VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ DUPLICIT A VYHODNOCENÍ KVALITY FOTOGRAFIÍ

SEARCH FOR DUPLICITIES AND QUALITY EVALUVATION OF PHOTOS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ZDENĚK SKLENÁŘ**
AUTHOR

VEDOUCÍ PRÁCE **Prof. Dr. Ing. PAVEL ZEMČÍK**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Sklenář Zdeněk**
Obor: Informační technologie
Téma: **Vyhledávání duplicit a vyhodnocení kvality fotografií**
Search for Duplicities and Quality Evaluvation of Photos
Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte literaturu na téma informací obsažených v digitálních fotografiích a existující software pro zpracování takových informací.
2. Navrhněte postup vyhledávání duplicit ve fotografiích a skupinách fotografií podle informací ve fotografiích a k postupu navrhněte i uživatelské rozhraní.
3. Navrhněte prostředí a způsob zpracování dat pro efektivní implementaci výběru vhodného snímku z duplicitních.
4. Implementujte zobrazování duplicit fotografií i uživatelské rozhraní s demonstrujte funkčnost na vhodném příkladu.
5. Zhodnoťte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, prof. Dr. Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem, implementací a otestováním aplikace, která slouží k vyhledání duplikací ve fotografiích podle Exif metadat. Aplikace dále umožňuje zobrazení náhledu fotografie, včetně Exif metadat fotografie. Filtrování fotografií. Seskupení duplikací, následný výběr nejlepší fotografie pro ponechání podle parametru nastaveného uživatelem, manuální úpravu této volby a vymazání ostatních. Export vybraných fotografií do ZIP archivu.

Abstract

This bachelor thesis is about the analysis, design, and implementation and testing of an application, which is used to find duplicates in photographs according to it's Exif metadata. The app can also let you preview the photo, including Exif metadatas. It is possible to filter your photos. You can group duplicates with the original photo, and select the best photos to keep it in accordance with a user-defined parameter, then manually adjust this option, and deleting others. There is also a possibility to export selected photos to a ZIP archive.

Klíčová slova

Fotografie, Metadata, Exif, IPTC, Duplicita, Python3.x, pyQT5, cx_freeze, filtrování fotografií, export fotografií, ZIP, Git

Keywords

Photo, Metadata, Exif, IPTC, Duplicate, Python3.x, pyQT5, cx_freeze, Photos filtering, Photos export, ZIP, Git

Citace

SKLENÁŘ, Zdeněk. *Vyhledávání duplicit a vyhodnocení kvality fotografií.* Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

Vyhledávání duplicit a vyhodnocení kvality fotografií

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Zdeněk Sklenář
14.05.2018

Poděkování

Rád bych poděkoval hlavně svému vedoucímu práce panu Prof. Dr. Ing. Pavlovi Zemčíkovi za pevné nervy a jeho věcné připomínky k práci. Také své rodině a přátelům. Kteří dokázali vidět naději v momentech kdy jsem to já nedokázal, bez jejich podpory by tato práce nikdy nevznikla. V neposlední řadě také všem, kteří mě naučili to, co znám a nezlomili nade mnou hůl, popřípadě ochotně pořídili novou.

Zvláštní poděkování patří odbornému konzultantovi panu Ing. Martin Čacho. Děkuji za jeho úhel pohledu na tuto práci a jeho čas, který na konzultace vymezil.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V dnešní digitální době je snadné během krátkého času vytvořit velké množství fotografií. Všechny tyto fotografie se ale musí někam ukládat a často tak ve velkém počtu zaplňují naše úložiště. Proto vznikla snaha o jejich třídění a určení duplicitních fotografií, které je zbytečné uchovávat. Jejich vymazáním tak vznikne místo pro další, důležitější fotografie. Až tedy budete příště kupovat větší úložné médium pro svoji dlouholetou galerii, zamyslete se nejdříve nad tím, zda opravdu potřebujete všechny její položky.

Mým úkolem bylo nastudovat teorii k problematice duplikace ve fotografiích podle Exif metadat, a poté vytvořit aplikaci pro jejich detekci a vymazání. Jako volitelné cíle byly možnosti zobrazení metadat a jejich úprava, jednotlivá i hromadná. Také filtrování a export fotografií.

Aplikace by měla pomoci běžnému uživateli pročistit jeho dlouhodobé uložiště fotografií ve kterém se pravděpodobně mohou nacházet duplikace. Je zde například více verzí fotografie, protože byla postupem času několikrát upravována. Popřípadě můžou být tyto verze i naprosto totožné, ať už z důvodů zálohy nebo výběru do jiné složky. Aplikace tyto stejné a upravené verze fotografie seskupí a nabídne uživateli náhled na jednotlivé skupiny duplikací. Určení duplikace tedy není nijak závislé na obrazových datech fotografie.

Tato bakalářská práce se skládá ze shrnutí současného stavu popsaného v kapitolách 2, 3, 4. Specifikování požadavků, popsání návrhu, testování a vývoje aplikace v kapitole 5. Popis implementace aplikace v kapitole 6. Následuje otestování vlastností této implementace na reálných datech a jejich vyhodnocení v kapitole 7.

Kapitola 2

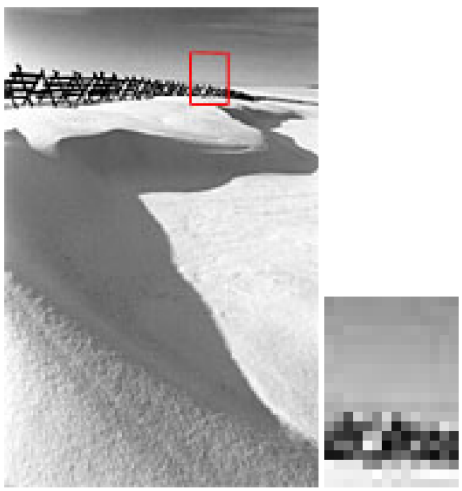
Fotografie a datové struktury

Klasická fotografie provází lidstvo již od předminulého století, prožila velký vývoj a stále se ještě vyvíjí. Přestože nehrozí, že by digitální fotografie zcela nahradila klasický film, trend posledních let ukazuje, že si stále více lidí pořizuje digitální fotoaparáty. Vzhledem k povaze práce zde zmíním jen některé nejdůležitější pojmy, potřebné pro její správné pochopení.

Digitální fotografie

Vzniká okamžitě po jejím pořízení a je tak možné jí ihned zkontrolovat. Což je jeden z hlavních důvodů velkého rozšíření digitálních fotoaparátů. [28] Vlastnosti digitálních fotoaparátů jsou neustále vylepšovány hlavně tedy jejich rozlišení, pomalý zápis snímků nebo jejich vysoká spotřeba.

Každou digitální fotografii si lze představit jako mozaiku, skládající se z jednotlivých čtverečků tedy pixelů. Síť vytvářející pixely se nazývá rastr. Pixely v rastru mohou mít menší nebo větší plochu. Záleží na tom, jak hustou síť máme k dispozici. Čím je síť hustší, tím ostřejší obraz získáme. [22] Rastr digitálního obrazu se objeví na každém monitoru v okamžiku, kdy obraz dostatečně zvětšíte.



Obrázek 2.1: Detail fotografie

Jedním z nejdůležitějších parametrů pro uchování digitální fotografie je formát souboru, do kterého je uložena. Formát může výrazně ovlivnit například kvalitu a velikost fotografie. [19] Více o procesu digitalizaci fotografie zde [28].

- **JPEG** (JPG)
  Tento formát se hodí na fotografie, u kterých se díky ztrátové kompresi dá ušetřit hodně datového objemu a lehká ztráta kvality při běžném pozorování příliš nevadí.

  Formát byl navržen speciálně pro fotografie, takže většinou nenabízí uspokojivou kvalitu, pokud je obsahem něco jiného než obraz. Pokud je ale potřeba kvalitní fotografie, je tento formát vhodný spíše pro konečný export upravené fotografie. Díky své malé velikosti je často používán na webu.

  Je vhodný pro uchování rastrové grafiky.

- **PNG**
  Je vhodný typicky pro webovou/počítačovou grafiku. Jeho komprese je bezeztrátová, čili výsledek 100% odpovídá originálu. Nevýhoda je vysoká datová velikost oproti JPG při ukládání fotografií. Lze vytvářet i průhledné obrázky. Průhlednost může být úplná nebo částečná - někdy označováno jako průhlednost a průsvitnost.

  Je vhodný pro uchování vektorové grafiky.

- **GIF**
  Jde často používat jako alternativu k PNG - je rovněž bezeztrátový. Nevýhoda oproti PNG je ale jeho omezený maximální počet barev. Kromě toho má GIF horší kompresi a není možné ho udělat průsvitný (chybí podpora alfa kanálu), podporována je jen obyčejná průhlednost.

  Nedává tedy příliš smysl dát přednostu GIFu před PNG. S jedinou výjimkou, což je animace, kde není na výběr.

2.1 Archivace

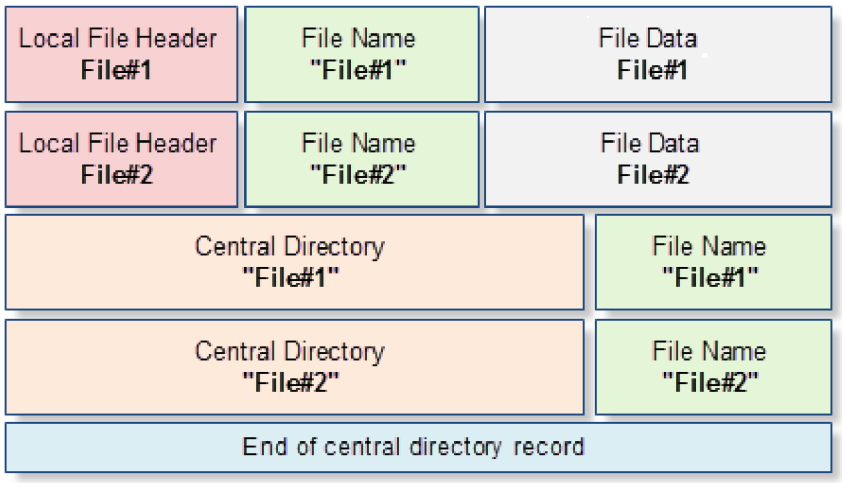
Je způsob pro dlouhodobé a trvalé uchování nepoškozených dat. S možností bezeztrátové komprese.
Je možné použít pro export fotografií se zachováním metadat. Export slouží hlavně pro výběr a přenos fotografií na jiné úložiště. Lze tedy například udělat filtrovaný výběr z většího množství a poté výběr vložit do jednoho strukturovaného archivu.
Existuje spoustu formátu a možností archivace, také kompresních algoritmů s různou mírou komprese.

ZIP

Je jeden z typů pro archivaci a kompresi dat.

Použití datové komprese není podmínkou. Je zde několik kompresních algoritmů, ze kterých si může uživatel vybrat. Archiv by měl obsahovat alespoň jeden nebo více souborů. Také může obsahovat složky. [11]

Tento formát byl zvolen hlavně pro svoji jednoduchost a volitelnost komprese. Také má vestavěnou podporu ve většině operačních systémů.



Obrázek 2.2: Schéma ZIP archivu, který obsahuje dva soubory: File#1 a File#2

ZIP archiv se tedy skládá z: [25]

- **Local File Header** (Hlavička komprimovaného souboru)
  Tento datový blok obsahuje základní informace o komprimovaném souboru. Například velikost souboru před a po provedení komprese, čas poslední modifikace souboru, kontrolní součet CRC-32 a místní ukazatel na název souboru. Kromě toho obsahuje tento blok také údaj o archivační verzi nezbytnou pro dekomprimaci souboru.

- **File Name** (Název komprimovaného souboru)
  Je posloupnost bajtů s libovolnou délkou, která tvoří název komprimovaného souboru. Délka názvu souboru by neměla překročit 65 536 znaků.

- **File Data** (Obsah komprimovaného souboru)
  Obsahuje komprimovaný soubor ve formě libovolného délkového byte pole. Je-li soubor prázdný nebo obsahuje adresář, pak se toto pole nepoužívá a název záhlaví lokálního souboru, který popisuje další soubor, následuje název souboru nebo adresáře.

- **Central Directory** (Centrální složka)
  Poskytuje rozšířené zobrazení dat v Local File Header. Vedle dat obsažených v hlavičce obsahuje také atributy souborů, místní odkaz na strukturu hlavičky místního souboru a jiné většinou nepoužité informace.

- **End of central directory record** (Ukončení záznamu)
  Tato struktura je prezentována jako jednotná šablona v každém archivu a je uložena na konci archivu. Nejzajímavější data, která obsahuje, jsou počet záznamů archivu (nebo řada souborů a adresářů) a místní odkazy na začátek bloku Central Directory.

2.2 Metadata

Metadata jsou skupina informací o datech. Metadata ve fotografiích umožňují zachovat informace, které jsou přenášeny s obrazovými daty.

Způsob uložení metadat si definuje každý formát zvlášť. Je důležité, aby grafické editory s metadaty u fotografie počítaly a při úpravě tak nedošlo k jejich smazání. Tato nepříjemnost se projevovala hlavně v minulosti, dnes už většina grafických editorů s metadaty počítá. V některých případech je dokonce i automaticky mění v závislosti na úpravě. [21]

Metadata zapisuje fotoaparát při pořízení snímku. Zapíše například datum, GPS polohu, svůj typ a značku výrobce. V dnešní době sociálních sítí tak vzrůstala obava z informací, které by se daly z fotografií vyčíst. Proto sociální sítě promazávají například GPS údaje z metadat fotografie.

Metadata ve fotografiích se používají zejména pro třídění a hledání, v případě práce s velkou galerií. Pro vyhledání upravené fotografie a jejího originálu, popřípadě kopie. [23]

Metadata mohou být uchována dvěma způsoby interně nebo externě. Interní metadata jsou uložena vedle obrazových dat v jednom souboru. Externí metadata jsou uložena do samostatného XML souboru. Postupem času vznikly následující formáty.

Exif

Z grafických formátů nepodporuje GIF. Také nepodporuje video formáty. Informace jsou ukládány interně v určené části souboru. Tento formát není od verze 2.2, která vyšla v roce 2002 dále vyvíjen. [10]

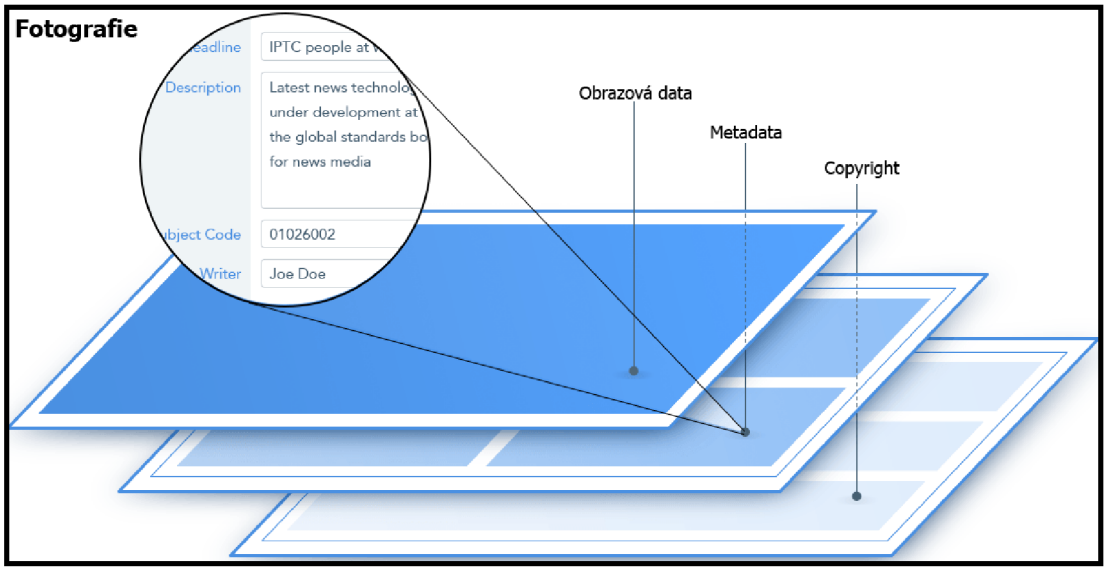
Kromě ukončeného vývoje a toho, že podporuje pouze fotografie, má tento formát i několik dalších nevýhod. Například propojení jednotlivých atributů, se kterým se velké množství editorů nedokáže vypořádat, čímž dochází k poškození, nebo odstranění již zapsaných metadat. Podpora barevné hloubky pouze 24 bitů.


```
{
 'Make': 'LENOVO',
 'Model': 'Lenovo S1',
 'Orientation': '1',
 'XResolution': '(72, 1)',
 'YResolution': '(72, 1)',
 'GPSInfo': "16: 'M', 17: (0, 1)",
 'ResolutionUnit': '2',
 'Software': 'Lenovo Camera',
 'DateTime': '2016:06:24 17:06:24',
 'ColorSpace': '1',
 'ExifImageWidth': '2560',
 'LightSource': '255',
 'FocalLength': '(350, 100)',
 'ExifImageHeight': '1920',
 'Flash': '0',
 'ExifInteroperabilityOffset': '812',
 'ISOSpeedRatings': '91',
 'DigitalZoomRatio': '(100, 100)'
}
```


Obrázek 2.3: Ukázka Exif matadat

IPTC

Podporuje všechny druhy médií například text, fotografie, video i zvuk. Informace jsou uloženy interně v určené části souboru. Význam tohoto formátu je v současnosti na vzestupu. [4]



Obrázek 2.4: Detail fotografie s IPTC metadaty

XMP

Vychází ze značkovacího jazyka XML. Metadata jsou ukládána do samostatného souboru, který je přidružen k multimediálnímu souboru. Open-source standard byl vytvořen v roce 2001 společností Adobe.

Při editaci RAW souborů ukládá poslední nastavení editoru při editaci. [15] V případě ztráty upravených fotografií tedy stačí pouze načíst originál, k němu příslušný XMP soubor a vytvořit nové.

Kapitola 3

Nástroje pro vývoj aplikace

Při vývoji aplikací je dobré zachovat jeho historii, aby bylo možné jakékoliv rozhodnutí v průběhu vývoje změnit. Také použít ověřenou a jednoduchou šablonu pro jeho správný průběh. Dále je nutné zvolit jazyk ve kterém bude aplikace a případné ostatní věci kolem vývoje napsány. Vzhledem k omezenému rozsahu práce jsou zde zmíněné pouze nejvíce relevantní nástroje a metodiky pro vývoj software.

3.1 Verzovací systémy

Slouží nejen pro uchování historie vývoje projektu, kde se lze vrátit k jakémukoliv zaznamenanému kroku. Ale v případě týmových projektů také pro efektivní spolupráci. Také slouží pro snadné sdílení celého projektu.

SVN

Apache Subversion je multiplatformní systém pro správu a verzování zdrojových kódů, slouží jako náhrada za starší systém CVS. Snaží se zachovat podobný způsob a styl práce, ale odstranit nedostatky CVS jako například nemožnost kopírování adresářů, časová a prostorová náročnost větvení a podobně.

Jednou z výhod systému SVN je existence velmi dobré dokumentace, která je volně dostupná v anglickém jazyce. [13] Další je možnost více přístupových metod k repozitáři. SVN je, tak jako CVS, založeno na principu centrálního repozitáře.

SVN uspokojuje základní potřeby při správě verzí. Je šířen pod licencí, která umožňuje jeho bezplatné komerční použití, k dispozici jsou zdrojové kódy.

Jak již bylo řečeno SVN je inspirován starším systémem CVS, bere si z něj některé jeho vlastnosti, je však mnohem flexibilnější a jeho používání je snazší.

Skládá se ze dvou hlavních částí – klientská část a serverová. Klientská část poskytuje nástroje pro práci s verzemi přímo v pracovním adresáři a komunikaci se serverovou částí, která se stará o repozitář (centrální, sdílené úložiště). [3]

Existuje několik klientských nástrojů pro práci s SVN systémem, od příkazového řádku, přes webové rozhraní až po nástroje integrované do GUI operačního systému.

Git

Je systém správy verzí vytvořený Linusem Torvaldsem pro vývoj jádra Linuxu. Návrh Gitu byl ovlivněn projekty BitKeeper (dříve používán pro vývoj jádra Linuxu) a Monotone. CVS zde neslouží jako inspirace, ale naopak pouze jako odstrašující případ. Jedná se o svobodný software.

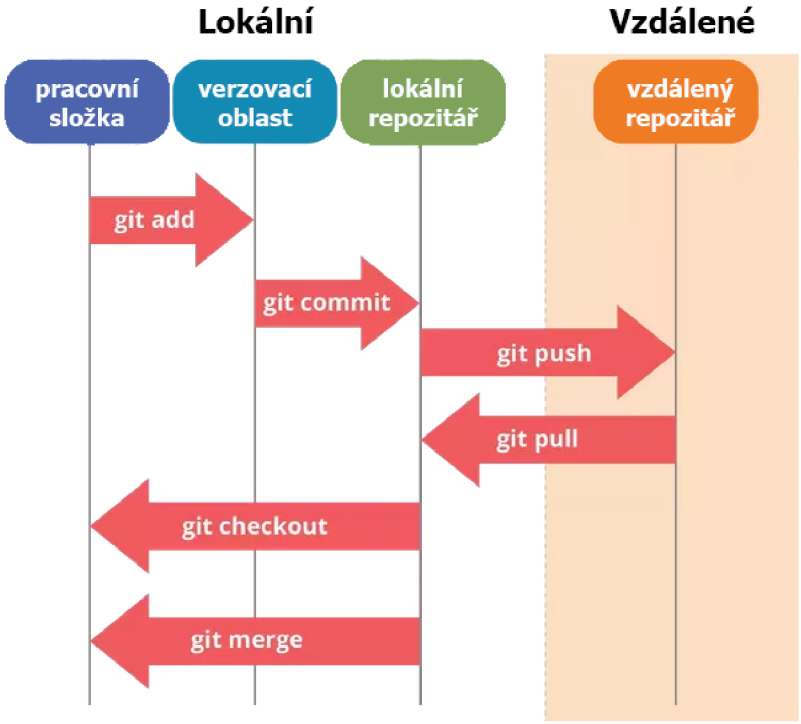
V současnosti je Git používán v mnoha známých firmách, například: Google, Microsoft. Pro práci na jejich projektech

Velká podpora pro nelineární vývoj. Git podporuje rychlé vytváření větví a jejich rychlé slučování. Obsahuje specifické nástroje pro vizualizaci a navigaci v nelineární historii vývoje projektu. Zajišťuje integritu dat.

Podobně jako je tomu u systémů Darcs, BitKeeper nebo třeba Monotone poskytuje Git vývojáři lokální kopii celé historie vývoje. Změny se kopírují z jednoho úložiště do jiného. Tyto změny se importují v podobě dalších vývojových větví, které mohou být začleněny do jiné větve stejným způsobem, jako lokálně vyvíjené větve.

Repozitář může být zveřejněn prostřednictvím protokolů HTTP, FTP, rsync nebo protokolu Git realizovaného buď přes obyčejné sockety nebo přes SSH. [1]

Existuje také spousta nástrojů pro efektivní práci s repozitářem. Například nástroj pro integraci systému git přímo do operačního systému TortoiseGit. [2]



Obrázek 3.1: Význam základních příkazů verzovacího systému Git při synchronizaci

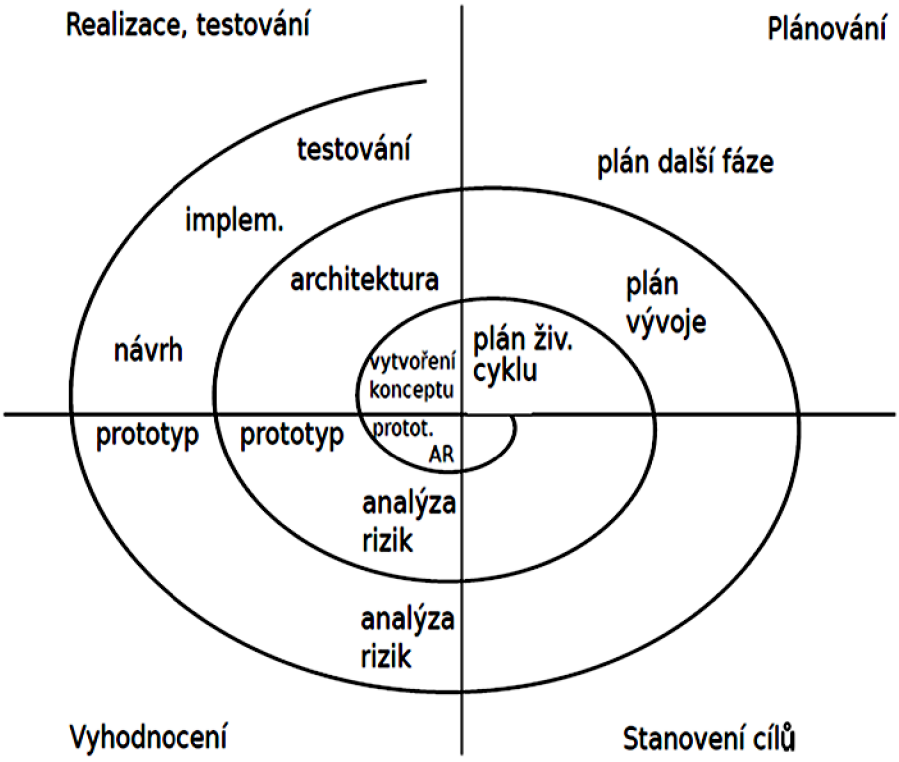
3.2 Klasické metodiky vývoje softwaru

Metodika je souhrn doporučených praktik a postupů pro vývoj software. Popisuje řízení vývoje, ale i vývoj samotný. Pomocí modelů. Modely popisují jednotlivé fáze, jejich vstupy, výstupy a nutné činnosti. Které je nutné udělat v dané fázi. Vzhledem k povaze práce přicházeli v úvahu dva modely, také není nutné zmiňovat agilní metodiky.

Spirálový model

Založen na kombinaci prototypování a analýzy rizik. Jednotlivé kroky vývoje se vždy opakují na vyšším stupni. Jeho autorem je Barry Boehm, model se začal používat v 80. letech. [18]

Jedná se o komplexní model vhodný hlavně pro složitější projekty, kde se požadavky mohou měnit v průběhu vývoje. Model je závislý na analýze rizik a chyby jsou tak odhaleny dříve. Tato analýza ale musí být pravidelně prováděna odborníky. Což zvyšuje cenu a složitost projektu. Vyžaduje stálou spolupráci se zákazníkem. Čímž se vývoj značně zpomaluje, ale výrazně se tak snižuje riziko nespokojenosti uživatele při finálním odevzdání projektu. Problematické přesné plánování termínu a cen.



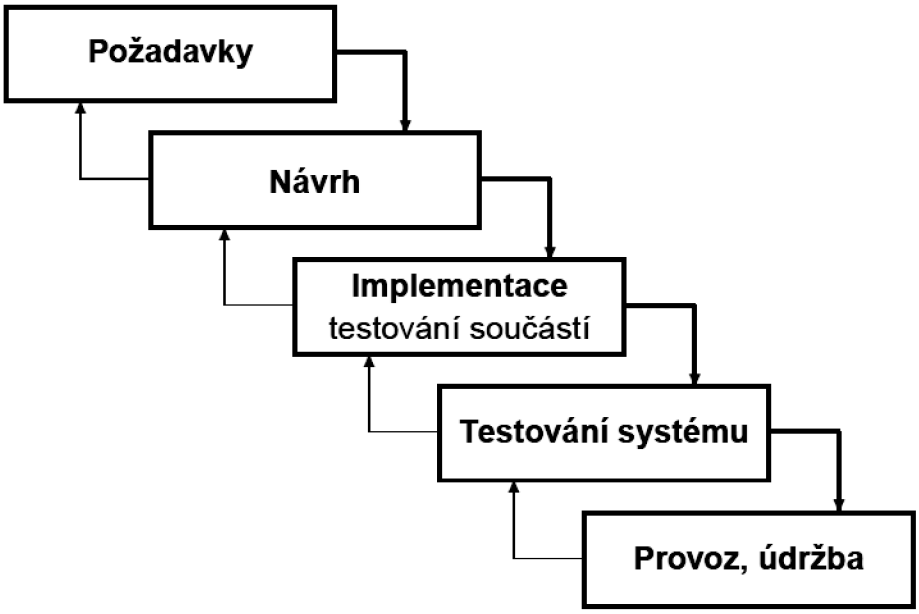
Obrázek 3.2: Spirálový model metodiky vývoje

Vodopádový model

Zástupce lineárních modelů. Každá fáze tedy začne až po ukončení předchozí, je možný pouze návrat k přímo předcházející etapě.

Jedním z prvních modelů, intenzivně používán v 70. letech. Cílem bylo zavést do vývoje jednoduchý řád, který umožní řešit náročnější problémy. Je zde typická postupná dekompozice problému. [18]

Model celkově má minimální interakci se zákazníkem, pouze při zadáni a předání projektu. Z toho plyne několik nevýhod tohoto modelu. Požadavky se jen těžko zpětně mění poté co vývoj postoupí do další fáze. Také zákazník vidí až finální verzi projektu. Nemusí tedy být vždy spokojen, je tedy důležité nepodcenit definici požadavků na software v první fázi vývoje. Navzdory tomu má model řadu výhod, které plynou právě z jeho jednoduchosti. Pokud jsou požadavky dobře definovány v úvodu vývoje, tak je velmi jednoduchý a účinný.



Obrázek 3.3: Vodopádový model metodiky vývoje

3.3 Skriptovací jazyky

Primárním důrazem pro směr těchto jazyků není budování vlastních aplikací. Zaměřují se spíše na řízení již existujících. Jsou navrženy s ohledem na rychlé pochopení a efektivní práci. Jedná se o interpretované jazyky.

PHP

Pracuje na straně serveru. S PHP můžete ukládat a měnit data webových stránek. Původní význam zkratky PHP byl Personal Home Page. Vzniklo v roce 1996, od té doby prošlo velkými změnami a nyní tato zkratka znamená spíše PHP: Hypertext Preprocessor.

PHP není nijak těžké pochopit a už se základy si lze vystačit. Umí ukládat, měnit a mazat data. Vše se odehrává na webovém serveru. Skript se nejprve provede na serveru a potom odešle prohlížeči pouze výsledek. Proto ve zdrojovém kódu najdete jen výsledek (to je rozdíl oproti JavaScriptu, který počítá přímo v prohlížeči). [6]

Pomocí PHP je možné vytvořit diskuzní fórum, knihu návštěv, počítadlo, anketu, graf, a dokonce si pomocí jednoduchého kódu můžete zlikvidovat celý obsah webu. Navíc máte možnost propojit vaše stránky s databázemi, např. MySQL.

Pro porovnání jazyků se nejčastěji uvádí příklad kódu HelloWorld. [17]

```
<?php
// Hello world in PHP
echo 'Hello World!';
?>
```

JavaScript

Je skriptovací jazyk, který umožňuje vytvořit hodiny, hodnotit data ve formuláři, počítat, dynamizovat data, umožňuje tvorbu všemožných prvků k oživení webu, přes blikající texty po jednoduché hry.
Je základem dynamického webu. Abyste mohli pracovat s JavaScriptem měli byste znát základy HTML. JavaScript je závislý na prohlížeči (uživatel jej může vypnout). V různých verzích prohlížečů nemusí skript vždy korektně fungovat. [5]

Pro porovnání jazyků se nejčastěji uvádí příklad kódu HelloWorld. [17]

```
// Hello world in JavaScript
console.log("Hello World");
```

Python

Existuje spousta článků zabývajících se tím, zda Python patří mezi skriptovací jazyky. Určitě má z této oblasti mnoho vlastností, ale dnes je samotnými tvůrci označován jako interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou. [7] Je tedy vhodný pro skriptování, vývoj webových aplikací, vývoj GUI aplikací, vědecké a numerické výpočty, softwarový vývoj, systémová administrace.

Nepatří mezi silně typované programovací jazyky. Vyvíjen jako open source. Používá systém výjimek. Které lze za použití bloku try-except zachytit a ošetřit. Umožňuje vícenásobnou dědičnost.

Klade důraz na produktivitu programátora. Má jednoduché konstrukce a dobrou čitelnost programu. Pro oddělení bloků programu jsou zde použity hlavně tzv. bílé znaky (mezera, tabulátor, konec řádku), hloubka zanoření odpovídá počtu odsazení bloku.

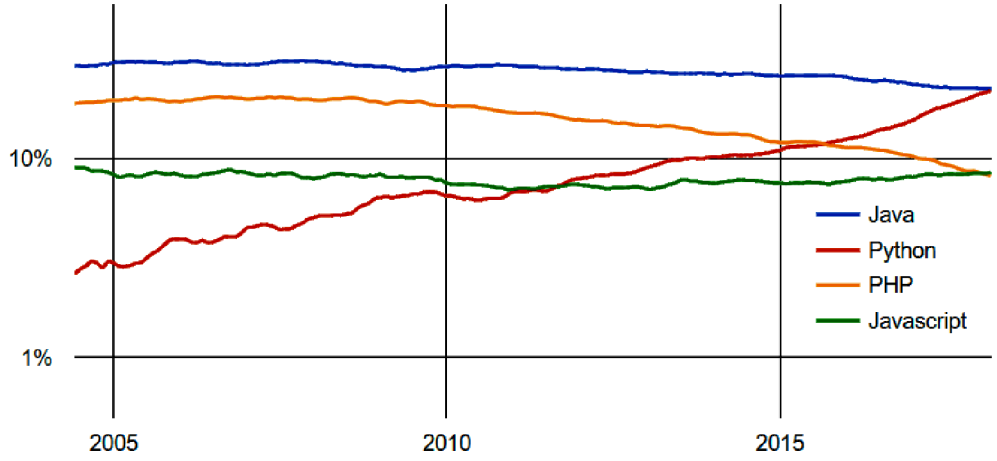
Má velkou zásobu snadno použitelných knihoven včetně jejich dokumentace. Je třeba zkompilovat program pro každý typ operačního systému zvlášť. Verze 2.x není kompatibilní s verzí 3.x.

Pro správné pochopení jazyka je také dobré znát několik zásad od jeho tvůrců. [24]

Pro porovnání jazyků se nejčastěji uvádí příklad kódu HelloWorld. [17]

```
# Hello world in Python 3 (aka Python 3000)
print("Hello World")
```

Popularita jazyka je dlouhodobě na vzestupu. Podle internetových žebříčků, například poněkud nevyrovnaný TIOBE [8] nebo níže ukázaný PYPL [14]. Na zmíněném odkazu se i ve FAQ lze dozvědět porovnání technik obou žebříčků.



Obrázek 3.4: První čtyři nejvíce oblíbené jazyky, podle PYPL žebříčku

Můžeme tedy vidět, že popularita skriptovacích a jim podobných jazyků je na vzestupu. V budoucnu se tak určitě neztratí.

Kapitola 4

Existující řešení

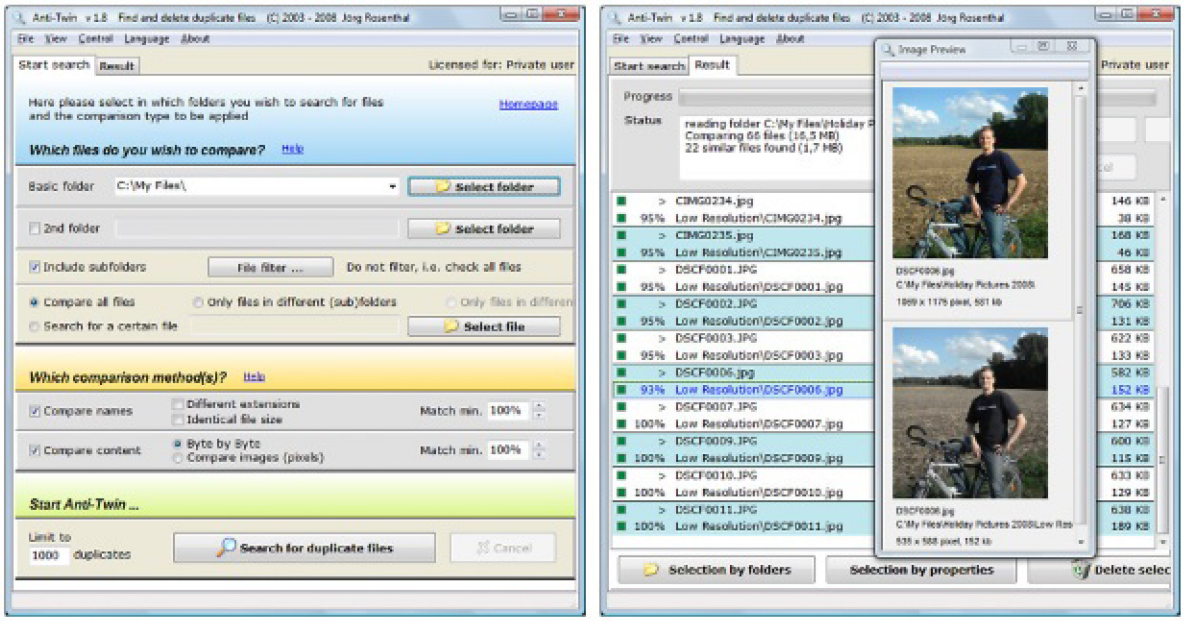
V rámci shrnutí aktuálního stavu, byly vyhledány aplikace. Které se zbývají podobnou problematikou. Pro výběr aplikací byly využity zdroje [9] a [20].

4.1 Anti-Twin

Přímo podporuje pouze platformu Windows. Nepřímo přes nástroj Wine také Linux a Mac.

Pro určení duplikace používá porovnání typu byte-to-byte nebo pixel-to-pixel. Také lze porovnávat podle obsahu a názvu souboru. Při porovnání lze také vzít v úvahu název a velikost souborů. Nalezené duplikace lze vymazat do koše nebo úplně.

Lze použít nejen na vyhledání duplikací ve fotografiích. Ale třeba i na hudební soubory MP3. Všechny údaje uvedené v této práci platí pro verzi 1.8d.



Obrázek 4.1: Ukázka vyhledání duplikací v aplikaci Anti-Twin

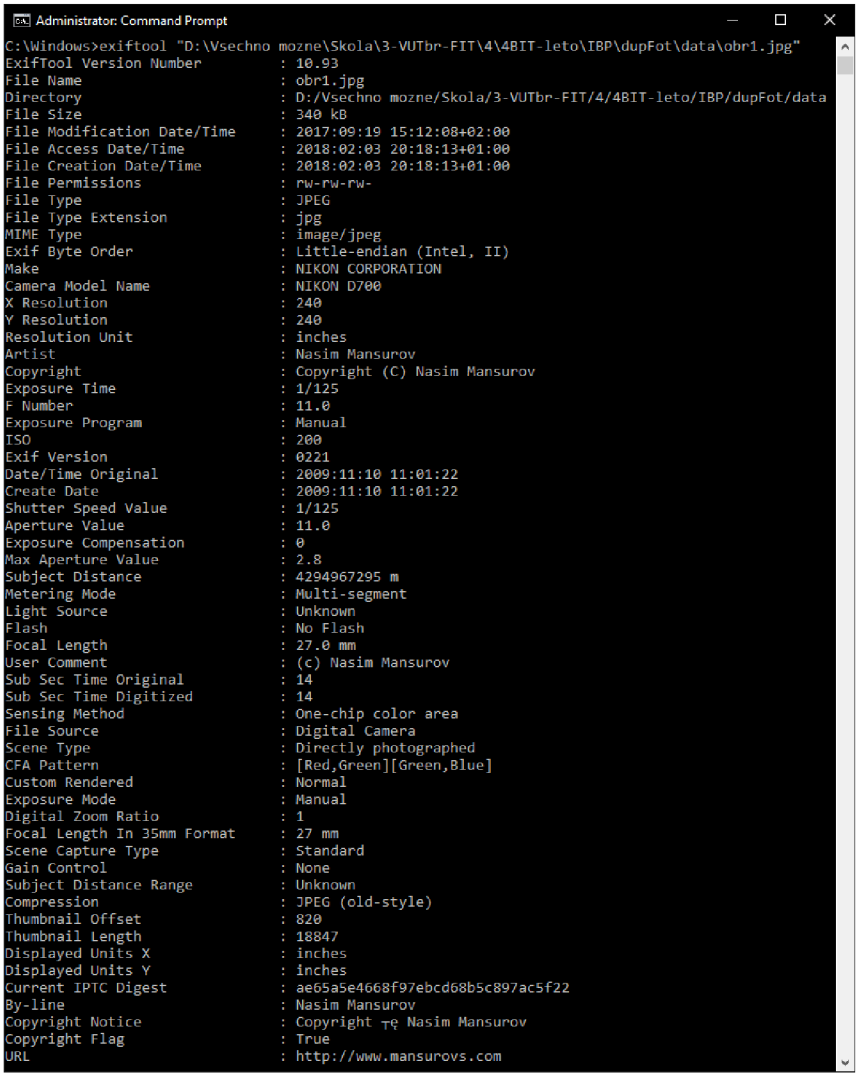
4.2 ExifTool

Podporuje Windows, Linux a Mac platformy.

Pouze konzolová aplikace bez GUI. Lze také použit jako knihovnu pro další projekty. Rozsáhlá internetová dokumentace [16]. Možnost číst a upravovat Exif, IPTC, XMP metadata. Hromadně nebo pro skupinu fotografií. Má více jazykových verzí. Podporuje téměř všechny multimediální formáty. Dlouhodobě testováno na fotografiích z více jak tisíc typů zařízení. Nelze bez dalších rozšíření použít pro vyhledání duplicitních Exif metadat fotografií.

Na vytvoření GUI už vznikl samostatný projekt pyExifToolGui [27], který ale podle slov autora není ani zdaleka dokončen.

Aplikace i projekt na grafickou nástavbu je poskytován zdarma. Nástavba má dostupné zdrojové kódy. Zdrojové kódy pro aplikaci nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 10.94.



Obrázek 4.2: Ukázka výpisu metadat fotografie pomocí ExifTool

4.3 Duplicate Photos Fixer Pro

Podporuje Windows, iOS, Mac a Android platformy. Aplikace má moderní, tmavé grafické rozhraní. Obsahuje překlad do 14 jazyků, čeština bohužel chybí.

Pro určení duplikace používá volitelný interval shody GPS a datumu pořízení fotografie. Také volitelnou míru shody obrazových dat.

Seskupí fotografie do skupin duplikací a nabídne jejich souhrnný náhled. S automatickým výběrem nejvhodnější pro ponechání. Je možná manuální úprava volby. Smaže označené fotografie.

Plná verze není poskytována zdarma. Lze aplikaci vyzkoušet v Trial verzi, která ale nedovolí vymazaní vybraných fotografií. Zdrojové kódy nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 1.1.1086.5815.

Obrázek 4.3: Ukázka zobrazení duplikaci v aplikaci Duplicate Photos Fixer Pro

4.4 Visual Similarity Duplicate Images Finder

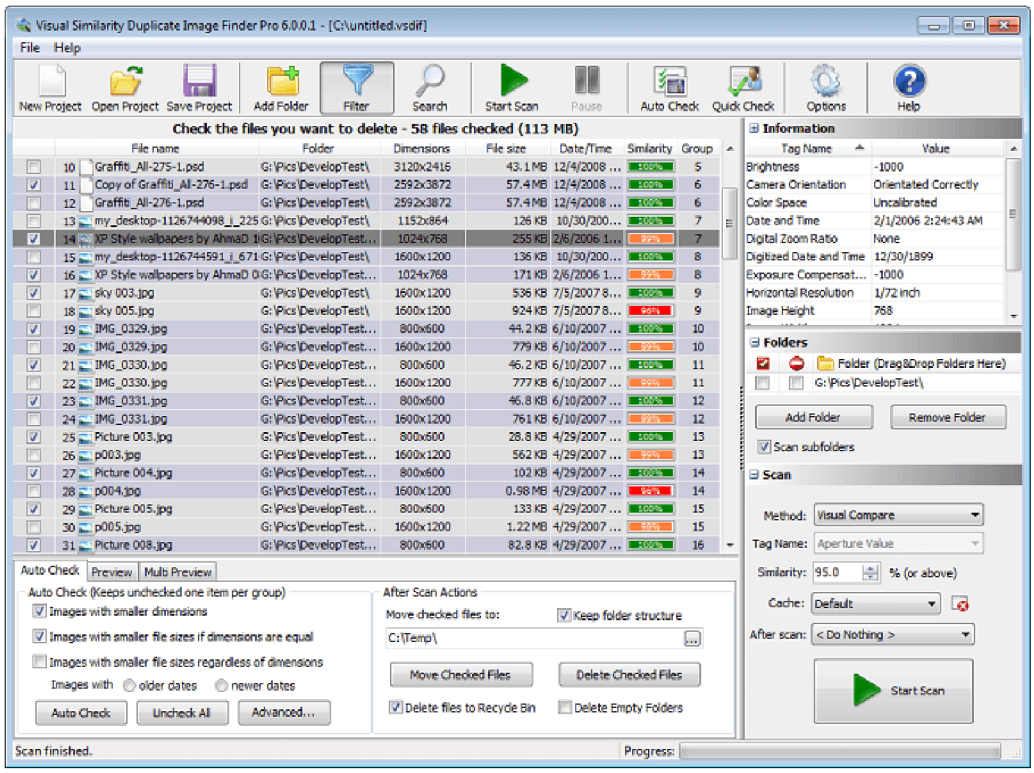
Podporuje pouze platformu Windows.

Umožňuje nastavit procentuální shodu duplikací. Jejich seskupení a automatický výběr nejvhodnější fotografie pro ponechání.

Mezi metodami pro určení duplikace si lze vybrat mezi: Obraz, Exif datum (lze nastavit interval), volitelný Exif záznam, velikost souboru, hash. Lze nastavit procentuální shodu. Umí zobrazit Exif metadata u fotografie.

Označené fotografie lze smazat do koše, smazat úplně, kopírovat, přesunout. Projekt lze uložit a načíst později.

Plná verze aplikace není poskytována zdarma. Lze stáhnout pouze velice omezenou Demo verzi. Všechny údaje uvedené v této práci platí pro verzi Demo 6.7.0.1.



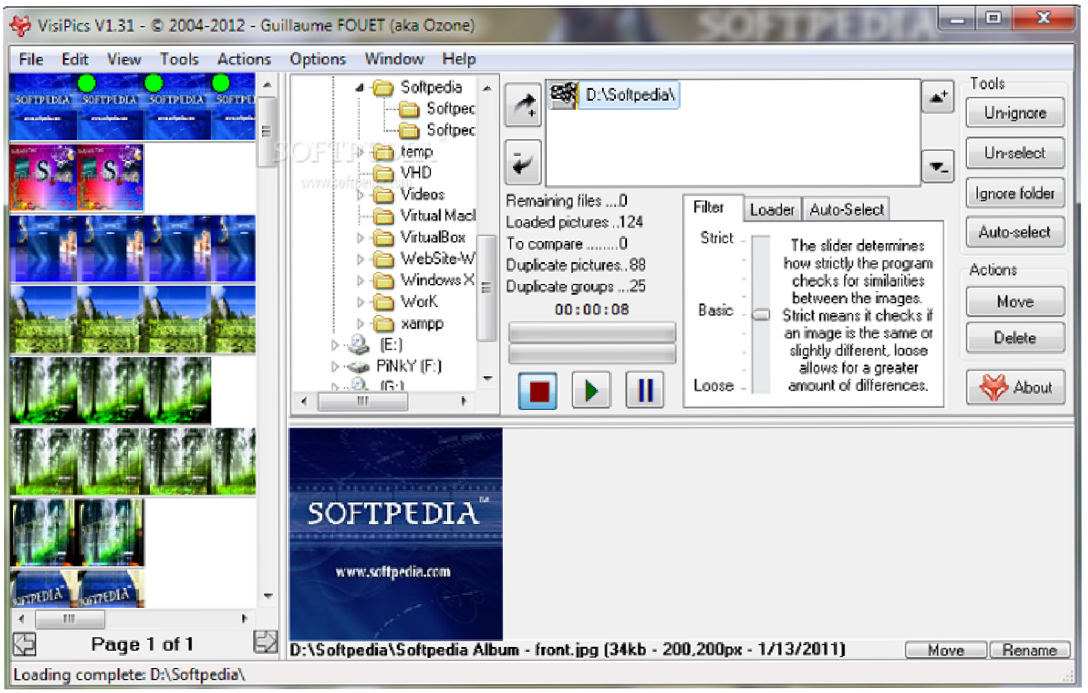
Obrázek 4.4: Ukázka zobrazení duplikaci v aplikaci Duplicate Photos Finder

4.5 VisiPics

Podporuje pouze Windows platformu.

Funkce automatického výběru fotografie pro ponechání ze skupiny duplikací. Hromadné zobrazení skupin duplikací. Duplikaci určuje podle obrazových dat. Duplikace mohou mít rozdílné rozlišení, jiný formát, minimální barevné efekty. Aplikace má tři módy pro vyhledání duplikace. Pro nastavení tolerance velikosti odlišností fotografií.

Aplikace je poskytována v plné verzi zdarma. Zdrojové kódy nejsou dostupné. Všechny údaje uvedené v této práci platí pro verzi 1.31.



Obrázek 4.5: Ukázka aplikace Visipics

Kapitola 5

Proces vývoje aplikace

Po dokončení shrnutí současného stavu. Začal proces vývoje. Proces se zakládal na použití Vodopádového modelu. Tento model je blíže popsán v kapitole 3.2 na straně 10.

5.1 Analýza požadavků

Ze zadání vyplynuly následující funkce:

- Aplikace najde a setřídí duplikace do skupin. V těchto skupinách umožní uživateli vybrat, které si chce ponechat.
- Zobrazení metadat fotografie.
- Zobrazení ignorovaných fotografií bez metadat a umožnit jejich smazání.
- Omezený výběr a následný bezeztrátový export fotografií na základě zadaných filtrů.
- Duplikace je určena podle Exif metadat.

Při analýze byl také proveden průzkum existujících řešení, který je podrobněji popsán v kapitole 4. Z něho vyplývají další specifikace:

- Možnost určit duplikace také podle IPTC metadat.
- GUI v češtině
- Možnost volby úplné nebo částečné duplikace metadat.
- Automatický výběr fotografie pro ponechání.

Také byly zjištěny nefunkční požadavky na aplikaci:

- Hlavní podporovaná platforma je Windows, a to hlavně ve verzi 7 a 10. Sekundární je pak podpora operačních systémů typu Linux a Mac. Hlavní je podpora JPEG formátu.
- Aplikace nesmí zamrznout, naopak musí uživateli dát zpětnou vazbu v jakékoliv fázi zpracovávání. Může být spuštěno pouze jedno prohledávání v jeden okamžik. Je tedy nutné zablokovat příslušná tlačítka uživateli. Důležitá je také stabilita a správnost určení duplikace.

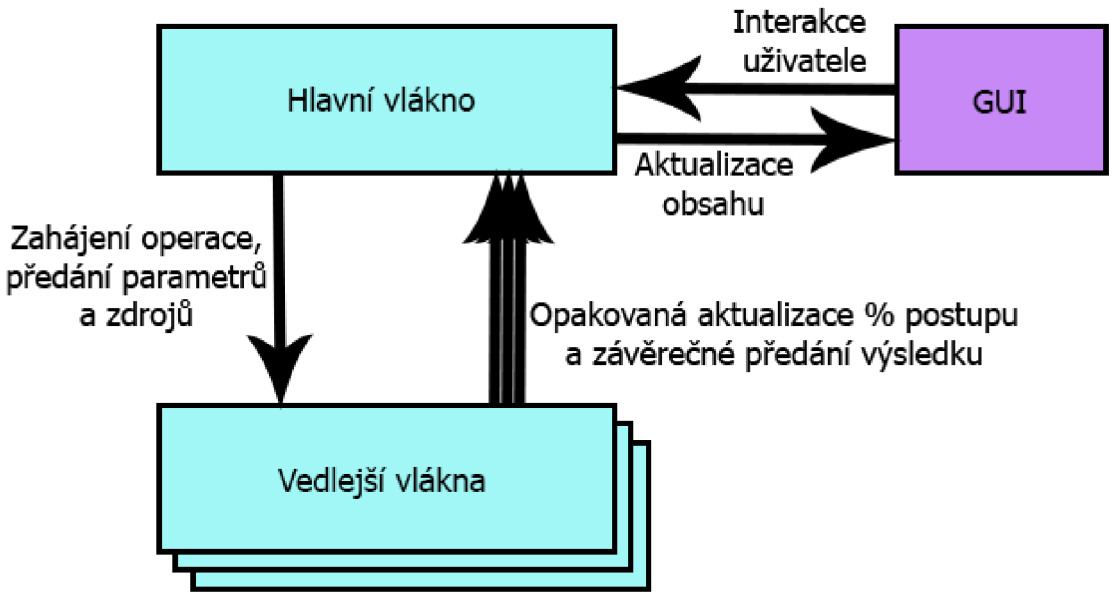
Plán vývoje

Nejdříve bylo nutné nastudovat teorii a existující řešení. Poté začít s vývojem funkcí, tedy hlavně backendu aplikace. Následoval návrh a vývoj GUI. Vzájemné propojení GUI a funkcí z backendu aplikace. Pak přišla na řadu verifikační část, kde jsem ověřoval hlavně zda aplikace splňuje zadané požadavky. Pomocí výstupů testovacího skriptu se stanovenými vzorovými výstupy. Nakonec byly v plánu výkonnostní testy a případné optimalizace. V poslední fázi také bylo nutné vyřešit a pokud možná co nejvíce zjednodušit instalaci aplikace.

Pro vývoj byl využit verzovací systém Git. Hlavně pro svoji výhodu vrácení jakékoliv změny ale také pro možnost okamžitého sdílení repozitáře s vedoucím. Git je blíže popsán v kapitole 3.1 na straně 8. Repozitář byl založen 17.09.2017. Součástí repozitáře je i subrepozitář se zdrojovými kódy dokumentace. Více o repozitáři na straně 42, repozitář se nachází zde [12].

5.2 Návrh aplikace

Multiplatformní aplikace, která bude snadno rozšiřitelná o další funkce. Hlavní význam aplikace je v pročištění dlouhodobého úložiště fotografií. Zejména pokud jsou zde upravené fotografie a jejich originály. Pokud uživatel pracoval s editory, které zachovaly metadata fotografie. Aplikace bude schopná tyto fotografie najít a seskupit s originálem. Uživatel pak výběrem z této skupiny určí, kterou fotografii si chce ponechat, zbytek je automaticky smazán po potvrzení výběru. Všechny požadované funkce musí být přístupné přes GUI.



Obrázek 5.1: Náhled vláken aplikace

Testování aplikace

Na začátku je nutné stanovení parametrů testovací galerie a její vytvoření. Sepsání parametrů testovacího stroje.

- **Scénaře**
  Scénáristické testy budou prováděny v průběhu vývoje. Mají za úkol průběžně ověřit plnění požadavků na aplikaci. Také správný směr vývoje.

  Aplikace bude v průběhu vývoje prověřována těmito scénáři:

  - První bude ověřovat možnost opatrného a podrobného přístupu. Uživatel si chce zadat všechny zdrojové složky sám. Nastavit možnost automatického výběru a mazání do koše, aby v případě chyby nepřišel o fotografii úplně. Rozhodně si chce také zobrazit podrobné údaje o každé fotografii. Při upravování výběru fotografií pro ponechání. Chce vidět upozornění, že fotografie budou smazány. Na konci si překontroluje jejich počet.
  - Druhý bude zaměřen spíše na rychlý přístup k uvolnění volného místa. Uživatel chce zadat pouze jednu složku a zahrnout všechny její podsložky automaticky. Popřípadě přidat další složku rychleji než tu první. Do nastavení se nedívat opakovaně, pouze při prvním spuštění aplikace. Zobrazit a smazat ignorované fotografie. Neupravit automatický výběr a okamžitě smazat fotografie nevybrané pro ponechání. Potvrdit a prohlédnout si jejich počet.
  - Poslední scénář je zaměřen na zálohu vybraných fotografií. Uživatel zadá zdrojové složky. Zaškrtne možnost podsložek. Zadá filtry pro vyhledání fotografií. Poté proběhne automatický výběr nejvhodnějších fotografií pro ponechání podle nastaveného parametru. Uživatel zvolí export, zadá název a umístění archivu. Pokud se mu nelíbí výchozí hodnoty.

- **Automatické**
  Automatické testy budou prováděny spíše v závěru vývoje aplikace. Jejich úkolem bude zatížit aplikaci a prověřit její stabilitu. Zjistit její výkon a odhalit možné slabiny algoritmu pro vyhledání duplikací. Hlavně jeho správnost a opakovatelnost.

- **Instalace**
  V úplném závěru vývoje. Bude testována také přenositelnost aplikace. Jednoduchým ověřením bude spuštění aplikace na jiném než testovacím stroji bez vývojového prostředí a minimální instalace vedlejšího softwaru.

Programovací jazyk

Jednou z prvních a nejdůležitějších věcí je volba jazyka ve kterém bude celá aplikace napsána.

Nároky na jazyk byly dány hlavně zadáním, bylo tedy nutné vytvořit GUI pro interakci s uživatelem tzv. frontend, který bude využívat funkcí z backendu. Nakonec byl pro implementaci zvolen jazyk Python, a to ve verzi 3.6.2. Při výběru jsem zvážil všechny vlastnosti jazyka popsané v kapitole 3.3 na straně 12. Zajímalo mě také jak bude vypadat vývoj složitější aplikace v tomto spíše skriptovacím jazyce.

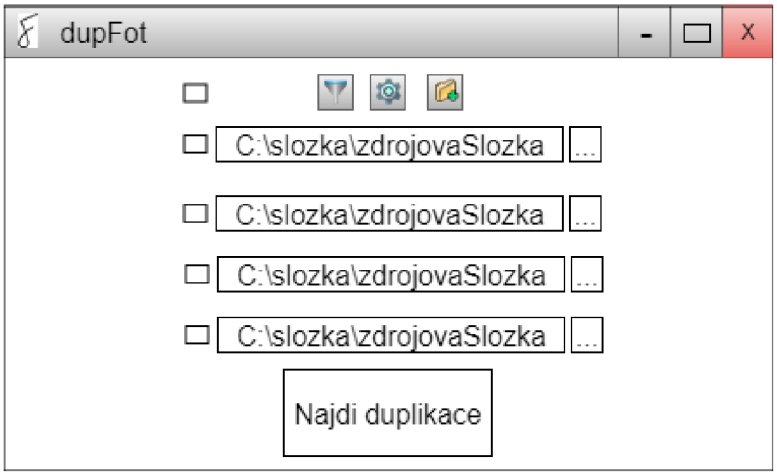
Grafické rozhraní

Grafické uživatelské rozhraní bylo navrhnuto pomocí nástroje www.draw.io. Návrh čtyř nejvíce důležitých a nejsložitějších oken je uveden níže.

- **Okno s informacemi o fotografii**
  Musí uživateli zobrazit všechny dostupné informace o fotografii. Pro usnadnění identifikace fotografie mezi ostatními. Uživatel si tedy po zobrazení tohoto okna musí být jistý o jakou fotografii se jedná.

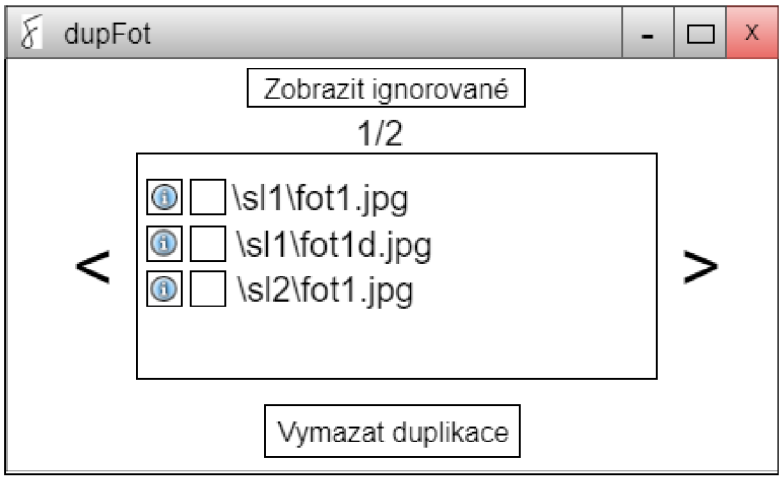
Obrázek 5.2: Náčrt okna s informacemi o fotografii

- **Základní okno**
  Musí uživateli umožnit zadání zdrojových složek, nastavení parametrů vyhledání duplikací a také jeho spuštění.



Obrázek 5.3: Náčrt základního okna aplikace

- **Okno duplikací**
  Zobrazí vyhledané skupiny duplikací. Na základě zdrojů a nastavení. Musí uživateli umožnit manuální výběr, ale zároveň zobrazit automatický výběr aplikace. Také zde musí být možnost zobrazit ignorované fotografie, které neobsahují Exif metadata.



Obrázek 5.4: Náčrt okna duplikací aplikace

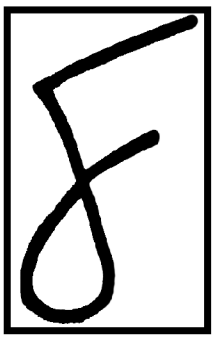
Některé funkce jako třeba export fotografií byly do aplikace přidány až v samotném závěru vývoje a v původním návrhu tak jejich ovládací prvky chybí.

- **Okno nastavení**
  Musí uživateli umožnit změnu a zobrazení aktuálního nastavení. Také návrat k původnímu nastavení aplikace.

Obrázek 5.5: Náčrt okna s nastavením aplikace

Název a logo aplikace

Pro prezentování a snadnější identifikaci aplikace. Je dobré vytvořit jednoduché logo a snadno zapamatovatelný logický název. Název vznikl spojením dvou slov: duplikace a fotografie. Slova jsou navzájem spojena stylem camel, kde druhé slovo začíná velkým písmenem a jde o běžnou programátorskou praktiku například při pojmenovávání proměnných. Tedy dupFot. Logo vzniklo spojením počátečních písmen těchto slov písmene d a F. Základní myšlenkou loga a názvu je jednoduchost, logo třeba lze nakreslit jedním tahem.



Obrázek 5.6: Logo aplikace

Kapitola 6

Implementace

Po definování všech potřebných funkcí v předchozí části vývoje. Bylo potřeba vymyslet reálné řešení zadaných specifikací.

Vícevláknová aplikace se signály pro synchronizaci. Hlavní vlákno obsluhuje GUI, na pozadí probíhají ostatní procesy, které mohou vytvořit další vlákna. Hlavní vlákno je synchronizováno pomocí signálů. Pokud je úloha v pozadí dokončena je emitován příslušný signál a GUI je aktualizováno. Více informací v kapitole 5.2.
Vlákna jsou vytvořena za pomocí vestavěné knihovny `threads`.

Grafické uživatelské rozhraní je implementováno použitím knihovny `PyQt5` v souboru gui.py. Každá grafická komponenta má zde svoji třídu, která dědí od třídy `QWidget`. Také funkci pro její správné zobrazení a uzavření. Všechny aktivní grafické komponenty jsou uloženy do společného seznamu, mohou tak být hromadně uzavřeny. Zároveň je nutné uchovávat referenci na instanci kvůli `garbage collectoru`.

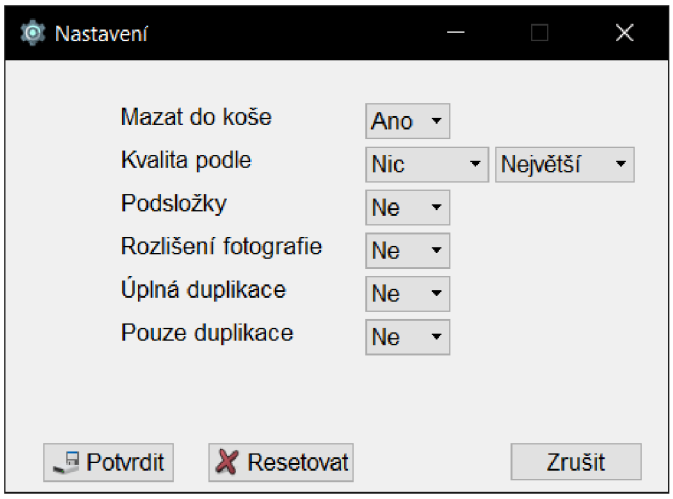
Pokud se v průběhu vykonávání aplikace objeví výjimka, je zachycena a její popis je zapsán do logovacího souboru. Soubor se vytvoří automaticky při první chybě a poté už jenom navazuje na předchozí. Nemaže tedy žádnou chybu. Tento soubor má nastavený pevný název `dupFot.log`.
Uzavření hlavního okna ukončí celou aplikaci.

6.1 Nastavení aplikace

Snad každá dobrá aplikace má své nastavení. Aby ji mohl uživatel lépe používat právě pro své konkrétní potřeby. Ani moje tedy není výjimkou. Lze nastavit:

- způsob mazání fotografií(koš, úplné)
- preferenci automatického výběru (nic, velikost, datum poslední změny, priorita složky, každou možnost lze znegovat)
- prohledávání podsložek pro nově zadané složky
- hledání duplikace (částečná nebo úplná duplikace metadat, vyloučení rozlišení z metadat)
- zobrazení pouze skupin duplikací nebo i jednotlivých fotografií (první se zobrazí skupiny duplikací, teprve poté jedinečné fotografie)

Nastavení se ukládá do souboru při jeho první změně a potvrzení. Je vytvořen soubor dupFot.sett. Který obsahuje uloženou instanci třídy Settings. Při spuštění aplikace je primárně načítán tento soubor, pokud neexistuje jsou nastaveny výchozí hodnoty.



Obrázek 6.1: Výchozí nastavení aplikace

6.2 Načtení fotografií a vyhledání duplikací

Uživatel zadá zdrojové složky a u každé nastaví, zda mají při hledání být zahrnuty i podsložky. Úroveň zanoření potom už není nijak omezena. Počet zdrojových složek je v GUI omezen na 18.

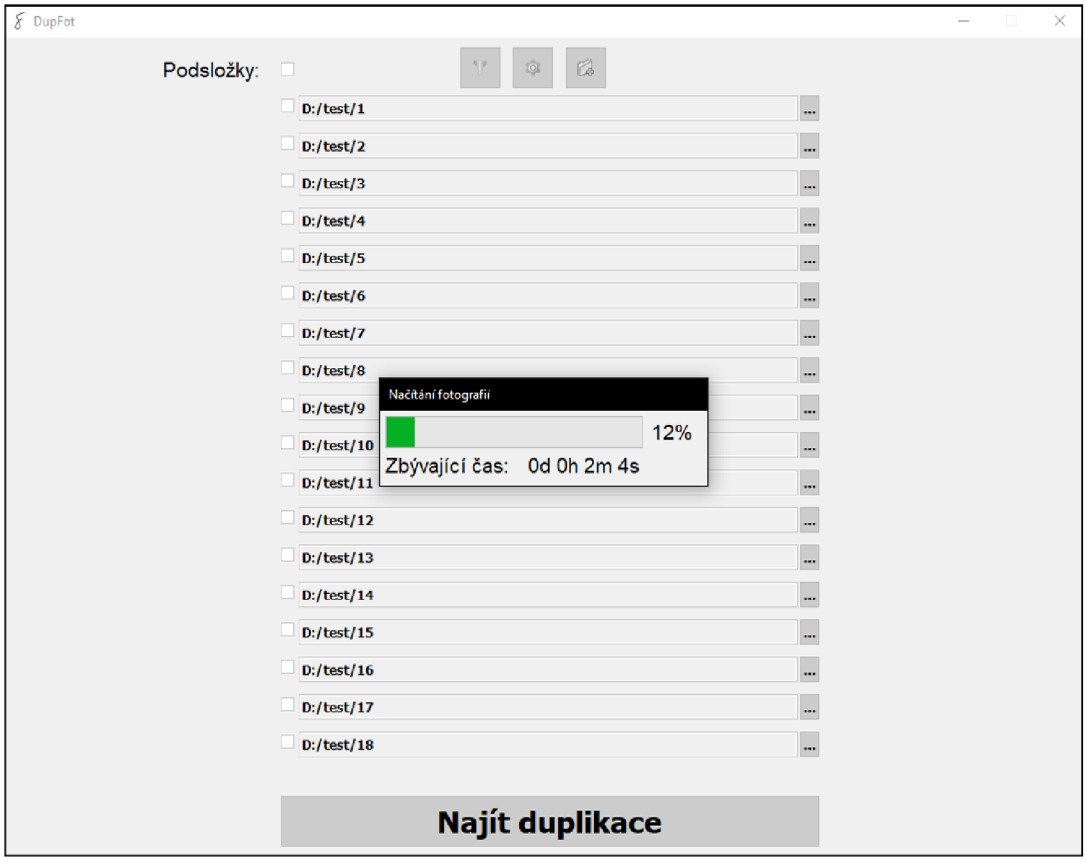
Jako výchozí adresář pro selekci je zvolena nadřazená složka poslední zadané složky. Pokud ji nemá, tak je to poslední vybrána složka. Pro první výběr je nastavena výchozí složka na `\home`.

Pro každou zdrojovou složku je vytvořeno jedno vlákno, to pak projde všechny podsložkuy a soubory samostatně. Vlákna se synchronizují na konci procházení.

Načteny jsou všechny fotografie, které se nacházejí v zadaných složkách, popřípadě podsložkách.

Primárně jsou vyhledány ExIf metadata, sekundárně jsou pak hledány IPTC údaje o fotografii. Pokud fotografie nemá ani jeden typ z těchto metadat je zahrnuta do seznamu ignorovaných. Výstupem načtení jsou tedy dvě skupiny fotografií.

Odhadovaný čas je vypočítán jako rozdíl aktuálního času a počátečního času načítání. Tento rozdíl je vydělen počtem hotových procent. Tím je získán průměrný čas na jedno procento. Tento průměrný čas je vynásoben počtem zbývajících procent. Tedy rozdílem 100 a počtem hotových procent. Tento čas se mění každé další procento. Je tedy čím dál více přesnější a zmenšuje se.

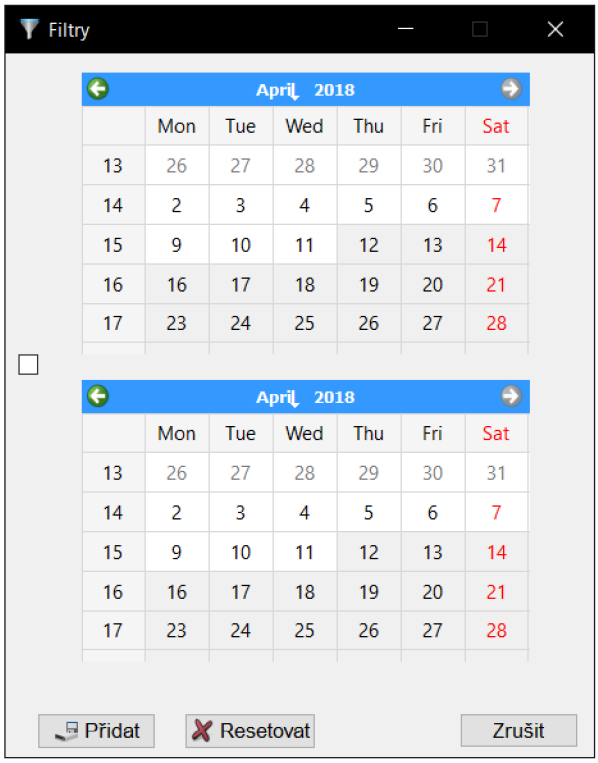


Obrázek 6.2: Zobrazení načítání fotografií

Filtrace

Ještě před samotným hledáním duplikace načtené fotografie projdou nastaveným filtrem. Dále pokračují pouze vyhovující.

- Časové intervaly
  Neomezené pole časových intervalů. Které se buď mají zahrnout nebo vyloučit z výběru.



Obrázek 6.3: Nastavení filtrů podle časových intervalů

- Velikost
  Minimální a maximální velikost fotografie.

V aktuální verzi 1.1 je funkční pouze filtr časových intervalů.

Hledání duplikace

V načtených a vyfiltrovaných fotografiích jsou vyhledány duplikace.

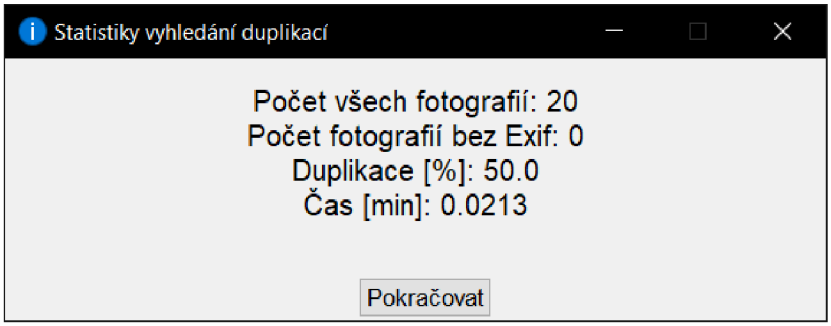
Všechny fotografie jsou načteny do jednoho pole. Toto pole je postupně procházeno. Každá fotografie je porovnána s následujícími od své pozice. Vyhledávání jde tedy pouze dopředu. Nalezené duplikace jsou vyloučeny z dalšího porovnávání. Pro každou fotografii tedy platí, že bude mít minimálně o jedna méně porovnání než předchozí. Ignorované fotografie nejsou porovnávány.

V případě ExIf jde nastavit, zda se má jednat o úplnou, nebo částečnou duplikaci. Při částečné shodě jsou vyloučeny některá metadata, která v průběhu testování upravili grafické editory. Je tedy pravděpodobné, že hledání úplné duplikace metadat najde pouze kopie, nikoliv upravené fotografie. Upravenou fotografií je zde myšlena modifikace obrazových dat, metadata by se v takovém případě měla přenést. Z testování ale vyplynulo, že některé grafické editory metadata upravují.

Duplicitní fotografie jsou seskupeny a je proveden automatický výběr nejvhodnější pro ponechání. Uživatel jej může úplně vypnout, popřípadě zvolit jiné kritérium pro automatickou volbu.

Statistiky vyhledávání

Na konci každého vyhledání duplikací jsou uživateli zobrazeny jeho statistiky.



Obrázek 6.4: Zobrazení statistik vyhledávání duplikací

6.3 Výběr fotografií a operace s nimi

Po seskupení duplikací je podle nastavení proveden automatický výběr fotografie vhodné pro ponechání. Poté je umožněn pře GUI výběr manuální.

- **Manuální**
  Uživateli jsou zobrazeny skupiny duplikací a z každé může vybrat libovolný počet fotografií pro ponechání. U každé fotografie lze prohlédnout její metadata a obrazová data. Do výběru lze v nastavení zahrnout i jednotlivé fotografie pro které není nalezena duplikace kvůli exportu.

- **Automatický**
  Je tedy proveden ještě před zobrazením skupin uživateli. Může být proveden podle:
  - priority složek (vrchní složka má nejvyšší prioritu)
  - velikost fotografie
  - datum poslední modifikace

  Také může být úplně deaktivován. Všechny možnosti lze použít sestupně i vzestupně.

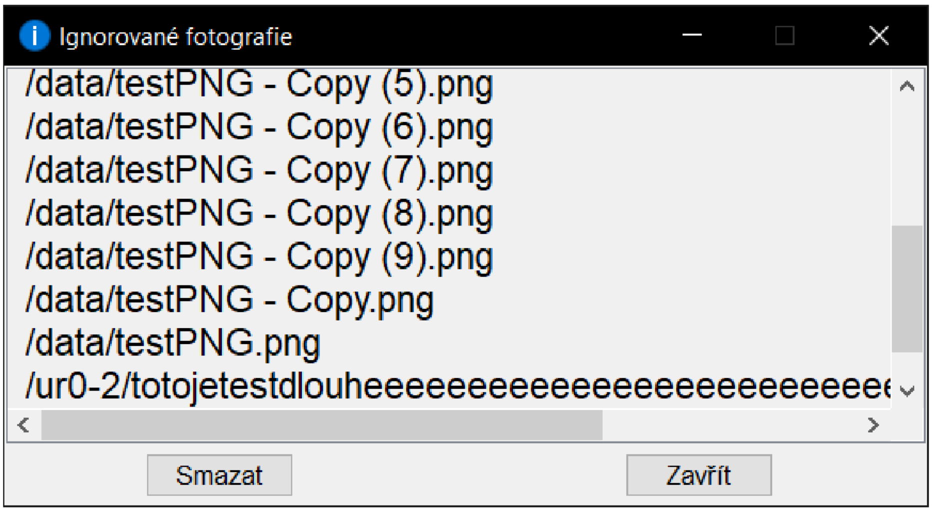
Export fotografií

Všechny označené fotografie jsou exportovány do archivu. Jehož umístění uživatel zvolí. Fotografie se třídí podle jejich přímo nadřazené složky.

Vymazání

- **Duplikací**
  Po potvrzení jsou smazány všechny neoznačené fotografie.
- **Ignorovaných fotografií bez metadat**
  Je možné pouze hromadné vymazání.

V nastavení lze zvolit mazání do koše nebo úplné. Mazání proběhne až po souhlasu uživatele. Po provedení je zobrazen počet smazaných fotografií.



Obrázek 6.5: Zobrazení ignorovaných fotografií

Kapitola 7

Vyhodnocení aplikace

Pro automatické, výkonnostní testování byl vytvořen skript test.py. Který každé vyhledání provede desetkrát, zaznamená dobu trvání každého vyhledání a zapíše medián z těchto výsledných časů. Poté přidá další složku do zdrojů a pokračuje. První kolo testování probíhá bez podsložek. Všechny fotografie jsou tedy jedinečné. Druhé kolo testování zahrne i podsložky, to tedy znamená mnohem více fotografií, ale také mnoho duplikací. Všechny výsledky jsou automaticky zapsány do xlsx souborů. Cílem je ukázat optimalizaci pro výskyt duplikací. Lze pomocí něj ověřit i stabilitu celé aplikace.
Také jej lze použit pro rychlé ověření správnosti nalezeného počtu fotografií a jejich duplikací v zadaných složkách. Pro tento účel je nutné snížit počet opakování tedy konkrétně proměnnou `count` na 1.

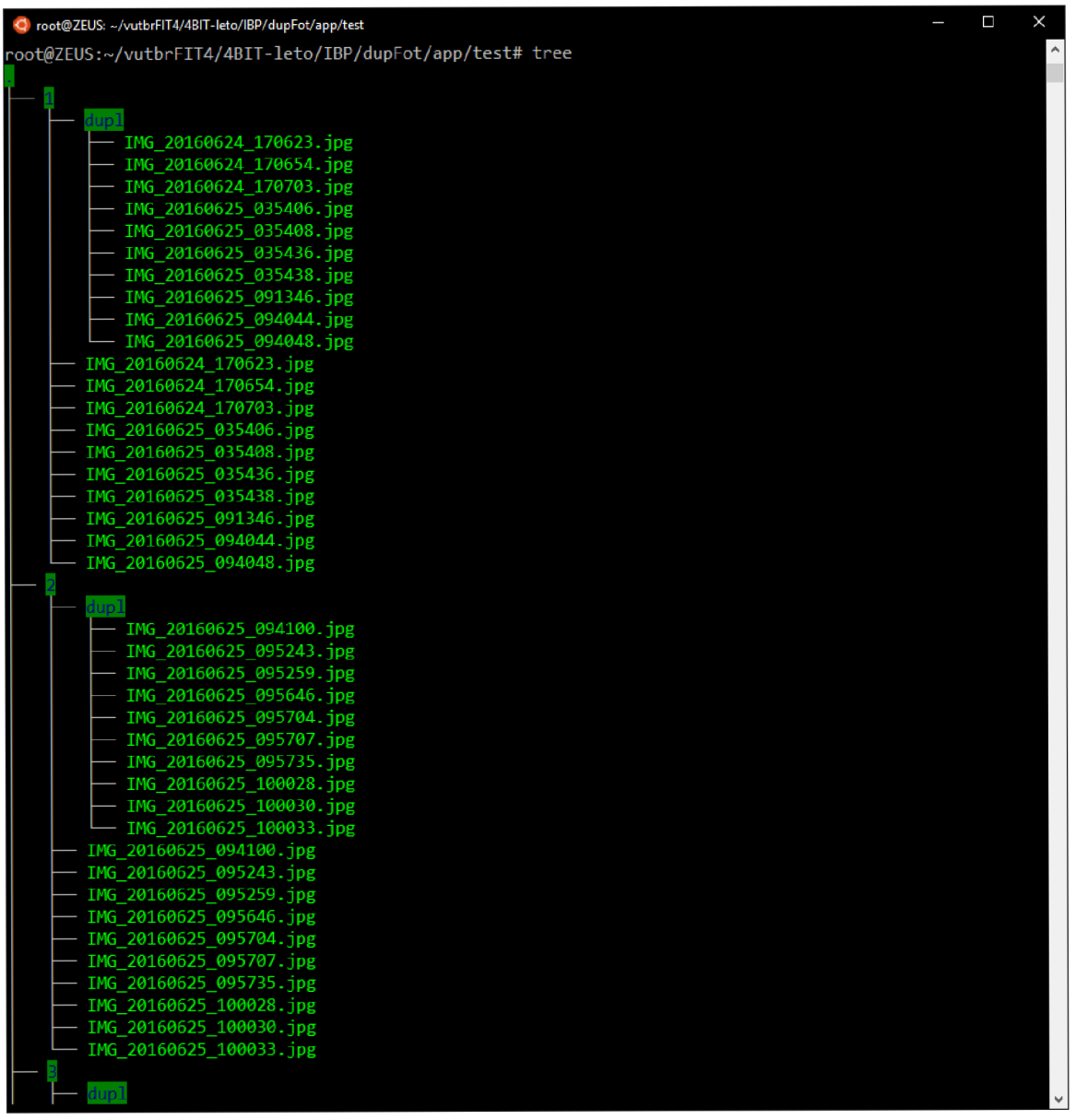
7.1 Konfigurace testovacího stroje

- **Operační systém**
  Windows10 Education, 16299.371, 64bit (na SSD)
- **Základní deska**
  Gigabyte 970A-D3
- **SSD**
  KINGSTON SHSS37A240G
- **Procesor**
  AMD FX-4350, Quad-Core 4.20 GHz
- **RAM**
  24 GB, DDR3, 2133 MHz, CL11
- **HDD**
  WDC WD20EZRX-00D8PB0, 2TB, 7200 RPM, cache 64 MB
- **GPU**
  AMD Radeon R9 200 Series, 3072 MB, GDDR5, 1100 MHz

Jako HDD je uveden disk na kterém byla umístěna aplikace a testovací galerie.
Všechny testy proběhly v minimálním zatížení stroje ostatními aplikacemi či procesy.

Testovací galerie

Skládá se z 40 složek. Každý adresář vždy obsahuje 10 fotografií a jejich duplikace umístěné ve svém podadresáři. Duplikace mohou být přesné kopie ale i upravené fotografie. Testované editory jsou Adobe Photoshop CS6 a Gimp. V testovací galerii jsou pouze fotografie formátu JPG.
Celkově je tedy zde 400 unikátních a 400 duplicitních fotografií. Průměrná velikost fotografie je 2,6125 MB.

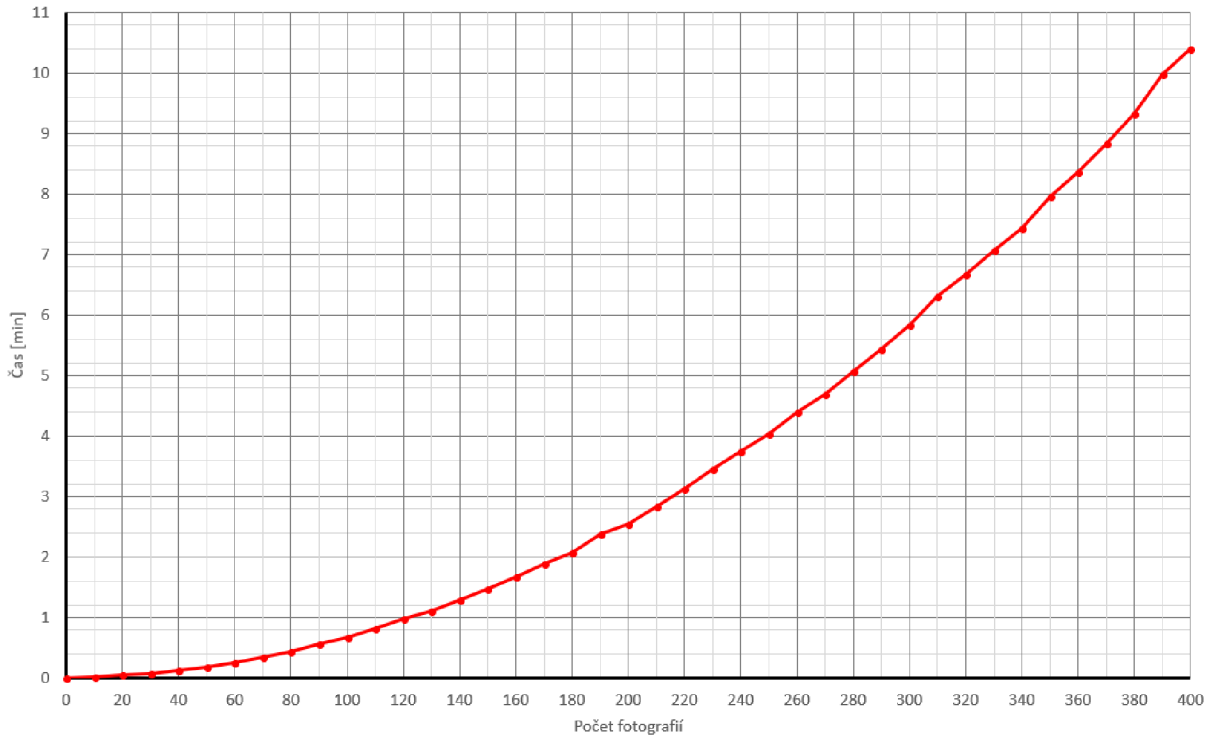


Obrázek 7.1: Ukázka stromového výpisu testovací galerie

7.2 Grafy z automatizovaného testování

Každý graf byl sestaven z průměrných hodnot dvou měření. Všechny časové údaje měření jsou zapsané v minutách z důvodu zbytečně velkých čísel pro sekundy. Měření bylo spuštěno, nad aplikací prováděné interpretem, která tedy nebyla nijak zkompilována.

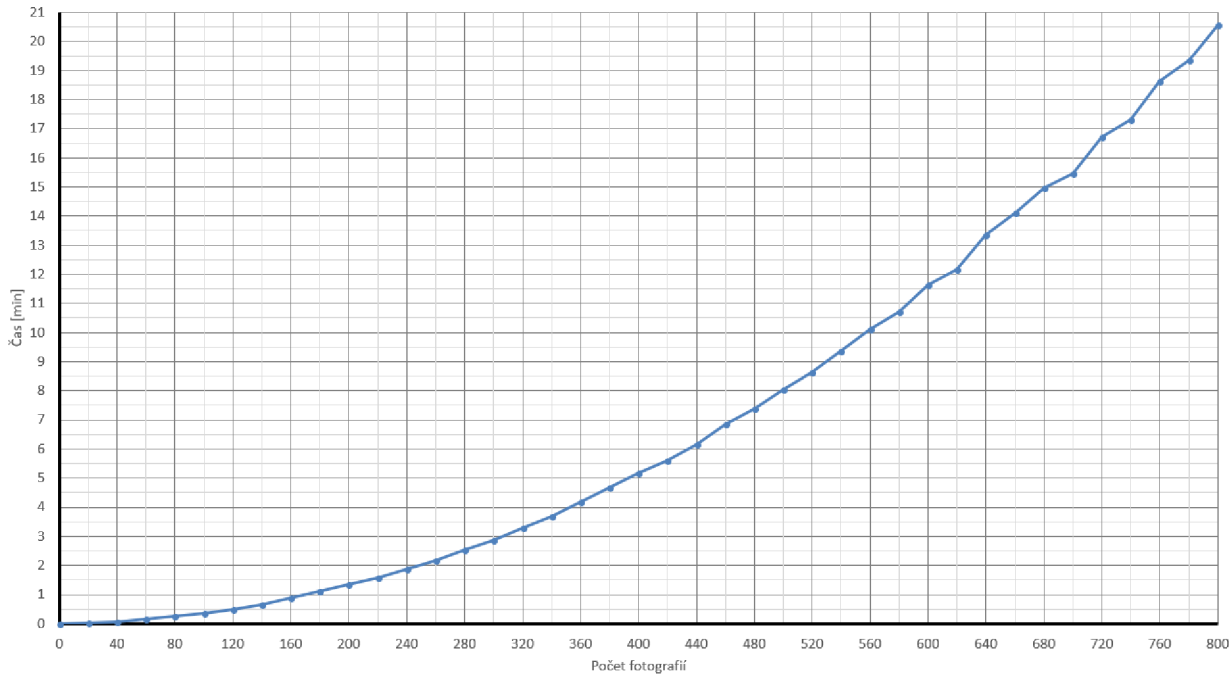
- **400 fotografií bez duplikace**
  Test **bez** zahrnutí podsložek s duplikacemi testovací galerie.



Obrázek 7.2: Graf testu 400 fotografií bez duplikace

Můžeme sledovat předpokládaný postupný nárůst potřebné doby zpracování na více fotografií.

- **800 fotografií s 50% duplikací**
  Test celé testovací galerie, pro ukázání pomalejšího nárůstu doby zpracování. V případě výskytu duplikací. Díky optimalizacím kódu.



Obrázek 7.3: Graf testu 800 fotografií s 50% duplikací

Na tomto grafu stojí za povšimnutí hlavně údaj o zpracování 400 fotografií s 50% duplikací. Díky optimalizacím pro výskyt duplikací jde o méně než poloviční hodnotu (5.1743), oproti údaji z předchozího testu pro stejný počet fotografií bez duplikace (10.4117).

Podrobné údaje ze všech měření lze najít ve zdrojových souborech dokumentace v složce tests.

7.3 Porovnání s konkurencí

Pro objektivní hodnocení aplikace je také důležité porovnání s konkurencí. Aplikace by měla nabídnout něco navíc a také se něčím odlišit. Do porovnání s aplikací dupFot jsou zahrnuty vlastnosti: vzhled, dostupnost, způsob určení duplikace a jeho účinnost.

Všem aplikacím bylo zadáno vyhledat duplikace z celé testovací galerie. U této operace byly zkoumány i ostatní aspekty aplikací. Všechny aplikace jsou blíže popsány v kapitole 4 na straně 13.

Název aplikace	**GUI**	**Určení duplikace**	**Licence**	**Správně**	**Čas [min]**
dupFot	Ano	Metadata	Otevřená	Ano	20.58
dupFot kompil.					11.16
VisiPics	Ano	Obraz	Zdarma	Ne	X
ExifTool	Ne	Metadata	Zdarma	Nelze	X
DP Fixer Pro	Ano	Obraz	Omezená	Ano	2.55
VS DIF Demo	Ano	Obraz	Omezená	Přijatelné	1.57
		Metadata			1.23
Anti-Twin	Ano	Obraz	Zdarma	Ano	146.78
		Bitové porovnání			25.16

Tabulka 7.1: Porovnání aplikace dupFot s konkurencí

DP Fixer - Duplicate Photos Fixer
VS DIF - Visual Similarity Duplicate Images Finder
dupFot kompil. - zkompilovaná aplikace dupFot pomocí knihovny cx_freeze [26]

Výsledek byl správně, pokud aplikace nalezla všechny duplikace v testovací galerii pouze zadáním nejvyšší zdrojové složky. Přijatelný výsledek znamená sloučení některých hodně podobných skupin. Demo aplikace VS DIF nezobrazí všechny skupiny duplikací.

7.4 Testování stability aplikace

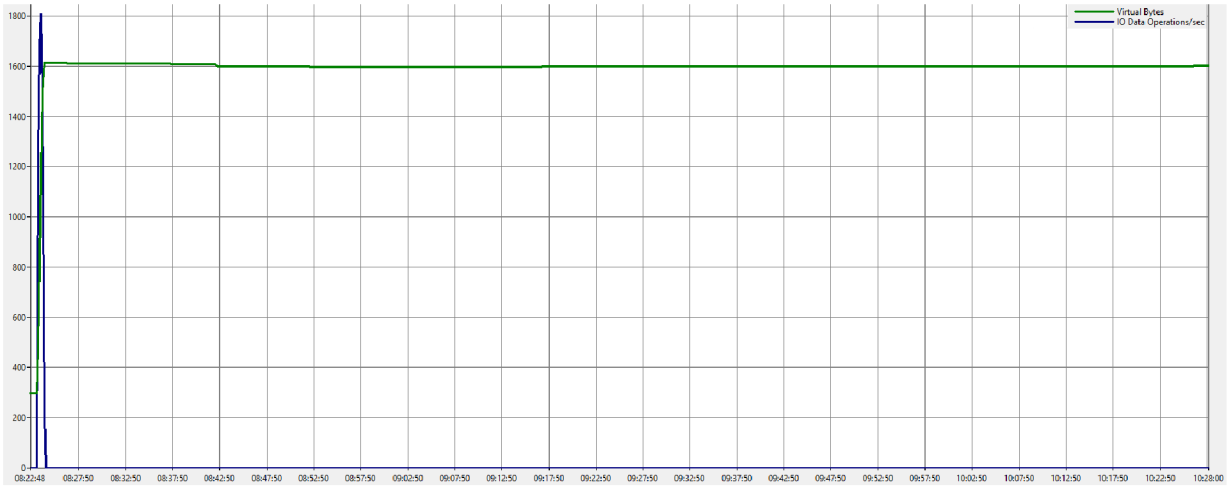
V rámci ověření stability zkompilované aplikace byly provedeny testy na velkou a dlouhodobou zátěž. Zadání rizikových hodnot do vstupů aplikace.

Velikost galerie [GB]	**Počet fotografií**	Duplikace [%]	**Správně**	**Čas [min]**
0.0	0	0.00	Ano	0.0004
2.0	757	49.80	Ano	10.14
25.0	11 880	87.99	Ano	124.82
50.0	23 760	88.05	Ano	130.65

Tabulka 7.2: Objemové testování zkompilované aplikace dupFot

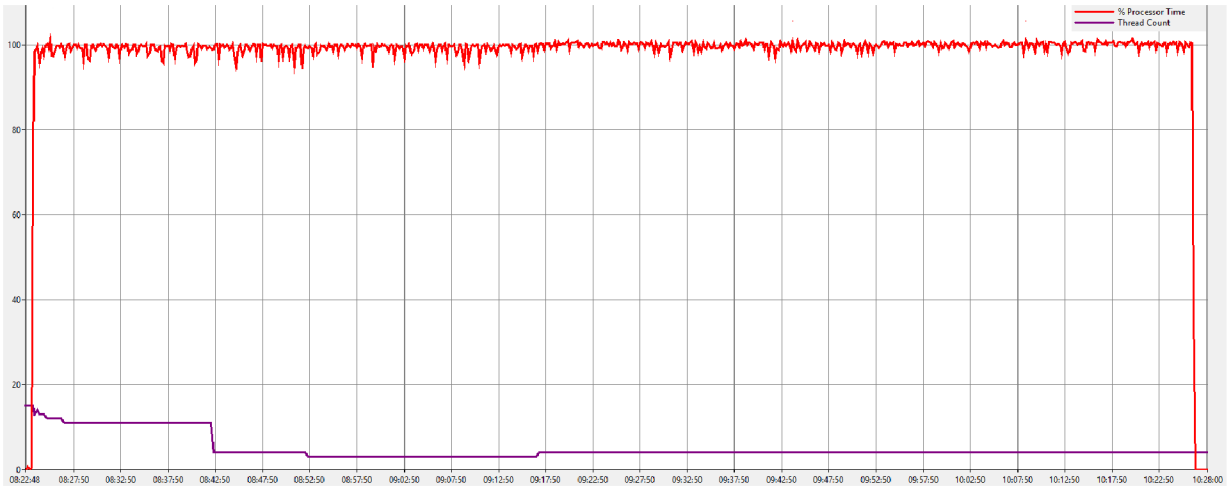
V každé skupině duplikací je vždy jedna fotka brána jako originál. Takže 100% výskyt duplikací nikdy nenastane. Fotografie byly ve formátu JPEG. Zapsané časy pocházejí z jednoho měření, slouží tedy pouze pro orientaci.

- **Graf využití HDD a RAM**



Obrázek 7.4: Graf využití HDD a RAM aplikací dupFot při zpracování 50 GB galerie s 88% duplikací

- **Graf využití CPU**



Obrázek 7.5: Graf využití CPU aplikací dupFot při zpracování 50 GB galerie s 88% duplikací

% Processor Time – je procento uplynulého času, které procesor tráví, aby provedl podproces, který není nečinný. Jde o % vytížení všech jader procesoru, kde každé jádro znamená 100%. Pro použitý testovací stroj se 4 jádry je tedy maximum této hodnoty 400%.

Stroj měl minimální vedlejší zatížení zdrojů, které bylo odfiltrováno. Výsledné zatížení je tedy pouze od procesů aplikace. Data byla posbírána přes Windows Performace Monitor. Více v přílożené XML šabloně ve složce tests.

Z naměřených grafů využití je zřejmé, že aplikace využívá pevný disk pouze při úvodním načtení fotografií. Poté už žádná práce s diskem neprobíhá a pracuje pouze s fotografiemi načtenými do virtuální paměti.

Kvůli jednovláknovému algoritmu pro vyhledání duplikací aplikace nedokáže využít celý procesor. Této věci jsem si vědom avšak pro verzi 1.1 platí, že výkon musel ustoupit správné a stabilní funkčnosti aplikace. Více vláknová implemetace vyhledávacího algortimu je uvažována v rozšíření. Jak lze vidět na začátku, načítání fotografií již vícevláknové je.

Aplikace byla testována i na jiných strojích a galeriích. Tyto testy zde ale kvůli rozsahu práce neuvádím, protože jejich výsledky byly přibližně stejné.

Testování hraničních hodnot

Vzhledem k možnostem vstupů od uživatele, bylo jako nejvíce rizikové místo vyhodnoceno textové pole pro zadání složky. Je to jediný vstup, kde uživatel může zadat cokoliv. Jinak je omezen vždy nabídnutými možnostmi. Tento vstup byl tedy otestován ručně zadáním několika dlouhých a náhodných hodnot. Také byly vyzkoušeny různé kombinace nastavení aplikace.

7.5 Zhodnocení

Po porovnání mé aplikace dupFot s konkurencí jsem do budoucího rozšíření přidal několik dalších funkcí. Celkově ale aplikace vyšla v porovnání dobře. Ostatním aplikacím vždy něco chybí ať už podpora více platforem, otevřená licence. Většinou porovnávají pouze na základě obrazových dat a vůbec tak nepracují s metadaty což je pomalé. Také tímto způsobem nemohou najít více modifikované fotografie. Některé výsledky ale také ukazují, že by aplikaci ještě mělo být možné více optimalizovat.

Limity aplikace

V rámci analýzy a následně během testování na reálných datech byly zjištěny následující omezení. Vzhledem k určení duplikace pouze na základě metadat fotografie, je zde velká nevýhoda pro fotografie stažené z internetu (hlavně ze sociálních sítí). Kde jsou některá, důležitá metadata promazávána automaticky (datum, poloha, ...). Také pro starší foťáky, které neumí zaznamenat datum či polohu pořízení fotografie. Tyto údaje jsou pro správné určení duplikace velmi důležité.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo nastudovat problematiku určení duplicitních fotografií podle Exif metadat a vytvořit aplikaci, která tyto duplicity najde a umožní uživateli jejich hromadné vymazání či přesun.

Výsledná aplikace prošla všemi stanovenými scénáristickými testy a splnila všechny jejich předpoklady, viz kapitolu 5.2. Svými funkcemi prošla verifikačním testováním sestaveném na základě zadání. Některé funkce byly přidány nad rámec zadání. Všechny z přidaných funkcí ale nejsou, v odevzdané verzi 1.1, přístupné uživateli přes GUI viz kapitolu 5.1.

V budoucím vývoji aplikace je potřeba se zaměřit hlavně na otestování dalších platforem. Také zapracovat na další optimalizaci algoritmu pro vyhledání duplicitních fotografií. Důležité je také zpřístupnění všech funkcí uživateli přes GUI. Některé funkce totiž z časových důvodů zůstaly naprogramované pouze v pozadí aplikace. Například hromadná i jednotlivá změna metadat. Nebo seskupení fotografií podle zadaného rozsahu vzdálenosti z GPS údajů, také jejich případná změna o zadaný úsek.

S volbou programovacího jazyka jsem spokojen, splnil vše, co se od něj očekávalo a v průběhu vývoje nebyl důvod k přehodnocení této volby. Vývoj složitější aplikace v jazyce Python 3.6.2 byl velmi zajímavý a v mnoha ohledech jiný než u jeho konkurence. Nicméně nakonec se mi vždy podařilo dosáhnout toho co jsem potřeboval.

Literatura

[1] Domovká stránka Git. Software Freedom Conservancy, [Online; navštíveno 30.04.2018].
URL https://git-scm.com/

[2] Domovká stránka nástroje TortoiseGit. TortoiseGit and contributors, [Online; navštíveno 30.04.2018].
URL https://tortoisegit.org/

[3] Domovká stránka SVN. The Apache software foundation, [Online; navštíveno 30.04.2018].
URL https://subversion.apache.org

[4] IPTC domovská stránka. International Press Telecommunications Council, [Online; navštíveno 28.03.2018].
URL https://iptc.org/standards/photo-metadata

[5] Shrnující popis jazyka JavaScript. Tvorba-webu.cz, [Online; navštíveno 17.04.2018].
URL https://www.tvorba-webu.cz/javascript

[6] Shrnující popis jazyka PHP. Tvorba-webu.cz, [Online; navštíveno 17.04.2018].
URL https://www.tvorba-webu.cz/php/

[7] Shrnující popis jazyka Python. Python Software Foundation, [Online; navštíveno 07.04.2018].
URL https://docs.python.org/3/tutorial

[8] TIOBE žebříček popularity jazyků. TIOBE software BV, [Online; navštíveno 26.04.2018].
URL https://www.tiobe.com/tiobe-index/python

[9] Top 5 Best Duplicate Photo Finder Tools to Remove Duplicate Photos. Ashisoft, [Online; navštíveno 13.04.2018].
URL https://www.ashisoft.com/blog/top-5-best-duplicate-photo-finder-to-delete-duplicate-photos

[10] *JEITA CP-3451*. Japan Electronics and Information Technology Industries Association, 2012, [Online; navštíveno 25.03.2018].
URL http://www.exif.org/Exif2-2.PDF

[11] *.ZIP File Format Specification, version 6.3.4*. 2014, [Online; navštíveno 02.04.2018].
URL https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT

[12] (Artonix14), Z. S.: Repozitář s aplikací. [Online; navštíveno 11.04.2018].
URL https://github.com/Artonix14/dupFot.git

[13] Ben Collins-Sussman, C. M. P., Brian W. Fitzpatrick: *Version Control with Subversion*. 2002, [Online; navštíveno 30.04.2018].
URL http://svnbook.red-bean.com/en/1.7

[14] Carbonnelle, P.: PYPL žebříček popularity jazyků. [Online; navštíveno 26.04.2018].
URL http://pypl.github.io/PYPL

[15] Dolejší, T.: XMP soubor, nenápadný dříč. [Online; navštíveno 25.03.2018].
URL https://www.fotoradce.cz/xmp-soubor-nenapadny-dric

[16] Harvey, P.: ExifTool Home Page. [Online; navštíveno 14.04.2018].
URL http://owl.phy.queensu.ca/~phil/exiftool

[17] helloworldcollection: 574 Hello World programs. [Online; navštíveno 17.04.2018].
URL https://helloworldcollection.github.io/

[18] Ing. Radek Kočí, P.; Ing. Bohuslav Křena, P.: Úvod do softwarového inženýrství IUS (2. přednáška), 2014.
URL https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIUS-IT%2Flectures%2FZS-2014-15%2FIUS2.pdf&cid=10027

[19] Jahoda, B.: Jaký formát obrázku použít na webu. 2015, [Online; navštíveno 18.04.2018].
URL http://jecas.cz/format-obrazku

[20] Khatri, M.: 5 Best Duplicate Photo Finder Tools to Delete Duplicate Photos. [Online; navštíveno 13.04.2018].
URL https://blogs.systweak.com/2016/04/5-duplicate-photo-finder-tools-to-delete-duplicate-photos

[21] Kužel, J.: *Editor EXIF a IPTC údajù z fotografií*. Bakalářska práce, Vysoké učení technické, Fakulta informačních technologií, Brno, 2011.

[22] Macenauer, A.: Co je digitální fotografie. 2002, [Online; navštíveno 18.04.2018].
URL https://www.fotoaparat.cz/clanek/tisk/230

[23] Nazaruk, T.: Metadata: Skryté informace o fotografii. [Online; navštíveno 23.03.2018].
URL https://www.stopfake.org/cz/metadata-skryte-informace-o-fotografii/

[24] Peters, T.: Zásady programování v jazyce Python. [Online; navštíveno 01.04.2018].
URL https://www.python.org/dev/peps/pep-0020

[25] Sokolov, V.: Handling ZIP Archives in pure MQL5. 2015, [Online; navštíveno 16.04.2018].
URL https://www.mql5.com/en/articles/1971

[26] Tuininga, A.: *cx_Freeze Documentation*. 2017, [Online; navštíveno 25.03.2018].
URL https://media.readthedocs.org/pdf/cx-freeze/latest/cx-freeze.pdf

[27] van der Wolf, H.: pyExifToolGui Home Page. [Online; navštíveno 14.04.2018].
URL https://hvdwolf.github.io/pyExifToolGUI

[28] Černoch, P.: Digitální fotografie - vznik, historie, současnost a vývojové trendy. 2005, [Online; navštíveno 18.04.2018].
URL https://www.fi.muni.cz/usr/jkucera/pv109/2003/xcernoc1

Příloha A

Obsah přiloženého média

Přiložené médium obsahuje repozitář aplikace dupFot 1.1, aktuální k datumu 14.05.2018.

- **zdrojový kód**
  Včetně komentářů ke každé třídě, metodě, funkci a souboru samotnému na jeho na začátku.
- **dokumentace**
  Vytvořené PDF a všechny zdrojové soubory LaTeXvčetně Makefile a použitých obrázků.
- **zkompilovaná aplikace**
  Pro OS typu Windows. Na ostatních OS nebylo provedeno testování ani kompilace, nicméně kompilovací skript by měl být připraven na OS typu Linux a MacOS.

Rozdělení zdrojového kódu

Kvůli lepší orientaci ve zdrojovém kódu aplikace. Byla zavedena následující hierarchie. Každý soubor tvoří samostatný celek a využívá předchozích souborů ve směru odshora dolů v následujícím pořadí. Více informací lze naleznout v komentářích v každém z nich.

- **classes.py**
  Soubor pro třídy, které se netýkají GUI.
- **globalVars.py**
  Soubor pro vytvoření globálních proměnných.
- **lib.py**
  Soubor pro funkce backendu aplikace.
- **gui.py**
  Soubor pro třídy a jejich metody, funkce pro vytvoření a práci s GUI.
- **main.py**
  Soubor pro spuštění aplikace.

- **setup.py**
  Soubor pro kompilaci aplikace.

- **test.py**
  Soubor pro testování aplikace. Může také sloužit jako ukázka pro další použití v jiných programech. Není potřeba pro kompilaci ani běh aplikace. viz kapitola 7.1 na straně 31.

Instalace aplikace

Je provedena spuštěním instalačního skriptu. Který je naprogramován jako vše v jazyce Python. Pro jeho spuštění je nutné zadat v příkazové řádce/terminálu následující příkaz:
`~/dupFot/app> python ./setup.py build`

Pro kompilaci aplikace je tedy nutné mít nainstalovaný Python 3.6.x. Potřebné knihovny doinstaluje skript automaticky. Testováno na verzi 9.0.1 nástroje `pip`.

Skript umožňuje kompilaci a vytvoření spouštěcího souboru pro daný typ operačního systému. Je zde možnost pro Linux, Windows a Mac. Kompilace je provedena za pomocí knihovny *cx_freeze.* [26]

Zkompilovaná aplikace je kompatibilní v rámci stejného typu operačního systému na kterém byla kompilace provedena. Aplikace má předpoklady být multiplatformní na základě použití Pythonu 3.6.2 a jeho standartních, volně dostupných knihoven.

Na Windows doporučuji přidat aplikaci do výjimek Windows Defender.

Repozitář s aplikací

Repozitář obsahuje i subrepozitář se zdrojovými soubory dokumentace. Pro získání veškerého obsahu repozitáře je tedy nutné do terminálu napsat příkaz:
`git clone -recurse-submodules https://github.com/Artonix14/dupFot.git`