

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

EDITOR EXIF A IPTC ÚDAJŮ Z FOTOGRAFIÍ

BAKALÁŘSKÁ PRÁCE

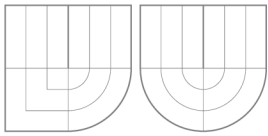
BACHELOR'S THESIS

AUTOR PRÁCE

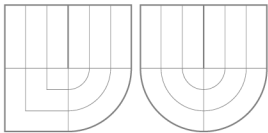
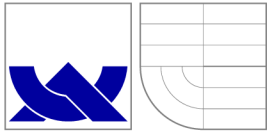
AUTHOR

JAN KUŽEL

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

EDITOR EXIF A IPTC ÚDAJŮ Z FOTOGRAFIÍ

EDITOR OF EXIF AND IPTC VALUES FROM IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN KUŽEL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN HORÁČEK

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a implementací programu, který slouží k editaci EXIF a IPTC údajů u fotografií formátu jpeg. Aplikace dále umožňuje zobrazování náhledů fotografií, úpravu fotografií pomocí zásuvných modulů, promítání fotografií a vyhledávání fotografií s využitím pokročilých voleb, mezi které patří i GPS souřadnice se zadaným rozsahem.

Abstract

This bachelor's thesis deals with analysis, design and implementation of the programme that is used to edit EXIF and IPTC data in JPEG images. The application also allows for previewing photos, editing photos using the plug-ins, showing photos and finding photos using advanced features, including the GPS coordinates of the specified range.

Klíčová slova

EXIF, GPS, HViewer, IPTC, grafické uživatelské rozhraní, metadata, zásuvný modul, XMP, vyhledávání, fotografie

Keywords

EXIF, GPS, HViewer, IPTC, graphical user interface, metadata, plugin, XMP, search, photograph

Citace

Jan Kužel: Editor EXIF a IPTC údajů z fotografií, bakalářská práce, Brno, FIT VUT v Brně, 2011

Editor EXIF a IPTC údajů z fotografií

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Horáčka a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Kužel
12. května 2011

Poděkování

Rád bych poděkoval svému vedoucímu, Ing. Janu Horáčkovi, za vedení práce a poskytnuté věcné připomínky, které mi pomohly při jejím zpracování.

© Jan Kužel, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

| | |
|---|-----------|
| 1 Úvod | 5 |
| 1.1 Členění práce | 5 |
| 2 Metadata u fotografií | 7 |
| 2.1 EXIF | 7 |
| 2.2 IPTC | 8 |
| 2.3 XMP | 9 |
| 3 Požadavky na výsledný program | 10 |
| 4 Návrh programu | 11 |
| 5 Implementace programu | 14 |
| 5.1 Grafické uživatelské rozhraní | 14 |
| 5.2 Náhledy fotografií | 15 |
| 5.3 Metadata EXIF | 16 |
| 5.4 Metadata IPTC | 17 |
| 5.5 Zásuvné moduly | 18 |
| 5.5.1 Bezeztrátová rotace | 20 |
| 5.6 Vyhledávání fotografií | 21 |
| 5.6.1 Funkce undo a redo | 22 |
| 5.7 Promítání fotografií | 22 |
| 5.8 Hierarchie tříd a modulů | 23 |
| 6 Popis výsledného programu | 24 |
| 6.1 Požadavky na systém | 24 |
| 6.2 Náhledy fotografií | 25 |
| 6.3 Zásuvné moduly | 26 |
| 6.3.1 Bezeztrátová rotace | 27 |
| 6.3.2 Grayscale | 27 |
| 6.4 Exif a Iptc metadata | 27 |
| 6.4.1 GPS | 29 |
| 6.5 Vyhledávání | 29 |
| 6.5.1 Využití GPS souřadnic | 30 |
| 6.6 Promítání fotografií | 32 |

| | |
|--|-----------|
| 7 Porovnání se stávajícími programy | 34 |
| 7.1 Grafické uživatelské rozhraní a náhledy fotografií | 34 |
| 7.2 Promítání fotografií | 35 |
| 7.3 Správa metadat | 36 |
| 7.4 Vyhledávání fotografií | 37 |
| 7.5 Ostatní funkce | 38 |
| 8 Možnosti rozšíření | 40 |
| 9 Závěr | 41 |
| A Obsah CD | 45 |

Seznam obrázků

| | | |
|-----|---|----|
| 4.1 | Návrh programu HViewer. | 11 |
| 5.1 | Vzorový příklad hierarchie tříd týkající se modulu <code>modul</code> | 14 |
| 5.2 | Hierarchie tříd modulu <code>mainwindow</code> | 15 |
| 5.3 | Hierarchie tříd modulu <code>exif</code> | 17 |
| 5.4 | Hierarchie tříd modulu <code>iptc</code> | 18 |
| 5.5 | Hierarchie tříd modulu <code>plugins</code> se třídou <code>LossLessRotate</code> | 21 |
| 5.6 | Hierarchie tříd modulu <code>search</code> | 22 |
| 5.7 | Hierarchie tříd modulu <code>slideshow</code> | 23 |
| 5.8 | Hierarchie modulů programu HViewer. | 23 |
| 6.1 | Zobrazení náhledů fotografií. | 25 |
| 6.2 | Rozhraní pro práci se zásuvnými moduly (po aplikaci bezztrátové rotace o 180°). | 26 |
| 6.3 | Dialogové okno zásuvného modulu pro bezztrátovou rotaci obrazu. | 27 |
| 6.4 | Zobrazení rozhraní pro práci s IPTC a EXIF metadaty. | 28 |
| 6.5 | Práce s GPS souřadnicemi. | 29 |
| 6.6 | Zobrazení rozhraní pro vyhledávání fotografií. | 30 |
| 6.7 | Rozhraní pro nastavení GPS souřadnic u vyhledávání. | 31 |
| 6.8 | Mapa pro nastavení GPS souřadnic u vyhledávání. | 31 |
| 6.9 | Promítání fotografií. | 32 |

Seznam tabulek

| | | |
|-----|--|----|
| 4.1 | Ukázka třídy, realizující rozhraní pro připojování zásuvných modulů. | 13 |
| 4.2 | Ukázka třídy v zásuvném modulu, aneb v dynamicky sestavené knihovně. . | 13 |
| 5.1 | Ukázka formátu konfiguračního souboru <code>saveFile.txt</code> | 16 |
| 5.2 | Používání programu <code>exiv2</code> a funkce <code>system()</code> v modulu <code>exif</code> | 17 |
| 5.3 | Používání programu <code>exiv2</code> a funkce <code>system()</code> v modulu <code>iptc</code> | 18 |
| 5.4 | Výpis informací týkající se určitého zásuvného modulu. | 19 |
| 5.5 | Načtení zásuvného modulu, jehož absolutní umístění, v rámci souborového systému, udává řetězec <code>pluginNameAbsolutePath</code> | 20 |
| 5.6 | Používání programu <code>exiftran</code> a funkce <code>system()</code> v zásuvném modulu pro bezeztrátovou rotaci. | 20 |
| 5.7 | Použití metody <code>ApplyCommand</code> | 21 |

Kapitola 1

Úvod

S rostoucí velikostí a počtem sbírek fotografií, nejen na internetových portálech k tomu určených (Picasa, flickr, MyPhotoAlbum, Windows Live Gallery, . . .), ale i u běžných uživatelů, se zvětšuje potřeba fotografie nějakým způsobem značit, popisovat, podle popisu pak organizovat, třídít, seskupovat či automaticky upravovat. S těmito potřebami se pak váže hledání vhodné aplikace, či skupin aplikací, které by vykonaly co největší množství práce s co nejmenším úsilím, za co nejkratší časový úsek.

Prvním krokem, při zpracování fotografií podle jejich popisu, je vložení samotného popisu do fotografie, tzn. vložení metadat. Za tímto účelem vznikají, a v minulosti již vznikaly, stále nové aplikace, funkce, či zásuvné moduly, které k výše zmíněnému slouží. Ty jsou pak využívány samotnými prohlížeči fotografií, kteří, jako celek, nabízí různorodé možnosti ovládání, vyhledávání, promítání fotografií, či odlišný vzhled a většinou ještě něco navíc, aby se od ostatních prohlížečů nějakým způsobem odlišily.

Byla tedy snaha o vytvoření programu, který by poskytoval dobrou a jednoduchou funkčnost a zvládal to, co se od něj očekává (3). Byl tedy vytvořen uživatelsky velmi jednoduše použitelný program, který se věnuje převážně zpracování metadat a vyhledávání pomocí nich, včetně GPS souřadnic s grafickým znázorněním na mapě, podporující zobrazování náhledů, promítání fotografií, v případě potřeby rozšířitelný pomocí zásuvných modulů, které jsou poměrně jednoduché na vytvoření a jehož zdrojové kódy jsou volně dostupné.

Zadání, návrhu, implementaci, popisu, porovnání a možnostem vylepšení zmíněného programu se pak věnuje následující dokument, jehož podrobnější členění je uvedeno v následující podkapitole.

1.1 Členění práce

V tomto dokumentu je nejprve, v kapitole 2, vysvětlen pojem metadata, za kterým jsou uvedeny jednotlivé typy metadat, typ IPTC (2.2), EXIF (2.1) a XMP (2.3), které jsou využívány ve velkém množství programů a knihoven, jejich výhody a nevýhody, případně atributy, jež je pomocí nich možné zapsat.

Poté, v kapitole 3, jsou uvedeny, rozčleněny a blíže popsány požadavky, které byly kladeny na výsledný editor.

Následuje kapitola číslo 4, která se věnuje návrhu výsledného programu, ve které jsou rozebrány požadavky a navrhnutá jejich řešení, za kterou se nachází kapitola další (5), která

poměrně detailně popisuje implementaci výsledného programu, obsahuje diagramy tříd jednotlivých modulů, nejdůležitější metody v nich používané a klíčové úseky kódu.

Kapitola je logicky rozčleněna na podkapitoly podle jednotlivých částí programu. První (5.1) se týká grafického uživatelského rozhraní, druhá (5.2) náhledů fotografií, další dvě (5.3, 5.4) pak metadat, která jsou výsledným editorem zpracovávána, pátá podkapitola (5.5) zásuvným modulům, další dvě (5.6 a 5.7) vyhledávání a promítání fotografií.

Kapitola 6 je věnována výslednému programu, kdy je vysvětlena a detailně popsána jeho funkčnost z uživatelského hlediska. Kapitola je členěná podobně, jako kapitola číslo 5, tj. na sekci s náhledy fotografií (6.2), se zásuvnými moduly (6.3), s metadaty (6.4), na sekci věnovanou vyhledávání (6.5) a promítání fotografií (6.6).

Kapitola číslo 7 je věnována porovnání výsledného programu, popsaného v kapitole předchozí (6), se stávajícími programy s podobnou funkčností. Porovnání probíhá v několika kategoriích. První je porovnání z hlediska grafického uživatelského rozhraní a zobrazování náhledů (7.1), druhá porovnává promítací funkce jednotlivých programů (7.2), další porovnává programy podle toho, jak si poradí se zpracováním metadat (7.3), předposlední pak s vyhledáváním (7.4) a poslední (7.5) se věnuje ostatním funkcím.

Kapitola číslo 8 popisuje možnosti, kterými by mohl být dosavadní program HViewer rozšířen, nebo vylepšen a hrubý nástin, jakým způsobem by mohlo být rozšíření či vylepšení realizováno.

Kapitola 9 pak obsahuje zhodnocení práce jako celek, konstatování o splnění cílů a dosažené výsledky.

Kapitola 2

Metadata u fotografií

Metadata jsou informace, které se týkají obsahu digitálních fotografií (a ne jen jich). Tyto informace jsou pak načítány či ukládány pomocí různých programů, nebo jsou vkládány přímo digitálním fotoaparátem, mobilním telefonem či digitální kamerou, zkrátka zařízením, kterým je fotografie pořízena.

Metadata mohou sloužit, a v grafických editorech, prohlížečích fotografií, nebo na internetových portálech, často také slouží, k vyhledávání fotografií, jejich třídění, úpravě a dalším operacím.

V dnešní době existuje větší množství formátů metadat (různí výrobci si někdy definují své vlastní formáty, ale protože nejsou nijak standardizovány, nesetkávají se s velkým úspěchem), z nichž nejrozšířenější jsou následující.

1. Formát EXIF, od japonského sdružení JEIDA (Japan Electronic Industries Development Association), vytvořený roku 1988, jehož specifikace je dostupná na internetové stránce [19].
2. Formát IPTC, neboli IPTC-NAA-Standard, vyvinutý asociací NAA (Newspaper Association of America) a koncilem IPTC (International Press Telecommunications Council) zhruba v roce 1990 (pojmenovaný IIM – Information Interchange Model), jehož specifikace je dostupná na [20].
3. Formát XMP, což je zkratka Adobe Extensible Metadata Platform, vytvořený firmou Adobe Systems Inc. . Kompletní specifikace, dokumentace i knihovny pro práci s XMP metadaty jsou ke stažení na internetových stránkách firmy Adobe [21].

2.1 EXIF

Metadata EXIF, jež jsou vkládána přímo do grafických souborů formátu jpeg či tiff (bližší informace o grafických formátech viz. [27] a [28], což je kniha, která se blíže zabývá kódováním či zpracováváním signálů u obrazových i jiných formátů), která jsou, mimo digitálních fotoaparátů a kamer, využívána také například skenery a kapesními počítači, pokrývají poměrně široké spektrum, které obsahuje:

- Informace o datu a čase, ve kterém byla fotografie uložena, pořízena či digitalizována.

- Nastavení fotoaparátu, které obsahuje statické informace (tj. informace, které se při focení nemění), jako je značka fotoaparátu a jeho výrobce, a dynamické informace (tj. informace, které se při focení dynamicky, podle potřeby, mění), jako je orientace fotografie, světelnost, rychlost závěrky, clona, ohniskovou vzdálenost apod.
- Tzv. thumbnail, což je náhled, který je využíván při zobrazení na LCD displeji digitálního fotoaparátu, v prohlížečích souborů, atd.
- Informace o autorovi, které se většinou vkládají při dodatečném zpracování.
- GPS informace, které jsou vkládány některými digitálními fotoaparáty či mobilními telefony, které mají vestavěný GPS přijímač, nebo dodatečně, uživateli, pomocí vhodných programů.

Samotný standard EXIF obsahuje několik nevýhod, mezi které patří propojení jednotlivých atributů, se kterým se velké množství editorů nedokáže vypořádat, čímž dochází k poškození, nebo odstranění již zapsaných metadat.

Podpora pouze dvou zmíněných grafických formátů (**jpeg**, **tiff**) a barevné hloubky pouze 24 bitů (některé fotoaparáty podporují 36 bitů).

Nemožnost vkládat EXIF informace k video formátům, což je v dnešní době, kdy téměř všechny fotoaparáty dokáží video natáčet, poměrně velká nevýhoda.

Další nevýhody jsou uvedeny na internetové stránce [11], ze které jsem čerpal v rámci celé podkapitoly (mimo zmíněné specifikace z internetové stránky [19]), na které je možné nalézt další užitečné informace.

2.2 IPTC

Druhý typ metadat, metadata IPTC, jsou textová data, která se, podobně jako metadata EXIF, využívají pro vkládání informací přímo do grafických souborů **jpeg** a **tiff** a jsou rozdělena do dvou částí (schémat). První je pojmenovaná **iptc Core shema**, druhá pak **iptc Extension shema**, která rozšiřuje část první.

Z počátku se formát IPTC využíval u všech typů multimediálních souborů. Mimo souborů grafických i u textových či zvukových, u kterých však nezískal příliš na popularitě a dnes už se využívá pouze u souborů grafických.

I metadata IPTC jsou často využívána grafickými editory či prohlížeči digitálních fotografií, ale na rozdíl od metadat EXIF, které se týkaly především nastavení, výrobce a modelu fotoaparátu, kterým jsou také vkládány, se metadata IPTC týkají především autora fotografie a samotného obsahu fotografie a bývají přidávána dodatečně.

Umožňují uložit adresu, kontakt, autorská práva, titulek, klíčová slova, která jsou velmi využívána při vyhledávání, kategorie, do které snímek spadá a mnoho dalšího.

Tabulka jednotlivých parametrů je na internetové stránce [8], další informace pak na [7], což jsou, kromě specifikace ([20]), stránky, ze kterých bylo v současné podkapitole čerpáno.

2.3 XMP

Posledním formátem metadat je textový formát XMP, který vychází z formátu XML a začleňuje formát IPTC, který je určený pro zpracování souborů (grafických i jiných) podle informací, které se týkají jejich obsahu. Formát metadat XMP je opět využíván v editorech, prohlížečích fotografií a například i v pdf souborech. Hlavním rozdílem oproti předešlým formátům je to, že se jedná samostatný soubor, nikoliv o součást fotografie.

Formát XMP definuje vlastnosti, které jsou užitečné pro získání informací o „historii fotografie“, tj, jestli byla fotografie vyfocena, naskenována, editována v nějakém vhodném editoru, atd. Umožňuje tedy každému zařízení, či programu, kterým je fotografie zpracována, přidat vlastní informace.

Základními daty, které je možné uchovávat v XMP jsou: popis, autor, předmět fotografie, vydavatel, datum, typ, formát, apod. Díky rozšířitelnosti XMP si pak další data může uživatel přidávat sám.

Výhodou, oproti předchozím typům metadat, je větší podpora typů souborů, která není omezena na typy jpeg a tiff, ale podporuje spoustu dalších, jejichž výčet je uveden na stránce [5], ze které bylo, mimo specifikace ([21]), v této části čerpáno. Na stránce [5] jsou také uvedeny informace o podpoře formátu XMP v současných programech apod.

Kapitola 3

Požadavky na výsledný program

V této kapitole jsou uvedeny požadavky, které byly kladeny na výsledný program, jež je možné rozdělit do dvou následujících částí.

1. Základní požadavky, které byly kladeny na výsledný program, jsou následující:
 - Funkčnost pod operačním systémem GNU/Linux.
 - Zobrazování, editace a odstraňování metadat typu EXIF a IPTC u fotografií ve formátu jpeg.
 - Načítání fotografií a zobrazování náhledů.
 - Rozšířitelnost pomocí zásuvných modulů.
 - Vytvoření vzorového zásuvného modulu, který umožňuje bezztrátovou rotaci fotografie.
 - Vyhledání fotografií.
2. Požadavky uvedené výše byly při implementaci programu blíže specifikovány, nebo byly přidávány požadavky nové, které zvýšily použitelnost a funkčnost výsledného programu (která je popsána v kapitole 6). Viz. následující výčet.
 - Přizpůsobování výšky náhledů výšce buněk (možný byl i opačný postup, tj. přizpůsobování výšky buněk výšce náhledům).
 - Zobrazování GPS lokace na mapě a zadávání či úprava GPS souřadnic pomocí téže mapy, nikoliv pomocí textových údajů.
 - Zajištění modality dialogového okna, které je vytvořeno určitým zásuvným modulem.
 - Vyhledávání pomocí GPS souřadnic v určitém rozsahu pomocí mapy, opět nikoliv jen pomocí textových údajů.
 - Možnost návratu k předchozímu a posuv na následující vyhledávání (funkce `undo` a `redo`).
 - Dva typy vyhodnocování vyhledávání.
 - Promítání fotografií.
 - Zobrazování základních informací po kliknutí na načtenou fotografii a při promítání fotografií.

Kapitola 4

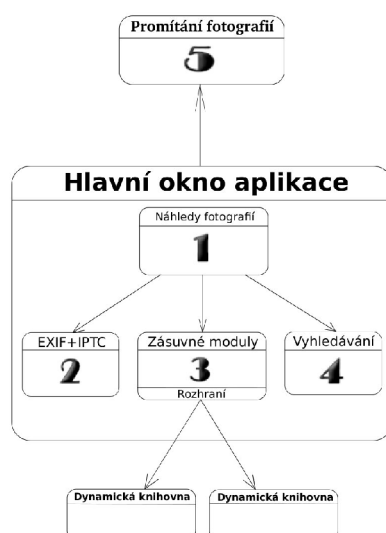
Návrh programu

Kapitola se zabývá zvolenými technologiemi pro realizaci programu a samotným návrhem výsledného programu na jisté, poměrně velké, úrovni abstrakce, tj. nezachází příliš do detailů.

Co se zvolených technologií týče, bylo rozhodnuto, že výsledný program bude určený pro operační systém Linux (což vyplývá i ze zadání) a že bude naprogramován v programovacím jazyce C++ (další možností bylo využití například jazyka Python) s využitím OOP (objektově orientovaného přístupu), který byl zvolen kvůli předešlým zkušenostem a jeho velké rozšířenosti, a pomocí Frameworku Qt [15], který poskytuje velmi užitečné nástroje pro návrh grafického uživatelského rozhraní a mnoho předdefinovaných tříd.

Po analýze požadavků na výsledný program (3) bylo rozhodnuto, že se bude skládat z pěti na sobě nezávislých vzájemně různě propojených částí, které si mezi sebou budou vyměňovat malé množství informací (většinou pouze ukazatele, nebo textové řetězce). Z nichž první čtyři budou obsaženy v jednom, hlavním, okně aplikace.

Celkový návrh programu je znázorněn na obrázku 4.1, kde jednotlivé bloky představují jednotlivé části programu.



Obrázek 4.1: Návrh programu HViewer.

Jako první byla zvolena část věnovaná náhledům fotografií, do které bylo nutné zakomponovat, mimo samostatné náhledy, ovládací prvky celé aplikace. Ta byla tedy navržena jako část hlavní, řídicí, propojená se všemi částmi ostatními. Také bylo určeno, že samotné načítání fotografií je nutné provádět ve speciálním vlákne, aby se výsledná aplikace po dobu načítání fotografií nestala nepoužitelnou.

Druhou částí bylo zvoleno rozhraní pro správu zpracovávaných metadat (EXIF a IPTC). Samotná část pak byla intuitivně rozdělena podle typu metadat na podsekcí spravující metadata typu EXIF a podsekcí spravující metadata typu IPTC. Co se týče voleb u jednotlivých typů metadat, tak byl program inspirován aplikací *digiKam*. Díky možnosti využívání externích nástrojů v operačním systému Linux byl odstraněn problém získávání metadat přímo z fotografií, vznikla ale nutnost zpracování výstupů zmíněného externího nástroje, kterým se stal program `exiv2` ([25]), který zmíněná metadata vypisuje na standardní výstup. Vhodným kandidátem byl ještě program `exiftool` ([10]). `Exiv2` byl zvolen především kvůli lepšímu formátu výstupu na základě vhodných přepínačů a kvůli vyhovující funkčnosti. Již v návrhu programu bylo jasné, že samotné volání programu `exiv2` bude probíhat pomocí funkce `system()`, jejíž úplný funkční prototyp je `int system(const char *command)` a je obsažena v hlavičkovém souboru `stdlib.h`.

Funkce `system()` vykoná jakýkoliv příkaz určený parametrem `command` nezávisle na aplikaci s využitím libovolných dostupných systémových prostředků, jako jsou vlákna či procesy. Úplný popis funkčnosti je dostupný v rámci manuálových stránek (viz. `man system`).

Jako třetí část byla vybrána část pro práci se zásuvnými moduly. Po prostudování článku [26], který se věnuje zásuvným modulům, bylo rozhodnuto, že samotný zásuvný modul bude tzv. dynamická knihovna a že výsledná aplikace bude muset obsahovat rozhraní pro načítání zmíněných dynamických knihoven. Po seznámení se s podporou zásuvných modulů přímo ve zmíněném frameworku [14] a po prozkoumání programů [12] a [13], které se týkají uvedené problematiky a jejichž zdrojové kódy jsou volně dostupné (ať na internetových stránkách, nebo jako příklady ve Frameworku Qt), byla ucelena představa, jak bude rozhraní pro načítání zásuvných modulů vytvořeno a jak budou samotné zásuvné moduly realizovány (což je obecněji popsáno v následujících dvou odstavcích).

Tvorba rozhraní pro načítání zásuvných modulů

K tvorbě rozhraní pro načítání zásuvných modulů, je nutné, v rámci aplikace (vytvářené pomocí Frameworku Qt), ke které budou zásuvné moduly připojovány, vytvořit zvláštní třídu, která bude obsahovat pouze virtuální metody. Ukázka takové třídy je v tabulce 4.1. Aplikace pak ještě musí obsahovat speciální makro `Q_DECLARE_INTERFACE()` a funkci pro načtení zásuvného modulu, která obsahuje instanci třídy `QPluginLoader` a funkci `qobject_cast()`. Implementace funkce je uvedena v tabulce 5.5.

Tvorba zásuvných modulů

K vytvoření zásuvného modulu (pomocí Frameworku Qt), aneb knihovny, která bude k programu připojována tzv. „za běhu“, je nutné dodržet následující podmínky.

1. Ve zmíněné knihovně vytvořit třídu, která dědí ze třídy `QObject` a ze třídy, kterou je realizováno rozhraní (v našem případě ze třídy `Interface`).
2. Ve třídě, vytvořené v bodě 1, použít makro `Q_INTERFACES()`.


```

class Interface
{
public:
    //Destruktor, který nemusí být více využívaný
    virtual ~Interface() {}
    //Metody, pro ovládání zásuvných modulů
    virtual void Start() = 0;
    virtual void Stop() = 0;
    virtual int CheckStatus() = 0;
};

```

Tabulka 4.1: Ukázka třídy, realizující rozhraní pro připojování zásuvných modulů.

3. Exportovat zásuvný modul pomocí dalšího speciálního makra, `Q_EXPORT_PLUGIN2()`.
4. Sestavit program tak, aby výsledkem nebyl spustitelný soubor, ale dynamická knihovna, což je zajištěno úpravou tzv. `.pro` souboru, která spočívá především změnou řádku `TEMPLATE = app` za `TEMPLATE = lib` a přidáním `CONFIG += plugin`.

Ukázka třídy, která by mohla být obsažena v zásuvném modulu, je v následující tabulce 4.2. Detailnější návod, společně s příklady, jak vytvořit zásuvný modul a rozhraní, které jej bude načítat, je popsán na zmíněné internetové stránce [14], ze které jsem v posledních dvou odstavcích čerpal.

```

class Plugin : public QObject, public Interface
{
    Q_INTERFACES(Interface)
public:
    //Stejné metody jako v rozhraní Interface
    void Start();
    void Stop();
    int CheckStatus();
};

```

Tabulka 4.2: Ukázka třídy v zásuvném modulu, aneb v dynamicky sestavené knihovně.

Jako předposlední část bylo navrženo vyhledávání. Po prozkoumání ostatních, podobných programů (některé jsou uvedeny v kapitole 7) a jejich voleb, podle kterých je možné vyhledávat, bylo rozhodnuto, že se i vyhledávání nejspíš rozdělí na další podseky (jak se později také stalo – viz. 6.5). Na základě programu *digiKam* (podobně jako u metadat) pak byly zvoleny dva typy vyhodnocení vyhledávání a vybrány některé volby vyhledávání. Také, podobně jako u načítání náhledů, bylo rozhodnuto, že samotné vyhledávání bude probíhat pomocí speciálního vlákna, aby neomezovalo běh celé aplikace.

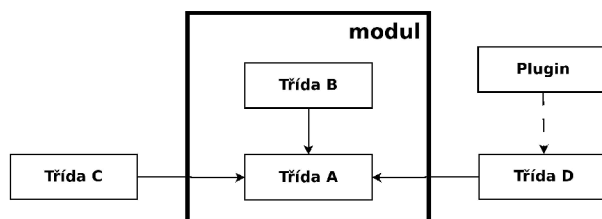
Poslední částí programu se stalo promítání fotografií, které bylo mírně inspirované programem *Eye of GNOME*. Pro promítání fotografií pak bylo navrženo nové okno, aby neprobíhalo v hlavním okně aplikace a bylo více přizpůsobitelné.

Kapitola 5

Implementace programu

V této kapitole se budeme zabývat implementací jednotlivých částí programu HViewer, které byly uvedeny v kapitole 4. U každé z jednotlivých částí bude popsáno, jakým modulům, tzn. souborům, ve výsledném programu, zmíněná část odpovídá. Dále pak jakými uživatelskými třídami jsou jednotlivé moduly realizovány a k čemu slouží. A jaké třídy jsou ve zkoumaném modulu používány (nemusí být v modulu přímo obsaženy). Také bude vždy zobrazena tzv. hierarchie tříd, tj. grafické znázornění propojení jednotlivých tříd, které se zmíněného modulu týkají (na konci kapitoly pak hierarchie modulů celého programu). Budou také uvedeny nějaké implementační detaily, případně metody, které jsou pro zmíněnou část klíčové.

Jednoduchý vzorový příklad hierarchie tříd, týkající se modulu s názvem `modul`, je znázorněn na obrázku 5.1. Modul obsahuje třídy A a B a využívá instance tříd C a D. Plné čáry zakončené šipkami směřující od zmíněných tříd B, C a D ke třídě A určují, že jsou jejich instance využity právě ní (ve třídě A) a že jsou součástí samotného programu. Čárkovaná čára vedoucí od třídy `Plugin` ke třídě D určuje, že je třída `Plugin` pomocí třídy D k programu připojena, ale není jeho součástí (tzn. je obsažena v externí knihovně). Tato syntaxe je dodržena v rámci celého dokumentu.



Obrázek 5.1: Vzorový příklad hierarchie tříd týkající se modulu `modul`.

5.1 Grafické uživatelské rozhraní

Celé grafické uživatelské rozhraní aplikace, tj. hlavní okno aplikace i okno pro promítání fotografií, je vytvořeno pomocí nástroje Qt Designer, který je součástí Qt Frameworku. Samotná implementace grafického rozhraní hlavního okna aplikace je umístěna v souboru `mainwindow.ui`. Implementace okna určeného pro promítání fotografií pak v souboru `slideshow.ui`. Vnitřní struktura zmíněných souborů odpovídá formátu XML v. 1.0, jehož

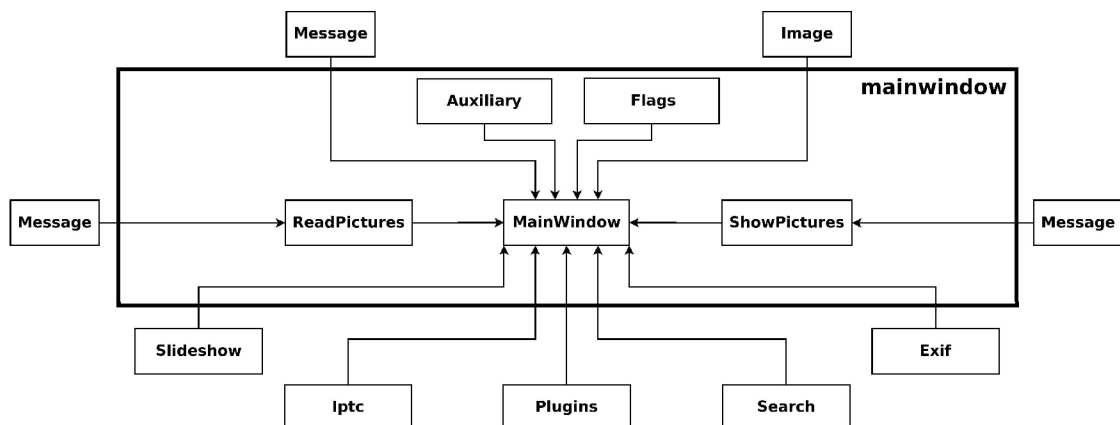
specifikace je dostupná na internetové stránce [16]. Veškeré ikony a grafické prvky jsou pak načítány pomocí souboru `graphics.qrc`

5.2 Náhledy fotografií

Implementace náhledů fotografií, aneb hlavní obrazovky aplikace, je realizována soubory `mainwindow.cpp` a `mainwindow.h`, které obsahují rozhraní a metody třídy `MainWindow`, což je hlavní třída celé aplikace, obsahující, mimo jiné, všechny metody pro ovládání grafického uživatelského rozhraní. Další třída, jež je v souborech obsažena, je `ShowPictures`, která slouží ke zobrazování načtených fotografií. Zmíněné načítání fotografií pak probíhá ve vlastním vlákne, aby neomezovalo běh aplikace, a je realizováno třídou `ReadPictures`. Další třída, obsažená v modulu `mainwindow`, je `Auxiliary`, která, podobně jako třída `ReadPictures`, spouští vlastní vlákno, a slouží k zobrazování informací o vybrané fotografii a informacím zobrazovaným při promítání fotografií. Poslední obsažená třída je třída `Flags` určená pro správu různých příznaků.

Dalšími třídami, které jsou v modulu používány, jsou třídy `Message`, `Slideshow`, `Plugins`, `Exif`, `Iptc`, `Search` a `Image`, tj. téměř všechny třídy aplikace. Třída `Message`, umístěná v souborech `message.h` a `message.cpp`, slouží k zobrazování kritických a informačních zpráv v rámci celé aplikace. Ostatní zmíněné třídy budou popsány později.

Hierarchie tříd, které jsou s modulem `mainwindow` spojeny, tj. jsou v něm přímo obsaženy, nebo v něm používány, je na obrázku 5.2.



Obrázek 5.2: Hierarchie tříd modulu `mainwindow`.

Nejdůležitějšími metodami v modulu `mainwindow` jsou metody následující.

- `ShowImagesInFilelist()` – metoda určená k řízení zobrazování náhledů, spuštění vlákna a přípravě tabulky, do které jsou náhledy vkládány.
- `ShowInfoAboutFiles(row, column)` – metoda ke zobrazování informací o vybrané fotografii. Parametry `row` a `column` slouží k výpočtu indexu do seznamu souborů.
- `SaveConfiguration()` – metoda, která slouží k uložení současné konfigurace hlavního okna, jeho velikosti, pozice a názvů fotografií, jejichž náhledy jsou právě zobrazeny.

- `LoadConfiguration()` – což je metoda s reverzní funkčností metody předešlé, tzn. slouží k načítání minulé konfigurace hlavního okna aplikace.

Některé zmíněné metody jsou uvedeny zjednodušeně, bez datových typů návratových hodnot a parametrů. Zmíněná konfigurace hlavního okna aplikace je ukládána do souboru `saveFile.txt`, ze kterého je poté načítána, jehož vnitřní struktura je uvedena v tabulce 5.1.

| | |
|-----------------|--|
| Záhlaví: | ##### Save file for program HViewer ##### |
| Rozměry: | x:1 y:80 width:1248 height:591 |
| Náhledy: | /home/Pictures/IBP/001.jpg /home/Pictures/IBP/002.jpg /home/Pictures/IBP/003.jpg /home/Pictures/IBP/004.jpg /home/Pictures/IBP/005.jpg /home/Pictures/IBP/006.jpg /home/Pictures/IBP/007.jpg /home/Pictures/IBP/008.jpg /home/Pictures/IBP/009.jpg /home/Pictures/IBP/010.jpg : : |
| Adresář: | /home/Pictures/IBP |
| Zápatí: | ##### Save file for program HViewer ##### |

Tabulka 5.1: Ukázka formátu konfiguračního souboru `saveFile.txt`.

5.3 Metadata EXIF

Implementace týkající se zpracování metadat typu EXIF je obsažena v modulu `exif`, který obsahuje pouze jedinou třídu, `Exif`. Třída `Exif` dále využívá třídu `Message` a třídu `Map`, umístěnou v modulu `gpsmap`.

Třída `Map` slouží k načítání a úpravě HTML kódu (metoda `returnHtmlCode`), ke zobrazování GPS souřadnic na mapě (metoda `ShowHtmlInWebView`) a k získávání hodnot z návratového řetězce od okna (metoda `ParseValue`), ve kterém je mapa zobrazována. Samotný HTML kód, který obsahuje velkou část kódu v JavaScriptu, je umístěn v souboru `maps.html`, který je optimalizován pro načítání ve třídě `Map`, tudíž obsahuje minimální počet bílých znaků a žádné komentáře. Totožný, lépe naformátovaný, soubor s detailními komentáři se jmenuje `mapsOriginal.html`, který není v programu nijak využíván.

Implementace obou souborů zahrnuje rozhraní pro používání map od společnosti Google ([24]). Jako inspirace při jejich tvorbě posloužil zdrojový kód programu *digiKam*, konkrétně

části pro zobrazování souřadnic, který je volně dostupný na internetové stránce [2]. Přesné označení částí kódu, které byly přebrány, je ve zmíněném souboru `mapsOriginal.html`.

Třída `Exif` obsahuje velké množství metod, z nichž nejdůležitější jsou následující.

- `LoadExifMetadata(filename)`, která, jak její název napovídá, slouží k *načítání* metadat souboru `filename`.
- `SaveExifMetadata(filename)`, která má opačnou funkci než předešlá metoda, tzn. slouží k *ukládání* metadat souboru `filename`.
- `ShowOnMap()`, což je metoda k zobrazování pozice na mapě.
- `InitializeHash()`, která je určená k inicializaci hash tabulky `checkboxesAndCommands`, která obsahuje příkazy programu `exiv2`, který je využíván v celém modulu a slouží k vlastnímu ukládání a načítání metadat pomocí zmíněné (4) funkce `system()`.

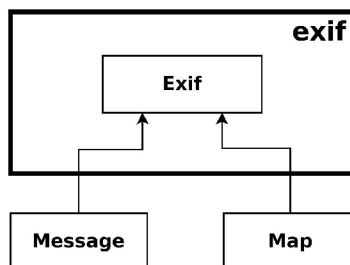
Ukázka používání programu `exiv2` společně s funkcí `system()` je v tabulce 5.2 a slouží k uložení atributu, který je nazvaný `DocumentName`, u souboru, jehož jméno je `filename`.

```
QString command("exiv2 -M \"set Exif.Image.DocumentName docName\" filename");
system(command);
```

Tabulka 5.2: Používání programu `exiv2` a funkce `system()` v modulu `exif`.

Ukázka kódu je opět poměrně zjednodušená, neobsahuje konverzi mezi jednotlivými druhy řetězců, kontrolu návratového kódu funkce `system()`, apod.

Samotná hierarchie zmíněných tříd, týkající se modulu `exif`, je znázorněna na obrázku 5.3.



Obrázek 5.3: Hierarchie tříd modulu `exif`.

5.4 Metadata IPTC

Implementace načítání, úprav a odstraňování metadat typu IPTC je umístěna v modulu `iptc`, tzn. v souborech `iptc.h` a `iptc.cpp`. Vnitřní struktura modulu `iptc` je velmi podobná modulu `exif` (5.3). Obsahuje také pouze jedinou třídu, která se jmenuje `Iptc`, která dále využívá funkčnosti třídy `Message`. Třída `Iptc` obsahuje několik desítek metod, z nichž jsou zřejmě nejdůležitější následující.

- `LoadIptcMetadata(filename)`, sloužící k načítání metadat týkajících se souboru s názvem `filename`.

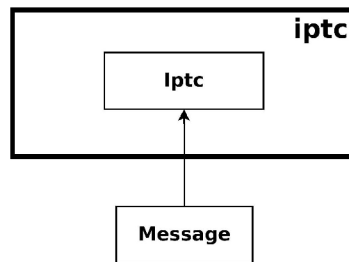
- `SaveIptcMetadata(filename)`, která poskytuje reverzní funkci metody předcházející, tzn. slouží k ukládání iptc metadat týkajících se souboru `filename`.
- `InitializeHash()`, což je metoda, která má stejnou funkci, jako v modulu `exif` (5.3), tzn. slouží k inicializaci hash tabulky `checkboxesAndCommands`, ve které jsou uloženy příkazy programu `exiv2`, který je opět společně s funkcí `system()` využíván v celém modulu.

Ukázka toho, jak je v modulu `iptc` využíván program `exiv2` společně se zmíněnou funkcí `system()`, je uvedena v tabulce 5.3 a je velmi podobná ukázce předešlé (5.2). Úsek kódu slouží k nastavení atributu, který je nazvaný `Headline`, u souboru, jehož jméno je `filename`.

```
QString command("exiv2 -M \"set Iptc.Application2.Headline head\" filename");
system(command);
```

Tabulka 5.3: Používání programu `exiv2` a funkce `system()` v modulu `iptc`.

Hierarchie tříd modulu `iptc` je opět velmi jednoduchá a je na obrázku 5.4.



Obrázek 5.4: Hierarchie tříd modulu `iptc`.

5.5 Zásuvné moduly

Implementace rozhraní pro správu a aplikaci jednotlivých zásuvných modulů je umístěna v modulu `plugins`. Modul `plugins` sestává z jediné třídy `Plugins` a využívá tříd `Message` a `PluginsInterface`. Třída `PluginsInterface` je obsažena ve „stejnojmenném“ souboru `pluginsinterface.h` a slouží jako vlastní rozhraní pro připojování zásuvných modulů. Implementace rozhraní, stejně jako samotné zásuvné moduly, je vytvořena na základě návodu a příkladů, které jsou uvedeny výše, konkrétně v kapitole 4, a obsahuje následující virtuální metody s následující funkcí.

- `CreateGUI(modal)`, která slouží k vytvoření grafického uživatelského rozhraní určitého modulu, tzn. jeho dialogového okna, jejíž parametr `modal` udává, jestli bude zmíněné dialogové okno modální.
- `GetPluginName()`, což je metoda, navracející jméno určitého zásuvného modulu.
- `GetPluginType()`, což je metoda, navracející typ určitého zásuvného modulu, který udává, jestli zásuvný modul pracuje z souborem (typ je `image`), nebo z adresářem (typ je `directory`).

- `GetPluginInfo()`, což je metoda, navracející informace o určitém zásuvném modulu, aneb popis, co zkoumaný zásuvný modul dělá. Poslední tři zmínění metody pak slouží k získání kompletních informací o zásuvném modulu, jejichž výpis je uveden v následující tabulce (5.4).

| |
|--|
| Plugin name: Lossless Rotate Plugin type: Image Plugin info: Plugin pro program HViewer, který provádí bezztrátovou rotaci zvoleného obrazu. |
|--|

Tabulka 5.4: Výpis informací týkající se určitého zásuvného modulu.

- `SetImages(inputImagePath,outputImagePath)`, určená k předání vstupního a výstupního umístění fotografie.
- `SetPreviewLabel(image2preview)`, sloužící k předání ukazatele na label, což je prostor, ve kterém je zobrazen náhled po úpravě fotografie. Náhled je vkládán samotným zásuvným modulem.
- `SetDirectory(directoryPath)`, určená k nastavení aktuálního adresáře.
- `SetNameLabel(image2name)`, sloužící k předání ukazatele na label, ve kterém je zobrazeno jméno upravované fotografie, které může být zásuvným modulem změněno.
- `UpdatePlugin()`, která, jak název napovídá, slouží k aktualizaci informací, které jsou předávány zásuvnému modulu.
- `SetParent(parent)`, která slouží k nastavení tzv. rodiče zásuvného modulu, kterým je hlavní okno aplikace HViewer.

Opět jsou, kvůli jednoduchosti, uvedeny spíše názvy metod, nikoliv úplné funkční prototypy. Samotná třída `Plugins` neobsahuje příliš mnoho metod. Nejdůležitějšími jsou následující uvedené.

- `LoadPluginsNames()`, která slouží k automatickému načtení seznamu zásuvných modulů, tj. souborů s příponou `so` z adresáře `plg`.
- `LoadPlugins(pluginName)`, což je metoda učená k samotnému načtení určitého zásuvného modulu, podle jeho jména `pluginName`, které je uloženo v seznamu, tzn. je načteno metodou předešlou. Ukázka toho, jak je načítání realizováno přímo v programu, je zobrazena v tabulce 5.5.
- `ApplyPlugin()`, což je metoda, sloužící ke spuštění zásuvného modulu, využívající jeho rozhraní standardním způsobem (například tvorba dialogového okna pluginu je spuštěna příkazem `pluginInterface→CreateGUI(modal)`).

Následuje diagram tříd modulu `plugins`, společně se třídou `LossLessRotate`, který je na obrázku 5.5.

```

QPluginLoader pluginLoader(pluginNameAbsolutePath);
QObject *plugin = pluginLoader.instance();
if (plugin){
    //Instance třídy PluginsInterface.
    pluginInterface = qobject_cast<PluginsInterface *>(plugin);
    if(pluginInterface){
        //Zásuvný modul v pořádku načten.
        return true;
    }
}
//Zásuvný modul nebyl načten.
return false;

```

Tabulka 5.5: Načtení zásuvného modulu, jehož absolutní umístění, v rámci souborového systému, udává řetězec `pluginNameAbsolutePath`.

5.5.1 Bezeztrátová rotace

Vzorový zásuvný modul, který je určen k programu HViewer, se jmenuje LossLess Rotate a je realizován dynamickou knihovnou `libLossLessRotate.so`. Vlastní implementace pluginu je pak v souborech `losslessrotate.h` a `losslessrotate.cpp`.

Modul `losslessrotate` je poměrně jednoduchý, obsahuje jedinou třídu `LossLessRotate`, která využívá třídy `PluginsInterface`, kterou je definováno rozhraní mezi aplikací a zásuvným modulem, o kterém musí zásuvný modul „vědět“. Třída `LossLessRotate` dále obsahuje naprosto stejné metody, které jsou uvedeny u popisu rozhraní `PluginsInterface` v kapitole 5.5 a to právě kvůli kompatibilitě.

Samotná rotace obrazu je pak prováděna pomocí programu `exiftran` a již známe funkce `system()`. Ukázka použití je v následující tabulce (5.6). Funkce použitých parametrů je následující.

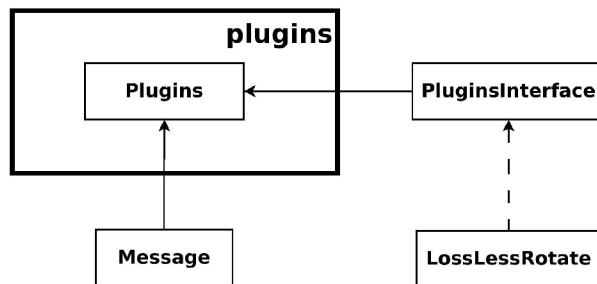
1. Parametr `-p` zajistí zachování tzv. časové značky souboru.
2. `-g` slouží k aktualizaci náhledu fotografie.
3. Parametr `-i` pak k vypnutí výstupního souboru, kdy se výstupním souborem stává soubor vstupní, tj. `inputImage`.
4. `degrees`, který může nabývat hodnot `-9`, pokud chceme rotovat o 90° , `-1`, pokud chceme rotovat o 180° , `-2`, pokud chceme rotovat o 270° a hodnoty `-a`, pokud chceme k rotaci využít metadat fotografie.

```

QString command("exiftran -pgi degrees inputImage");
system(command);

```

Tabulka 5.6: Používání programu `exiftran` a funkce `system()` v zásuvném modulu pro bezeztrátovou rotaci.



Obrázek 5.5: Hierarchie tříd modulu `plugins` se třídou `LossLessRotate`.

5.6 Vyhledávání fotografií

Implementace vyhledávání fotografií se nachází v modulu `search`, který obsahuje jednoduchou třídu `Image`, která slouží k reprezentaci jednoho náhledu. Dále třídu `AuxiliaryThread`, která, jak její název napovídá, využívá speciální vlákno (`thread`) a je určena k vlastnímu vyhledávání. Poslední, hlavní třídou, je třída `Search`, která `AuxiliaryThread` využívá.

Modul `search` pracuje také s třídami `Message`, jejíž význam je známý, `Map` z modulu `gpsmap`, jejíž funkčnost je popsána výše (5.3) a třídou `SearchInformation`, umístěnou v souboru `undo+redo.h`, která je využívána u funkcí `undo` a `redo`, kterým je věnována podkapitola 5.6.1.

Modul `search` obsahuje několik klíčových metod, jejichž výčet je následující.

- `ApplyCommand(command, outFile, output, filename, type)`, což je metoda, která je využívána u každé z voleb vyhledávání a slouží k aplikaci příkazu `command`, který se týká metadat `type` u souboru `filename`. Výsledek je pak uložen do výstupního parametru `output`. Parametr `outFile` je pomocný soubor. Příklad využití metody je v tabulce 5.7.

```

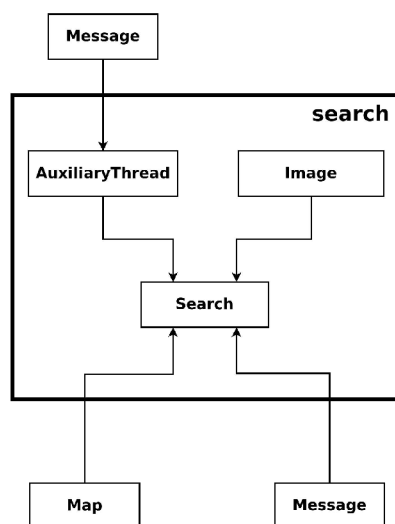
outFile("txtDirectoryPath/search_white");
command("exiv2 -PEt filename -g Exif.Photo.WhiteBalance > outFile);
ApplyCommand(command, outFile, output, filename, "WhiteBalance");
  
```

Tabulka 5.7: Použití metody `ApplyCommand`.

- `AddItemToResultListReciver(filename, image, list)`, která přijímá, požadavky na přidání souboru `filename` s ikonou `image` do seznamu `list`.
- `AddItemToResultListSender(filename, image, list)`, která požadavky metodě předešlé odesílá.
- `CreateMapWindow()`, využívána k vytvoření nového okna s mapou a k načtení počátečních informací.
- `SetSearchInformation(SearchInformation)`, což je metoda, využívána funkcemi `undo` a `redo`, která slouží k *uložení* aktuálních vyhledávacích informací.

- `GetSearchInformation(SearchInformation)`, což je metoda, podobně jako metoda předešlá, využívaná funkcemi `undo` a `redo`, která slouží k načtení požadovaných vyhledávacích informací.

Hierarchie tříd modulu `search` je znázorněna na obrázku 5.6.



Obrázek 5.6: Hierarchie tříd modulu `search`.

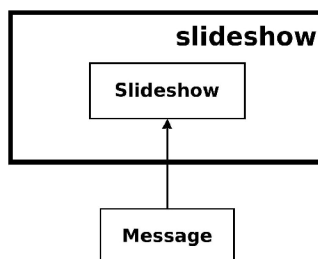
5.6.1 Funkce `undo` a `redo`

Jak již bylo zmíněno, funkce slouží k návratu k předchozímu vyhledávání, nebo posuvu na další vyhledávání, pokud je to možné. Jednotlivé volby vyhledávání jsou uloženy ve třídě `SearchInformation`, jejichž instance jsou vkládány do vektoru `SI`, který je deklarován jako `QVector<SearchInformation>SI`, čímž jsou jednotlivé volby uchovány. K uložení a načtení jednotlivých voleb slouží dvě naposledy zmíněné metody v kapitole předešlé, tj. `SetSearchInformation` a `GetSearchInformation`. Maximální velikost vektoru je programově ošetřena na deset prvků, tzn. není možný návrat více jak o deset vyhledávání zpět, nebo o deset vyhledávání vpřed.

5.7 Promítání fotografií

Implementace je umístěna v modulu `slideshow`, který obsahuje pouze jednu třídu, `Slideshow`, a využívá třídu `Message`. Co se týče jednotlivých metod, je třída poměrně jednoduchá. Obsahuje převážně metody pro ovládání prezentace. Pro její spuštění (`StartSlideshow()`), pozastavení (`PauseSlideshow()`), přechod na předchozí (`PreviousPicture()`) a další fotografii (`NextPicture()`), apod.

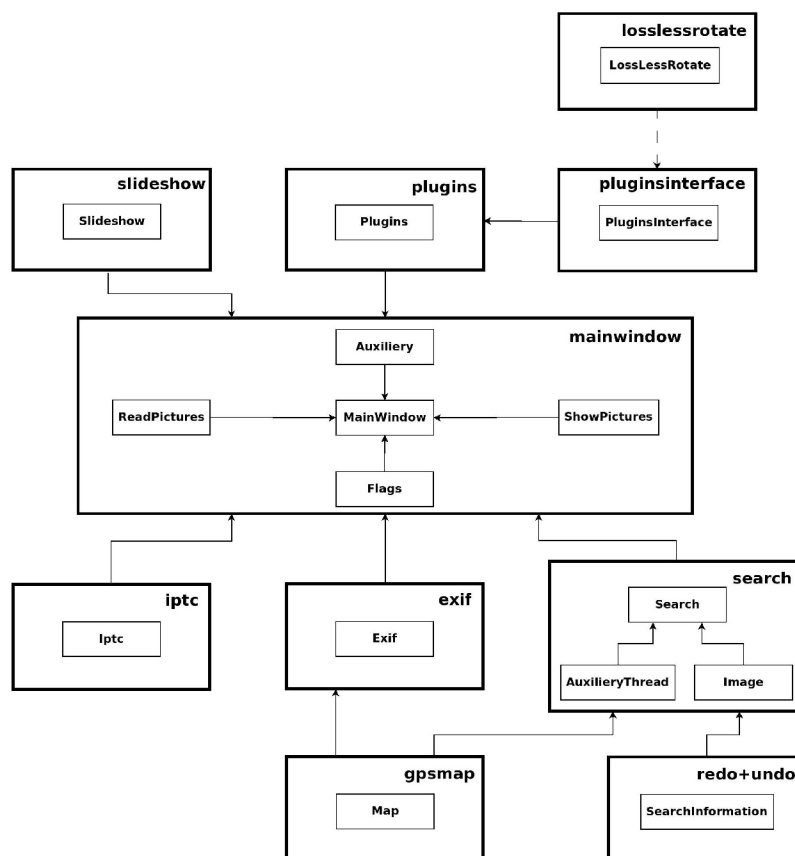
Hierarchie tříd modulu `slideshow` je znázorněna na následujícím obrázku (5.7).



Obrázek 5.7: Hierarchie tříd modulu `slideshow`.

5.8 Hierarchie tříd a modulů

Na obrázku 5.8 je zobrazena hierarchie modulů¹ celého programu HViewer, která obsahuje všechny doposud zmíněné moduly a třídy, které jsou uvedeny na obrázcích 5.2, 5.3, 5.4, 5.5, 5.6 a 5.7. Každý ze zobrazených modulů, kromě modulu `losslessrotate`, který je jako externí plugin od vlastní aplikace implementačně oddělen, navíc využívá třídu `Message` z modulu `message`, která není, pro přehlednost, v obrázku 5.8 zahrnuta.



Obrázek 5.8: Hierarchie modulů programu HViewer.

¹Bloky pojmenované `pluginsinterface` a `redo+undo` neznačí přímo moduly (tj. hlavičkové soubory společně se soubory zdrojovými), ale pouze hlavičkové soubory.

Kapitola 6

Popis výsledného programu

V následující kapitole se detailně seznámíme s výsledným programem HViewer. Budeme se věnovat jednotlivým hlavním částem grafického prostředí programu, které se jmenují:

- Náhledy fotografií.
- Zásuvné moduly.
- Exif a Iptc metadata.
- Vyhledávání.
- Promítání fotografií.

Nejprve se ale dozvíme, jaké programy či knihovny musí být v systému, pro plnou funkčnost programu, obsaženy.

6.1 Požadavky na systém

Pokud požadujeme plnou funkčnost programu, která bude popsána níže, je nutné, aby systém splňoval následující požadavky:

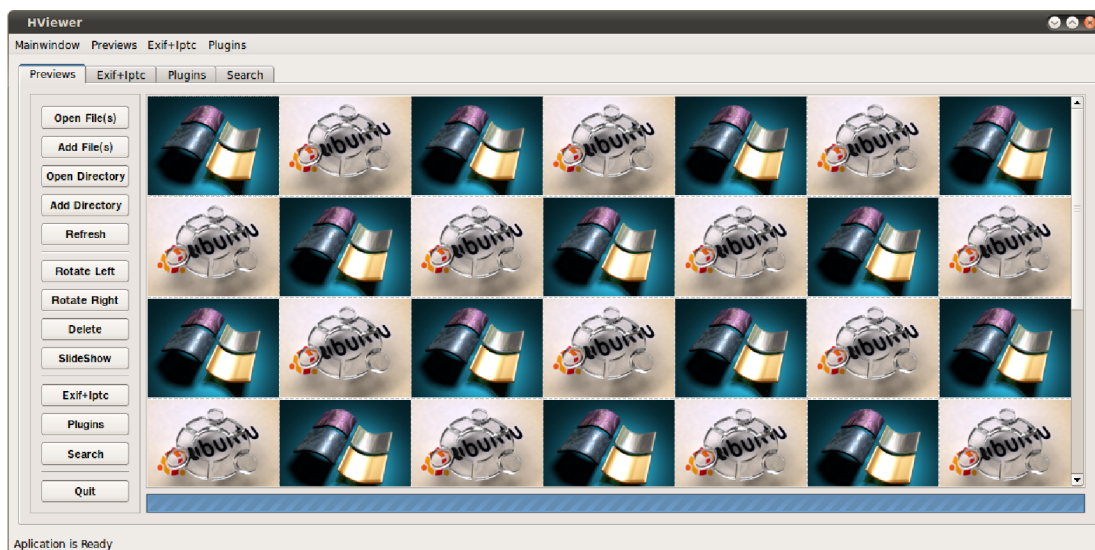
- Qt Framework nebo Qt Libraries v. 4.7.0 a vyšší¹, jež jsou volně dostupné na internetové stránce [15].
- Program `qmake` v. 2.01a, který je automaticky instalován společně s Qt Frameworkem.
- Program `exiv2` [25] pro výpis a zápis metadat u fotografií.
- Program `exiftran`, sloužící k transformacím obrazu.
- Program `jpegtran`, který slouží například k převodu barevného obrazu na šedobílý a podporuje spoustu dalších transformací, jež je obsažen v balíčku `libjpeg-progs`.
- Program pro překlad zdrojových kódů `g++` a program pro kompilaci s názvem `make`.

Všechny výše popsané nástroje jsou uživateli volně dostupné a některé z nich bývají standardní součástí systému, tzn. uživatel je nemusí explicitně instalovat.

¹Dostatečné jsou i nižší verze, není v nich ale zaručen vzhled uvedený v náhledech

6.2 Náhledy fotografií

První část grafického rozhraní, v programu pojmenována Previews, zobrazující náhledy, je k vidění jako první, po spuštění aplikace. Mimo plochy, pro zobrazování náhledů fotografií, obsahuje ovládací prvky, kterými je ovládána celá aplikace. Vzhled rozhraní je možné vidět na obrázku 6.1.



Obrázek 6.1: Zobrazení náhledů fotografií.

Funkčnost jednotlivých ovládacích prvků, které jsou v hlavním okně aplikace zakomponovány, je uvedena v následujícím výčtu:

- Open File(s) – vyvolá dialog pro výběr a následné *načtení* souborů.
- Add File(s) – vyvolá dialog pro výběr a následné *přidání* souborů.
- Open Directory – vyvolá dialog pro výběr a následné *načtení* obsahu adresáře.
- Add Directory – vyvolá dialog pro výběr a následné *přidání* obsahu adresáře.
- Refresh – aktualizuje náhledy.
- Rotate Left – s vybranou fotografií provede rotaci o 90° doleva.
- Rotate Right – s vybranou fotografií provede rotaci o 90° doprava.
- Delete – odstraní vybranou fotografii.
- Slideshow – spustí promítání fotografií, které jsou právě načteny.
- Exif+Iptc – zobrazí metadata první, nebo vybrané fotografie.
- Plugins – s první, nebo s vybranou fotografií, přejde k nabídce načtených zásuvných modulů.
- Search – přepne do části programu sloužící k vyhledávání podle zadaných kritérií.

- Quit – uloží aktuální stav aplikace a uzavře ji.

Na obrázku 6.1 je také možné vidět tzv. **progress bar** [4], který informuje uživatele o procentuálním počtu načtených náhledů fotografií a tzv. **status bar** [6], což je panel, ve kterém jsou zobrazovány zprávy pro uživatele, jež ho informují o činnosti aplikace, nebo o *základních* údajích, týkající se vybraného náhledu.

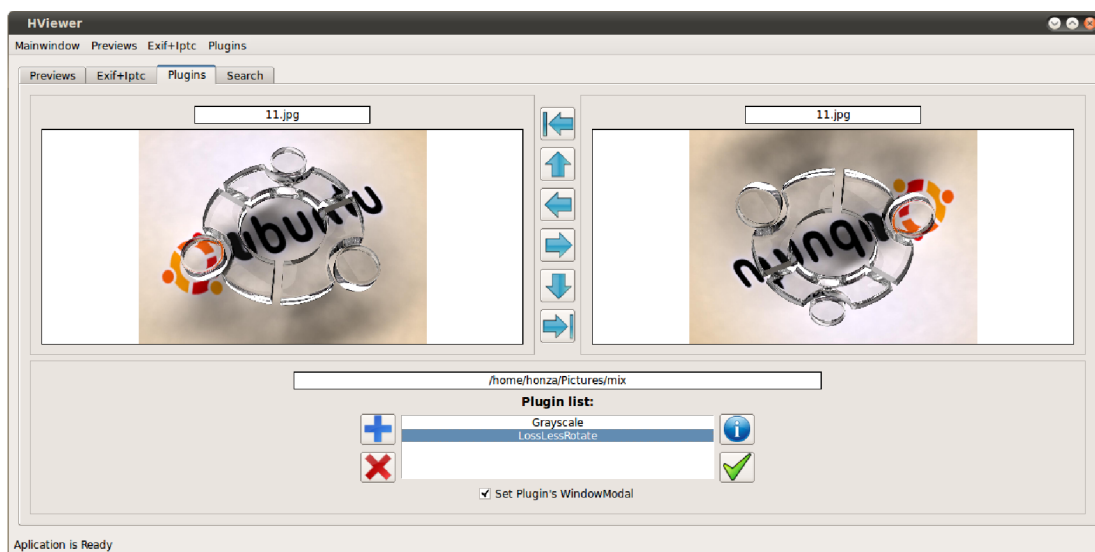
Program zobrazuje dvě zprávy. První, **Application is Ready**, znamená, že je aplikace připravena pracovat. Druhá zpráva, **Application is Running**, vyjadřuje zaneprázdněnost aplikace, tzn. že případné požadavky uživatele nemusí být vykonány a uživateli je zobrazeno okno, informující o stavu aplikace.

Mezi zmíněné základní údaje patří jméno fotografie (**Name**), rozlišení (**Resolution**), cesta k adresáři (**Path**), ve kterém je fotografie umístěna a velikost fotografie (**Size**) v bytech. Celkový výpis pak může vypadat například takto:

Name: ubuntu.jpg, Res: 1600x1200, Path: /home/Pictures/ibp Size: 114557B.


6.3 Zásuvné moduly






Další částí je rozhraní pro práci se zásuvnými moduly, v programu označené anglickým slovem **Plugins**, které je na obrázku 6.2, na němž, mimo jiné, vidíme, že obsahuje dva náhledy zpracovávané fotografie.







Obrázek 6.2: Rozhraní pro práci se zásuvnými moduly (po aplikaci bezetrátové rotace o 180°).

Levý náhled zobrazuje fotografii před aplikací určitého zásuvného modulu, pravý po aplikaci zásuvného modulu. Nad oběma náhledy je zobrazeno jejich jméno, pod nimi pak název adresáře, ve kterém je náhled umístěn. Mezi náhledy se nachází ovládací prvky, jejichž význam je následující:

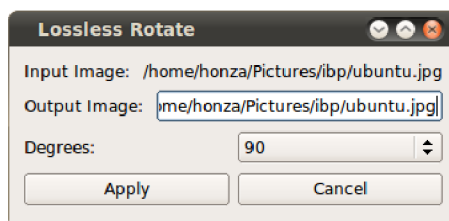
-  – posuv na první náhled.

-  – posuv o deset náhledů zpět, nebo na první náhled, pokud posuv o deset náhledů zpět není možný.
-  – posuv na předchozí náhled.
-  – posuv na další náhled.
-  – posuv o deset náhledů vpřed, nebo na poslední náhled, pokud posuv o deset náhledů vpřed není možný.
-  – posuv na poslední náhled.

Spodní část rozhraní je určena pro práci se samotnými zásuvnými moduly. Obsahuje jejich seznam, ovládací prvky  pro přidání a  pro odebrání zásuvného modulu,  pro aplikaci vybraného zásuvného modulu a  pro zobrazení informací o vybraném zásuvném modulu.

6.3.1 Bezeztrátová rotace

Prvním zásuvným modulem, který je zobrazen v jejich seznamu, viz. obrázek 6.2, je zásuvný modul umožňující bezeztrátovou rotaci obrazu (Lossless Rotate). Po aplikaci zásuvného modulu je vyvoláno dialogové okno 6.3, které informuje uživatele o vstupním (Input Image) a výstupním (Output Image) souboru a umožňuje zvolit mezi rotací o 90°, 180° a 270°, nebo automatickou rotací, při níž se využívá metadat ve fotografii.




Obrázek 6.3: Dialogové okno zásuvného modulu pro bezeztrátovou rotaci obrazu.

6.3.2 Grayscale

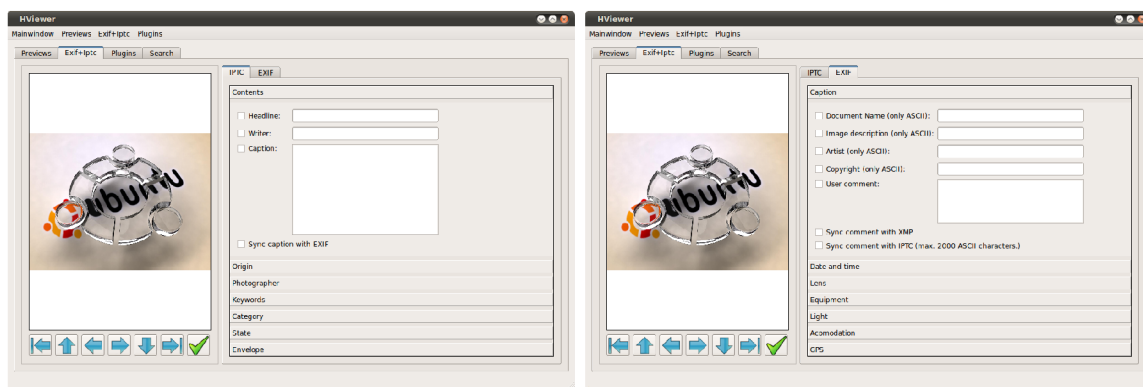
Druhý zásuvný modul slouží k převodu barevného obrazu na obraz ve stupních šedi (anglicky Grayscale). Jeho vstupní rozhraní, tzn. dialogové okno, je velmi podobné dialogovému oknu zásuvného modulu pro bezeztrátovou rotaci, které je uvedeno na obrázku 6.3, na rozdíl od něj, ale neobsahuje žádnou dodatečnou volbu, jako je stupeň rotace (Degrees). Obsahuje tedy pouze název vstupního a výstupního souboru a ovládací prvky pro aplikaci zásuvného modulu a uzavření dialogového okna.

6.4 Exif a Iptc metadata

Další část programu, určená pro zobrazování, úpravu, přidávání a odebrání metadat u fotografií, obsahuje náhled fotografie, jejíž metadata se chystáme zpracovávat, pod kterým

jsou umístěny ovládací prvky, jejichž funkce je *totožná* s těmi v části se zásuvnými moduly (6.3), tzn. slouží k posuvu mezi náhledy. Navíc je ještě obsažen ovládací prvek , který slouží k potvrzení změn.

Vlastní část pro práci s metadaty je rozdělena do dvou dalších sekcí a to právě podle formátu metadat, jež je možné spravovat. Pro ucelení představy slouží obrázek 6.4, v jehož levé polovině je zobrazen náhled rozhraní pro práci s IPTC metadaty a v polovině pravé pak rozhraní pro práci s metadaty EXIF.



Obrázek 6.4: Zobrazení rozhraní pro práci s IPTC a EXIF metadaty.

Z náhledu je patrné další členění jednotlivých metadat, jež je nutné, kvůli jejich rozsahu. Metadata IPTC jsou dělena na sekce:

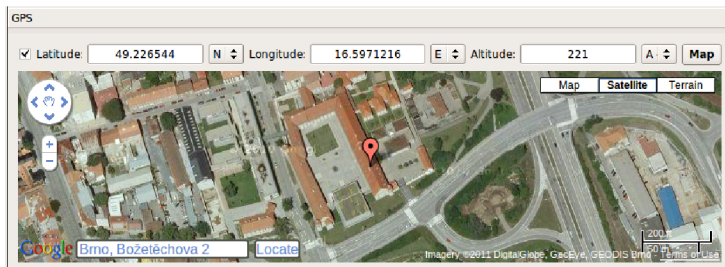
- Contents – popis vzhledu fotografie.
- Origin – popis formálních informací o fotografii (datum, město, ...).
- Photographer – popis informací o fotografovi.
- Keywords – správa klíčových slov.
- Category – zařazení fotografie do kategorií.
- State – popis průběhu činnosti.
- Envelope – popis tzv. redakčních detailů.

Metadata EXIF pak na:

- Caption – popis vzhledu fotografie (detailněji než v IPTC sekci Contents)
- Date and time – zápis dat a časů, týkající se fotografie.
- Lens – popis nastavení čočky.
- Equipment – popis a nastavení zařízení, kterým byla fotografie pořízena.
- Light – informace o zdroji světla, při pořizování fotografie.
- Accommodation – přizpůsobení vlastností fotografie.
- GPS – zadání, změna či odstranění GPS souřadnic a zobrazení na mapě (viz. 6.4.1).

6.4.1 GPS

Za zvláštní zmínku stojí metadata týkající se GPS souřadnic a jejich zadávání, upravování a zobrazování na mapě. Všem zmíněnému, jak je psáno výše (6.4), se věnuje část programu, jejíž vzhled je možno vidět na obrázku 6.5.



Obrázek 6.5: Práce s GPS souřadnicemi.

Zobrazování map je realizováno pomocí služby *google maps* [18] s využitím *google maps API V2* [24]. GPS souřadnice jsou automaticky načteny (jestliže jsou ve fotografii obsaženy) a výsledná poloha je zobrazena na mapě. Při jejich editaci, tzn. při změně některé z hodnot určující polohu fotografa, jsou změny automaticky přeneseny do mapy. Zadání nových GPS souřadnic je také možné kliknutím na určité místo na mapě, nebo přetažením ukazatele do mapy na nové místo. V tomto případě jsou pak nové hodnoty automaticky přeneseny směrem z mapy do rozhraní aplikace.

Samotná mapa pak obsahuje ovládací prvky pro změnu typu mapy, kde je na výběr ze čtyř druhů (základní, satelitní, hybridní a terénní), pro pohyb na mapě, který je možný i tzv. metodou *drag & drop*, pro přibližování a pro vyhledávání. V levém spodním rohu je ještě možné vidět tzv. *scale control*, který informuje o měřítku mapy.

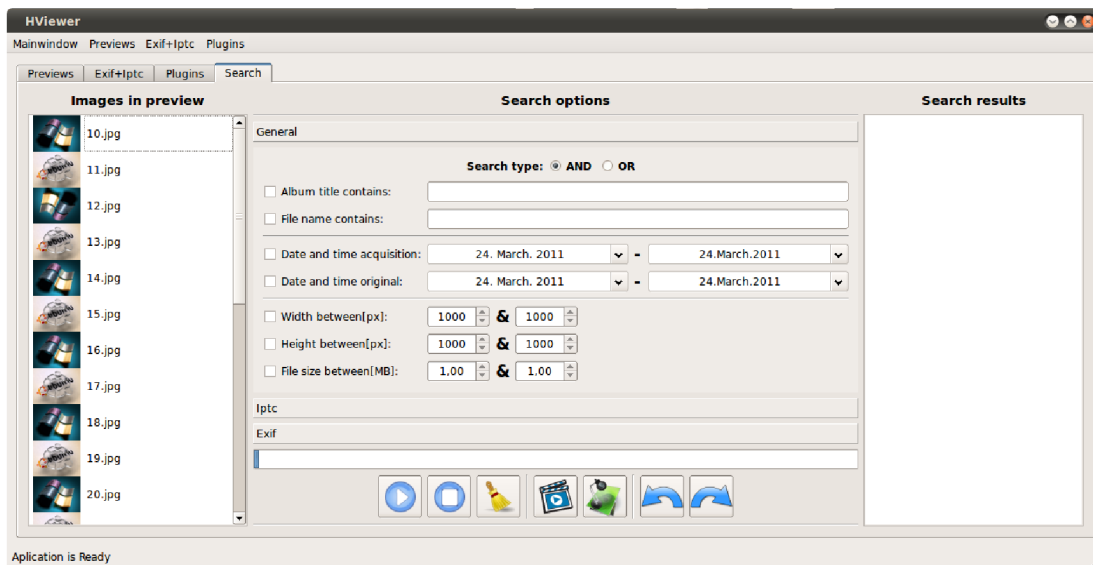
6.5 Vyhledávání

Předposlední hlavní částí programu HViewer je rozhraní pro rozsáhlé vyhledávání fotografií podle zvolených kritérií. Rozhraní ve své levé části obsahuje seznam fotografií, jejichž náhledy jsou právě načteny, v pravé části pak seznam fotografií, které vyhovují nastavenému vyhledávacímu filtru (viz 6.6).

Mezi oběma seznamy se nachází jednotlivé volby vyhledávání, které může uživatel vložit a typ vyhledávání, který může být AND, nebo OR. V případě prvního typu vyhledávání, typu AND, musí fotografie vyhovovat *všem* zadaným volbám aby byla zařazena do seznamu nalezených fotografií. V případě druhém, při vyhledávání typu OR, stačí, když fotografie vyhovuje alespoň *jedné* z voleb. Volby jsou pro přehlednost rozděleny do tří částí, z nichž první nese název *General* a obsahuje obecné volby vyhledávání.








Druhá část, pojmenovaná *lptc*, obsahuje volby, které jsou podmnožinou IPTC metadat, jež je možné zadat v sekci 6.4.

Poslední část vyhledávání se jmenuje *Exif* a podobně jako část druhá, obsahuje volby, které jsou podmnožinou EXIF metadat, jež je možné zadat v sekci 6.4. Obdobně, jako je tomu v kapitole 6.4, tak i tady je detailněji zmíněna práce s GPS souřadnicemi, pomocí kterých je možno vyhledávat. Vše je popsáno v podkapitole 6.5.1.



Obrázek 6.6: Zobrazení rozhraní pro vyhledávání fotografií.

Poté následují ovládací prvky s následujícím významem:

-  – spustí vyhledávání fotografií podle zadaných voleb.
-  – zastaví právě probíhající vyhledávání.
-  – odstraní všechny volby předchozího vyhledávání a dříve nalezené fotografie v seznamu.
-  – spustí promítání fotografií, které jsou obsaženy v seznamu nalezených fotografií, tj. odpovídají zadaným volbám vyhledávání.
-  – zobrazí náhledy fotografií, které jsou obsaženy v seznamu nalezených fotografií, tj. odpovídají zadaným volbám vyhledávání.
-  – vrátí se k předchozímu vyhledávání, pokud je to možné, a provede jej.
-  – přejde na následující vyhledávání, pokud je to možné, a provede jej.

6.5.1 Využití GPS souřadnic

V programu HViewer je možné vyhledávat pomocí zeměpisné šířky, zeměpisné délky i pomocí nadmořské výšky. Ke každé z hodnot je možné navíc přidat rozsah v jakém se bude vyhledávat. Celkový prostor vyhledávání je pak tvořen buď přímkou, pokud je vyhledáváno pouze pomocí jedné ze souřadnic, nebo obdélníkem, pokud je vyhledáváno pomocí dvou souřadnic, nebo kvádrem, pokud je vyhledáváno pomocí všech tří souřadnic.

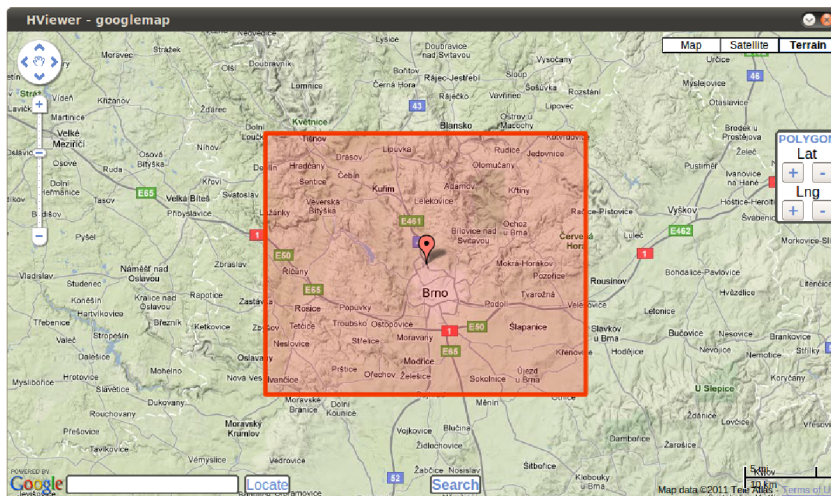
Maximální rozsah, nastavitelný k zadané nadmořské výšce je 500 m ($+250\text{ m}$ a -250 m) při kroku 1 m . Rozsah, který je možný nastavit k zeměpisné šířce a délce, je 2° ($+1^\circ$ a -1°) od zadaného bodu, kdy nastavování probíhá pomocí menších jednotek, minut, s krokem jedna minuta.

Zmíněná jedna minuta zeměpisné šířky odpovídá asi 1850 m (jeden stupeň pak 111 km) a jedna minuta zeměpisné délky asi 1315 m (jeden stupeň pak 78.8 km). Minimální prostor, který je tedy možný prohledávat (větší než zadaný bod), je asi $2 \cdot 1315\text{ m} \times 2 \cdot 1850\text{ m} \times 2 \cdot 1\text{ m}$. Maximální prostor je asi $2 \cdot 78.8\text{ km} \times 2 \cdot 111\text{ km} \times 2 \cdot 250\text{ m}$. Násobení dvěma je vždy uvedeno kvůli kladné i záporné hodnotě, která se k bodu přičítá. Výpočty jednotlivých délek byly prováděny na základě údajů na internetové stránce [3].

Všechny výše popsané údaje týkající se GPS souřadnic je možné nastavit pomocí jednoduchého rozhraní (6.7),

Obrázek 6.7: Rozhraní pro nastavení GPS souřadnic u vyhledávání.

nebo pomocí mapy (6.8), k jejímuž otevření slouží tlačítko **Use map**, které je možné vidět na obrázku 6.7.



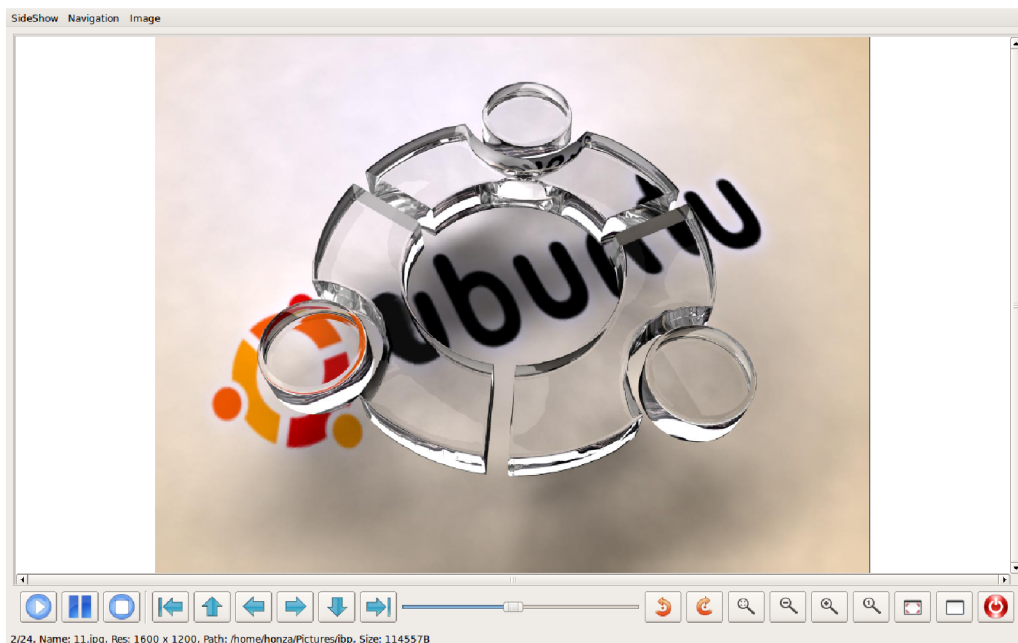
Obrázek 6.8: Mapa pro nastavení GPS souřadnic u vyhledávání.

Prvky obsažené na mapě 6.8 mají stejnou funkci, jako na mapě pro zobrazování polohy fotografa (6.4.1) a jsou zobrazovány *stejným* způsobem (za pomoci *totožných* služeb). Navíc je ještě obsažen ovládací prvek s nápisem **Search**, po jehož stisku se spustí samotné vyhledávání, a ovládací prvky pro změnu velikosti plochy (nastavení rozměrů zobrazovaného polygonu [9]), ve které se bude vyhledávat.

Samotná mapa je propojena s rozhraním 6.7, tzn. jakákoliv změna na mapě je přenesena do zmíněného rozhraní a naopak, jakákoliv změna v rozhraní je přenesena zpět do mapy.






6.6 Promítání fotografií

Poslední částí programu HViewer je rozhraní k promítání fotografií (Slideshow), jejichž náhledy jsou aktuálně načtené, nebo fotografií, které byly právě vyhledány. Velkou část rozhraní, pro promítání fotografií, zabírá vlastní fotografie, pod kterou jsou zobrazeny ovládací prvky a výše zmiňovaný status bar, jehož funkce je totožná jako v okně s náhledy 6.2, jen navíc, na prvním místě, před jménem souboru, zobrazuje aktuální fotografii ku celkovému počtu fotografií, tzn. například číslo 2/24, viz. 6.9.



Obrázek 6.9: Promítání fotografií.

Ovládací prvky fungují následujícím způsobem (funkčnost některých navigačních prvků je podobná jako v sekci se zásuvnými moduly 6.3, nebo v sekci s metadaty 6.4, nicméně je zde uveden úplný výčet):

-  – spuštění promítání, pokud je pozastaveno, nebo zastaveno.
-  – pozastavení promítání.
-  – zastavení promítání a skok na první fotografii.
-  – skok na první promítanou fotografii.
-  – návrat o deset promítaných fotografií, nebo na fotografii první, pokud návrat o deset fotografií není možný.

-  – návrat na předchozí fotografii.
-  – posuv na následující fotografii.
-  – posuv o deset následujících fotografií, nebo na fotografii poslední, pokud posuv o deset fotografií není možný.
-  – posuv na poslední fotografii.
-  – rotace fotografie o 90° doleva.
-  – rotace fotografie o 90° doprava.
-  – nastavení velikosti fotografie tak, aby se na výšku přesně vlešla do okna.
-  – oddálení fotografie.
-  – přiblížení fotografie.
-  – nastavení originální velikosti fotografie.
-  – přepnutí okna do maximálního režimu.
-  – přepnutí okna do celoobrazovkového režimu.
-  – uzavření okna.

Mezi výše popsanými ovládacími prvky se nachází „posuvník“, kterým lze nastavit rychlost promítání, tzn. jakou rychlostí se budou měnit jednotlivé fotografie.

Kapitola 7

Porovnání se stávajícími programy

V této kapitole se budeme věnovat porovnání programu HViewer, jehož funkce jsou popsány v předešlé kapitole (6), s podobnými stávajícími programy, tj. s programy, které jsou dostupné pro operační systém Linux, ve kterých jsou zobrazovány náhledy, které podporují práci s metadaty, vyhledávání fotografií podle různých kritérií a promítání fotografií.

Programy vyhovující většině předchozím kritériím jsou Gwenview, MaPiVi, digiKam a imgSeek. Ty budou porovnávány s programem HViewer v následujících „disciplínách“.

1. Grafické uživatelské rozhraní a náhledy fotografií.
2. Promítání fotografií.
3. Správa metadat.
4. Vyhledávání fotografií.
5. Ostatní funkce.

Samotné porovnání pak nebude vždy přímo ku programu HViewer, ale někdy bude uveden spíše obecnější popis, ze kterého rozdíly jednotlivých programů a programu HViewer vyplývají.

7.1 Grafické uživatelské rozhraní a náhledy fotografií

Gwenview: program je určený pro prostředí KDE [22], což se samozřejmě promítá na jeho celkovém vzhledu. Základní obrazovka je poměrně jednoduchá. Na rozdíl od programu HViewer neobsahuje náhledy fotografií, které byly načteny při minulém vypnutí programu, ale obsahuje historii adresářů (ať už místních, nebo vzdálených), ze kterých byly fotografie v minulosti načítány.

MaPiVi: rozhraní programu je rozděleno do tří svislých bloků. V prvním je tzv. adresářový strom, v prostředním náhledy fotografií, které jsou obsaženy ve vybraném adresáři, společně s krátkými informacemi o metadatech, v posledním pak zvětšený náhled vybrané fotografie. Program MaPiVi, stejně jako program HViewer, načte po spuštění fotografie a název adresáře, jež byly načteny při posledním vypnutí programu.

digiKam: „nejmocnějším“ ze zmíněných programů, je program digiKam. Jeho síla nespočívá ve složitosti grafického rozhraní, nýbrž v množství nabídek a funkcí, které poskytuje

a ve funkčnosti zásuvných modulů. Rozhraní programu, po jeho spuštění, je opět poměrně jednoduché. Na rozdíl od všech porovnávaných programů neobsahuje adresářový strom ani tlačítka, pomocí kterých dochází k výběru souborů či adresářů, ale obsahuje vlastní strukturu alb, která jsou automaticky, společně s náhledy, načtena při každém spuštění. Strukturu alb je možné nadefinovat v nastavení programu. DigiKam je stejně jako Gwenview určen pro prostředí KDE.

imgSeek: po spuštění programu se zobrazí poměrně složité rozhraní, obsahující adresářový strom, prostor pro náhledy fotografií v adresáři a pro jednu zvětšenou fotografii. Program po spuštění nenačítá žádné náhledy, ani naposledy otevřené složky, ale vybrané adresáře, které je možné nadefinovat v nastavení programu.

HViewer: rozhraní programu, po jeho spuštění, je detailně rozebráno, společně s náhledy aplikace, v kapitole 6.2. Možnou nevýhodou (nebo výhodou) rozhraní může být absence adresářového stromu, který byl nahrazen jednotlivými ovládacími prvky, které jsou úspornější, co se týče rozměrů, ale možná méně „pohodlné“. Tento způsob není možné vidět u žádného z porovnávaných programů. Další odlišnost, která může být nevýhodnou (stejně tak jako výhodnou), je absence jakýchkoliv popisků u jednotlivých náhledů, která znamená větší prostor pro samostatné náhledy, ale může ubírat na přehlednosti.

Nejrychlejším programem, co se týče rychlosti prvního načtení náhledů fotografií, je imgSeek, ve kterém jsou náhledy načteny téměř okamžitě. Poměrně rychlé jsou i programy gwenview a HViewer, z nichž těsně vítězí první zmíněný. Velmi pomalé jsou pak programy MaPiVi a digiKam, kde je načítání náhledů asi o řád pomalejší. Všechny zmíněné programy načítají fotografie pomocí speciálního vlákna nebo procesu, takže neomezují běh aplikace.

7.2 Promítání fotografií

Gwenview: obsahuje dva typy promítání fotografií. První je součástí aplikace, druhý je dostupný jako část zásuvného modulu, jehož jméno je KIPI, což je zkratka KDE Image Plugin Interface, který kromě promítání, poskytuje spoustu dalších funkcí, a je využíván i programem digiKam (což je další z jejich společných vlastností), kde jsou jeho „promítací“ schopnosti blíže popsány a je dostupný na internetové stránce [23]. V této části bude popsáno promítání dostupné přímo v aplikaci gwenview.

Při promítání je fotografie vyhrazena celá obrazovka, na které, na rozdíl od programu HViewer, nejsou žádné ovládací prvky. Ty se objeví, společně s řadou náhledů fotografií, které se budou promítat, po přijetí kurzorem myši do horní části obrazovky. Co se samotných ovládacích prvků týče, obsahuje program pouze ty nejzákladnější. Jedním z nich je nastavení parametrů prezentace, z nichž nejzajímavější je možnost nastavit zobrazovaná metadata u právě promítaných fotografií. Nic takového program HViewer neumožňuje.

MaPiVi: jako jediný z porovnávaných programů, nezobrazuje promítané fotografie v novém okně, ale v hlavním okně aplikace, konkrétně v posledním svislém bloku (s rozdělením okna aplikace jsme se setkali v předchozí kapitole 7.1). Promítány jsou pak pouze označené fotografie, namísto všech načtených, jak je tomu u většiny porovnávaných programů, včetně programu HViewer. Program neobsahuje žádné ovládací prvky, týkající se prezentace, ani žádné možnosti nastavení prezentace.

digiKam: promítání fotografií je řešeno pomocí zásuvného modulu KIPI (jak bylo řečeno u programu gwenview). Je možné promítat všechny načtené fotografie, výběr, nebo všechny

fotografie ze všech alb (alba viz. 7.1). Před promítáním je možné nastavit velké množství voleb. Samotné promítání je pak opět v režimu „celé obrazovky“ a obsahuje pouze nejnútnejší ovládací prvky.

imgSeek: spouští prezentaci fotografií v novém okně, které je zobrazeno přes celou obrazovku. Okno ale neobsahuje žádné ovládací prvky ani žádné možnosti nastavení. Právě zobrazovaný obrázek není nijak vystředěný ani optimalizovaný pro velikost okna.

HViewer: detailní popis rozhraní pro promítání fotografií je uveden výše, konkrétně v kapitole 6.6. Ze zmíněných programů obsahuje prezentace nejvíce ovládacích prvků. Obsahuje ale i nejvíce tzv. rušivých elementů, které je při promítání, kromě vlastní fotografie, možné vidět na obrazovce (z aplikací, které promítají obrázky v novém okně). Obsahuje také menší množství nastavení (pouze rychlost promítání), než některé porovnávané programy (gwenview, digiKam).

Nejpropracovanější program, co se podpory promítání fotografií týče, je gwenview, s podporou dvou typů promítání, následuje aplikace pojmenovaná digiKam. Poté program HViewer, s dobrou podporou promítání fotografií. Předposlední „místo“ patří programu MaPiVi a poslední aplikaci imgSeek, v němž si není, co se týče promítání, možné zvolit prakticky nic.

7.3 Správa metadat

Gwenview: zobrazování, editace a odstraňování metadat je opět řešeno pomocí zásuvného modulu KIPi (7.2), který využívá knihovnu LibEXiv2, což je nástavba nad programem exiv2, který je využívána i programem HViewer. Program gwenview kromě metadat IPTC a EXIF spravuje i metadata XMP, od kterých bylo v programu HViewer upuštěno. Gwenview, na rozdíl od programu HViewer, neposkytuje žádné nástroje pro práci s GPS souřadnicemi. Další nevýhodou, oproti programu HViewer, je *nemožnost* změny fotografie při zobrazených metadatach, tzn. není možné posouvat fotografie při zobrazeném dialogovém okně s výpisem metadat.

MaPiVi: poskytuje poměrně dobré rozhraní pro editaci části IPTC metadat. Druhá část rozhraní je nepřehledná a není příliš jasné, jakých hodnot mohou jednotlivé parametry nabývat. Metadata EXIF a XMP je možné pouze zobrazit. MaPiVi neobsahuje žádné nástroje pro správu GPS souřadnic. Fotografiemi při zobrazených informacích není možné listovat.

digiKam: stejně jako první zmíněný program, využívá i digiKam plugin KIPi. Podpora práce s metadaty je tedy velmi podobná. Navíc však pracuje s další funkcí zmíněného pluginu a tou je zobrazování či zadávání GPS souřadnic pomocí mapy. DigiKam opět neumožňuje změnu fotografie, když je zobrazeno okno s metadaty, nebo mapa s polohou objektu.

imgSeek: podporuje pouze metadata typu EXIF, které je možné jak zobrazovat, tak i editovat, ale protože, podobně jako u programu MaPiVi, není příliš jasné, jakých hodnot mohou jednotlivé parametry nabývat, tak se program hodí spíše pro zobrazování, než pro editaci metadat. Výhodou může být, že jako jediný, z doposud popisovaných programů, umožňuje posun fotografií při zobrazeném výpisu metadat. Práce s GPS souřadnicemi opět není podporována.

HViewer: detailní popis rozhraní pro práci s metadaty je uveden v kapitole 6.4. HViewer dokáže pracovat s takřka totožnou množinou metadat, jako programy gwenview a digiKam.

Programu digiKam se navíc vyrovná i zobrazením místa na mapě, kdy je využíváno stejných technologií, a takřka totožného zpracováním GPS souřadnic. Velmi výhodnou funkcí je možnost změny fotografie při zobrazených metadatech, kterou jsme zatím viděli pouze u programu imgSeek.

Pomyslné první místo, co se týče práce s metadaty, získává program HViewer. Následuje program digiKam, za kterým je umístěna aplikace s názvem gwenview. Předposlední pak program imgSeek, který podporuje spíše zobrazování, než plnohodnotnou editaci. Poslední pak MaPiVi, v němž je podpora metadat nejslabší.

7.4 Vyhledávání fotografií

Gwenview: program, jako jediný z porovnávaných, neobsahuje žádné vyhledávání fotografií, což ho v této kategorii, automaticky posunuje na poslední místo.

MaPiVi: obsahuje více typů vyhledávání. Prvním z nich je vyhledávání v tzv. databázi, což jsou fotografie, o nichž má program informace v paměti (například byly v minulosti načítány). Vyhledávat pak lze pomocí několika voleb s využitím regulárních výrazů a to jak na shodnost, tak i neshodnost názvu fotografie se vzorem.

Druhý typ vyhledávání je pomocí klíčových slov, která jsou načítána (ukládána do paměti) společně s fotografiemi. Nejedná se tedy o vyhledávání v pravém slova smyslu, ale spíše o jakýsi výběr klíčových slov a zobrazení fotografií, které vybrané klíčové slovo obsahují. Dalším, velmi zajímavým typem vyhledávání, je vyhledávání pomocí časové osy, kdy je možné, z přehledného histogramu, vybrat měsíc a zobrazit fotografie v něm vytvořené.

Posledním typem je pak vyhledávání podle lokace. Nejedná se však o GPS souřadnice, nýbrž o seznam načtených lokací, které jsou uloženy formou metadat ve fotografiích. Seznam lokací si program, stejně jako klíčová slova, ukládá do paměti. MaPiVi nabízí ještě možnost vyhledání duplicit a vyhledání sta nejlépe hodnocených fotografií. Nic takového, stejně jako regulární výrazy, program HViewer nepodporuje.

digiKam: aplikace digiKam, podobně jako program MaPiVi, poskytuje více typů vyhledávání. První je velmi rozsáhlé. Je možné zvolit velké množství parametrů, týkajících se fotografií a jejich metadat. Navíc obsahuje čtyři typy vyhodnocení vyhledávání, které by bylo možné nazvat AND, OR, $\overline{\text{AND}}$ a $\overline{\text{OR}}$, čímž převyšuje program HViewer (který obsahuje jen dva typy).

Druhé je vyhledávání podle lokace, kdy je možné na modelu zeměkoule, nebo pomocí Open Street Map¹, vybrat úsek, ze kterého chceme fotografie zobrazit.

Třetím typem je tzv. „pružné“ vyhledávání, kdy lze vyhledávat buď duplicitu, podobné fotografie, nebo fotografie podle náčrtku. Program HViewer nemá ani jednu ze zmíněných funkcí.

Další typ vyhledávání, nebo spíše výběru, je podobně jako u programu MaPiVi, pomocí klíčových slov, což vyvolá zobrazení fotografií, které vybrané klíčové slovo obsahují, nebo data, což vyvolá zobrazení fotografií, které byly vytvořeny ve zvolený rok, či měsíc.

imgSeek: podporuje vyhledávání podle parametrů fotografií (jméno souboru, velikost souboru, . . .), podle skupiny a podle obsahu fotografie. Při vyhledávání podle parametrů fotografie a podle skupiny, do které fotografie patří, je možné zvolit typ vyhodnocení, podobně

¹<http://www.openstreetmap.org>

jako v programech HViewer a digiKam, který je přímo v programu nazván AND a OR. Při vyhledávání podle obsahu fotografie je možné vybrat vzor, nebo ho přímo načrtnout. Program navíc podporuje vyhledávání duplicit.

HViewer: vyhledávání fotografií je věnována celá kapitola 6.5. Oproti výše popsaným programům poskytuje program HViewer nejkomfortnější vyhledávání pomocí GPS souřadnic. Také oproti všem programům, kromě programu digiKam, nabízí nejvíce voleb, podle kterých je možné vyhledávat. Nepodporuje ale například vyhledávání duplicit, vyhledávání podle vzoru, podle náčrtku, ani pomocí regulárních výrazů (jak je psáno výše).

Nejpropracovanějším programem, co se týče vyhledávání, je program digiKam, který umožňuje vyhledávat téměř všemi možnými způsoby a podle mnoha kritérií. Druhé místo patří programu HViewer, který má, ze zbylých programů, nepřehlednější rozhraní pro vyhledávání, obsahuje nejvíce voleb a podporuje vyhledávání podle GPS souřadnic (i když neobsahuje tolik typů vyhledávání). O třetí a čtvrté místo se dělí programy MaPiVi a imgSeek, jejichž vyhledávání je sice poměrně rozdílné, ale je velmi těžké posoudit, které je lepší a které horší. Na posledním místě je prvně zmíněný program gwenview, bez podpory vyhledávání.

7.5 Ostatní funkce

Gwenview: program obsahuje velké množství funkcí, které jsou určeny pro práci s fotografiemi (rotace, ořez, zrcadlení, změna velikosti, mazání, redukce červených očí, ...). Navíc využívá zmíněný plugin KIPI, jež mu přidává další funkce (dávkové zpracování, tvorba kalendáře, široké možnosti importu a exportu fotografií, ...) a činí ho plnohodnotným nástrojem pro práci s fotografiemi.

MaPiVi: aplikace dokáže překvapit počtem a rozmanitostí ostatních funkcí, které jsou například změna velikosti, rotace, ořez, převod barevného obrazu na šedobílý apod. (z těch standardních) a porovnání fotografií i adresářů, tisk seznamu souborů, vytvoření vodoznaku apod. (z těch méně standardních). Některé funkce bohužel nejsou příliš dobře zpracovány, což jim ubírá na jejich použitelnosti.

digiKam: digiKam obsahuje stejné funkce jako program gwenview a přidává spoustu dalších. Navíc v maximální míře využívá plugin KIPI. Obsahuje i vlastní editor, kde je možné pracovat s bitovou hloubkou, nastavovat úroveň, provádět různé transformace a vylepšení obrazu apod. . . Je to tedy, jak bylo zmíněno v kapitole 7.1, nejmocnější program, z programů námi porovnávaných.

imgSeek: program, na rozdíl od doposud popsaných, neobsahuje velké množství ostatních funkcí. Obsahuje rozhraní pro transformace fotografií, kdy je možné měnit ostrost, jas, kontrast, ... Dále podporuje tvorbu HTML alb a gqview kolekcí (viz [1] a dokumentace k programu). Nabízí také základní podporu dávkového zpracování, čímž výčet jeho ostatních funkcí končí.

HViewer: poslední z programů neobsahuje takové množství funkcí jako výše zmíněné, ale pouze funkci pro rotaci fotografie a smazání fotografie. Největší výhodou, oproti ostatním programům, je možnost velmi snadno vytvořit téměř libovolný zásuvný modul, tzn. zásuvný modul s téměř libovolnou funkčností. Kapitola, věnující se grafickému uživatelskému rozhraní pro práci se zásuvnými moduly, je 6.3. V kapitole 5.5 je pak popsáno rozhraní (tzn. funkce), které musí zásuvný modul obsahovat, aby byl z programem HViewer kompatibilní.

Program obsahující obecně nejvíce funkcí je samozřejmě digiKam. Na druhém místě je aplikace gwenview, jež je programu digiKam svou funkčností dost podobná. Třetí místo patří programu MaPiVi, který obsahuje poměrně velké množství funkcí. Předposlední pak programu HViewer, především za zásuvné moduly a poslední aplikaci imgSeek.

Kapitola 8

Možnosti rozšíření

Velký prostor, co se týče zlepšení aplikace, zbývá pro různé volby uživatelských nastavení, kterých v programu není příliš mnoho. Uživatel by mohl mít možnost volby nastavování popisků náhledů fotografií, které by mohly být podmnožinou metadat (tzn. hodnota ISO, vyvážení bílé, použití blesku, GPS souřadnice, ohnisková vzdálenost, apod.). Podobné nastavení by pak bylo užitečné i při promítání fotografií. Další užitečnou volbou by mohla být programová změna ovládacího bloku s tlačítky za adresářový strom (či přidání adresářového stromu), která by, kvůli propojení tlačítek z jednotlivými funkcemi, byla velmi složitá.

Velmi užitečné by také bylo rozhraní pro řetězové zpracování pomocí skriptů, či filtrů. Kdy by bylo možné přidávat vybrané fotografie do tzv. fronty, na kterou by se aplikoval seznam příkazů. Rozhraní by pak bylo realizováno jako samostatné okno programu HViewer. Co se samotných filtrů týče, jsou možnosti poměrně rozmanité. Od vyvážení bílé, úpravy křivek, převod na jiné bitové hloubky, rozmazání fotografie, zaostření fotografie, převody do různých formátů a mnohé další. Tvorba takového rozhraní by se, za určitých podmínek, mohla svou náročností rovnat samotnému programu HViewer.

Dalšími funkcemi, kterými by mohl být HViewer rozšířen jsou další druhy vyhledávání. Například vyhledávání pomocí náčrtku, pomocí podobnosti fotografie, nebo vyhledávání duplicit, což jsou funkce, které některé programy v kapitole 7 obsahují.

Co se týče využití GPS souřadnic u fotografií, byla by vhodná funkce pro tzv. rekonstrukci trasy, která by se aplikovala na sadu fotografií, které obsahují GPS pozice. Trasa by pak byla znázorněna pomocí *google maps API* s využitím funkcí, kterými je možné vkládat body, které by reprezentovaly jednotlivé pozice, či úsečky, které by reprezentovaly jednotlivé přechody mezi pozicemi.

Další užitečné vlastnosti, které v programu prozatím chybí, jsou volby importu a exportu fotografií. Import či export fotografií by mohlo probíhat z internetových galerií, které jsou v dnešní době hojně využívány, či ze sociálních sítí.

Kapitola 9

Závěr

Cílem práce bylo vytvořit editor EXIF a IPTC metadat u fotografií formátu `jpeg` s dalšími funkcemi, které již byly několikrát popsány (například v kapitole 3).

Práce tedy nejprve seznamuje s přehledem používaných typů metadat (u digitálních fotografií), mezi které metadata EXIF a IPTC bezesporu patří, s typy souborů, u kterých jsou jednotlivá metadata používána a s typy atributů, které je možné v rámci určitého typu metadat zapsat.

Poté jsou představeny kompletní požadavky na výsledný program, tj. veškeré cíle, na základě kterých byl program navrhován a některé postupy, či technologie, které jsou v programu využívány. Následuje poměrně detailní popis implementace programu, který, společně se zdrojovými kódy aplikace, která byla vyvíjena jako aplikace typu **Open Source**, může napomoci k rozšíření či úpravě programu (pokud by uživatel nechtěl využít možnosti zásuvných modulů, nebo by nabízené možnosti nebyly dostatečné). Následuje detailní popis výsledné aplikace, tzn. návod na její ovládání a používání, která splňuje všechny požadavky na výsledný program, tedy stanovené cíle. Dokument je zakončen detailnějším porovnáním programu se stávajícími aplikacemi společně s konstatováním rozdílů mezi nimi a s prozkoumáním jejich výhod či nevýhod, které jsou dostupné pro operační systém Linux, a možnostmi rozšíření a vylepšení konečné aplikace.

Vývoj i testování aplikace byl prováděn na operačním systému Linux, konkrétně na distribuci Ubuntu 10.04 LTS (viz [17]), pro kterou byl vzhled aplikace průběžně optimalizován, se zmíněnými nástroji (6.1), které jsou pro plnou funkčnost aplikace nutné. Pro systém Linux je výsledná aplikace také určena a je v něm bez problémů použitelná. Na systému Microsoft Windows je aplikace prozatím nefunkční a to kvůli způsobu, jakým jsou některé nástroje (`exiv2`, `exiftran`, `jpegtran`), na kterých je program funkčně závislý, využívány.

Literatura

- [1] Internetová stránka věnovaná programu QGview [online].
Poslední modifikace: 2006-03-12 [cit. 2011-04-15].
URL <http://gqview.sourceforge.net>
- [2] Využití Google maps API [online].
Poslední modifikace: 2010-01-30 [cit. 2011-04-22].
URL <http://digikam3rdparty.free.fr/gpslocator/getlonlatalt.php>
- [3] Stránka věnovaná nadmořské výšce a nadmořské šířce [online].
Poslední modifikace: 2011-01-26 [cit. 2011-04-01].
URL http://www.nationalatlas.gov/articles/mapping/a_latlong.html
- [4] Vysvětlení pojmu Progress bar [online].
Poslední modifikace: 2011-02-26 [cit. 2011-04-02].
URL http://en.wikipedia.org/wiki/Progress_bar
- [5] Popis formátu XMP [online].
Poslední modifikace: 2011-03-14 [cit. 2011-04-14].
URL http://en.wikipedia.org/wiki/Extensible_Metadata_Platform
- [6] Vysvětlení pojmu Status bar [online].
Poslední modifikace: 2011-03-15 [cit. 2011-04-02].
URL http://en.wikipedia.org/wiki/Status_bar
- [7] Popis formátu IPTC v anglickém jazyce [online].
Poslední modifikace: 2011-04-04 [cit. 2011-04-13].
URL http://en.wikipedia.org/wiki/IPTC_Information_Interchange_Model
- [8] Popis formátu IPTC v českém jazyce [online].
Poslední modifikace: 2011-04-06 [cit. 2011-04-13].
URL <http://cs.wikipedia.org/wiki/IPTC>
- [9] Vysvětlení pojmu Polygon [online].
Poslední modifikace: 2011-04-09 [cit. 2011-04-10].
URL <http://en.wikipedia.org/wiki/Polygon>
- [10] Popis programu ExifTool [online].
Poslední modifikace: 2011-04-12 [cit. 2011-03-22].
URL <http://en.wikipedia.org/wiki/ExifTool>
- [11] Popis formátu EXIF [online].
Poslední modifikace: 2011-05-04 [cit. 2011-04-13].
URL http://en.wikipedia.org/wiki/Exchangeable_image_file_format

- [12] Příklad zásuvného modulu echo [online].
© 2008–2010 [cit. 2011-04-08].
URL <http://doc.qt.nokia.com/latest/tools-echoplugin.html>
- [13] Příklad zásuvného modulu plug and paint [online].
© 2008–2010 [cit. 2011-04-08].
URL <http://doc.qt.nokia.com/latest/tools-plugandpaint.html>
- [14] Tvorba zásuvných modulů ve Frameworku Qt [online].
© 2008–2010 [cit. 2011-04-08].
URL <http://doc.qt.nokia.com/latest/plugins-howto.html>
- [15] Domovská stránka Frameworku Qt [online].
© 2008–2011 [cit. 2011-03-20].
URL <http://qt.nokia.com>
- [16] Popis formátu XML verze 1.0 [online].
© 2008 [cit. 2011-03-30].
URL <http://www.w3.org/TR/xml>
- [17] Internetová stránka operačního systému Ubuntu verze 10.04 LTS [online].
© 2010 [cit. 2011-04-14].
URL <http://www.ubuntu.cz/ziskejte/poznamkykvydani/1004poznanky>
- [18] Internetová stránka Google maps [online].
© 2011 [cit. 2011-04-10].
URL <http://maps.google.com>
- [19] Domovská stránka formátu EXIF [online].
© 2011 [cit. 2011-04-13].
URL <http://www.exif.org>
- [20] Domovská stránka formátu IPTC [online].
© 2011 [cit. 2011-04-13].
URL http://www.iptc.org/site/Photo_Metadata
- [21] Domovská stránka formátu XMP [online].
© 2011 [cit. 2011-04-13].
URL <http://www.adobe.com/devnet/xmp.html>
- [22] Domovská stránka projektu KDE [online].
© 2011 [cit. 2011-04-15].
URL <http://www.kde.org>
- [23] Internetová stránka věnovaná pluginu KIPi [online].
© 2011 [cit. 2011-04-15].
URL <http://www.kipi-plugins.org/drupal>
- [24] GOOGLE: Google Maps JavaScript API V2 [online].
© 2011 [cit. 2011-04-01].
URL <http://code.google.com/apis/maps/documentation/javascript/v2>

- [25] HUGGEL, A.: Exiv2 – Image metadata library and tools [online].
Poslední modifikace: 2011-02-14 [cit. 2011-03-25].
URL <http://www.exiv2.org>
- [26] MARTINEK, D.: Zásuvné moduly [online].
Poslední modifikace: 2010-10-24 [cit. 2011-04-08].
URL <http://www.fit.vutbr.cz/~martinek/clang/plugins.html>
- [27] MURRAY, J. D.; VANRYPER, W.: *Encyklopedie grafických formátů*. Praha: Computer Press, první vydání, 1995, ISBN 80-85896-18-4, 736 s.
- [28] RAO, K. R.; HWANG, J. J.: *Techniques standards for image video, audio coding*. New Jersey: Prentice Hall, 1996, ISBN 0-13-309907-5, 564 s.

Příloha A

Obsah CD

Příložené CD obsahuje jednoduchou adresářovou strukturu, která musí být pro bezproblémový chod aplikace zachována, která je následující.

- app** – následující soubory a adresáře týkající se aplikace HViewer.
 - **img** – grafické soubory používané v programu HViewer.
 - **plg** – zkompileované zásuvné moduly a následující adresáře.
 - **Grayscale** – zdrojové kódy zásuvného modulu Grayscale.
 - **LossLessRotate** – zdrojové kódy zásuvného modulu LossLessRotate.
 - **pos** – plakát k aplikaci HViewer.
 - **src** – zdrojové kódy aplikace HViewer.
 - **txt** – pomocné soubory aplikace.
 - **Makefile** – překlad všech zdrojových kódů.
 - **README** – základní informace o programu HViewer.
- doc** – text práce.
- pic** – sada obrázků pro testování aplikace, se kterými byly vytvářeny náhledy.