VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ VODNÍCH OBJEKTŮ V OBRAZE

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE DAVID ČELOUD
AUTHOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ VODNÍCH OBJEKTŮ V OBRAZE

WATER OBJECT DETECTION IN IMAGE

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE DAVID ČELOUD
AUTHOR

VEDOUCÍ PRÁCE ING. JANA ŠILHAVÁ
SUPERVISOR

BRNO 2008

Abstrakt

Bakalářská práce popisuje historii remote sensing, strukturu satelitních snímků, jejich zpracování a analýzu. Definuje multispektrální prostor a vysvětluje základy kvantitativní analýzy a rozdíly mezi supervised a unsupevised klasifikací. Sekce implementace popisuje návrh a vývoj programu, který bude schopen otevřít a zpracovat satelitní snímek a vyhledat v něm vodní objekty.

Klíčová slova

dálkový průzkum Země, satelitní snímky, fotointerpretace, skládání umělých barev, saturační lineární metoda posílení kontrastu, kvantitativní analýza, multispektrální prostor, spektrální třídy, unsupervised klasifikace, pohyblivé středy, euklidovská vzdálenost

Abstract

Bachelor’s thesis describes history of remote sensing, image data structure, their processing and analyzing. It defines mulstispectral space and explains basics of quantitative analysis and differences between supervised and unsupervised classification. Implementation section describes designing and developing of program, which will be able to open and process image data and detect water objects in him.

Keywords

remote sensing, image data, photointerpretation, false color composite, saturating linear contrast enhancement, quantitative analysis, multispectral space, spectral classes, unsupervised classification, migrating means, Euclidean distance

Citace

Čeloud David: Vyhledávání vodních objektů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2008.

Vyhledávání vodních objektů v obraze

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jany Šilhavé. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
David Čeloud
13. května 2008

Poděkování

Děkuji Ing. Janě Šilhavé za skvělé vedení, shovívavost a velmi inspirativní a konstruktivní rady, připomínky a návrhy k vypracování této bakalářské práce.

Obsah

Úvod

Je dávným posláním člověka objevovat, poznávat a odhalovat. Byla to zvědavost, která nutila lidstvo ptát se, co se nachází za dalším kopcem, a už v dávných dobách průzkumníci sledovali z vyvýšeného místa počínání nepřátelské armády. Rozvoj elektrotechniky způsobil obrovský převrat. Díky němu můžeme na kopec místo průzkumníka posadit stroj a ze vzdáleného sledovacího střediska uvidíme co dělá nepřátelská armáda dokonce i v noci.

Zařízení umožňující plnit lidem jejich poslání z pohodlí domova jsou s velkým úspěchem umisťována všude. V obchodě vás hlídač ze všech různých úhlů sleduje kamerami, na letišti vám personál vidí skoro až do žaludku a fotografie s vašim silničním rychlostním rekordem vám přijde až do domu. Oblast využití elektronických očí, uší a nosů je široká a rozmanitá a neustále se rozšiřuje.

Jednou z takových oblastí je právě již dříve zmíněné sledování armád, krajin a cest pod nimi a vůbec celkově pozorování zemského povrchu jako takového. Stejně jako Jeníček ze známé pohádky i senzory musí sledovat povrch z výšky a je třeba je tam dostat. Obecně platí, že čím větší výška, tím lepší rozhled.

Doba, kdy kolem Země obíhaly jen Měsíc a Sputnik, jsou dávno pryč. Dnes různé satelity nepřetržitě snímají zemský povrch v mnoha různých částech spektra elektromagnetického záření. Data posílaná z těchto satelitů zpět na Zem je třeba zaznamenat a zpracovat například za účelem nalezení vodních objektů nacházejících se na snímaném povrchu. O tom je moje bakalářská práce

Technická zpráva je členěna do dvou hlavních sekcí. Kapitola Něco teorie (1) a její podkapitoly představí čtenáři data ze satelitů, popíší jejich vlastnosti a uvedou techniky jaké lze použít pro získání co nejvíce informací.

Podkapitola Remote sensing (1.1) slouží čtenáři k obecnému uvedení do historie a problematiky dálkového vidění. Podkapitola Satelitní snímky (1.2) přibližuje základní strukturu satelitních dat a rozdíly, v nichž se liší například od normálních počítačových obrázků. V podkapitole Analýza satelitních snímků (1.3) se čtenář může dočíst o principech zpracování obsahu satelitních snímků pomocí různých matematických a logických metod.

Druhou část zprávy tvoří kapitola Implementace (2). Ta čtenáři podrobně vysvětlí průběh realizace zadání bakalářské práce.

Podkapitola Hlavní menu (2.1) popisuje vytváření hlavního formuláře programu, nejdůležitější části GUI. Následující dvě kapitoly popisují dvě základní implementační části bakalářské práce. Kapitola Zpracování satelitních snímků (2.2) rozebírá postup při vytváření funkčních částí programu zodpovědných za načtení satelitních dat ze souboru a jejich následné zobrazení. Kapitola Klasifikace satelitních snímků (2.3) popisuje postup vytváření funkčních částí programu provádějícíh analýzu satelitních snímků a vyhledávání vodních objektů v nich.

1 Něco teorie

Kapitola obsahuje teoretické skutečnosti, které jsem zpracoval před samotnou tvorbou bakalářské práce. Snaží se přiblížit čtenáři problematiku remote sensing a analýzy satelitních snímků v tom rozsahu, aby mu čtení dalších kapitol nečinilo žádné závažné problémy. Nepovažuji sám sebe za autora žádné myšlenky zaznamenané v této kapitole.

1.1 Remote sensing

1.1.1 Remote sensing obecně

Definice říká, že: „Dálkový průzkum (český ekvivalent pro remote sensing) je věda i umění získávat užitečné informace o objektech, plochách či jiných jevech prostřednictvím dat měřených na zařízeních, která s těmito zkoumanými objekty, plochami či jevy nejsou v přímém kontaktu.“ [1]

Remote sensing je moderní metoda získávání údajů o jevech, které nejsou v přímém kontaktu se zařízením, které tyto údaje zaznamenává (jako třeba radar, letadlo, loď, satelit, atd.). V praxi se jedná o sběr dat o daném objektu nebo oblasti pomocí různých druhů zařízení. Takže například satelity zkoumající Zemi a sledující počasí nebo ultrazvuk pro zjišťování pohlaví dítěte, magnetická rezonance užívaná v lékařství , to vše jsou příklady remote sensing. [2]

Existují dva druhy dálkového průzkumu. Pasivní senzory zaznamenávají záření, které je vyzařováno, nebo se odráží od pozorovaného objektu nebo plochy. [2] Odražené sluneční světlo je nejčastějším zdrojem záření zaznamenávaného pasivními senzory. Mezi pasivní senzory patří fotoaparát, kamera, infračervená kamera, atd. Aktivní senzory na druhou stranu nevyužívají přirozeného záření z okolí, ale samy ozařují pozorované objekty a následně detekují a měří odražené záření od cíle. [2] Typickým představitelem této skupiny je radar, kde časové zpoždění mezi vyzářeným a odraženým paprskem pomůže určit polohu, výšku, rychlost a směr pohybu objektu.

Vznik Remote sensing umožnil do té doby problematické sledování objektů v nebezpečných nebo nepřístupných oblastech. [2] Můžeme tak sledovat mizení deštných pralesů v povodí Amazonky, vliv změny klimatu na ledovce v Arktidě a Antarktidě nebo provádět průzkum dna mořského pobřeží i hlubin oceánů. V období Studené války se remote sensing často využíval pro získávání důvěrných informací o druhé straně konfliktu. [2]

Orbitální platformy (družice a satelity) shromažďují data z různých částí elektromagnetického spektra, což ve spojení s velkou zkoumanou oblastí umožňuje vědcům sledovat jevy jako el niño a jiné krátkodobé či dlouhodobé fenomény. [2] Další užití zahrnuje různé oblasti průzkumu Země jako natural resource management, sledování růstu zemědělských plodin, národní bezpečnost, špionáž, atd. [2]

1.1.2 Historie remote sensing

První začátky této moderní metody spadají do období vzniku létajících dopravních prostředků. Vzduchoplavec G. Tournachon, přezdívaný Nadar, roku 1858 vyfotil ze svého balónu Paříž a stal se tak prvním leteckým fotografem. [1] První taktické využití nové metody bylo během Americké občanské války. [2] Poštovní holubi, papíroví draci, rakety a neřízené balony byli používáni pro pořizování prvních obrázků. [1] Tyto obrázky však nebyly vhodné pro vytváření map nebo pro vědecké účely.

Systematické letecké fotografování bylo vyvinuto pro vojenské sledovací a průzkumné potřeby během První světové války a vyvrcholilo v průběhu Studené války vývojem špionážních průzkumných letadel jako např. *U-2*. [2] Pozdější vývoj se zaměřil na zmenšování senzorů a jejich seskupování na řízených i neřízených platformách (letadlech, lodích, …)

Rozvoj umělých satelitů v druhé polovině 20. století umožnil remote sensing v globálním měřítku. Senzory instalované na různé satelity zkoumající povrch Země nebo pro sledování počasí jako *Landsat*, *Nimbus* nebo novější *RADARSAT* a *UARS* poskytly mnoho dat pro civilní, věděcký nebo vojenský sektor. [2] Vesmírné sondy k jiným planetám poskytly také příležitost provádět remote sensing studie v mimozemských prostředích. Například radar na sondě *Magellan* poskytl detailní topografické mapy Venuše a senzory sondy *SOHO* umožnily vědcům studovat Slunce a jeho solární větry. [2]

V 60. a 70. letech došlo k dalšímu zdokonalení vlivem rozvoje metod zpracování satelitních snímků. Několik výzkumných skupin v Silicon Valley jako *NASA Ames Research Center*, *GTE* a *ESL Inc*. vyvinulo techniky Fourierovy transformace, vedoucí k prvnímu významnému zlepšení kvality satelitních snímků. [2]

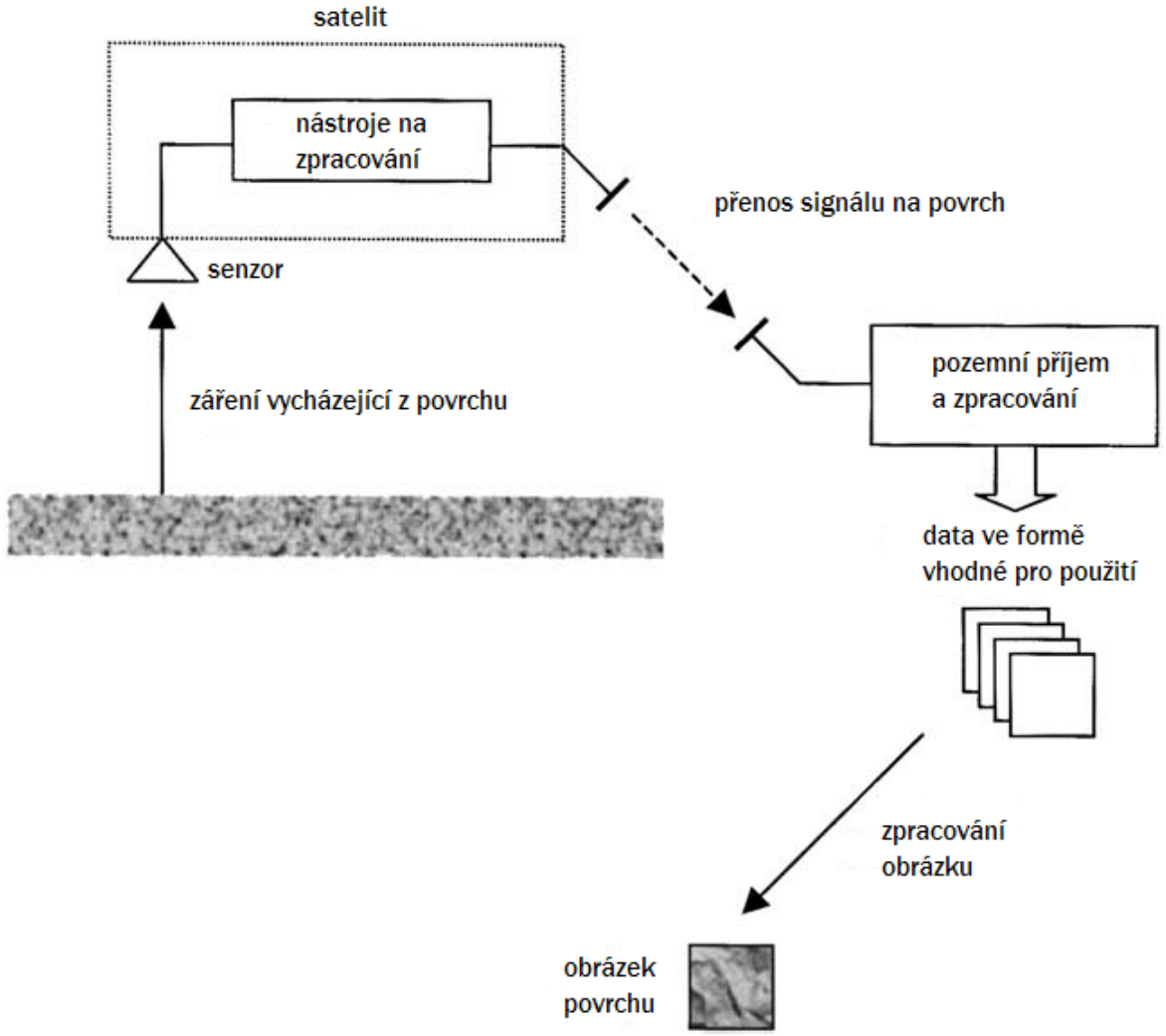
Rozvoj online webových služeb pro jednoduchý přístup k remote sensing datům v 21. století (jako např. Gogle Earth) zpopularizoval remote sensing pro širokou veřejnost. [2]

1.2 Satelitní snímky

1.2.1 Jak vznikají?

Satelitní snímky vznikají měřením energie vyzařované ze zemského povrchu senzorem umístěným na satelitu. Toto měření je použito ke konstrukci snímku popisujícího povrch pod satelitem. [3]

Měřenou energií může být odražené sluneční záření, takže vytvořené snímky připomínají obrázek, který by viděl kosmonaut koukající se dolů ze satelitu, jen s tím rozdílem, že vlnové délky používané při remote sensing jsou často mimo rozsah lidského oka. Dalším možným zdrojem zachycovaného záření může být samotná Země díky své teplotě. A jako poslední možnost to může být nějaká rozptýlená energie pocházející z umělého zdroje, jako je třeba laser nebo radar umístěný na satelitu. [3]



Obrázek 1: Postup zpracování satelitního snímku [3]

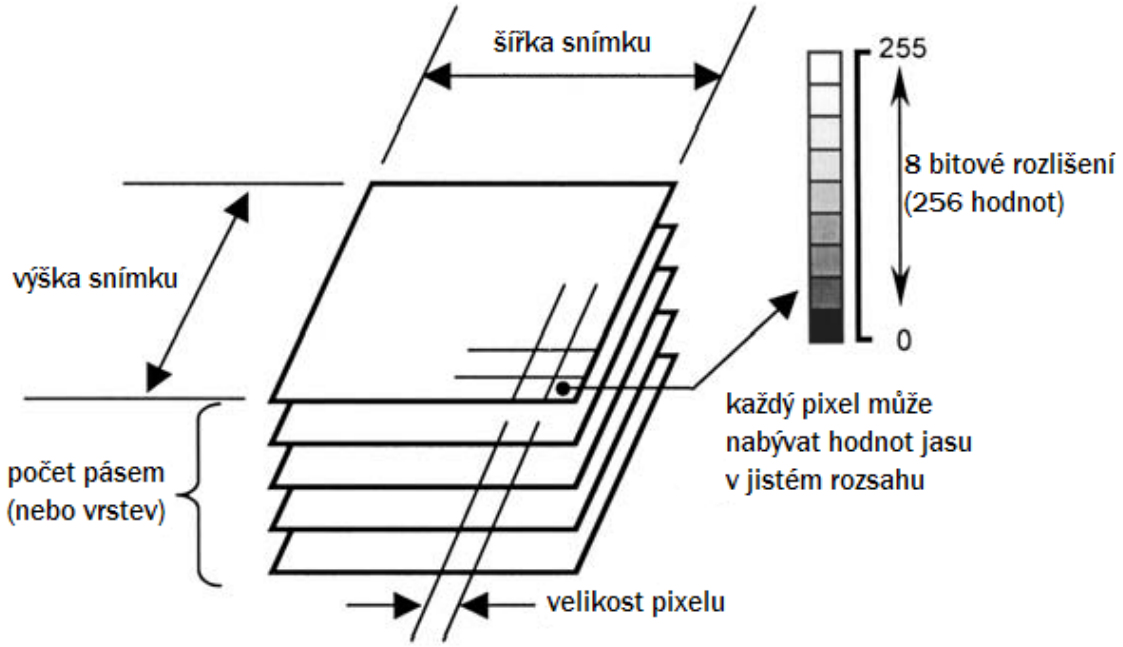
Odražená nebo vyzářená energie z povrchu Země prochází atmosférou k satelitnímu senzoru, poté dojde k přenosu dat zpět na zem a jejich zpracování do podoby použitelné pro uživatele.

1.2.2 Vlastnosti satelitních snímků

Jednou z hlavních přínosných vlastností satelitních snímků je, že jsou běžně dostupné v digitální podobě. To znamená, že plocha snímku je rozdělena na diskrétní obrazové prvky, pixely, a i jejich hlavní radiometrická hodnota je převedena na diskrétní hodnoty. [3] Navíc data, která nebyla původně zaznamenávána digitálně, lze jednoduše digitalizovat. V počátcích remote sensing byla většina satelitních snímků v analogové podobě. V současnosti je většina dostupná přímo v podobě digitální. [3]

Velkou výhodou digitální podoby satelitních snímků je jejich snadná počítačová zpracovatelnost, ať už za účelem počítačové analýzy nebo kvůli posílení vizuální kvality pro lepší možnost analýzy odborníkem.

Pravděpodobně nejvýznamnější charakteristikou dat v systému remote sensing je vlnová délka nebo rozsah vlnových délek použitých při snímacím procesu. [3] Jedná-li se o odražené sluneční záření, snímky mohou být zaznamenávány v ultrafialové oblasti, oblasti viditelné části spektra a v oblasti blízkého a středního infračerveného záření. Vzhledem k výraznému útlumu v atmosféře se v ultrafialové oblasti měření neprovádí. Nejběžnější remote sensing systémy zaznamenávají v rozsahu od viditelného spektra, přes spektrum blízkého infračerveného až ke střednímu infračervenému záření. [3]



Obrázek 2: Formální charakteristika satelitních snímků [3]

Z pohledu struktury dat a pro analytické účely jsou nejdůležitější tyto vlastnosti (Obrázek 2):

- počet a umístění spektrálních měření provedených příslušnými senzory (takzvané spektrální vrstvy, pásma nebo kanály)
- rozloha, kterou popisuje plocha jednoho pixelu
- radiometrické rozlišení jednotlivých pixelů

Radiometrické rozlišení popisuje v jakém rozsahu se mohou vyskytovat diskrétní hodnoty jasu pixelů. [3] Často se radiometrické rozlišení vyjadřuje v počtech bitů potřebných pro zobrazení všech hodnot rozsahu. Takže například data s 8 bitovým radiometrickým rozlišením mají 256 úrovní jasu.

Všechny tyto údaje, jako šířka a výška jedné spektrální vrstvy, počet spektrálních vrstev, radiometrické rozlišení a rozloha jednoho pixelu, jsou charakteristické pro různé satelity.

Pokud je ve snímku uložena více, než jedna spektrální vrstva, označují se takové snímky jako *multispektrální*. [3]

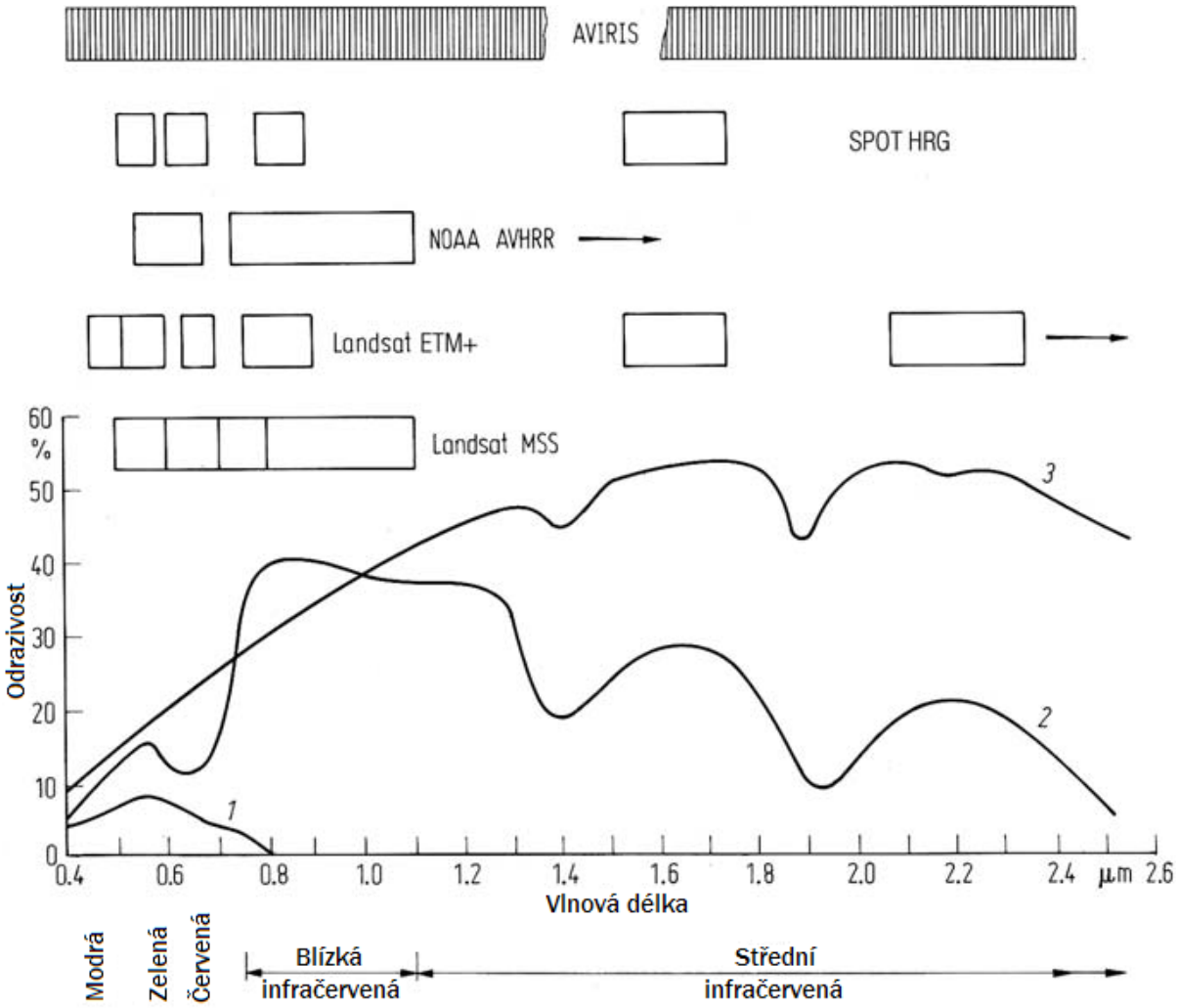
1.2.3 Spektrální rozsahy nejčastěji používané v remote sensing

Teoreticky mohou remote sensing systémy měřit energii vyzařující ze zemského povrchu v jakémkoliv představitelném rozsahu vlnových délek. Nicméně technologické možnosti, vliv atmosféry způsobující silný útlum některých frekvencí a rozptyl signálu částicemi vody v atmosféře přinutily některé vlnové délky vyloučit. Nejpoužívanější rozsahy pro sledování zemského povrchu jsou v rozsahu 0,4 μm až 12 μm (oblast viditelného a infračerveného spektra) a v rozsahu 30 mm až 300 mm (oblast mikrovlnného záření). [3]

Významnost těchto dvou rozsahů souvisí s interakcí mezi elektromagnetickým zářením a zkoumanými materiály. [3] V oblasti viditelného a infračerveného spektra závisí energie naměřená senzory na vlastnostech jako barva, obsah vlhkosti a buněčná struktura u rostlin, obsah minerálů a vlhkost půdy a úroveň sedimentace vodních ploch. Na termálním konci infračerveného spektra (směrem k větším vlnovým délkám) je množství vyzářené energie ovlivněno například tepelnou kapacitou a dalšími tepelnými vlastnosti povrchu i mělkého podloží. V oblasti mikrovlnného spektra vyzářenou energii ovlivňuje nerovnost terénu a jeho elektrické vlastnosti (silně závisející na jeho vlhkosti). [3]

Obrázek 3 ukazuje, jak tři dominantní povrchové materiály (půda, vegetace a voda) odráží sluneční záření v oblasti viditelného a infračerveného spektra. Z grafu je patrno, že voda odráží méně než 10 % v modré a zelené oblasti, malé procento v červené oblasti a prakticky téměř žádnou energii v infračervené části spektra. Pokud je voda zkalená, nebo dostatečně mělká, že je vidět na dno, dochází k malému, ale významnému odrazu v oblasti blízkého infračerveného spektra.

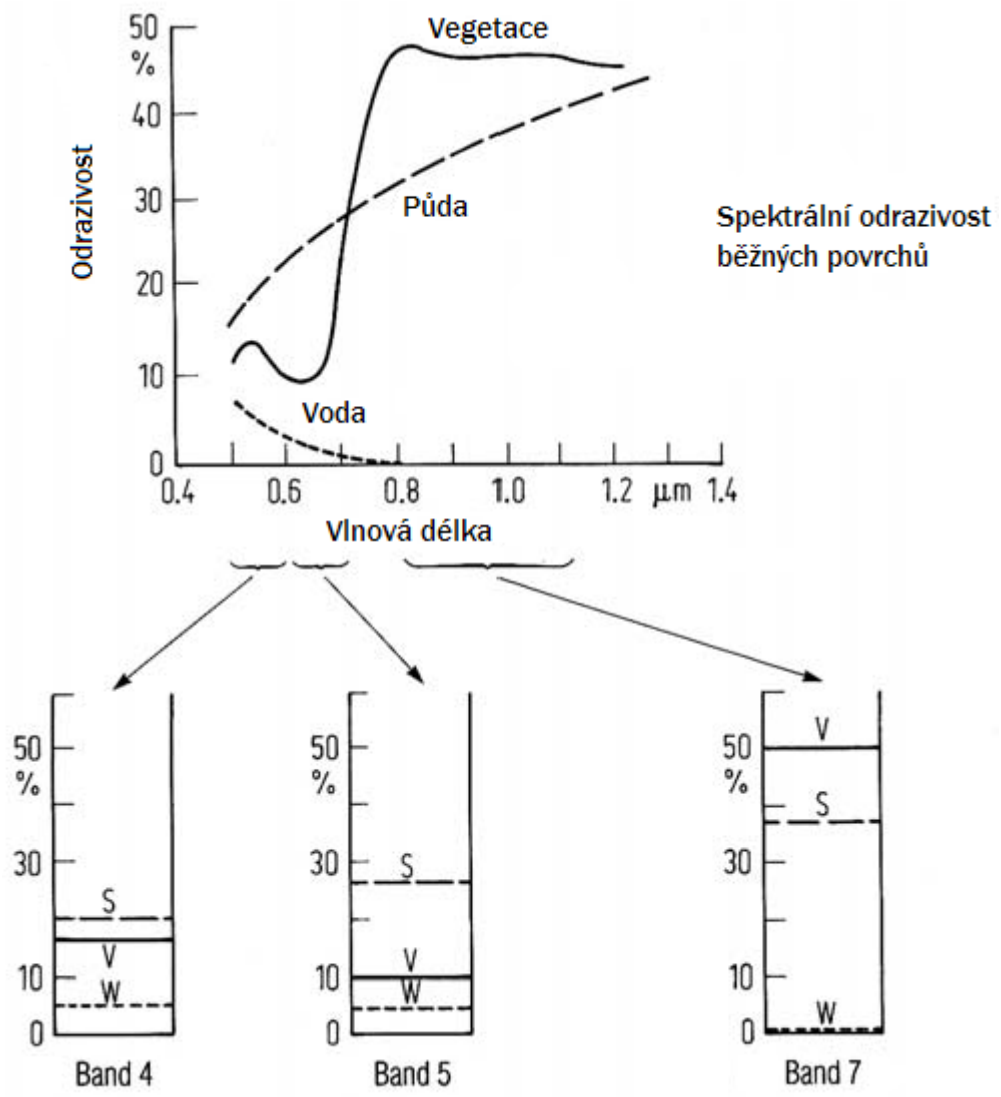
Křivka odrazivosti půdy roste přibližně monotónně s malými poklesy v 1,4 μm, 1,9 μm a 2,7 μm vlivem vlhkosti. [3] Tyto poklesy jsou téměř nepostřehnutelné u velmi suché půdy a písku.



Obrázek 3: Charakteristiky odrazivosti běžných materiálů povrchu Země v závislosti na vlnové délce v oblasti viditelného a blízkého a středního infračerveného záření. *1* voda, *2* vegetace, *3* půda [3]

Oproti tomu jílovité půdy mají tyto propady v 1,4 µm a 2,2 µm. [3]

Křivka vegetace je podstatně komplexnější, než předchozí dvě křivky. V oblasti 1,4 µm, 1,9 µm a 2,7 µm se nachází známé poklesy způsobované absorpcí vody. [3] Oblast mezi 0,7 µm a 1,3 µm je ovlivňována buněčnou strukturou vegetace a ve viditelné oblasti spektra hraje nejdůležitější roli barva. [3] Z grafu je také velmi dobře patrné, jak chlorofyl absorbuje záření z modré a červené oblasti a propouští záření ze zelené oblasti spektra. To je také důvod, proč vidíme rostliny zeleně.



Obrázek 4: Souvislost charakteristiky odrazivosti běžných povrchů s hodnotami jasu v některých spektrálních vrstvách satelitního snímku. **V** – vegetace, **S** – půda, **W** – voda [3]

1.3 Analýza satelitních snímků

1.3.1 Analýza obecně

Při analýze satelitních snímků v digitální podobě existuje několik různých postupů ve snaze získat informace. Jeden postup zahrnuje využití počítače k prozkoumání každého pixelu v satelitním snímku a jeho posouzení čistě na základě jeho vlastností. Tento postup je označován jako *kvantitativní analýza*, jelikož pixely podobných vlastností jsou počítány kvůli posouzení celého snímku. [3] Další přístup spočívá v analýze a získávání informací vizuálním průzkumem obrázku vytvořeného vhodným zobrazením satelitního snímku. Tento postup je označován jako *fotointerpretace*. [3] Jeho

úspěšnost závisí na schopnosti analytického odborníka správně identifikovat a používat prostorové, časové a spektrální prvky obsažené ve vytvořeném obrázku. Prostorová informace je přítomna například v popisech tvarů, velikostí, orientace a textur. [3] Silnice, pobřeží a vodní toky, velké terénní nerovnosti a zlomy obecně jsou obvykle jednoduše identifikovatelné podle jejich prostorového uspořádání. Časové informace, jako například změna jednotlivých objektů nebo typu pokrytí v satelitních snímcích stejné oblasti, ale s různým datem snímání, mohou být často používány odborníkem třeba pro rozlišení listnatých (opadavých) a jehličnatých (neopadavých) lesů. [3] Odborník mající dostatečné zkušenosti a znalosti spektrální charakteristiky odrazivosti běžných materiálů je schopen na základě spektrálních informací odhadnout typy povrchů zobrazených na vytvořeném obrázku. [3]

Tyto dva výše popsané přístupy k analýze satelitních snímků mají svoje vlastní uplatnění a často se navzájem doplňují. Tam, kde je fotointerpretace silná, kvantitativní analýza nemá dostatečnou analytickou schopnost a naopak. [3] Obecně platí, že fotointerpretace, jejíž výkonnou jednotkou je mozek analytického odborníka, je vhodnější a výkonnější v oblasti rozeznávání tvarů a hranic mezi různými zobrazenými povrchy, zatímco kvantitativní analýza, jejíž výkonnou jednotkou je procesor počítače, je naopak výkonnější v oblasti přesného měření, jako jsou odhady rozloh různých povrchů nebo spektrální analýzy, jelikož je počítač na rozdíl od lidského oka schopen rozlišovat mnohem více úrovní jasu. [3]

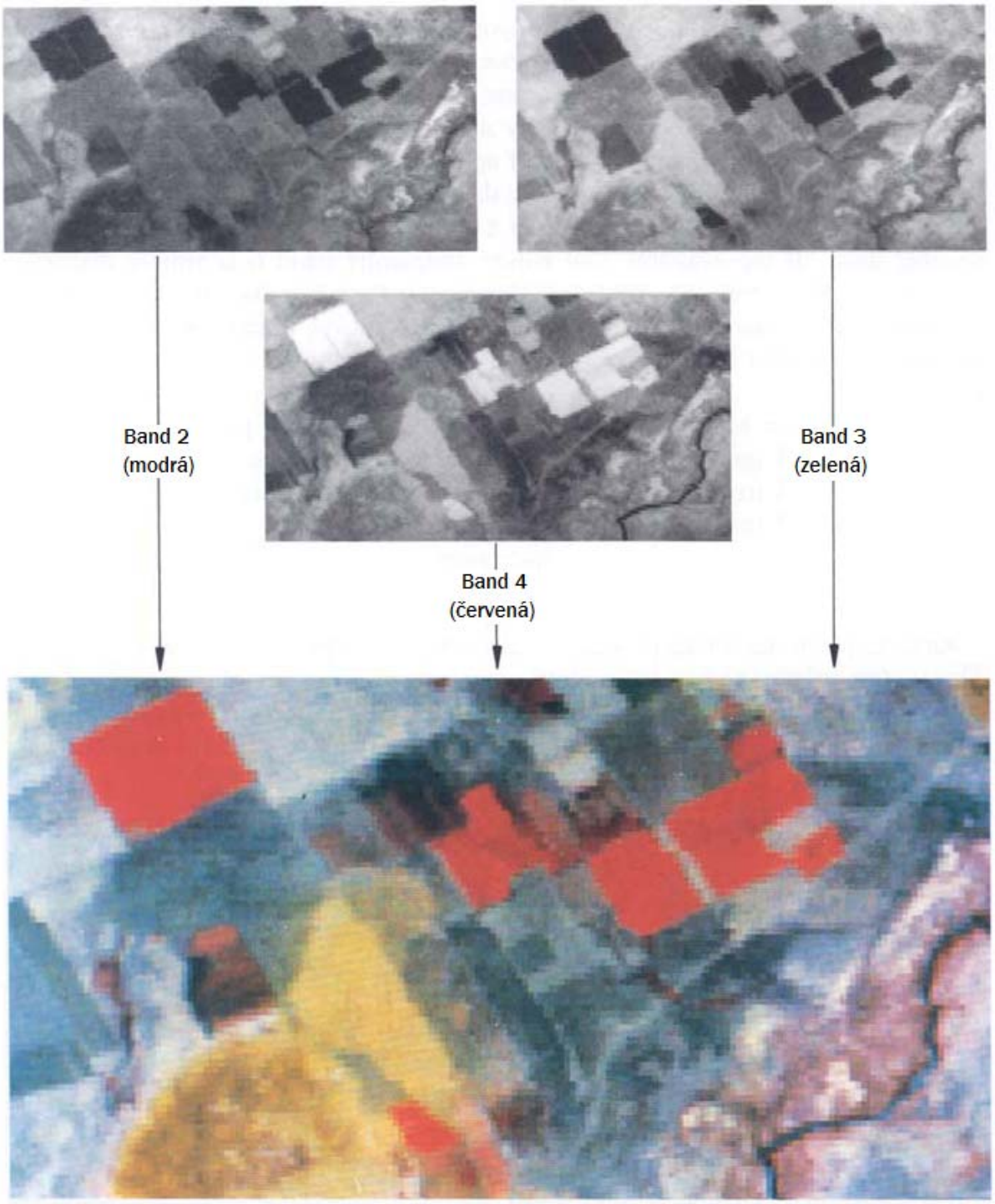
1.3.2 Zobrazovací metody používané při fotointerpretaci

Samotná skutečnost, že o satelitních datech mluvíme jako o satelitních snímcích, nejmenší uložená plocha tohoto snímku se jmenuje pixel a hodnoty, kterých mohou jednotlivé pixely nabývat, se nazývají hodnoty jasu, poukazuje na jistou podobnost mezi satelitním snímkem a například obrázkem v počítači. Zobrazit normální obrázek je vcelku jednoduché, ale jak je to se satelitním snímkem?

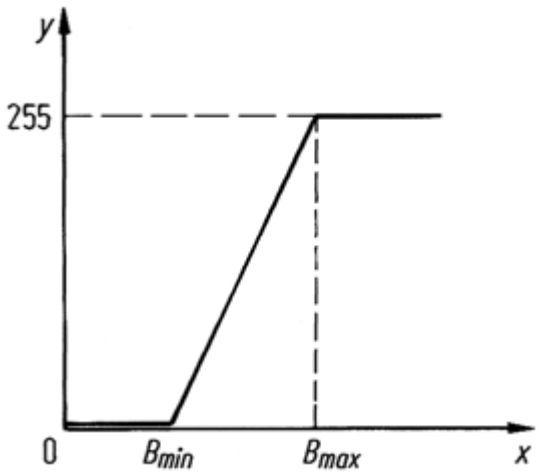
Hlavním problémem satelitního snímku je ten, že je multispektrální a navíc pixely v každé vrstvě mohou nabývat naráz pouze jedné hodnoty. [3] To prakticky znamená, že se každý satelitní snímek skládá z několika různých černobílých obrázků. Samotné černobílé obrázky však nemají příliš velkou informační hodnotu. Lidské oko nedokáže dostatečně kvalitně posoudit rozdíly dvou různých obrázků a jeho rozlišovací schopnost zdaleka nestačí na všechny úrovně šedi, které dokáže počítač vykreslit.

Pixel běžně používaný v počítačové grafice má tři barevné složky, neboli tři hodnoty jasu – červenou, zelenou a modrou. To nám umožňuje zobrazit naráz tři černobílé obrázky (spektrální vrstvy) v jednom obrázku, barevném. Obrázek 5 ukazuje, jak vypadá takové spojování více spektrálních vrstev dohromady. V praxi se tato metoda nazývá *false color composite*, česky skládání umělých / falešných barev. [3] Název souvisí se skutečností, že jsme schopni vhodnou kombinací spektrálních vrstev vytvořit obraz, který by z vesmíru uviděl kosmonaut s velmi dobrým zrakem, ale

jsme také schopni kombinací vrstev vytvořit obrázky, které sice vychází ze skutečnosti zachycené satelitními senzory, avšak od reality se barevně odlišují. Metoda false color composite je klíčovou hlavně pro fotointerpretaci. [3] Různé kombinace různých spektrálních tříd totiž dokáží zvýraznit některé vlastnosti a informace zachycené na satelitním snímku a zjednodušit tak odborníkovi jeho analýzu.



Obrázek 5: Ukázka principu vytváření barevného obrázku ze tří spektrálních pásem: infračerveného pro červenou barvu, červeného pro zelenou barvu a zeleného pro modrou barvu [3]



Obrázek 6: Princip metody saturating linear contrast enhancement [3]

Dalším z problémů satelitních snímků je jejich nedostatečný kontrast, kvůli kterému většinu snímků není možné hned zobrazit. Důvodem je fakt, že satelitní senzor se od lidského oka liší a navíc vzhledem k velkému rozsahu vlnových délek zaznamenávaných jedním senzorem je pochopitelné, že povrch nevykazující žádné razantní změny pokrytí bude mít snímané hodnoty jasu jednotlivých pixelů spíše blízko u sebe. Výsledný vykreslený černobílý obrázek pak vypadá, jako by byl tvořen pouze několika stěží od sebe rozeznatelnými úrovněmi šedi. [3] Pokud by analytik nebyla žena, která má mnohem větší barevnou rozlišovací schopnost než muž, byla by fotointerpretace takových snímků velice nekvalitní a složitá.

Řešením je použití některé z metod posílení kontrastu. [3] Tyto metody dokáží upravit jasy jednotlivých pixelů tak, aby při zachování jejich informační hodnoty zlepšily možnost zpracovatelnosti lidským okem. Představme si, že v nějakém spektrálním pásmu nějakého satelitního snímku jsou zaznamenány hodnoty v rozsahu od 30 do 55. V počítačové grafice běžně používané rozsahy pro jednotlivé barvy jsou 8bitové, neboli hodnoty od 0 do 255. Výsledkem posílení kontrastu je stejná spektrální vrstva, ale s hodnotami v rozsahu 0 až 255. Nejlepší z jednodušších je *saturating linear contrast enhancement*, česky saturační lineární metoda posílení kontrastu. [3] Využívá statistických skutečností o dané vrstvě. Největší množství pixelů má hodnotu jasu v rozsahu ± 2σ (σ je symbol používaný pro směrodatnou odchylku) od střední hodnoty jasu. [3] Graf na Obrázku 6 vysvětluje princip její funkce. Osa X představuje skutečné hodnoty jasu pixelu uložené ve spektrální vrstvě. Osa Y stejným způsobem souvisí s hodnotami použitými při vykreslování této spektrální vrstvy. Hodnota B_{min} je určena jako střední hodnota jasu - 2σ a hodnota B_{max} jako střední hodnota jasu + 2σ. [3] Hodnoty jasu nad nebo pod těmito hranicemi jsou automaticky převáděny na 255, respektive 0.

Principielně vytvářejí metody posílení kontrastu překladovou tabulku používanou při zobrazování. Nazývá se *LookUp Table*, nebo LUT. [3] Indexy této tabulky jsou skutečné hodnoty jasu v dané spektrální vrstvě a hodnoty v ní obsažené jsou ty, co se vykreslují. Každá zobrazovaná spektrální vrstva má svoji vlastní LUT. Zobrazuje-li se barevný obrázek, jsou třeba tři LUT, černobílý

vyžaje pouze jednu LUT(v černobílém obrázku mají všechny barevné složky pixelu stejnou hodnotu jasu).

1.3.3 Úvod do klasifikace

Analýza satelitních snímků pomocí fotointerpretace je použitelná, pokud nás zajímají geometrické charakteristiky a pouze základní rozdělení různých typů povrchu. To je nicméně nepraktické při použití na úrovni samotných pixelů, pokud jich není jen pár významných. Výsledkem je velmi malá schopnost přesného určení hranic a rozlohy rozdílných povrchů. [3] Je to způsobeno tím, že výsledky fotointerpretace závisí na možnosti odborníka prozkoumat a zhodnotit pouze něco kolem tří spektrálních pásem z několika dostupných v satelitním snímku. [3] Ne, že by byla všechna pásma ve snímku nutně potřebná k identifikaci pixelu, avšak chceme-li k nim přihlížet a zahrnout je do rozhodovacího procesu, pak je fotointerpretace jasně nedostačující metodou. Navíc není žádný odborník schopen pouhým okem rozlišovat všechny hodnoty daného radiometrického rozlišení. Pro počítač není žádným problémem analyzovat pixel po pixelu a používat přitom celý rozsah radiometrického rozlišení ve všech spektrálních pásmech. [3]

V počítačově založené kvantitativní analýze jsou atributy každého pixelu prozkoumávány za účelem jeho pojmenování podle příslušnosti do některé z definovaných skupin. Tento proces je často nazýván *klasifikace*. [3] Toto pojmenování je prováděno počítačem na základě identifikace pixelů s podobnými spektrálními vlastnosti.

1.3.4 Multispektrální prostor a spektrální třídy

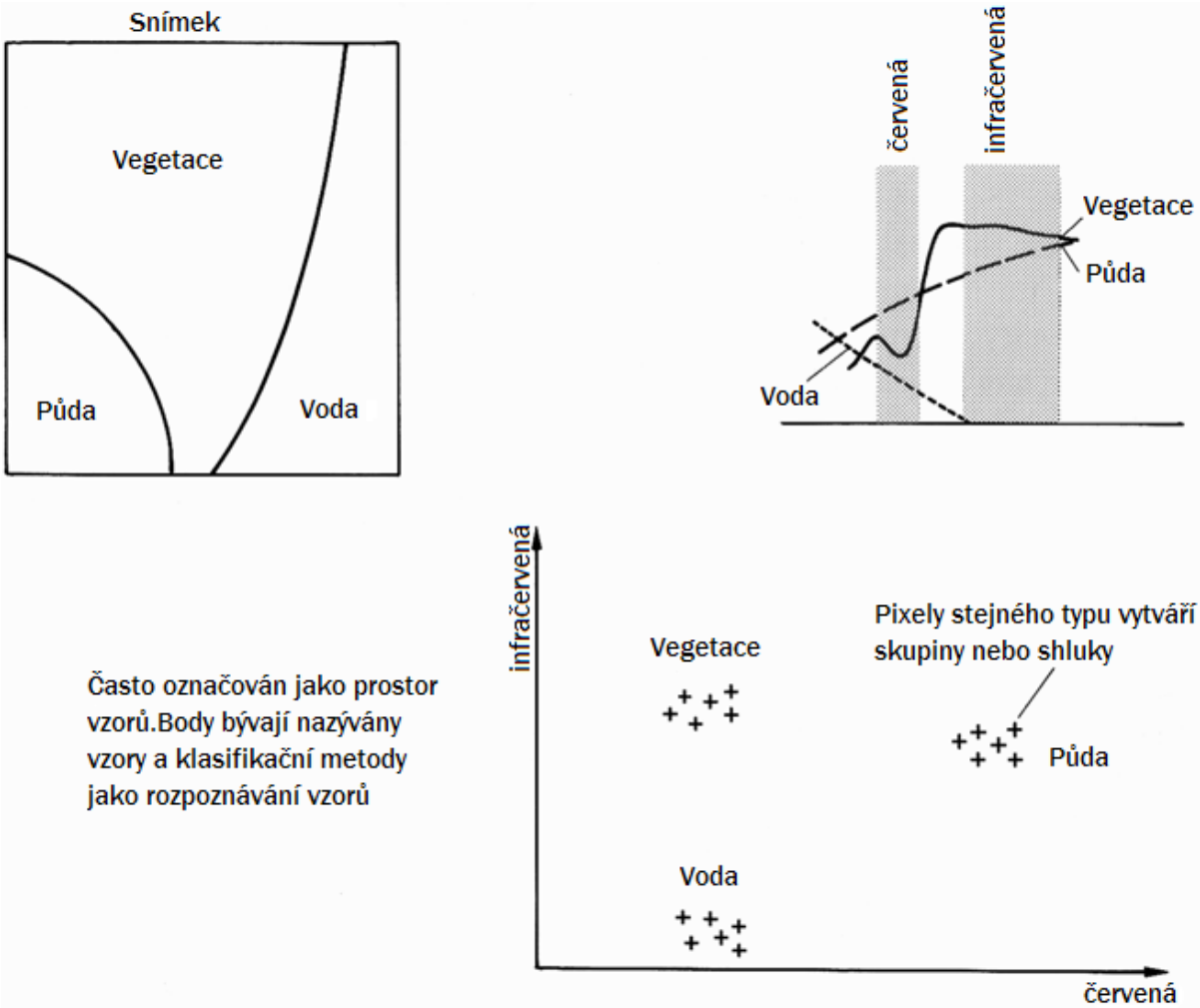
Nejefektivnější způsob, jakým mohou být reprezentována multispektrální data pro kvantitativní analýzu, je jejich zobrazení v N-rozměrném prostoru, kde N je počet spektrálních vrstev v satelitním snímku. [3] V tomto prostoru je každý pixel zobrazen jako bod, jehož souřadnice jsou hodnoty jasu tohoto pixelu v jednotlivých spektrálních pásmech. [3] Obrázek 7 ukazuje jeden takový prostor. Pro zjednodušení (a také kvůli dvourozměrnému papíru) se jedná o pouze dvourozměrný prostor vytvořený z červeného a infračerveného spektrálního pásma. Na obrázku je velmi dobře patrné, že body patřící pixelům, které popisují stejný zemský povrch, tvoří v multispektrálním prostoru shluky. Velikost a tvar těchto shluků závisí na různorodosti popisovaného povrchu, zaznamenaném rušení a topografických vlivech. [3] Tyto shluky pixelů jsou označovány jako *informační třídy*, protože to jsou ty informace, které počítač potřebuje předem znát, aby mohl správně rozpoznávat a pojmenovávat pixely. [3]

Ve skutečnosti nebývají informační třídy tak kompaktními shluky, jak je vidět na Obrázku 7. Často můžeme najít několik různých shluků popisujících půdu ve stejné oblasti nebo několik různých shluků popisujících stejný les a to samé platí i pro další povrchy. Je to způsobeno nejen rozdílným typem povrchu (jiný druh vegetace, atd.), ale také rozdíly v množství vlhkosti, složením půdy, na

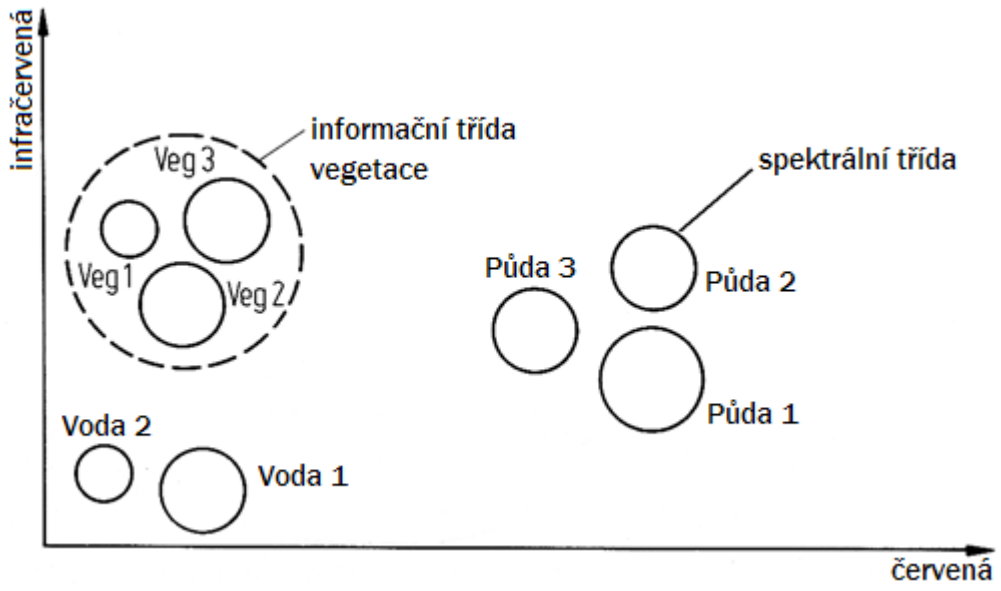
které roste daná vegetace, a topografickými vlivy. V důsledku toho vypadá multispektrální prostor mnohem více jako na Obrázku 8, kde je každá informační třída složena z několika *spektrálních tříd*. [3]

V mnoha případech netvoří ty informační třídy, které obvykle uživatele zajímají, takto zřetelné shluky nebo skupiny shluků, ale jsou neoddělitelně namíchány v multispektrálním prostoru. Stává se to například pokud satelitní senzory snímají povrch, kde se vyskytuje různě hustá vegetace, takže satelit snímá různé kombinace baldachýnu a povrchu pod ním. [3] Informační třídy v tomto případě odpovídají spíše procentuálnímu poměru obou povrchu, než dobře odděleným a ohraničeným třídám jako na Obrázku 8. V tomto případě je tedy velice důležité správně definovat sadu spektrálních tříd, která efektivně popisuje žádanou informační třídu. [3]

Při kvantitativní analýze jsou to právě spektrální třídy, které bude počítač používat, jelikož popisují přirozený způsob shlukování pixelů v satelitním snímku. [3] Po provedení kvantitativní analýzy může analytik roztřídit vytvořené spektrální třídy do správných informačních tříd.



Obrázek 7: Dvourozměrný multispektrální prostor a jeho vztah k charakteristice odrazivosti běžných materiálů [3]

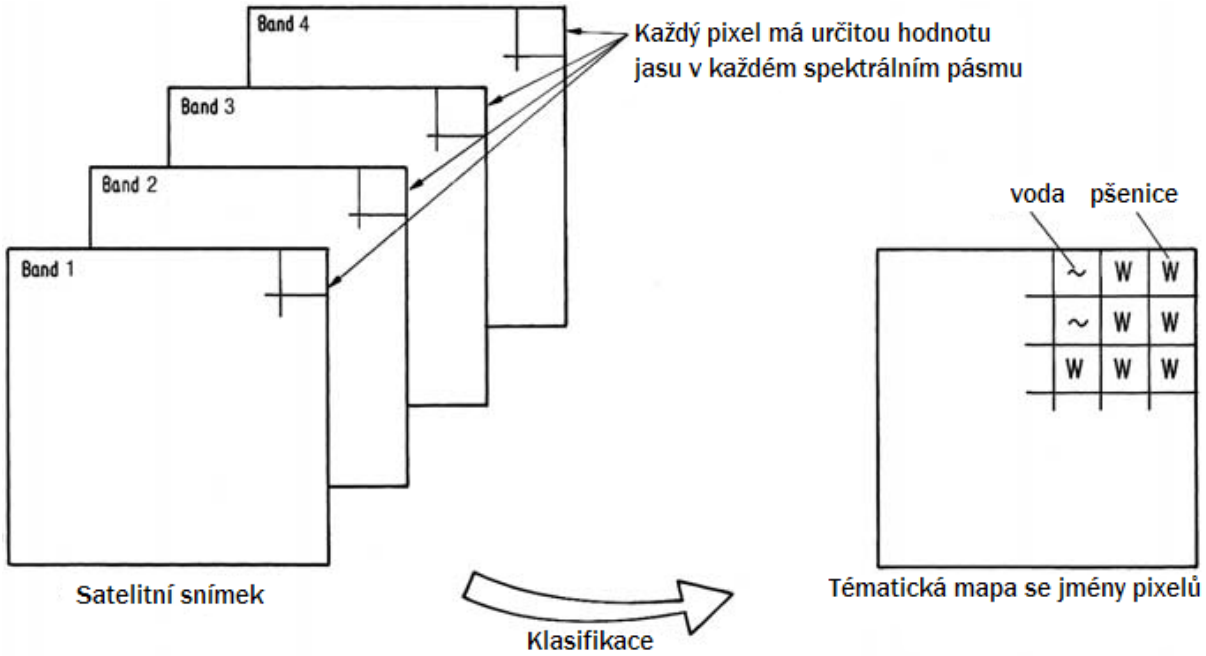


Obrázek 8: Informační třída jako skupina spektrálních tříd [3]

1.3.5 Principy klasifikace

Skutečnosti, že data v satelitním snímku lze rozdělit do několika skupin spektrálních tříd a že tyto třídy přímo souvisí s typem povrchu zachyceného na snímku, se využívá v technikách matematického rozeznávání vzorů, klasifikaci vzorů a v jejich moderních variantách se strojovým učením. Vzory jsou samotné pixely, nebo přesněji vektory obsahující hodnoty jasu daného pixelu. [3]

Klasifikace zahrnuje pojmenování pixelů podle jejich příslušnosti do určité spektrální a tím pádem i informační třídy. [3]



Obrázek 9: Význam klasifikace při pojmenovávání pixelů v satelitním snímku [3]

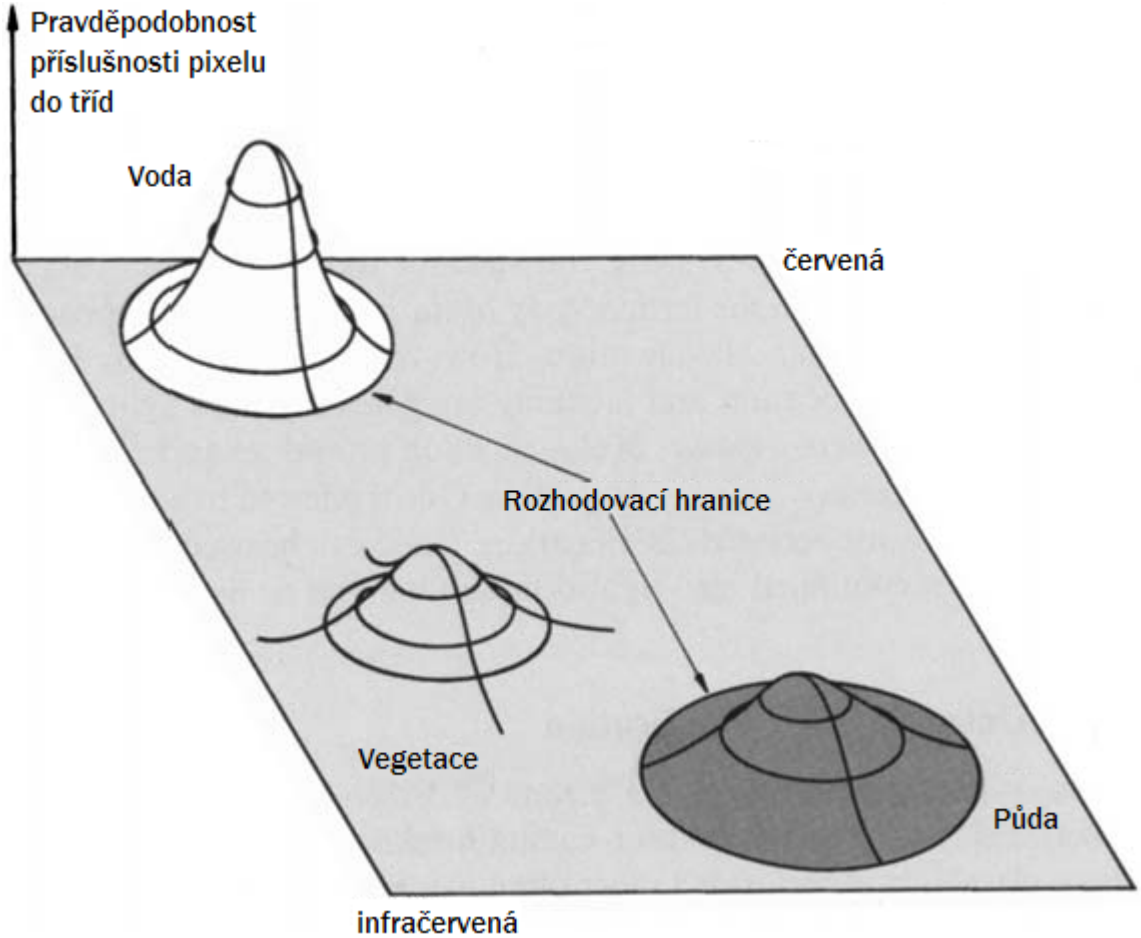
Existují dvě velké skupiny klasifikačních metod a obě nachází své uplatnění při analýze remote sensing dat. Jedna z nich je unsupervised klasifikace a druhá supervised klasifikace. V praxi se často používají různě kombinované hybridní metody. [3]

1.3.6 Supervised a unsupervised metody

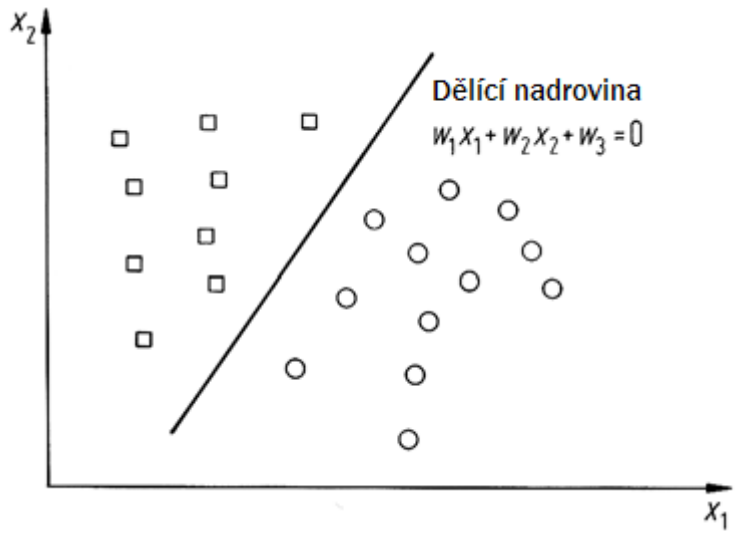
1.3.6.1 Supervised klasifikace

Supervised klasifikace popisuje způsob, jakým lze roztřídit pixely v satelitním snímku do spektrálních tříd na základě předem vytvořených rozhodovacích pravidel nebo parametrů. [3] Existuje několik různých metod používaných pro supervised klasifikaci. První a nejstarší skupinou metod jsou *statistické metody*. [3] Ty pracují s předpokladem, že každá spektrální třída se dá vyjádřit pomocí normálního rozložení v multispektrálním prostoru, jak ukazuje Obrázek 10. [3] Pixel je poté pojmenován podle třídy, k jejíž příslušnosti má největší pravděpodobnost. Na Obrázku 10 je *rozhodovací hranice* hranicí, za kterou má pixel stejnou pravděpodobnost příslušnosti ke dvěma spektrálním třídám. [3]

Další skupinou jsou nestatistické, *geometrické metody*. [3] Ty naopak pracují se skutečností, že



Obrázek 10: Dvourozměrný multispektrální prostor se spektrálními třídami reprezentovanými normálním rozložením [3]



Obrázek 11: Dvourozměrný spektrální prostor se dvěma skupinami pixelů rozdělenými lineární hranicí [3]

mohou od sebe být spektrální třídy v multispektrálním prostoru odděleny hranicemi, které se nazývají *nadroviny*. [3] Díky nim je pak možné určit, uvnitř kterých hranic daný pixel leží, a tím pádem i která spektrální třída ho pojmenuje. Obrázek 11 ukazuje jednoduchý multispektrální prostor se dvěma spektrálními třídami oddělenými lineární hranicí.

Obecně má supervised klasifikace tři hlavní kroky. V prvním kroku je pro každou spektrální třídu vybráno několik známých, pojmenovaných pixelů. Tato skupina pixelů se nazývá *training set*, česky trénovací sada. [3] K tomu lze použít pozemní průzkumy, letecké fotografie, topografické mapy, atd. Ve druhém kroku se z trénovací sady vytvoří sada pravidel, parametrů nebo rovnic (v závislosti na používané metodě) pro každou popisovanou spektrální třídu. [3] Tím končí fáze učení. Třetí krok je klasifikační fáze. Spočívá v pojmenování zbylých pixelů za použití dříve vytvořených pravidel nebo parametrů. [3]

1.3.6.2 Unsupervised klasifikace

Naproti tomu unsupervised klasifikace popisuje způsob, jakým jsou pixely v satelitním snímku přiřazeny do spektrálních tříd bez předchozí znalosti o jejich existenci nebo umístění v multispektrálním prostoru. [3] Metody unsupervised klasifikace mohou být použity pro zjištění počtu a umístění spektrálních tříd, do kterých se shlukují pixely satelitního snímku a pro určení do které z těchto tříd daný pixel patří. K tomu se většinou používá metoda *clustering,* česky shlukování. [3] Analytik může poté jednoduše identifikovat vzniklé spektrální třídy pomocí předem pojmenovaných pixelů (například z leteckého nebo pozemního průzkumu). Clustering je obecně výpočetně náročný proces, přesto je velmi důležitým při remote sensing analýze. Jelikož zatímco informační třídy zkoumaného snímku bývají známy, odborník často nemá žádnou představu o spektrálních třídách, které jsou v informačních třídách obsaženy. [3]

2 Implementace

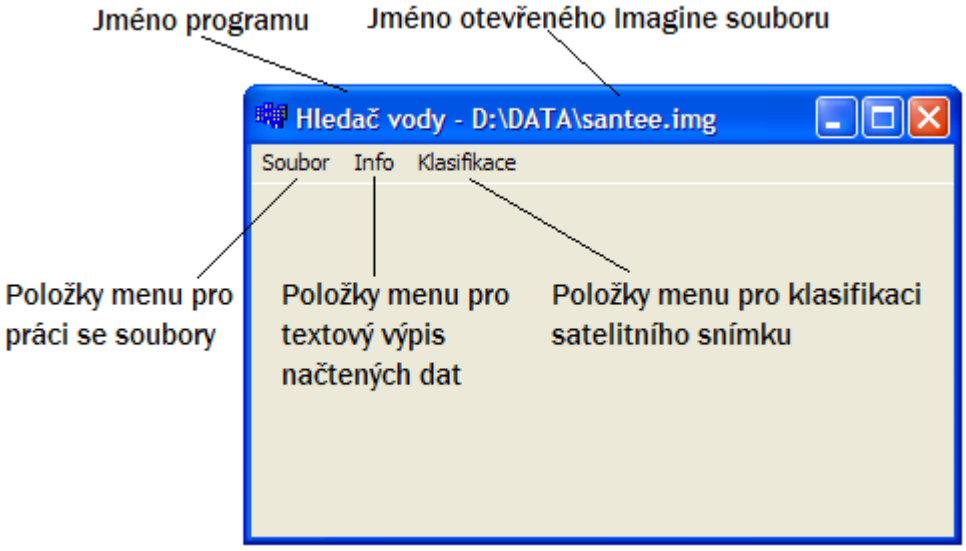
Tato kapitola popisuje podobu programové části bakalářské práce, jednoduše vysvětluje použití a účel programu a snaží se čtenáři přiblížit postup, překážky a jejich odstranění při tvorbě programu.

Celý program jsem napsal v jazyce C++ v prostředí Borland C++ Builder. Je to velice kvalitní nástroj a díky integrovaným prefabrikovaným stavebním dílům (tzv. VCL komponentám [4]) je vcelku jednoduché stavět základní dialogové boxy používané v programu. Bohužel je Borland C++ Builder vývojové prostředí pro aplikace v prostředí MS Windows, proto je celý můj program také určen pouze pro MS Windows.

Jako autor uznávám, že existuje mnoho částí programu, které by šlo napsat nebo navrhnout jinak s výsledkem stejného a v některých případech i lepšího výkonu. Program jako celek je brán jako část většího skupiny nástrojů pro analýzu satelitních snímků.

2.1 Hlavní menu

Hlavní menu funguje jako vstupní bod programu. Zobrazí se hned po startu programu přes celou horní část obrazovky. To z toho důvodu, aby menu vypadalo na většině rozlišení stejně. V popisku okna je název programu a v případě, že program otevřel nějaký IMAGINE soubor, je zde uveden i jeho celý název. Program jsem nazval poeticky **Hledač vody**, jelikož je to podle zadání jeho primární funkce. Pod popiskem okna je standardně umístěno menu, jak známe z většiny jiných programů. Nejinak je tomu i tady. Menu má tři základní skupiny položek. Obvyklý **Soubor** a pak méně obvyklejší **Info** a **Klasifikace**. Prázdný prostor pod menu byl původně určen pro grafická tlačítka reprezentující položky v menu, ale nakonec jsem se k nim nedostal.



Obrázek 12: Hlavní menu programu

Skupina položek **Soubor** obsahuje základní příkazy pro otevření IMAGINE souboru se satelitním snímkem, jeho zavření a ukončení celého programu. Po startu programu je jedinou aktivní skupinou položek. Ostatní skupiny nelze použít, dokud není otevřen nějaký IMAGINE soubor. Při výběru položky **Otevřít** se objeví `OpenDialog` [4] pro přehledný výběr žádaného souboru. Program se poté snaží otevřít a zpracovat tento soubor a načíst potřebná data do paměti. Provede-li se načítání souboru úspěšně, zobrazí se několik oken s grafickým náhledem satelitních dat (více v kapitole 2.2.2.2) a zpřístupní se ostatní skupiny položek, v opačném případě o tom program uživatele informuje výpisem chyby. Je-li otevřen nějaký IMAGINE soubor, položka **Zavřít** vymaže z paměti všechna data o otevřeném souboru a opět zablokuje ostatní skupiny položek. Program navíc umí pracovat naráz pouze s jedním otevřeným souborem, proto je třeba před otevřením nového IMAGINE souboru ten starý zavřít. Položka **Konec** provede to samé jako položka **Zavřít** a navíc ukončí celý program.

Skupina položek **Info** obsahuje pouze jedinou položku **Ukaž**. Je přístupná pouze pokud je otevřen nějaký IMAGINE soubor. Položka **Ukaž** otevře nové okno s textovým náhledem satelitních dat (více v kapitole 2.2.2.1). Původně sloužila jen jako součást ladících informací při vytváření knihovny pro práci s IMAGINE soubory (kapitola 2.2.1), ale nakonec zůstala zachována i v konečné verzi.

Poslední skupina, **Klasifikace**, nabízí uživateli možnost analýzy otevřeného a načteného satelitního snímku. Stejně jako skupina **Info** je také přístupná pouze pokud je otevřen nějaký soubor. Položka **Klasifikace** otevře další okno s nastavením parametrů klasifikace (více v kapitole 2.3.2). Položka **Zruš rozdělení** zruší všechny výsledky předchozích klasifikací a tím připraví satelitní data na další klasifikaci.

Programově je hlavní menu tvořeno třídou `TFrm_MainForm`. Ta krom výše vypsaných prvků ještě obsahuje ukazatel na třídu `Bands` spravující načtená satelitní data v paměti (kapitola 2.2.1), ukazatele na `TFrm_Viewer` – okna s grafickým náhledem satelitních dat (kapitola 2.2.2.2), ukazatel na třídu `ImgClasses` starající se o výsledná data klasifikace (kapitola 2.3.1) a ukazatel na třídu `ErrorReport` používanou pro zobrazení chyby uživateli (kapitola 2.2.1).

Jedním z hlavních problému, na které jsem při implementaci hlavního menu narazil, byl problém přístupnosti. Stávalo se, že vlivem zapouzdření se kvůli nedostatečně propracovanému objektově orientovanému návrhu objevila situace, kdy určitá funkce v jedné třídě neměla přístup k nějaké proměnné, která se většinou nacházela v jiné třídě. V tu chvíli bylo třeba stávající koncepci mírně pozměnit a umožnit tak přístup funkci k žádané proměnné. Řekl bych, že všechny tyto problémy se mi podařilo řešit celkem elegantně a téměř ihned.

2.2 Zpracování satelitních snímků

Podkapitola čtenáři popisuje postup při zpracování souborů se satelitními snímky jejich načtením počínaje přes několik různých druhů zobrazení načtených dat, snahy o zvyšování výkonu aplikace a popisem dekompresní metody ESRI GRID [5] konče.

2.2.1 Knihovna pro práci s IMAGINE soubory

Existuje velké množství různých formátů používaných pro uložení satelitních dat. Jedním z nejpoužívanějších a nejlépe dokumentovaných je formát IMAGINE vytvořený firmou Leica Geosystems GIS & Mapping, LLC. [5] Ten jsem si také vybral pro svůj program. Obecně platí, že sehnat satelitní snímky je velice obtížné, protože satelit přeci jen nemá každý. Nakonec se mi na Internetu podařilo objevit pár IMAGINE souborů a tak mohla moje práce započít.

Mým úkolem bylo vytvořit knihovnu v jazyce C++, která by dokázala nějakým způsobem načíst a zpracovat IMAGINE soubor a data v něm obsažená (alespoň ta, která potřebuji pro klasifikaci a se kterými budu dál pracovat) uložit do paměti. Vycházel jsem ze skvěle zpracované dokumentace popisující IMAGINE formát, která je součástí dokumentace ERDAS IMAGINE. [5] Struktura IMAGINE souboru tak, jak jsem ji potřeboval a použil já, je uvedena v Příloze 1.

Vytvořená knihovna se skládá ze tří tříd. První třída `ErrorReport` je pomocná třída používaná pro výpis krátkého textu chyby vzniklé při provádění některé z funkcí v knihovně. Je pravidlem, že funkce, jejichž běh může postihnout nějaká chyba, tuto třídu vrací a procedura nebo funkce, která volala „hlídanou“ funkci, pak může na základě této skutečnosti jednoduše zareagovat například výpisem chyby, skončením provádění nějaké činnosti, změnou chování programu, atd.

Druhá třída `Bands` je hlavní třída zprostředkující načítání hodnot pixelů a spektrálních vrstev ze souboru a uložení těchto údajů do paměti. Obsahuje pole záznamů s informacemi o spektrálních vrstvách. Každý záznam obsahuje tabulku s hodnotami pixelů ve vrstvě, jméno vrstvy, údaje o šířce a výšce vrstvy, statistické hodnoty jako střední hodnota, směrodatná odchylka, maximální a minimální hodnota pixelu ve vrstvě a také údaje potřebné pro správně zpracování vrstvy: šířka a výška jednoho bloku, počet bloků v řádku, počet bloků ve vrstvě a datový typ pixelu.

Metody třídy `Bands` pracují s třídou `ErrorReport`, takže každá chyba při načítání je správně ošetřena a program na tuto skutečnost může reagovat. Pomineme-li standardní metody, které se vyskytují v téměř každé třídě, jako je konstruktor, destruktor, případně funkce utvářející rozhraní třídy (umožňují přístup k hodnotám proměnných uvnitř třídy z prostoru mimo tuto třídu ať už za účelem jejich čtení nebo změny), je zde hlavní metoda `LoadBand`.

Metoda `LoadBand` je nejpodstatnější funkcí celé třídy. Bývá volána, když metoda třídy `ImgFile` (viz níže) narazí při načítání .img (přípona IMAGINE souborů) souboru na uzel ještě nezpracované spektrální vrstvy. Metoda převezme otevřený soubor od třídy `ImgFile` a pokračuje

v jeho zpracovávání. Vytvoří nový záznam o vrstvě a uloží do něj základní údaje (šířka, výška, atd.) Poté postupně prochází soubor a hledá jeden ze dvou hledaných uzlů, `Edms_State` (Příloha 1) a `Esta_Statistics` (Příloha 1). První z uzlů obsahuje hodnoty pixelů dané vrstvy, metoda zjistí počet bloků a datový typ uložených pixelů, alokuje dostatečný prostor, pixely načte do tabulky a tabulku uloží do záznamu. Druhý uzel obsahuje statistické hodnoty, metoda je načte a uloží do připraveného záznamu. Po celou dobu kontroluje stav souboru a stane-li se, že soubor najednou nečekaně končí, nebo že data nejsou konzistentní, okamžitě ukončí načítání a skončí s chybou, kterou ve formě objektu třídy `ErrorReport` předá třídě `ImgFile`. Nakonec pak vrátí řízení zpět třídě `ImgFile`, která pokračuje ve zpracovávání souboru.

Poslední třída `ImgFile` zajišťuje otevření na pevném disku uloženého .img souboru a práci s tímto otevřeným souborem. Obsahuje celý název otevíraného souboru (ten se poté vypíše v popisku okna Hlavního menu – kapitola 2.1), číselný identifikátor otevřeného souboru používaný funkcemi pro přístup k datům souboru, ukazatel na strukturu `TEhfaFile`, která shrnuje hlavní informace o otevřeném souboru z pohledu IMAGINE formátu a ukazatel na strukturu ukládající kořenový uzel celé stromové struktury otevřeného souboru.

Metody třídy `ImgFile` se zaměřují na práci s IMAGINE soubory. Metoda `AssignFile`, která se používá pro přiřazení uživatelem nově vybraného souboru, spolu s metodou `FileName`, sloužící ke zjištění přiřazeného jména souboru, zprostředkují rozhraní třídy. Krom těchto funkcí a destruktoru, má třída `ImgFile` jednu hlavní metodu `OpenFile`.

Metoda `OpenFile` otevírá .img soubor, jehož jméno je ve třídě předem definováno metodou `AssignFile`. Je první funkcí, která se spustí při načítání IMAGINE souboru do paměti. Jejím parametrem je ukazatel na objekt třídy `Bands`, který bude obsahovat data ze souboru. Pokusí se otevřít zadaný soubor a uspěje-li, poznamená si číselný identifikátor otevřeného souboru a začne postupně načítat data z tohoto souboru. Metoda je schopná rozpoznat, zda má otevíraný soubor strukturu odpovídající IMAGINE formátu. Pokud struktura neodpovídá, soubor nejde otevřít nebo najednou nečekaně končí, funkce zareaguje vytvořením objektu třídy `ErrorReport` s příslušným hlášením o vzniklé chybě. Je-li vše v pořádku, funkce vytvoří nový záznam struktury souboru a načte do něj informace ze souboru. V této struktuře je uložena informace o pozici kořenového uzlu v souboru. Ten se zkopíruje do nově vytvořeného záznamu připraveného v rámci třídy `ImgFile`. Se znalostí kořenového uzlu můžeme procházet celým souborem a hledat mezi potomky kořenového uzlu výskyty uzlu `Eimg_Layer` (Příloha 1) obsahujícího všechny informace jedné spektrální vrstvy. V tu chvíli funkce volá metodu `LoadBand` třídy `Bands` a předá jí všechny důležité parametry potřebné pro zpracování spektrální vrstvy. Poté, co metoda `LoadBand` vrátí řízení, funkce pokračuje v prohledávání potomků kořenového uzlu a hledání dalších uzlů `Eimg_Layer`. Po prohledání všech potomků kořenového uzlu funkce končí a vrací `NULL`, jelikož během

vykonávání funkce nedošlo k žádné chybě. Objekt třídy `Bands`, jehož ukazatel byl použit jako parametr funkce v tuto chvíli obsahuje všechny spektrální vrstvy vyskytující se v IMAGINE souboru.

Knihovna také obsahuje množství různých podpůrných struktur. Struktury, jejichž názvy začínají `TE`, například `TEfhaFile`, jsou struktury používané při načítání dat z .img souboru – shodují se s vnitřní strukturou IMAGINE souboru. Ostatní struktury slouží pro uložení načtených satelitních dat do paměti.

2.2.2 Zobrazování načtených dat

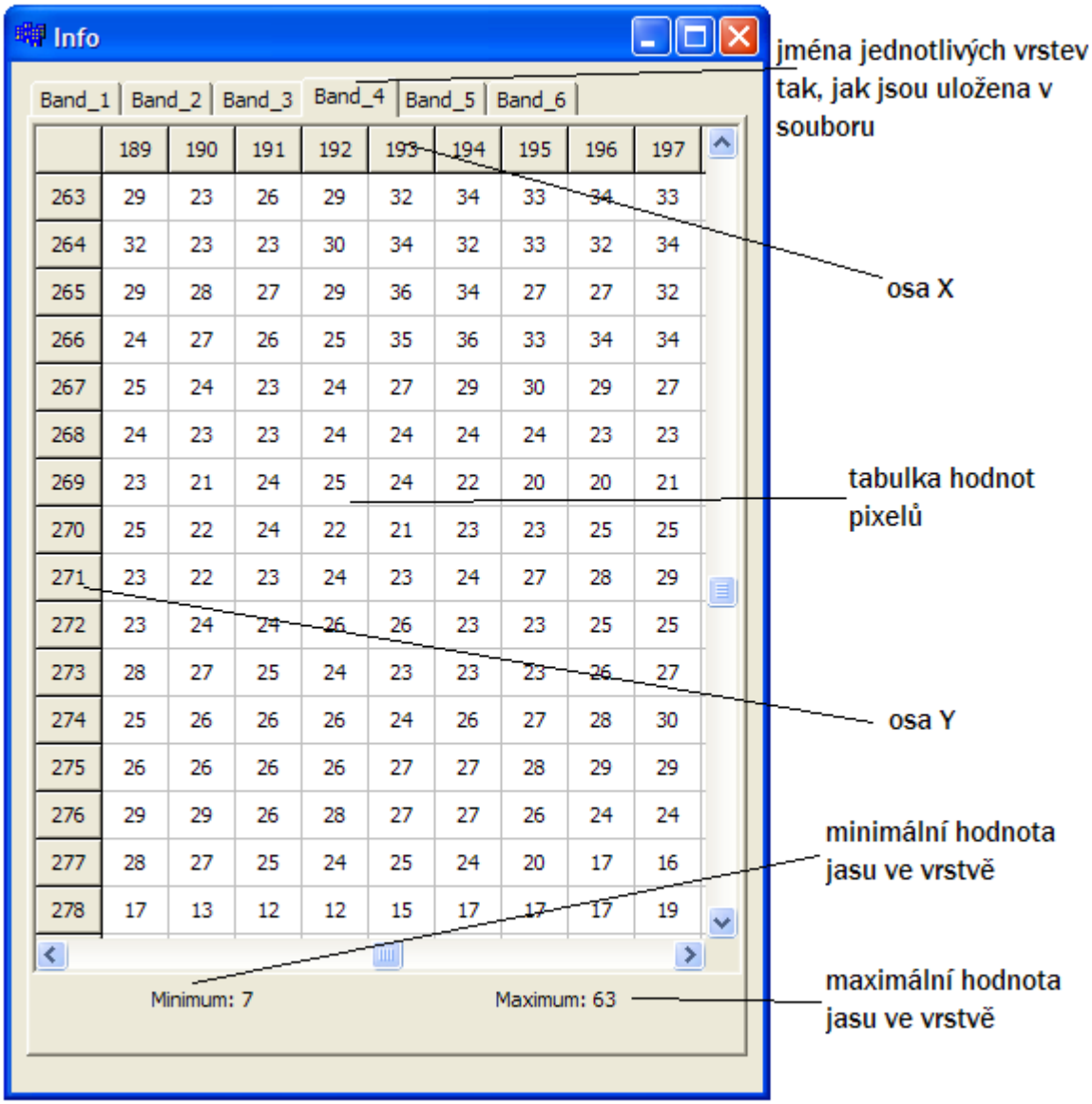
Po vytvoření první funkční podoby knihovny pro práci s IMAGINE soubory (kapitola 2.2.1) vyvstala logická potřeba hodnoty pixelů načtených v paměti nějakým způsobem reprezentovat uživateli. Následující dvě podkapitoly jsou zaměřeny na popis vývoje nástrojů schopných vhodně zobrazovat data načtená v paměti.

2.2.2.1 Textový náhled načtených satelitních dat

Textový náhled původně sloužil pouze pro potřeby implementace a zlepšování knihovny pro práci s IMAGINE soubory, ale postupem času se z něj stal plnohodnotný nástroj. Na rozdíl od grafického náhledu, který používá techniky posílení kontrastu (kapitola 1.3.3) a tím upravuje hodnoty pixelů, textový náhled ukazuje uživateli skutečně načtené hodnoty. Celé okno má velmi účelový design. Hlavním prvkem je tabulka se samotnými hodnotami pixelů. Sloupce tabulky korespondují s osou X a řádky s osou Y. Tak je zachována standardní orientace. Vzhledem ke skutečnosti, že satelitní snímky jsou multispektrálními daty, bylo zapotřebí tuto vlastnost zakomponovat i do okna textového výpisu. Slouží k tomu záložky pod popiskem okna. Každá záložka zpřístupňuje jednu spektrální vrstvu. Její název je shodný s názvem vrstvy, jak je pojmenována v IMAGINE souboru. Pod tabulkou s hodnotami pixelů je zvlášť vypsána maximální a minimální hodnota jasu vyskytující se v dané spektrální vrstvě.

Programově je okno náhledu tvořeno třídou `TFrm_InForm`. Centrální tabulka s výpisem hodnot pixelů je objekt třídy `TStringGrid` [4], záložky tvoří třída `TTabControl` [4] a dvojice komponent vypisujících maximální a minimální hodnotu, jsou objekty třídy `TLabel` [4]. Vzhledem k tomu, že objekt třídy `TTabControl` ve skutečnosti nemění celou „svoji“ plochu, bylo třeba naprogramovat změnu hodnot zobrazených v objektu třídy `TStringGrid`. To se ukázalo jako velmi neefektivní, jelikož při každé změně vybrané spektrální vrstvy byl program nucen znovu vyplnit celou tabulku daty z paměti a vykreslit ji.

Řešením tohoto problému bylo vytvoření hned několika objektů třídy `TStringGrid` (jeden pro každou načtenou spektrální vrstvu) uložených do připravené tabulky. Tak vznikne několik stejných tabulek položených na sebe. Hodnoty pixelů jsou do jednotlivých tabulek ukládány ještě před zobrazením celého okna, poté jsou všechny tabulky skryty a vidět zůstane pouze tabulka



Obrázek 13: Okno s textovým náhledem načtených dat

zobrazující hodnoty pixelů první spektrální vrstvy. Změna záložky poté nemusí provádět znovunačtení dat z paměti, ale pouze skryje zobrazenou tabulku a zobrazí tabulku obsahující hodnoty pixelů té vrstvy, jejíž záložka byla vybrána. Tím se ušetří velké množství času v porovnání s předchozí variantou řešení.

Objekty používané pro zobrazování minimální a maximální hodnoty jasu vybrané spektrální vrstvy reagují na změnu svého obsahu velice rychle a proto nebylo třeba původní koncept jejich použití nijak měnit.

2.2.2.2 Grafický náhled načtených satelitních dat

Na rozdíl od textového náhledu byl nástroj umožňující grafický náhled cíleně vytvářen pro zobrazování načtených satelitních dat uživateli. Dokáže zobrazovat jednotlivé spektrální vrstvy i

barevný false color composite obrázek a používá metodu saturating linear contrast enhancement pro posílení kontrastu. Dalo by se říci, že grafický náhled je nejdůležitějším oknem vyskytujícím se v programu.

Po otevření IMAGINE souboru se satelitním snímkem a jeho načtení do paměti se vykreslí jeden černobílý náhled každé spektrální vrstvy v satelitním snímku a je-li dostatečný počet spektrálních vrstev (tři a více), zobrazí se i barevný false color composite (kapitola 1.3.3) náhled. Barevný náhled se podobně jako hlavní menu roztáhne přes celou šířku obrazovky, zatímco všechny černobílé náhledy jsou defaultně minimalizovány.

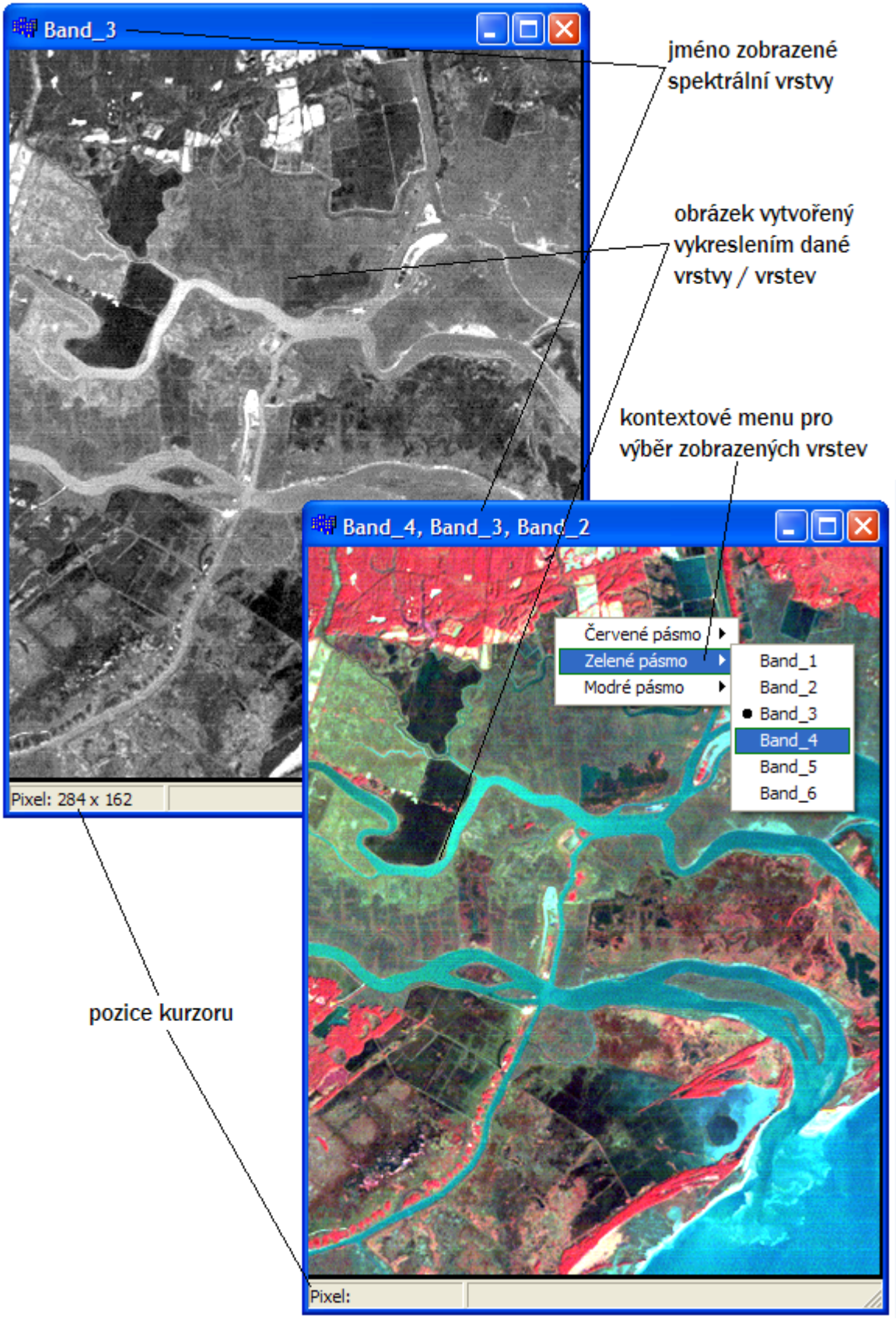
Na Obrázku 14 jsou zobrazeny obě varianty grafického náhledu. V popisku každého okna je název zobrazené spektrální vrstvy. V barevném náhledu jsou seřazeny v pořadí červené spektrum, zelené spektrum a modré spektrum. Hlavní částí okna je vlastní obrázek dané spektrální vrstvy položený na černém pozadí. Velikost celého okna lze libovolně měnit. Pokud je velikost zobrazované plochy menší než velikost obrázku, vytvoří se vertikální a horizontální posuvná lišta umožňující prohlížení celé zobrazené vrstvy. Ve spodní části okna náhledu se nachází stavový řádek. Pohybuje-li uživatel kurzorem myši po ploše zobrazené spektrální vrstvy, v levé části stavového řádku se vypisují souřadnice ve formátu X : Y určující pixel, na kterém se nachází kurzor myši. Za povšimnutí stojí kontextové menu v barevném náhledu. Umožňuje různé kombinace spektrálních vrstev za účelem odhalení co největšího množství detailů. Černobílý náhled kontextové menu nemá.

Programově je okno náhledu tvořenou třídou `TFrm_Viewer`. Hlavní plocha okna umožňující grafické zobrazení dat je objekt třídy `TImage` [4]. Ten je spojen s objektem třídy `TScrollBox` [4] zajišťujícím korektní vytváření a rušení posuvných lišt v závislosti na potřebě. Kontextové menu zprostředkovává třída `TPopupMenu` [4] a stavový řádek vespod okna je objektem třídy `TStatusBar` [4]. Třída `TFrm_Viewer` dále obsahuje tři ukazatele na strukturu `LUT` (pro každou spektrální vrstvu jeden), což je struktura LookUp tabulky použité při vykreslení dané spektrální třídy.

Třída používaná pro černobílý náhled je ve skutečnosti shodná s třídou pro náhled barevný. Každé vytvořené okno je schopno vykreslit tři spektrální třídy (jako červenou, zelenou a modrou složku výsledné barvy pixelu) a má údaj o tom, jaké vrstvy to jsou. Jsou-li všechny tři zobrazované vrstvy různé, jedná se o barevný false color composite obrázek. Jsou-li stejné, výsledný obrázek bude černobílý. Skutečnost, zda se jedná o černobílý nebo barevný náhled je také zaznamenána.

Mimo obvyklých metod (konstruktoru, destruktoru, metod pro vytváření rozhraní) má třída `TFrm_Viewer` několik hlavních metod. Jsou to metody `CreateLUT`, `LookUp`, `MixColor` a `BandPaint`.

Metoda `MixColor` provádí jednoduché přetypování. Třída `TImage` používá pro popis barvy jednoho pixelu datový typ `TColor` [4]. Hodnota tohoto typu je 4bajtová. Nejvyšší bajt není v tuto chvíli důležitý, ale dalšítři bajty představují hodnoty jasu každé ze složek barvy v pořadí



Obrázek 14: Černobílý grafický náhled na jednu spektrální vrstvu a barevný false color composite obrázek

modrá, zelená, červená. Tři zobrazované spektrální vrstvy – tři jednobajtové hodnoty. Samotná funkce tedy provádí jednoduché bitové posuny a sčítání.

Metoda `CreateLUT` zajišťuje vytvoření LookUp tabulky používané při posílení kontrastu zobrazované spektrální vrstvy. Pokud se jedná o barvený náhled, je metoda `CreateLUT` volána pro každou z vybraných spektrálních vrstev. Funkce používá metodu saturating linear contrast enhancement (kapitola 1.3.3). V případě, že datový typ použitý pro uložení hodnot jasu pixelů dané vrstvy v paměti, je celočíselný, spočítá funkce hraniční hodnoty `BrkDown` a `BrkUp` a vytvoří tabulku později používanou při vykreslování metodou `LookUp`. V případě datového typu v plovoucí řádové čárce funkce pouze vypočítá hraniční hodnoty `BrkDown` a `BrkUp`.

Metoda `LookUp` je využívána pro zjišťování nových hodnot jasu při vykreslování spektrální vrstvy. Spolupracuje se strukturou LUT dané vrstvy. Existuje ve dvou variantách. První varianta je používána v případě, že jsou hodnoty jasu pixelů dané spektrální vrstvy celočíselného datového typu. V tom případě metoda zkontroluje, zda není hodnota mimo rozsah daný hraničními hodnotami `BrkDown` a `BrkUp` a v případě, že je, vrátí určenou hodnotu (0 nebo 255). Pokud je hodnota v určeném intervalu, použije se jako index do vytvořené tabulky a metoda vrátí hodnotu nalezenou v tabulce na této pozici. Druhou variantu program použije, pokud jsou hodnoty uloženy v datovém typu plovoucí řádové čárky. Potom metoda nepřistupuje k tabulce, jelikož žádná neexistuje a po zkontrolování správnosti rozsahů `BrkDown` a `BrkUp` výslednou hodnotu spočítá a vrátí. Původně existovala pouze jedna varianta, ale vyhledávání hodnot ve vytvořené tabulce šetří čas na rozdíl od jejich počítání.

Metoda `BandPaint` je hlavní funkcí celé třídy. Zařizuje samotné vykreslování spektrálních vrstev do objektu třídy `TImage`. Je poprvé volána v konstruktoru třídy `TFrm_Viewer`, který musí specifikovat, které tři spektrální vrstvy se budou zobrazovat. Barevný náhled tuto metodu navíc používá při každé změně vybraných spektrálních vrstev. Na začátku si funkce vytvoří metodou `CreateLUT` LookUp tabulky pro zobrazované vrstvy, změní rozměry objektu třídy `TImage` a zapíše název zobrazované vrstvy (zobrazovaných vrstev) do popisku okna. Poté prochází všechny pixely daných vrstev, upravuje jejich hodnotu metodou `LookUp` a z výsledku vytvoří jednu hodnotu typu `TColor` funkcí `MixColor`. Tuto hodnotu posléze přiřadí pixelu na vybraných souřadnicích. Po projití všech pixelů metoda končí a vykreslovaný snímek se zobrazí.

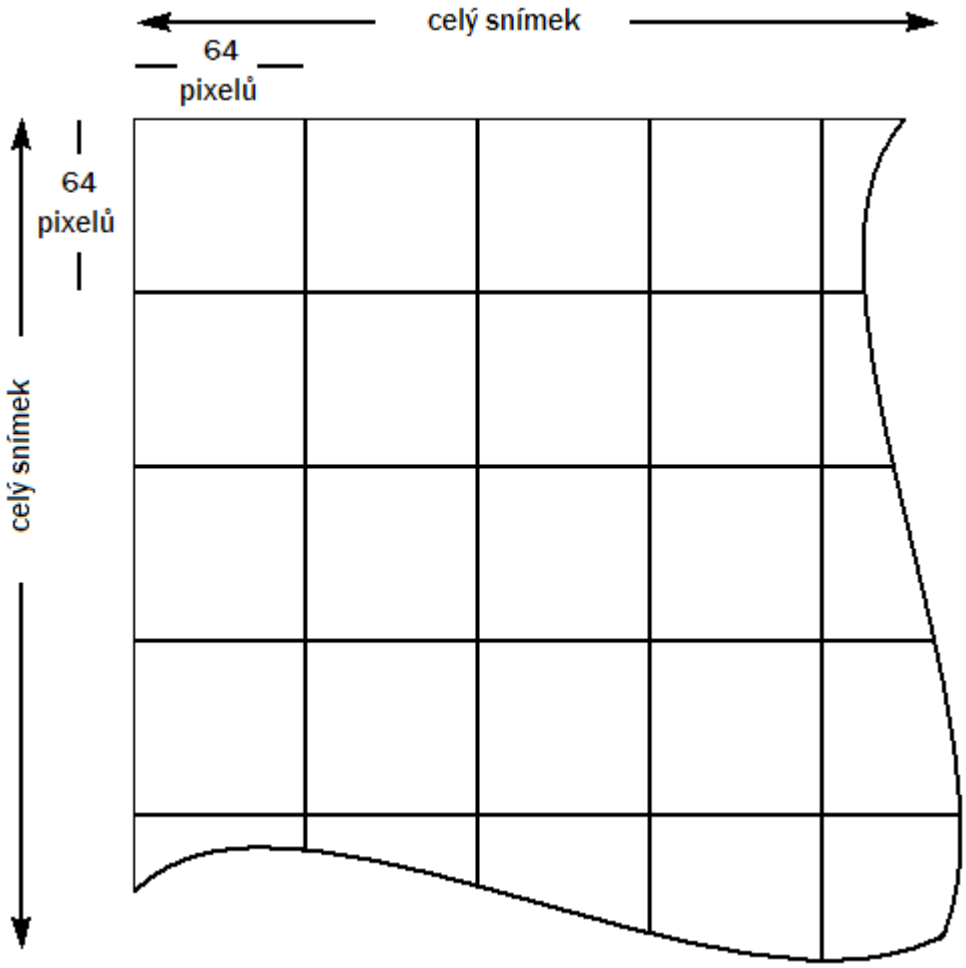
Jedním z problému, které jsem musel při vytváření nástroje pro grafický náhled řešit, byl problém s posuvnými lištami, které by ošetřily skutečnost, že okno je menší, než vykreslený obrázek. V první verzi totiž lišty nebyly. Snažil jsem se problém řešit provázáním objektu třídy `TImage` s objekty jiných vhodných tříd, například `TScrollBar` [4], ale žádná z variant nepřinášela kýžený výsledek. Až objevení, zprovoznění a propojení objektu třídy `TImage` s objektem třídy `TScrollBox` umožnilo požadovanou a korektní práci s posuvnými lištami.

První verze barevného grafického náhledu navíc neuměla měnit zobrazované spektrální vrstvy. Byly vybrány tři spektrální vrstvy (4,3,2, v případě, že satelitní snímek měl jen 3 vrstvy, tak 3,2,1) napevno. Úprava v podobě přidání kontextového menu však na sebe nenechala dlouho čekat. Možnost přepínat zobrazené spektrální vrstvy tak například umožňuje používat barevný grafický náhled při fotointerpretaci (kapitola 1.3.2).

2.2.3 Snaha o zvyšování výkonu

Při prvních testech nového nástroje grafického náhledu jsem byl velmi nemile překvapen rychlostí celého načítání a zobrazení. S růstem velikosti plochy zaznamenané v satelitním snímku rostla i doba potřebná pro jeho načtení a zobrazení. Tato skutečnost mi příliš nevyhovovala a proto jsem se pokusil zbavit Hledač vody jeho velkého neduhu.

Důvod této komplikace souvisí s formátem IMAGINE souboru (Příloha 1). Všechny spektrální vrstvy jsou uloženy v takzvaném dlaždicovém formátu. To znamená, že hodnoty pixelů ve spektrální vrstvě nejdou v souboru za sebou, tak jak by se dalo očekávat, ale jsou seskupovány do čtvercových bloků a tyto bloky jsou ukládány jako celky. Autoři formátu IMAGINE souboru tvrdí, že díky dlaždicovému formátu je možné načítat a zobrazovat hodnoty ze souboru velice rychle. Problém však



Obrázek 15: Dlaždicový formát satelitního snímku s velikostí bloku 64 x 64 pixelů

byl, že jsem nezjistil jak.

První verze knihovny pro práci s IMAGINE soubory při načítání spektrálních vrstev tyto bloky upravovala a ukládala hodnoty za sebe, jak je běžné například u tabulek. To vyžadovalo načítání ze souboru jedné hodnoty za druhou, což bylo velmi pomalé a neefektivní. Řešením problému bylo akceptování skutečnosti zvláštního uskupení dat a úprava načítací metody třídy `Bands`. Tak vznikla nová metoda `LoadBandv2` (původní verze `LoadBand` se ve finální verzi nezachovala vzhledem k její nadbytečnosti) načítající naráz ne jednotlivé hodnoty, ale celé bloky. Struktura dat v paměti se tak stala shodnou se strukturou v souboru. Díky této inovaci se podařilo několiktisíckrát zrychlit načítání.

To ale s sebou přineslo i potíže. Nástroje pro textový i grafický náhled pracovaly s daty uloženými v paměti jako tabulka. Změna metod načítajících data pro zobrazení byla tedy dalším logickým a nutným krokem. To se však ukázalo jako velice problematické. Třídy `TSringGrid` a `TImage` nejsou defaultně stavěny na blokový zápis. Úprava metod v tomto smyslu tedy nepřipadala v úvahu. Proto přibyly ve třídě `Bands` dvě nové, často používané a navzájem komplementární metody `xy2ij` a `ij2xy` pro transformace souřadnic. Pixely ve vrstvě jsou v paměti uloženy v blocích a do zobrazovacích komponent je třeba je načítat jako tabulku.

Metoda `ij2xy` převádí pomocí jednoduchého dělení a sčítání souřadnice používané v blokovém uspořádání na souřadnice, jaké by měl daný pixel v tabulkovém uspořádání. Pro svoji činnost používá metoda hodnoty udávající šířku a výšku jednoho bloku, počet bloků v řádku a počet bloků ve vrstvě uložené speciálně pro tento účel zvlášť v každé spektrální vrstvě. `i` je index bloku v rámci celé vrstvy, `j` udává index pixelu v daném bloku. Souřadnice `x` a `y` odpovídají kartézskému uspořádání.

Metoda `xy2ij` naopak převádí souřadnice vyjádřené ve smyslu tabulkového uspořádání na souřadnice používané pro přístup k pixelům v paměti. Používá se stejně jako předchozí metoda a i její algoritmus je podobný.

Vytvoření transformačních metod umožnilo zachovat metody načítající hodnoty jasu pixelů z paměti v téměř nezměněné podobě. Nutnost používání transformačních metod při načtení každého jednotlivého pixelu však výrazně zvyšuje časovou složitost celého řešení. Za zvýšení rychlosti při načítání dat ze souboru do paměti se tak platí daň ve formě zpomalení samotného vykreslování načtených dat.

Řešením tohoto nového problému by nejspíš byla výrazná modifikace načítacích metod a možná i úprava tříd `TStringGrid` a `TImage`, ale žádná z mých snah nevedla ke zvlášť uspokojivým výsledkům.

2.2.4 ESRI komprese v IMAGINE souborech

Po nepříliš úspěšné snaze o zvýšení výkonu jsem začal pracovat na posledních nedostatcích omezujících zpracování některých .img souborů. IMAGINE soubory mají svůj vlastní vestavěný kompresní mechanismus, *ESRI GRID*, kvůli kterému byly některé soubory programem nečitelné. Schéma ESRI GRID komprese je popsáno v Příloze 1.

ESRI komprimuje každý blok zvlášť, proto nejsou výjimkou soubory mající některé bloky normální a některé komprimované. Skutečnost, zda je načítaný blok komprimován, je snadno zjistitelná přímo při načítání daného bloku. Upravená metoda `LoadBandv2` třídy `Bands` dokáže na tuto informaci reagovat a jedná-li se o komprimovaný datový blok, funkce k němu přistupuje zcela jinak.

Při analýze .img souborů jsem narazil na dvě varianty komprese. První nepoužívá run-length prvek a pouze od všech hodnot odečte minimální hodnotu jasu v bloku. Tabulka délek opakování neexistuje a počet záznamů této tabulky je uveden jako -1. Komprimovaných hodnot je poté stejný počet jako v nekomprimované podobě bloku, jen zabírají méně místa. Druhá varianta používá při kompresi run-length prvek a tabulka opakování prvků má nenulový počet záznamů. Komprimovaných hodnot jasu je shodný počet, ke každému záznamu tabulky jedna hodnota.

Nově vytvořená metoda `Deko` třídy `Bands` umožňuje dekompresi komprimovaných bloků a jejich načtení do paměti. Bývá volána metodou `LoadBandv2` v případě, že metoda `LoadBandv2` při zpracovávání načítané spektrální vrstvy narazí na blok komprimovaný první variantou ESRI GRID. V případě bloku komprimovaného druhou variantou metoda `LoadBandv2` data načte a předzpracuje do podoby, jaká by měla v případě první varianty komprese a poté zavolá metodu `Deko`.

Práce s kompresními algoritmy a se soubory na bitové úrovni je obecně velmi delikátní záležitost. Algoritmus metody `Deko` neumí zpracovat všechny typy komprese používané v IMAGINE souborech a to z toho důvodu, že se mi nepodařilo sehnat .img soubory se všemi typy komprese, abych na nich mohl metodu zprovoznit a doladit. V případě, že tato situace nastane, metoda `Deko` vytvoří objekt třídy `ErrorReport` a ohlásí tak chybu. Celý program pak dále korektně funguje, a otevíraný soubor je pro program nezpracovatelný.

2.3 Klasifikace satelitních snímků

Po zpracování souboru se satelitním snímkem, jeho načtení do paměti a zobrazení přichází na řadu samotná analýza. Podkapitola čtenáři přiblíží průběh vývoje knihovny funkčních prvků pro kvantitativní analýzu načtených satelitních dat metodou unsuservised klasifikace a tvorbu okna sloužícího jako nastavení parametrů prováděné analýzy.

2.3.1 Knihovna pro klasifikaci

Satelitní snímek je otevřen a zpracován, v paměti jsou v objektu třídy `Bands` načteny všechny spektrální vrstvy a na řadě je analýza. Jelikož má být program schopen sám vyhledat vodní plochy, fotointerpretace (kde je hlavním analytickým prostředkem odborník) logicky není použitelnou metodou analýzy snímků. Zbývá tedy kvantitativní analýza – klasifikace prováděná počítačem (kapitola 1.3.2). Ze dvou hlavních metod klasifikace jsem po odborné debatě s vedoucí bakalářské práce vyloučil supervised klasifikaci za prvé kvůli problému se získáváním předem pojmenovaných pixelů pro trénovací fázi klasifikátoru (kapitola1.3.6.1) a za druhé poněvadž nás zajímají pouze vodní plochy a tak uživatel nemusí mít povědomí o ostatních typech povrchu na zobrazeném snímku, což může být komplikace pro některé varianty supervised klasifikace, které potřebují pojmenované pixely ze všech na satelitním snímku se vyskytujících typů povrchu.

Úkolem je tedy v jazyce C++ vytvořit knihovnu nabízející funkce pro unsupervised klasifikaci satelitních dat načtených v paměti. Z existujících variant unsupervised klasifikace se jako nejvhodnější a nejefektivnější jeví varianta *migrating means*, česky pohyblivé středy, a její algoritmus ISODATA (více v Příloze 2), který jsem v klasifikační knihovně implementoval.

Knihovna obsahuje jedinou třídu `ImgClasses` používanou v souvislosti s klasifikací. Metoda migrating means vyžaduje předem znát počet spektrálních tříd (clusterů), do kterých se pokusí rozdělit všechny pixely v satelitním snímku podle hodnot jejich jasu (kapitola 1.3.4). Tento údaj spolu se souřadnicemi všech středů vytvořených clusterů v rámci multispektrálního prostoru jsou ve třídě uloženy. Dále třída obsahuje tabulku rozměrově kopírující klasifikovaný snímek. Je stejně blokově orientovaná jako spektrální vrstvy načtené v paměti. Hodnoty v tabulce udávají příslušnost pixelů stejných souřadnic do příslušných spektrálních tříd. Po vytvoření objektu třídy ImgClasses patří všechny pixely do třídy -1. To je výchozí stav, kdy pixely ještě nejsou klasifikovány. Třída obsahuje kromě konstruktoru, destruktoru a funkcí utvářejících rozhraní třídy několik hlavních metod. Jsou to metody `CreateVector`, `Distance`, `Initialize` a `uClass`.

Metoda `CreateVector` vytváří z pixelu zadaných souřadnic vektor v multispektrálním prostoru. Používá k tomu hodnoty jasu tohoto pixelu ze všech spektrálních vrstev. Je to pomocná metoda, která převádí hodnoty jasu uložené v různých datových typech do hodnoty pouze jednoho

datového typu a tím usnadňuje práci dalším metodám. Navíc je obvykle každý vektor při klasifikaci potřeba hned několikrát za sebou (tolikrát, do kolika shluků chceme rozdělit pixely ve snímku), urychluje jeho vytvoření práci a šetří čas.

Metoda `Distance` je důležitou měřící metodou. Její výsledky slouží jako hlavní metrika a jsou esenciální pro správné rozhodování a přiřazení zkoumaného pixelu do daného shluku. Počítá *Euklidovskou vzdálenost* (Příloha 2) dvou bodů v multispektrálním prostoru. Jedním z nich je koncový bod vektoru pixelu vytvořeného metodou `CreateVector` a druhým jeden ze středů vytvářených shluků.

Metoda `Initialize` je používána v konstruktoru třídy `ImgClasses` a také vždy, když uživatel vybere v hlavním menu položku **Zruš rozdělení**. Jejím úkolem je smazat všechny vytvořené středy shluků (pokud již nějaké existují) a anulovat rozdělení pixelů do shluků. Provede to změnou všech hodnot v tabulce určující příslušnost pixelů do spektrálních tříd na hodnotu -1.

Metoda `uClass` je hlavní metodou celé třídy. Provádí unsupervised klasifikaci otevřeného satelitního snímku pomocí implementace algoritmu ISODATA metody migrating means (Příloha 2). Vzhledem ke skutečnosti, že je algoritmus ISODATA iterační, a může se stát, že bude iterovat v nekonečné smyčce, vyžaduje metoda před spuštěním zadat limitní parametry. Prvním parametrem je informace o maximálním počtu iteračních cyklů, které má metoda provést. Druhým parametrem je desetinné číslo v intervalu <0, 1> a udává poměr mezi počtem pixelů, které jsou ve stejném shluku, jako byly v minulém iteračním cyklu, a počtem všech pixelů ve snímku. Nazývám tento parametr konvergentní zarážka. Její význam je velký, protože vlastně určuje přesnost, o jakou se má klasifikační algoritmus snažit. Čím blíže je hodnota konvergentní zarážky 1, tím více cyklů musí metoda provést a tak tím déle trvá celý průběh klasifikace. K ukončení iterace stačí, aby metoda překročila jeden ze zadaných limitních parametrů. Dalším potřebným parametrem je samozřejmě počet shluků, do kterých se bude algoritmus snažit rozdělit pixely snímku.

Po spuštění metody dojde k inicializaci středů vznikajících shluků (více v Příloze 2). Jelikož metoda pracuje s blokovým uspořádáním dat, je třeba kontrolovat přesahy. K tomu slouží několik proměnných a kontrolních přepočtů. Následuje samotný iterační cyklus. Na začátku každého cyklu dojde k přepočítání nových středů pro každý shluk. Souřadnice nových středů se počítají jako průměrné hodnoty všech bodů pixelů patřících do daného shluku zvlášť v každém rozměru multispektrálního prostoru. Poté metoda postupně prochází všechny pixely ve snímku, metodou `CreateVector` vytvoří z hodnot jasu zkoumaného pixelu vektor a metodou `Distance` počítá vzdálenost daného vektoru od každého středu shluku. Poté přiřadí zkoumaný pixel do toho shluku, od jehož středu má vektor nejmenší vzdálenost. Pokud je zkoumaný pixel už členem tohoto shluku, dojde tak k potvrzení jeho příslušnosti počítadlo `inSameClass` sledující počet pixelů přiřazených opět do stejného shluku se inkrementuje. Až metoda projde všechny pixely snímku, dojde k rozhodování, zda bude cyklus probíhat znovu, nebo skončí. Hodnota počítadla `inSameClass` se

vydělí počtem pixelů ve snímku a výsledná hodnota se porovná s hodnotou konvergentní zarážky zadané jako limitní parametr, hodnota určující počet provedených cyklů se porovná s maximálním počtem cyklů zadaných jako druhý limitní parametr. Pokud některá z kontrolovaných hodnot přesáhne limitní parametry, cyklus a tím pádem celá metoda `uClass` končí, jinak pokračuje další iteračním cyklem. Po ukončení jsou výsledné rozdělení pixelů do shluků a středy těchto shluků uloženy ve třídě.

V průběhu vytváření knihovny a jejím pozdějším dolaďování jsem narazil na velký počet problémů a komplikací. První verze klasifikační knihovny například neměly zvlášť vytvořenou metodu `Initialize` a tak po každé provedené klasifikaci bylo třeba otevřený soubor zavřít a znovu načíst do paměti. Metoda `CreateVector` také neexistuje hned od začátku existence knihovny. Její činnost prováděla metoda `Distance`, která tak byla zbytečně zdlouhavá a složitá. Špatně navržená počáteční inicializace středů shluků také komplikovala a zpomalovala klasifikaci. Chybu způsoboval špatný výpočet. Velmi složitým a zdlouhavým procesem také byla implementace zarážek přesahů z důvodu práce s blokovým uspořádáním dat.

Nejnovější částí knihovny je možnost při spuštění metody `uClass` zaregistrovat callback funkci umožňující výpis hodnot probíhající klasifikace. Registrovaná callback funkce musí mít čtyři parametry, tři celočíselné a jeden desetinný. Jeden z celočíselných a desetinný parametr předávají funkci informace o počtu cyklů a současném poměru nepřeřazených pixelů. Současný poměr je znám pouze na konci celého cyklu, avšak callback funkce je volána i v průběhu cyklu. V tuto chvíli předává metoda `uClass` callback funkci -1 místo hodnoty poměru.

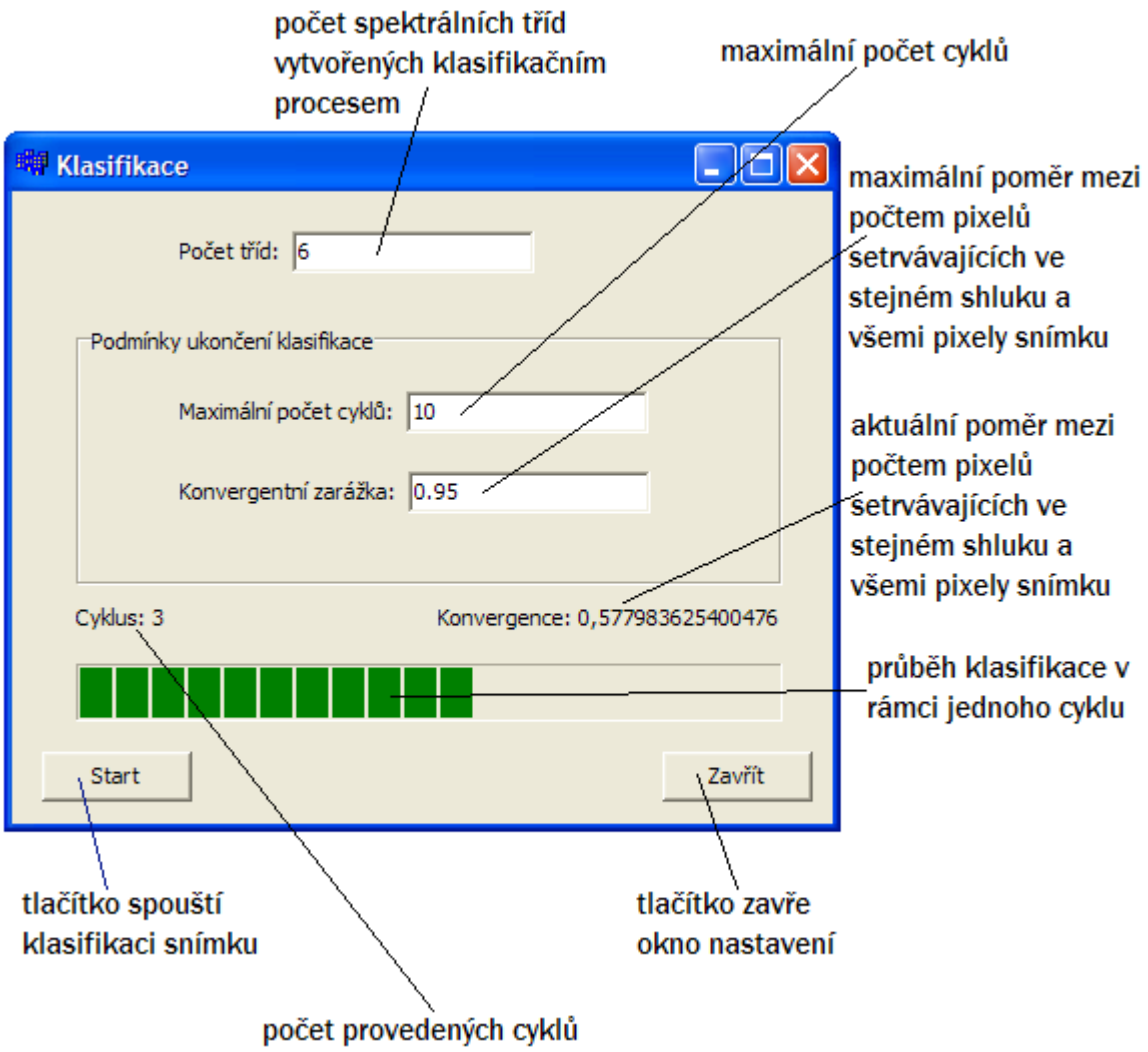
Další dva celočíselné parametry se uplatní právě v průběhu cyklu. Jeden z nich udává index právě zpracovávaného bloku a ten druhý určuje celkový počet bloků. Těchto dvou údajů lze využít například při vykreslování průběhu klasifikace.

2.3.2 Nastavení parametrů klasifikace

S vytvořenou klasifikační knihovnou je třeba nějakým způsobem komunikovat. Metoda `uClass` třídy `ImgClasses` vyžaduje pro svůj běh parametry, které je třeba nastavovat a měnit podle přání uživatele. Pro tyto potřeby bylo vytvořeno okno s nastavením parametrů klasifikace.

Výsledné okno je tvořeno ve velmi jednoduchém designu a na jeho výrobu jsou použity většinou pouze základní třídy nabízené vývojovým prostředím C++ Builder. Okno obsahuje tři textová pole, která použije uživatel pro zadávání potřebných parametrů. Popisky významů jednotlivých polí by měly uživateli dostatečně objasnit jejich význam. Dále tlačítko Start spouštějící klasifikaci a tlačítko Zavřít, které zavře celé okno s nastavením, a ve spodní části okna nad tlačítky se nachází dva informační bloky a ukazatel průběhu právě prováděné klasifikace.

Programově je celé okno tvořeno třídou `TFrm_ClassSetting`. Textová pole pro zápis parametrů klasifikace tvoří objekty třídy `TEdit` [4], tlačítka jsou třídy `TButton` [4], informační



Obrázek 16: Okno nastavení parametrů klasifikace

popisky jsou objekty třídy `TLabel` a ukazatel průběhu je tvořen třídou `TProgressBar` [4]. Textová pole pro nastavování parametrů klasifikace jsou od zbytku okna oddělena blokem třídy `TGroupBox` [4] z čistě estetického hlediska.

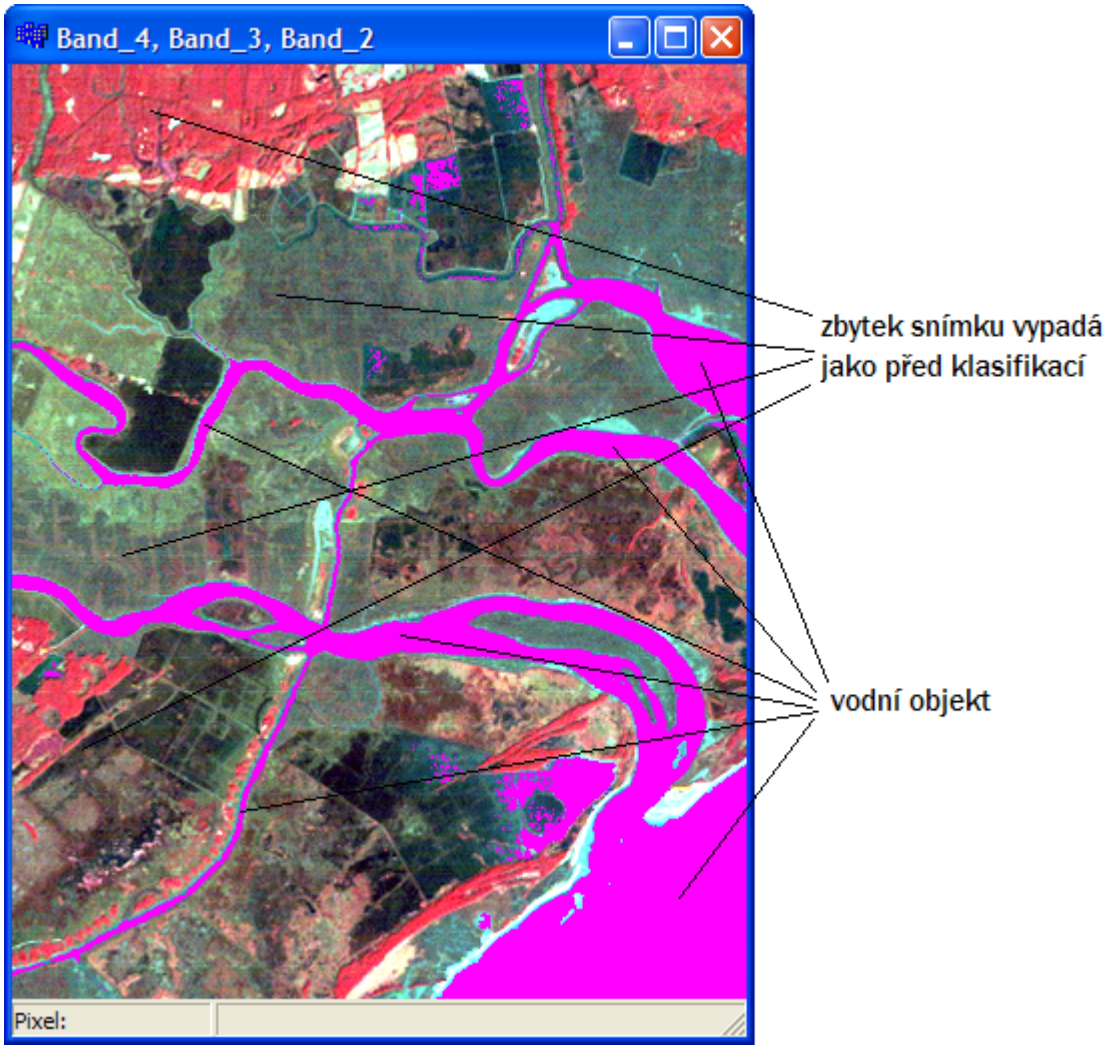
Jediná metoda třídy `TFrm_ClassSetting`, o které bych se rád zmínil, je metoda `ZobrazPostup`. Tato metoda slouží pro zobrazování průběhu klasifikace a je zaregistrována jako callback funkce používaná klasifikační knihovnou.

Při vytvářen okna pro nastavení se vyskytnulo několik menších problémů. Jedním z nich byla potřeba převodu textové interpretace desetinného čísla na skutečnou hodnotu datového typu plovoucí řádové čárky. Hlavním problémem s převodem byla existence nejednoznačnosti v definici znaku používaného jako řádový oddělovač. V našich končinách se používá čárka, v americky mluvících zemích je to tečka. Řešení umožnila změna globální proměnné `DecimalSeparator` [4] používané pro tyto účely v programech vytvořených vývojovým prostředím C++ Builder.

Další komplikací byla potřeba zabránit uživateli manipulovat s ostatními částmi programu, když je zobrazeno okno s nastavením. Řešením poskytlo zobrazování tohoto okna modálně. [6]

2.3.3 Zobrazování výsledků klasifikace

Uživatel otevřel satelitní snímek, který se načetl do paměti, nechal si zobrazit okno s nastavením klasifikace, poté nastavil parametry a klasifikaci spustil. Po skončení klasifikace by však také rád viděl nějaký výsledek jeho (ač ve skutečnosti počítal stroj) snažení. Díky relativně dobře propracovanému nástroji pro grafický náhled je pochopitelné jeho využití pro zobrazení výsledků provedené klasifikace. Důležité však je, co je kýženým výsledkem. Objektem zájmu jsou vodní objekty, ale unsupervised klasifikace nepracuje se znalostí jednotlivých spektrálních tříd. Je tedy třeba identifikovat některou ze spektrálních tříd jako třídu popisující vodní objekty.



Obrázek 17: Výsledek klasifikace zobrazen do původního obrázku

Ve většině programem analyzovaných snímcíh vytvářely spektrální třídy popisující odrazivost vody kompaktní shluky, a tak byla převážná většina pixelů zobrazujících vodní objekty přiřazena do jedné spektrální třídy. Z obrázků v kapitole 1, zvlášť z Obrázků 4, 7 a 8 je patrné, že spektrální třídy

popisující vodní objekty se nachází nejblíže počátku souřadnic multispektrálního prostoru. Inicializace středů shluků na začátku algoritmu ISODATA (Příloha 2) zajišťuje, že první střed uložený ve třídě `ImgClasses` je střed shluku nejblíže počátku souřadnic multispektrálního prostoru. Usoudil jsem tedy, že ve většině případů je spektrální třídou popisující vodní objekty třída první.

Zobrazování výsledků klasifikace řeší modifikovaná metoda `BandPaint` třídy `TFrm_Viewer`. Při průchodu jednotlivými pixely satelitního snímku metoda kontroluje tabulku rozdělení pixelů do shluků a řídí vykreslování podle ní. Před samotnou klasifikací (kdy patří všechny pixely do skupiny -1) se zobrazí načtený satelitní snímek normálně. Po klasifikaci dojde k překreslení a pixely patřící do první spektrální třídy (podle tabulky příslušnosti patří do skupiny 0) jsou vykreslovány jinou barvou, na rozdíl od ostatních, jichž se změna nedotkne. Tento způsob zpracování má výhodu v tom, že zachová ostatní informace a snímek tak vypadá jako před klasifikací, jen má vodní objekty vykresleny jinou barvou. Pomocí kontextového menu lze libovolně měnit výběr zobrazených spektrálních vrstev, aniž by se nějak ovlivnil vykreslený výsledek klasifikace.

Barva vybraná pro vyznačení vodních objektů byla vybrána náhodně a ve vývojovém prostředí C++ Builder má název `clFuchsia` [4]. Měl jsem pocit, že v určitých případech není dostatečně kontrastní, ale po konzultaci s vedoucí bakalářské práce byla vybraná vykreslovací barva zachována.

Na Obrázku 17 je patrno, že některé pixely jsou označeny chybně jako pixely obsahující vodní objekt a některé pixely, které popisují vodní plochu, tak naopak označeny nejsou. Může to být způsobeno mnoha různými faktory. Například vliv elektromagnetických poruch ovlivňující zaznamenané hodnoty, nebo pixel, jehož spektrální vlastnosti jsou natolik podobné, že ho metoda `uClass` chybně označila jako pixel vodního objektu. Implementovaná metoda uClass ani zdaleka není bezchybná a ideální a tak jsou takovéto chyby očekávatelné a pochopitelné.

Závěr

Na závěr patří shrnutí celé mé dosavadní činnosti v souvislosti se zadáním bakalářské práce.

1. Prostudoval jsem základy remote sensing (kapitola 1.1), strukturu satelitních snímků a jejich vlastmosti (kapitola 1.2) metody analýzy a zpracování obrazu (kapitola 1.3.1) a klasifikace (1.3.3).
2. Seznámil jsem se s metodami používanými pro klasifikaci obrazových materiálů v družicových snímcích (kapitoly 1.3.5 a 1.3.6)
3. Navrhnul jsem příznakovou sadu vhodnou pro detekci vodních objektů v družicových snímcích. Jsou to spektrální prvky uložené v satelitním snímku a spektrální prostor z nich vytvořený. (kapitoly 1.3.4 a 2.3.3)
4. Navrhnul jsem postupy, které mají za úkol rozeznávat vodní objekty v družicových snímcích. Použil jsem unsupervised klasifikaci (kapitola 1.3.6.2) a její algoritmus ISODATA (Příloha 2)
5. Postupy jsem implementoval. Detailnější popis celého průběhu implementace následuje.

Vytvořil jsem hlavní menu jako základní prvek GUI a zároveň spojovací můstek celého programu (kapitola 2.1) Je hlavním ovládacím nástrojem celého programu.

Po vytvoření základní podoby hlavního menu mě čekal úkol, který mi zabral asi nejvíce času a přitom o něm není v zadání bakalářské práce žádná zmínka. Musel jsem z velkého množství různých existujících formátů satelitních snímků vybrat jeden, který by mi nejvíce vyhovoval a „naučit“ můj program s tímto formátem pracovat (více v kapitole 2.2.1). Až dlouho po vykonání celé této složité a zdlouhavé práce jsem objevil několik DLL knihoven schopných pracovat s IMAGINE formátem. Na druhou stranu díky tomu, že jsem celou část programu, která má na svědomí zpracování souborů, psal sám, dobře jsem pronikl do problematiky a pochopil celou strukturu IMAGINE formátu.

Poté, co jsem měl satelitní data načtená ze souboru, bylo třeba je nějakým způsobem zobrazit uživateli za prvé textově (kapitola 2.2.2.1) a za druhé graficky (kapitola 2.2.2.2). Po úspěchu se zobrazovacími prvky jsem však byl velmi nemile překvapen nedostatečnou rychlostí zobrazování a proto jsem se pokusil úpravou načítacích algoritmů zvýšit výkon aplikace (více v kapitole 2.2.3). Nakonec jsem za účelem rozšíření aplikace upravil knihovnu pro práci s IMAGINE soubory (popsanou v kapitole 2.2.1) a implementoval podporu ESRI komprese (kapitola 2.2.4).

Následovala implementace části programu zodpovědné za analýzu a klasifikaci satelitních snímků.

Hlavním účelem programu bylo analyzovat satelitní snímek a identifikovat v něm vodní objekt (řeku, jezero, moře, …) pomocí některé z klasifikačních metod. Vybral jsem si unsupervised klasifikaci (kapitola 1.3.6.2) pro její jednoduchou implementaci a ovladatelnost. Přesněji řečeno metodu ISODATA spadající do skupiny migrating means. Její bližší specifikace je v Příloze 2.

Vytvořil jsem tedy klasifikační knihovnu umožňující aplikaci zmíněného algoritmu na načtených satelitních snímcích. (kapitola 2.3.1) Vytvořenou knihovnu bylo třeba připojit k programu, proto jsem vytvořil nastavovací rozhraní pro jednoduchou práci s klasifikačním nástrojem. (kapitola 2.3.2) Samotná klasifikace však nestačila pro úspěšnou identifikaci vodní plochy a proto jsem začal experimentovat s hybridizací této metody. Postup, průběh a výsledky jsou shrnuty v kapitole 2.3.3.

6. Program jsem testoval jako hotový celek i během jeho vývoje na několika satelitních snímcích, které přikládám na CD ve složce **Ukázka**. Výsledná klasifikace a zobrazení výsledku hodnotím vcelku kladně. Obvykle se nalezená oblast shoduje z více, než 90% se skutečnou vodní plochou zobrazenou na snímku. Existují však snímky, které program zpracovat neumí (výsledky nedávají smysl). Způsobuje to skutečnost, že klasifikační i zobrazovací nástroj byly vytvářeny a laděny za asistence jen malého procenta všech existujících satelitních snímků. Jak zmiňuje [3], výsledek jakékoliv analýzy vždy závisí na znalostech a zkušenostech uživatele; samotný analytický a klasifikační proces nelze jednoduše automatizovat. Alespoň zatím.

7. Hodnocení jsem částečně provedl již v předchozím bodě, proto se zaměřím na možnosti budoucího vývoje a rozšiřování programu. Těch je hned několik.

Program by mohl umět zpracovávat víc formátů souborů používaných pro ukládání satelitních dat. Stejně tak by mohl ukládat analyzované snímky zpátky do souboru a provádět tak zálohy výsledků klasifikace. Další možností vývoje je klasifikační knihovna. Vedle unsupervised klasifikace by mohla přibýt supervised klasifikace se vším, co k ní náleží (načítání a zpracování trénovaní sady, atd.) Zobrazování výsledků je možno také modifikovat. V programu by mohl být nástroj pro práci s vytvořenými spektrálními třídami, jejich spojováním a rozdělováním na základě postupů popsaných v [3] Program by se také mohl rozvinout více pro fotointerpretaci. V grafickém náhledu by mohla přibýt technika zoom a program by mohl nabízet víc technik posilování kontrastu. Je to jen pár možností rozvoje, jistě jsem nevyjmenoval všechny.

8. Přiznávám, že plakát jsem vypustil a nepokoušel jsem se o jeho vytvoření, jelikož nejsem dobrý grafik a tak by víc než plakát reprezentující dosažené výsledky a použité postupy připomínal strašidelnou koláž. O to podrobněji jsem se snažil propracovat technickou zprávu a připojit k ní co nejvíce obrázků z programu.

Nezbývá, než zhodnotit vlastní přínos. Téma práce jsem si vybral, protože mě zajímalo, jak taková analýza probíhá. V průběhu studia materiálů a implementace výsledného programu byla moje zvědavost uspokojena. Pochopil jsem základy práce se satelitními snímky, jejich složitost, ale i důležitost zejména v dnešní době. Zjistil jsem, jak vypadá struktura .img souborů a co vše je třeba uložit spolu s naměřenými hodnotami, aby měl vytvořený snímek nějakou analytickou a geografickou hodnotu. Zopakoval jsem si teorii z oblasti umělé inteligence při načítání problematiky supervised a unsupervised klasifikace. Až teprve při psaní této technické zprávy jsem si uvědomil, jak spolu souvisí metoda fotointerpretace a mnou vytvořený nástroj pro grafický náhled.

Literatura

[1] *Dálkový průzkum Země*.
Dokument dostupný na URL http://cs.wikipedia.org/wiki/Dálkový_průzkum_Země (květen 2008)

[2] *Remote sensing*. Dokument dostupný na URL http://en.wikipedia.org/wiki/Remote_sensing (květen 2008)

[3] Richards, J.A., Xiuping, J. *Remote sensing digital image analysis: an introduction*, 4th ed., Berlin, Springer 2006.

[4] Borland Software Corporation. *Developer Studio 2006 C++ Builder Reference*, 2005.

[5] Leica Geosystems GIS & Mapping, LLC. *ERDAS IMAGINE .img file format documentation*. Dokument dostupný na URL http://home.gdal.org/projects/imagine/iau_docu0.pdf (květen 2008) a http://home.gdal.org/projects/imagine/iau_docu1.pdf (květen 2008).

[6] *Tvorba uživatelských rozhraní: Dialogové boxy, základní prvky*. Dokument dostupný na URL https://www.fit.vutbr.cz/study/courses/ITU/private/lectures/itu-dialog-cs.pdf

[7] Leica Geosystems Geospatial Imaging, LLC. *ERDAS Field Guide: Volume Two*, 2008.
Dokument dostupný na URL
http://gi.leica-geosystems.com/documents/pdf/FieldGuide_Vol2_March2008.pdf (květen 2008)

Seznam příloh

Příloha 1. Struktura IMAGINE souboru

Příloha má za úkol čtenáři přiblížit implementační detaily formátu IMAGINE .img souboru v takové míře, v jaké jsem je nastudoval a použil já při vytváření knihovny pro práci s IMAGINE soubory (kapitola 2.2.1). Obsah přílohy vychází z [5].

Program ERDAS IMAGINE používá .img soubory pro ukládání rastrových dat. IMAGINE soubory používají ERDAS IMAGINE *Hierarchal File Format* (HFA), česky hierarchický formát souboru. [5]

Informace o vrstvě

Každá spektrální vrstva v .img souboru má vlastní podpůrná data:

- výška a šířka vrstvy (počet řádků a sloupců)
- typ vrstvy (tématická nebo kontinuální – s hodnotami naměřenými satelitními senzory)
- datový typ použitý pro uložení pixelů (8bit, plovoucí řádová čárka, …)
- typ komprese (žádná, ESRI GRID)
- velikost bloku

Tyto údaje jsou obvykle shodné pro všechny spektrální vrstvy ve snímku.

Velikost bloku

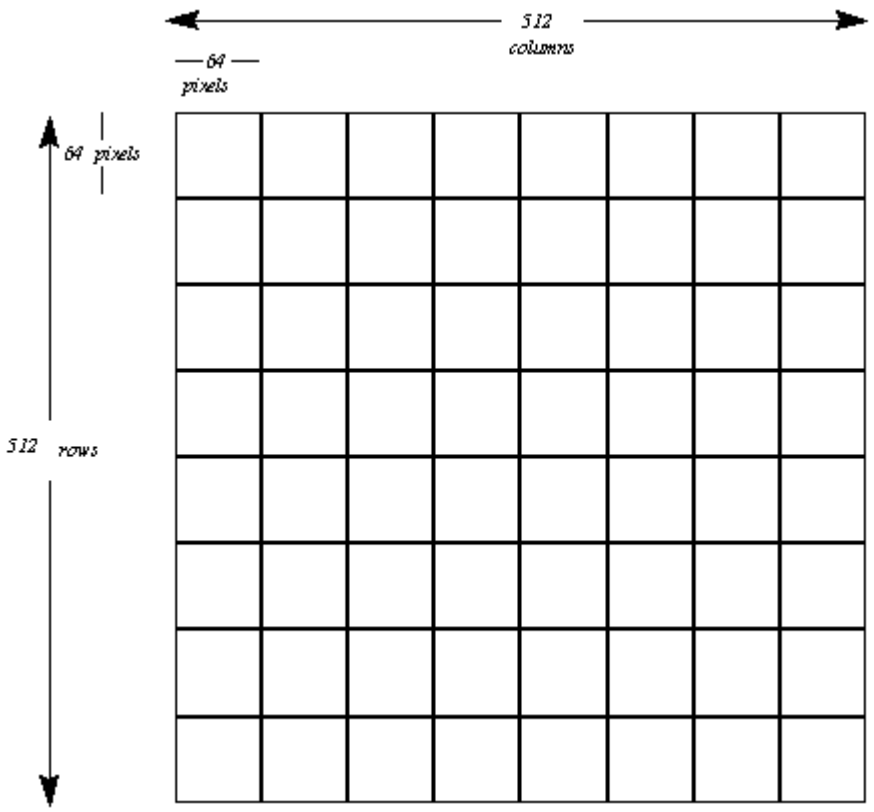
.img soubor používá *dlaždicový formát* pro uložení rastrových dat. Dlaždicový formát umožňuje rychlé načítání a zobrazování spektrálních vrstev. Spektrální rastrová vrstva je rozdělena na dlaždice (bloky). Standardní velikost bloku je 64 na 64 pixelů. [5]

HFA Objektový adresář

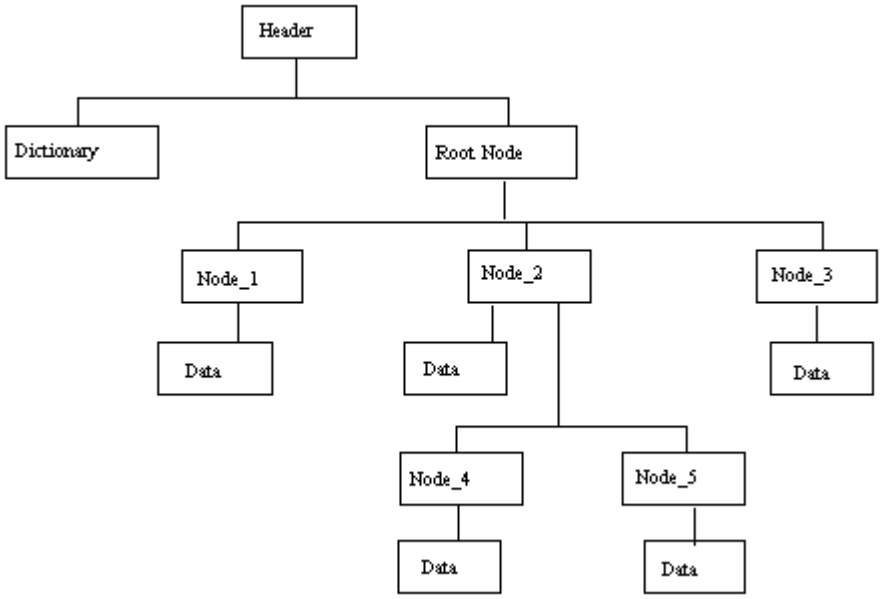
IMAGINE soubory používají HFA formát. Tento formát umožňuje uložení několika různých typů dat do jednoho fyzického souboru za použití stromové struktury. Každý .img soubor je tvořen stromem, jehož uzly jsou objekty různých dat. Obsah těchto uzlů (stejně jako informace o struktuře) je uložen pomocí *machine independent format* (MIF), česky formát nezávislý na stroji, což umožňuje používání .img souborů na počítačích odlišné architektury. [5] Většina typů definovaných v MIF se shoduje s typy používanými v jazyce C++.

Hierarchická architektura souboru

Hierarchická architektura souboru udržuje objektově orientovanou strukturu dat pomocí stromové struktury. Každý objekt se nazývá *entry*, česky záznam, a je uložen v jednom uzlu stromu. Každý záznam má vlastní jméno a typ. Typ vypovídá o struktuře dat uložených v záznamu. Navíc každý záznam může obsahovat ukazatel na podstrom nebo další uzly. Všechny záznamy jsou uloženy v MIF. [5]



Obrázek P-1: Příklad 512 x 512 spektrální vrstvy rozdělené na bloky velikosti 64 x 64 pixelů [5]



Obrázek P-2: HFA struktura souboru [5]

Krom velkého množství typů existuje několik předdefinovaných, jejichž uspořádání a význam zná každý. Tyto předdefinované typy spadají do tří oblastí:

- Základní HFA objekty
- .img objekty
- objekty hlaviček externích souborů

Základní HFA objekty

Následuje seznam typů základních objektů vyskytujících se v každém .img souboru

Ehfa_HeaderTag

Ehfa_HeaderTag je používán na začátku souboru pro jednoznačnou identifikaci .img formátu. Vždy musí zabírat prvních 20 bajtů souboru. [5]

TYP	JMÉNO	POPIS
CHAR[16]	label	Úvodní řetězec "EHFA_HEADER_TAG"
ULONG	headerPtr	Ukazatel na záznam typu *Ehfa_File*.

Tabulka P-1: Detailní struktura záznamu Ehfa_HeaderTag [5]

Ehfa_File

Záznam Ehfa_File je tvořen z několika hlavních částí, zahrnujících ukazatel na free list, slovník a kořenový uzel stromové struktury. Tyto objekty mohou být kdekoliv po souboru a proto existuje záznam Ehfa_File. [5]

TYP	JMÉNO	POPIS
LONG	version	Definuje verzi záznamu Efha_File. V současnosti 1.
ULONG	freeList	Ukazatel na seznam volných bloků v souboru. Tento seznam je prohledáván, kdykoliv je potřeba nové místo.
ULONG	rootEntryPtr	Ukazatel do souboru na kořenový uzel.
SHORT	entryHeader Length	Definuje délku záznamové části každého uzlu. Každý uzel se skládá ze dvou částí. První je záznamová část obsahující jméno uzlu, typ a informace o okolí v rámci stromu. Druhou částí jsou data uzlu.
ULONG	dictionaryPtr	Ukazatel do souboru na MIF slovník. Slovník musí být přečten a dekódován před každým dalším zpracování souboru.

Tabulka P-2: Detailní struktura záznamu Ehfa_File [5]

Ehfa_Entry

Záznam Ehfa_Entry obsahuje hlavičkové informace pro každý uzel stromové struktury jako jméno a typ uzlu, stejně jako informace o umístění ve struktuře. [5]

TYP	JMÉNO	POPIS
ULONG	next	Ukazatel do souboru na následující „bratrský“ uzel v rámci stromové struktury. Pokud je to poslední uzel, pole obsahuje 0.
ULONG	prev	Ukazatel do souboru na předchozí „bratrský“ uzel v rámci stromové struktury. Pokud je to první uzel, pole obsahuje 0.
ULONG	parent	Ukazatel do souboru na předchozí „rodičovský“ uzel v rámci stromové struktury. Pokud je to kořenový uzel, pole obsahuje 0.
ULONG	child	Ukazatel do souboru na první „synovský“ uzel v rámci stromové struktury. Pokud uzel nemá syny, pole obsahuje 0.
ULONG	data	Ukazatel do souboru na data daného uzlu. Pokud uzel neobsahuje žádná data, pole obsahuje 0.
LONG	dataSize	Velikost datové části uzlu v bajtech
CHAR[64]	name	Nulou ukončený řetězec se jménem tohoto uzlu. Zabírat může maximálně 64 bajtů i s ukončující 0.
CHAR[32]	type	Nulou ukončený řetězec s typem tohoto uzlu. Typ musí být definován ve slovníku. Zabírat může maximálně 32 bajtů i s ukončující 0.
TIME	modTime	Obsahuje čas poslední změny v uzlu provedené.

Tabulka P-3: Detailní struktura záznamu Ehfa_Entry [5]

.img objekty

Následuje výňatek ze seznamu typů předdefinovaných objektů obvykle používaných v .img HFA souborech. Výňatek zahrnuje ty typy, které jsem použil ve své práci.

Eimg_Layer

Eimg_Layer je kořenový uzel podstromu objektů vztahujících se k jedné uložené spektrální vrstvě. Objekt popisuje základní informace o vrstvě, jako je její výška a šířka v pixelech, datový typ použitý pro jejich uložení, a šířka a výška jednoho bloku v pixelech. Ostatní informace o spektrální vrstvě (hodnoty jasu pixelů, statistické údaje, atd.) jsou uloženy jako synovské uzly. [5]

TYP	JMÉNO	POPIS
LONG	width	Šířka vrstvy v pixelech
LONG	height	Výška vrstvy v pixelech.
ENUM	layerType	Typ uložené vrstvy. 0="tématická" 1="kontinuální"
ENUM	pixelType	The type of the pixels. 0="unsigned 1bit" 1=" unsigned 2bit" 2=" unsigned 4bit" 3=" unsigned 8bit" 4="signed 8bit" 5=" unsigned 16bit" 6=" signed 16bit" 7=" unsigned 32bit" 8= "signed 32bit" 9="float 32bit" 10="float 64bit" 11="komplex 64bit" 12="komplex 128bit"
LONG	blockWidth	Šířka každého bloku v pixelech.
LONG	blockHeight	Výška každého bloku v pixelech.

Tabulka P-4: Detailní struktura záznamu Eimg_Layer [5]

Edms_VirtualBlockInfo

Objekt Edms_VirtualBlockInfo popisuje jeden rastrový blok spektrální vrstvy. Popisuje, kde se nachází hodnoty jasu daného bloku, kolik bajtů dat je v bloku uloženo, a také informaci, zda je blok komprimován. Popis používané komprese je pod tabulkou. [5]

TYP	JMÉNO	POPIS
SHORT	fileCode	Položka slouží k rozdělení vrstvy do více souborů. Hodnota identifikuje soubor použitý pro uložení bloku. V současnosti je vždy 0, jelikož systém ukládání do více souborů není zatím implementován.
ULONG	offset	Ukazatel do souboru na místo, kde začíná blok hodnot jasu.
LONG	size	Počet bajtů v bloku.
ENUM	logvalid	Indikuje, zda blok skutečně obsahuje validní data. To umožňuje existenci bloku v mapě, ale ne v souboru. 0="false" 1="true"
ENUM	compression Type	Oznamuje, zda je blok komprimován. 0="žádná komprese" 1="komprese ESRI GRID" Pokud není použita žádná komprese, blok je uložen jako proud za sebou uložených hodnot definovaného datového typu.

Tabulka P-5: Detailní struktura záznamu Edms_VirtualBlockInfo [5]

ESRI GRID je dvouprůchodový run-length kompresní algoritmus. Používá se pro kompresi vrstev, jejichž hodnoty jasu jsou uloženy jako hodnoty celočíselného datového typu a komprimuje zvlášť jednotlivé bloky. Průběh komprese je následující:

- lokalizuje se maximální a minimální hodnota jasu v daném bloku
- podle velikosti rozdílu mezi maximální a minimální hodnotou se určí bitová velikost výsledných komprimovaných hodnot. Pokud je rozdíl menší nebo roven 256, je výsledná velikost 8bitová, v případě menšího rozdílu než 65536 je 16bitová a v ostatních případech 32bitová.
- minimální hodnota se odečte od každé hodnoty v bloku
- run-length metoda je použita pro kompresi několika za sebou jdoucích pixelů stejné jasové hodnoty

Data uložená v souboru mají následovanou strukturu. Minimální hodnota jasu daného bloku je uložena na prvních 4 bajtech, následuje údaj o počtu run-length segmentů na dalších 4 bajtech a další 4 bajty obsahují relativní offset na začátek komprimovaných hodnot pixelů. V následujícím bajtu je informace o počtu bitů použitých na jednu hodnotu. Za tímto bajtem následuje tabulka udávající počty opakování určeného komprimovaného pixelu. Počet jejich položek je určen dříve ve druhé čtveřici bajtů. Za tabulkou se nachází samotná komprimovaná data bloku. Sem ukazuje relativní offset. Každý záznam tabulky určující počet opakování může mít od jednoho do čtyř bajtů (bráno z pohledu jak jsou data uložena v souboru). První dva bity prvního bajtu určují, kolik bajtů je použito na daný záznam (1, 2, 3 nebo 4). Následujících x bajtů určuje počet opakování komprimované hodnoty pixelu, která je zrovna na řadě. Na každou komprimovanou hodnotu připadá jeden záznam v tabulce počtu opakování.

Edms_RLCParams

Objekt Edms_RLCParams popisuje parametru run-length komprese použité na bloku dat. Tento objekt bývá uložen na začátku komprimovaného bloku. [5]

TYP	**JMÉNO**	**POPIS**
LONG	min	Minimální hodnota jasu v komprimovaném bloku.
LONG	numsegments	Počet záznamů v tabulce opakování
LONG	dataoffset	Relativní ukazatel do souboru od počátku záznamu Edms_RLCParams j začátku komprimovaných hodnot.
UCHAR	numbitspervalue	Počet bitů použitých pro uložení jedné komprimované hodnoty.

Tabulka P-6: Detailní struktura záznamu Edms_RLCParams [5]

Esta_Statistics

Objekt typu Esta _Statistics je používán pro uložení statistických informací o vrstvě [5]

TYP	JMÉNO	POPIS
DOUBLE	minimum	Minimální hodnota jasu vyskytující se ve vrstvě.
DOUBLE	maximum	Maximální hodnota jasu vyskytující se ve vrstvě.
DOUBLE	mean	Střední hodnota všech hodnot jasu ve vrstvě.
DOUBLE	median	Medián všech hodnot jasu ve vrstvě.
DOUBLE	mode	Modus všech hodnot jasu ve vrstvě.
DOUBLE	stddev	Směrodatná odchylka hodnot jasu ve vrstvě.

Tabulka P-7: Detailní struktura záznamu Esta _Statistics [5]

Edms_State

Objekt Edms_State popisuje umístění všech bloků patřících do jedné vrstvy. V podstatě se jedná o tabulku indexů jednotlivých bloků vrstvy. [5]

TYP	JMÉNO	POPIS
LONG	numvirtualblocks	Počet bloků v dané vrstvě.
LONG	numobjectsperblock	Počet hodnot jasu uložených v jednom bloku.
LONG	nextobjectnum	Tato hodnota se nepoužívá. Je rezervována pro budoucí použití.
ENUM	compressionType	Indikuje, zda se ve vrstvě vyskytují komprimované bloky. 0="žádné komprimované bloky" 1="některé bloky jsou komprimovány ESRI GRID"
Edms_VirtualBlockInfo p	blockinfo	Tabulka záznamů Edms_VirtualBlockInfo.
Edms_FreeIDList p	freelist	Tato hodnota se nepoužívá. Je rezervována pro budoucí použití.
TIME	modTime	Čas poslední modifikace dat ve vrstvě.

Tabulka P-8: Detailní struktura záznamu Edms_State [5]

Příloha 2. Klasifikační metoda ISODATA

V autorství algoritmu ISODATA jsou nesrovnalosti. [3] tvrdí, že ISODATA představila v roce 1965 dvojice Ball a Hall. Naproti tomu [7] její autorství připisuje dvojici Tou a Gonzales roku 1974. Oba zdroje se však shodují v definici samotného algoritmu.

ISODATA, neboli *Iterative Self-Organizing Data Analysis Technique*, česky iterativní samo organizační technika k analýze dat je jednou z metod migrating means unuspervised klasifikace. [7] Používá metriku spektrální vzdálenosti k přiřazení zkoumaného pixelu do nějakého shluku. Proces začíná se specifikovaným počtem předem vybraných středů shluků a poté iteruje, posouvá tyto středy doprostřed nalezených shluků a opět znovu přiřazuje všechny zkoumané pixely do shluků na základě spektrální vzdálenosti. Díky tomu, že je iterativní, není ovlivněna strukturou prvních řad pixelů jako jiné, jednoprůchodové klasifikační metody. [7]

ISODATA metoda vyžaduje tři parametry pro svoji funkci:

- Počet shluků, do kterých má metoda roztřídit pixely
- Konvergentní zarážka, která udává maximální množství pixelů, jejichž příslušnost může zůstat nezměněna mezi dvěma sousedními iteracemi.
- Maximální počet iterací, které se mají provést

Spektrální vzdálenost

Euklidovská spektrální vzdálenost je vzdálenost v multispektrálním prostoru. Je to hodnota, která umožňuje porovnat podobnost dvou spektrálních vektorů.[7] Euklidovskou spektrální vzdálenost dvou bodů lze spočítat pomocí vzorce:

$D = \sqrt{\sum_{i=1}^{n} \left(d_i - e_i \right)^2}$ [7], kde

D – spektrální vzdálenost

n – počet spekter (rozměrů)

i – příslušné spektrum

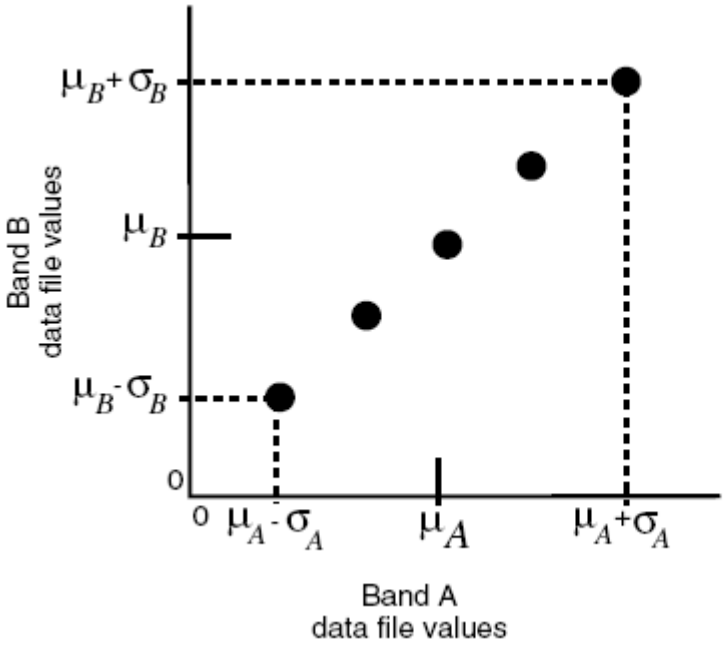
d_i – hodnota jasu pixelu d ve spektru i

e_i – hodnota jasu pixelu e ve spektru i

Ve dvourozměrném prostoru je vzorec pro výpočet spektrální vzdálenosti shodný s Pythagorovou větou.

Inicializace středů shluků

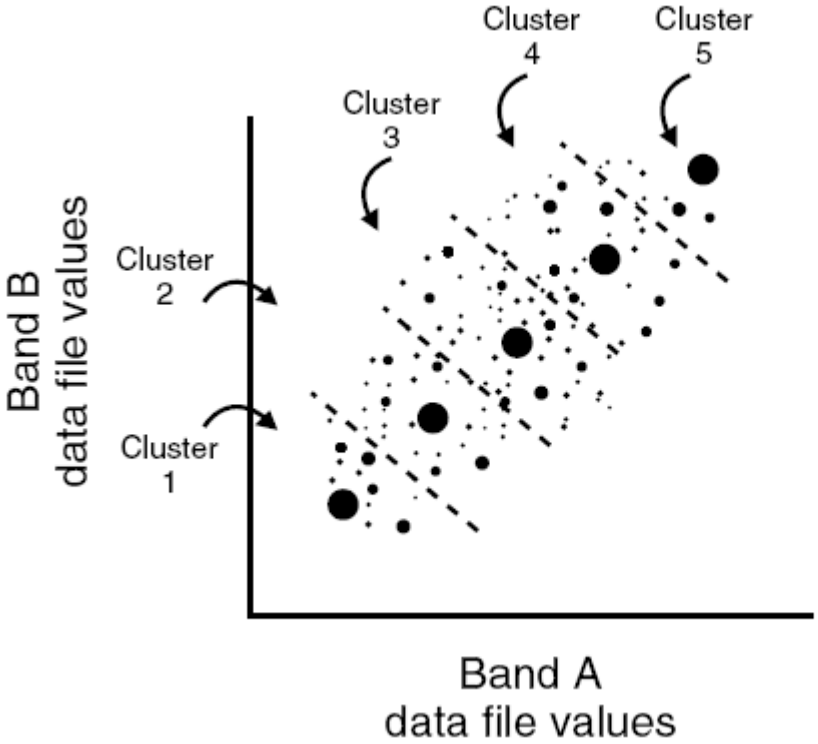
Při prvním cyklu algoritmu ISODATA je třeba inicializovat středy budoucích shluků, jejichž počet byl zadán jako parametr. Při ostatních cyklech se středy shluků počítají pomocí spektrálních pozic



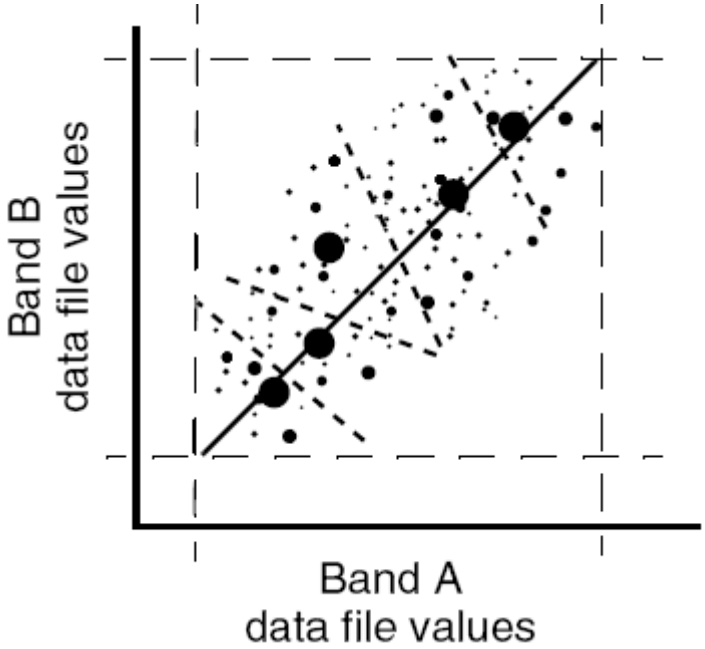
Obrázek P-3: 5 inicializovaných středů shluků ve dvourozměrném multispektrálním prostoru [7]

pixelů do daného shluku přiřazených. Poté jsou tyto nové shluky namísto starých používány při rozhodování. [7]

Inicializované středy shluků leží v multispektrálním prostoru rozmístěny na vektoru vytvořeném mezi body o souřadnicích (μ_1-σ_1, μ_2-σ_2, μ_3-σ_3 … μ_n-σ_n) a (μ_1+σ_1, μ_2+σ_2, μ_3+σ_3 … μ_n+σ_n). [7] μ je znak používaný pro střední hodnotu, σ je znak používaný pro směrodatnou odchylku.



Obrázek P-4: Po prvním cyklu [7]



Obrázek P-5: Po druhém cyklu [7]

Na Obrázku P-3 je zobrazen jednoduchý multispektrální prostor s pěti inicializovanými středy shluků v rozsahu (μ_A-σ_A, μ_B-σ_B) a (μ_A+σ_A, μ_B+σ_B).

Analýza pixelů

Algoritmus začíná v levém horním rohu spektrální vrstvy a postupuje zleva doprava blok po bloku, spočítá spektrální vzdálenost mezi vybraným pixelem a každým ze středů shluků a přiřadí pixel do shluku, k jehož středu je nejblíž. [7]

Uvažujeme-li standardní inicializaci středů shluků, výsledek prvního cyklu ISODATA algoritmu bývá zpravidla velmi blízce podobný Obrázku P-4. Pro druhý cyklus se všechny středy shluků přepočítají a posunou ve spektrálním prostoru (odtud název metody migrating means) a poté se celý přiřazovací cyklus opakuje s novými středy. Obrázek P-5 ilustruje stav po druhém cyklu, kdy jsou středy shluků posunuty oproti počátečnímu stavu.

Příloha 3. CD

Na zadních deskách je kapsa s CD. Na CD je v adresáři **Technická zpráva** uložena elektronická podoba této technické práce, zdrojové soubory k projektu v adresáři **Projekt**, v adresáři **Dokumentace** se nachází programová dokumentace vytvořená nástrojem Doxygen a v adresáři **Ukázka** je zkompilovaný program spolu s pár IMAGINE soubory pro otestování funkčnosti.