvolní paměť obsazená editorem. Historie úprav však musí zůstat zachována. Naopak malé náhledy fotografií při zobrazení galerie zabírají minimum paměti a očekáváme, že galerií nebude tolík, ale bude mezi nimi uživatel často přecházet. Ponecháme tak veškeré náhledy v paměti až do vypnutí programu.

Kapitola 4

Realizace

Program je rozdělen do modulů, kde každý je řešen s důrazem na minimální závislost na ostatních. Pro oddělení v rámci zdrojových kódů jsou použity jmenné prostory¹ - mgui (třídy pracující s GUI), mcore (třídy pracující s vnitřními strukturami), mnumeric (matematické funkce) a mimage (funkce pro zpracování obrazu).

4.1 Vnitřní informační struktura

Základní strukturou, která se otevře při spuštění programu je MProject, do kterého je celá vnitřní struktura zabalena. Stavebním prvkem projektu je následně třída MObject, která se pomocí dědičnosti dělí na MGallery, reprezentující složky označované jako galerie a MPhoto reprezentující jednotlivé fotografie. Každá struktura MGallery v sobě obsahuje datový kontejner std::set[3], který ukládá ukazatele na instance třídy MObject. To nám umožňuje vytvořit stromovou strukturu galerií a fotografií o libovolné hloubce, kde každá galerie může obsahovat jak fotografie, tak i další podgalerie.

MPhoto a MGallery jsou především pojítkem s GUI a zpracováním obrazu. Zeměpisné nebo i fotografické údaje o jednotlivých galeriích a fotografiích jsou řešeny samostatnými třídami. MObject obsahuje strukturu MGPSInfo se zeměpisnými informacemi, která je spořečná jak pro fotografie, tak i pro galerie. Každá galerie a fotografie obsahuje ještě struktury pro práci se specializovanými informacemi MGalleryInfo a MPhotoInfo.

4.2 Grafické uživatelské rozhraní

Jednotlivé prvky GUI, stejně tak jako vnitřní struktura, využívají dědičnosti. Většina zobrazovaných objektů je implementována jako rozšíření Qt GUI tříd.

Obsah projektu je zobrazován hned na několika místech. Za prvé v podobě stromové struktury v levé části GUI, kde je zobrazován celý projekt. Za druhé v hlavním okně, kde se zobrazuje obsah galerií nebo grafický editor. Ukazatele na jednotlivé instance GUI tříd jsou ukládány i do instancí MObject. V případě smazání MObjectu se tak smažou i všechny GUI reference.

¹namespace

	APP1 Marker ...
IFD0	Image width Image height ... ExifIFD offset GPS offset IFD1 offset Data
IFD1	Compression Resolution

Tabulka 4.1: Struktura EXIF dat

Při rozmišťování jednotlivých prvků se využívá Qt tříd QLayout, konkrétně QGridLayout, které je umožňují poskládat do mřížky, kde se zadává pouze řádek a sloupec, nemusí se tak vše posouvat, přidáme-li nový objekt.

4.3 EXIF údaje fotografií

Načítaná metadata jsou uložena v datových strukturách EXIF v hlavičce obrazového souboru JPEG. Data takového obrázku začínají sekvencí 0xFFD8 a končí 0xFFD9. Mimo jiné uvnitř existují sekvence značené podobně, konkrétně 0xFF – nazývané Markery. Markery od 0xFFE0 do 0xFFEF jsou nazývané Application Markers a značí data zaznamenaná uživatelskými aplikacemi. EXIF začíná Application Markerem APP1 (0FFE1), následují různé hlavičky a za nimi IFD0² a IFD1, což jsou námi požadované EXIF informace. IFD0 obsahuje informace jakými jsou šířka, výška, ale i offsety na jednotlivé informační položky, mezi kterými je i odkaz na tzv. ExifIFD. To je podsložka IFD0 obsahující veškeré dostupné fotografické informace. Offsety končí odkazem na následující IFD, tedy IFD1, obsahující náhled obrázku. Dále pokračuje datová část, na kterou ukazovaly jednotlivé offsety.

Při čtení EXIFů se musí nejdříve zvolit správné IFD, ze kterého se vybere potřebný tag - označení konkrétní EXIF informace. Samotné čtení už za nás řeší funkce knihovny libexif `exif_content_get_entry`, které předáme pouze informace, co má načíst.

4.4 Úprava fotografií

Aktuálně podporuje GalleryManager pouze formát JPEG, jehož podporu implementuje knihovna libjpeg. Po importu a volbě konkrétní fotografie umožňuje GalleryManager grafické úpravy. Ty probíhají nad 8-bitovým RGB prostorem³. Ke zpracování obrazu byla zvolena knihovna Boost::GIL od Adobe Systems Inc. Ta umožňuje nahlížet na fotografií pomocí

²Image File Directory

³Každá barva (červená, zelená, modrá) má 2^8 , tedy 256 úrovní (od 0 do 255), můžeme tedy nakombinovat až $256^3 = 16777216$ různých barev.

dvou datových typů, jako `image` a jako `view`. Konkrétní implementace datových typů nesou jména jako `rgb8_image_t` nebo `rgb8_view_t`, které značí, že pracují nad 8-bitovým RGB prostorem. `Image` obsahuje kompletní informace o fotografii, kdežto `view` je pouze jednoduchá struktura uchovávající informace o samotných pixelech. Je proto daleko výhodnější provádět úpravy nad pohledem.

`Boost::GIL` umožňuje s obrazy pracovat stejným způsobem jako s STL datovými kontejnery. Programátor tak může jednoduše procházet pixely obrazu pomocí iterátorů.

Jak již bylo zmíněno, GUI je řešeno pomocí frameworku Qt, který také obsahuje třídy pro práci s obrazem. Ty jsou ale velice omezené a pro budoucí rozšiřování práce nedostatečné. Propojení mezi zpracováním obrazu a vykreslováním náhledu je řešeno přes zkopirování jednotlivých pixelů z obrazu v `Boost::GIL` do `QImage`, Qt třídy vymezené pro práci s obrazem. `QImage` už lze jednoduše zobrazit v GUI překreslením do třídy `QPixmap`.

`GalleryManager` umožňuje následující úpravy - kontrast, jas, změna sytosti, otočení, transformace velikosti a převod do černobílé.

4.4.1 Kontrast

Kontrast nám určuje viditelnost věcí, tedy rozdíl barevných složek obrazu. Nízko-kontrastní fotografie obsahují z velké části střední šedé, naopak vysoko-kontrastní fotografie obsahují velké barevné rozdíly na hranách fotografovaných objektů. `GalleryManager` využívá, pro úpravu kontrastu následujícího výpočtu na množině reálných čísel:

1. Pro každý pixel zvlášť vezme jeho hodnoty RGB, které se pohybují v rozmezí 0 až 255
2. Od každé odečte 127 (polovinu maximální hodnoty 255), vyjde tak odchylka od střední hodnoty
3. Odchylka se vynásobí koeficientem z intervalu $<0.0, 2.0>$, uživatel zadává přívětivější hodnotu v rozmezí $<-100, 100>$
4. K výsledku se zpět přičte 127

Záporné hodnoty zadávané uživatelem značí pokles kontrastu. Kladné hodnoty jeho zvýšení. Lze si tak lépe představit požadovanou změnu.

Výsledek je omezen na interval $<0, 255>$. Pokud před převodem ležel mimo tento interval, dochází ke ztrátě barevné informace. Na obrazu tak jsou vidět buď bílé nebo čistě černé plochy. V extrémních případech, bude-li kontrast opakován navyšován, narostou nebo klesnou jednotlivé barevné složky pixelů natolik, že budou dosahovat pouze hodnot 0 nebo 255. Z původních různobarevných ploch tak vznikají plochy jednobarevné a zpětným snížením kontrastu už nelze vrátit obraz do původního stavu.

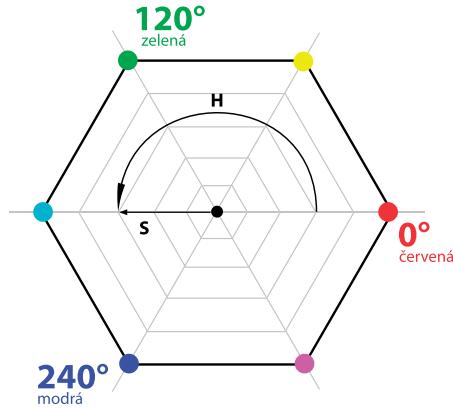
4.4.2 Jas

Jas se také označuje jako světlost obrazu, tedy jak blízko jsou jednotlivé barvy k bílé. Jeho implementace je celkem jednoduchá. RGB je aditivní barevný model, kde se jednotlivé barvy přičítají. Na rozdíl například od CMYK, kde se naopak odčítají. U RGB to znamená, že černá barva má hodnoty $(0, 0, 0)$. Bílá naopak $(255, 255, 255)$. Jas se tak zvyšuje přičítáním zvolené konstanty ke všem třem složkám a snižuje odečítáním.

4.4.3 Sytost a převod do černobílé

Změna sytosti je založena na barevném modelu HSV, který má sytost S jako jednu ze svých složek. Stačí tak převést každý pixel do barevného prostoru HSV, upravit sytost a převést zpět. Sytost se nachází v intervalu $<0, 1>$, uživatel zadává hodnoty opět v intervalu $<-100, 100>$. Ten se převádí na interval $<0, 2>$, kterým se původní sytost násobí.

HSV modeluje způsob, kterým pracují výtvarníci. Nejprve vyberou odstín barvy, který chtějí použít a následně přimíchávají černou nebo bílou, ke ztmavení resp. zesvětlení. Výběr odstínu lze pozorovat na obrázku 4.1, kde odstín H je dán úhlem a sytost S vzdáleností od středu.



Obrázek 4.1: Hodnoty HS barevného prostoru HSV

Převod do HSV nicméně není bezproblémový. Můžeme si všimnout, že v případě, kdy je vzdálenost od středu (sytost) nulová, není možné určit odstín a taková barva je nedefinovaná. K tomu dochází v případě, že se jedná o odstín šedé. V modelu RGB tak mají délčí barevné složky stejné hodnoty. Převod nám nejlépe popíše následující pseudokód:

```

min      = min(red, green, blue)
max      = max(red, green, blue)

value    = max
saturation = (max-min)/max

if saturation == 0
    hue = UNDEFINED
else
    diff = max - min
    if max == red
        hue = (green - blue) / diff
    else
        if max == green
            hue = 2 + (blue - red) / diff

```

```

else
if max == blue
    hue = 4 + (red - green) / diff

hue = hue * 60
if hue < 0
    hue = hue + 360

```

Lze pozorovat, že při určování odstínu se dělí rozdílem maximální a minimální hodnoty RGB. V případě šedé jsou všechny tři barvy stejné a rozdíl je nulový. V definičním oboru reálných čísel tak není výpočet definován.

Sytost lze mimo jiné chápat i jako intenzitu nebo čirost dané barvy, jinými slovy ji lze snižovat přidáváním bílé, kdy se barva stává zakalenější. Hodnota V naopak určuje odchylku od černé. Na základě maximální hodnoty RGB se pak určí, v jaké třetině barevného kola se barva nachází. Kterým směrem a o jakou hodnotu se barva posune, se určí rozdílem sousedních barev.

Převod do černobílé se provádí snížením sytosti na nulu. U barev, které mají jednu RGB složku nulovou a další 255, tedy maximální, dostaneme vždy bílou. Na 4.1 se jedná o barvy na okraji šestiúhelníku. Nevýhodu tohoto přístupu i přes jeho matematickou správnost je, že tak vznikají bílé plochy, které lidské oko vnímá jako vypálené.

4.4.4 Rotace a transformace

Rotace je v GalleryManageru řešena přes maticové násobení za pomoci matematického rozšíření knihovny Boost::GIL. Nejprve se vygeneruje rotační matice, kterou se vynásobí všechny souřadnice pixelů. Nastává však problém, že rotace probíhá kolem souřadnic [0, 0], tedy dolního-levého rohu a obraz se dostane mimo plátno. Musí se tak nejprve přizpůsobit velikost plátna a výsledek následně i posunout.

Boost::GIL využívá matic 3x2, které jsou ve skutečnosti maticemi 3x3, kde poslední řádek je vždy [0 0 1], protože veškeré úpravy probíhají ve 2D prostoru. To umožňuje provádět nejen rotaci, ale i translaci pomocí maticového násobení. Stačí tak vygenerovat jednu transformační matici, která se aplikuje na každou souřadnici. Výpočet probíhá následovně:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

Hodnoty t_x a t_y se získávají funkcemi sin a cos na základě toho, jak se každý roh plátna natočil. Výpočet je různý pro každý kvartál rotace.

Při transformaci obrazu, tedy změně jeho velikosti, dochází k jeho deformaci. Je tedy třeba vypočítat výslednou barvu každého nového pixelu. K tomu existuje více metod, kde každá má své výhody i nevýhody. GalleryManager využívá bilineárního vzorkování, které dosahuje nejlepších výsledků, jsou-li rozměry výsledného obrazu nejvýše dvakrát menší nebo dvakrát větší. Pro srovnání se vzorkováním dle nejbližšího souseda, kde se vybírá vždy barva nejbližšího pixelu, se zde barvy průměrují, a proto dosahuje výsledný obraz plynulejších přechodů na kontrastních hranách.

Bilineární vzorkování používá čtyři okolní pixely, jejichž barvu zprůměruje na základě vzdálenosti od výsledného pixelu. Představíme-li si obraz jako mřížku, výsledný pixel bude v poli této mřížky. Nejprve se vypočte na základě velikosti zvětšení obrazu vážený průměr barvy ve směru osy x u horního a dolního páru pixelů. Následně se vypočte vážený průměr těchto dvou hodnot na základě zvětšení ve směru osy y. Váha poměru, na základě které probíhá průměrování pro barvy v rámci výpočetních párů, záleží na koeficientu, kterým se obraz zvětšuje nebo zmenšuje.

Kapitola 5

Testování

Testování funkčnosti jednotlivých algoritmů probíhalo porovnáním výsledků s existujícími produkty, zejména s programem Adobe Photoshop.

5.1 Porovnání úprav s programem Adobe Photoshop

5.1.1 Převod do černobílé a vykreslování obrazu



Obrázek 5.1: Převod do černobílé a vykreslování obrazu

Na obrázku 5.1 si lze všimnout především dvou věcí. První jsou algoritmy, které se používají při automatickém převodu do černobílého prostoru. Fotografie převedená ve Photoshopu má kontrastnější oblohu, než-li fotografie upravená v GalleryManager. Převod na černobílou fotografii se provádí na základě sytosti barev. Syté barvy jsou převedeny na světlé a naopak. Matematicky je sytá barva taková, která má velký rozdíl mezi maximální a minimální hodnotou RGB. Čistá modrá (0, 0, 255), zelená (0, 255, 0) nebo červená (255, 0, 0) budou převedeny na bílou. Lidské oko zároveň vnímá sytost každé barevné složky RGB jinak. Toho využívá Photoshop a nepřevádí lineárně, ale například z modrých, které jsou vnímány jako méně syté, vytváří tmavší barvy. Ze zelených naopak zase barvy světlejší.

Druhou věcí je vykreslování náhledu obrazu. Photoshop využívá nejspíše i při samotných úpravách antialiasingu¹ a vyhlazuje tak přechody mezi barvami, u GalleryManageru lze pozorovat stejnobarvené plochy a mezi nimi náhlé tonální skoky.

5.1.2 Úprava kontrastu



Obrázek 5.2: Úprava kontrastu

¹Vyhlažování hran, oblouky nebo křivky tak nejsou zubaté, ale dodávají zarovnaný dojem

Při úpravě kontrastu nabízí Adobe Photoshop dvě možnosti. Novou a starou metodu. Stará metoda funguje stejně jako algoritmus implementovaný v GalleryManageru. Zajímavější je tak porovnat výsledky s novou metodou. Hlavním rozdílem je práce s barvami na hranici bílé. Úprava kontrastu funguje na principu, že čím dále jsou jednotlivé barevné složky od střední šedé (127, 127, 127), tím více danou barevnou složku snížíme, resp. zvýšíme. Extrémně světlé barvy tak začnou nabývat hodnot bílé (255, 255, 255) a tmavé naopak hodnot černé (0, 0, 0). Na 5.2 je toto vidět především na mraku v horní části, kde dochází ke ztrátě kresby. Photoshop se zastaví na minimálních barevných hodnotách a dále už mrakové kontury nezničí.

Zároveň barvám blíže středu barevného prostoru je přidáváno na sytosti. Tato vlastnost mi nepřijde u Photoshopu příliš dobře zvolená, z fotografického hlediska tak získávají barvy nerealistickou podobu. V GalleryManageru naopak vypadají věrohodněji. Na druhou stranu, Photoshop obsahuje i možnosti úprav jednotlivých barevných kanálů nebo přímo konkrétních odstínů a může tak tyto vlastnosti odstranit jinými způsoby.

5.1.3 Úprava sytosti



Obrázek 5.3: Úprava sytosti

Změna sytosti dopadla v obou editorech velice podobně. Ve Photoshopu lze pozorovat, že oranžový strom na obrázku 5.3 v popředí je světlejší, než tomu je u GalleryManageru. Lidské oko vnímá sytost každé barvy trochu jinak. Vnímá tak zelenou nebo i červenou více, než tomu

je u modré. GalleryManager bere v úvahu pouze maximální a minimální hodnoty jednotlivých barevných složek. Úpravu sytosti pak provádí lineárně. Photoshop tento fakt bere v úvahu, a tak přísytí některé části fotografie více. Zřetelné to je právě u stromu, kde oranžová obsahuje velké množství zelené a červené, které hodně přispívají k sytosti vnímané lidským okem.

5.2 Testování exportu webové prezentace

Vyexportovaná webová prezentace byla otestována přes HTML validator², který zaručuje, že HTML a CSS kód dodržuje standardy dané konsortiem W3C, a je tak zobrazitelný ve všech dnes používaných webových prohlížečích.

²<http://validator.w3.org/>

Kapitola 6

Závěr

6.1 Zhodnocení splnění cílů

Byl vytvořen program GalleryManager, který dává uživateli možnost organizovat a popisovat soubory fotografií. Program dokáže načítat metadata typu EXIF, která jsou vkládána do obrazu každým moderním digitálním fotoaparátem a získat tak všechny důležité údaje o pořízení fotografie (nastavení fotoaparátu - čas, zeměpisná poloha a podobně).

Grafický editor umožňuje u fotografií provádět následující úpravy:

- Úprava jasu a kontrastu
- Změna sytosti barev
- Otáčení snímku o libovolný úhel
- Zmenšování a zvětšování
- Převod barevné fotografie na černobílou

Účinnost zvolených algoritmů byla porovnána s algoritmy používanými programem Adobe Photoshop CS4 za použití testovacích souborů ve formátu JPEG a 8-bitové barevné hloubce. Při použití malých obrazových souborů, které se využívají pro web, je účinnost jednoduchých algoritmů GalleryManageru srovnatelná. Výhody pokročilejších algoritmů používaných ve Photoshopu se projeví při práci s většími soubory a při vícenásobných a rozsáhlých úpravách snímku. Konkrétní výsledky jsou popsány v sekci testování.

Z názvu repositáře na [githubu.com](https://github.com) (MGalleryManager v2) je patrné, že se jedná o druhou verzi programu. V průběhu práce na první vyšlo najevo, že původní myšlenka napojování jednotlivých modulů na GUI přes datové tabulky způsobovala nedostatek záznamů o provázanosti mezi třídami. Důsledkem bylo, že s každou novou funkcí bylo nutné rozsáhlé přepracování zdrojového kódu.

Ve druhé verzi je již zcela odděleno uživatelské rozhraní od části programu, která se využívá pro zpracování obrazu. Díky tomu je jednoduché přidávat další funkce, bez rozsáhlých zásahů do celého programu. Aplikaci tak lze snadno rozšiřovat. Kód je volně ke stažení z

repositáře na webovém serveru github.com¹ a okomentován pomocí syntaxe Doxygen, aby na něm mohli pokračovat i další vývojáři.

6.2 Návrhy na zlepšení a cíle do budoucnosti

Každý program se dá vždy vylepšovat a kromě malých změn v podobě přívětivosti uživatelského prostředí lze práci do budoucnosti rozšiřovat následujícími projekty:

- Grafické úpravy implementovat jako samostatnou knihovnu - rozšíření Boost::GIL, oddělit ji tak jako nezávislý projekt
- Podpora více-bitových barevných hloubek (16-bit, 32-bit)
- Podpora dalších formátů používaných ve fotografii, především TIFF, RAW (specifický pro jednotlivé výrobce fotoaparátu)
- Export do formátu PDF nebo v podobě SQL dotazů pro databáze

Obrovskou výhodou knihovny Boost::GIL, pro zpracování obrazu v různých barevných hloubkách nebo formátech, je způsob, kterým je navržena. Už podle názvu Generic Image Library, je většina funkcí napsána jako template². Umožňuje tak pracovat stejně s 8-bit RGB prostorem jako s 16-bit CMYK. Je třeba implementovat pouze podporu pro jednotlivé formáty a algoritmy pro úpravu obrazu napsat dostatečně genericky, aby mohly pracovat nad libovolným barevným prostorem nebo hloubkou.

Mělo by se tak navázat na myšlenku generického programování i při implementaci samostatné knihovny pro úpravu obrazu.

Cílem je zahrnout práci do redakčního systému a navázat ji na databázi rozpracovaných webových serverů FotoAtlas.cz a PhotographicAtlas.com. V oblasti grafických úprav bude přínosem implementovat pokročilejší algoritmus pro zmenšení obrazu, jako například bikubické převzorkování. Aktuálně GalleryManager využívá bilineární převzorkování. Pro uživatele by byla přínosem i ochrana fotografií pomocí individuálních vodoznaků.

V průběhu svého studia bych se i nadále chtěl věnovat algoritmům pro úpravu obrazu. Rád bych tedy rozšířil práci do podoby samostatného projektu grafické C++ knihovny implementující i pokročilejší algoritmy, než-li jsou použity v GalleryManageru.

¹<https://github.com/macicek/MGalleryManager-v2>

²česky šablona - datové typy nejsou zadány konkrétně, ale vygenerují se jednotlivé verze funkce na základě toho, s jakými datovými typy je použita

Literatura

- [1] C. GALLOWAY, L. MUELLER. The libexif C EXIF library.
<http://libexif.sourceforge.net/api/>, stav z 12. 12. 2011.
- [2] CIPA DC-008-Translation 2010. Exchangable image file format for digital still cameras:
Exif version 2.3, 2010.
http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf.
- [3] cplusplus.com. C++ STL Containers.
<http://wwwcplusplus.com/reference/stl/>, stav z 29. 11. 2011.
- [4] D. VAN HEESCH. Doxygen Manual.
<http://www.stack.nl/~dimitri/doxygen/index.html>, stav z 20. 12. 2011.
- [5] E. HAMILTON. JPEG File Interchange Format.
<http://www.jpeg.org/public/jfif.pdf>.
- [6] K. SUGIYAMA. Desktop Color Handbook 09/10, 2009.
- [7] L. BOURDEV, H. JIN, C. HENNING. Generic Image Library Documentation.
http://www.boost.org/doc/libs/1_47_0/libs/gil/doc/index.html, stav z 28. 11. 2011.
- [8] Nokia Corporation. Qt Reference Documentation.
<http://doc.qt.nokia.com/4.7/index.html>, stav z 29. 11. 2011.
- [9] Přispěvatelé L^AT_EXWikibook. L^AT_EX Wikibook.
<http://en.wikibooks.org/wiki/LaTeX/>, stav z 2. 1. 2012.
- [10] Přispěvatelé Wikipedie. Transformation matrix, 2011.
http://en.wikipedia.org/wiki/Transformation_matrix.
- [11] Přispěvatelé Wikipedie. Bilinear filtering, 2011.
http://en.wikipedia.org/wiki/Bilinear_filtering.
- [12] Přispěvatelé Wikipedie. Rotation matrix, 2011.
http://en.wikipedia.org/wiki/Rotation_matrix.
- [13] S. CHACON. Git community book.
<http://book.git-scm.com/>, stav z 29. 11. 2011.

Příloha A

Seznam použitých zkratek

API Application Programming Interface

CMYK Cyan Magenta Yellow Black

EXIF Exchangeable Image File Format

GIL Generic Image Library

GUI Graphical User Interface

HSV Hue Saturation Value

IPTC Information Interchange Model - International Press Telecommunications Council

IFD Image File Directory

JPEG Joint Photographic Experts Group

LGPL GNU Lesser General Public License

RGB Red Green Blue

STL Standard Template Library

TIFF Tagged Image File Format

W3C World Wide Web Consortium

XMP Extensible Metadata Platform

:

Příloha B

Instalační a uživatelská příručka

Na přiloženém CD je k nalezení spustitelný *.exe soubor s přiloženými dynamickými knihovnami pro operační systému Windows. V případě potřeby spustit projekt pod operačním systémem Linux je třeba projekt znova zkompilovat.

Projekt je závislý na knihovnách libjpeg, Boost::gil (verze 1.47) s rozšířeními, Qt (verze 4.7) a libexif.

Při komplaci pod OS Linux je třeba postupovat následovně:

1. Nainstalovat knihovnu libjpeg, která je k nalezení na <http://www.ijg.org/>.
2. Stáhnout knihovnu Boost (je otestována na verzi 1.47, u verze 1.48, která vyšla během psaní bakalářské práce, docházelo ke kolizím jmen s knihovnou Qt) . K nalezení je na <http://sourceforge.net/projects/boost/files/boost/1.47.0/>. Knihovna je tvořena hlavičkovými soubory, není tak třeba instalace. Pro GIL je třeba i numerické rozšíření z <http://opensource.adobe.com/wiki/display/gil/Downloads>, které se pouze nakopíruje do stávající knihovny Boost.
3. Nainstalovat knihovnu libexif, která je k nalezení na <http://sourceforge.net/projects/libexif/files/libexif/0.6.20/>.
4. Nainstalovat Qt SDK z <http://qt.nokia.com/downloads>.
5. Projekt spustit v prostředí Qt Creator.
6. Je třeba upravit projektový soubor MGalv2.pro, kde se do LIBS přidají odkazy na knihovny libjpeg a libexif a do INCLUDEPATH cesty k hlavičkovým souborům Boost::GIL, libexif a libjpeg.
7. Spustit komplaci v prostředí Qt Creator.

Pro instalace knihoven libexif a libjpeg by mělo stačit změnit cestu v konzoli na složku, do které byly knihovny nakopírovány a následně jen spustit příkazy:

```
./configure  
make  
make install
```


Příloha C

Obsah přiloženého CD

Kromě tohoto dokumentu je na přiloženém CD i program ve spustitelné podobě společně se všemi závislostmi, zdrojové kódy, testovací soubory, z nich vytvořený projekt a z projektu vyexportovaná webová stránka. Struktura CD je následující:

- /kořenový adresář
 - /**exe** - složka obsahující spustitelný program a všechny závislosti
 - * gallerymanager.exe - spustitelný program
 - * *.dll - dynamické knihovny potřebné ke spuštění programu
 - /**images** - složka obsahující obrázky využívané programem
 - *.png - obrázky ve formátu PNG využívané programem
 - *.jpg - obrázky ve formátu JPG využívané programem
 - /**export** - složka obsahující testovací vyexportovaný projekt z aplikace
 - /**sample** - složka obsahující testovací fotografie
 - * /**musala** - složka obsahující vzorové obrázky
 - *.jpg - vzorové obrázky ve formátu JPG
 - * /**rilskajezera** - složka obsahující vzorové obrázky
 - *.jpg - vzorové obrázky ve formátu JPG
 - * /**rilskymonastyr** - složka obsahující vzorové obrázky
 - *.jpg - vzorové obrázky ve formátu JPG
 - * *.jpg - vzorové obrázky ve formátu JPG
- /**src** - složka obsahující zdrojové kódy
 - * /**core** - složka obsahující cpp/h soubory s třídami jádra, tj. pro práci s fotografiemi a galeriami
 - *.png - obrázky ve formátu PNG využívané programem
 - *.jpg - obrázky ve formátu JPG využívané programem
 - * /**gui** - složka obsahující cpp/h soubory s GUI třídami
 - /**mgridwidget** - složka obsahující cpp/h soubory prvku MGridWidget (část GUI, kde se zobrazuje editor a obsah galerií)
 - *.cpp - C++ soubory
 - *.h - hlavičkové soubory

- /**mtreewidget** - složka obsahující cpp/h soubory prvku MTreeWidget (část GUI zobrazující stromovou strukturu projektu)
 - *.cpp - C++ soubory
 - *.h - hlavičkové soubory
- *.cpp - C++ soubory
- *.h - hlavičkové soubory
- * MGalv2.pro - projektový soubor editoru Qt Creator
- * MGalv2.pro.user - uživatelská nastavení projektu
- * *.cpp - C++ soubory
- * *.h - hlavičkové soubory
- /**text** - složka obsahující dokumentaci k bakalářské práci
 - * bp.pdf - vyexportovaná práce ve formátu PDF
 - * bp.tex - zdrojový kód dokumentace práce
 - * bp.bbl - zdrojový kód použité literatury
 - * readme.txt - návody ke spuštění a komplikaci programu, technické parametry
 - * /**figures** - složka obsahující obrázky používané v dokumentaci
 - *.pdf - obrázky ve formátu PDF