

Hledání duplikátů pomocí metod morfologické analýzy a analýzy histogramu

Josef Fojtů

Bakalářská práce

2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

ABSTRAKT

Abstrakt česky

Tato práce se zabývá problematikou hledání podobných obrázků označovaných jako duplikáty. Práce se dělí na dvě základní části. Teoretická část obsahuje úvod do základních pojmů rovinné grafiky, dále definuje problém vzniku duplikátů a popisuje konkrétní metody na jejich rozpoznání, obsahuje také kapitolu o morfologické analýze. V praktické části čtenář nalezne porovnání popsaných metod, výsledky testování metod na konkrétních skupinách zkušebních i běžných obrázků. Další součástí praktické části je popis programu Najdi duplikáty. Tento program je charakteristický především funkcí pro hledání duplikátů.

Abstrakt ve světovém jazyce

This work deals with the problem of searching similar images, these are labeled as duplicates. Work is divided in two parts. The first part is theoretical and it contains introduction to elementary terms of plane graphics, defines problem of duplicates creation and describes concrete methods for recognition of duplicates. It contains chapter about morphological analysis too. The second part is practical, the reader finds there comparison of described methods, results of methods testing on concrete groups of testing also real images. The next section of practical part is description of program "Najdi duplikáty". That program is characteristic by function for find duplicates.

Chtěl bych především poděkovat Ing. Tomáši Dulíkovi za velmi dobré odborné vedení a pomoc při řešení problémů této práce. Také bych chtěl poděkovat svým přátelům, kteří mě v této práci podporovali.

OBSAH

ÚVOD.....	6
I. TEORETICKÁ ČÁST	7
1 POJMY POČÍTAČOVÉ ROVINNÉ GRAFIKY	8
1.1 DIGITÁLNÍ OBRAZ.....	8
1.1.1 Vzorkování obrazu	9
1.1.2 Kvantizace obrazu	9
1.2 DALŠÍ DŮLEŽITÉ POJMY	10
1.2.1 Intenzita barvy.....	10
1.2.2 Barevný model RGB	10
1.2.3 Odstíny šedi.....	11
1.2.4 Histogram	11
1.2.5 Histogram složek.....	12
1.2.6 Digitální obraz.....	12
1.2.7 Obrazový bod	13
1.2.8 Obrazová data.....	13
2 PRINCIPY HLEDÁNÍ PODOBNÝCH OBRAZKŮ.....	14
2.1 OBRAZOVÉ DUPLIKÁTY	14
2.1.1 Duplikát.....	14
2.1.2 Přesný duplikát.....	14
2.2 OBRAZOVÉ ÚPRAVY	15
2.3 VLIV OBRAZOVÝCH ÚPRAV NA HISTOGRAM.....	15
2.3.1 Úvod.....	15
2.3.2 Změna jasu	15
2.3.3 Změna kontrastu.....	16
2.3.4 Změna velikosti	17
2.3.5 Změna orientace	18
2.3.6 Změna formátu	18
2.4 METODY HLEDÁNÍ DUPLIKÁTŮ	19
2.4.1 Korelační analýza histogramu	19
2.4.2 Analýza histogramu korelační funkcí	21
2.4.3 Spektrální analýza	24
2.4.4 Problém matoucích histogramů.....	27
2.5 MORFOLOGICKÁ ANALÝZA.....	28
II. PRAKTICKÁ ČÁST	30
3 POROVNÁNÍ METOD	31
3.1 ÚPRAVA KORELAČNÍ ANALÝZY HISTOGRAMŮ	31
3.1.1 Neupravená verze.....	31
3.1.2 Upravená verze.....	32
3.2 TESTOVÁNÍ METOD.....	33
4 PROGRAM NAJDI DUPLIKÁTY	37
4.1 VLASTNOSTI PROGRAMU	37
4.1.1 Hlavní funkce	37
4.1.2 Ovládání programu.....	37

4.1.3	Podporované formáty	39
4.1.4	Systémové nároky	39
4.2	VÝVOJ PROGRAMU	39
4.2.1	Kód aplikace.....	39
4.2.2	Problémy při vývoji.....	40
4.2.3	Průběh vývoje.....	40
4.2.4	Možnosti vylepšení	40
4.3	PODOBNÉ PROGRAMY	41
4.4	FUNKCE HLEDÁNÍ DUPLIKÁTŮ	42
4.4.1	Porovnání dvojice obrázků.....	42
4.4.2	Algoritmus analýzy obsahu adresáře	42
4.4.3	Rychlost.....	45
SEZNAM POUŽITÉ LITERATURY		49
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		50
SEZNAM OBRÁZKŮ		51
SEZNAM TABULEK.....		52
SEZNAM PŘÍLOH.....		53

ÚVOD

Při organizaci obrázků na počítači si vždycky nevystačíme s tím, co nám nabízí prohlížeč obrázků a souborový manažer. Někdy chceme ušetřit místo na disku nebo uspořádat obrázky a proto hledáme takové, které jsou upravenými kopiemi původního obrázku. Právě v takovém případě je dobré použít nástroj, který nám práci usnadní.

V rámci bakalářské práce byl takový nástroj vytvořen. Program nepatří do komerční oblasti a proto může být pomůckou všem, kteří by ho využili.

I. TEORETICKÁ ČÁST

1 POJMY POČÍTAČOVÉ ROVINNÉ GRAFIKY

1.1 Digitální obraz

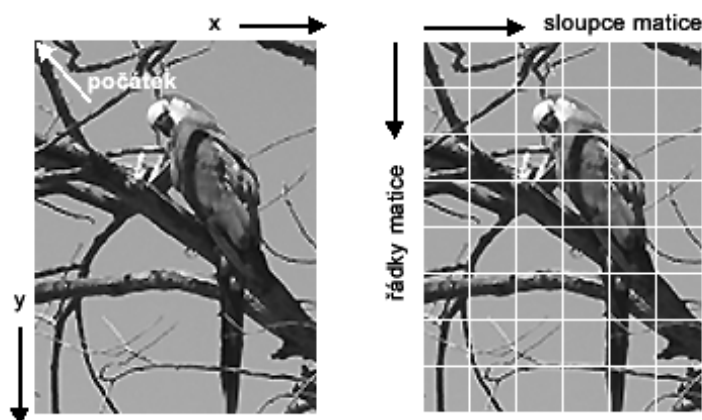
Vzniká digitalizací původního obrazu. Digitalizace je v obrazové technice převod spojité dvojrozměrné obrazové informace do dvojrozměrné obrazové matice. Digitalizace se skládá ze vzorkování (diskretizace v ploše) a následné kvantizace (diskretizace v úrovních). Digitalizace obrazu je nutným předpokladem pro jeho následující digitální zpracování číslicovými počítači (zobrazení, uložení, úpravy apod.).

Proces digitalizace můžeme jednoduše popsat jako zobrazení

$$f(x, y) \xrightarrow{D} g(j, k)$$

kde $x, y \in R$, $j, k \in N$. Funkce $f(x, y)$ je spojitá dvojrozměrná obrazová funkce, která představuje rozložení určité fotometrické veličiny (např. barevné intenzity) po ploše a u které jak funkční hodnoty f , tak souřadnice x, y mohou nabývat libovolných hodnot z oboru reálných čísel. Funkce $g(j, k)$ je obrazová matice, u které jak funkční hodnoty, tak indexy mohou nabývat pouze diskretních hodnot.

Obrazová matice může mít obecně libovolnou geometrii. V praxi se však používá téměř výhradně pravoúhlá matice (obr. 1). Jeden prvek obrazové matice se nazývá obrazový bod. Často se můžeme setkat i s názvem pixel, což je umělé slovo, vytvořené z anglických slov picture element. Poloha obrazového bodu je určena řádkovým indexem j a sloupcovým indexem k .



Obr. 1 – Pravoúhlá obrazová matice

1.1.1 Vzorkování obrazu

Vzorkování obrazu si lze intuitivně představit jako rozdělení obrazu, který chceme vzorkovat na zpravidla čtverce, přičemž část obrazu v jednu čtverci nahradíme jednou reprezentační hodnotou, konstantní pro všechny body čtverce. Je zřejmé, že tím může dojít k omezení rozlišovací schopnosti. Na jednoduchém příkladu si však ukážeme, že při nesprávném vzorkování může dojít nejen k omezení rozlišovací schopnosti, ale i k naprostému znehodnocení obrazu, neboť nevzorkovaný obraz bude obsahovat struktury, které digitální obraz vůbec neobsahoval.

Za podmínky, že vzorkovaný obraz má konečnou velikost (reálné vzorkování) je navzorkovaný obraz dán vztahem

$$f_p(x, y) = f_i(x, y) \cdot s(x, y),$$

kde f_i je vstupní obrazová funkce a $s(x, y)$ vzorkovací funkce. Ta je daná vztahem

$$s(x, y) = \sum_{j=-J}^J \sum_{k=-K}^K p(x - j\Delta x, y - k\Delta y),$$

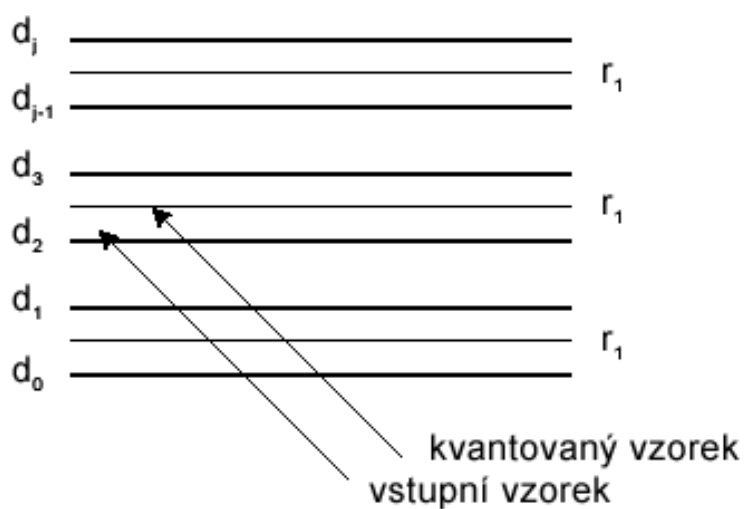
kde funkce p jsou reálné vzorkovací impulsy.

1.1.2 Kvantizace obrazu

Druhou část digitalizace obrazové informace představuje diskretizace parametrů obrazové informace. Tato diskretizace má na rozdíl od plošné diskretizace jednorozměrný charakter a bývá označována jako amplitudové kvantování.

Amplitudovým kvantováním se rozumí proces reprezentace analogových, plošně diskretizovaných vzorků konečným počtem diskrétních úrovní. Tento proces lze popsat takto: amplituda analogového vzorku f je porovnána se sadou rozhodovacích úrovní d a jestliže leží mezi rozhodovacími úrovněmi d_i a d_{i+1} je její hodnota nahrazena rekonstrukční úrovní r_i .

Celý proces je znázorněn na obr. 2. Pro správnou kvantizaci obrazu je nutné stanovit vhodný počet kvantizačních hladin, dále pak rozhodovacích a rekonstrukčních úrovní.



Obr. 2 – Princip kvantizace signálu

1.2 Další důležité pojmy

1.2.1 Intenzita barvy

Intenzita barvy I udává, jakou měrou je tato barva zastoupena. Běžně nabývá hodnot z intervalu

$$I \in < 0, 255 >$$

Pokud je $I = 255$, pak je barva zastoupena maximální možnou měrou. Pokud je intenzita $I = 0$, pak barva není zastoupena vůbec.

Intenzita barvy se označuje jako celková intenzita nebo jas. Lidské oko vnímá body určité barvy v maximální intenzitě jako velmi jasné, naopak body s barvou minimální intenzity jako velmi tmavé.

1.2.2 Barevný model RGB

Každý bod v obraze má nějakou barvu. Na obrazovce počítače je barva bodu určena intenzitami tří barevných složek: červené (red – R), zelené (green – G) a modré (blue – B).

V tomto barevném modelu se bílé barvy dosáhne nastavením barevných složek na maximální intenzitu ($R=255$, $G=255$, $B=255$). Oproti tomu černé barvy se dosáhne nastavením barevných složek na minimální intenzitu ($R=0$, $G=0$, $B=0$).

1.2.3 Odstíny šedi

Jsou tvořeny různou intenzitou černé barvy. Jde o přechod od bílé barvy k černé přes všechny intenzity černé barvy. Obrazový bod již není popsán intenzitami všech tří barevných složek, ale pouze celkovou intenzitou.

V tomto modelu je barva nahrazena odstínem šedi o intenzitě I v rozmezí 0 – 255.

Barevné body lze převést do odstínů šedi. Ztratí se tím informace o barvě, která se již nedá zpětně určit. Převod do odstínů šedi nelze provést pouhým zprůměrováním intenzit barevných složek R , G a B . Lidské oko je na jednotlivé barvy různě citlivé. Nejcitlivěji vnímá barvu zelenou, naopak nejméně citlivé je na barvu modrou.

Převod barvy jednoho bodu na odstín šedi lze provést podle vztahu

$$I = 0.299R + 0.587B + 0.114G,$$

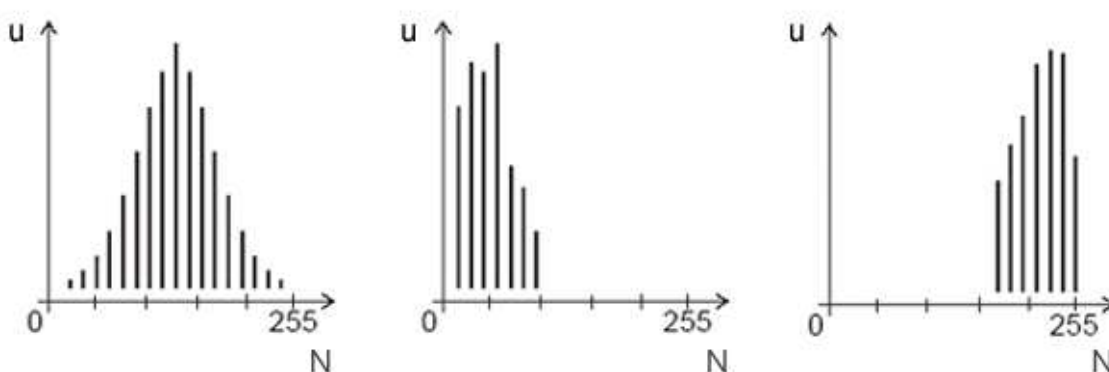
kde R , G , B jsou intenzity těchto složek, I je intenzita vzniklého odstínu šedé.

1.2.4 Histogram

Histogram je definován jako funkce četnosti výskytu určité úrovně. Platí

$$N = N(u),$$

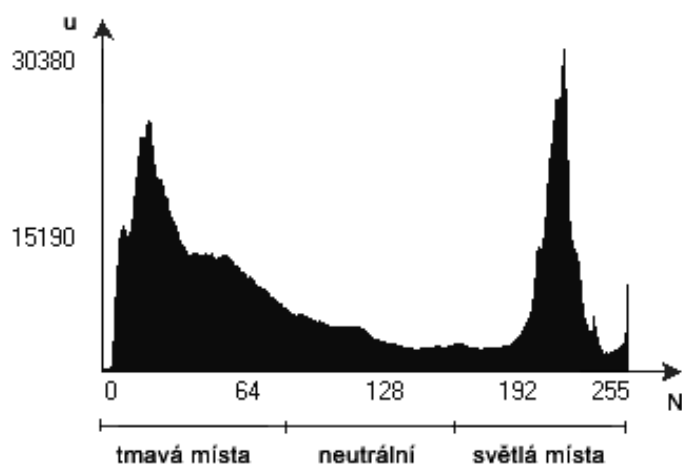
kde u je příslušná úroveň, N četnost výskytu úrovně u . Histogram především udává, jak je využit dynamický rozsah barev a odstínů a zda jsou všechny úrovně obsazeny se stejnou četností.



Obr. 3 - Optimální histogram (vlevo), tmavý obraz, světlý obraz (vpravo)

Pojmem histogram se obvykle rozumí histogram intenzit. Ten je tedy funkcí četnosti výskytu určité celkové intenzity. Určí se převedením barevného obrázku do odstínů šedi a

následným procházením celkových intenzit bodů a počítáním četností výskytu v celém rozsahu.

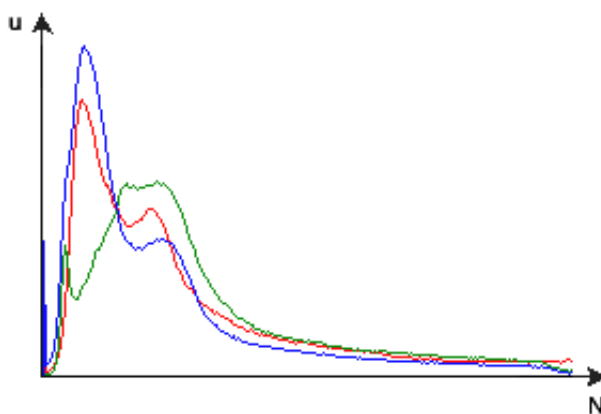


Obr. 4-1 Reálný histogram intenzit

Z křivky histogramu lze prakticky usuzovat o expozici obrázku. Běžně je histogram intenzit využíván jako důležitá kontrola při upravování digitálních fotografií.

1.2.5 Histogram složek

Speciálním případem histogramu je histogram složek. Ten je funkcí četnosti výskytu určité intenzity jedné ze základních barevných složek. Jsou histogramy zelené složky, červené i modré.



Obr. 4-2 Reálný histogram zelené, červené a modré barevné složky

1.2.6 Digitální obraz

Digitální obraz se skládá z obrazových bodů uspořádaných do matice určité šířky a výšky. Rozměry této matice se označují jako rozlišení obrazu.

1.2.7 Obrazový bod

Je nejmenším elementem digitálního obrazu. Každý bod má celočíselné souřadnice a barvu. Souřadnice bodu je v digitální obrazové technice běžně dvousložková – x a y .

Pro barevný model RGB je barva obrazového bodu dána intenzitami základních barevných složek tohoto modelu. Například pro červený obrazový bod je jeho barva dána $R, G, B = 255, 0, 0$.

1.2.8 Obrazová data

Je posloupnost hodnot určujících barvu všech obrazových bodů v obrázku. Obrazová data také nesou informaci o souřadnicích všech bodů, protože jsou logicky uspořádány do pole. Nejsnadněji jsou obrazové data reprezentovány celkovými intenzitami barev jako na obrázku.

56	53	48	49	101
38	22	69	16	36
76	84	196	27	21
22	18	109	14	16
74	27	14	8	22

Obr. 5 - Obrazová data obrázku o velikosti 5x5 obrazových bodů

2 PRINCIPY HLEDÁNÍ PODOBNÝCH OBRAZKŮ

2.1 Obrazové duplikáty

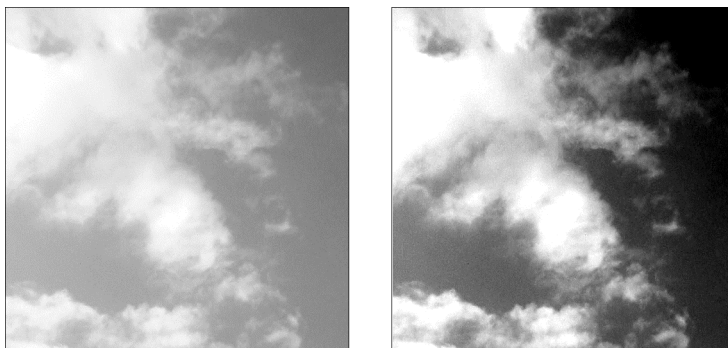
Obrazové duplikáty vznikají kopírováním obrázku nebo úpravou originálu a uložením výsledku do nového souboru.

2.1.1 Duplikát

Mějme dva podobné obrázky, obrázek A a obrázek B. Obrázek B vznikl úpravou obrázku A. Pak platí, že obrázek B je duplikát obrázku A. Obrázek A je původní obrázek, tedy originál.

Duplikát může vzniknout aplikováním pouze jedné konkrétní obrazové úpravy, pak mluvíme o jednoduché úpravě originálu, nebo několika úprav, například změny jasu, kontrastu, velikosti i orientace zároveň. V tomto případě se jedná o kombinovanou úpravu originálu.

V dalším textu se vždy rozumí duplikát jako obrazový duplikát.



Obr. 6 – Originální obrázek (vlevo), jeho duplikát (vpravo)

2.1.2 Přesný duplikát

Mějme dva podobné obrázky, obrázek A a obrázek B. Obrázek B vznikl zkopírováním obrázku A. Pak platí, že obrázek B je přesný duplikát obrázku A. Obrázek A je původní obrázek a tedy originál.

Přesný duplikát se odlišuje od originálu názvem a příponou. Obrazové data přesného duplikátu a originálu jsou naprosto stejná. Přesný duplikát a originál mají často stejnou velikost na disku počítače.

2.2 Obrazové úpravy

Mezi základní obrazové úpravy patří

- změna jasu
- změna kontrastu
- změna velikosti
- změna obrazové orientace
- změna formátu

Tyto úpravy se běžně používají při práci s digitálními fotografiemi na počítači. Uživatel počítače může upravit fotografii a nechat grafický software vytvořit nový soubor s upravenou fotografií. Tímto vznikají duplikáty.

2.3 Vliv obrazových úprav na histogram

2.3.1 Úvod

Duplikát a originál mají do určité míry podobné histogramy. Míra této podobnosti je ovlivněna tím, jakou úpravou původního obrázku duplikát vznikl, jak tato úprava byla výrazná a zda byla provedena jednoduchá nebo kombinovaná úprava.

Platí, že histogram duplikátu vytvořeného jednoduchou úpravou originálu a histogram originálu jsou si velmi podobné. Po kombinované úpravě originálu je histogram duplikátu odlišný výrazněji.

2.3.2 Změna jasu

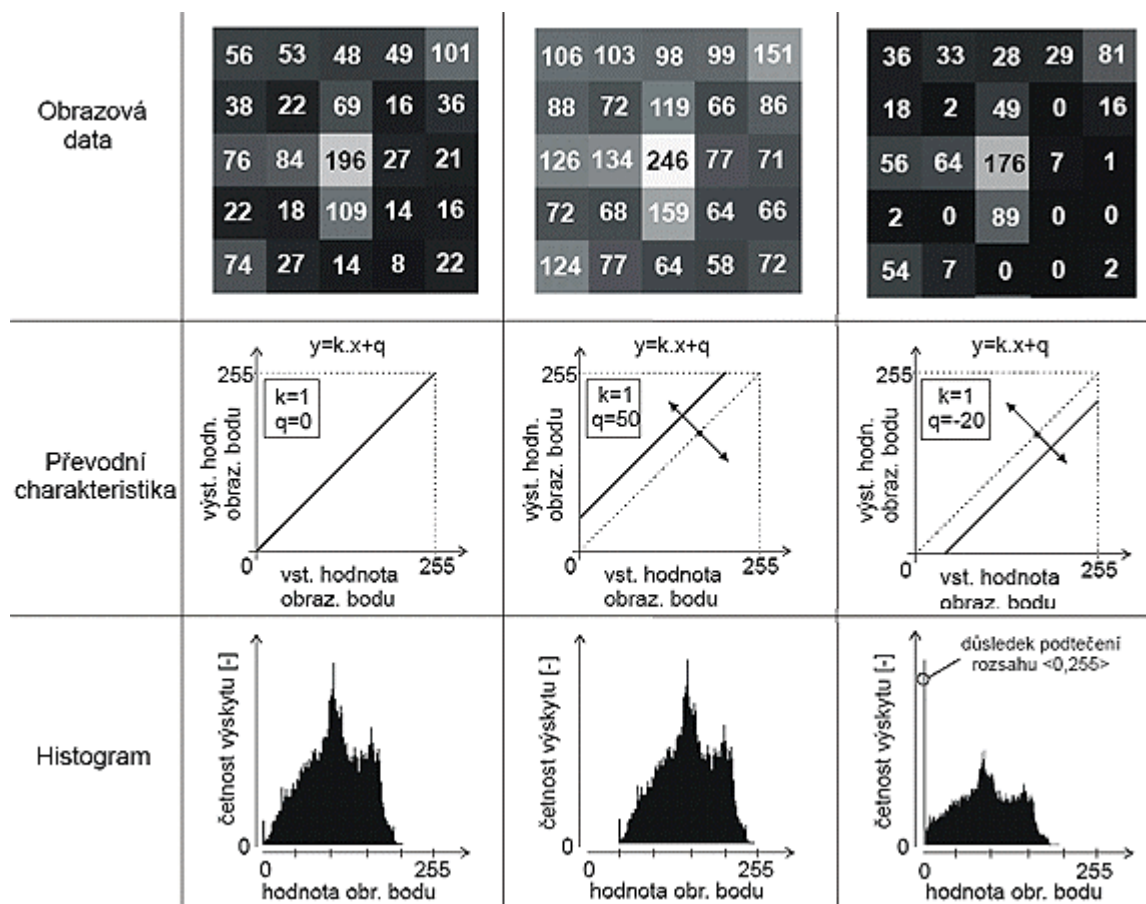
Po změně jasu je obrázek zesvětlen nebo ztmaven. Změnou jasu dojde k plošné změně celkových intenzit bodů obrazu. K původnímu obrazu se přičte nebo odečte konstanta. Pokud je jas změněn, pak celková intenzita pro každý bod bude

$$I_V = I_P + k \text{ (zvýšení jasu),}$$

$$I_V = I_P - k \text{ (snížení jasu),}$$

kde I_P je původní intenzita, I_V výsledná intenzita a k velikost změny.

Křivka histogramu je touto úpravou posunuta ve vodorovném směru. Pokud je jas zvýšen, pak histogram duplikátu je vůči původnímu posunut doprava směrem k bílé barvě. Pro snížení jasu se histogram posouvá doleva.



Obr. 7 – Vliv změny jasu na histogram obrázku

Často se po změně jasu na stranách histogramu kumulují větší množství bodů a tak vznikají špičky. Je to důsledkem toho, že některým bodům se výsledná intenzita dostane mimo meze a takovým je pak přiřazena intenzita minimální ($I_V = 0$) nebo maximální ($I_V = 255$).

2.3.3 Změna kontrastu

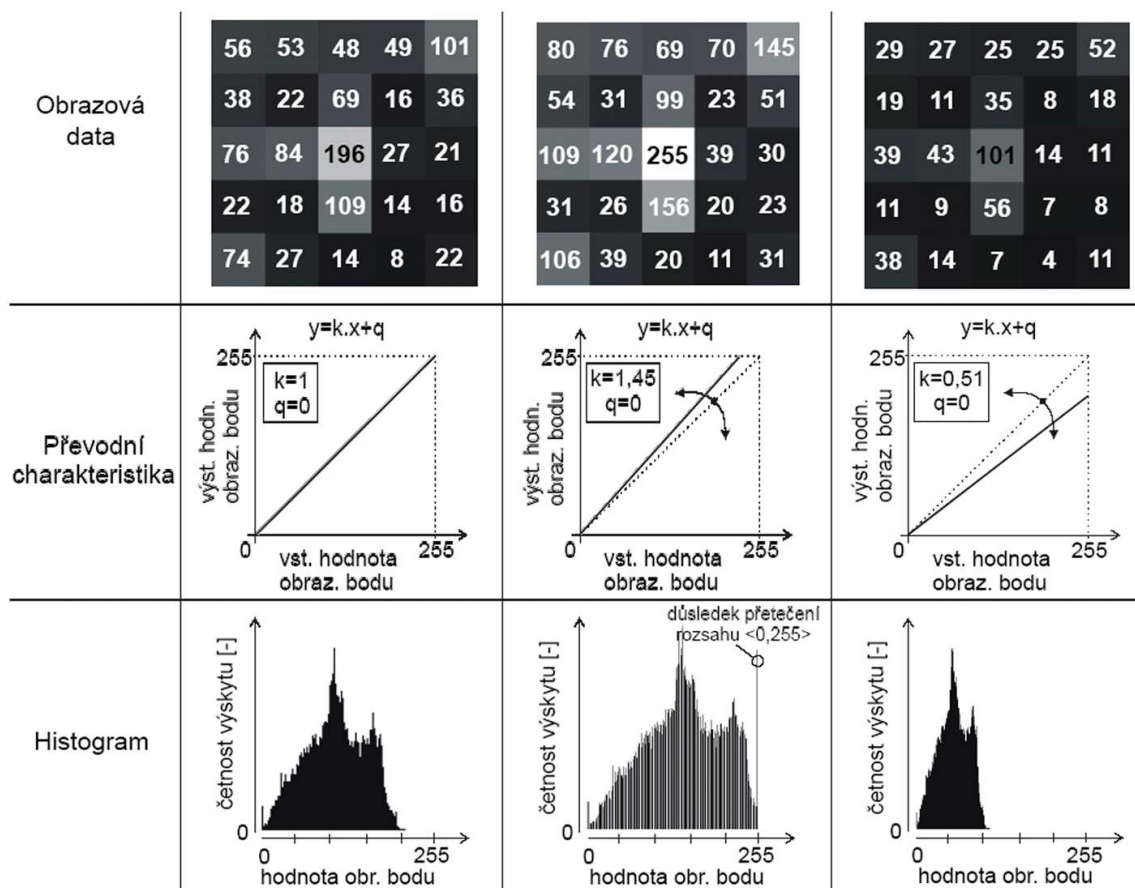
Změnou kontrastu dojde také k plošné změně celkových intenzit bodů obrazu. Původní obraz se násobí nebo dělí konstantou. Pro každý obrazový bod nového obrazu platí

$$I_V = I_P \cdot k \text{ (zvýšení kontrastu),}$$

$$I_V = I_P \div k \text{ (snížení kontrastu),}$$

kde I_P je původní intenzita, I_V výsledná intenzita a k velikost změny.

Křivka histogramu je touto změnou rozšířena nebo zúžena. Histogram duplikátu je méně podobný histogramu originálu než po změně jasu.



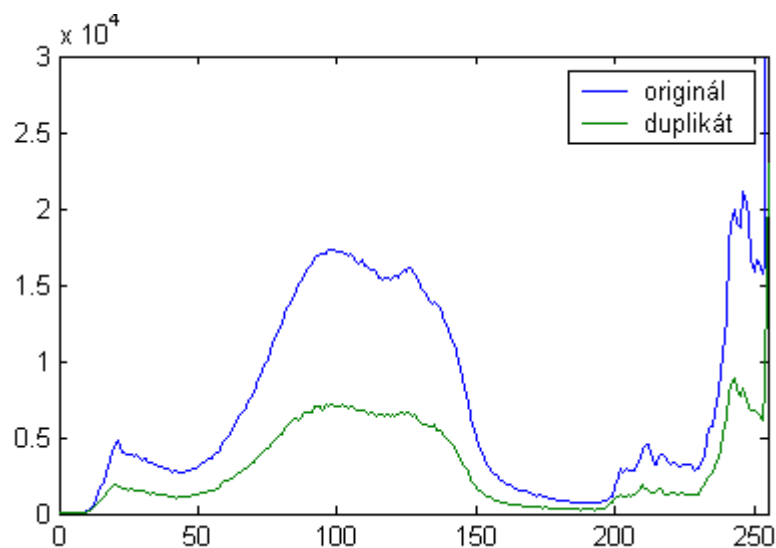
Obr. 8 – Vliv změny kontrastu na histogram obrázku

Po změně kontrastu se také často kumulují po stranách intervalu větší množství bodů.

2.3.4 Změna velikosti

Změnou velikosti dojde ke zvětšení nebo zmenšení rozlišení obrázku. Po změně velikosti originálu je změna obrazových dat minimální. Histogram originálu a duplikátu má pouze rozdílné počty četností.

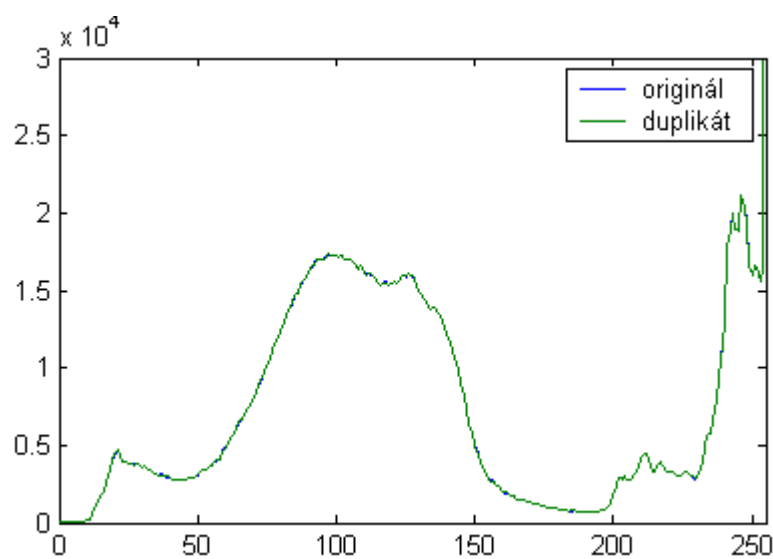
Tyto histogramy mají jinou „výšku“, ale stejný průběh. Jsou silně lineárně závislé a míra této závislosti je ovlivněna pouze kvalitou algoritmu použitého ke změně velikosti původního obrázku. Dnes používané algoritmy jsou velmi kvalitní, míra lineární závislosti je téměř stoprocentní.



Obr. 9 – Vliv změny velikosti obrázku na histogram obrázku

2.3.5 Změna orientace

Při změně orientace dochází k otočení obrazu. Pro skokovou změnu orientace, tedy v násobcích 90° , se histogramy originálu a duplikátu rovnají.



Obr. 10 – Vliv změny orientace na histogram obrázku

2.3.6 Změna formátu

Změnou formátu se rozumí převod obrázku z jednoho formátu do jiného. Tato konverze může být bezztrátová či ztrátová.

Při bezztrátové konverzi (např. z formátu JPEG do formátu BMP) nedojde k žádné změně obrazových dat (za předpokladu zachování barevné hloubky). Histogramy originálu a duplikátu jsou proto stejné.

Při ztrátové konverzi záleží na tom, k jak výrazným úbytkům kvality obrazu dochází. Podle toho se budou lišit i histogramy.

2.4 Metody hledání duplikátů

Při hledání duplikátů lze využít faktu, že obrazové úpravy mění barvu obrazových bodů na celé ploše obrazu. Prakticky to znamená, že plošné charakteristiky obrazu zůstanou do značné míry zachovány, budou pouze posunuty.

Existuje několik metod jak najít podobné obrázky. Liší se svou přesností a náročností výpočtu nutného pro určení podobnosti dvojice obrázků.

2.4.1 Korelační analýza histogramu

Tato metoda využívá podobnosti histogramů originálu a duplikátu. Je založena na vyhodnocení korelačního koeficientu dvou porovnávaných histogramů.

Teoretický úvod

Korelační koeficient pro soubory dat x, y je dán jako

$$\rho_{1,2} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu_x) \cdot (y_i - \mu_y)}{\sigma_x \cdot \sigma_y},$$

kde $-1 \leq \rho_{1,2} \leq 1$, μ_x a μ_y jsou aritmetické průměry souborů dat, σ_x a σ_y jsou jejich směrodatné odchylky.

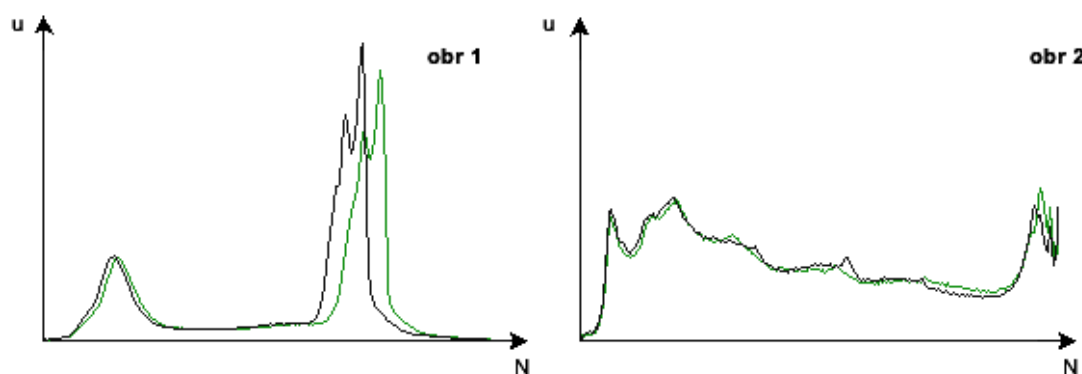
Korelační koeficient je ukazatelem lineární závislosti dvou souborů dat. Čím víc se toto číslo blíží k 1.00, tím silnější závislost. Prakticky lze za lineárně závislé soubory prohlásit už takové, které mají korelační koeficient alespoň 0.65.

Princip metody

Metoda určí histogramy dvojici obrázků a vypočte korelační koeficient levé a pravé poloviny histogramů zvlášť. Kvůli horším výsledkům se neurčuje celkový korelační koeficient (viz. kapitola 3.1).

Korelační koeficienty se po výpočtu upraví. Pokud je některý z nich záporný, tak se nuluje. Podobnost obrázků je stanovena jako průměr těchto upravených koeficientů.

Metoda může pro zrychlení pracovat s histogramem zelené složky, protože histogram intenzit je běžně velmi podobný histogramu zelené složky. Podobnost je patrná z obrázku (obr. 11). Metoda je takto rychlejší, protože čas nutný pro určení histogramu zelené složky je 3x kratší než pro určení histogramu intenzit.



Obr. 11 – Podobnost histogramů intenzit a zelené složky

Hodnocení metody

Metoda korelační analýzy histogramů není pro hledání duplikátů úplně ideální. Úspěšnost této metody ve správném vyhodnocení podobnosti obrázků je asi 60%.

Nevýhodou metody je, že udává malou podobnost originálu a duplikátu vzniklého úpravou kontrastu a jako duplikát často nevyhodnotí obrázek kopie originálu se silně upraveným jasnem. V praxi to však není nepřekonatelným problémem, protože při výskytu více duplikátů jsou správně identifikovány alespoň některé vazby mezi nimi a z nich jsou pak vytvořeny vcelku dobré skupiny.

Největší výhody této metody jsou rychlost, snadná programová realizace a naprostá úspěšnost v odhalování duplikátů vzniklých otočením nebo změnou velikosti originálu. Její další předností je univerzálnost. Může být použita jako dobrý základ a pro lepší výsledky doplněna ještě jinou, zpřesňující metodou.

Praktické příklady analýz pomocí této metody jsou v kapitole 3.2.

2.4.2 Analýza histogramu korelační funkcí

Tato metoda využívá podobnosti histogramů originálu a duplikátu. Metoda je založena na analýze průběhu vzájemné korelační funkce.

Vzájemnou korelační funkci dvou signálů lze získat několika způsoby. V rámci metody se určuje s využitím Fourierovy transformace.

Teoretický úvod

Fourierova transformace převádí původní časový průběh do frekvenční oblasti, proto platí

$$\tilde{f}(A, \omega) = f(t),$$

kde f je původní průběh a $\tilde{f}$ jeho Fourierův obraz, A je amplituda, ω fázová frekvence, t čas.

Vztah pro výpočet spojitě Fourierovy transformace je dán jako

$$\tilde{E}(f) = \int_{-\infty}^{\infty} E(t) \cdot e^{2\pi f t \cdot i} dt,$$

kde $E(t)$ je původní časový průběh, f je frekvence a i značí imaginární jednotku.

Pro diskrétní signály se však používá diskrétní Fourierova transformace. Označuje se jako DTFT. Diskrétní Fourierova transformace je definována jako

$$\tilde{E}[k] = \frac{1}{N} \sum_{n=0}^{N-1} E[x] \cdot e^{\frac{-2\pi k i}{N}},$$

kde N je počet vzorků a $k \in C$.

Výpočet této transformace běžným způsobem je velmi výpočetně náročné. Proto byl vyvinut optimalizovaný algoritmus označovaný jako Fast Fourier Transformation (FFT). Tento se používá pro přímou transformaci. Pro výpočet zpětné transformace se používá inverzní Fourierova transformace označována Inverse Fast Fourier Transformation (IFFT).

Vzájemná korelační funkce vyjadřuje určitou podobnost dvou signálů. Zkoumá míru podobnosti dvou signálů v závislosti na jejich posunutí. Z této funkce lze kromě podobnosti signálů také určit, o jakou vzdálenost je jeden signál posunutý vůči druhému v případě, že posunuté jsou.

Vzájemná korelační funkce je definována jako

$$\tilde{R}_{xy}(f) = \tilde{x}^*(f) \cdot \tilde{y}(f),$$

kde $\tilde{x}^*(f)$ je komplexně sdružená funkce k Fourierově transformaci prvního časového průběhu, $\tilde{y}(f)$ je Fourierova transformace druhého časového průběhu.

Vzájemnou korelační funkci můžeme převést z frekvenční do časové oblasti. Tento převod se provádí zpětnou Fourierovou transformací funkce $\tilde{R}_{xy}(f)$.

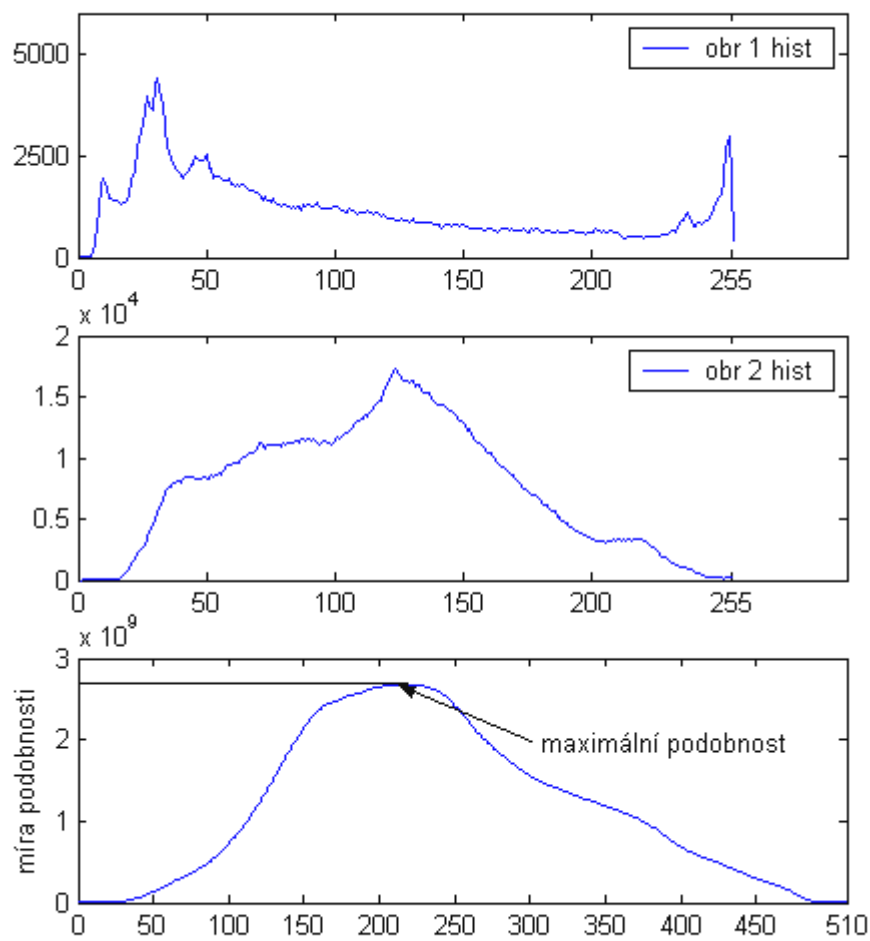
Výsledná funkce již podává informaci o podobnosti srovnávaných signálů, ale pro metodu vyhodnocení podobnosti obrázků se ještě znormuje. Normovaná vzájemná korelační funkce pak nabývá hodnot z intervalu $\langle -1, 1 \rangle$. Maximum této funkce udává, jak jsou zkoumané signály podobné.

Princip metody

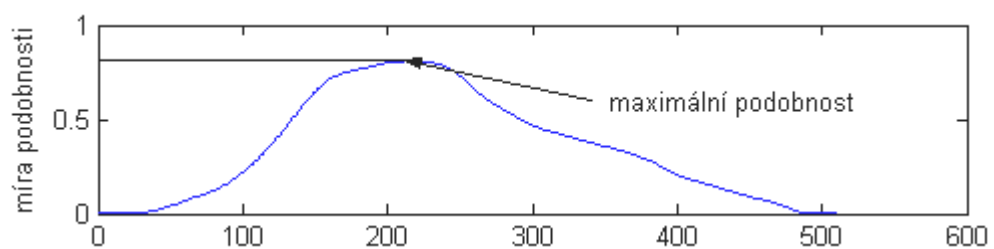
K vyhodnocení podobnosti obrázků je určena vzájemná korelační funkce jejich histogramů. Funkce je následně znormována. Pokud jsou srovnávané obrázky podobné, pak má tato funkce maximum blízké číslu 1.00 (obr. 13).

Pro stanovení míry podobnosti dvou obrázků se zjistí velikost maxima normované funkce. Pokud maximum leží v nastavené toleranci, pak jeden obrázek je prohlášen za duplikát toho druhého. Nastavená tolerance určuje, do jaké míry podobnosti ještě bude obrázek považován za duplikát.

Příklady testování této metody jsou uvedeny v kapitole 3.2.



Obr. 12 – Histogramy porovnávaných obrázků a jejich vzájemná korelační funkce



Obr. 13 – Normovaná vzájemná korelační funkce z Obr. 12

Porovnávané histogramy (obr. 12 nahoře a uprostřed) nejsou příliš podobné. Normovaná vzájemná korelační funkce (obr. 13) má však maximum kolem hodnoty 0.8. Z něho se dá usuzovat, že by se mohlo jednat o podobné obrázky.

Hodnocení metody

Zdá se, že tato metoda je v úspěšnosti o něco lepší než korelační analýza histogramu. Má lepší výsledky při odhalování duplikátů vzniklých úpravou kontrastu, také velmi dobře vyhodnocuje duplikáty vzniklé změnou orientace nebo rozlišení originálu. Úspěšnost této metody může být až 70%.

Z testování metody je patrné, že koeficient podobnosti je pro všechny porovnávané obrázky vysoký. Mezi jeho velikostí pro podobné i nepodobné obrázky nebývá dostatečně velký rozdíl. Při praktickém použití této metody to může být problém, metoda tak má menší úspěšnost ve správném vyhodnocování porovnávaných obrázků. Často považuje za duplikáty i obrázky, které jimi ve skutečnosti nejsou. Praktické příklady v kapitole 3.2.

2.4.3 Spektrální analýza

Stanovení podobnosti dvou obrázků pomocí spektrální analýzy je další metodou, kterou je možné využít při porovnávání obrázků. Tato metoda je založena na analýze spekter obrázků. Nepracuje s histogramy obrázků.

Teoretický úvod

Spektrum obrazu udává prakticky informaci o tom, jakou měrou jsou v obraze zastoupeny větší plochy stejné nebo blízké barvy a místa s velmi rychle se měnící barvou. Pro analýzu obrázků hledáme dvojrozměrné obrazové spektrum. Obrazové spektrum je vhodné určit výpočtem dvojrozměrné diskrétní Fourierovy transformace. Ta je definována jako

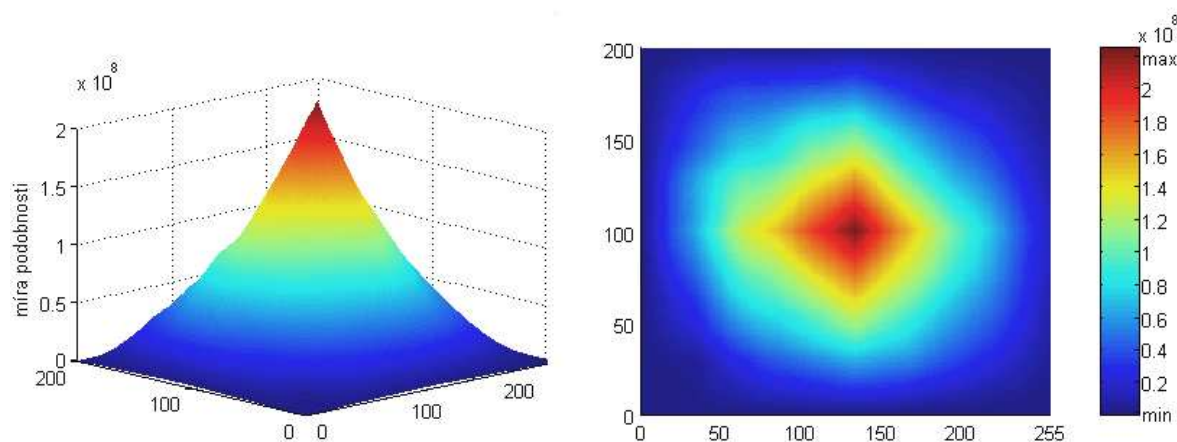
$$\tilde{E}[u, v] = \frac{1}{N_1 N_2} \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} E[x, y] \cdot e^{-2\pi i \left(\frac{xu}{N_1} + \frac{yv}{N_2} \right)},$$

kde N_1, N_2 je počet vzorků, $u, v \in C$.

Dvojrozměrná vzájemná korelační funkce se používá pro určení podobnosti funkcí dvou proměnných. Pro stanovení této funkce se získané spektrum obrázků převede z komplexních hodnot na reálné vypočtením absolutní hodnoty pro všechny body spektra.

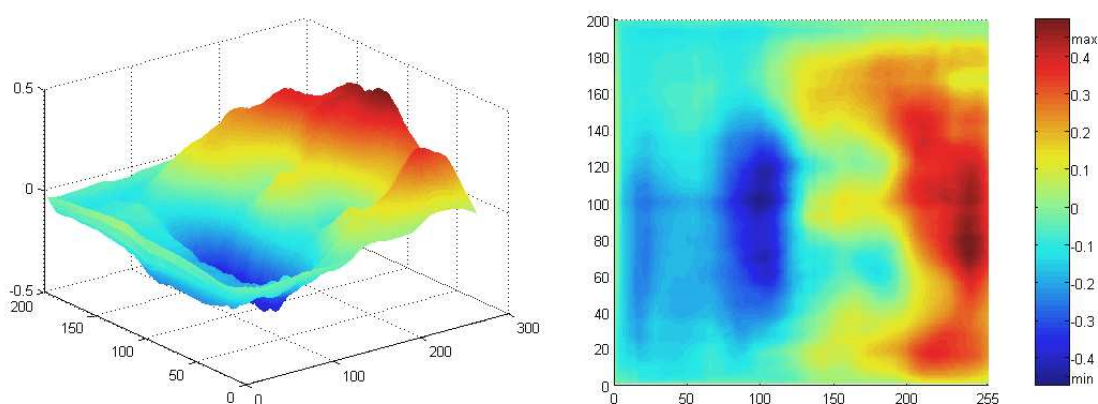
Pro takto upravená spektra určíme dvojrozměrnou diskrétní vzájemnou korelační funkci. Ta je definována obdobně jako obyčejná vzájemná korelační funkce s tím rozdílem, že pracuje s dvojrozměrnými Fourierovými transformacemi.

Výsledkem je průběh ve frekvenční oblasti, proto se funkce převede zpět do diskrétní oblasti provedením zpětné dvojrozměrné Fourierovy transformace.



Obr. 14 – Dvourozměrná vzájemná korelační funkce srovnávaných obrázků

Po znormování získané funkce určíme její maximum (na obr. 15 tmavě červenou barvou). Jeho velikost udává zda jsou průběhy podobné nebo nikoliv.



Obr. 15 – Dvourozměrná normovaná vzájemná korelační funkce

Na obrázku 15 je maximum pod hodnotou 0.5. Dá se usoudit, že porovnávané obrázky nebyly podobné.

Princip metody

Proces srovnávání dvojice obrázků probíhá v několika krocích. Srovnávaným obrázkům se vypočte dvourozměrná normovaná vzájemná korelační funkce. Součástí tohoto výpočtu je dvourozměrná Fourierova transformace srovnávaných obrázků. Transformace je časově velmi náročná, proto se provádí pouze s „profily“ obrázků. Profil je velmi výrazně

zmenšená kopie původního obrázku se zachovaným poměrem stran. Výška profilu by měla být co nejmenší, v rozmezí asi 100-200 obrazových bodů.

Podobnost srovnávaných obrázků se určí nalezením maxima normované vzájemné korelační funkce jejich profilů. Pokud se blíží maximum k číslu 1.00, pak jeden obrázek je duplikátem toho druhého.

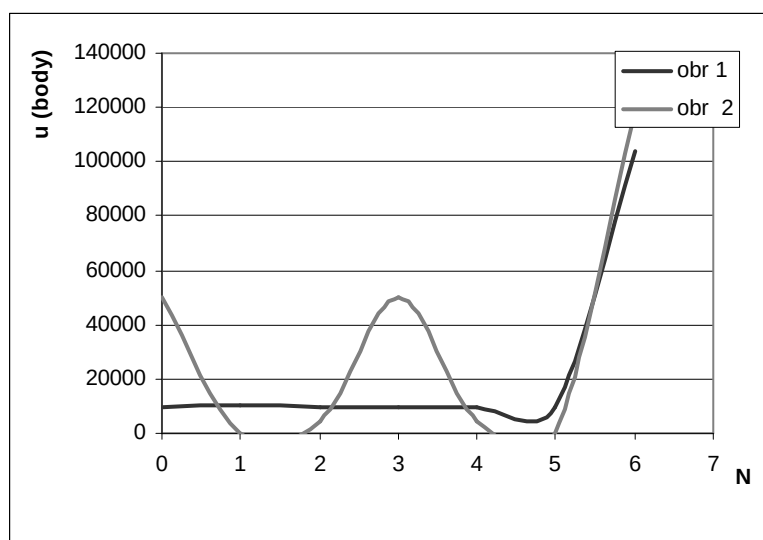
Hodnocení

Tato metoda pracuje velmi přesně. Její úspěšnost je z výsledků testů stoprocentní (viz. kapitola 3.2 – testování metod). Je proto na hledání duplikátů velice vhodná, je také velmi rychlá. Nevýhody této metody jsou pouze ve složitosti programové realizace.

2.4.4 Problém matoucích histogramů

Některé průběhy histogramů způsobují problémy u metod, které stanovují korelační podobnost. Srovnávané obrázky mohou mít histogramy s velmi podobnými špičkami. Pokud jsou špičky dostatečně velké, tak korelační analýza vyhodnotí tyto histogramy jako podobné, i když mají průběh velmi odlišný. Označme tyto histogramy jako „matoucí“.

V příkladu mějme dva nepodobné soubory dat se špičkou v podobné výšce. Provede se dvakrát výpočet korelačního koeficientu. Při prvním výpočtu špička nebyla zahrnuta. Jak je vidět (tab. 1 vpravo), že korelační koeficient správně nevypovídá o lineární závislosti. Pokud se však do výpočtu zahrne i špička, pak korelační koeficient najednou vypovídá o silné lineární závislosti (tab. 1 vpravo). Díky špičce přestal korelační koeficient prakticky vypovídat o celkové podobnosti srovnávaných průběhů.



graf 1 - Průběh porovnávaných dat

N	u1(body)	u2(body)
0	9909	50000
1	9990	0
2	9859	4300
3	9711	50000
4	9636	4531
5	9884	0
6	104224	118673

výpočet	$\rho_{x,y}$
1	-0,178
2	0,859

Tabulka 1 – Data srovnávaných průběhů (vlevo), vypočtené korelační koeficienty

2.5 Morfologická analýza

Morfologická analýza nehledá obrazové duplikáty. Tato metoda umožňuje hledat obrázky, které obsahují tvar zastoupený ve vzorovém obrázku. Vstupem analýzy je vzorový obrázek a seznam obrázků, který se má prohledat. Výstupem analýzy je seznam obrázků, které obsahují daný tvar.

Úvod

Matematická morfologie je široké odvětví, které se zabývá identifikováním a popisem tvarů. V oblasti digitálního zpracování obrazů jde o sadu nástrojů k předzpracování a segmentaci obrazů. Samostatnou kapitolou morfologie je analýza obrysů.

Analýza obrysů

Analýza obrysů zkoumá v obraze série bodů podél zkoumaných objektů. Tyto body mohou být uspořádány pravidelně nebo nepravidelně. K věrnému zachycení studované křivky (obrysu) je však většinou nutné použít velké množství bodů, což z hlediska dalšího matematického zpracování není příliš praktické. Metody používané k analýze obrysů proto většinou získaným hodnotám bodů přiřazují určitou funkci menšího počtu popisných proměnných, které lze použít pro další morfologickou analýzu.

Jednou z metod morfologické analýzy obrysů je **analýza charakteristického tvaru**. Tato metoda přímo využívá některé vybrané popisné proměnné, které určitým způsobem charakterizují studovaný objekt. Takovými proměnnými jsou například úhly tečen k povrchu křivky.

Hledání obrázků podle vzoru

Metoda hledání obrázků podle vzoru není snadná záležitost. V rámci této práce jsem se zabýval pouze studiem morfologické analýzy a několika experimenty. Sestavení a implementace metody do programu nebylo z časových důvodů možné. Proto je následující popis metody pouze hypotézou.

Metoda hledání obrázků podle vzoru by využívala morfologické analýzy charakteristického tvaru k nalezení takových obrázků, ve kterých se vyskytuje tvar stejný jako je na vzorovém obrázku. Takovým tvarem může být například obličej člověka.

Vzorový obrázek by musel splňovat určitá kritéria. Musel by mít výrazný pouze jeden tvar umístěný na jednobarevném pozadí (nejlépe bílém). Dále by musel mít malé rozlišení, aby analýza byla provedena v přijatelném čase.

Srovnávaný obrázek by byl nejprve předzpracován. Do předzpracování by patřila např. změna velikosti tohoto obrázku na velikost vzoru, úprava jasů obrazu, odstranění rušivých tvarů na okraji obrázku. Po předzpracování by metoda pokračovala určením charakteristického tvaru. Výsledkem by byly hodnoty popisující identifikovaný tvar. Ty by se srovnaly s hodnotami popisující tvar ve vzoru a zjistila by se míra shody. Pokud by byla v tolerančním intervalu, pak by obrázek byl zařazen do seznamu nalezených obrázků.

Hodnocení

Správnost výsledků této metody by se odvíjela od charakteru srovnávaných obrázků. Pokud by na srovnávaném obrázku bylo rušivé pozadí neodstraněné při předzpracování, pak by ani přes obsah vzorového tvaru nebylo možné vyhodnotit obrázek jako podobný. Rušivé pozadí (například další výrazný tvar jako je dům nebo jiné obličejy) by znemožnilo provést správnou identifikaci charakteristického tvaru.

Rychlost analýzy by pro rozsáhlý počet srovnávaných obrázků nebyla velká. Záleželo by na velikosti vzoru a na optimalizaci analýzy.

II. PRAKTICKÁ ČÁST

3 POROVNÁNÍ METOD

V této kapitole jsou uvedeny výsledky analýz při použití metod na testovací obrázky i běžné digitální fotografie. Simulace byly prováděny v prostředí Matlab.

3.1 Úprava korelační analýzy histogramů

3.1.1 Neupravená verze

zadání:

- ❑ původní obrázek (originál) - dup j1 k1.jpg,
- ❑ celkem 11 obrázků ke srovnání, kde většina vznikla úpravou jasu nebo kontrastu originálu,
- ❑ 3 obrázky zcela odlišné od originálu,

Pro dané zadání byla provedena analýza neupravenou verzí korelační analýzy histogramů. Výsledky analýzy jsou v tabulce.

obrázek 1	obrázek 2	$\rho_{x,y}$	P
dup j1 k1.jpg	dup j0-8 k1.jpg	0,57	ne
	dup j1-2 k1.jpg	0,58	ne
	dup j0-6 k1.jpg	0,19	ne
	dup j1-4 k1.jpg	0,28	ne
	dup j1 k20b.jpg	0,25	ne
	dup j1 k20w.jpg	0,91	ano
	dup j1 k40b.jpg	-0,06	ne
	dup j1 k40w.jpg	0,76	ano
	vrstvy-2.jpg	-0,42	ne
	bitmap11.bmp	0,03	ne
	bitmap13.bmp	-0,15	ne
tolerance		30%	
úspěšnost		45%	

Tabulka 2 – Testování neupravené verze korelační analýzy histogramů

Celkový korelační koeficient vypovídá o podobnosti srovnávaných obrázků jen při menších obrazových úpravách originálu. Podobnost je vyhodnocena při změnách jasu do 20%. Pro silnější změny jasu již nevypovídá o žádné podobnosti. Podobně je tomu v případě zmenšení kontrastu o 20-40%, tedy žádná podobnost (v tab. 2 červeně zvýrazněno). Z tohoto důvodu se korelační koeficienty stanovují pro každou polovinu histogramu zvlášť. Tím vzniká upravená verze korelační analýzy histogramů.

3.1.2 Upravená verze

Pro stejné zadání byla provedena upravená verze analýzy. Výsledky jsou v tabulce.

obrázek 1	obrázek 2	$\rho_{x,y1}$	$\rho_{x,y2}$	průměr ¹	P
dup j1 k1.jpg	dup j0-8 k1.jpg	0,31	0,79	0,55	ne
	dup j1-2 k1.jpg	0,38	0,79	0,58	ne
	dup j0-6 k1.jpg	-0,13	0,61	0,31	ne
	dup j1-4 k1.jpg	0,08	0,75	0,42	ne
	dup j1 k20b.jpg	-0,09	0,85	0,43	ne
	dup j1 k20w.jpg	0,92	0,65	0,78	ano
	dup j1 k40b.jpg	-0,22	0,78	0,39	ne
	dup j1 k40w.jpg	0,80	0,62	0,71	ano
	vrstvy-2.jpg	-0,45	-0,33	0	ne
	bitmap11.bmp	-0,26	0,16	0,08	ne
	bitmap13.bmp	0,18	-0,32	0,09	ne
tolerance		30%			
úspěšnost		45%			

Tabulka 3 – Testování upravené verze korelační analýzy histogramů

¹ – průměr neuvažuje záporné hodnoty koeficientů $\rho_{1x,y}$, $\rho_{2x,y}$

Úspěšnost metody po úpravě je stále stejná. Výhoda této metody je ale v tom, že míra podobnosti je pro duplikáty vzniklé výraznou změnou originálu vyšší (v tab. 3 modře zvýrazněno) a to při zachování velkého „odstupu“ od míry podobnosti odlišných obrázků. Tato změna podstatně zlepšuje úspěšnost metody v programové realizaci.

3.2 Testování metod

Př. 1:

- původní obrázek (originál) - dup.jpg (výrazné tvary v obrázku, žádná bílá plocha),
- celkem 11 obrázků ke srovnání, z toho 8 podobných originálu, které vznikly úpravou jasu nebo kontrastu originálu, 3 obrázky odlišné,

		KorAnHist				AnHistKF		SpektAn	
obr. 1	obr. 2	$\rho_{1x,y}$	$\rho_{2x,y}$	průměr*	P	max _{1D}	P	max _{2D}	P
dup.jpg	dup j0-8 k1.jpg	0,31	0,79	0,55	ne	0,96	ano	0,997	ano
	dup j1-2 k1.jpg	0,38	0,79	0,58	ne	0,97	ano	0,998	ano
	dup j0-6 k1.jpg	-0,13	0,61	0,31	ne	0,86	ano	0,986	ano
	dup j1-4 k1.jpg	0,08	0,75	0,42	ne	0,94	ano	0,995	ano
	dup j1 k20b.jpg	-0,09	0,85	0,43	ne	0,87	ano	0,998	ano
	dup j1 k20w.jpg	0,92	0,65	0,78	ano	0,98	ano	0,998	ano
	dup j1 k40b.jpg	-0,22	0,78	0,39	ne	0,48	ne	0,994	ano
	dup j1 k40w.jpg	0,80	0,62	0,71	ano	0,95	ano	0,994	ano
	vrstvy-2.jpg	-0,45	-0,33	0	ne	0,64	ne	0,652	ne
	bitmap11.bmp	-0,26	0,16	0,08	ne	0,83	ne	0,684	ne
	bitmap13.bmp	0,18	-0,32	0,09	ne	0,60	ne	0,684	ne
tolerance		35%				15%		15%	
úspěšnost		45%				91%		100%	

Tabulka 4 – První testování metod na souboru převážně podobných obrázků

* – průměr neuvažuje záporné hodnoty koeficientů $\rho_{1x,y}$, $\rho_{2x,y}$

Korelační analýza histogramu má pro tyto obrázky nejmenší úspěšnost. Při jejím praktickém použití však velmi záleží na konkrétním obsahu adresářů s obrázky, běžně je její úspěšnost vyšší.

Př. 2:

- původní obrázek original_2.jpg (málo výrazné tvary a velká bílá plocha v obrázku),
- celkem 11 obrázků ke srovnání, z toho 8 podobných originálu, které vznikly úpravou jasu nebo kontrastu originálu, 3 obrázky odlišné,

		KorAnHist				AnHistKF		SpektAn	
obr. 1	obr. 2	$\rho_{1x,y}$	$\rho_{2x,y}$	průměr ¹	P	max	P	max	P
dup.jpg	dup j0-8 k1.jpg	-0,18	-0,05	0,00	ne	0,39	ne	0,990	ano
	dup j1-2 k1.jpg	0,06	-0,13	0,03	ne	0,79	ne	0,999	ano
	dup j0-6 k1.jpg	-0,22	-0,03	0,00	ne	0,28	ne	0,941	ano
	dup j1-4 k1.jpg	-0,15	-0,07	0,00	ne	0,37	ne	0,975	ano
	dup j1 k20b.jpg	0,50	0,35	0,42	ne	0,92	ano	0,998	ano
	dup j1 k20w.jpg	0,37	0,27	0,32	ne	0,89	ano	0,999	ano
	dup j1 k40b.jpg	0,04	-0,18	0,02	ne	0,88	ano	0,994	ano
	dup j1 k40w.jpg	-0,27	-0,23	0,00	ne	0,83	ne	0,997	ano
	vrstvy-2.jpg	-0,10	-0,24	0,00	ne	0,71	ne	0,672	ne
	bitmap11.bmp	-0,14	-0,04	0,00	ne	0,83	ne	0,754	ne
	160_6073.jpg	-0,12	0,09	0,00	ne	0,60	ne	0,725	ne
tolerance		35%				15%		15%	
úspěšnost		27%				55%		100%	

Tabulka 5 – Druhé testování metod na souboru převážně podobných obrázků

Pro metody založené na podobnosti histogramů byl originální obrázek jeden z nejhorších. Obsahuje velké plochy stejné barvy a ty se úpravami jasu a kontrastu spojily. Proto tady tyto metody mají tak malou úspěšnost. Analýza histogramu pomocí vzájemné korelační funkce je na tom však stále lépe než běžná korelační analýza.

Př. 3:

- ❑ původní obrázek original_3.jpg,
- ❑ celkem 3 obrázky ke srovnání, z toho všechny podobné, vznikly úpravou orientace a velikosti originálu,

		KorAnHist				AnHistKF		SpektAn	
obr. 1	obr. 2	$\rho_{1x,y}$	$\rho_{2x,y}$	průměr ¹	P	max	P	max	P
dup.jpg	dup o-90.jpg	1,00	1,00	1,00	ano	1,00	ano	-	-
	dup v0-6.jpg	1,00	1,00	1,00	ano	1,00	ano	1,00	ano
	dup v1-4.jpg	1,00	1,00	1,00	ano	1,00	ano	1,00	ano
tolerance		35%				15%		15%	
úspěšnost		100%				100%		66%	

Tabulka 6 – Testování metod na souboru otočených a zmenšených obrázků

Metoda spektrální analýzy nebyla použita pro první obrázek, protože není schopná nalézt duplikát vzniklý otočením originálu. V tomto případě je jasně vidět výhoda metod založených na analýze histogramu. Jejich úspěšnost je stoprocentní a navíc žádná jiná metoda nedovede najít duplikát vzniklý otočením originálu.

Př. 4:

- ❑ původní obrázek original_4.jpg,
- ❑ celkem 2 obrázky ke srovnání, oba jsou velmi odlišné od originálu,

		KorAnHist				AnHistKF		SpektAn	
obr. 1	obr. 2	$\rho_{1x,y}$	$\rho_{2x,y}$	průměr ¹	P	max	P	max	P
dup.jpg	160_6071.jpg	0,84	0,77	0,80	ano	0,90	ano	0,378	ne
	160_6073.jpg	0,77	0,85	0,81	ano	0,91	ano	0,401	ne
tolerance		35%				15%		15%	
úspěšnost		0%				0%		100%	

Tabulka 7 – Testování metod na souboru zcela odlišných obrázků

Metody založené na analýze histogramu tady selhaly díky „matoucím“ histogramům. Nelze se už nezmínit o naprosto vynikající úspěšnosti metody hledání duplikátů založené na spektrální analýze.

Př. 5:

Testování metod na vybraném souboru obyčejných fotografií z digitálního fotoaparátu. Mezi fotografiemi není žádná podobnost. Metody byly aplikovány opakovaně pro stanovení všech podobností, tedy metodou srovnávání každého s každým.

Zadání:

- ❑ dvě sady obrázků (každá obsahuje 6 obrázků),
- ❑ sada 1 obsahuje převážně obrázky s „matoucími“ histogramy,

metoda	porovnávání	tolerance (%)	špatně	chyba (%)
KorAnHist	sada 1	35	2	33
		20	2	33
	sada 2	35	1	17
		30	1	17
AnHistKF	sada 1	30	6	100
		25	0	0
	sada 2	30	0	0
		25	0	0
SpektAn	sada 1	10	0	0
		5	0	0
	sada 2	10	0	0
		5	0	0

Tabulka 8 – Testování metod při porovnávání obrázků stylem každý s každým

Korelační analýza histogramu v tomto testu obstála dobře. Pro zkoumané sady obrázků není její chybovost nejvyšší. To dokazuje, že metoda je použitelná. Metoda analýzy histogramů pomocí korelační funkce si neporadila se sadou 1. Na matoucí histogramy je náchylná ještě víc než obyčejná korelační analýza. Metoda využívající spektrální analýzy obrázků se jednoznačně projevila jako nejlepší.

4 PROGRAM NAJDI DUPLIKÁTY

4.1 Vlastnosti programu

4.1.1 Hlavní funkce

Hlavní funkcí programu Najdi duplikáty je pomoc při organizaci adresářů s digitálními fotografiemi. S jeho pomocí lze hledat duplikáty v zadaném adresáři či adresářích. Po vyhodnocení obsahu výběru je uživateli nabídnut seznam skupin podobných obrázků.

Najdi duplikáty je aplikace jednoduchá a proto vhodná pro každého uživatele.

4.1.2 Ovládání programu

Najdi duplikáty má velice jednoduché ovládání. Po spuštění programu se zobrazí jeho hlavní okno. Na levé straně tohoto okna je adresářový strom, na straně pravé se zobrazuje seznam obrázků v daném adresáři nebo seznam nalezených duplikátů. Hlavní okno má dále plochu pro zobrazování náhledu obrázků, nástrojovou lištu a jednoduché menu.

prvek	příkaz	akce
adresářový strom	levá myš (klepnutí)	výběr jednoho adresáře (podržení klávesy CTRL, výběr několika adresářů)
	levá myš (poklepání)	zobrazení všech podporovaných obrázků v adresáři (pokud jde o adresář, který už nemá žádné podadresáře)
seznam	levá myš (klepnutí)	výběr obrázku, zobrazí se náhled v levém dolním rohu okna (pokud je držena klávesa CTRL, pak lze vybrat několik obrázků)
	klávesa DELETE	odstranění vybraných obrázků
	šipka nahoru, dolů	pohyb v seznamu

Tabulka 9 – Ovládání prvků hlavního okna

Nástrojová lišta

hledat identické obrázky – hledá přesné duplikáty (v současné verzi programu ještě neimplementováno)

hledat podobné obrázky – hledá podobné obrázky v adresáři nebo několika adresářích (maximálně však ze 500 souborů)

odstranit vybrané – smaže vybrané obrázky (nepoužívá Koš)

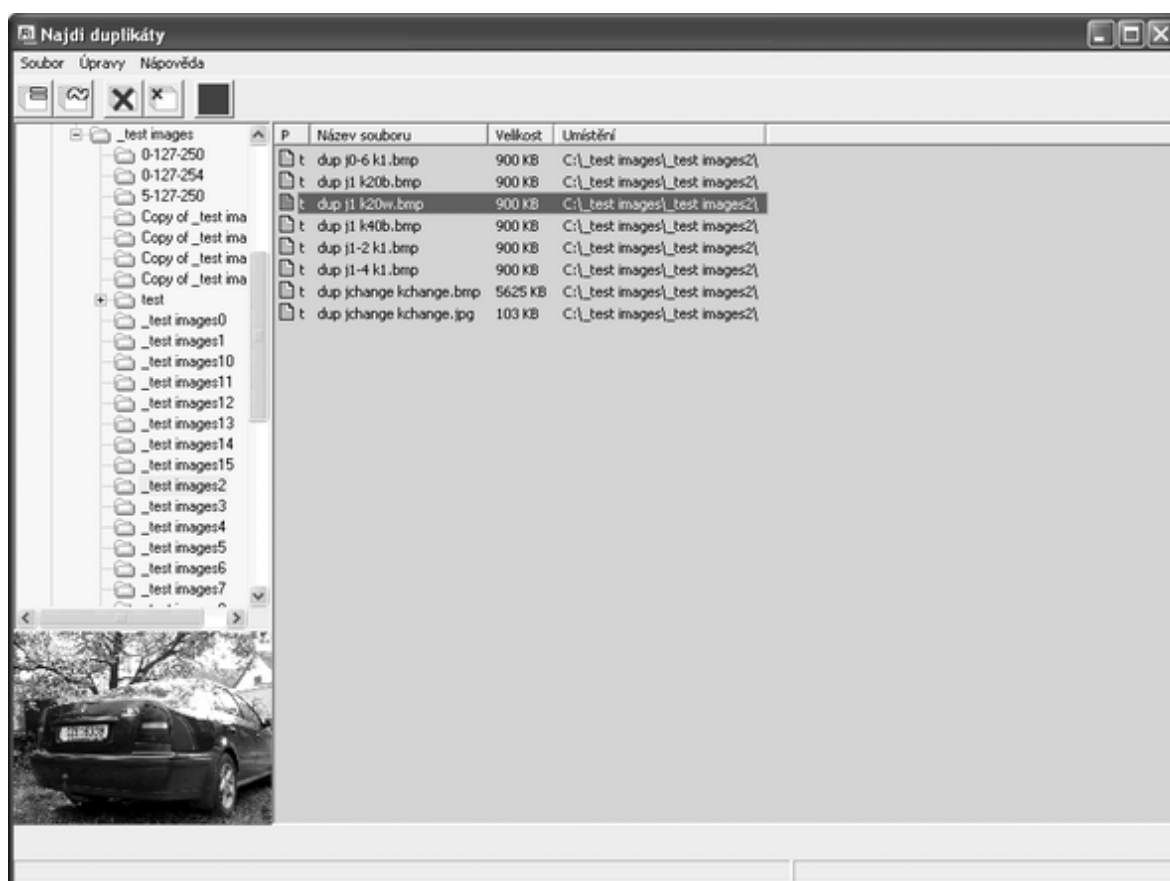
Menu aplikace

soubor: konec – zavře aplikaci

úpravy: nastavení – nastavení programu

- používaná složka – výběr barevné složky, pro kterou program vytváří v paměti histogram (při hledání duplikátů). Je doporučeno ponechat nastavení zelené nebo nastavit modrou složku.
- přesnost popisu obrázků – určuje, jak přesně se histogram obrázku počítá (100% - ze všech bodů, 90% - z poloviny bodů, 80% - ze třetiny bodů)

nápověda: o aplikaci – zobrazí dialog s autory a verzí programu



Obr. 16 – Hlavní okno programu Najdi duplikáty

4.1.3 Podporované formáty

Typy obrázků, se kterými program pracuje, se odvíjí od možností knihovny wxWidgets. Současná verze programu pracuje s obrázky typu: bmp, gif, jpg, pcx, png, tif.

4.1.4 Systémové nároky

Najdi duplikáty má tyto požadavky:

- Intel Pentium III 600 nebo rychlejší
- 256 MB operační paměti
- grafická karta podporující 24-bitový barevný režim
- operační systém Windows XP

4.2 Vývoj programu

4.2.1 Kód aplikace

Najdi duplikáty je aplikace napsaná v jazyce C++. Byly používány některé funkce dostupné ve standardních knihovnách vývojového prostředí, ale především knihovna wxWidgets verze 2.4.1 (www.wxwindows.org), pro tvorbu grafického rozhraní ve spojení s prostředím wxGlade verze 0.3.5.1 (wxglade.sourceforge.net).

Jazyk C++ byl vybrán jako jediná možnost, programování v tomto jazyce bylo obsahem samostatného předmětu v druhém ročníku, mám s ním nejvíc zkušeností. O použití knihovny wxWidgets bylo rozhodnuto kvůli těmto vlastnostem:

- volná dostupnost (licence je zdarma)
- optimální konstrukce (jednoduchost, praktičnost)
- dostatečná funkčnost (velký počet objektů, tříd a metod)
- možnost rychlého překompilování aplikace na jiný operační systém (např. Linux)

Využití této knihovny však není optimální z hlediska rychlosti aplikace. Především práce s obrázky, jejich načítání a manipulace s nimi, je značně omezena jak rychlostí, tak funkčností. Například tato knihovna neobsahuje metodu, která by změnila rozlišení bitmapy. Z tohoto důvodu trvá hledání duplikátů poměrně dlouhou dobu.

4.2.2 Problémy při vývoji

První překážky se objevily již na začátku. Neměl jsem téměř žádné zkušenosti se systémovým programováním. Všechno jsem se teprve učil. Nejprve jsem musel získat představu o pozadí systémového programování, a proto jsem začal s programováním pomocí knihovny Windows API. Napsal jsem tak několik zkušebních aplikací. Při programování mi bylo jasné, že možnosti knihovny jsou dost omezené a nebude možné napsat aplikaci Najdi duplikáty s její pomocí. Především knihovna neobsahuje metody pro načítání obrázků formátu jpeg.

Pak jsem si myslel, že vhodná knihovna by byla Windows MFC verze 6. Opět jsem psal zkušební aplikace, ale zjistil jsem, že ani tato nemá implementovanou podporu obrázků formátu jpeg. Nezbylo než zapomenout na MFC a napsat kód s takovou knihovnou, která toto omezení nemá. Vedoucí této práce mi doporučil knihovnu wxWidgets. Samotnou aplikaci Najdi duplikáty jsem díky těmto komplikacím začal psát až od ledna tohoto roku.

Další velkou překážkou bylo dobře napsat funkci pro hledání duplikátů. Podařilo se mi to po mnohem delší době než jsem předpokládal v časovém plánu. Toto zpoždění způsobilo, že Najdi duplikáty nemá ani základní implementaci metody na hledání duplikátů pomocí morfologické analýzy a funkčnost aplikace je silně omezená.

4.2.3 Průběh vývoje

Doba vývoje byla velmi dlouhá. Tři měsíce mi trvalo získávání zkušeností na programování aplikace takového rozsahu. V tomto období jsem pracoval i s Matlabem a ověřoval, zda metoda korelační analýzy histogramu bude v aplikaci fungovat. Dále jsem si upřesňoval, jak budu při psaní aplikace postupovat. Program Najdi duplikáty jsem začal psát koncem ledna tohoto roku. Vývoj aplikace trval s přípravou asi 7 měsíců a věnoval jsem se mu velice intenzivně.

Práce na programu bude pokračovat. Chtěl bych program dále vylepšovat a nabídnout na internetu volně ke stažení.

4.2.4 Možnosti vylepšení

Možnosti vylepšení jsou především v oblasti funkčnosti a rychlosti programu.

Funkčnost programu by se dala zlepšit:

- úpravou metody tvořící skupiny duplikátů, využití korekční metody na vyloučení nesprávně vyhodnocených duplikátů
- přidáním dalších možností do nastavení programu
- přidáním funkce na zobrazení vybraného obrázku v celoobrazovkovém režimu
- přidáním funkce na uložení a načtení všech výsledků předchozích analýz do a ze souboru
- přidáním funkce na hledání přesných duplikátů
- přidáním funkcí pro smazání výsledků analýzy, smazání výsledků všech předešlých analýz, smazání všech duplikátů k daném originálu, smazání celé skupiny (i s originálem)
- přidáním možnosti zobrazit podobné obrázky jako pole náhledů
- vytvořením verze pro operační systém Linux
- využitím vláken (například při práci se seznamem duplikátů by běžela na pozadí funkce načítající další náhledy obrázků v seznamu, minimalizovalo by se tak čekání uživatele, dále by aplikace mohla přijímat zprávy operačního systému i při provádění analýzy, tím by „nezamrzla“)

Rychlost programu bude možné zvýšit

- ukládáním již načtených histogramů do programové databáze (při analyzování stejného adresáře i po znovuspuštění programu by se nemusel všem nezměněným obrázkům počítat histogram znovu)
- lepším návrhem tříd a úpravou některých metod aplikace ()

4.3 Podobné programy

Funkce na hledání duplikátů zatím není běžnou výbavou prohlížečů obrázků. Nejznámější program pro prohlížení a správu obrázků, ACD See, zatím umožňuje najít pouze přesné duplikáty. Hledání duplikátů je tedy v současnosti funkce, která je dostupná jen ve specializovaných nástrojích. V komerční sféře je těchto nástrojů několik, zatímco v té nekomerční existuje jen Dup Detector, kterému program Najdi duplikáty konkuruje.

Jako příklad dostupných nástrojů pro hledání duplikátů uvádím dva programy:

ABC View

- vysoce výkonný nástroj pro práci s obrázky
- pokročilá funkce pro hledání duplikátů (nenajde však duplikát, který vznikl změnou orientace originálu – jeho otočením)
- cena asi 700,- Kč

Dup Detector

- méně výkonný nástroj než ABC View
- je pomalejší a specializuje se jen na hledání duplikátů (neobsahuje běžné funkce pro prohlížení obrázků)
- je zdarma

4.4 Funkce hledání duplikátů

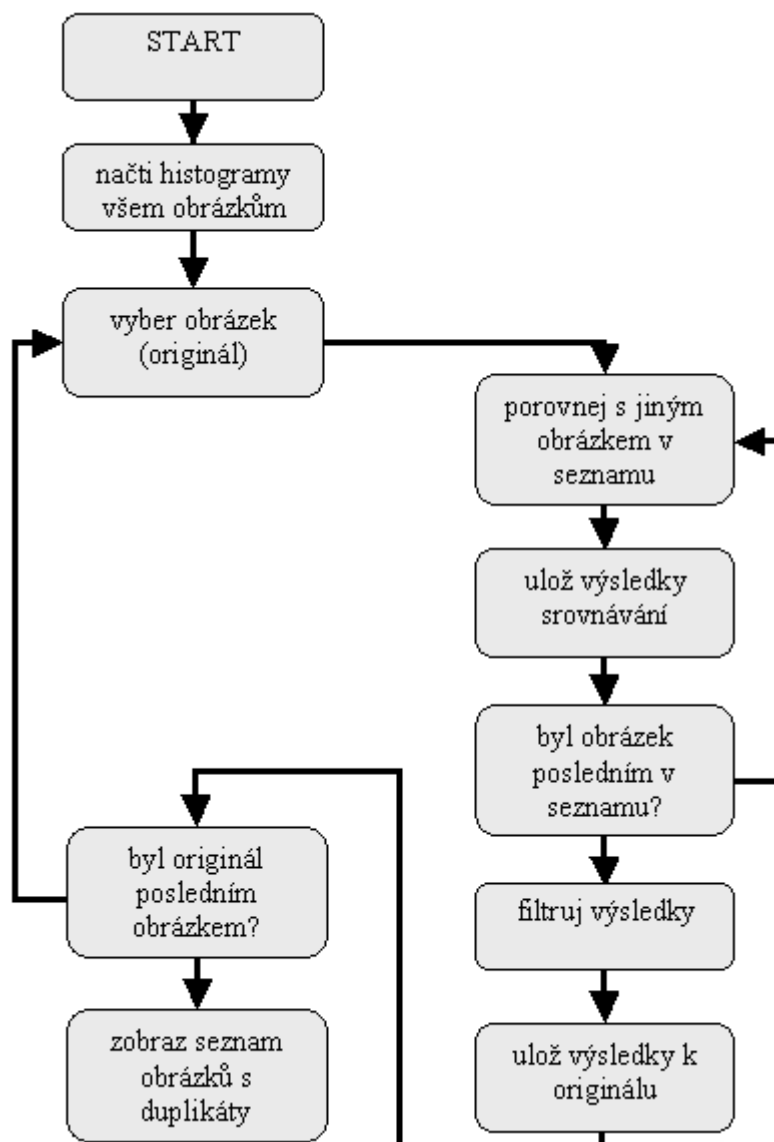
4.4.1 Porovnání dvojice obrázků

Program Najdi duplikáty používá k určení míry podobnosti dvou obrázků pouze korelační analýzu histogramů. Tato metoda byla popsána v kapitole 2.4.1.

Porovnání na základě této metody je spolehlivé přibližně na 60%. Taková spolehlivost není velká a způsobuje problémy. Místo ní by mohla být použita přesnější metoda porovnávající obrázky na základě spektrální analýzy, ale tím by nebyly splněny všechny body zadání této práce. Program by s touto metodou totiž nemohl rozpoznat duplikáty vzniklé otočením originálu.

4.4.2 Algoritmus analýzy obsahu adresáře

Vymyslet dobrý algoritmus pro funkci hledání duplikátů v adresáři bylo obtížné. Základní návrh algoritmu určil již dříve vedoucí této bakalářské práce. Podle něho bylo dobré využít porovnání každého obrázku s každým. Začal jsem tedy tímto postupem a navrhl následující základní algoritmus.



Popis důležitých kroků:

- načti histogram všem obrázkům - program nejprve vytvoří seznam všech obrázků patřících do vybraného adresáře a jeho podadresářů
- porovnej s jiným obrázkem v seznamu – postupně porovná histogram originálu s histogramy všech ostatních obrázků v seznamu
- ulož výsledky srovnávání – do datové struktury příslušející aktuálnímu obrázku uloží korelační koeficienty vyjadřující jeho míru podobnosti první a druhé poloviny histogramů (vůči originálu)
- filtruj výsledky – ze seznamu duplikátů odstraní takové, jejichž součet korelačních koeficientů nedosahuje hranice nastavené uživatelem programu, dále takové, jejichž poměr stran je jiný než poměr stran originálu

- ulož výsledky k originálu – originálu přidá ukazatel na pole datových struktur duplikátů splňujících dané podmínky

Problém tohoto základního algoritmu je, že netvoří vhodné skupiny podobných obrázků. Výsledkem je pouze seznam všech obrázků (z vybraného adresáře), kde ke každému je přiřazen ukazatel na seznam duplikátů, které splňují uživatelem nastavenou toleranci.

P	Název souboru	Velikost	Podobnost	Umístění	Rozlišení
O	dup j1 k20b.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j0-6 k1.bmp	900 KB	54%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k40b.bmp	900 KB	87%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup jchange kchange.jpg	103 KB	79%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup j0-6 k1.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k20b.bmp	900 KB	54%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup jchange kchange.jpg	103 KB	59%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup j1 k40b.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k20b.bmp	900 KB	87%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup jchange kchange.jpg	103 KB	68%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup j1-2 k1.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k20w.bmp	900 KB	59%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1-4 k1.bmp	900 KB	71%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup j1 k20w.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1-2 k1.bmp	900 KB	59%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup j1-4 k1.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1-2 k1.bmp	900 KB	71%	C:_test images_test images2\	640 x 480 (24 bit)
O	dup jchange kchange.jpg	103 KB		C:_test images_test images2\	640 x 480 (24 bit)
D	dup j0-6 k1.bmp	900 KB	59%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k20b.bmp	900 KB	79%	C:_test images_test images2\	640 x 480 (24 bit)
D	dup j1 k40b.bmp	900 KB	68%	C:_test images_test images2\	640 x 480 (24 bit)

Obr. 17 – Výsledky analýzy po porovnání metodou každý s každým

Z obrázku je patrná zbytečná délka seznamu. Ta bývá způsobena tím, že jeden obrázek je běžně podobný k několika obrázkům a je tak v seznamu zastoupen několikrát, pokaždé v jiné skupině. V seznamu na obr. 17 existují mezi skupinami křížové vazby (zvýrazněny). Pojem křížových vazeb je znázorněn v tabulce 10.

skupina	obrázek	podobnost
skupina 1	160_6019.jpg	-
	160_6071.jpg	76%
	160_6012.jpg	74%
	160_6017.jpg	72%

	160_6070.jpg	87%
skupina 2	160_6017.jpg	-
	160_6012.jpg	70%
	160_6019.jpg	72%

Tabulka 10 – Příklad křížové vazby mezi dvojicí skupin

Jak je vidět z tabulky 10, tak křížová vazba je vlastnost dvojice skupin, které obsahují dva stejné prvky originál-duplikát se stejnou podobností, ale v obráceném pořadí.

Úkolem úprav původního algoritmu bylo doplnit ho tak, aby výsledkem byl seznam především s menším počtem skupin. Nejprve jsem ho doplnil o metodu, která jednu ze skupin odstraní v případě, že jsou skupiny spojeny křížovou vazbou a mají téměř stejný počet duplikátů. V případě křížové vazby a výrazněji rozdílného počtu duplikátů se zachovávají obě skupiny, jinak by zůstávaly jen ty největší a tak by vznikala velká chyba.

Další úpravou bylo doplnění původního algoritmu o metodu, která odstraní duplicitní obrázky se seznamu. Tato metoda projde celý seznam a smaže všechny obrázky, které mají předchůdce v jiné skupině se silnější podobností k originálu.

P	Název souboru	Velikost	Podobnost	Umístění	Rozlišení
	dup j1 k20b.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
	dup j0-6 k1.bmp	900 KB	54%	C:_test images_test images2\	640 x 480 (24 bit)
	dup j1 k40b.bmp	900 KB	87%	C:_test images_test images2\	640 x 480 (24 bit)
	dup jchange kchange.jpg	103 KB	79%	C:_test images_test images2\	640 x 480 (24 bit)
	dup j1-2 k1.bmp	900 KB		C:_test images_test images2\	640 x 480 (24 bit)
	dup j1 k20w.bmp	900 KB	59%	C:_test images_test images2\	640 x 480 (24 bit)
	dup j1-4 k1.bmp	900 KB	71%	C:_test images_test images2\	640 x 480 (24 bit)

Obr. 18 – Výsledek upravené verze analýzy

Výsledek po analýze s využitím těchto metod je přehledný (obr. 18), v seznamu jsou tvořeny dobré skupiny duplikátů. Jednotlivé skupiny jsou pro lepší přehlednost odděleny prázdným řádkem.

4.4.3 Rychlost

Rychlost použitého algoritmu je menší. Je to způsobeno především tím, jak jsou obrázky nahrávány do paměti. Je použita funkce knihovny wxWidgets wxImage::LoadImage, která je pomalá.

Zrychlení postupu algoritmu by bylo třeba právě při nahrávání obrázků. Jedním z řešení by bylo využívat k načítání obrázků některou ze systémově závislých funkcí. Takové funkce

jsou napsány tak, aby na daném operačním systému pracovaly rychle. Toto řešení by postup algoritmu nejspíše urychlilo. S kombinací platformou závislého a nezávislého programování však nemám zkušenosti a proto nemůžu zhodnotit, jestli by se zvolené řešení opravdu vyplatilo.

ZÁVĚR

Problematika hledání duplikátů se může zdát složitá. Existují různé metody, které jsou schopny duplikáty identifikovat. Některé jsou složité, jiné velmi jednoduché jako například korelační analýza histogramu. V této práci bylo popsáno několik metod. Z výsledků testování se ukázala jako jednoznačně nejpřesnější metoda založená na spektrální analýze. Testování nebylo obsáhlé, ale myslím si, že výsledky mají vypovídající hodnotu.

Součástí této práce bylo především napsat program pro hledání duplikátů. Program byl vytvořen, dal jsem mu název Najdi duplikáty. Při používání funkce programu na hledání duplikátů nejsou výsledky stoprocentní. Většinou jsou výsledky dobré, avšak občas velmi špatné. Nepřesnost programu je způsobena použitou metodou pro porovnávání dvojic obrázků. K tomu byla zvolena pouze korelační analýza histogramu, protože je velice jednoduchá na praktickou realizaci v použitém jazyce C++ a zpočátku se zdálo, že s ní nebudou problémy. Přesnost výsledků hlavní funkce programu by podle mého názoru bylo dobré zvýšit nějakou jinou, korekční metodou. Ta by z výsledků vyřadila takové duplikáty, které ve skutečnosti podobnými obrázky nejsou. K tomuto by bylo nejlepší zvolit metodu založenou na spektrální analýze. Pokud by se podruhé už neporovnávaly obrázky s mírou podobnosti poukazující na to, že jde o duplikáty vzniklé otočením originálu, pak by toto spojení dvou metod program zlepšilo tak, že by ve hledání duplikátů podával výborné výsledky. Spektrální analýzu nebylo možné použít jako hlavní metodu, protože nerozpoznává duplikáty vzniklé otočením originálu.

Hledání duplikátů je užitečná funkce k lepší organizaci a správě obrázků na počítači. Na internetu se nabízí již dost programů s touto funkcí, většina však pouze v placené verzi. Vývoj je takový, že brzo bude funkce na hledání duplikátů součástí každého lepšího programu určeného ke správě obrázků. V současnosti je vlastní pouze specializovaným nástrojům.

V průběhu vývoje programu Najdi duplikáty, testování jednotlivých metod a různých experimentech se Matlab projevil jako velmi silný nástroj a pomocník. Velkým množstvím funkcí umožňuje programátorovi odzkoušet pouze papírově navržený algoritmus v praktických simulacích. To vede k lepšímu návrhu algoritmu ještě před jeho samotným programováním a tím i úspoře času.

Vývoj programu Najdi duplikáty nebyl snadný zvláště proto, že jsem před touto prací ještě noprogramoval žádnou aplikaci pro operační systém Windows. Hodně jsem se toho musel

učit. Jedním z velkých problémů při vývoji bylo vytvořit funkci na hledání duplikátů tak, aby podávala co nejlepší výsledky. Dlouhou dobu a obtížně ji bylo třeba ladit. Ve vývoji programu chci pokračovat. V současnosti ještě není dobře odladěný a postrádá některé základní funkce.

Bakalářská práce dále zkoumala aplikaci matematické morfologie v metodě, která by umožňovala hledat obrázky obsahující podobný tvar jako na vzorovém obrázku. Dospěl jsem k závěru, že nejlepší by bylo použít k popisu vzoru metodu Eigen Shape Analysis. Stejným způsobem by se zkoumal i nejvýraznější tvar v porovnávaných obrázcích. Bylo zjištěno, že konkrétní realizace takové metody by byla velice složitá. V rámci této práce ji nebylo možné provést.

Matematická morfologie je široké odvětví a nabízí spoustu konkrétních aplikací. Většina prací, do kterých jsem nahlédl, morfologii uplatňuje v biologii a lékařství. Například ke studování mikrobů, konkrétně vývoje jejich populace, byla použita metoda využívající matematickou morfologii k identifikaci mikrobů a jejich počítání.

Tato bakalářská práce mě velmi motivovala k další práci v oboru informačních technologií. Pracoval jsem na ní velice intenzivně.

SEZNAM POUŽITÉ LITERATURY

- [1] JAN, J. : Číslíková filtrace, analýza a restaurace signálů
- [2] PROSISE, J. : Programování ve Windows pomocí MFC
- [3] PETZOLD, C. : Programování ve Windows
- [4] POKORNÝ, P. : Základy počítačové grafiky
- [5] Klíma, M., Bernas, M., Hozman, J., Dvořák P. : Zpracování obrazové informace
- [6] HOZMAN, J. : Základní metody předzpracování obrazu
- [7] MILITKÝ, J. : Obrazová analýza a Matlab
- [8] BLIŽŇÁK, M. : přednášky předmětu Systémové programování
- [9] JAN, J. : přednášky předmětu Systémy, procesy, signály
- [10] ŽITNÝ, R. : přednášky předmětu Experimentální metody 2
- [11] MACHOLÁN, M. : Analýza fenotypu
- [12] návod na vývojového prostředí Matlab

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

jx	x udává jak byl změněn u obrázku jas (např. 0-8, jas snížen o 20%)
kx	x udává jak byl změněn u obrázku kontrast (např. 1-2, kontrast zvýšen o 20%)
vx	x udává jak byla obrázku změněna velikost (např. 0-6, obrázek zmenšen o 40%)
rx	x udává jak byla změněna orientace obrázku (např. -90, obrázek otočen o 90° doleva)
KorAnHist	korelační analýza histogramu
AnHistKF	analýza histogramu korelační funkcí
SpektAn	spektrální analýza

SEZNAM OBRÁZKŮ

Obr. 1 – Pravoúhlá obrazová matice.....	8
Obr. 2 – Princip kvantizace signálu	10
Obr. 3 - Optimální histogram (vlevo), tmavý obraz, světlý obraz (vpravo).....	11
Obr. 4-1 Reálný histogram intenzit	12
Obr. 5 - Obrazová data obrázku o velikosti 5x5 obrazových bodů	13
Obr. 6 – Originální obrázek (vlevo), jeho duplikát (vpravo).....	14
Obr. 7 – Vliv změny jasu na histogram obrázku	16
Obr. 8 – Vliv změny kontrastu na histogram obrázku.....	17
Obr. 9 – Vliv změny velikosti obrázku na histogram obrázku	18
Obr. 10 – Vliv změny orientace na histogram obrázku	18
Obr. 11 – Podobnost histogramů intenzit a zelené složky.....	20
Obr. 12 – Histogramy porovnávaných obrázků a jejich vzájemná korelační funkce	23
Obr. 13 – Normovaná vzájemná korelační funkce z Obr. 12	23
Obr. 14 – Dvourozměrná vzájemná korelační funkce srovnávaných obrázků	25
Obr. 15 – Dvourozměrná normovaná vzájemná korelační funkce	25
Obr. 16 – Hlavní okno programu Najdi duplikáty.....	38
Obr. 17 – Výsledky analýzy po porovnání metodou každý s každým	44
Obr. 18 – Výsledek upravené verze analýzy	45

SEZNAM TABULEK

Tabulka 1 – Data srovnávaných průběhů (vlevo), vypočtené korelační koeficienty	27
Tabulka 2 – Testování neupravené verze korelační analýzy histogramů	31
Tabulka 3 – Testování upravené verze korelační analýzy histogramů.....	32
Tabulka 4 – První testování metod na souboru převážně podobných obrázků	33
Tabulka 5 – Druhé testování metod na souboru převážně podobných obrázků	34
Tabulka 6 – Testování metod na souboru otočených a zmenšených obrázků.....	35
Tabulka 7 – Testování metod na souboru zcela odlišných obrázků	35
Tabulka 8 – Testování metod při porovnávání obrázků stylem každý s každým.....	36
Tabulka 9 – Ovládání prvků hlavního okna	37
Tabulka 10 – Příklad křížové vazby mezi dvojicí skupin.....	45

SEZNAM PŘÍLOH

- P 1 Korelační analýza histogramu (Matlab program)
- P 2 Analýza histogramu korelační funkcí (Matlab program)
- P 3 Spektrální analýza (Matlab program)

PŘÍLOHA P I: KORELAČNÍ ANALÝZA HISTOGRAMU

Program v prostředí Matlab porovnáváající první obrázek ze seznamu se všemi ostatními metodou korelační analýzy histogramu. Výsledkem je tabulka s podobnostmi obrázků k originálu a vyhodnocením zda jde o duplikát či nikoliv.

```
1 clc, clear all;
2
3 seznam_name='obrázky\test1\seznam.txt';%seznam obrazku
4 fid=fopen(seznam_name);
5
6 original=fgetl(fid);%nacteni nazvu, resp. cesty k prvniemu souboru v
    seznamu
7 disp('Hledání duplikátů metodou analýzy histogramu');
8 disp(' ');
9 disp('ORIGINAL je:');
10     disp(original)
11
12 obr1=imread(original); %nacte original
13 obr1gray=rgb2gray(obr1); %prevede do odstínu šedi
14 hist1=imhist(obr1gray); %spocte histogram obrazku
15
16 vysledky(1, 1:5, 1)=0;
17 i=1;
18
19 while 1
20     duplikat=fgetl(fid);
21     if ~ischar(duplikat), break, end
22
23 obr2=imread(duplikat); %nacte srovnávaný obrázek
24 obr2gray=rgb2gray(obr2);
25 hist2=imhist(obr2gray);
26 clear obr2;
27
28 %rozdelení histogramu hist 1 do dvou částí
29 hist1part1=hist1;
30 hist1part2=hist1;
31 hist1part1(129:256,:) = [];
32 hist1part2(1:128,:) = [];
33
34 %rozdelení histogramu hist 2 do dvou částí
35 hist2part1=hist2;
```

```

36 hist2part2=hist2;
37 hist2part1(129:256,:) = [];
38 hist2part2(1:128,:) = [];
39
40 %korelacni koeficienty:
41 vysledky(i, 1, 1)=i+1;
42
43 R1=corrcoef(hist1part1,hist2part1); %korelacni analyza prvni
    poloviny histogramu
44 vysledky(i, 2, 1) = R1(2);
45
46 R2=corrcoef(hist2part2,hist1part2); %korelacni analyza druhe
    poloviny histogramu
47 vysledky(i, 3, 1) = R2(2);
48
49 %nulovani zapornych koeficientu
50 r1t=0;
51 r2t=0;
52 if (R1(2) > 0)
53     r1t=R1(2);
54 end;
55 if (R2(2) > 0)
56     r2t=R2(2);
57 end;
58
59 vysledky(i, 4, 1) = (r1t+r2t)/2; %vypocteni podobnosti srovnavanych
    obrazku
60
61 %rozhodnuti zda jde o podobne obrazky nebo nikoliv na zaklade
    nastavene
62 %tolerance
63 if ((r1t+r2t)/2 >= 0.65) %tolerance = 0.65
64     vysledky(i, 5, 1) = 1;
65 else vysledky(i, 5, 1) = 0;
66 end;
67
68 i=i+1;
69 end
70 fclose(fid);
71
72 disp(' ');
73 disp(vysledky); %vypis vysledku porovnani
74

```

```
75 fid=fopen(seznam_name);
76 disp('seznam srovnanych obrazku: '); %vypis porovnavanych obrazku
77 while 1
78     tline = fgetl(fid);
79     if ~ischar(tline), break, end
80     disp(tline)
81 end
82 fclose(fid);
```

PŘÍLOHA P II: ANALÝZA HISTOGRAMU KORELAČNÍ FUNKCÍ

Program v prostředí Matlab porovnáváající první obrázek ze seznamu se všemi ostatními metodou analýzy histogramu korelační funkcí. Výsledkem je tabulka s podobnostmi obrázků k originálu a vyhodnocením zda jde o duplikát či nikoliv.

```
1 clc, clear all;
2
3 seznam_name='obrazky\test1\seznam.txt';%seznam obrazku
4 fid=fopen(seznam_name);
5
6 original=fgetl(fid);%nacteni nazvu, resp. cesty k prvniemu souboru v
    seznamu
7 disp('Hledání duplikátů metodou analyzy histogramu pomoci vzajemne')
8 disp('korelacni funkce');
9 disp(' ');
10 disp('ORIGINAL je:');
11     disp(original)
12
13 obr1=imread(original); %nacte original
14 obr1gray=rgb2gray(obr1); %prevede do odstínu sedí
15 hist1=imhist(obr1gray); %spocte histogram obrazku
16
17 vysledky(1, 1:3, 1)=0;
18 i=1;
19
20 while 1
21     duplikat=fgetl(fid);
22     if ~ischar(duplikat), break, end
23
24 obr2=imread(duplikat); %nacte srovnávaný obrázek
25 obr2gray=rgb2gray(obr2);
26 hist2=imhist(obr2gray);
27 clear obr2;
28
29 vysledky(i, 1, 1)=i+1;
30
31 %stanovení vzajemné korelační funkce
32 Rfunkce=xcorr(hist1,hist2,'coeff');
33 maximum=max(Rfunkce);
34 vysledky(i, 2, 1) = maximum;
35
```

```

36 %rozhodnuti zda jde o podobne obrazky nebo nikoliv na zaklade
    nastavene
37 %tolerance
38 if (maximum >= 0.85) %tolerance = 0.85
39     vysledky(i, 3, 1) = 1;
40 else vysledky(i, 3, 1) = 0;
41 end;
42
43 i=i+1;
44 end
45 fclose(fid);
46
47 disp(' ');
48 disp(vysledky); %vypis vysledku porovnavani
49
50 fid=fopen(seznam_name);
51 disp('seznam srovnavanych obrazku: '); %vypis porovnavanych obrazku
52 while 1
53     tline = fgetl(fid);
54     if ~ischar(tline), break, end
55     disp(tline)
56 end
57 fclose(fid);

```

PŘÍLOHA P III: SPEKTRÁLNÍ ANALÝZA

Program v prostředí Matlab porovnáváající první obrázek ze seznamu se všemi ostatními metodou založenou na spektrální analýze. Výsledkem je tabulka s podobnostmi obrázků k originálu a vyhodnocením zda jde o duplikát či nikoliv.

```
1 clc, clear all;
2
3 seznam_name='obrázky\test1-res\seznam.txt';%seznam profilu obrazku
  (zmensenin)
4 fid=fopen(seznam_name);
5
6 original=fgetl(fid);%nacteni nazvu, resp. cesty k prvnimu souboru v
  seznamu
7 disp('Hledání duplikátů metodou zalozenou na spektralni analyze')
8 disp(' ');
9 disp('ORIGINAL je:');
10     disp(original)
11
12 obr1=imread(original); %nacte original
13 obr1gray=rgb2gray(obr1); %prevede do odstínu sedí
14 hist1=imhist(obr1gray); %spocte histogram obrazku
15
16 vysledky(1, 1:3, 1)=0;
17 i=1;
18
19 while 1
20     duplikat=fgetl(fid);
21     if ~ischar(duplikat), break, end
22
23 obr2=imread(duplikat); %nacte srovnávaný obrázek
24 obr2gray=rgb2gray(obr2);
25 hist2=imhist(obr2gray);
26 clear obr2;
27
28 vysledky(i, 1, 1)=i+1;
29
30 %stanovení vzájemné dvourozměrné korelační funkce
31 RXYfunkce=normxcorr2(obr1gray, obr2gray);
32 maximum=max(max(RXYfunkce));
33 vysledky(i, 2, 1) = maximum;
34
```

```
35 %rozhodnuti zda jde o podobne obrazky nebo nikoliv na zaklade
    nastavene
36 %tolerance
37 if (maximum > 0.90)
38     vysledky(i, 3, 1) = 1;
39 else vysledky(i, 3, 1) = 0;
40 end;
41
42 i=i+1;
43 end
44 fclose(fid);
45
46 disp(' ');
47 disp(vysledky); %vypis vysledku porovnavani
48
49 fid=fopen(seznam_name);
50 disp('seznam srovnavanych obrazku: '); %vypis porovnavanych obrazku
51 while 1
52     tline = fgetl(fid);
53     if ~ischar(tline), break, end
54     disp(tline)
55 end
56 fclose(fid);
```