

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

**WEBOVÁ APLIKACE PRO ANALÝZU
OBRAZOVÝCH MATERIÁLŮ S VYUŽITÍM
SENZOROVÝCH DAT**

Diplomová práce

Štěpán ROUBALÍK

Vedoucí práce doc. RNDr. Vilém PECHANEC, Ph.D.

Olomouc 2019

Geoinformatika

ANOTACE

Diplomová práce pojednává o tvorbě webové aplikace pro analýzu obrazových materiálů s využitím dat ze sensorové sítě. V praxi se jedná o kombinaci metod Dálkového průzkumu Země a postupů aplikovaných v oblasti vývoje informačních technologií a softwarového inženýrství. Důležitým bodem je využití volně dostupných dat a technologií pro samotný vývoj aplikace. Vytvořením funkčního řešení aplikace má být docíleno přesnějších výsledků, získaných z analýzy obrazových dat. Zpřesnění těchto výpočtů má zajistit integrace sensorových dat do aplikace.

Práce je rozdělena na dva hlavní oddíly, část teoretickou a praktickou. V teoretické části jsou vymezeny cíle práce, metody zpracování, a podrobněji je rozveden obsah literární rešerše. Hlavní rešeršní kapitola obsahuje popis vegetačních indexů, jako jedné z metod hodnocení zdravotního stavu vegetace. Dále je součástí teoretické části přehled technologických řešení a použitelnost datových sad pro účely praktické části práce. Obsahem praktické části je podrobný popis vývoje webové aplikace.

KLÍČOVÁ SLOVA

webová aplikace, obrazová data, sensorová data, PostgreSQL/PostGIS, vegetační indexy

Počet stran práce: 52

Počet příloh: 6 (z toho 1 volná a 1 elektronická)

ANOTATION

This master thesis is focused on development of web application for raster image analysis combine with sensor data. It means that is a combination of two methods (remote sensing and software engineering). Only open source tools and programs are used for purpose of the thesis. User will be able to get a more accurate result from raster processing data. More accuracy is achieved by implementation the sensor data into application. That is a main aim of the thesis.

A text part is divided in two section. The theoretical part describes a main aim of thesis, methods which are used and a literature review. The main chapter of literature review describes a vegetation indices method, implementation of spatial databases and basic technical information about satellite datasets which are used in practical part. An overview of existing web solutions is also includes in the research chapter. The practical part contains a detailed description of web application development.

KEYWORDS

web application, satellite data, sensor data, PostgreSQL/PostGIS, vegetation indices

Number of pages: 52

Number of appendixes: 6

Prohlašuji, že

- bakalářskou/diplomovou práci včetně příloh, jsem vypracoval(a) samostatně a uvedl(a) jsem všechny použité podklady a literaturu. *(Např. Ve své programové aplikaci jsem použil modul pro transformaci vektorových dat mezi prostorovými referenčními systémy, vytvořený)*

- jsem si vědom(a), že na moji bakalářskou/diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou/diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské/diplomové práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské/diplomové práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užit výsledky a výstupy mé bakalářské/diplomové práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské/diplomové práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne 12. 8. 2019

Štěpán Roubalík

Děkuji vedoucímu práce doc. RNDr. Vilému Pechancovi, Ph.D za podněty, připomínky, a příkladné vedení diplomové práce. Dále děkuji konzultantu Mgr. Tomáši Pohankovi, Ph.D za spolupráci v oblasti serverových technologií a programovacích technik.

Vevázaný originál **zadání** bakalářské/diplomové práce (s podpisem vedoucího katedry a razítkem katedry). Ve druhém výtisku práce je vevázána fotokopie zadání.

Zde je konec prvního oddílu, kde není číslování stránek. Následující strana patří již do druhého oddílu, který má nastaveno číslování stránek.

OBSAH

SEZNAM POUŽITÝCH ZKRATEK	8
ÚVOD	9
1 CÍLE PRÁCE.....	10
2 METODY A POSTUPY ZPRACOVÁNÍ.....	11
2.1 Použité metody	11
2.2 Použitá data	12
2.3 Použité programy	14
2.4 Postup zpracování.....	15
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY	18
3.1 Analýza obrazových materiálů	18
3.1.1 Vegetační indexy	18
3.2 Prostorové databáze	20
3.2.1 Open source řešení	20
3.2.2 Komerční řešení	24
3.3 Využití satelitních a senzorových dat	27
3.3.1 Multispektrální data.....	27
3.3.2 Senzorová data.....	29
3.4 Webové technologie používané pro DPZ	30
4 VÝVOJ APLIKACE	33
4.1 Uložení dat	33
4.2 Frontend aplikace	39
4.3 Komunikace aplikace s databází.....	42
5 FUNKCIONALITA APLIKACE.....	45
5.1 Metadatové informace	45
5.2 Zobrazování dat	46
5.3 Výpočet indexů	49
5.4 Implementace senzorových dat.....	51
6 VÝSLEDKY	55
7 DISKUZE	59
8 ZÁVĚR	61
POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
SEZNAM ILUSTRACÍ	
SEZNAM TABULEK	
PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
AJAX	Asynchronous JavaScript and XML
API	Application Program Interface
CSS	Cascade Style Sheet
DPZ	Dálkový průzkum Země
ELMG	Elektromagnetické spektrum
EPSG	European Petroleum Survey Group
ESA	European Spatial Agency
ESRI	Environmental System Research Institute
ETRS	Earth Resource Technology Satellite
GDAL	Geospatial Data Abstraction Library
GIS	Geografický informační system
HTML	HyperText Markup Language
JPEG	The Joint Photographics Experts Group
KGI	Katedra geoinformatiky
NDVI	Normalized Difference Vegetation Index
NDMI	Normalized Difference Moisture Index
NIR	Near Infrared
OGC	Open Geospatial Consorsium
PHP	Hypertext Preprocessor
RDBMS	Relation Database Management System
RGB	Red Green Blue
RUP	Rational Unified Process
SAVI	Soil Adjusted Vegetation Index
SDE	Spatial Database Engine
SQL	Structured Query Language
SWIR	Short wave infrared
TIFF	Tag Image File Format
UML	Unified Model Language
USGS	United Statets Geological Survey
WKB	Well Know Binary
WKT	Well Know Text

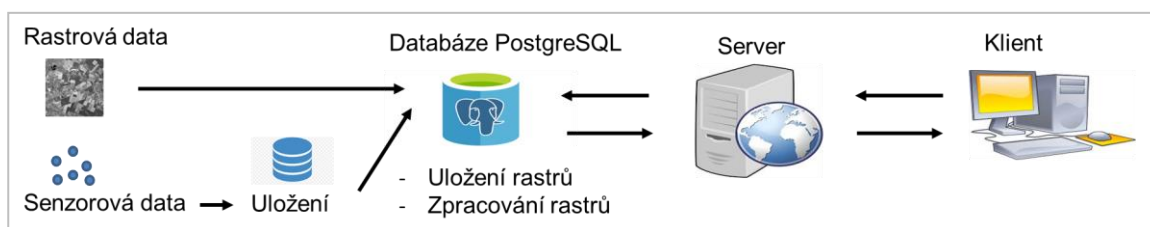
ÚVOD

Oblast Dálkového průzkumu Země prochází stejně jako informační technologie neustálým vývojem. Analýza družicových dat je nejen obohacována o nové teoretické postupy, ale i v technologické části dochází za posledních 10 let k viditelným posunům. Z mnoha zavedených postupů zpracování a integrace dat DPZ jsou v posledních letech identifikovatelné dva směry. Jedná se o rozvoj nástrojů pro výpočty nad leteckými a družicovými daty v tzv. ultra lehkém klientovi. Tedy aplikaci, pro jejichž chod uživatel nepotřebuje žádné specifické softwarové vybavení s výjimkou webového prohlížeče v požadované verzi. Dalším směrem je integrace dat pořizovaných ze senzorů. Data z těchto zařízení jsou schopná v reálném čase poskytovat informace, ze kterých je zjistitelný aktuální stav půdy a okolních podmínek. To přináší novou možnost spojit a zahrnout tato senzorová data do analýz nad daty satelitními. Kombinace družicových a senzorových dat byla naposledy tématem i na Copernicus fóru konaném v červnu 2019 v Praze. Online nástrojů na výpočty s daty DPZ existuje celá řada a slouží pro různé účely. Některé jsou postaveny na komerční bázi. Senzorová data jsou do některých řešení postupně integrována. V České republice představitele tvoří například společnost *Clever Farm*.

Myšlenkou práce je z nabídky jednotlivých open source technologií navrhnout prototyp webové aplikace. Tento prototyp bude mít do budoucna ambice být nadále rozvíjen a poskytovat komplexní řešení pro zpracování satelitních a senzorových dat dohromady. Záměrem je také ověření, jestli je nasazení open source nástrojů na podobný úkol dostatečné a poskytují možnou alternativu k enterprise řešením.

1 CÍLE PRÁCE

Cílem práce je návrh a tvorba prototypu webové orientované aplikace, která bude provádět vybrané analýzy obrazových materiálů a pracovat se sensorovými daty. Webová aplikace bude postavená na bázi komunikace klient server. Uživatel prostřednictvím aplikace bude schopen operovat s daty na straně serveru. V praxi to znamená že, pomocí ovládacích prvků aplikace budou řízeny a zadávány parametry dotazu, který bude následně na aplikačním serveru proveden. Výsledek zpracovaného dotazu bude pomocí webových protokolů předán zpět klientovi (webové aplikaci) jako odpověď zadaný požadavek na server. Hlavní myšlenkou je snaha vytvořit online nástroj, který za využití geoinformačních technologií umožní rychlé a nenáročné zpracování kombinace rastrových a sensorových dat. Stěžejní pro vývoj aplikace je využití open source nástrojů pro jednotlivé programové komponenty. Zdrojová data použitá pro testování funkcionality aplikace budou z větší části také z kategorie otevřených dat. Plánované využití aplikace je v enviromentálních, zemědělských a lesnických odvětvích. Stručné schéma aplikace je uvedeno na obrázku 1.1.



Obrázek 1.1 Obecné schéma aplikace

2 METODY A POSTUPY ZPRACOVÁNÍ

Vývoj webové aplikace zahrnuje několik fází, které vychází z metod softwarového inženýrství. Následující kapitola je věnována problematice použitých metod, zdrojových dat a programů využitých pro vývoj aplikace. Součástí je i popis základního postupu tvorby aplikace a zpracování diplomové práce jako celku.

2.1 Použité metody

Softwarové inženýrství

Softwarové inženýrství je technická disciplína, která se zabývá aspekty produkce softwaru od fáze prvotní specifikace požadavků až po jeho provoz a údržbu (Sommerville, 2013). Toto odvětví je podle Sommerville (2013) klíčové ze dvou důvodů: tvorba pokročilých systémů v odpovídajícím čase i kvalitě a systematický přístup k tvorbě softwaru. Tento uplatňovaný přístup je nazýván softwarovým procesem. Pro všechny softwarové procesy jsou definovány čtyři základní části (specifikace, vývoj, validace, evoluce softwaru). Práce zahrnuje primárně první dva kroky. Pro proces vývoje softwaru je využíváno několik modelů a přístupů. Některé z nich lze i částečně kombinovat. Pro účely a teoretický background v praktické části jsou stručně popsány následující pojmy z odvětví softwarového inženýrství.

Jazyk UML

Jazyk UML (Unified Model Language) je sada 13 různých typů diagramů umožňující modelovat softwarové systémy. Je univerzálně uznávaný jako standardní přístup k vývoji modelů softwarových systémů (SoftwareEngineering.com, 2018).

Proces RUP

Proces RUP je příkladem modelování procesů. Odvozuje se z práce na jazyce UML a souvisejícím procesu Unified Software Development Process (Arlow a Neustadt, 2005). RUP je fázový model, který na rozdíl od vodopádového přístupu (jedna fáze odpovídá jednomu procesu) pracuje s iteracemi fází vývoje. Iterace v RUP procesu identifikuje čtyři typy fází (založení, rozpracování, konstrukce a přechod). I vývoj aplikace k diplomové práci využíval při vývoji iteraci fází u jednotlivých komponent.

Wireframe a prototyp

V prvotní fázi vývoje každého softwaru nebo aplikace je pracováno s pojmy wireframe a prototyp. Wireframe je drátěný model aplikace. Pomocí něj si vývojář dělá na základě předešlé analýzy první obrázek o vzhledu a logické struktuře aplikace. Prototypem je potom chápána raná verze aplikace již s potřebnou funkcionalitou.

Metoda vegetačních indexů

Významnou metodou analýzy rastrových dat je metoda vegetačních indexů, postavená na práci s různými vlnovými délkami pásem elektromagnetického spektra. Základem je využití rozdílné odrazivosti objektů v různých částech barevného spektra. Kombinací těchto pásem lze zvýraznit objekty určitých vlastností. Specifické kombinace těchto barevných pásem jsou dnes známy jako vegetační indexy. K výpočtu indexu musí být k dispozici alespoň dvě pásma rozdílné vlnové délky. Nejčastěji je využíváno rozdílu odrazivosti (reflektance) ve vlnových délkách červené a blízké infračervené části barevného spektra.

2.2 Použitá data

V rámci práce bylo využito několika skupin datových zdrojů. Primárně jsou data rozdělena na satelitní, získaná pomocí družic, a sensorová, získaná automatickým měřením pomocí bezdrátové sensorové sítě umístěné v zájmové oblasti. Stručný popis konkrétních datových sad je obsahem následující podkapitoly. Podrobné technické specifikace datových sad, způsob jejich implementace a manipulace s metadaty jsou obsahem podkapitoly 3.3 jako součást teoretické části práce, a tvoří i součást přílohy 1. Zájmová oblast pro umístění měřicího senzoru byla původně plánována pro oblast „Velké pole“ u Holešova. Z legislativních důvodů však toto umístění nebylo možné pro testování použít. Jako náhradní varianta byla využita sensorová zařízení umístěná v blízkosti Katedry geoinformatiky. Důsledky této neplánované změny na výsledky zpracovaných dat jsou jedním z bodů závěrečné diskuze.

Multispektrální data

Pro analýzu obrazových materiálů byla využita data z klasických multispektrálních satelitních senzorů. V tomto případě šlo především o volně dostupná data ze systémů Sentinel-2 a Landsat 8. Byla použita i data z Landsat 7. Kvůli poruše páskování na této družici nebylo ve zvoleném časovém období (červen 2019) možné pořídit data nad stejnou zájmovou oblastí jako data z předchozích dvou družicových systémů. Také odpadla možnost propojit tento typ dat s daty sensorovými.

Systém Sentinel-2 je zástupce nekomerční sféry jako součást projektu Copernicus pod správou Evropské kosmické agentury (ESA) a Evropské komise. Data ze satelitů Sentinel jsou zdarma poskytována na Scientific Data Hub (<https://scihub.copernicus.eu/dhus/#/home>) nebo v prohlížeči Earth Explorer (<https://earthexplorer.usgs.gov/>). V rámci mise Sentinel-2 obíhají po stejné orbitě dvě identické družice. Senzory těchto družic snímají zemský povrch a poskytují satelitní snímky, pokrývající významnou část elektromagnetického spektra, s vysokým prostorovým rozlišením v poměrně krátké časové periodě. Prostřednictvím multispektrálních senzorů jsme schopni data ze Sentinelu 2 zaznamenat v 13 spektrálních pásmech od viditelné části ELMG spektra přes blízkou infračervenou část (NIR) po střední infračervenou část (SWIR). Čtyři pásma jsou v prostorovém rozlišení 10 metrů, šest pásem v prostorovém rozlišení 20 metrů a tři pásma v prostorovém rozlišení 60 metrů (Copernicus.gov, 2018). Přehled všech pásem Sentinelu 2, jejich vlnových délek s prostorovým rozlišením, je uveden v příloze 1.

Druhým zástupcem nekomerční otevřené sféry mezi družicovými daty je systém Landsat 8. Provozovatelem je od jeho počátku v roce 1972 americká vládní agentura United States Geological Survey (USGS). Sesbíraná data jsou zdarma ke stažení v aplikacích EarthExplorer (<https://earthexplorer.usgs.gov/>), GloVis (<https://glovis.usgs.gov/>) nebo LandsatLook Viewer (<https://landsatlook.usgs.gov/>). V intervalu 24 hodin po pořízení snímků jsou nově pořízená data v těchto rozhraních dostupná. Časová perioda snímání povrchu Země u tohoto družicového systému je 16 dnů s offsetem 8 dnů od Landsat 7. Pomocí submodulů OLI a TIRS jsou pořizovány snímky v prostorovém rozlišení 15 a 30 metrů na pixel ve viditelné, blízké infračervené a střední infračervené části ELMG spektra. Jedná se o snímky multispektrální nebo panchromatické. Spektrální pásma jsou pořizována jako 16 - bitové rastry (USGS, 2018).

Senzorová data

Senzor (čidlo) je zařízení, které snímá definované fyzikální veličiny. Data fyzikálních charakteristik (teplota, vlhkost, srážky) jsou sbírány pomocí bezdrátové senzorové sítě a ukládány do databáze. Obecně existují dva základní typy senzorů, a to analogové a digitální senzory (Pechanec 2014). Čidlo je vždy připojeno k měřicímu zařízení (Julina, 2016).

Základním údajem o senzoru je jeho identifikační číslo ID. Tento údaj slouží k jednoznačné identifikaci senzoru, resp. jeho měřených dat, od měření, přes přenos, až po uložení do databáze. ID by tedy mělo být uloženo v měřicí jednotce a odesláno při každém přenosu. Další informace slouží k podrobnějšímu popisu senzorů, případně k jednodušší orientaci mezi senzory např. v mapovém podkladu. Tyto informace mohou být uloženy v měřicí jednotce a k dispozici na vyžádání (možnost vyčtení všech údajů o senzoru v terénu). Alternativně je možné mít tyto informace pouze v databázi na serveru. Jedná se zejména o tyto údaje: textový popis senzoru, ID nebo popis lokality, poloha GPS, fyzikální jednotka měřených dat, název výrobce, typové označení výrobce, typová verze senzoru, sériové číslo senzoru, limitní hodnoty apod. (Pechanec a kol., 2017).

Typ údaje	Velikost dat [byte]	Poznámka
ID (16bit)	2	
ID (32bit)	4	
ID	3	[číslo sítě].[číslo jednotky].[číslo senzoru]
čas	6	datum 3byte + čas 3byte
časová značka	4	unix timestamp
číslo měření (16bit)	2	číslo měření sloužící jako časová značka

Obrázek 2.1. Velikost základních informací o senzoru (Julina, 2016)

Typ údaje	Velikost dat [byte]	Poznámka
popis senzoru	8 - 32	text
ID lokality	2	binární 16bit
popis lokality	8 - 16	text
poloha GPS	10 / 22	binární / text
fyzikální jednotka	8 - 16	text
název výrobce	8 - 16	text
typové označení	8 - 16	text
typová verze	3 - 6	text
sériové číslo	4 / 6 - 16	binární 32bit / text

Obrázek 2.2. Velikost doplňkových informací o senzoru (Julina, 2016)

2.3 Použité programy

V souladu se zadáním práce byly využity pouze programy publikované pod volně šiřitelnou licenci. Pro databázový server bylo vybráno řešení *PostgreSQL* ve verzi 11. Pro umožnění analýz nad prostorovými daty bylo do databáze přidáno rozšíření *PostGIS* ve verzi 2.4.4. Komunikace mezi klientskou aplikací a databází zajišťuje webový server *Apache*.

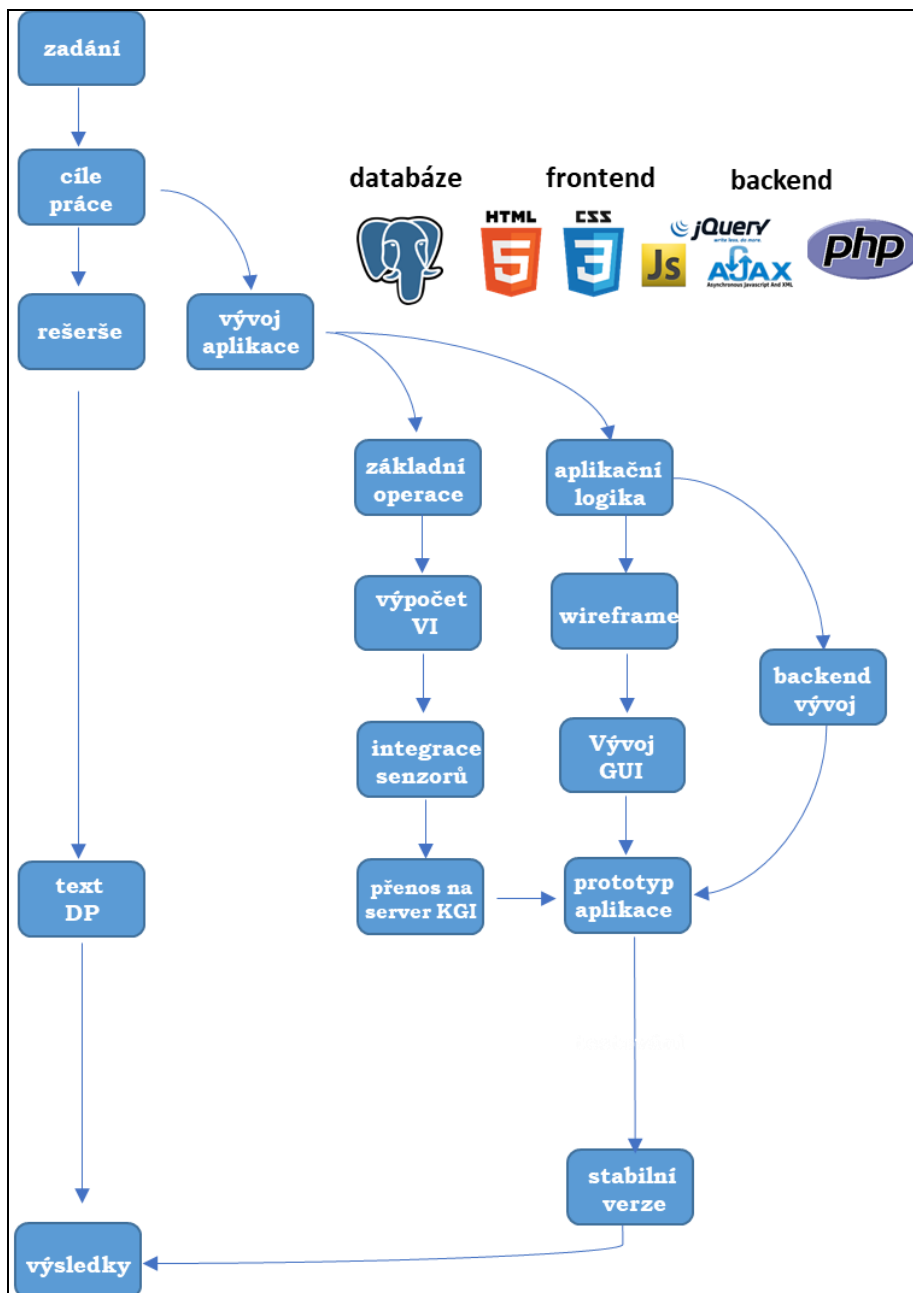
Procesní část na straně serveru vykonávají funkce skriptovacího jazyka *PHP*. Na straně klienta jsou využity tři technologie používané pro tvorbu libovolné webové aplikace. Značkovací makrojazyk *HTML* pro obsahovou část aplikace, kaskádové styly *CSS* pro definici vzhledu aplikace a *Javascript* pro interaktivní ovládání aplikace. V rámci kaskádových stylů je použita knihovna *Bootstrap* ve verzi 3.3.7. Pro tvorbu mapové části aplikace zahrnuje v část *Javascriptu* integraci knihovny *Leaflet* a dalších externích zdrojů. Za účelem vyhnout se kompletnímu načítání obsahu webové aplikace vždy při vytvoření nového dotazu na data byla využita technologie *AJAX*. Využity jsou v tomto případě knihovny napsané v *Javascriptu* podporující asynchronní načítání obsahu webových stránek. Mimo této výhody načítání přináší využití této technologie také několik nevýhod. Například při špatně zvoleném použití může výrazně stoupnout počet http požadavků na server a dochází potom k jeho přetížení.

Při zpracování byly okrajově použity i desktopové GIS programy. *QGIS* pro kontrolní zobrazování obsahu *PostgreSQL/PostGIS* databáze a *SNAP Desktop* pro částečný pre-processing rastrových dat. Veškeré prováděné změny byly ihned verzovány pomocí nástroje *Git*. Textová a grafická část práce byla vypracována pomocí proramů z balíku *Microsoft Office 2016* pod univerzitní licenci.

2.4 Postup zpracování

Vypracování praktické části je rozděleno na několik částí schematicky znázorněných na obrázku 2.3. Po zadání práce a prvotní teoretické rešerši zadané problematiky přišla na řadu analýza požadavků na technologické řešení aplikace. Bylo rozhodnuto, na jakých technologiích budou jednotlivé části (databáze, serverové komponenty, frontend) prototypu vyvíjeny. Analýza zohledňovala následující požadavky, ze kterých nakonec vzešly použité technologie uvedené v předešlé podkapitole:

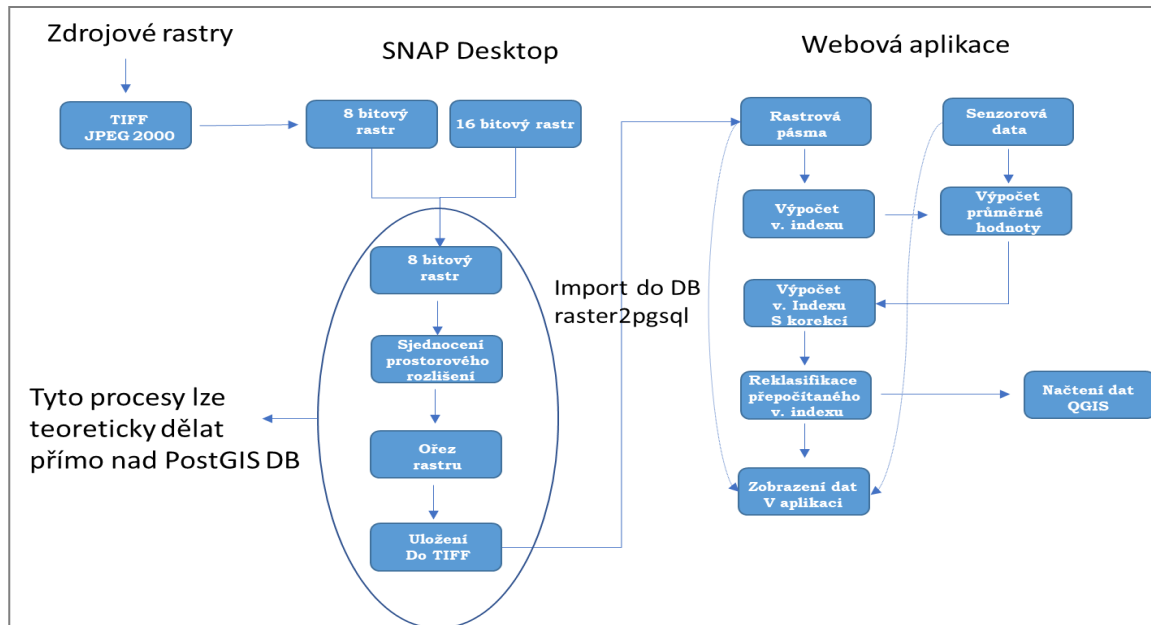
- jedná se o open source řešení,
- schopnost zpracovat prostorová data,
- možnost propojení s webovou stránkou na frontendu.



Obrázek 2.3. Postup zpracování teoretické a praktické části

Workflow zpracování dat

Proces zpracování dat začíná již samotným stažením z výše uvedených zdrojů. Rastrové snímky byly ve výchozí podobě předzpracovány v programu *SNAP Desktop* tak, aby po jejich importu do databáze byl zjednodušený proces dalších výpočtů. Předzpracování znamenalo sjednocení prostorového a spektrálního rozlišení rastrů. Pro rychlejší import bylo přidáno i zmenšení extenty rastru. Po importu dat do *PostGIS* databáze již bylo přistoupeno k samotnému výpočtu vegetačních indexů. V první fázi se jednalo pouze o práci se samotnými pásmy rastrů, ve druhé fázi do výpočtu vstoupila i data ze senzorů. Stručné schéma celého procesu je součástí obrázku 2.4.



Obrázek 2.4 Workflow zpracování dat

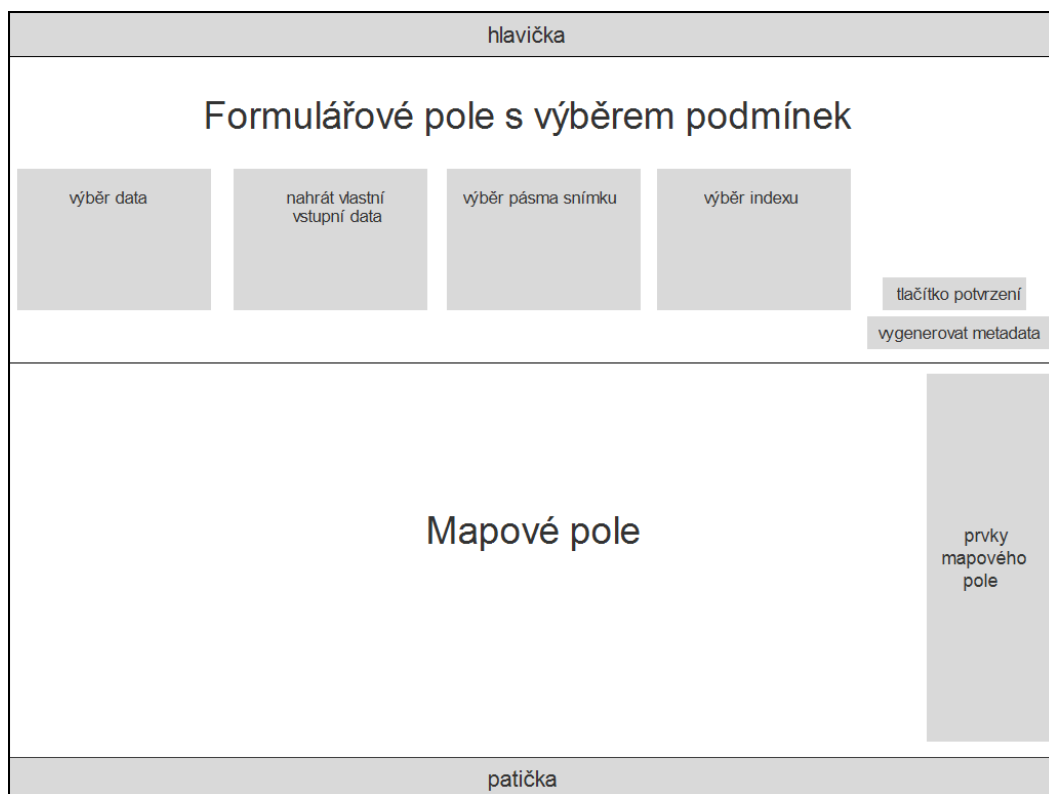
Návrh databáze

Po vybrání databáze PostgreSQL následoval návrh tabulek pro uložení dat v databázi a naplnění těchto tabulek testovacími daty. Struktura tabulek u rastrových dat byla odvíjena od možností importu rastrových dat. U sensorových dat bylo schéma tabulek převzato z již zřízené a provozované databáze na webovém serveru Katedry geoinformatiky. Schémata tabulek v databázi se podrobněji zabývá kapitola 4.1. Import rastrových dat do databáze byl proveden s využitím nástroje *raster2pgsql.exe*, který je součástí instalace databáze *PostgreSQL*.

Po vyřešení ručního importu dat do databáze se ve vývoji přešlo na tvorbu prostorových dotazů, prováděných přímo na databázovém serveru. Pro tvorbu dotazů byl použit jazyk SQL s nabízenými konstruktory zvolené databáze. Obecně se jednalo o dotazy zajišťující výběr jednotlivých záznamů z tabulek obsahující sensorová data a dotazy na zpracování rastrových dat (získání metadat, výpočetní operace nad rastry).

Návrh uživatelského rozhraní

Souběžně s tvorbou SQL dotazů nad databází byl rozpracován i prvotní návrh uživatelského rozhraní. Základní wireframe (obrázek 2.5) vycházel ze záměru obsáhnout dvě hlavní části aplikace (mapové pole, ovládací panel). Postupem času došlo ke změně, kdy postranní panel v základním načtení byl předělán na rozbalovací. Z upraveného wireframu byl v dalších krocích postupně dotvářen finální vzhled rozhraní aplikace. Více se tímto zabývá kapitola 4.2.



Obrázek 2.5 Prvotní wireframe aplikace

Vývoj backendu aplikace

Navázání vytvořených prostorových SQL dotazů na webové rozhraní aplikace bylo dalším krokem ve workflow praktické části. Všechny procesy na straně serveru byly pomocí skriptovacího jazyka *PHP* a technologie *AJAX* navázány na interaktivní prvky webového rozhraní, naprogramovaných pro změnu pomocí jazyka *Javascript*. Programování procesů na straně serveru úzce korespondovalo s vývojem na straně Frontendu.

Přenos z localhostu na server

Prakticky celý vývoj aplikace probíhal v prostředí na localhostu. K přenosu konfiguračních souborů došlo až v samotném závěru praktické části. Přenos a nastavení databáze byl proveden přes webové rozhraní *pgAdmin 4*. V prvním kroku bylo již dříve vytvořené schéma databáze s rastrovými daty v lokálním prostředí pomocí příkazu *Backup* vyexportováno. Následovalo nahrání tohoto obsahu databáze na databázový server KGI pomocí příkazu *Restore*. Přenos zahrnoval veškeré dříve provedené nastavení, vytvořené uživatele a obsah databáze. Tato část rovněž obsahovala editaci zdrojového kódu konfiguračních souborů, kde byly změněny přístupové údaje k databázím provozovaným na katederním serveru místo lokálně používané databáze.

Zpracování výsledků

Po dokončení praktické části následovala sumarizace výsledků, kompletace textové části a příloh práce. Dále potom vytvoření repozitáře pro zdrojový kód na úložišti *Github*, a sepsání manuálu pro import dat do databáze pomocí *raster2pgsql.exe* a vytvoření návodu pro práci v aplikaci.

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Následující kapitola je věnována teoretickému shrnutí nejdůležitějších poznatků z oblasti Dálkového průzkumu Země a prostorových databází, které jsou klíčové pro vývoj aplikace v praktické části. Nejprve jsou popsány vegetační indexy využitelné primárně pro oblast zemědělství a obecná metodika jejich výpočtu. Následuje přehled vybraných databázových systémů s důrazem na jejich prostorové rozšíření. Další část je věnována technickým parametrům jednotlivých datových sad, použitých jako vstupní data pro analýzu. Poslední kapitola teoretické části je věnována přehledu existujících řešení v oblasti webových technologií využívaných k analýze obrazových materiálů.

3.1 Analýza obrazových materiálů

Pořizováním, zpracováním a analýzou obrazových materiálů se zabývá dálkový průzkum Země. Podle Dobrovolného (1998) se jedná o metodu zjišťování informací o objektech a jevech na dálku – bez přímého kontaktu s těmito jevy a objekty. Vymezením pojmu se zabývalo více českých i zahraničních autorů Čapek (1988), Campbell (1996), Lillesand, Kiefer (1994), Alföldi et al. (1993), Zemek et al. (2014), Žizala (2016), Galambošová (2016), Lukeš (2018) a další. Důležitou vlastností pro analýzu objektů a jevů pomocí dálkového průzkumu Země je spektrální chování objektů. Sluneční záření dopadající na zemský povrch je z části absorbováno, zčásti vedeno do nižších vrstev půdy a z části odraženo zpátky do atmosféry. Množství odraženého záření je možné charakterizovat tzv. spektrální odrazivostí, tj. poměr odraženého a dopadajícího záření na určité vlnové délce (Dobrovolný, 1998). Spektrální charakteristika vyjadřuje průběh závislosti odrazivosti a emisivity dané látky na vlnové délce. Tato závislost je pro různé typy povrchů definována tzv. spektrální křivkou odrazivosti (Halounová, 2005). Z křivek spektrálního chování lze určit vlnovou délku, která je vhodná k identifikaci daného objektu. Pro praktickou část jsou klíčové spektrální charakteristiky půdy a především vegetace.

Spektrální projev půdy závisí na obsahu biotických a abiotických složek půdy. Spektrální vlastnosti půdy určují především minerální složení, půdní vlhkost, obsah organických látek a textura půdního povrchu. Vegetační složka je zastoupena prakticky na všech obrazových snímcích. Odrazové vlastnosti vegetace jsou formovány pomocí uspořádání vegetačního krytu, vodního obsahu, zdravotního stavu vegetace a půdního substrátu. Spektrální chování vegetace tvoří základ pro koncept spektrálních vegetačních indexů, které dávají do poměru odrazivost vegetační složky ve viditelné a blízké infračervené části elektromagnetického spektra.

3.1.1 Vegetační indexy

Jedna z metod zabývajících se biofyzikálními vlastnostmi vegetace za využití dálkového průzkumu Země je metoda vegetačních indexů. Metoda poskytuje vyhodnocení stavu vegetace za použití zvýraznění specifické spektrální odrazivosti vegetace. Vegetačními indexy lze charakterizovat fyziologické stresy rostlin, obsah vody, chemické a mechanické poškození nebo průběh fenofáze. Dobrovolný (1998) dále uvádí: „*Některé z vegetačních indexů vypovídají i o vlastnostech půdního substrátu*“. Oproti jiným metodám DPZ má výhodu, je výpočetně relativně jednoduchá a lépe univerzálně aplikovatelná (Frampton, 2013). Podle způsobu výpočtu lze vegetační indexy dělit na diferenční, poměrové a ortogonální (Halounová, 2005).

NDVI – Normalized Difference Vegetation Index

NDVI je jeden z nejčastěji používaných vegetačních indexů. Jedná se o poměrový vegetační index, který využívá spektrální pásma blízké infračervené a viditelné červené části elektromagnetického spektra. Poprvé byl tento index použit na univerzitě v Texasu během projektu, při kterém byl zkoumán efekt zelené vlny během vegetačního období v Oblasti Velkých planin. Bylo zjištěno, že při správné kombinaci pásem 5 (Red) a 7 (NIR – blízké infračervené záření) u systému ETRS-1 (Earth Resources Technology Satellite) výsledné hodnoty korelují se stavem vegetace (Rouse a kol., 1974). Kombinace pásem je vyjádřena rovnicí:

$$\text{NDVI} = (\text{NIR}-\text{RED})/(\text{NIR}+\text{RED}) \quad (1)$$

Index bude použitý pro výpočet v praktické části. Pásma Sentinelu-2 a Landsatu 8 a jejich použití pro výpočet indexů jsou obsahem následujících kapitol.

NDMI – Normalized Difference Moisture Index

Index vlhkosti (NDMI) normalizovaného rozdílu je index odvozený z pásem blízkých infračervených (NIR) a krátkovlnných infračervených kanálů (SWIR). Odrazivost SWIR odráží změny jak obsahu vegetační vody, tak i houbové mezofylové struktury ve vegetačním krytu. Zatímco odrazivost NIR je ovlivněna vnitřní strukturou listů a obsahem sušiny listů, nikoli však obsahem vody. Kombinace NIR a SWIR odstraňuje odchylky vyvolané vnitřní strukturou listů a obsahem sušiny listů, což zlepšuje přesnost při získávání obsahu vegetační vody. Množství vody dostupné ve vnitřní struktuře listů ovlivňuje spektrální odrazivost ve SWIR intervalu elektromagnetického spektra.

$$\text{NDMI} = (\text{SWIR}-\text{NIR})/(\text{SWIR}+\text{NIR}) \quad (2)$$

SAVI – Soil Adjusted Vegetation Index

Index SAVI byl odvozen v roce 1988 z NDVI pro minimalizaci odrazivosti půdy. Odrazivost půdy je limitována L faktorem půdy, který je v indexu obsažen a nabývá hodnot v intervalu 0 až 1. Huete (1988) uvádí fakt že: „*nízká vegetace dosahuje hodnot blízkých $L = 1$, pro střední vegetaci se hodnoty blíží 0,5 a u husté vegetace je počítáno s hodnotou 0,25.*“ Pechanec a kol. (2017) zmiňují: „*že pro většinu případů je přípustná hodnota faktoru $L = 0,5$.*“ Pokud je hodnota faktoru 0, jsou hodnoty SAVI velmi podobné hodnotám z NDVI. Ale pro velké procento případů platí, že výsledné hodnoty SAVI jsou menší než u poměrového indexu NDVI. Podle Qi a kol. (1994) to značí: „*že SAVI dokáže eliminovat samotnou odrazivost půdy*“. Index bude použitý pro výpočet v praktické části pro pásma Sentinelu-2, Landsatu 7 a Landsatu 8. Obecná rovnice SAVI potom zní:

$$\text{SAVI} = (\text{NIR}-\text{RED})/(\text{NIR}+\text{RED}+L) \times (1+L) \quad (3)$$

3.2 Prostorové databáze

Pojmem prostorová databáze lze definovat jako databázi, ve které jsou uložena i prostorová data, případně informace o objektech z reálného světa reprezentované prostorovými daty (VÚGTK, 2016). Obe a Hsu (2015) definují prostorovou databázi nejen jako úložný prostor pro data, ale i jako významný analytický a organizační nástroj pro řešení geoprostorových úloh. Prostorová databáze může tvořit nadstavbu nad klasickým databázovým systémem. Lze ji klasifikovat různými způsoby. Prvním kritériem je dělení na komerční a open source databáze. Mezi komerční řešení lze zařadit Oracle, IBM DB2 databáze, včetně jejich prostorových nástaveb. Naopak za významné open source řešení jsou považovány systémy PostgreSQL, SQLite, MySQL a další. Druhým způsobem lze dělit databáze na skupinu založenou na dotazovacím jazyku SQL, kam patří všechna výše zmíněná řešení, a prostorové NoSQL databáze. Práce se zabývá v praktické části nejvíce využitím databázového systému PostgreSQL a jeho prostorovou extenzí PostGIS a je mu v teoretické části věnována největší pozornost. Jsou zde ale zmíněny i další příklady prostorových databázových systémů.

3.2.1 Open source řešení

PostgreSQL

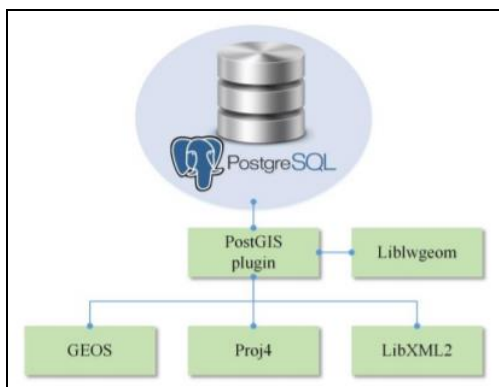
PostgreSQL je výkonný objektově relační databázový systém s otevřeným zdrojovým kódem vyvíjený v rámci projektu na University of California v Berkeley od roku 1986. Postupně se funkcionalitou a spolehlivostí vyrovnal nebo předčil komerční databázové systémy. Běží nativně na všech rozšířených operačních systémech jako například Windows, Linux, macOS, Solaris nebo BSD (postgre.org, 2018). Systém splňuje soubor vlastností ACID zaručujících spolehlivé zpracování transakcí, podporuje cizí klíče, operace JOIN, pohledy, trigery a uložené procedury. Obsahuje většinu datových typů používaných v SQL. Tento systém má nativní programovací rozhraní pro řadu jazyků (Perl, Python, PHP, C++, Java, .Net, Ruby a další). V neposlední řadě PostgreSQL obsahuje řadu rozšíření jako například prostorovou extenzi PostGIS a další, díky kterým je PostgreSQL velmi často využívané řadou organizací nejen pro potřeby geoinformačních technologií. Systém PostgreSQL prochází neustálým vývojem. V současné době je aktuální verze 10.x a v beta vývoji verze 11. Starší verze s označením 9.3–9.6 mají plánovaný End of Life mezi lety 2018 až 2021 (obrázek 3.1).

Release	Release date	End of life
PostgreSQL 9.6	September 29, 2016	September, 2021
PostgreSQL 9.5	January 7, 2016	January, 2021
PostgreSQL 9.4	December 18, 2014	December, 2019
PostgreSQL 9.3	September 9, 2013	September, 2018

Obrázek 3.1 End of Life verze PostgreSQL 9 (Postgre.org, 2017)

Prostorová extenze PostGIS

PostGIS reprezentuje open source geodatabázi, tvořící nadstavbu nad objektově relační databází PostgreSQL (Obe a Hsu, 2015). Hlavním přínosem extenze je rozšíření funkcionality databázového systému o možnost ukládat prostorová data a provádět prostorové dotazy v SQL jazyce. PostGIS je primárně vyvíjen firmou Refraction Research Inc. s otevřeným zdrojovým kódem pod licencí *GNU General Public Licence* a zároveň implementuje standardy *Simple Features for SQL Specification* organizace OGC (OpenGIS Implementace pro Simple feature, 2010). OGC definuje dva základní způsoby vyjádření prostorových objektů WKT (Well Known Text) a WKB (Well Known Binary). Funkcionalita PostGIS je zvyšována dalšími významnými knihovnamy pro prostorová data jako Proj4 a GDAL (obrázek 3.2). Společně s těmito projekty spadá do skupiny OSGeo. Od verze 0.9 PostGIS podporuje všechny prvky a objekty definované v OGC Simple Features for SQL. Díky PostGIS je databáze PostgreSQL schopná ukládat, a především pracovat s rastrovými daty. Extenzí PostGIS je podporován dvojí způsob uložení rastrů tzv. *in-database* a *out-database* (Obe a Hsu, 2015). Autoři také zdůrazňují, že obojí uložení má výhody i nevýhody.



Obrázek 3.2 Propojení databáze Postgre a prostorové extenze PostGIS (Postgre.org, 2018)

Práce s rastry v PostGIS

Při použití způsobu uložení *in-database* jsou uvnitř databáze uloženy samotné pixely rastru v nativním formátu PostGIS rastr. Rastry mohou být uloženy ve dvou strukturách. Prvním způsobem je uložení kdy jedna tabulka odpovídá jednomu rastru a obsahuje pouze jeden řádek s datovým typem rastr. Druhou možností je rozdělení struktury na dlaždice kdy tabulka obsahuje jednotlivé dlaždice rastru (jedna dlaždice znamená jeden záznam v tabulce). Dohromady dává tabulka jeden celek (jeden rastr). Z pohledu webových technologií podle Obe a Hsu (2015) je lepší použít systém kdy jsou rastry rozděleny do dlaždic o rozměrech 50×50 až 400×400 pixelů. Smyslem dělení na dlaždice je zrychlení prostorových operací, manipulace a načítání dat. Import kromě samotných dat umožňuje nahrávat i rastrové pyramidy, které jsou v databázi uloženy jako další tabulky s rastry. Velikost pixelu těchto náhledů je úměrně přizpůsobena hladině náhledu. Výhody tohoto uložení lze shrnout do následujících bodů:

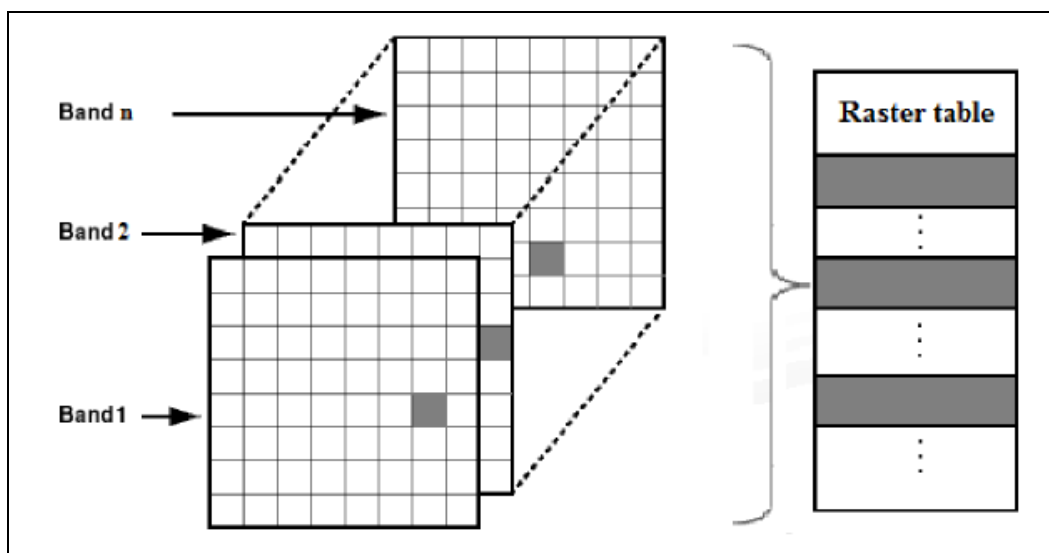
- rastry využívají všech vlastností relačních databází,
- způsob uložení je více otestovaný než *out-database*,
- zajištění plné transakční integrity při editaci rastrů,
- rychlejší čtení dat a procesní zpracování,
- záloha rastrů společně s databází (Obe a Hsu, 2015).

Uložení rastru způsobem out-database je založeno na souborovém uložení dat mimo databázi. V databázi je uložen pouze indexovaný geografický rozsah a cesta k rastrovým datům v souborovém systému. Řadu nevýhod převažující nad výhodami tohoto způsobu uložení uvádí kromě Obe a Hsu (2015) také Racine (2011) nebo Blasby (2001). Za největší nevýhody je bráno:

- ztráta všech výhod uložení v databázi,
- rastry nemají uloženou vazbu na databázi, po smazání rastru v systému je zneplatněna cesta z databáze k souboru,
- přístupová práva k souborům,
- pomalejší analýza dat.

Parametry rastru

Rastrová data jsou charakterizována velikostí pixelu, počtem sloupců a řad tvořících grid, počtem spektrálních pásem a většinou je definován souřadnicový systém rastru. Každý rastrový snímek od panchromatických až po hyperspektrální je složen z jednotlivých kanálů (bandů), přičemž je vždy zapotřebí nejméně jeden kanál. Od verze PostGIS 2.0 je umožněno načítat do databáze rastry s více než 255 kanály (Obe a Hsu, 2015). Pixely mají pro každé pásmo rozdílné hodnoty (obrázek 3.3), na které se lze dotazovat pomocí funkce *ST_Value*. Tato funkce na výstupu vrací hodnotu pixelu pro definovanou oblast a barevné pásmo. Za předpokladu že jsou do PostGIS nahrány georeferencované rastry je kód pro souřadnicový systém spárovaný s tabulkou SRID obsaženou v PostGIS. Kódy obsažené v tabulce Spatial Reference Raster Identifier (SRID) jsou převzaty z EPSG kódů souřadnicových systémů. Lze provádět transformaci souřadnicových systémů přímo v PostGIS pomocí příkazu *ST_Transform*.



Obrázek 3.3 Uložení vícepásmového rastru v PostGIS (PostGIS.net, 2018)

Načítání metadat z hlavičky rastru v PostGIS

Pro načítání rastrových dat do PostGIS databáze se využívá nástroj *raster2pgsql.exe*. S podporou GDAL knihovny je možné načíst širokou škálu rastrových formátů. Správnou funkčnost lze v příkazovém řádku ověřit pomocí příkazu *raster2pgsql -G*, který vypíše seznam všech podporovaných rastrových formátů pro import do PostGIS. Z prostorových indexů pro geodata je využíván RTree Index implementovaný jako GiST. Umožňuje indexování i 3D prostorových dat.

Pro zjištění základních metadat rastru, uložených přímo v hlavičce, lze použít příkaz *ST_Metadata* nebo *ST_BandMetadata*. Mezi základní metadata patří:

- velikost pixelu,
- rotace rastru,
- pozice levého horního pixelu,
- souřadnicový systém rastru,
- počet pásem rastru.

Nelze z hlavičky rastru zjistit údaje o vlnové délce rastru, datu pořízení nebo úrovni zpracování. Tato metadata musí být zjišťována externě z přiložených metadataových souborů datových sad.

SQLite – Raster Lite

SQLite je relační databázový open source systém napsaný v knihovně jazyka C. Na rozdíl od jiných řešení není založen na principu klient-server. Databázový server je spuštěn jako samostatný proces a každá databáze je uložena v samostatném souboru *.dbm. Pro SQLite je implementovaný prakticky celý standard SQL92. Předkompilovaná databáze je k dispozici pro Linux, Mac OS i Windows. Podobně jako PostGIS pro databázový systém PostgreSQL i v SQLite je operace s rastrovými daty je možná na základě rozšíření s názvem RasterLite. Prostorové rozšíření pro vektorová data nese označení Spatial Lite.

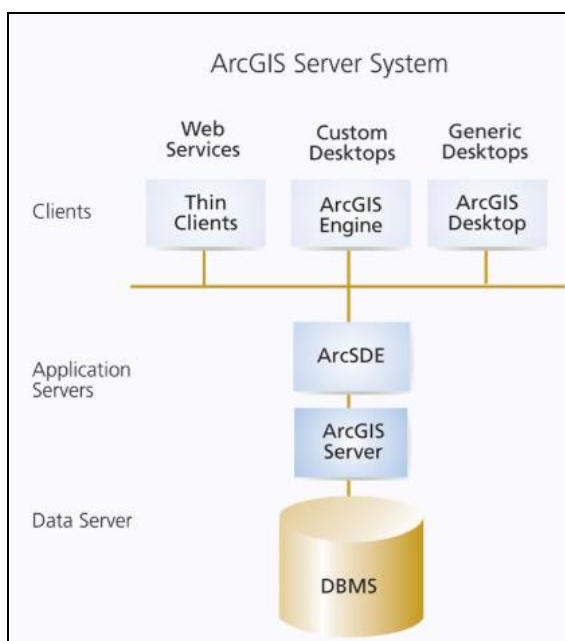
MySQL – MyGIS

Systém MySQL zahrnoval prostorovou extenzi poprvé ve verzi 4 (Obe, Hsu, 2015). Oproti konkurenčním systémům obsahuje řadu nevýhod pro prostorová data. V první fázi MySQL podporovalo minimum geometrických operací. Velmi podstatným faktorem pro načítání dat byla absence prostorových indexů. Prostorové indexy začali být podporovány až od verze MySQL 5.6. Podle Obe (2015) lze konstatovat, že pro řešení komplexnějších prostorových dotazů a operací je databáze MySQL nedostačující. Pro účely práce je vyloučeno použití MySQL pro absenci podpory rastrových dat.

3.2.2 Komerční řešení

ArcSDE

ArcSDE (Spatial Database Engine) je technologické řešení firmy Esri pro ukládání a správu prostorových dat v prostředí zvoleného relačního databázového systému (esri.com, 2018). Z hlediska architektury klient/server funguje ArcSDE jako tzv. middleware, který zajišťuje komunikaci mezi klientem a serverem. Požadovaná prostorová data uložená v databázi jsou pomocí middleware zasílána například do ArcGIS for Desktop, ArcGIS online i dalších aplikačních řešení společnosti Esri. Systém umožňuje rozložení úloh mezi podporovaný databázový server, uživatelskou aplikaci a ArcSDE aplikační server. Výhodou této třívrstvé architektury je vysoká výkonnost celého systému. Limitujícím faktorem je vysoká pořizovací cena. Základní balíček služeb ArcGIS for Server začíná na hodnotě 10 000 dolarů a roste s rozšiřováním základní funkcionality.

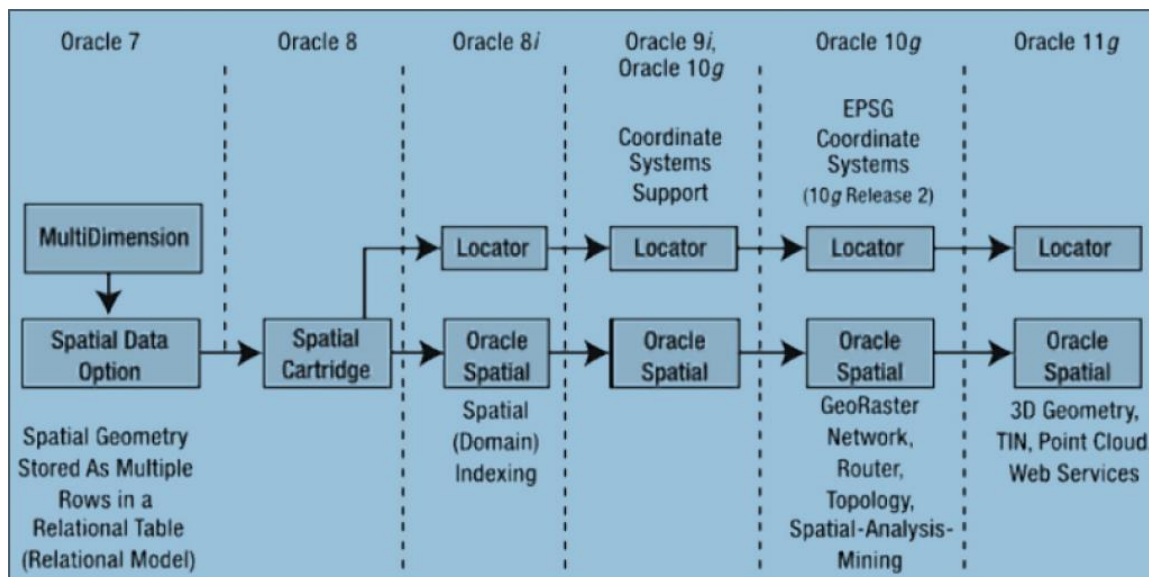


Obrázek 3.4 Schéma ArcSDE (esri.com, 2017)

Správa dat je v ArcSDE možná ve formátech RDBMS (Oracle, MS SQL Server, Informix, IBM DB2 a nově také PostgreSQL). Pro ukládání dat je používán unifikovaný datový model, který uchovává prostorová a atributová data v jedné databázi RDBMS. Prostorové objekty tak mohou být načteny společně s atributovými daty pomocí jediného SQL dotazu. Při ukládání rastrových dat ArcSDE podporuje výpočet tzv. pyramidových vrstev. Při použití pyramidových vrstev je původní raster rozdělen v jednotlivých úrovních na určitý počet dlaždic. Velikost dlaždice je závislá na několika parametrech (bitová hloubka rastru, rychlost síťového připojení, nastavení databáze). Nastavení velikosti rastrových dlaždic obecně závisí na provozovateli databáze. Obe a Hsu (2015) uvádí že: „Optimální rozměry rastrových dlaždic v PostGIS jsou 50×50 až 400×400 pixelů pro potřeby webových služeb“. Výchozí hodnota velikosti dlaždice v ArcSDE je 128x128 pixelů. Vytvářením pyramid je obecně zrychlováno načítání v klientské aplikaci. Rastrová data lze v ArcSDE ukládat v komprimované podobě pomocí podporovaných kompresí (LZ77, JPEG a JPEG 2000) i v nekomprimované podobě.

Oracle Spatial

Společnost Oracle patří mezi hlavní členy OGC. Nadstavba Oracle Spatial poskytuje SQL schéma a funkce pro ukládání, aktualizaci a dotazování prostorových dat v Oracle databázi (Murray, 2009). První verze Oracle Spatial byla vydána s verzí Oracle 8iR1. Podmnožinou Oracle Spatial je geografický databázový systém Oracle Locator spadající pod licenci Oracle Database Enterprise Edition. Pro používání Oracle Locatoru je zapotřebí licence Oracle Spatial.

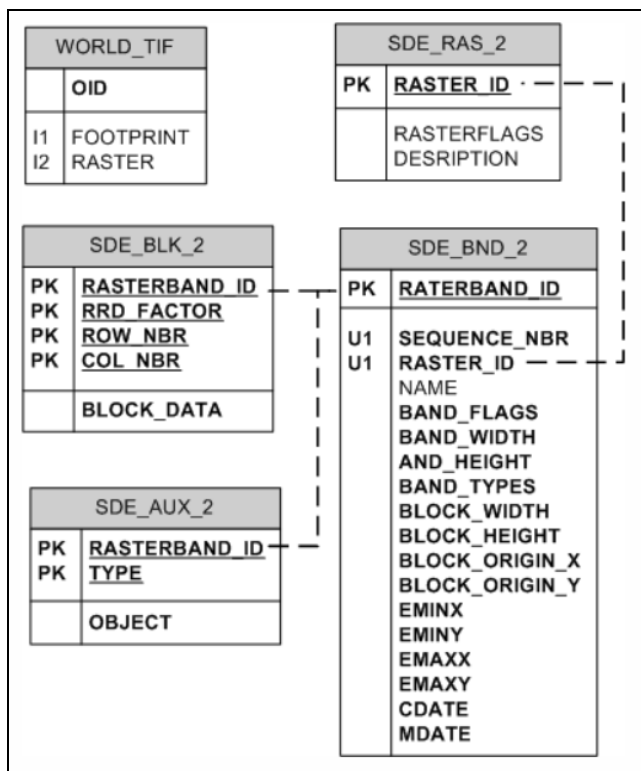


Obrázek 3.5 Vývojová větev Oracle Spatial od verze 7 (oracle.com 2018)

Pro ukládání rastrových dat je v řešení Oracle implementován Georaster objektově relační model. V tomto modelu jsou pixely dány do bloků a každý blok je ukládán jako jeden záznam do rastrové tabulky. Stejně jako u PostGIS lze pro import rastrů do databáze použít knihovnu GDAL. Pro indexování jsou použity indexy QuadTree a R strom. Oracle Spatial má oproti PostGIS slabší podporu ve standardech webových služeb (podporovány jsou WMS, WFS a WCS). Procedurální zpracování SQL příkazů je v Oraclu pomocí procedurálního jazyka PL/SQL (Oracle Corporation, 2018).

IBM DB2

Dalším komerčním řešením jsou databázové systémy společnosti IBM. Produkt IBM® Db2® v aktuální verzi 11.1 nabízí prostorovou extenzi Db2 Spatial extender. Dalším řešením je použití middlewaru ArcSDE od Esri. Geodatabáze s rastry uloženými v nativním binárním formátu je součástí databáze Db2. Rastrový dataset uložený v nativním formátu DB2 je složen z celkem 5 tabulek (business tabulka, atributová tabulka, geometrická tabulka, tabulka spektrálních pásem, ostatní metadata rastru). Business tabulka je navázána na atributovou tabulku samotných rastrů přes sloupec raster který obsahuje referenci na všechny jejich atributy.



Obrázek 3.6 Struktura uložení rastru v databázi DB2 (IBM knowlegde center, 2018)

3.3 Využití satelitních a senzorových dat

Pro praktické otestování vytvořené webové aplikace byla využita v první řadě multispektrální obrazová data. V budoucím nasazení je počítáno také s nasazením hyperspektrálních snímků. Zcela samostatně vystupují data získaná senzorovým měřením. Teoretická část se v tomto případě zaměřuje na technické parametry dat, způsob a dostupnost uložení metadat pro načtení, a další zpracování. Dalším bodem je využití spektrálních pásem těchto dat k výpočtu vegetačních indexů. Větší důraz byl kladen na volně dostupná data. U multispektrálních jsou volně dostupná data reprezentována družicovými systémy Sentinel-2 a Landsat 7 nebo 8. Komeční hyperspektrální data zastupují například družicový systém Hyperion nebo letecký systém CASI.

3.3.1 Multispektrální data

Na základě spektrálního rozlišení, které udává počet a rozsah spektrálních pásem ve kterých byla pořízena, jsou rozlišována:

- panchromatická data,
- multispektrální data,
- hyperspektrální data (esa.int, 2018).

Při pořizování multispektrálních dat jsou intervaly vlnových délek ukládány do oddělených pásem. Samostatně jsou pásma zobrazená jako černobílý obraz. Barevný obraz pomocí tzv. RGB syntézy je získán kombinací těchto jednotlivých pásem (Gisat, 2018). Multispektrální satelitní data lze získat nejen od komerčních poskytovatelů, ale existují i volně dostupná data.

Sentinel-2

Zástupcem otevřených družicových dat pro dálkový průzkum je Sentinel-2 pod správou ESA. Data ze Sentinelu-2 jsou zdarma poskytována na Scientific Data Hub (<http://scihub.copernicus.eu>). Satelit Sentinelu-2 snímá povrch ve 13 pásmech.

Pro výpočet vegetačních indexů pro konkrétní datové sady jsou využity obecné rovnice pro výpočet uvedené v kapitole 3.1. Index NDVI využívá u dat Sentinel-2 dvě spektrální pásma v prostorovém rozlišení 10 m: Band 4 – RED (665 nm) a VEGETATION RED EDGE (0.865 μm) v rozlišení 20. Výsledné hodnoty se pohybují v intervalu -1 až 1 (Index DataBase, 2018 a). Po dosazení hodnot do rovnice NDVI pro Sentinel-2 vypadá výpočet následovně:

$$\text{NDVI} = (\text{B8A} - \text{B4}) / (\text{B8A} + \text{B4}) \quad (4)$$

Vegetační index NDMI je využíván nejen pro Sentinel-2 (dále také pro Landsat 8, QuickBird nebo ASTER). U systému Sentinel-2 jsou do rovnice dosazena tato pásma: Band 11 – SWIR (1.610 μm) v rozlišení 20 m, Band 8A – VEGETATION RED EDGE (0.865 μm) v rozlišení 20 m.

$$\text{NDMI} = (\text{B8A} - \text{B11}) / (\text{B8A} + \text{B11}) \quad (5)$$

SAVI využívá u Sentinelu-2 stejně jako NDVI dvě spektrální pásma v prostorovém rozlišení 10 m: Band 4 – RED C a Band 8A VEGETATION RED EDGE (Index DataBase, 2018c). Dosazením zbývajících proměnných dostáváme rovnici ve tvaru:

$$\text{SAVI} = (\text{B8A} - \text{B4}) / (\text{B8A} + \text{B4} + \text{L}) \times (1 + \text{L}) \quad (6)$$

Landsat 7 & 8

Snímky Landsat 8 jsou pořizovány pomocí senzorů OLI a TIRS. Senzor OLI pořizuje snímky pro viditelnou, blízkou infračervenou a střední infračervenou část elektromagnetického spektra. Prostorové rozlišení těchto snímků je 30 metrů. Zároveň je senzor schopný snímat i panchromatické snímky s velikostí pixelu 15 metrů. Každé ze spektrálních pásem L8 jsou pořizována jako 16 bitové rastry. V průměru je nasnímáno 500 snímků denně. Všechny tyto snímky jsou ze surové podoby postupně zpracovány do úrovně zpracování s označením L1.

Úrovně L1GT a L1TP obsahují radiometrické a geometrické korekce a jsou dostupná v datovém formátu GeoTIFF. Tento formát rozšiřuje obrazová data TIFF (Tagged Image File Format) o kartografické a geodetické údaje vázané pro konkrétní rastr. Struktura TIFF souboru umožňuje část metadat a obrazová data ukládat do jednoho souboru. Snímky v této úrovni zpracování jsou volně stažitelné pro běžné uživatele. Celkem je v této úrovni zpracování zahrnuto 13 souborů (spektrální pásma 1–9 nasnímaných senzorem OLI, dvě spektrální pásma nasnímané senzorem TIRS, metadatový soubor ve formátu *.txt, QA band soubor zdroj: USGS, 2018). Oproti Landsatu 7 došlo k mírným změnám v označení spektrálních pásem (obrázek 3.7). Výpočty vegetačních indexů z pásem Landsat 8 definují následující rovnice 7 až 9 pro Landsat 7 rovnice 10 až 12.

$$\text{L8 NDVI} = (\text{B5} - \text{B4}) / (\text{B5} + \text{B4}) \quad (7)$$

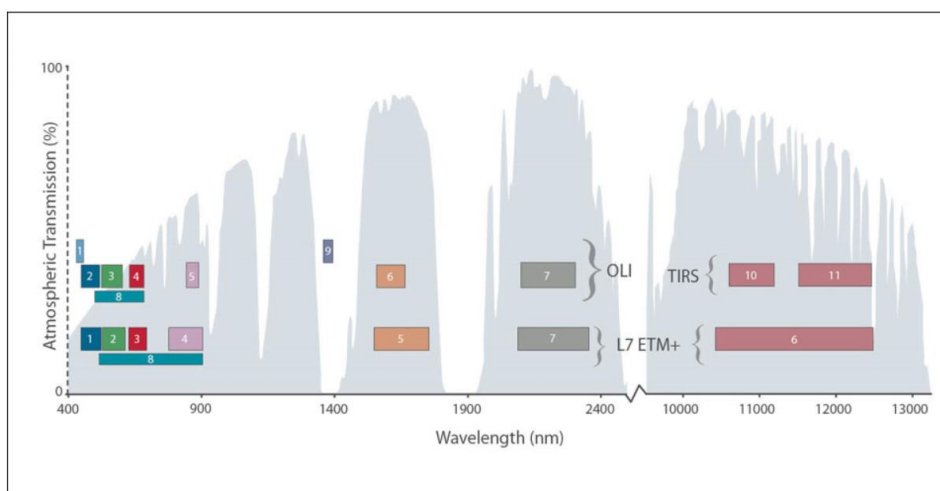
$$\text{L8 NDMI} = (\text{B5} - \text{B6}) / (\text{B5} + \text{B6}) \quad (8)$$

$$\text{L8 SAVI} = (\text{B5} - \text{B4}) / (\text{B5} + \text{B4} + \text{L}) \times (1 + \text{L}) \quad (9)$$

$$\text{L7 NDVI} = (\text{B4} - \text{B3}) / (\text{B4} + \text{B3}) \quad (10)$$

$$\text{L7 NDMI} = (\text{B4} - \text{B5}) / (\text{B4} + \text{B5}) \quad (11)$$

$$\text{L7 SAVI} = (\text{B4} - \text{B3}) / (\text{B4} + \text{B3}) \times (1 + \text{L}) \quad (12)$$



Obrázek 3.7 Porovnání pásem Landsat 8 a Landsat 7

3.3.2 Senzorová data

Data pořizovaná pomocí pozemních čidel (senzorů) jsou sbírána online v určitém časovém intervalu. Podle Pechance (2014) jsou senzory děleny na dva základní typy, analogové a digitální senzory prostředí. Analogový senzor poskytuje měřený údaj nejčastěji ve formě napětí (proti společné zemi nebo diferenciallyně), méně často pak ve formě proudu, odporu (vodivosti), elektrické kapacity apod. V některých případech je výstup senzoru tzv. vysoko-ohmový, vyžadující speciální předzesilovač. Některá čidla vyžadují galvanické oddělení měřeného systému. Některá čidla, např. odporové čidlo teploty PT100, lze zapojit a měřit několika různými způsoby a tomu odpovídá i výsledná přesnost. Oproti tomu digitální senzor poskytuje již předzpracovaný údaj po některé ze sériových (drátových) linek, například UART, RS-232, RS-485, SDI-12, SPI, I2C, 1Wire atd., případně pomocí bezdrátového přenosu (v pásmu ISM 433/868/2400MHz).

Rozdíl mezi analogovými a digitálními senzory je v odlišných nárocích na měření. Klíčový je požadavek na aplikaci napájecího napětí před samotným měřením tzv. excitací senzoru. Liší se i časová odezva a perioda měření. Rozdíly v technických parametrech různých senzorů jsou uvedeny v následující tabulce (Julina, 2016).

Senzor, veličina	Excitace [sec]	Napájecí napětí [V]	Odezva senzoru [sec]	Perioda měření [sec]	Poznámka, typ senzoru
analogový, kontinuální	-	různé	<0,001	<0,001	trvalé měření
analogový, vlhkost	2	5,5 - 18	<0,1	2	Virrib
analogový, vlhkost	0,01	2,5 - 3,6	<0,001	>0,01	EC-5
analogový, kombinovaný	-	4 - 20	<0,001	>0,1	HumiAir9
analogový, tenzometr	>0,01	10,6	<0,001	>0,1	T4e
analogový, teplota	0,1 - 1	0,1 - 0,5mA	<0,001	>0,1	PT100
digitální, I2C, teplota	-	1,4 - 3,6	<0,001	0,25	TMP102
digitální, bezdrátový, teplota	-	různé	5 - 20	5 - 20	bezdrátový teploměr
digitální, SDI-12, SRS NDVI	-	3,6 - 15	>0,03	1 - 2	SRS-Ni, SRS-Nr
digitální, UART, kamera	>1,8	4,5 - 9	>0,5	>0,5	uCAM-II

Obrázek 3.8 Technické parametry senzorů

3.4 Webové technologie používané pro DPZ

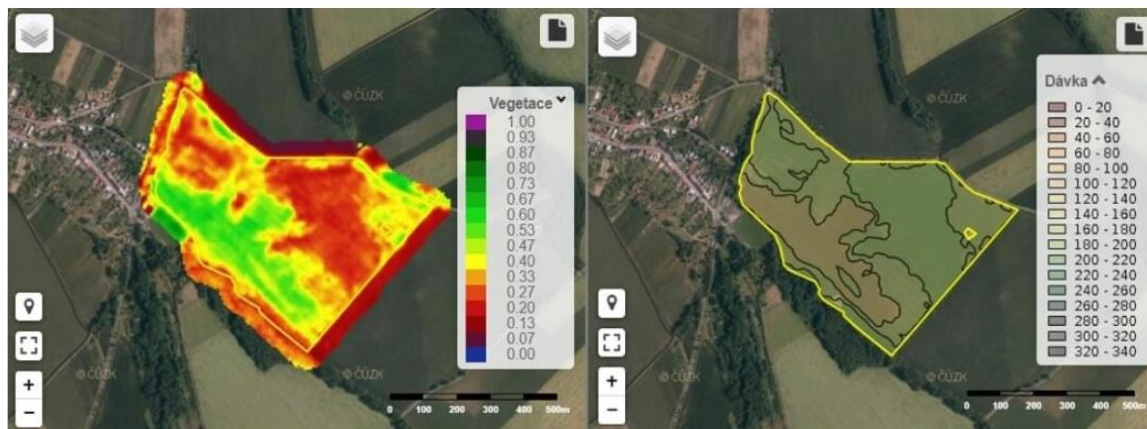
Data z Dálkového průzkumu Země začala být v poslední dekádě stále ve větší míře používána pro podporu precizního zemědělství. Pro usnadnění práce odborníkům ze zemědělské praxe a snížení požadavků na software se vedle desktopových programů stále více prosazují webové aplikace s důrazem na tenkého klienta. Následující kapitola je věnována vybraným webovým řešením pro práci s daty pořízenými prostřednictvím dálkového průzkumu Země.

Crop Explorer

Aplikace Crop Explorer začala být využívána ministerstvem zemědělství Spojených Států pro odhad zemědělské produkce již na počátku nového tisíciletí. Aplikace nabízí prakticky v reálném čase informace nutné pro odhady úrody založené na satelitních snímcích. Prostorová geodatabáze ze které nástroj čerpá je založená na gridu. Data jsou ukládána v denních, 10 denních a měsíčních intervalech. Pro měření kvality vegetace se počítají vegetační indexy. Sucho nebo extrémní vlhkost terénu může být identifikováno množstvím zeleně podle normalizovaných vegetačních indexů a měření ze satelitního snímku pomocí radiometru AVHRR. Americké ministerstvo zemědělství je od roku 2001 vybaveno produkty Esri a samotná mapová služba Crop Explorer je postavena na prvcích ArcSDE.

Clever Farm

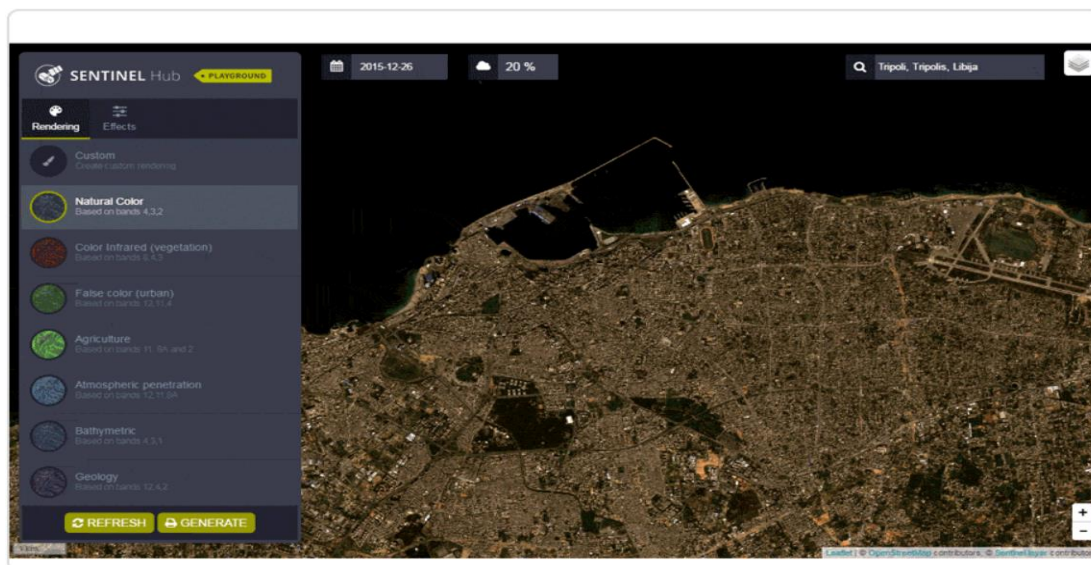
V roce 2013 vznikl v České republice projekt Clever Maps, ve kterém se postupně vytrídily 4 samostatné produktové týmy. Produktový tým Clever Farm v roce 2016 začal vystupovat jako samostatný partner původního Clever Maps se zaměřením na vývoj a správu aplikace pro agronomické aktivity (Clever Maps, 2018). Základní aplikace Clever Farm je používána pro evidenci půdy a hlídání nitrátových směrnic a je přístupná zcela zdarma. Pro vyšší funkcionalitu je nezbytné dokoupení zásuvných modulů aplikace. Zabudování sensorových dat do aplikace se věnuje modul Sensory. Lze tak v reálném čase monitorovat základní charakteristiky půdy (teplota, vlhkost). Dalším pluginem v Clever Farm je modul Sentinel. Pomocí funkcionality Sentinelu lze monitorovat zdravotní stav vegetace, vývoj biomasy, vegetační indexy nebo úhrn srážek. V modulu lze pracovat s daty Sentinel-2, Landsat 8 a SPOT-6. Na základě satelitních snímků lze v aplikaci předpřipravit mapu hnojení nebo osevu a optimalizovat tak zemědělskou produkci. Kromě aktuálních snímků má uživatel přístup i ke snímkům staršího data.



Obrázek 3.9 Aplikace Clever Farm

Sentinel Playground

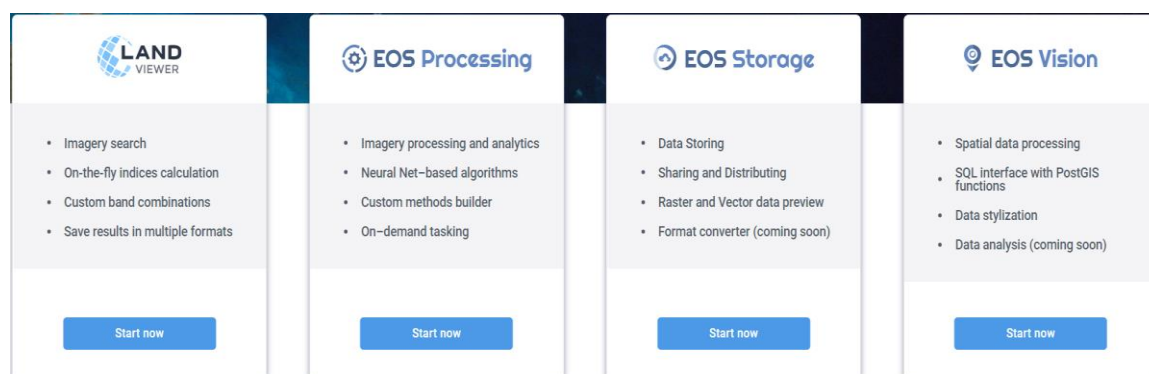
Jedná se o grafický interface, který využívá data ze Sentinel Hubu. Pomocí rozhraní lze zobrazovat satelitní data včetně archivovaných dat a provádět analýzy obrazu. V Sentinelu Playground lze pracovat s datovými zdroji z 12 senzorů (Sentinel, Landsat, MODIS a další). V rozhraní je umožněno zobrazovat snímky ze všech nasnímaných částí světa v libovolném čase, dále provádět operace s barevnými pásmy. Pomocí API webové aplikace lze vytvářet vlastní skripty na operaci s barevnými pásmy, generování mapových služeb, úpravu atmosférických korekcí a další. Podrobná technická dokumentace je dispozici na: <https://www.sentinel-hub.com/develop/documentation>.



Obrázek 3.10 Rozhraní Sentinel Playground

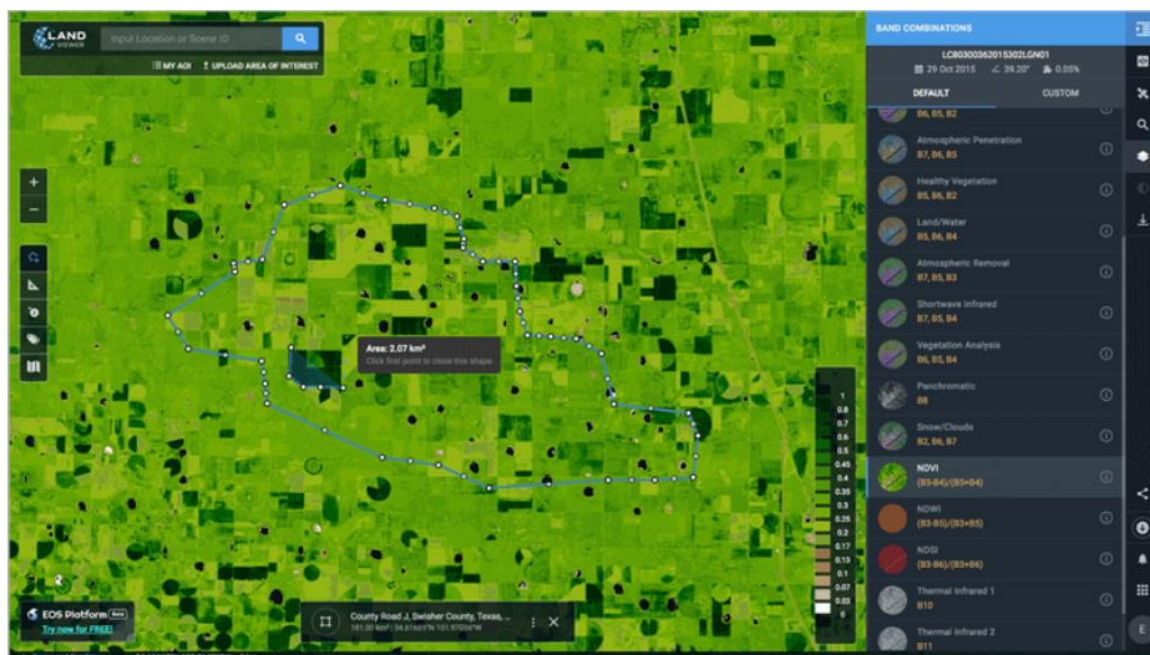
Platforma EOS

Analýzu obrazových materiálů, na kterou je běžně potřeba desktopových softwarů jako Erdas nebo Envi, je nyní možné dělat v cloudovém prostředí díky EOS Plattform. Zahrnutý jsou čtyři moduly (Landviewer, Processing, Vision, Storage). Toolboxy dohromady poskytují výkonný nástroj pro vyhledávání, analýzu, ukládání a vizualizaci objemných rastrových geodat (GIS Lounge, 2018). Pro práci v aplikaci je potřeba mít vytvořený účet na EOS a být přihlášen.



Obrázek 3.11 Toolboxy platformy EOS

V aplikaci lze pracovat s rastrovými daty v nástrojích Processing, Vision a Storage s různými datovými sadami. Kromě nativního formátu platformy lze nahrát data ve formátech Geotiff, JPEG, JPEG 2000 a pomocí API využít i vlastní algoritmy pro zpracování. Z velkého množství algoritmů zpracování dat (převzorkování, změna projekce, pansharpening) je podporován i výpočet vegetačních indexů *on the fly*. Vedle přednastavených vegetačních indexů (NDVI, ReCI, ARVI, SAVI, AVI) je možná i vlastní kombinace barevných pásem rastrů (GIS Lounge, 2018).



Obrázek 3.12 Výpočet vegetačního indexu NDVI v platformě EOS

4 VÝVOJ APLIKACE

Obsahem praktické části je podrobný rozbor vývoje aplikace od vstupní analýzy požadavků na řešení až po výslednou podobu prototypu aplikace. Vývoj jednotlivých komponent aplikace probíhal v mnoha případech souběžně, a zdrojový kód byl pravidelně verzován nástrojem *Git*. Repoziťář je volně dostupný pod odkazem <https://github.com/stepanroubalik/sensor-web-app.git>.

4.1 Uložení dat

V první fázi analytické části byl navrhnout způsob importu dat do databáze *PostgreSQL*, využívaný webovou aplikací jako hlavní úložiště a nástroj pro zpracování dat. Oproti původnímu předpokladu bylo pro import rastrových dat zvoleno řešení třetí strany. Pro potřeby prozatímního vývoje byl zvolen nástroj „*raster2pgsql.exe*“, který je součástí běžné instalace *PostgreSQL*. Pro systém Windows i další operační systémy je spustitelný z příkazové řádky následujícím způsobem:

```
C:\Program Files\PostgreSQL\11\bin>raster2pgsql.exe.
```

Pro samotné spuštění importu nestačí zadat pouze cestu k souboru *raster2pgsql.exe*, ale musí být vydefinovány i základní parametry pro uložení rastru. Následující tabulka obsahuje parametry, které jsou pro úspěšné nahrání rastru do PostGIS databáze a možnosti dalšího zpracování doporučené.

Tabulka 4.1 Parametry importu rastrových dat přes *raster2pgsql.exe*

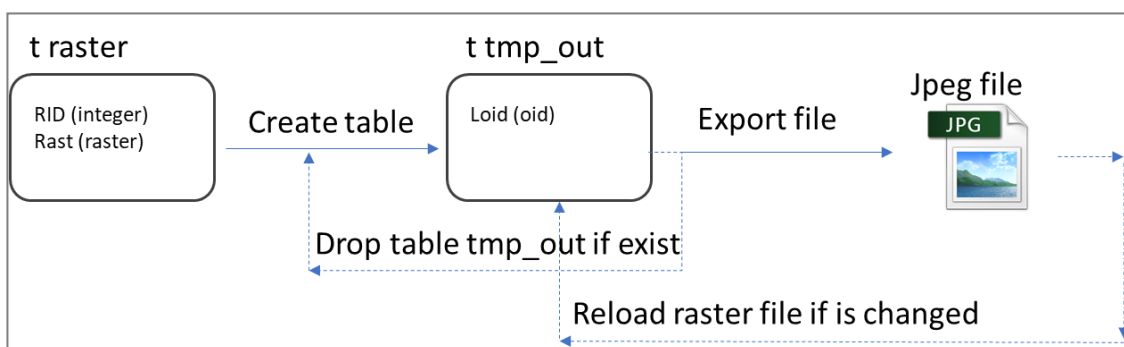
Parametr	Funkce
. \bin>raster2pgsql.exe	Volání programu v příslušném adresáři
-I	Vytvoření Gist indexu nad sloupcem rastru.
-C	Vytvoření omezení pro rastr (extent, srid, velikost pixelu) a uložení do hlavičky.
-s	Definice souřadnicového systému, který je primárně zobrazován. Ve skutečnosti data lze dotazovat a transformovat pro všechny SRID, které jsou součástí tabulky <i>spatial_ref_sys</i> .
-c	Vytvoření nové tabulky v rámci schématu. Používá se pouze poprvé, potom už se pro záznamy používá příkaz <i>append</i> .
-a	Nahraje vybraný soubor do existujícího databázového schématu a tabulky jako další záznam.
-n „Název sloupce“	Vytvoření sloupce s názvem zdrojového souboru.
-f	Specifikace cesty ke zdrojovému rastru v souborovém systému.

Výsledný zápis importu v příkazové řádce potom může vypadat následovně: *C:\Program Files\PostgreSQL\11\bin>raster2pgsql.exe -I -C -s 32633 -M -a "C:\git\rastrbandx.tif" public.L7B04 | "C:\Program Files\PostgreSQL\11\bin\psql.exe" -d "diplomka" -U "postgres" -p 5432.*

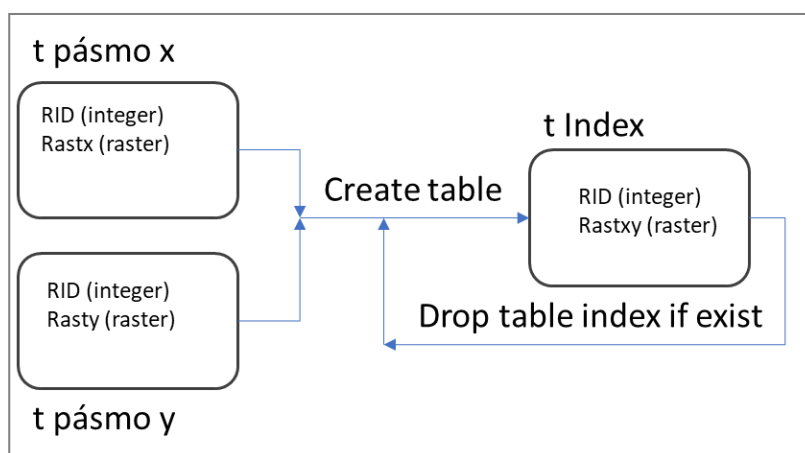
Nastavení databáze

Zdrojové databáze s názvy „db_roubalik“ a „sensor“ jsou vytvořeny na serveru Katedry geoinformatiky pod url adresou 158.194.94.120 a komunikují na portu 5432. Obsahem první databáze je schéma „public“ obsahující celkem 29 tabulek s rastrovými daty. Přehled tabulek je součástí přílohy číslo 1. Běžný uživatelský účet má pouze omezená práva na manipulaci s databází (insert a select záznamů). Oproti tomu tzv. superuser uživatel s administrátorským přístupem má umožněné mazání tabulek, schématů, rolí, přidělování práv, možnost zálohování a replikace databáze.

Následuje podrobný popis schématu tabulek pro uložení rastrů a senzorových dat a práci s daty na úrovni databáze. Následující čtyři schémata definují procesy nad databází. Všechny jsou v dalších kapitolách popsány i z pohledu tzv. use case, tedy aktivit, které vykonává uživatel ovládáním aplikace. Ovládání aplikace uživatelem vede ke spuštění těchto procesů. Prvním procesem je transformace rastru z databázové tabulky do souboru (obrázek 4.1). Každou tabulku související s rastrem je možné přes dočasnou tabulku exportovat do souboru, který je na rozdíl od tabulky zobrazitelný přímo v aplikaci formou obrazového záznamu. Proces, při kterém jsou generovány nové tabulky, je použit při výpočtu vegetačního indexu (obrázek 4.2). Příkazy *SQL* vedoucí k dosažení stavu znázorněného ve schématu jsou popsány níže v rámci kapitoly 4.1. Existence tabulky rastru je dočasná a při každém spuštění nového výpočtu s identickými vstupními parametry je přepsána tabulka vegetačního indexu.



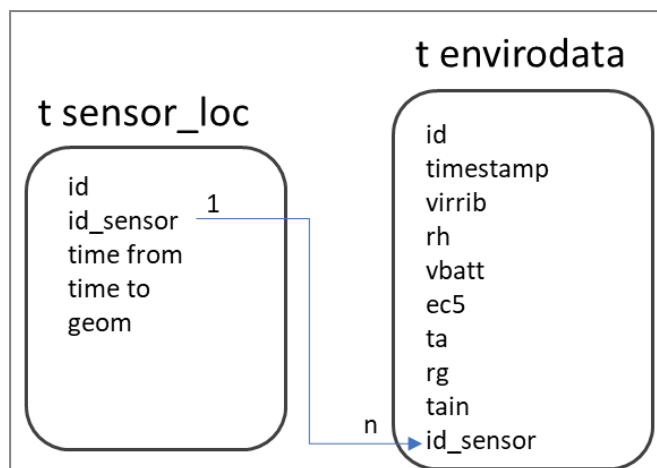
Obrázek 4.1 Proces exportu rastru z PostGIS databáze do formátu jpeg



Obrázek 4.2 Proces tvorby tabulky rastru vegetačního indexu v PostGIS databázi

Do procesních výpočtů vstupují kromě rastrů i sensorová data uložená v tabulkách *tEnvirodata* a *tSensor_location*. První tabulka obsahuje čistě atributová data. Kromě primárního klíče *id* tabulka obsahuje sloupce s názvem *timestamp* a *stanice*. Atribut *timestamp* poskytuje údaje o frekvenci měření fyzikálních veličin s periodou jedna hodina v tomto případě. Atribut *stanice* realizuje pomocí vazby 1:N propojení atributových dat s daty geolokačními napojením tabulky *tEnvirodata* na tabulku *tSensor_location*. Ostatní sloupce v *tEnvirodata* již poskytují informace o fyzikálních veličinách související s vlhkostními vegetačními indexy. Význam hodnot v jednotlivých sloupcích dekóduje následující seznam:

- virrib (objemová půdní vlhkost 1)
- rh (relativní vzdušná vlhkost)
- vbatt (napětí uvnitř baterie senzoru)
- ec5 (objemová půdní vlhkost 2)
- ta (teplota vzduchu)
- rg (srážky)
- tain (teplota uvnitř senzoru)

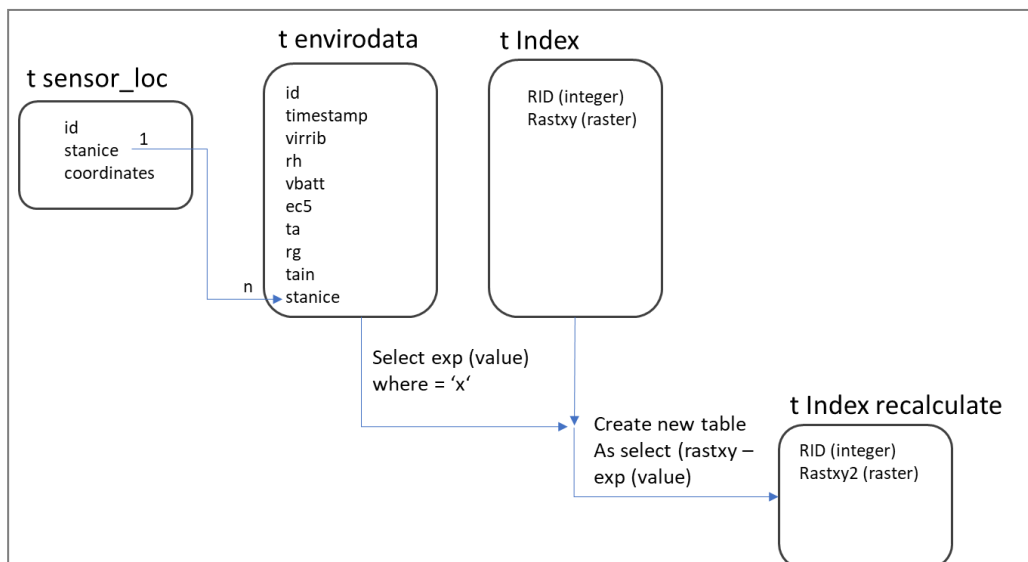


Obrázek 4.3 Možné provázání geometrické a atributové části sensorových dat

Nejnáročnější na technickou realizaci bylo navázání sensorových dat na hodnoty rastrů. Na obrázku 4.4 je zobrazena posloupnost mezikroků použitých pro výpočet výsledného modifikovaného rastru vegetačního indexu (*tIndexRecalculate*). Hodnoty pixelů jsou oproti původnímu vegetačnímu indexu (*tIndex*), který vznikl pomocí operací mapové algebry ze dvou nebo více pásem rastru (obrázek 4.2), změněny o hodnotu vybrané fyzikální veličiny pořízené senzorem. Tato hodnota je získána z tabulky *tEnvirodata*. Pro plošný odečet hodnoty získané z jednoho konkrétního bodu byly otestovány dva způsoby výpočtu:

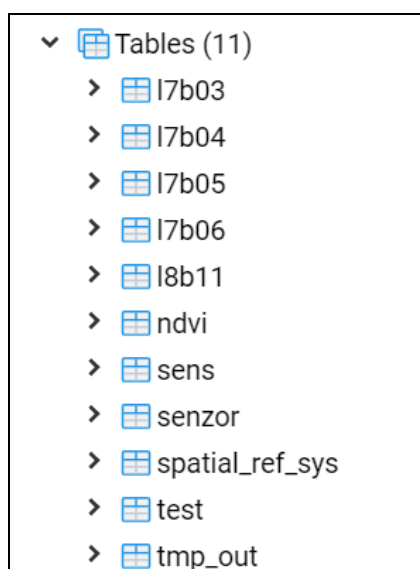
- rozdíl hodnoty pixelu a hodnoty sensorové v konkrétním čase,
- rozdíl hodnoty pixelu a aritmetického průměru,

Vlivem absence plošného rozmístění senzorů v zájmovém území nakonec nedošlo k interpolaci sensorových dat a následnému odečtu této interpolované vrstvy od rastru vegetačního indexu.



Obrázek 4.4 Proces odečtu hodnoty veličiny pořízené senzorem od hodnot pixelů vegetačního indexu

Zdrojová databáze obsahuje celkem 29 tabulek, do kterých jsou nahrávány jednotlivé typy rastrů, data ze senzorů, počítány vegetační indexy a další operace procesingu dat. Názvy tabulek rastrů byly voleny tak, aby asociativnost názvů co nejvíce korespondovala s jejich obsahem. Například kódový název tabulky *l7b03* odpovídá rastrovému snímku *Landsat 7* a jeho barevnému pásmu v červené viditelné části elektromagnetického spektra. Aplikační část se snažila minimalizovat požadavek na používání těchto zkrácených názvů přímo v prostředí aplikace. Prakticky ve všech případech byl tento problém vyřešen pomocí aliasů, které jsou v uživatelském rozhraní aplikace použity. Pro výše zmíněnou tabulku tak byl vytvořen například alias *Landsat 7 – RED pásmo*. Analogicky bylo postupováno pro rastry Landsatu 8 a Sentinelu-2 i další data. Náhled na vytvořené tabulky je obsahem obrázku 4.5.



Obrázek 4.5 Ukázka databázových tabulek

Použití prostorových funkcí v PostGIS

V práci byla použita celá řada funkcí PostGIS, které byly na úrovni databázového systému součástí datových výběrů. Pro možnost zobrazení metadat k rastrovým datům uložených v hlavičce rastru byla použita funkce *ST_MetaData* a *ST_BandMetaData*. Jako argumenty funkcí jsou souřadnice levého horního rohu (*.upperleftx*), souřadnicový systém (*ST_SRID*), velikost pixelu (*round(ST_PixelWidth(rast)::numeric,4)*), bitová hloubka rastru (*ST_BandPixelType*) a počet pásem rastru (*.numbands*). Celý zápis sql s použitím všech výše zmíněných parametrů je ve tvaru *SELECT rid As raster_id, ST_SRID(rast) AS srid, round(ST_PixelWidth(rast)::numeric,4) AS velikost_Pixelu, ST_BandPixelType(rast) AS bitová_hloubka FROM schema.raster*.

Důležitou složku pro načítání rastrových dat do aplikace je zjištění souřadnic rohových bodů rastrového snímku. Pro tento úkol databázová část využívá v dotazování funkce *ST_AsText*, *ST_Transform* a *ST_Envelope*. Samotný bounding box je vypočítán pomocí třetí funkce *ST_Envelope*, ostatní dvě slouží pro převod dat do stringového formátu a transformaci souřadnic do požadovaného souřadnicového systému. Celý zápis dotazu je potom v syntaxi „*SELECT ST_AsText(ST_Transform(ST_Envelope(rast), 4326)) As boundingbox FROM raster;*“.

Díky rozdílné bitové hloubce družicových snímků Landsat 7 (8 bitů) a Sentinel-2 (16 bitů) musela být pro převod využita funkce *ST_Reclass*. Převzorkování rastrů musí být provedeno pro snímky Sentinelu-2, snímky Landsat jsou v odpovídajícím, 8 bitovém formátu. Zápis pro převzorkování rastru je: *ST_Reclass(rast,'0-65535:0-255', '8BUI')*. Prvním argumentem funkce je sloupec s rastrovými daty, druhý definuje bitový rozsah a třetí bitovou hloubku. Funkce je součástí UPDATE příkazu. V tomto případě není vytvářen nový sloupec s převzorkovanými daty, ale je pouze přepsán původní sloupec „*rast*“. Vzorový příkaz pro převzorkování vypadá následovně: „*UPDATE tabulka SET sloupec = ST_Reclass (rast,'0-65535:0-255', '8BUI') WHERE name = "zaznam x";*“. Tato funkce se při testování potýkala s problémy, a proto převzorkování bylo pro zachování přesnosti výpočtu provedeno před importem do DB v programu *SNAP Desktop*.

Funkce mapové algebry

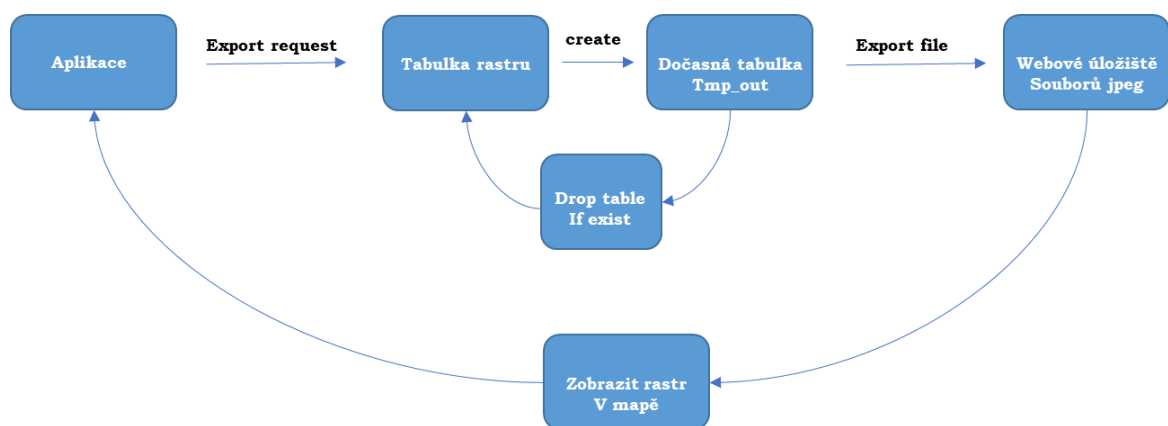
Výpočet vegetačního vegetačního zajišťuje procesní funkce *ST_MapAlgebra*. Vstupem do výrazu mapové algebry jsou dva nebo více pásem rastru. Výsledkem je pouze jeden výsledný rastr, v tomto případě na základě definovaného vzorce výpočtu vegetačního indexu. Následující ukázka SQL skriptu obecně rozebírá celý výpočet indexu. Na prvním řádku je vytvářena příkazem *create table* nová tabulka, do které je ukládána výsledná hodnota sloupce *rast* obsahující hodnoty pixelů výsledného rastru. Za příkazem *ST_MapAlgebra* následuje v závorce výpočetní výraz specifický pro každý index. Podle výrazu dojde k přepočítání hodnot vstupních pásem rastru. Třetí a čtvrtý řádek potom do vydefinovaného vzorce na řádku předchozím doplní číselné hodnoty ze vstupních tabulek. Průnik jednotlivých pixelů je zajištěn pomocí vazby *inner join* přes sloupce *rid*. I přes tuto vazbu je dobré vždy ověřit, zda mají vstupní rastry stejné prostorové rozlišení, stejnou bitovou hloubku a stejný bounding box. Všechny tyto parametry rastru lze získat přes tlačítko v aplikaci *zobrazit metadata*.

```
CREATE TABLE public.result AS
SELECT ST_MapAlgebra (arast, 1, brast, 1, '([rast1] - [rast2]) / ([rast1] + [rast2])::float', '32BF') AS rast
FROM (SELECT a.rast as arast, b.rast as brast FROM public.bandA
a INNER JOIN public.bandB b ON a.rid = b.rid) as joined
```

Obrázek 4.6 příklad SQL příkazu pro výpočet vegetačního indexu

Export dat

Pro možnost zobrazení dat v mapové aplikaci jsou rastry z databáze nejdříve načteny do paměti a uloženy jako soubory ve formátu jpeg nebo png do úložiště na serveru. Odtud jsou potom postupně načteny funkcemi PHP a javascriptu do webové aplikace. Pro uložení rastru z libovolné tabulky v databázi do paměti je použitý sql řetězec pro vytvoření dočasné tabulky: `'CREATE TABLE docasnaTabulka AS SELECT lo_from_bytea(0, ST_AsGDALRaster (ST_Union(rast), 'JPEG', ARRAY['QUALITY=50'])) AS loid FROM zdrojovyRastr;'`. Po vytvoření dočasné tabulky může být proveden vlastní export do souboru následujícím skriptem: `'SELECT lo_export(loid, 'c:\temp\exportovanyRastr.jpg') FROM docasnaTabulka;'`. V momentě kdy je export dokončený, není již dočasná tabulka potřebná a příkazem `'DROP TABLE IF EXISTS docasnaTabulka;'` je tahle tabulka odstraněna. Tato iterace může být zopakována v libovolném množství rastrů. Schéma tohoto procesu je v detailu na obrázku 4.7 a doplněný zdrojový kód na obrázku 4.8. Pro zahájení exportu musí být odeslán požadavek ze strany klienta použitím tlačítka aplikace „GENEROVAT RASTR“. Tímto je vytvořena dočasná tabulka „tmp_out“, která je shodná s obsahem dotazované rastrové tabulky. Pokud dočasná tabulka již existuje, je nejdříve pomocí sql smazána a opětovně vytvořena. Tento proces probíhá při každé nové iteraci exportu. Z dočasné tabulky je výše popsaným způsobem vytvořený a uložený soubor jpeg s uživatelsky definovaným názvem. Takto uložený soubor lze potom zobrazit přímo v aplikaci.



```
<?php
$tabulka = $_POST['export'];
$db = new PDO('pgsql:host=localhost;port=5432;dbname=diplomka;', 'postgres', 'sr');
$deleteTable = $db->prepare("DROP TABLE IF EXISTS tmp_out;");
$deleteTable->execute();
$createTable = $db->prepare("CREATE TABLE tmp_out AS SELECT lo_from_bytea(0,
ST_AsGDALRaster(ST_Union(rast), 'JPEG', ARRAY['QUALITY=50'])) AS loid
FROM $tabulka;");
$createTable->execute();

$tmp = $_POST['tmp'];
$db = new PDO('pgsql:host=localhost;port=5432;dbname=diplomka;', 'postgres', 'sr');
$createFile = $db->prepare("SELECT lo_export(loid, 'C:\\xampp\htdocs\diplomka\data\l7b05.jpg')
FROM $tmp;");
$createFile->execute();
?>
```

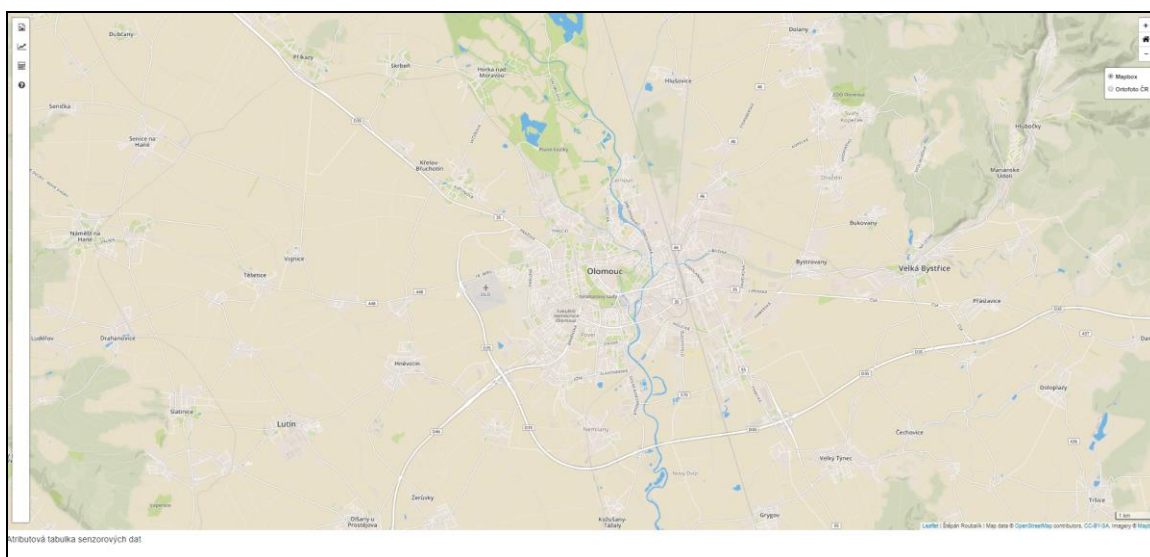
Obrázek 4.8 Zdrojový kód pro export rastru

4.2 Frontend aplikace

Pro tvorbu uživatelského rozhraní byla použita javascriptová knihovna Leaflet. Pomocí objektů a funkcí této open source knihovny jsou definovány základní prvky mapové aplikace. V rozložení jsou zakomponovány následující prvky:

- mapové pole,
- ovládací postranní panel,
- legenda,
- zoom button,
- měřítko.

Metody programovacího jazyka Javascript jsou použity i pro jednotlivé prvky zajišťující funkcionalitu, kterou aplikace nabízí. Následující text, ukázka zdrojového a náhled základního zobrazení (obrázek 4.9) popisuje základní konfiguraci Leafletu a načtení mapy v těle html webové stránky.



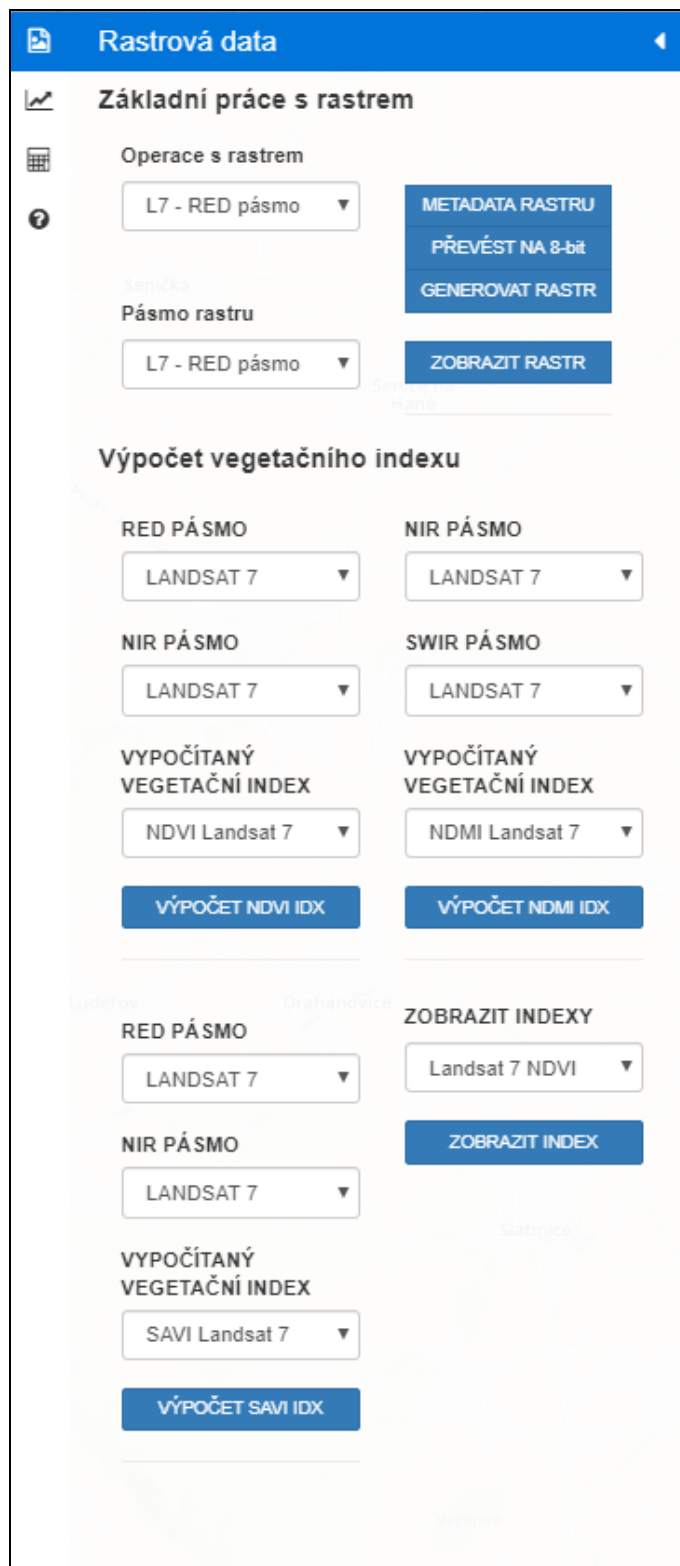
Obrázek 4.9 Základní zobrazení aplikace po spuštění

Inicializace Leafletu požaduje v první řadě provázání stránky `index.html` s dalšími konfiguračními soubory a externími skripty v sekci `<head></head>`. V tomto případě hlavička obsahuje celkem čtyři typy referencí:

- linky a skripty na knihovnu *Leaflet* a *jQuery*,
- linky na použité styly *CSS* a knihovnu *Bootstrap*,
- linky na *Zoomhome plugin*,
- linky na *Sidebar plugin*.

Zdrojový kód je součástí přílohy 6. Jako hlavní pro tvorbu kaskádových stylů a designu webové stránky byla využita knihovna *Bootstrap* ve verzi 3.3.7. Zahrnuto je zde stylování jednotlivých prvků, ale i rozmístění v rámci jednotlivých sekcí stránky. Extenze *Zoomhome* patří v *Leafletu* mezi běžně používané moduly. V mapovém okně je potom umožněné vykreslit ovládací prvky pro zoom in a zoom out a především volbu vrácení se do výchozího zoomu, ve kterém je aplikace při prvním načtení.

Samostatnou sekci tvoří postranní panel. V základním zobrazení mapového okna je minimalizovaný a na záložky je umožněno dostat se po jeho rozbalení. Panel pod sebe začleňuje tři záložky pro výpočty vegetačních indexů nad rastrovými daty, kartu pro manipulaci se sensorovými daty, kartu pro korekční výpočty nad NDMI vegetačním indexem a kartu nápovědy (viz obrázky 4.10 a, b, c).



Obrázek 4.10 První záložka rozbalovací postranní panelu pro rastrová data

Obrázek 4.10b Druhá záložka rozbalovací postranní panelu pro senzorová data

Obrázek 4.10c Třetí záložka rozbalovací postranní panelu pro korekční výpočet

4.3 Komunikace aplikace s databází

Pro fungování aplikace je nezbytné propojení uživatelského rozhraní aplikace a procesních funkcí v PostGIS databázi. Tato kontinuita je zajišťována skriptovacím jazykem PHP na straně serveru, metodou asynchronního načítání AJAX v rámci webové stránky a funkcemi samotného skriptovacího jazyka Javascript.

Konfigurační soubory aplikace

Spouštěčem aplikace na webovém serveru je soubor *index.html*. Jeho hlavička obsahuje tagy s referencemi ke všem externím knihovnám a souborům definující strukturu a vzhled aplikace. Tělo tohoto souboru je vyplněno HTML tagy pro obsahovou část aplikace a skripty programovacího jazyka Javascript pro interaktivní prvky aplikace. Souborů php pro navázání komunikace s databázovým serverem a zpracování dat na straně serveru je celkem 12: Celá souborová struktura je obsahem obrázku 4.11 a popis funkcionality je také obsahem přílohy 2.

css	07.08.2019 15:31	Složka souborů	
data	07.08.2019 15:31	Složka souborů	
js	07.08.2019 15:31	Složka souborů	
zoomhome	07.08.2019 15:31	Složka souborů	
displayIndex	07.08.2019 15:31	Soubor PHP	1 kB
displayRaster	07.08.2019 15:31	Soubor PHP	1 kB
index	07.08.2019 15:32	Chrome HTML Docu...	26 kB
rasterExport	07.08.2019 15:31	Soubor PHP	1 kB
rasterProperties	07.08.2019 15:31	Soubor PHP	2 kB
sensorCorrection	07.08.2019 15:31	Soubor PHP	1 kB
sensorData	07.08.2019 15:31	Soubor PHP	1 kB
sensorDataStatistics	07.08.2019 15:31	Soubor PHP	2 kB
sensorLocation	07.08.2019 15:31	Soubor PHP	1 kB
sensorReclass	07.08.2019 15:31	Soubor PHP	1 kB
vegetationIndicesNdm1	07.08.2019 15:31	Soubor PHP	1 kB
vegetationIndicesNdvi	07.08.2019 15:31	Soubor PHP	1 kB
vegetationIndicesSavi	07.08.2019 15:31	Soubor PHP	1 kB

Obrázek 4.11 Konfigurační soubory nezbytné pro chod aplikace

Vytvoření připojení k databázi

Připojení k databázi je součástí každého z výše zmiňovaných php souborů. Pro přehlednost je proměnná definující připojení ve všech případech pojmenována shodně *\$db*. Definice připojení je v základním tvaru „`$db = new PDO (argumenty);`“. Jako povinné jsou brány následující parametry (host, port, název databáze, uživatel, heslo). Celý zápis potom vypadá například takto: `$db = new PDO('pgsql:host="url";port=5432;dbname=dbname;', 'user', 'heslo');`

Pro připojení k databázi není vyžadována žádná aktivita ze strany klienta. Připojení proběhne automaticky při prvotním načtení. V dalším kroku již probíhá zpracování dat nad databází. Zde jsou jednotlivé sql dotazy ukládány do libovolně pojmenovaných proměnných obecným zápisem: „`$promenna = $db -> prepare („sql dotaz“);`“. Proměnná *\$db* referuje k výše nakonfigurované a připojené databázi a zbytek zápisu zpracovává samotný sql dotaz.

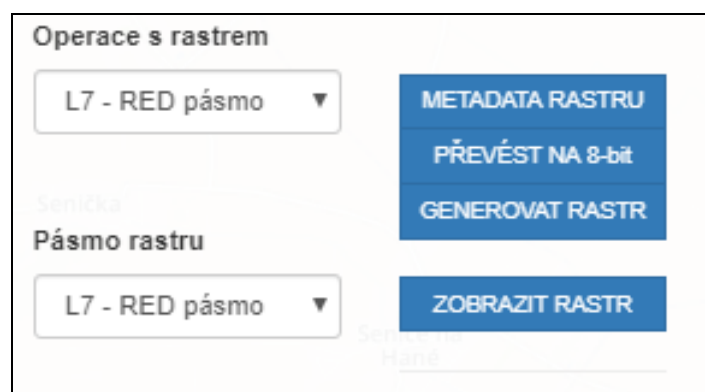
Přenos a zpracování vstupních parametrů

Tato část bude zaměřena na popis procesů, které probíhají na pozadí aplikace (navázání komunikace s databází, výběry dat, výpočty) po inicializaci na straně klientské aplikace. Jako příklad je uvedeno načtení metadat o rastroch uložených v databázi. Princip zpracování dat je popsán obecně a v různých obměnách je obsažen i při popisu konkrétní funkcionality. Podrobněji se fungování jednotlivých prvků zabývají části kapitoly 5 a také přílohy práce.

Zprv je na straně klientské aplikace pomocí html tagu zobrazitelný výběr rastru a tlačítko spuštění procesu načtení metadatových záznamů (obrázek 4.12 a 4.13). Podle toho jaký rastr z nabídky uživatel vybere, jsou ve výsledku zobrazena i příslušná metadata. Po stisknutí tlačítka je požadavek odeslán na server kde je pomocí php souborů a SQL skriptů vytvářena odpověď na definovaný požadavek.

```
<br><br><h5><strong>Pásmo rastru</strong></h5>
<select id="typ" class="form-control">
  <option value="17b03">L7 - RED pásmo</option>
  <option value="17b04">L7 - NIR pásmo</option>
  <option value="17b05">L7 - SWIR pásmo</option>
  <option value="18b04">L8 - RED pásmo</option>
  <option value="18b05">L8 - NIR pásmo</option>
  <option value="18b06">L8 - SWIR pásmo</option>
  <option value="s2b04">S2 - RED pásmo</option>
  <option value="s2b08a">S2 - NIR pásmo</option>
  <option value="s2b11">S2 - SWIR pásmo</option>
  <option value="test_s2b08a">testovací NIR pásmo</option>
  <option value="test_s2b11">testovací SWIR pásmo</option>
  <option value="18_ndmi">testovací NDMI index</option>
  <option value="18_savi">testovací savi index</option>
</select>
</div>
<div class="col-sm-6">
  <div class="btn-group-vertical">
    <br><br><button id="metadata" class="btn btn-primary btn-sm btn-block">METADATA RASTRU</button>
    <button id="transform" class="btn btn-primary btn-sm btn-block">PŘEVÉST NA 8-bit</button>
    <button id="save" class="btn btn-primary btn-sm btn-block">GENEROVAT RASTR</button><br>
    <button id="rasterBoundingBox" class="btn btn-primary btn-sm btn-block">ZOBRAZIT RASTR</button><hr>
  </div>
</div>
```

Obrázek 4.12 Ukázka zápisu ovládacích prvků v HTML



Obrázek 4.13 Ukázka ovládacích prvků aplikace

Soubor php, pomocí kterého jsou zpracovávány vstupní parametry požadavku (obrázek 4.14) vytváří připojení na databázi pomocí proměnné „\$db“. Po vytvoření připojení je vykonána procesní část nad daty pomocí sql dotazu uloženého do proměnné „\$sql“. V zápleti je proměnná zpracována příkazem „\$sql->execute();“. V poslední části php skriptu je funkcí jako například „echo“ vygenerovaný výsledek dotazu, který je i v této podobě načítán AJAXEM pomocí response funkce.

```

<?php
$meta=$ POST['meta'];
$db = new PDO('pgsql:host=localhost;port=5432;dbname=diplomka', 'postgres', 'sr');
$sql = $db->prepare("SELECT
    rid As r,
    ST_Width(rast) As w,
    ST_Height(rast) As h,
    round(ST_PixelWidth(rast)::numeric,0) AS pw,
    round(ST_PixelHeight(rast)::numeric,0) As ph,
    ST_SRID(rast) AS srid,
    ST_BandPixelType(rast,2) AS bt
FROM $meta");
$sql->execute();
echo "<table id= 'dtHorizontalVerticalExample' class= 'table table-striped table-bordered table-sm' c";
echo "<tr><th>rid</th><th>w</th><th>h</th><th>pw</th><th>ph</th><th>srid</th><th>bt</th></tr>";
while ($row = $sql->fetch(PDO::FETCH_ASSOC)) {
    echo "<tr>";
    foreach($row as $field=>$value) {
        echo "<td>{$value}</td>";
    }
    echo "</tr>";
}
echo"</table>";

```

Obrázek 4.14 Ukázka PHP souboru rasterProperties.php pro metadata

```

<script>
    $("#metadata").click(function() {
        $.ajax({
            url:'rasterProperties.php',
            type:'POST',
            data:{
                meta: $("#meta").val(),
            },
            success: function(response) {
                $("#rasterTable").html(response);
            }
        });
    });
</script>

```

Obrázek 4.15 Spuštění AJAXU a zpracování dat serveru

V dalším kroku je na straně frontendu cílem definovat co má aplikace po výběru dat na vstupu a potvrzení operace kliknutím na tlačítko „Zobrazit metadata“ dělat dál. To je zajišťováno pomocí následujícího javascript kódu (obrázek 4.15). Pokud uživatel klikne na tlačítko, jehož id odpovídá řetězci „metadata“, dojde ke spuštění *Ajaxy*, který odkáže na příslušný soubor php. Tento soubor na backendu zpracuje vstupní parametry pomocí proměnných. V momentu kdy jsou úspěšně provedeny operace v serverové části, je výsledek pomocí *Ajaxy* a funkce „function(response)“ načten pomocí úspěšného provedení funkce *function(response)* do sekce mapové aplikace s id „rasterTable“.

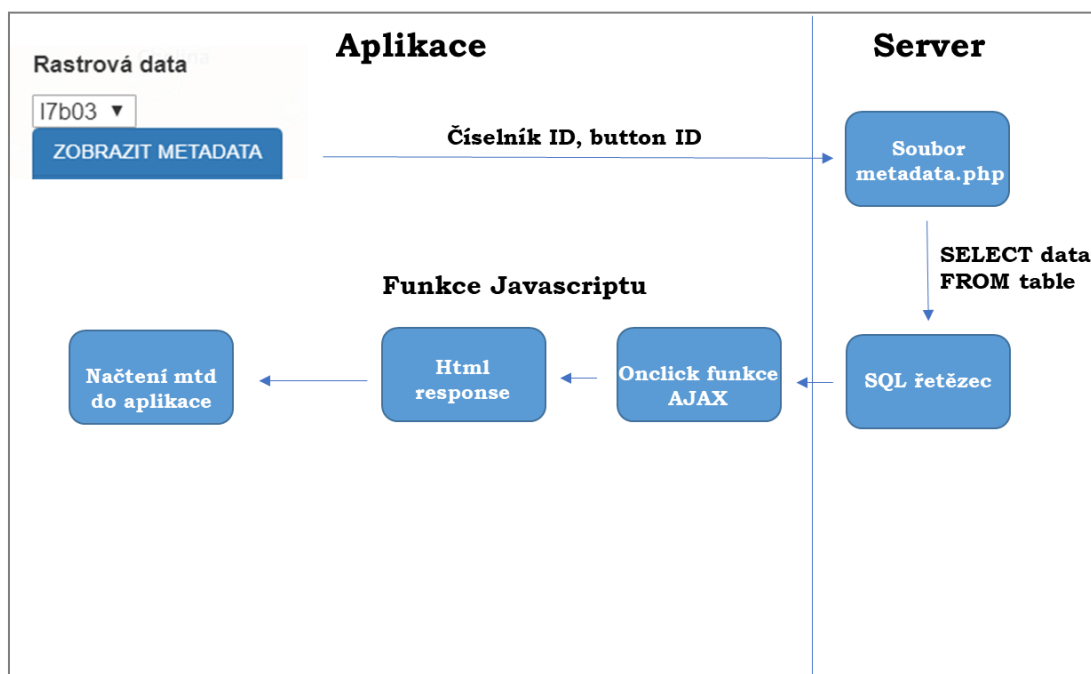
5 FUNKCIONALITA APLIKACE

Další kapitola bude věnována popsání konkrétních prvků aplikace a jejich fungování v rámci celého řešení od definice vzhledu na frontendu až po procesní zpracování na backendu.

5.1 Metadatové informace

Funkce pro načtení metadat

Načtení metadat příslušného snímku patří k základním procesům, které aplikace zvládá. Odpovědí na tento požadavek je zobrazení tabulky, která je naplněna daty podle použitého sql řetězce na straně serveru. Proměnné do tohoto řetězce za pomoci požadavku *POST* definuje sám uživatel na straně klienta. Zpracování dotazu je obsahem následujícího workflow na obrázku 5.1.



Obrázek 5.1 Proces dotazování na metadata rastrů

Funkce pro transformaci rastrů

Pro volbu změny bitové hloubky rastru funguje tlačítko „PŘEVÉST NA 8-bit“. Jeho spuštěním dojde k převodu sloupce rastr v tabulce PostGIS která se tímto aktualizuje. V prototypu aplikace byly prozatím vynechány další možnosti transformace rastrů. Sloupec *rast* definující hodnoty pixelů rastru je pomocí funkce *ST_Reclass* z intervalu pro 16 bitový rastr na rozpětí rastru 8 bitového. Kvůli problémům s exportem a načítáním jiných než 8 bitových rastrů však prozatím není možné využít potenciál této funkce naplno. Souvisí s tím i omezení s načítáním vypočítaných vegetačních indexů do aplikace. Ukázková data vegetačních indexů musela být na serverové úložiště nahrána ručně. Zobrazování nestandardních rastrových typů dat je probíraným bodem v diskuzi práce.

5.2 Zobrazování dat

Zobrazení dat se skládá ze dvou dílčích kroků, které musí uživatel na straně klienta provést. Zdrojová data uložená v tabulkách PostGIS jsou v první fázi exportována do úložiště na serveru v takovém formátu, aby mohla být ve druhém kroku načtena do mapového okna Leafletu. Reprezentaci rastru uloženou přímo v databázi nelze načíst do aplikace přímo. Protože však rastr pro potřeby výpočtu vegetačního indexu v PostGIS databázi uložený být musí, byl potřeby načítání obrazového záznamu do mapy v Leafletu vytvořen export z rastrové PostGIS tabulky do souboru jpeg.

Funkce pro export rastrů

Uživatel může libovolný rastr uložený z databáze načíst a zobrazit v mapovém okně. Ovládací prvky tvoří jeden číselník výběru tabulky pro export a zobrazení rastru a dvě potvrzovací tlačítka „*GENEROVAT RASTR*“, „*ZOBRAZIT RASTR*“. Číselník obsahuje hodnoty všech typů rastrových tabulek. Podle této hodnoty je na straně serveru za použití jazyka *SQL*, *PHP* a *Javascriptu* provedeno nejdříve uložení hodnot rastru do dočasné tabulky s názvem *tmp_out*. V následujícím kroku ze stejného číselníku vybrán název souboru rastru a ten je z této dočasné tabulky exportován na serverové úložiště rastrů ve formátu jpeg. Odtud je rastr v momentu požadavku na zobrazení v mapě spárován se souřadnicemi svého boundingboxu, na které se aplikace pomocí *sql* dotáže. Po provedení těchto dvou kroků je možné rastr zobrazit v aplikaci tlačítkem *ZOBRAZIT RASTR* nebo *ZOBRAZIT INDEX*.

Funkce pro zobrazování rastrů

Pro načítání rastru jsou na frontendu vytvořeny dva typy prvků. Číselník, ze kterého jsou vybírána rastrová data a tlačítko „*Zobrazit rastr*“. Tohle tlačítko spouští proces přebírající ze zvolené nabídky název rastru, který má být zobrazen. Funkce *Javascriptu* a *php* potom zajišťují dvě klíčové věci. Podle názvu datového zdroje je z databáze načten boundingbox rastru a jeho souřadnice rohových bodů jsou použity pro lokalizaci rastru v mapě (obrázek 5.2). To jaké pásmo nebo typ rastru se do daných souřadnic v mapě vykreslí, určuje vydefinovaný název v číselníku, který se shoduje s názvem vyexportovaného souboru z přecházejícího kroku (exportu) a názvem rastrové tabulky v databázi.

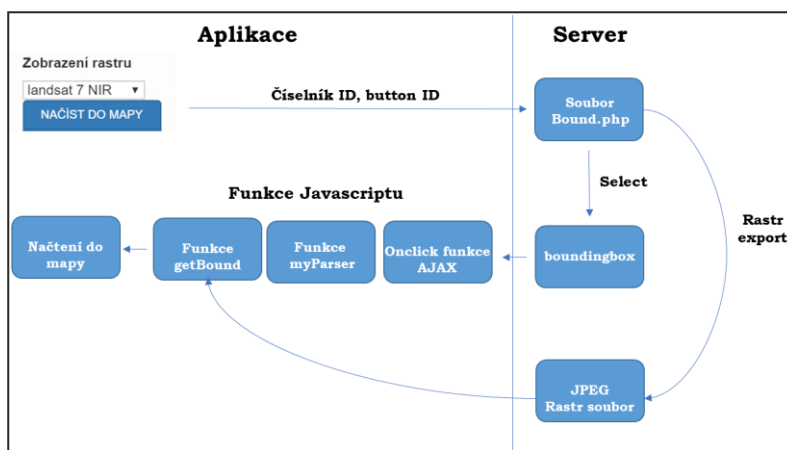
```
<?php
$typ = $_POST['typ'];
$db = new PDO('pgsql:host=localhost;port=5432;dbname=diplomka;', 'postgres', 'sr');
$sql = $db->prepare("SELECT ST_AsText(ST_Transform(ST_Envelope(rast),4326)) As boundwkt
FROM $typ;");
$sql->execute();

while ($row = $sql->fetch(PDO::FETCH_ASSOC)) {
    foreach($row as $field=>$value) {
        echo "{$value}";
    }
}
```

Obrázek 5.2 Skript PHP zajišťující processing na straně serveru

Po stisknutí tlačítka „ZOBRAZIT RASTR“ je z klienta na server odeslán požadavek. Ten v podobě sql dotazu provede na serveru několik operací. Dojde k připojení na požadovanou databázi. Z databáze dojde k výběru dat týkající se bounding boxu zvoleného rastru. Tato informace je po provedení *select* příkazu přebrána serverem a odeslána zpět. Po vrácení na stranu klienta je požadavek nadále zpracováván *Javascriptem*.

Jednotlivé části celého procesu jsou znázorněny ve schématu na obrázku 5.3. Výběrem na straně serveru byl jako odpověď vygenerován řetězec obsahující souřadnice rohových bodů vybraného rastru v následujícím formátu: *POLYGON((16.078 48.945,16.242 48.943,16.240 48.835,16.076 48.836,16.07848.945))*. V tomto tvaru přebere řetězec skript na obrázku 5.4, který definuje jaký prvek spustí vrácení této odpovědi, v jakém souboru proběhlo zpracování dotazu v serverové části, jaká data odpověď serveru je vrácena, a v neposlední řadě jak budou data nadále zpracovávána. Spuštěním tlačítka s *id=boundingBox* dojde k provedení sekvence příkazů a pomocí *AJAXU* je vrácena odpověď serveru funkcí *response* do skriptu i s *daty*, které definuje číselník pro výběr rastru. Data o boundingboxu rastru jsou v podobě textového řetězce nadále předána jako proměnná do funkce *myParser* a parsována až do výsledné podoby [*souřadnice x, souřadnice y*]. V tomto tvaru jsou souřadnice finálně použity při načtení rastru.



Obrázek 5.3 Schéma procesu zobrazení rastru

```

<script>
$("#rasterBoundingBox").click(function(){
$.ajax({
url:'displayRaster.php',
type:'POST',
data:{
typ: $("#typ").val(),
},
success: function(response){
myParser(response);
}
});
});
</script>

```

Obrázek 5.4 Skript přebírající ze serveru odpověď pomocí metody AJAX

Funkce *myParser* (obrázek 5.5) zpracuje vstupní řetězec uložený v proměnné *response* tak, aby výsledná proměnná *getBound()* odpovídala seznamu kde lze pořadí jeho prvků dotazovat pomocí indexování seznamu. Na jednotlivých pozicích tohoto seznamu jsou hodnoty souřadnic, které jsou použity ve funkci *getBound()*. Ta finálně spáruje souřadnice rastru se souborem obsahující data a vykreslí výsledek do mapy. V proměnných na druhém až pátém řádku dochází k postupnému parsování záznamu až do výsledného tvaru [hodnota x, hodnota y], který je předán funkci *getBound()*.

Funkce *getBound* přebírá rozparsovaný řetězec z předchozí funkce *myParser*. Druhý až pátý řádek kódu na obrázku 5.6 uloží hodnoty na příslušných místech řetězce do proměnných, které jsou ve finále uloženy do proměnné *var imageBounds*. Tím jsou vyřešeny souřadnice výsledného obrázku. Na řádcích 8 a 9 je pomocí metody *document.getElementById* k těmto souřadnicím přidělen i soubor uložený ve webovém úložišti ke kterému je cesta definovaná pomocí proměnné *var imageUrl*. Zobrazení je ve finále docíleno přidáním proměnné *var rastrSnimek* přidáním do mapového pole pomocí funkce *.addTo(map)*.

```
function myParser(response) {
    var res = response.split(",");
    var first = res[0].split("(");
    var firstA = first[1].split(" ");
    var third = res[2].split(" ");
    getBound(firstA, third);
}
```

Obrázek 5.5 Funkce *myParser*

Tabulka 5.1 Postup parsování odpovědi serveru

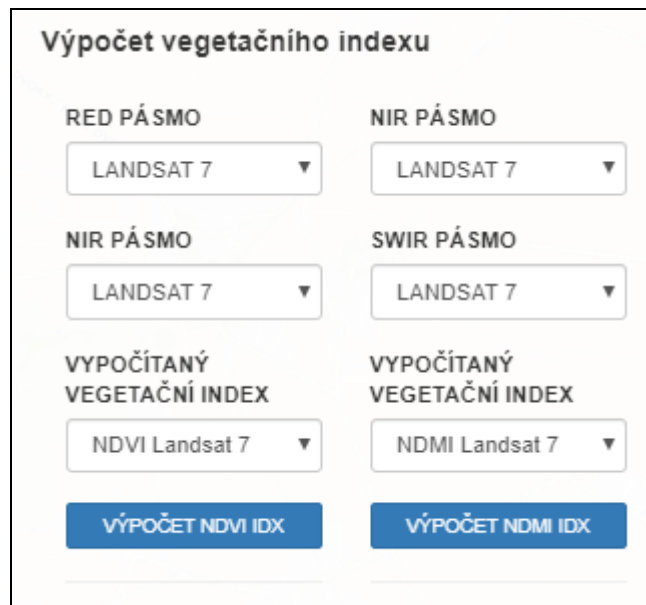
proměnná var	hodnota
response	POLYGON((string))
res	[POLYGON((xy1, xy2,...xyn))]
first	[POLYGON, xy1]
firstA	[x1, y1]
third	[x2, y2]

```
function getBound(first, third){
    var text1x = first[0],
        text1y = first[1],
        text2x = third[0],
        text2y = third[1];
    map.panTo([text1y, text1x]);
    var imageBounds = [[text1y, text1x],[text2y, text2x]];
    var typ = document.getElementById("typ");
    var vybranyTypRastru = typ.options[typ.selectedIndex].value;
    var imageUrl = './data/'+vybranyTypRastru+'.jpg';
    var rastrSnimek = L.imageOverlay(imageUrl, imageBounds).addTo(map);
}
```

Obrázek 5.6 Funkce *getBound* pro lokalizaci rastru v mapě

5.3 Výpočet indexů

Výpočet vegetačních indexů je aplikovaný na tři nejpoužívanější vegetační indexy podrobněji popsané v teoretické části (NDVI, NDMI, SAVI). Jedná se o klíčovou komponentu v navrhované webové aplikaci. Pro manipulaci s daty vegetačních indexů byly vytvořeny tyto následující ovládací prvky (číselníky pro navolení dvou pásem rastru, tlačítka pro realizaci výpočtu indexu a tlačítka pro zobrazení výsledku v mapovém okně. Ovládací prvky byly navrženy tak, aby definice výpočtu byla pro uživatele co nejjednodušší. Každý vegetační index má svou sadu ovládacích tlačítek (obrázek 5.7). Možnost výběru z číselníků tak logicky odpovídají počítanému vegetačnímu indexu.



Obrázek 5.7 Ovládací panel pro výpočet vegetačních indexů

Funkce pro definici výpočtu vegetačního indexu

V momentě kdy uživatel vybere ze dvou pásem rastru a spustí výpočet tlačítkem „VÝPOČET...“ jsou hodnoty číselníků pásma x a pásma y odeslány přes http protokol a metodou `_POST` zpracovány v souborech *php* definující výrazy pro výpočet indexů. V tomto případě jsou hodnoty z číselníků uloženy v proměnných `$redpasmo` a `$nirpasmo`. Odtud jsou potom použity v proměnné `$sql`, definující SQL dotaz pro výpočet vegetačního indexu (obrázek 5.8). Provedením dotazu je přes AJAX zaslána odpověď serveru zpátky na klienta.

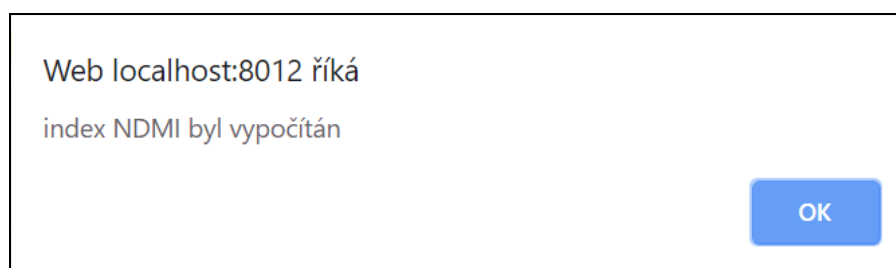
```
<?php
$redpasmo = $_POST['pasmo1'];
$nirpasmo = $_POST['pasmo2'];
$db = new PDO('pgsql:host=localhost;port=5432;dbname=diplomka;', 'postgres', 'sr');
$sql = $db->prepare("CREATE TABLE ndvi AS SELECT ST_MapAlgebra
((rast, 1, rast, 1, '((1 + 0.5) * ([rast1] - [rast2]) / ([rast1] + [rast2] + 0.5))::float', '32BF')
AS rast FROM
(SELECT a.rast as rast, b.rast as rast FROM $redpasmo a INNER JOIN $nirpasmo b ON a.rid = b.rid) as joined")
$sql->execute();
->
```

Obrázek 5.8 Volání SQL skriptu pro výpočet vegetačního indexu

Skript pro příjem zpracovaných dat (obrázek 5.9) na straně klienta je spuštěn funkcí, která je pomocí id svázána s konkrétním tlačítkem pro výpočet vybraného vegetačního indexu. Definice AJAXU začleňuje tři argumenty url php souboru, metodu POST pro zpracování dotazu, a data obsažená v odpovědi požadavku. Soubor php zastupuje výše uvedený skript „vegetacni_indexNDVI.php“, data jsou přejímána z číselníků opět pomocí jejich id (*#ndviRed.val*, *#ndviNIR.val*, *#ndvi.val*). Analogické skripty jsou i pro indexy NDMI a SAVI. V případě úspěšného provedení funkce pomocí *alert* je uživateli oznámeno úspěšné provedení výpočtu (obrázek 5.9b).

```
<script>
    $("#indexNDVI").click(function() {
        $.ajax({
            url:'vegetationIndicesNdvi.php',
            type:'POST',
            data:{
                ndviRed:$("#ndviRed").val(),
                ndviNir:$("#ndviNir").val(),
                ndvi:$("#ndvi").val(),
            },
            success: function(){
                alert("index NDVI byl vypočítán");
            }
        });
    });
</script>
```

Obrázek 5.9a Zpracování odpovědi na požadavek pro výpočet indexu AJAXEM



Obrázek 5.9b Dialog potvrzující provedení výpočtu indexu

5.4 Implementace senzorových dat

Senzorová data jsou měřena pomocí pěti senzorů umístěných v blízkosti katedry Geoinformatiky. Z těchto měřicích zařízení jsou data v intervalu 15 nebo 60 minut odesílána pomocí standardů Sigfox, Zigbee nebo GPRS na server katedry do PostgreSQL databáze „*senzor*“. Jeden odeslaný datový paket odpovídá jednomu databázovému záznamu v tabulce *envirodata* použitých pro korekční výpočet (obrázek 5.10).

	id_sensor	sensor	value	original_id	time	id
	text	text	numeric (5,2)	numeric	[PK] timestamp without time zone	[PK] bigint
1	enviro	Virrib	24.30	[null]	2019-07-20 00:00:00	143
2	enviro	RH	91.17	[null]	2019-07-20 00:00:00	144
3	enviro	BAT	9.17	[null]	2019-07-20 00:00:00	145
4	enviro	EC5	1.42	[null]	2019-07-20 00:00:00	146
5	enviro	TA	17.18	[null]	2019-07-20 00:00:00	147
6	enviro	RG	0.00	[null]	2019-07-20 00:00:00	148

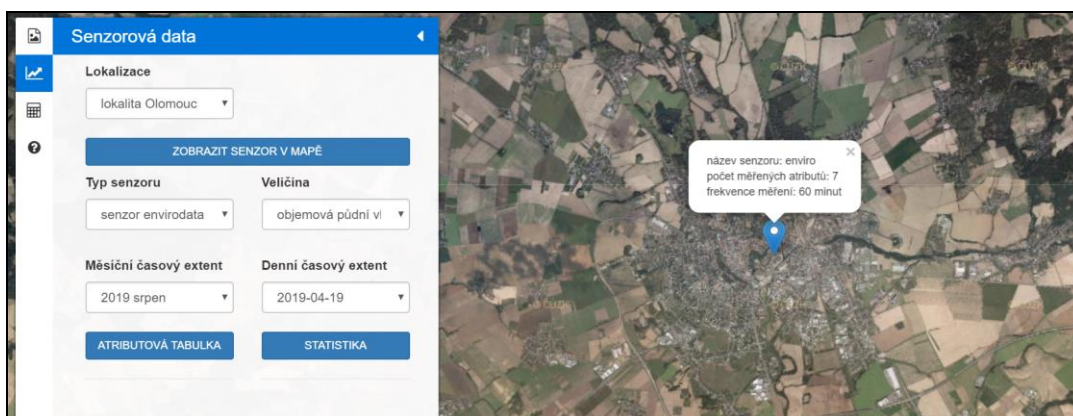
Obrázek 5.10 Datový záznam pro typ senzoru enviro

Lokalizace senzoru v mapě

Údaje senzorových dat jsou extrahovány z databáze na serveru stejným způsobem jako rastry pomocí PHP a AJAXU. Procesní část na straně serveru je vykonána php skriptem a výsledek je asynchronně načten Ajaxem do klienta. Data o poloze obsahuje tabulka *sensor_location*. Odtud jsou selectem získány souřadnice senzoru. Server vrátí string souřadnic na stranu klienta, kde jsou dále Javascriptem zpracovány. Řetězec souřadnic je funkcí *sensorParser* rozdělen na dvě části, které jsou předány do funkce *getSensorCoord*. Přidaný bod je v Leafletu definovaný elementem *L.marker([x,y]).addTo(map)* (obrázek 5.11). Pro zobrazení stačí v aplikaci vybrat z pole *Lokalizace* příslušný záznam s názvem lokality a výběr potvrdit tlačítkem „*zobrazit senzor v mapě*“ (obrázek 5.12).

```
function getSensorCoord(res) {
    var textlx = res[0],
        textly = res[1];
    map.panTo([textlx, textly]);
    L.marker([textlx, textly]).addTo(map).bindPopup("název senzoru: enviro  
<br> počet měřených atributů: 7 <br> frekvence měření: 60 minut").openPopup();
}
```

Obrázek 5.11 Funkce pro načtení senzoru v mapě



Obrázek 5.12 Zobrazení senzoru v aplikaci

Výpočet průměrné hodnoty měřené senzorové veličiny

Proces implementace senzorových dat postupně prochází přes výpočet mediánové nebo průměrné hodnoty měřených veličin. Statistický výpočet průměru nebo mediánu je prováděn nad tabulkou „*envirodata*“ v databázi „*sensor*“ na serveru KGI. Ke statistickým výpočtům nad databází byly použity dva SQL příkazy. Na obrázku 5.13 je obsažena syntaxe pro získání mediánu z vybraného počtu záznamů sloupce v tabulce. Medián je reprezentován funkcí „*ROUND(PERCENTILE_CONT(0.5)*“. Dále musí být použita funkce pro výběr sloupce, nad kterým má být medián počítaný „*WITHIN GROUP(ORDER BY nazev sloupce)::numeric, 3*“. Syntaxe pro aritmetický průměr je podobná (obrázek 5.14), pouze funkce pro výpočet je odlišná „*SELECT to_char(AVG(nazev_sloupce, 9999D9999))*“. Datový výběr lze omezit pomocí podmínky *WHERE* na konci *SELECT* příkazu. Výsledek datového výběru je pomocí funkce *echo* formátován a v podobě html tabulky vrácen jako odpověď v aplikaci.

```
$sqlMedian = $dbsensor->prepare
("SELECT ROUND(PERCENTILE_CONT(0.5)
WITHIN GROUP (ORDER BY $merenaVelicina)::numeric, 3) AS median_item_count
FROM $sensorType
WHERE $merenaVelicina <> 0 AND timestamp::text LIKE '$datetime%');
$sqlMedian->execute();

echo "<table id= 'dtHorizontalVerticalExample'
class= 'table table-striped table-bordered table-sm'
cellspacing='0', width= '100%'>";
echo "<tr><th>medián</th></tr>";
while ($row = $sqlMedian->fetch(PDO::FETCH_ASSOC)) {
    echo "<tr>";
    foreach($row as $field=>$value) {
        echo "<td>{$value}</td>";
    }
    echo "</tr>";
}
echo"</table>";
```

Obrázek 5.13 Server skript pro výpočet mediánu

```
$sqlMean = $dbsensor->prepare
("SELECT to_char(AVG ($merenaVelicina), '9999999999999999D999') AS průměr
FROM $sensorType WHERE timestamp::text LIKE '$datetime%');
$sqlMean->execute();

echo "<table id= 'dtHorizontalVerticalExample'
class= 'table table-striped table-bordered table-sm'
cellspacing='0', width= '100%'>";
echo "<tr><th>průměr</th></tr>";
while ($row = $sqlMean->fetch(PDO::FETCH_ASSOC)) {
    echo "<tr>";
    foreach($row as $field=>$value) {
        echo "<td>{$value}</td>";
    }
    echo "</tr>";
}
echo"</table>";
```

Obrázek 5.14 Server skript pro výpočet aritmetického průměru

V SQL řetězci jsou zadefinovány čtyři proměnné, kterými lze výsledek upravovat podle potřeby (typ senzoru, specifikace roku a měsíce, specifikace dne, měřená veličina). Po výběru hodnot z těchto číselníků (obrázek 5.15) je tlačítkem „STATISTIKA“ zahájen výpočet a dojde k načtení výsledku v podobě průměrné a mediánové hodnoty do tabulek v postranním panelu (obrázek 5.16). Přehled atributové tabulky obsahující všechny atributy je vyvolán tlačítkem „ATRIBUTOVÁ TABULKA“.

Obrázek 5.15 Grafické rozhraní pro definici statistického výpočtu z hodnot ze senzoru

medián
70.969
průměr
72,616

Obrázek 5.16 Výsledek dotazu na medián a průměr sensorové veličiny

Výpočet rozdílu sensorových a rastrových dat

Výpočet, při kterém je stanoven rozdíl mezi hodnotou pixelů vypočítaného vegetačního indexu, a průměrnou hodnotou vybrané sensorové veličiny je získán rozdílový rastr. Ten v závislosti na intervalu hodnot vegetačního indexu NDMI může nabývat hodnot od minina - 1 do maxima + 1 podle toho jak moc přesný byl výpočet vegetačního indexu. V optimální situaci by rozdíl hodnoty pixelu a hodnoty měřené sensorové veličiny měl být roven nule. V takovém případě lze hodnotu pixel vypočítaného vegetačního indexu v porovnání s aktuálními klimatickými a meteorologickými daty považovat za velice přesný. Výpočet byl v praktické části nasazen na vegetační index NDMI, který je z vybraných indexů nejlépe aplikovatelný na detekci půdní vlhkosti. Od vegetačních indexů NDMI z Landsatu 8 a Sentinelu 2 byla pro odečet ve výpočtu použita objemová půdní vlhkost.

Výpočet vegetačního indexu opraveného o korekční hodnotu objemové půdní vlhkosti (*virrib*) nebo relativní vzdušné vlhkosti (*rh*) je v prvním kroku definován stejně. Uživatel vybere dvě vstupní pásma zdrojového rastru. Ve druhém kroku zadá do pole „*Hodnota senzorové veličiny*“ číselnou hodnotu, která vznikla při dotazování na senzorová data *virrib* nebo *rh* na předchozí záložce. Nakonec stačí vybrat název pro výsledný korekční rastr. Po spuštění výpočtu tlačítkem „VYTVOŘIT PŘEPOČÍTANÝ INDEX“ dojde k uložení tohoto rastru do databáze. Ovládací prvky tohoto postranního panelu jsou uvedeny na obrázku 5.17. Po dokončení tohoto procesu má uživatel možnost u korekčního rastru provést reklasifikaci do kategorií. Základní reklasifikaci rastru umožňuje přímo *PostGIS* databáze. V aplikaci je tato funkce spuštěna tlačítkem „REKLASIFIKOVAT PŘEPOČÍTANÝ RASTR“, vstupem je vytvořený přepočítaný rastr, který uživatel vybírá z pole „*Vytvořit přepočítaný vegetační index NDMI*“ Nastavení intervalu hodnot pro reklasifikaci je uvedeno v kapitole 6.

The image shows a web application control panel titled "Výpočet indexu NDMI s korekcí". It contains several sections with dropdown menus and buttons:

- NIR pásmo:** A dropdown menu with "Landsat 8 NIR" selected.
- SWIR pásmo:** A dropdown menu with "Landsat 8 SWIR" selected.
- Hodnota senzorové veličiny:** An empty text input field.
- Vytvořit přepočítaný vegetační index NDMI:** A dropdown menu with "Landsat 8 podle relativní vzdušné vlhkosti (rh)" selected.
- Buttons:** A blue button labeled "VYTVOŘIT PŘEPOČÍTANÝ INDEX" and another blue button labeled "REKLASIFIKOVAT PŘEPOČÍTANÝ INDEX".
- Vstupní rastr pro reklasifikaci:** A dropdown menu with "Landsat 8 podle relativní vzdušné vlhkosti (rh)" selected.
- Výstupní rastr pro reklasifikaci:** A dropdown menu with "Reklasifikace Landsat 8 podle(rh)" selected.

Obrázek 5.17 Ovládací panel pro výpočet rozdílového rastru

6 VÝSLEDKY

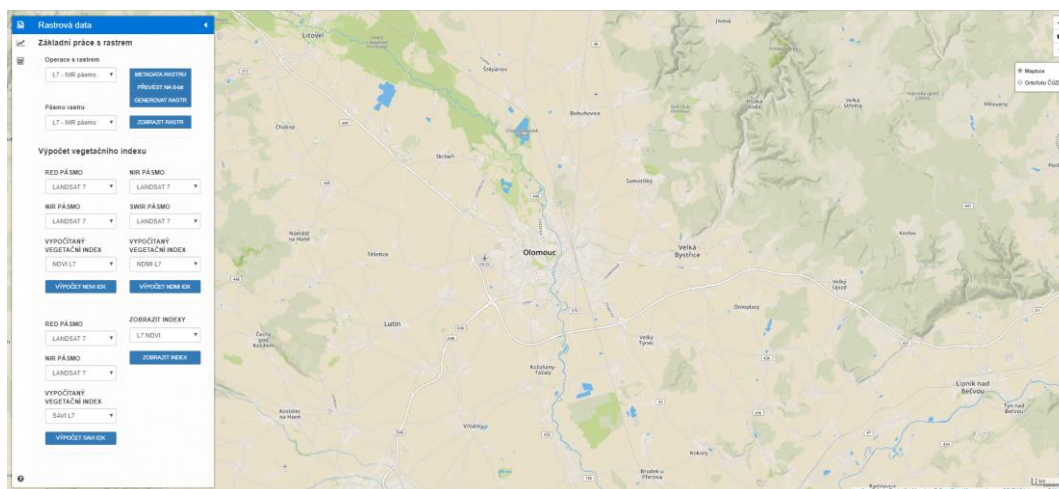
Hlavním výsledkem práce podle předlohy zadání je návrh webové aplikace, která zpracovává rastrová data a na výstupu produkuje výsledek v podobě vypočítaného vegetačního indexu. Zároveň aplikace integruje sensorová data, zobrazitelná nad rastrovými daty. Následně je s pomocí těchto dat možné provádět jednoduché korekční výpočty a srovnávat vypočítané vegetační indexy s daty získanými pozemním sensorovým měřením. Výsledky práce shrnují hlavní komponenty aplikace i doplňující výstupy vzniklé při tvorbě diplomové práce (zdrojový kód aplikace, manuál pro předzpracování a import dat do PostgreSQL/PostGIS databáze, manuál popisující práci s daty přímo v aplikaci, sadu zpracovaných rastrů a vegetačních indexů).

Mapová aplikace

Prvním výstupem je samotná mapová aplikace postavená na technologiích *PostgreSQL/PostGIS*, *PHP* a *Javascriptu*. Aplikace je provozovaná na webovém serveru KGI a je spustitelná z jakéhokoliv webového prohlížeče. Dostupná je pod odkazem <http://www.geoinformatics.upol.cz/dprace/magisterske/roubalik19/app/>.

Funkcionalita je určena na zpracování dvou typů dat, rastrových a sensorových. Součástí uživatelského rozhraní je hlavní mapové okno (obrázek 6.1) a čtyři varianty postranního ovládacího panelu.

První slouží k základní práci s rastrem (dotaz na metadata nebo tlačítko pro export rastru z databáze do jpeg formátu). Pomocí dalších tlačítek na záložce lze počítat tři typy vegetačních indexů (NDVI, NDMI, SAVI). Na druhé záložce jsou navrženy prvky a tlačítka pro dotazování databáze se sensorovými daty. Lze tak pro definované období v intervalu 24 hodin vypočítat průměrnou a mediánovou hodnotu pro teplotu, objemovou půdní vlhkost, srážky a další měrné jednotky. Předposlední záložka nabízí možnost výpočtu indexu NDMI upraveného o průměrnou nebo mediánovou hodnotu objemové půdní vlhkosti nebo relativní vzdušné vlhkosti získané pomocí sensorového měření. Výsledkem tohoto výpočtu je rastr rozdílových hodnot. Význam hodnot tohoto rastru je popsán dále v této kapitole. Poslední postranní záložka slouží jako základní okno pro nápovědu s odkazem na technickou dokumentaci aplikace. K těmto dvěma prvkům jsou doplněny další widgety jako měřítko, přepínač podkladových vrstev (Mapbox, Ortofoto ČR), tlačítka na Zoom In/Out, zobrazování souřadnic v závislosti na poloze kurzoru myši. Celkem aplikace nabízí 15 ovládacích prvků, 3 druhy výpočtu a 6 dalších typů operací, které byly podrobněji popsány v předchozích kapitolách 4 a 5.



Obrázek 6.1 Finální vzhled mapové aplikace

Zdrojový kód

Pro opětovnou použitelnost a další vývoj byl zdrojový kód verzován pomocí nástroje *Git*. V průběhu práce byly provedeny desítky aktualizací v souladu s principem agilního vývoje. Po dokončení prototypu aplikace byl zdrojový kód uložen na cloudové úložiště Github. Zároveň je součástí přílohy 6 (DVD).

Zpracované rastry a vegetační indexy

Rastrová data specifikovaná v kapitole 2.2 a 3.3 prošla úpravami buď přímo v prostředí aplikace, nebo ještě před importem došlo k předzpracování v programu *SNAP Desktop*. Tato testovací data na výstupu zahrnují pro každou ze tří datových sad (Landsat 7 a 8, Sentinel 2) vždy tři rastry reprezentující červené RED pásmo, blízké infračervené NIR pásmo, a střední infračervené SWIR pásmo. Výpočtem a kombinací těchto barevných pásem vzniklo celkem devět vypočítaných vegetačních indexů, vždy tři typy podle zdrojových dat. Výpočet indexů byl primárně proveden pomocí webové aplikace a pro kontrolu byl pro stejná data zopakován i v programu *SNAP Desktop*. Následně bylo provedeno srovnání výsledků, které je shrnuto v tabulce 6.1. Z tohoto srovnání je vidět že přesnost výpočtu vegetačního indexu nad databází *PostgreSQL/PostGIS* odpovídá výsledkům z programu *SNAP Desktop* s přesností na čtyři desetinná místa.

Tabulka 6.1 Srovnání výpočtu vegetačních indexů v PostGIS databázi a ve SNAP desktop

index	Interval hodnot PostGIS min	Interval hodnot PostGIS max	Interval hodnot SNAP min	Interval hodnot SNAP max
L7 NDVI	-0.111888	0.690476	-0.111888	0.690472
L7 NDMI	-0.585366	0.221477	-0.585362	0.221475
L7 SAVI	-0.167539	1.03264	-0.167539	1.03264
L8 NDVI	-0.999991	0.999996	-1.000000	1.000000
L8 NDMI	-0.999994	0.999992	-1.000000	1.000000
L8 SAVI	-1.49362	1.49706	-1.49362	1.49706
S2 NDVI	-0.999991	0.999996	-1.000000	1.000000
S2 NDMI	-0.999992	0.999989	-1.000000	1.000000
S2 SAVI	-1.49302	1.49703	-1.49302	1.49703

Pro typ vegetačního indexu NDMI z družic *Landsat 8* a *Sentinel 2* lze dále provést korekční výpočet. Od rastru vegetačního indexu je odečtena průměrná nebo mediánová hodnota objemové půdní vlhkosti nebo relativní hodnota vzdušné vlhkosti získaná za 24 měření v průběhu dne. Jednotky této veličiny jsou v procentech převedeny do intervalu přibližně od -1 do 1 v závislosti na indexu. Časový interval pro sensorová data musí odpovídat dnu pořízení satelitních dat, ze kterých je vypočítán vegetační index. Provedením výpočtu je na výstupu získán rastr rozdílů hodnot v intervalu od minus jedné do jedné. Je-li hodnota pixelu tohoto rastru rovna nule, znamená to, že hodnota téhož pixelu pro rastr vegetačního indexu odpovídá průměrné nebo mediánové hodnotě získané ze senzoru. Čím více záporná nebo kladná hodnota rozdílového rastru je, tím méně odpovídá hodnota pixelu vegetačního indexu hodnotě objemové půdní vlhkosti. Pro kategorizaci pixelů výsledného rastru lze provést reklasifikaci. Kvůli chybnému načítání tohoto typu rastru do aplikace byla vizualizace výsledku provedena v *QGISU*.

Po načtení rozdílového rastru z databáze (obrázek 6.2b) bylo provedeno nastavení reklasifikovaného rastru. Na základě vstupního intervalu byly stanoveny čtyři kategorie. Cílem bylo selektovat pixely na kategorie, kde je rozdíl hodnot do 10% jak v kladném tak záporném směru a kde je rozdíl větší než 10% v kladných i záporných hodnotách. Přehled kategorií je vysvětlen v tabulce 6.2 a podoba reklasifikovaného rozdílového rastru je na obrázku 6.3.



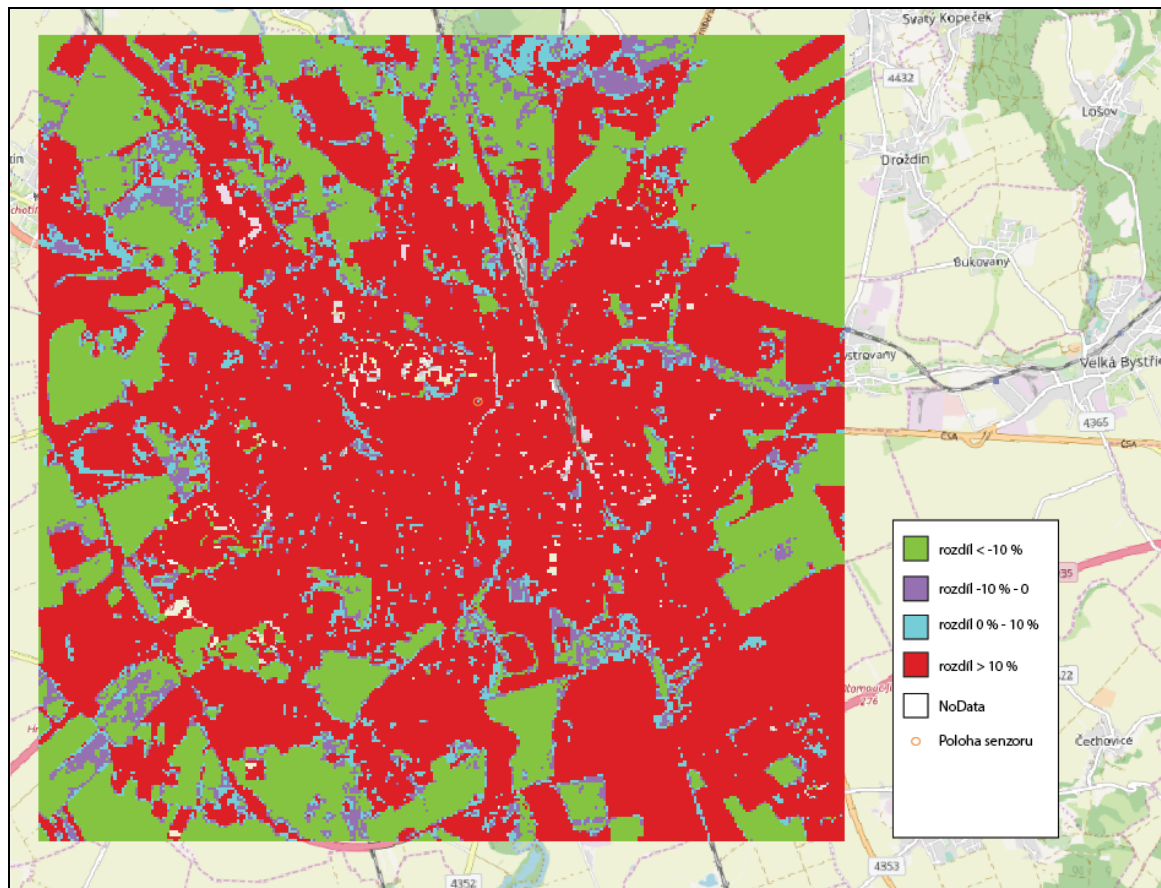
Obrázek 6.2a Načtení vypočítaného rozdílového rastru NDMI v QGISu bez korekce



Obrázek 6.2b Načtení vypočítaného rozdílového rastru NDMI v QGISu s korekcí

Tabulka 6.2 Převodní tabulka pro reklasifikaci rozdílového rastru

Číslo kategorie	Název kategorie	Interval hodnot	Barevné označení
1	Výrazně záporný rozdíl hodnot	(-MIN, -0.1)	zelená
2	Mírně záporný rozdíl hodnot	<-0.1, 0)	fialová
3	Mírně kladný rozdíl hodnot	<0, 0.1>	světle modrá
4	Výrazně kladný rozdíl hodnot	(0.1, MAX)	červená
NoData	NoData	NoData	Bílá



Obrázek 6.3 Reklasifikovaný rozdílový rastr do 4 kategorií zobrazený v QGISu

7 DISKUZE

Podmínkou úspěšného vývoje libovolného softwaru je dodržení základních principů softwarového inženýrství. Při tvorbě praktické části byla snaha klást důraz na jednotlivé části softwarového projektu. V průběhu vývoje prototypu byly objeveny technické nedostatky. Řešení části z nich je obsahem technického popisu předcházejících kapitol. Nevyřešené nedostatky do spuštění testovacího prototypu aplikace posloužily jako podklady pro sepsání následující diskuze.

Z mnoha požadavků se povedlo integrovat funkce pro výpočet tří typů indexů, což posloužilo jako základ pro budování další funkcionality. Procesní zpracování se povedlo nasadit také na senzorová data a jejich využití ve výpočtu vegetačních indexů. Mimo výpočetních úkolů se povedlo propojit rastrový obsah databáze PostGIS do mapové aplikace vytvořené v Leafletu. Načítaný obsah je díky funkcím Javascriptu interaktivní.

Podrobnější analýza požadavků

K základní analýze požadavků řešení a návrhu fungování aplikace došlo ještě před začátkem samotného vývoje. Průběh ale prokázal, že ne všechny předpokládané části řešení v kombinaci se zvolenými technickými řešeními umožní dosáhnout požadovaných výsledků. Před dalším vývojem by proto měla být prioritní podrobnější analýza, testování vytvořeného prototypu a zvážení dalších kroků vývoje.

Import dat do databáze

Prototyp aplikace počítal s procesní a vizualizační komponentou. Otázkou od samého počátku zůstávalo, jakým způsobem bude aplikace schopná číst data z externích zdrojů nativně bez nutnosti využívat nástroje třetích stran. V průběhu analýzy se nepovedlo nalézt adekvátní řešení zapadající do technického konceptu, a proto bylo ve finále představeno pouze prozatímní řešení importu do databáze. Tento import bere v potaz využití příkazové řádky a nástroje *raster2pgsql.exe*. Podrobněji je proces popsán v kapitole 4.1. Za velkou nevýhodu nutno považovat fakt, že toto řešení vyžaduje instalaci systému *PostgreSQL* a přístupové údaje ke zdrojové databázi na straně uživatele. Pro import přímo z uživatelského rozhraní bude do budoucna nezbytné nalézt způsob jak pomocí skriptovacího jazyka transformovat zdrojová rastrová data do schématu pod kterým jsou ukládány rastrová data v databázi *PostgreSQL/PostGIS*. Druhou větví spadající pod import datových zdrojů může být proces stahování dat z cloudových úložišť přímo do databáze pomocí již existujících API. Tento proces vynechá jeden krok oproti výše popsanému importu. Vynechání kroku stažení dat nejdříve na lokální disk ke klientovi, a posléze jejich ukládání z lokálního zařízení přes interface aplikace do prostorové databáze.

Pro potřeby testování na cvičných datech je objem dat v databázi snížen na minimální počet záznamů v každé tabulce. Jeden záznam ve struktuře rastrové tabulky odpovídá jednomu uloženému rastru. Pro odstraňování dočasně vytvořených dat a tabulek byly vypracovány skripty pro jejich redukci. Budoucí nasazení do ostrého provozu bude však hlavně pro rastrová data požadovat operace na straně databáze tak, aby byla zajištěna konzistence ukládaných záznamů. Protože současný import neumožňuje podrobnější odlišení záznamů v databázi, tak aby klient mohl na straně aplikace tyto záznamy od sebe spolehlivě rozlišit, bude nezbytné v budoucnu navrhnout funkce nebo triggeru které při importu tyto rozlišovací atributy vytvoří (například timestamp importu, geolokace) a v aplikaci budou sloužit jako filtr dotazu na rastr.

Skriptovací jazyk pro Backend

Na počátku vývoje existoval návrh na zpracování řešení využívající pro backend i jazyk *Python*. Tento paralelní vývoj byl pro potřeby diplomové práce dočasně pozastaven. Důvodem pro tohle rozhodnutí byla snadnější implementace a kombinace jazyka *PHP* s *Javascriptem* a *AJAXEM*. Díky výše popsaným rezervám v řešení není návrat k Pythonu a jeho začlenění do řešení vyloučený. Potenciál skrytý v prostorových knihovnách nabízí do budoucna rozvoj jak pro importní, procesní i vizualizační komponenty.

Rozvoj aplikační vrstvy řešení

Navrhnuté řešení je postaveno na vícevrstvé architektuře. Koncept tenkého klienta zahrnuje na straně prezentační vrstvy pouze jednoduché uživatelské rozhraní. Aplikační a databázová vrstva je koncentrována na serverovou část řešení. Webový server ale slouží pouze jako nástroj na přijímání a odesílání požadavků z klienta a veškeré procesní operace provádí až samotný databázový server. Oproti komplexnějším webovým řešením zde prozatím není mezi aplikací a databázový server postaven výkonnější server aplikační. Absence aplikačního serveru přináší nevýhody v podobě snížených možností vlastní konfigurace procesů, které jsou ve finále prováděny nad databází.

Knihovna *Leaflet* byla upřednostněna před dalšími open source webovými mapovými řešeními. Na výběr bylo několik dalších alternativ jako *Mapbox*, *OpenLayers*, nebo API různých webově orientovaných aplikací zaměřující se na prostorová data. Široká základna uživatelů, rozsáhlá technická dokumentace a velká flexibilita v rámci editace zdrojového kódu byly hlavní důvody sázky právě na *Leaflet*. Protože ale navržená architektura celého řešení důsledně odděluje frontend a backend je do budoucna ve hře varianta na začlenění další knihovny, API nebo vlastního řešení uživatelského rozhraní i s ohledem na zvyšující se nároky na manipulaci s daty v prostředí aplikace. Současné řešení prozatím nepodporuje v postranním panelu načítání všech vrstev, které obsahuje mapové okno. Za nedostatek musí být brán i fakt, že vypočítané vegetační indexy v *PostGIS* jsou podporovány pouze jako 32 bitové rastry. Tento typ nelze exportovat ve formátu *jpeg* ve kterém načítá data *Leaflet*. Výsledky tak musí být zobrazeny v jiném programu například *QGISU*. Tento nedostatek zůstává největším otazníkem pro budoucí řešení.

Snahou při návrhu a vývoji aplikace byla integrace sensorových dat. Praktická část ukázala, že ne všechny zamýšlené kroky v rámci integrace těchto dat byly naplněny. Podařilo se realizovat extrakci potřebných bodových sensorových dat a využití jejich hodnot při výpočtech rozdílu a srovnávání s hodnotami pixelů satelitních dat s těmito konkrétními hodnotami veličin měřených senzory. Část zabývající se integrací sensorových dat musela být ve finále okleštěna o interpolaci hodnot sensorových dat a také o výpočet rozdílu rastrů vegetačních indexů a těchto interpolovaných dat. Tvorbu těchto výpočtů zohledňující prostorové rozmístění více sensorů nad jedním rastrovým snímkem znemožnilo několik faktorů. Zaprvé omezené možnosti interpolace bodových dat přímo v databázi. Komponenta interpolace sensorových dat nabízí další příležitost pro budoucí vývoj operací na serveru. Druhým významným omezením byla skutečnost, že dlouhou dobu nebylo z provozních důvodů jasné, kde budou senzory v rámci studované oblasti umístěny. Naproti tomu se podařilo ověřit, že korekční výpočet je technicky možné pomocí aplikace provést. V budoucím testování by měla být vybrána oblast menšího území mimo město a s nasazením více měřících zařízení. Výpočet sensorových hodnot bude potom možné provést s větší přesností.

8 ZÁVĚR

Základním cílem práce bylo vyvinout prototyp webové aplikace s požadovanou funkcionalitou postavené na předem vytipovaných open source technologiích. Tento cíl pro klíčové požadavky zadání byl splněn.

Práce zahrnuje dvě hlavní části – teoretická popisuje výpočty vegetačních indexů, fungování databáze PostgreSQL/PostGIS nebo technické specifikace datových zdrojů. Základem praktické části je popis vývoje aplikace a všech jejích komponent. Řešení obsahuje jak návrh procesních funkcí pro výpočty nad databázemi, tak i provázání serverové komponenty s uživatelským rozhraním aplikace programované primárně pomocí jazyka Javascript. Po skončení vývoje byl prototyp nasazen do zkušebního provozu na server Katedry geoinformatiky. Byly provedeny zkušební výpočty vegetačních indexů a zkoušky načítání dat ze senzorů. Výpočty pomocí funkcí v PostGIS databázi odhalily, že zpracování rastrových dat může být tímto způsobem přesné a být rovnocenným řešením jiným aplikacím pro zpracování dat Dálkového průzkumu Země.

Zároveň se potvrdil předpoklad, že největší problém bude s datovými formáty. Import dat do databáze musel být řešen pomocí programu třetí strany a stejně tak export některých dat prozatím nezvládá aplikace nativně. Přes výčet omezení, se povedlo vytvořit základ pro budování platformy na bázi lehkého klienta. Navržený teoretický proces integrace hodnot ze senzorových dat do výpočtu vegetačního indexu prozatím byl testován jen na omezeném počtu dat. Pro ověření přesnosti a účinnosti výpočtu bude muset být testováno nasazení ve specifické oblasti primárně určené pro zemědělství. Aplikace je publikována pod volně šiřitelnou licencí a je otevřená možnostem dalšího vývoje.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

Knižní zdroje:

DOBROVOLNÝ, Petr. Dálkový průzkum Země: digitální zpracování obrazu. Brno: Masarykova univerzita, 1998. ISBN 80-210-1812-7.

HALOUNOVÁ, Lena. Dálkový průzkum Země. Praha: Vydavatelství ČVUT, 2005. ISBN 80-01-03124-1.

OBE, Regina O a Leo S HSU. PostGIS in action. Second edition. Shelter Island, NY: Manning, [2015]. ISBN 9781617291395.

PECHANEC, V.; REJŠEK, K.; MRÁZ, A.; SVOBODOVÁ, J.; KILIANOVÁ, H.; VRANOVÁ, V. Nasazení multispektrálních optických senzorů pro monitorování vlastností půdy. 2017. ISBN 978-80-244-5111-4. Metodika. Univerzita Palackého v Olomouci.

SOMMERVILLE, I. Softwarové inženýrství. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.

Elektronické zdroje:

ARÉVALO, J. PostGIS WKT Raster. An Open Source alternative to Oracle GeoRaster. konference FOSS4G Barcelona [online]., 2010 [cit. 2018-09-29]. Dostupné z URL: <<http://2010.foss4g.org/presentations/3814.pdf>>.

Clever Maps [online]. 2018 [cit 2018-09-29]. Dostupné z: <https://www.clevermaps.cz/projekty#farm>

Clever Farm [online]. 2018 [cit 2018-09-29]. Dostupné z: <https://www.cleverfarm.cz/>

Data Formats. ESA [online]. 2018f [cit. 2018-02-09]. Dostupné z: <https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/data-formats>

Družice Sentinel. GEO/COPERNICUS v České republice [online]. 2018 [cit. 2018-02-17]. Dostupné z: <http://copernicus.gov.cz/druzice-sentinel>

FRAMPTON, W. J.; DASH, J.; WATMOUGH, G.; MILTON, E. J. Evaluating the capabilities of Sentinel-2 for quantitative estimation of biophysical variables in vegetation. ISPRS Journal of Photogrammetry and Remote Sensing [online]. 2013, 82, s. 83-92 [cit. 2018-02-09]. DOI: 10.1016/j.isprsjprs.2013.04.007. ISSN 09242716.

Gisat [online]. 2018 [cit 2018-09-29]. Dostupné z: <http://gisat.cz/content/cz/druzicova-data/objednani-dat/zakladni-pojmy>

HUNT, E. R.; DORAISWAMY, P. C.; MCMURTREY, J. E.; DAUGHTRY, C. S. T.; PERRY, E. M. AKHMEDOV, B. A visible band index for remote sensing leaf chlorophyll content at the canopy scale. International Journal of Applied Earth Observation and Geoinformation [online]. 2013, 21, s. 103-112 [cit. 2018-02-09]. DOI: 10.1016/j.jag.2012.07.020. ISSN 03032434.

HUETE, A. R. A soil-adjusted vegetation index (SAVI). Remote Sensing of Environment [online]. 1988, 25(3), s. 295-309 [cit. 2018-02-09]. DOI: 10.1016/0034-4257(88)90106-X. ISSN 00344257.

Index: Green leaf index. Index DataBase [online]. 201a [cit. 2018-02-09]. Dostupné z: <https://www.indexdatabase.de/db/i-single.php?id=375>

Index: Normalized Difference NIR/Red Normalized Difference Vegetation Index, Calibrated NDVI - CDVI. Index DataBase [online]. 2018b [cit. 2018-02-09]. Dostupné z: <https://www.indexdatabase.de/db/i-single.php?id=58>

Index: Normalized Difference 680/430 Normalized Pigment Chlorophyll Index. Index DataBase [online]. 2018c [cit. 2018-28-09]. Dostupné z: <https://www.indexdatabase.de/db/i-single.php?id=62>

Index: Soil Adjusted Vegetation Index. Index DataBase [online]. 2018d [cit. 2018-02-09]. Dostupné z: <https://www.indexdatabase.de/db/i-single.php?id=87>

Leaflet tips and trick. [online]. 2017 [cit 2018-04-19] dostupné z: https://leanpub.com/leaflet-tips-and-tricks/read?fbclid=IwAR3_4Ci_KFuRP1uiU8oGxIhrNJ--6r11tasPldU8vxTvqFYqWinyV5XfPM

Murray, Chuck. ORACLE. Oracle Spatial Developer's Guide, 11g Release 1 (11.1) [online]. 2009 [cit. 2014-11-01]. Dostupné z: http://docs.oracle.com/cd/B28359_01/appdev.111/b28400.pdf.

RONNEY, Nick. Creating raster from scratch in PostGIS 1. [online]. 2017 [cit 2018-04-19] Dostupné z: https://nronnei.github.io/blog/2017/03/creating-rasters-from-scratch-in-postgis/?fbclid=IwAR2y51U_FWj5HvJSXBtoP4uMTFZYkm3Wbw-wzM7wcAUwRmPt4dFWWhoW4JxU

RONNEY, Nick. Creating raster from scratch in PostGIS 2. [online]. 2017 [cit 2018-04-19] Dostupné z: <https://nronnei.github.io/blog/2017/03/creating-rasters-from-scratch-in-postgis-pt2/?fbclid=IwAR2Xy229hPOLRkIcT0Z1EE2A8ZqY2-x8DsICqPTyCm5iZ-WZRdeuH-cp4hE>

OpenGIS Implementace pro Simple feature část 2. [online] [cit.2018-09-19]. Dostupné z URL: <<http://www.opengeospatial.org/standards/sfs>>.

PODHRÁZKÝ, Adam. Hodnocení vlivu přírodních liniových prvků na okolní agroekosystémy pomocí DPZ a UAV. Diplomová práce [online]. 2017 [cit 2018-05-12]. Dostupné z: https://is.muni.cz/th/lawo2/DP_Podhrazsky.pdf

POHANKA, T., PECHANEC, V., HEJLOVÁ v. Python web server for sensor data visualisation. NFORMATICS, GEOINFORMATICS AND REMOTE SENSING CONFERENCE PROCEEDINGS, SGEM 2016, VOL I. Sofia: STEF 92 Technology Ltd., 2016

PostGIS Manual [online]. 2018 [cit. 2018-09-01]. Dostupné z: <http://postgis.net>

PostgreSQL: Manuals [online]. 2012 [cit. 2016-03-05]. Dostupné z: <https://www.postgresql.org/docs/manuals/>

RACINE, Pierre. Store, manipulate and analyze raster data within the PostgreSQL/PostGIS spatial database. Sborník konference FOSS4G Denver [online]., 2011 [cit. 2018-09-29]. Dostupné z URL: <http://trac.osgeo.org/postgis/raw-attachment/wiki/WKTRaster/PierreRacine_FOSS4G-2011.pdf>.

RYCHLÝ, Marek. Prostorové a XML databáze In: [online]. 2014 [cit. 2015-03-20]. Dostupné z:http://www.fit.vutbr.cz/rychly/public/docs/slides-demo2-spatial_and_xml/sli.

QI, J.; CHEHBOUNI, A.; HUETE, A. R.; KERR, Y. H.; SOROOSHIAN, S. A modified soil adjusted vegetation index. *Remote Sensing of Environment* [online]. 1994, 48(2), s. 119-126 [cit. 2018-02-09]. DOI: 10.1016/0034-4257(94)90134-1. ISSN 00344257.

ROUSE, J. W. a kol. Monitoring vegetation systems in the Great Plains with ERTS. United States, 1974. s. 309-317. Texas A&M Univ.; College Station, TX.

Sentinel Hub [online]. 2019 [cit 2019-02-10]. Dostupné z: <https://www.sentinel-hub.com/eoproducts/ndvi-normalized-difference-vegetation-index>

VINCINI, M.; FRAZZI, E.; D'ALESSIO, P. A broad-band leaf chlorophyll vegetation index at the canopy scale. *Precision Agriculture* [online]. 2008, 9(5), s. 303-319 [cit. 2018-02-09]. DOI: 10.1007/s11119-008-9075-z. ISSN 1385-2256.

VINCINI, M.; FRAZZI, E.; D'ALESSIO, P. Comparison of narrow-band and broad-band vegetation indexes for canopy chlorophyll density estimation in sugar beet. *Precision agriculture '07*. Wageningen, The Netherlands: Wageningen Academic Publishers, 2007, s. 189-196. ISBN 9789086860241.

Výzkumný ústav geografický, topografický a kartografický. Terminologický slovník zeměměřictví a katastru nemovitostí [online]. 2016 [cit 2018-09-25]. Dostupné z: <https://www.vugtk.cz/slovník/termin.php?jazykovaverze=&tid=1216&l=prostorova-data>.

Webová stránka Agricolous [online]. 2018 [cit 2018-11-23]. Dostupné z: <https://www.agricolous.com/en/indici-vegetazione-ndvi-ndmi-istruzioni-luso/>

Workshop Postgis raster [online]. 2019 [cit 2019-05-11]. Dostupné z: <https://github.com/lcalisto/workshop-postgis-raster/blob/master/README.md>

SEZNAM ILUSTRACÍ

- Obrázek 1.1 – Obecné schéma aplikace
- Obrázek 2.1 – Velikost základních informací o senzoru
- Obrázek 2.2 – Velikost doplňkových informací o senzoru
- Obrázek 2.3 – Postup zpracování teoretické a praktické části
- Obrázek 2.4 – Workflow zpracování dat
- Obrázek 2.5 – Prvotní wireframe aplikace
- Obrázek 3.1 – End of Life verze PostgreSQL 9
- Obrázek 3.2 – Propojení databáze PostgreSQL a prostorové extenze PostGIS
- Obrázek 3.3 – Uložení vícepásmového rastru v PostGIS
- Obrázek 3.4 – Schéma ArcSDE
- Obrázek 3.5 – Vývojová větev Oracle Spatial od verze 7
- Obrázek 3.6 – Struktura uložení rastru v databázi DB2
- Obrázek 3.7 – Porovnání pásem Landsat 8 a Landsat 7
- Obrázek 3.8 – Technické parametry senzorů
- Obrázek 3.9 – Aplikace Clever Farm
- Obrázek 3.10 – Rozhraní Sentinel Playground
- Obrázek 3.11 – Toolbox platformy EOS
- Obrázek 3.12 – Výpočet vegetačního indexu NDVI v platformě EOS
- Obrázek 4.1 – Proces exportu rastru z PostGIS databáze do formátu jpeg
- Obrázek 4.2 – Proces tvorby tabulky vegetačního indexu v PostGIS databázi
- Obrázek 4.3 – Možné provázání geometrické a atributové části sensorových dat
- Obrázek 4.4 – Proces odečtu hodnoty veličiny pořízené senzorem od hodnot pixelů vegetačního indexu
- Obrázek 4.5 – Ukázka databázových tabulek
- Obrázek 4.6 – Příklad SQL příkazu pro výpočet vegetačního indexu
- Obrázek 4.7 – Schéma procesu exportu rastru
- Obrázek 4.8 – Zdrojový kód pro export rastru
- Obrázek 4.9 – Základní zobrazení aplikace po spuštění
- Obrázek 4.10 a – První záložka postranního panelu pro rastrová data
- Obrázek 4.10 b – Druhá záložka postranního panelu pro sensorová data
- Obrázek 4.10 c – Třetí záložka postranního panelu pro korekční výpočet
- Obrázek 4.11 – Konfigurační soubory nezbytné pro chod aplikace
- Obrázek 4.12 – Ukázka zápisu ovládacích prvků v HTML
- Obrázek 4.13 – Ukázka ovládacích prvků aplikace
- Obrázek 4.14 – Ukázka PHP souboru rasterProperties.php pro metadata
- Obrázek 4.15 – Spuštění AJAXU a zpracování dat serveru
- Obrázek 5.1 – Proces dotazování na metadata rastrů
- Obrázek 5.2 – Skript PHP zajišťující procesing na straně serveru
- Obrázek 5.3 – Schéma procesu zobrazení rastru
- Obrázek 5.4 – Skript přebýrající ze serveru odpověď pomocí metody AJAX
- Obrázek 5.5 – Funkce myParser

Obrázek 5.6 – Funkce getBound pro lokalizaci rastru v mapě
Obrázek 5.7 – Ovládací panel pro výpočet vegetačních indexů
Obrázek 5.8 – Volání SQL skriptu pro výpočet vegetačního indexu
Obrázek 5.9a – Zpracování odpovědi na požadavek pro výpočet indexu AJAXEM
Obrázek 5.9b – Dialog potvrzující provedení výpočtu indexu
Obrázek 5.10 – Datový záznam pro typ senzoru enviro
Obrázek 5.11 – Funkce pro načtení senzoru v mapě
Obrázek 5.12 – Zobrazení senzoru v aplikaci
Obrázek 5.13 – Server skript pro výpočet mediánu
Obrázek 5.14 – Server skript pro výpočet aritmetického průměru
Obrázek 5.15 – Grafické rozhraní pro definici statistického výpočtu z hodnot senzoru
Obrázek 5.16 – Výsledek dotazu na medián a průměr sensorové veličiny
Obrázek 5.17 – Ovládací panel pro výpočet rozdílového rastru
Obrázek 6.1 – Finální vzhled mapové aplikace
Obrázek 6.2a – Načtení vypočítaného rozdílového rastru NDMI v QGISu bez korekce
Obrázek 6.2b – Načtení vypočítaného rozdílového rastru NDMI v QGISu s korekcí
Obrázek 6.3 – Reklasifikovaný rozdílový rastr do 4 kategorií zobrazený v QGISu

SEZNAM ILUSTRACÍ V PŘÍLOHÁCH

Obrázek 1 – Import
Obrázek 2 – Základní načtení snímku a volba pro ořezání (Subset) ve SNAPU
Obrázek 3 – Nastavení extentu pro ořez snímku
Obrázek 4 – Konverze bitové hloubky rastru
Obrázek 5 – Nastavení 8 bitové hloubky rastru
Obrázek 6 – Export ořezaného a převzorkovaného rastru
Obrázek 7 – Příkaz pro import rastru do PostGIS databáze
Obrázek 8 – Autorizace importu heslem
Obrázek 9a – Úspěšné dokončení importu v příkazové řádce
Obrázek 9b – Vytvořená naimportovaná tabulka zobrazená v PgAdmin 4
Obrázek 10 – Výchozí stav pro výpočet vegetačního indexu ve SNAPU
Obrázek 11 – Dvoupásmový rastr
Obrázek 12a – Záložka Band Math
Obrázek 12b – Detail záložky Band Math pro úpravu výpočetního výrazu
Obrázek 12c – Vypočítaný vegetační index s možností exportu
Obrázek 13a – Okno nápovědy
Obrázek 13b – Oknonápovědy – korekční výpočet
Obrázek 14 – Dotaz na metadata testovacího NIR pásma
Obrázek 15a – Kontrolní zobrazení testovacího NIR pásma
Obrázek 15b – Kontrolní zobrazení testovacího SWIR pásma
Obrázek 16 – Zadání parametrů pro výpočet testovacího NDMI indexu ze Sentinelu 2
Obrázek 17 – Výsledek vypočítaného vegetačního indexu NDMI bez korekce
Obrázek 18 – Zjištění hodnoty Objemové půdní vlhkosti pro 21. dubna 2019

- Obrázek 19 – Výpočet NDMI s použitím hodnoty objemové půdní vlhkosti
Obrázek 20 – Připojení k databázi s rastrovými daty v QGIS
Obrázek 21 – Výsledek vypočítaného vegetačního indexu NDMI s korekcí podle objemové půdní vlhkosti načtený v QGISu – Sentinel 21. 4. 2019
Obrázek 22 – Reklasifikovaný rastr do 4 kategorií- testovací NDMI ze Sentinelu 2 pro 21. 4. 2019

SEZNAM TABULEK

- Tabulka 4.1 – Parametry importu rastrových dat přes raster2pgsql.exe
Tabulka 5.1 – Postup parsování odpovědi serveru
Tabulka 6.1 – Srovnání výpočtu vegetačních indexů v PostGIS db a ve SNAP Desktop
Tabulka 6.2 – Převodní tabulka pro reklasifikaci rastru

SEZNAM TABULEK V PŘÍLOHÁCH

- Tabulka I – Přehled datových zdrojů pro rastry
Tabulka II – Přehled databázových tabulek A
Tabulka III – Přehled databázových tabulek B
Tabulka IV – Přehled databázových tabulek C
Tabulka V – Přehled konfiguračních souborů aplikace
Tabulka VI – Obsah dalších adresářů v rootu aplikace

PŘÍLOHY

SEZNAM PŘÍLOH

Vázané přílohy:

- Příloha 1 Přehled zdrojových dat a databázových tabulek
- Příloha 2 Souborová struktura aplikace
- Příloha 3 Návod pro předzpracování a import rastrových dat do PostGIS
- Příloha 4 Návod pro práci s aplikací (Nápověda) + praktický příklad

Volné přílohy

- Příloha 5 Poster
- Příloha 6 DVD

Popis struktury DVD

Adresáře:

aplikace_zdrojovy_kod

text_prace

vstupni_data

 Landsat 7

 Landsat 8

 Sentinel 2

vystupni_Data

 vegetacni_indexy

web

Přehled datových zdrojů

První příloha mapuje názvosloví použitých datových zdrojů a metadatový popis vytvořených databázových tabulek. Přehledy jsou uspořádány do několika tabulek. První tabulka v příloze číslo 1 obsahuje informace o všech použitých typech rastrových datových zdrojů. Jedná vždy o tři barevná pásma: červené (RED), blízké infračervené (NIR), střední infračervené (SWIR).

I. Přehled datových zdrojů pro rastry

Datový zdroj	Pásmo rastru	Kód rastru	Spektrální rozlišení [μm]	Prostorové rozlišení [m]	Datový formát
Landsat 7	Červené	B03	0.63–0.69	30	*.TIFF
	Blízké infračervené	B04	0.77–0.90	30	*.TIFF
	Střední infračervené	B05	1.55–1.75	30	*.TIFF
Landsat 8	Červené	B04	0.64–0.67	30	*.TIFF
	Blízké infračervené	B05	0.85–0.88	30	*.TIFF
	Střední infračervené	B06	1.57–1.65	30	*.TIFF
Sentinel 2-A	Červené	B04	0.665	20	*.JPEG2000
	Blízké infračervené	B08A	0.865	10	*.JPEG2000
	Střední infračervené	B11	1.610	20	*.JPEG2000

V dalších přehledových tabulkách II. až IV. jsou vypsané všechny databázové tabulky obsažené v databázích „db_roubalik“ a „sensor“. Tabulky jsou agregovány do skupin podle původu zdrojových dat. Záznamy v tabulkách II. a III. jsou vázány k databázi „db_roubalik“, záznamy v tabulce IV. referují na obsah databáze „sensor“. Kódu tabulky je přidán ve druhém sloupci sémantický význam a rozlišen je i původ vzniku tabulky. Jeden způsob je import pomocí nástroje „raster2pgsql.exe“ (způsob import), druhým je vytvoření tabulky pomocí jazyka SQL přímo v databázovém schématu (způsob výpočet).

II. Přehled databázových tabulek A

Skupina	Název tabulky	Sémantický význam	Vytvoření
Landsat 7	L7B03	Červené pásmo	import
	L7B04	Blízké infračervené pásmo	import
	L7B05	Krátké infračervené pásmo	import
	L7_ndvi	NDVI index	výpočet
	L7_ndmi	NDMI index	výpočet
	L7_savi	SAVI index	výpočet
	L7_ndvi_snap	NDVI index ze SNAPU	import
	L7_ndmi_snap	NDMI index ze SNAPU	import
	L7B_savi_snap	SAVI index ze SNAPU	import
Landsat 8	L8B04	Červené pásmo	import
	L8B05	Blízké infračervené pásmo	import
	L8B06	Krátké infračervené pásmo	import
	L8_ndvi	NDVI index	výpočet
	L8_ndmi	NDMI index	výpočet
	L8_savi	SAVI index	výpočet
	L8_ndvi_snap	NDVI index ze SNAPU	import
	L8_ndmi_snap	NDMI index ze SNAPU	import
	L8_savi_snap	SAVI index ze SNAPU	import
Sentinel 2-A	S2B04	Červené pásmo	import
	S2B08A	Blízké infračervené pásmo	import
	S2B11	Krátké infračervené pásmo	import
	S2_ndvi	NDVI index	výpočet
	S2_ndmi	NDMI index	výpočet
	S2_savi	SAVI index	výpočet
	S2_ndvi_snap	NDVI index ze SNAPU	import
	S2_ndmi_snap	NDMI index ze SNAPU	import
	S2_savi_snap	SAVI index ze SNAPU	import

III. Přehled databázových tabulek B

Název tabulky	Sémantický význam	Vytvoření
L8_ndmi_virrib	Rastr rozdílu hodnot NDMI získaných z Landstatu 8 a hodnoty objemové půdní vlhkosti	výpočet
L8_ndmi_rh	Rastr rozdílu hodnot NDMI získaných z Landstatu 8 a hodnoty relativní vzdušné vlhkosti	výpočet
S2_ndmi_virrib	Rastr rozdílu hodnot NDMI získaných ze Sentinelu 2 a hodnoty objemové půdní vlhkosti	výpočet
S2_ndmi_rh	Rastr rozdílu hodnot NDMI získaných ze Sentinelu 2 a hodnoty relativní vzdušné vlhkosti	výpočet
Reclass_L8_ndmi_virrib	Reklasifikovaný rastr rozdílu hodnot NDMI z Landsatu 8 a hodnot relativní půdní vlhkosti	výpočet
Reclass_L8_ndmi_rh	Reklasifikovaný rastr rozdílu hodnot NDMI z Landsatu 8 a hodnot relativní půdní vlhkosti	výpočet
Reclass_S2_ndmi_virrib	Reklasifikovaný rastr rozdílu hodnot NDMI z Sentinelu 2 a hodnot relativní půdní vlhkosti	výpočet
Reclass_S2_ndmi_rh	Reklasifikovaný rastr rozdílu hodnot NDMI z Sentinelu 2 a hodnot relativní půdní vlhkosti	výpočet

IV. Přehled databázových tabulek C

Název tabulky	Sémantický význam	Použití tabulky
sensor_loc	Tabulka souřadnic jednotlivých měřicích senzorů.	ANO
envirodata	Tabulka pro typ senzoru envirodata.	ANO
libelium	Tabulka pro typ senzoru libelium.	NE
sigfox	Tabulka pro typ senzoru sigfox.	NE
data	Jednotná tabulka pro všechny typy senzorů.	NE

Popis struktury konfiguračních souborů aplikace

Funkce aplikace byly popsány podrobněji v kapitolách praktické části 4–6. Příloha 2 pouze formou tabulky shrnuje názvy a stručnou charakteristiku jednotlivých konfiguračních souborů (tabulka V a tabulka VI).

V. Přehled konfiguračních souborů aplikace

Název souboru	Popis
index.html	Spouštěcí soubor domovské stránky aplikace
rasterProperties.php	Soubor na zpracování dotazu na metadata rastru
rasterExport.php	Soubor na zpracování příkazu pro export rastru
displayRaster.php	Soubor pro pracování dotazu na extent rastru, které je dále použité pro načtení rastru do mapového okna
vegetationIndicesNdvi.php	Soubor definující argumenty a postup výpočtu pro vegetační index NDVI
vegetationIndicesNdmi.php	Soubor definující argumenty a postup výpočtu pro vegetační index NDMI
vegetationIndicesSavi.php	Soubor definující argumenty a postup výpočtu pro vegetační index SAVI
displayIndex.php	Soubor pro pracování dotazu na extent rastru vegetačního indexu, které je dále použité pro načtení vegetačního indexu do mapového okna
sensorLocation.php	Zpracování souřadnic rastru a jeho lokalizace
sensorData.php	Soubor na vytváření selektované atributové tabulky pro sensorová data
sensorDataStatistics.php	Soubor s definicí statistických výpočtů pro objemovou vlhkost půdy a relativní vlhkost vzduchu
sensorDataCorrection.php	Definiční soubor pro výpočet indexu NDMI doplněného o sensorové hodnoty objemové vlhkosti půdy nebo relativní vlhkosti vzduchu
sensorDataReclass.php	Soubor pro reklasifikaci přepočítaného korekčního rastru

VI. Obsah dalších adresářů v rootu aplikace

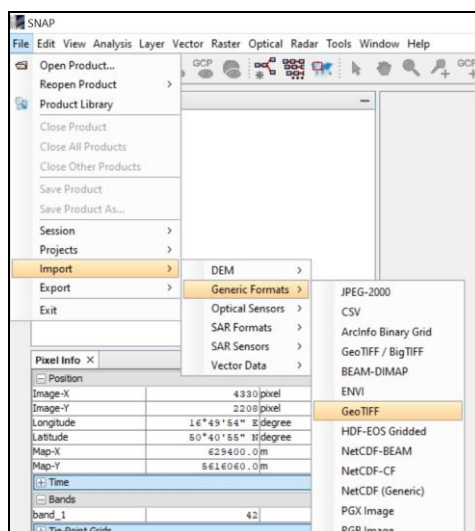
Název adresáře	Popis
css	Veškeré kaskádové styly použité pro design
js	Veškeré použité Javascript kódy třetích stran
zoomhome	Soubory definující Leaflet Zoomhome pluginu
data	Adresář pro exportované rastry z databáze

Manuál pro předzpracování a import družicových snímků

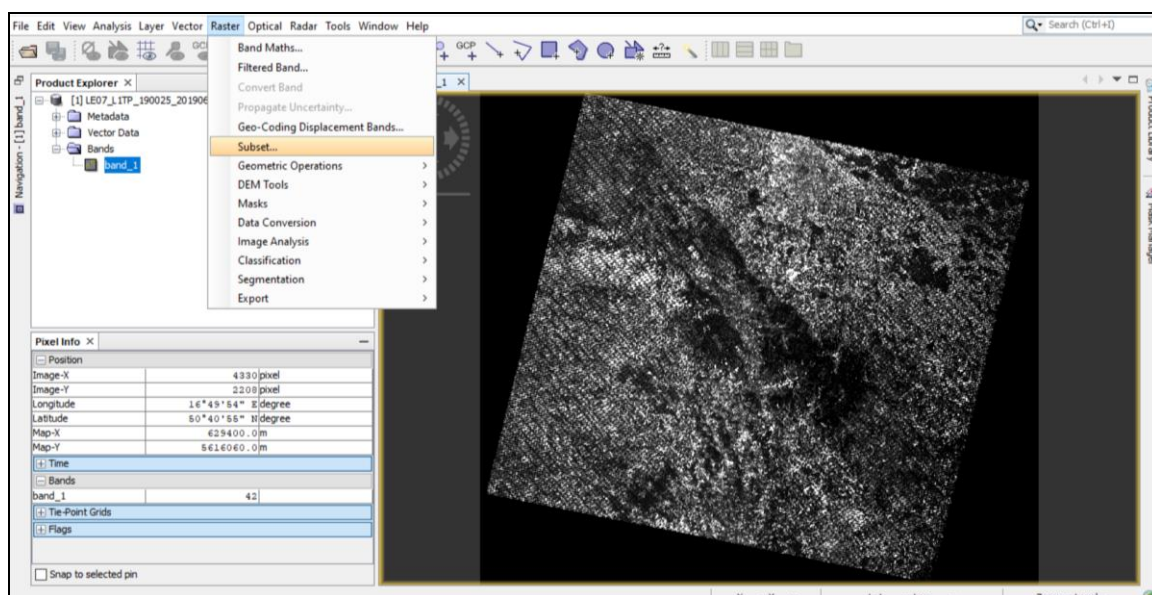
Tato příloha slouží jako ukázka zpracování stažených družicových dat v několika fázích. První fáze popisuje proces ořezání a opětovné uložení rastrových snímků v programu SNAP Desktop předtím než jsou v druhém kroku importovány do PostGIS databáze.

Předzpracování rastrových dat ve SNAP Desktop

Prvním krokem po otevření programu SNAP (odkaz ke stažení programu: <http://step.esa.int/main/download/snap-download/>) je import družicových dat stažených z různých zdrojů (<https://earthexplorer.usgs.gov/> nebo <https://www.sentinel-hub.com/>). Přes panel nástrojů a nabídku *File->Import->Generic Formats->GeoTIFF/JPEG-2000* (obrázek 1) je příslušný snímek načten do prohlížečeho okna (obrázek 2).

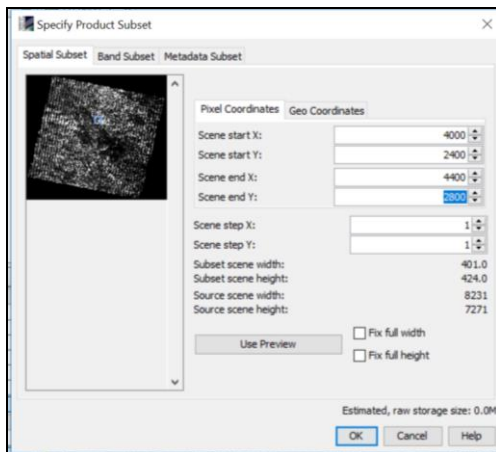


Obrázek 1. Import

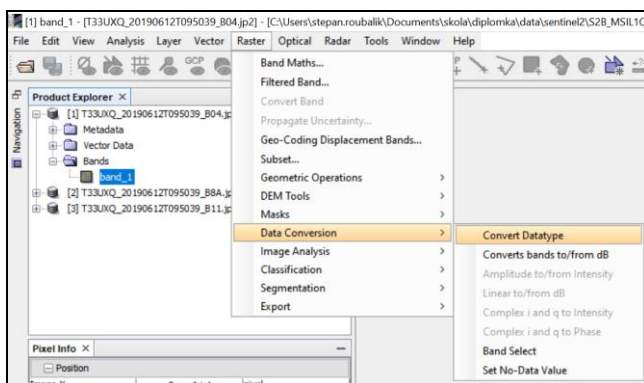


Obrázek 2. Základní načtení snímku a volba pro ořezání (Subset)

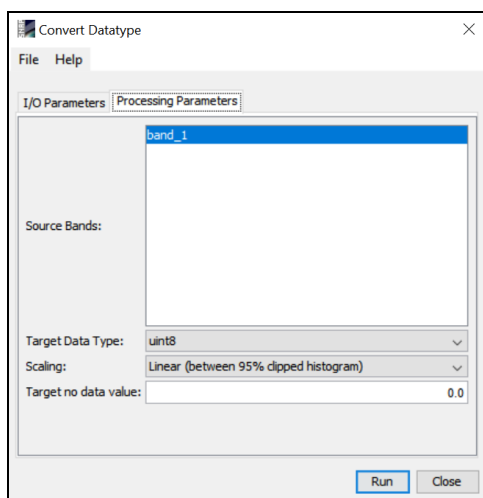
Ořezání rastru je provedeno pomocí nabídky v panelu nástrojů *Raster->Subset*. Po rozkliknutí proběhne otevření dialogového okna ve kterém má uživatel možnost nastavit extent ořezové masky rastru v pixelech (obrázek 3). U dat Landsatu 8 a Sentinelu 2 lze provést převzorkování bitové hloubky rastru z 16 bitového typu na 8 bitový (*uint8*) na obrázku 5. Převzorkování je spouštěno přes panel nástrojů *Raster->Convert Datatype* (obrázek 4).



Obrázek 3. Nastavení extentu pro ořez snímku

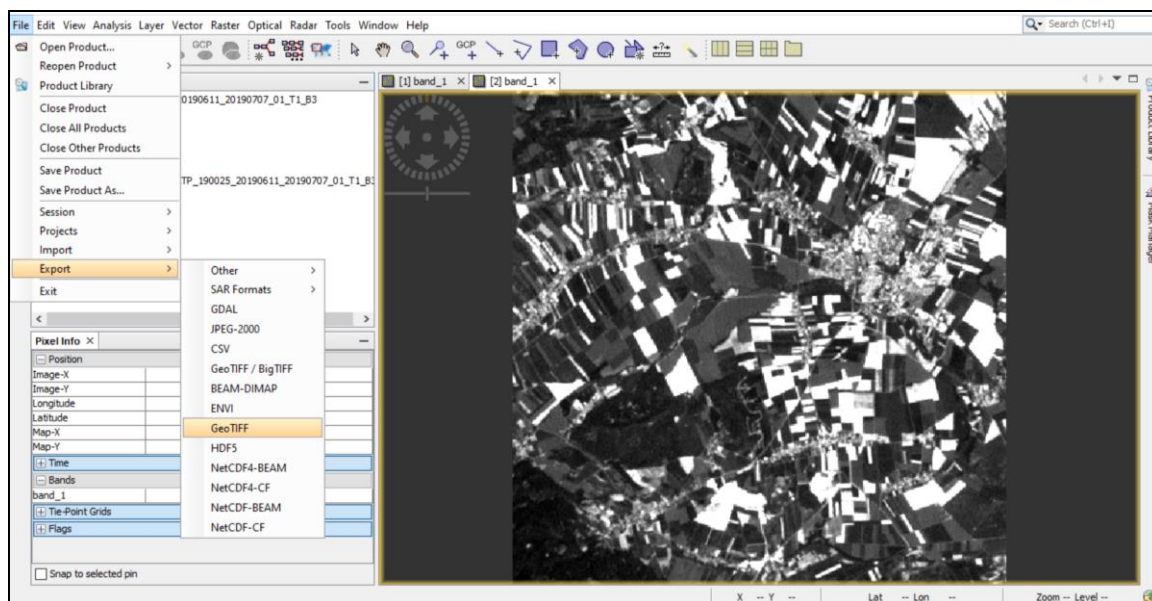


Obrázek 4. Konverze bitové hloubky rastru



Obrázek 5. Nastavení 8 bitové barevné hloubky rastru

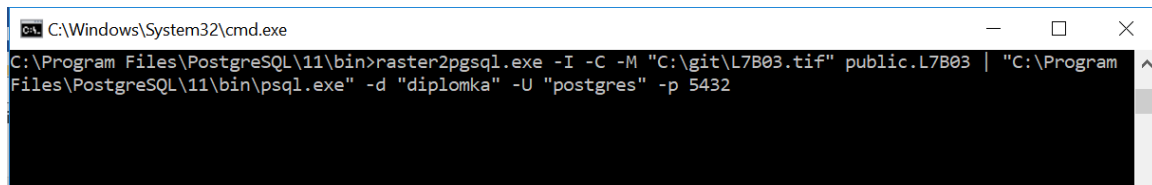
Potom co jsou nastaveny výše zmíněné parametry (ořez, bitové rozlišení, prostorové rozlišení) lze snímek exportovat v několika libovolných formátech. Pro další práci je doporučeno použít pro export formát GeoTIFF (obrázek 6).



Obrázek 6. Export ořezaného a převzorkovaného rastru

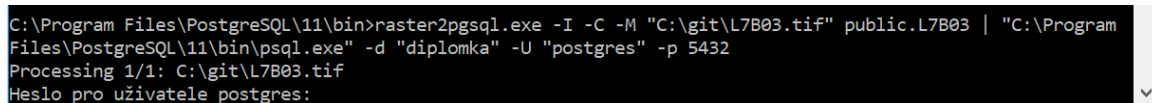
Import snímků do databáze PostGIS

Pro zahájení importu je potřeba otevřít příkazovou řádku v následujícím adresáři: „C:\Program Files\PostgreSQL\x\bin“. Písmeno x v tomto případě zastupuje číslo verze PostgreSQL (9, 10, 11...). Jakmile je uživatel ve správném adresáři může být zadán příkaz, který zahájí import rastru (obrázek 7).



Obrázek 7. Příkaz pro import rastru do PostGIS databáze

První část řetězce pro import „*raster2pgsql.exe*“ spouští importovací nástroj. Další tři argumenty „-I -C -M“ jsou povinné pro úspěšný import zadat (vyplnění hodnot proběhne automaticky). Následuje uvedení cesty k příslušnému úložišti na lokálním pc „C:\git\L7B03.tif“. Tato cesta je libovolná pro každého uživatele. Za svislicí „|“ je uvedena cesta k nástroji psql.exe který je umístěn na straně databázového serveru. Poslední povinné parametry jsou: (název databáze d ““ uživatel U ““ a port p ““ na kterém databázový server běží). Po kliknutí je uživatel vyzván k zadání přístupového hesla k databázi (obrázek 8).



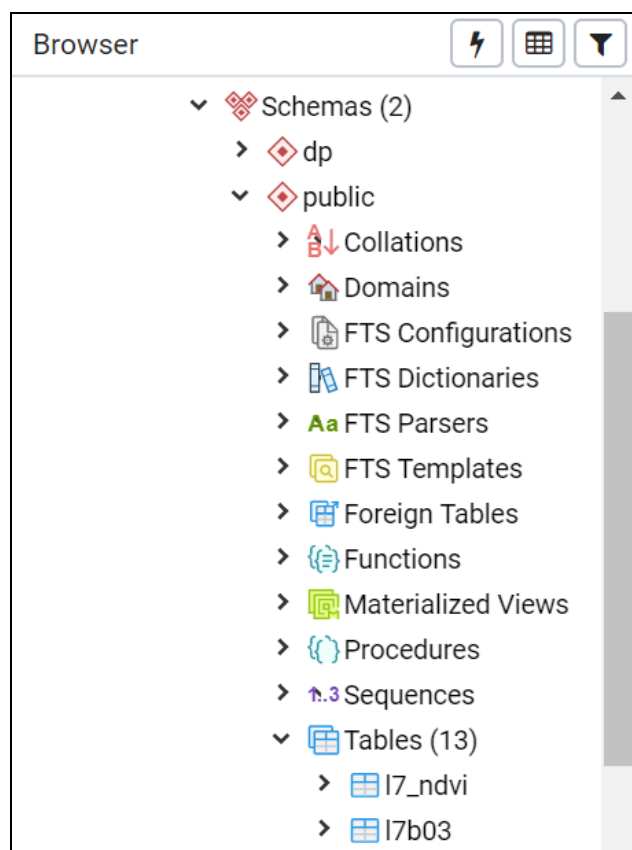
Obrázek 8. Autorizace importu heslem

Po zadání je spuštěn samotný proces importu. Že byl import úspěšný lze ověřit v rozhraní databáze PgAdmin 4 kde v příslušné databázi a schématu public je mezi

„tables“ tabulka názvem odpovídající importovanému souboru, v tomto případě *l7b03* (obrázek 9a, 9b). Jakmile je import dokončen, pracuje uživatel dále v prostředí webové aplikace.

```
BEGIN
CREATE TABLE
INSERT 0 1
CREATE INDEX
ANALYZE
NOTICE: Adding SRID constraint
NOTICE: Adding scale-X constraint
NOTICE: Adding scale-Y constraint
NOTICE: Adding blocksize-X constraint
NOTICE: Adding blocksize-Y constraint
NOTICE: Adding alignment constraint
NOTICE: Adding number of bands constraint
NOTICE: Adding pixel type constraint
NOTICE: Adding nodata value constraint
NOTICE: Adding out-of-database constraint
NOTICE: Adding maximum extent constraint
addrasterconstraints
-----
t
(1 řádka)
COMMIT
VACUUM
```

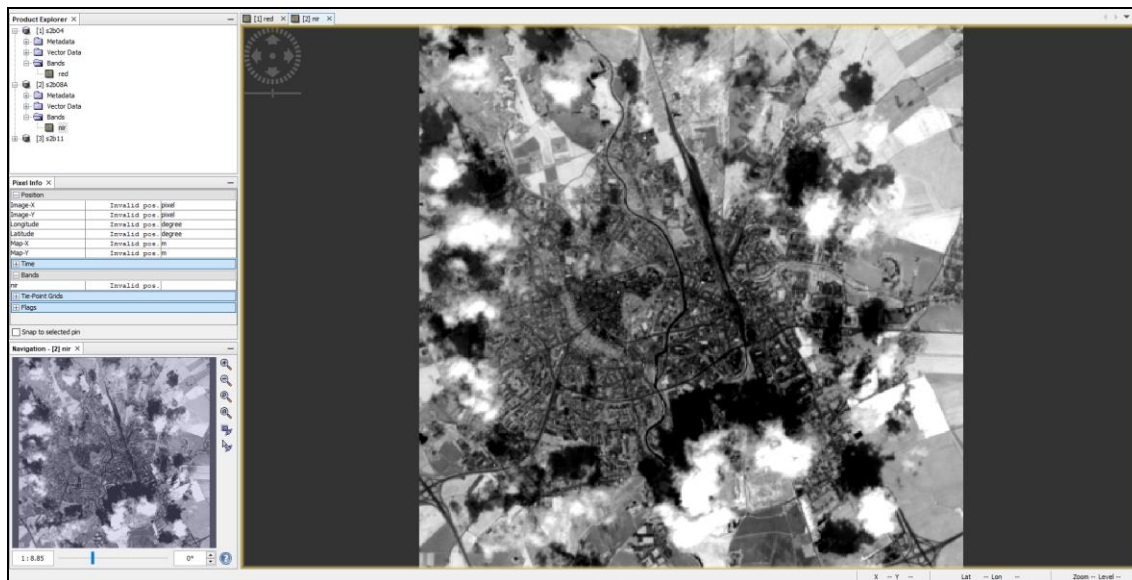
Obrázek 9a. Úspěšné dokončení importu v příkazové řádce



Obrázek 9b. Vytvořená naimportovaná tabulka zobrazená v PgAdmin 4

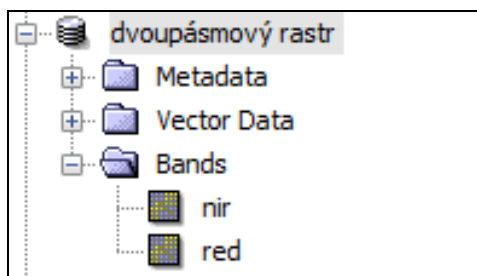
Kontrolní výpočet vegetačních indexů ve SNAPU

Výpočty vegetačních indexů jsou součástí funkcionality webové aplikace. V rámci testování a ověření přesnosti výpočtu indexu v *PostGIS* byl proveden výpočet i v prostředí programu *SNAP Desktop*. Takto vypočítaný index byl poté porovnán. Součástí přílohy je výpočet vegetačního indexu ve SNAPU. Ve výchozím stavu jsou podobně jako na obrázku 10 načteny dva rastry s jedním barevným pásmem. V tomto příkladu pásma 4 a 8A Sentinelu 2.

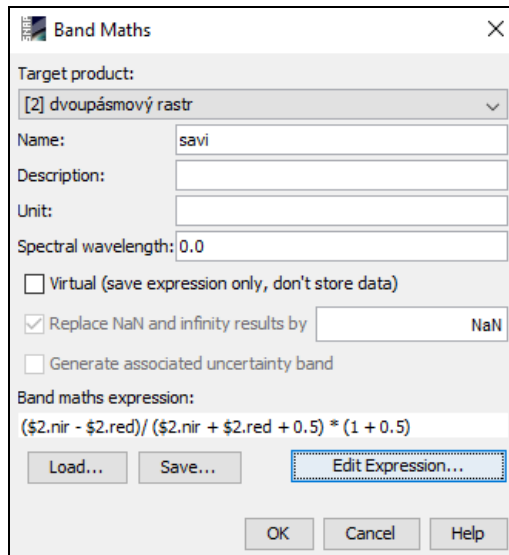


Obrázek 10. Výchozí stav pro výpočet vegetačního indexu ve SNAPU

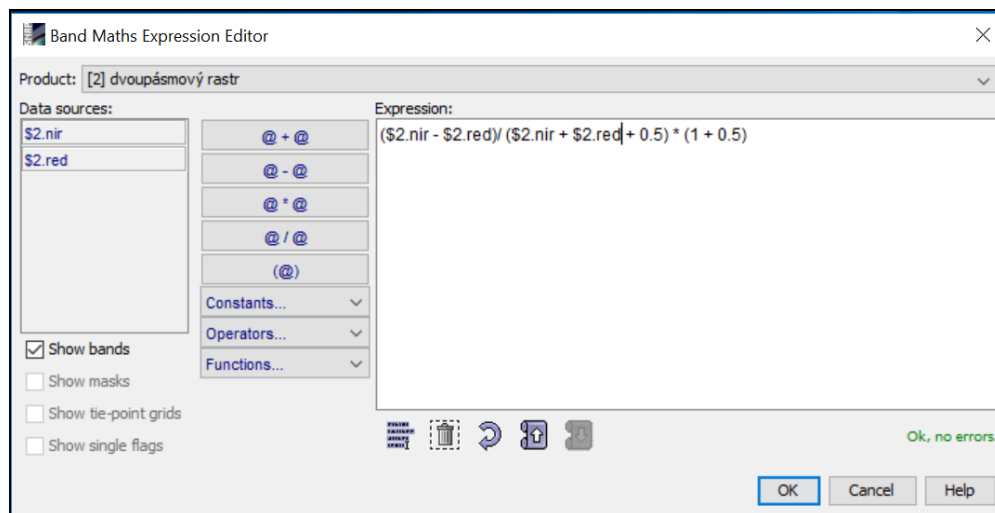
Přes nabídku Raster->Band Math lze každému rastru přidat libovolný počet pásem. Aby mohl být v dalším kroku vypočítán vegetační index, je nejprve vytvořen rastr který obsahuje obě pásma zároveň (obrázek 11). Nyní už lze opět přes tlačítko Raster -> Band Math zadefinovat rovnici podle které bude vypočítáno třetí pásmo, jehož hodnoty budou odpovídat příslušnému vegetačnímu indexu. Do rovnice je možno dosadit všechna dostupná pásma daného rastru (obrázek 12b). V základním nastavení (obrázek 12a) je doporučeno nově vytvářené pásmo pojmenovat stejně jaký je název vegetačního indexu, doplněna může být hodnota vlnové délky (pro index nepotřebná informace). Možnost vytvořit virtuální rastr musí být nezatržená. Kombinace pásem může být zapsána přímo v tomto okně nebo lze nahrát již vytvořený výraz pomocí tlačítka „Load“. Každý výraz potom lze nadále upravovat poklikem na tlačítko „Edit Expression“. Po nastavení všech změn a odkliknutí je vytvořeno další pásmo s hodnotami vegetačního indexu. Posledním většinou prováděným krokem je export pásma s širokým výběrem výstupních formátů (obrázek 12c).



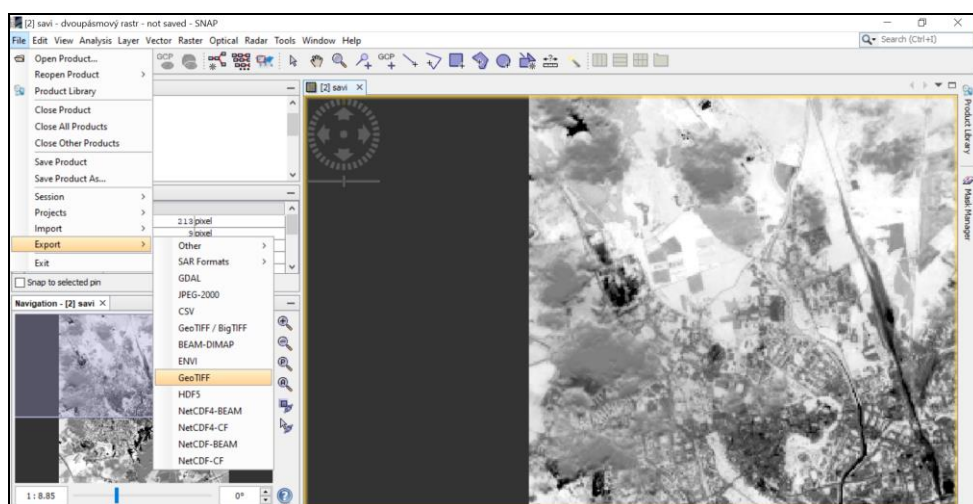
Obrázek 11. Dvoupásmový rastr



Obrázek 12a. Záložka Band Math



Obrázek 12b. Detail záložky Band Math pro úpravu výpočetního výrazu



Obrázek 12c. Vypočítaný vegetační index s možností exportu

Návod pro ovládání webové aplikace

Pomocná nápověda je dostupná přímo v aplikaci na záložce „Nápověda“. Popsány jsou tři hlavní procesy: výpočet vegetačního indexu bez korekce, výpočet průměrných hodnot ze sensorových dat, výpočet vegetačního indexu s použitím korekčních dat (obrázek 13a, 13b).

Nápověda

Výpočet vegetačního indexu

RED PÁSMO
 LANDSAT 7 ▼ Zadání prvního rastrového pásma pro výpočet

NIR PÁSMO
 LANDSAT 7 ▼ Zadání druhého rastrového pásma pro výpočet

VYPOČITANÝ VEGETAČNÍ INDEX
 NDVI L7 ▼ Zadání názvu vegetačního indexu pro typ zdroje Landsat 7, Landsat 8, Sentinel 2 a spuštění výpočtu tlačítkem "VÝPOČET..."
 Ve všech polích musí být zadán stejný název družicových dat. Jinak výpočet neproběhne správně.

VÝPOČET NDVI IDX

Zjištění průměrné hodnoty ze senzoru

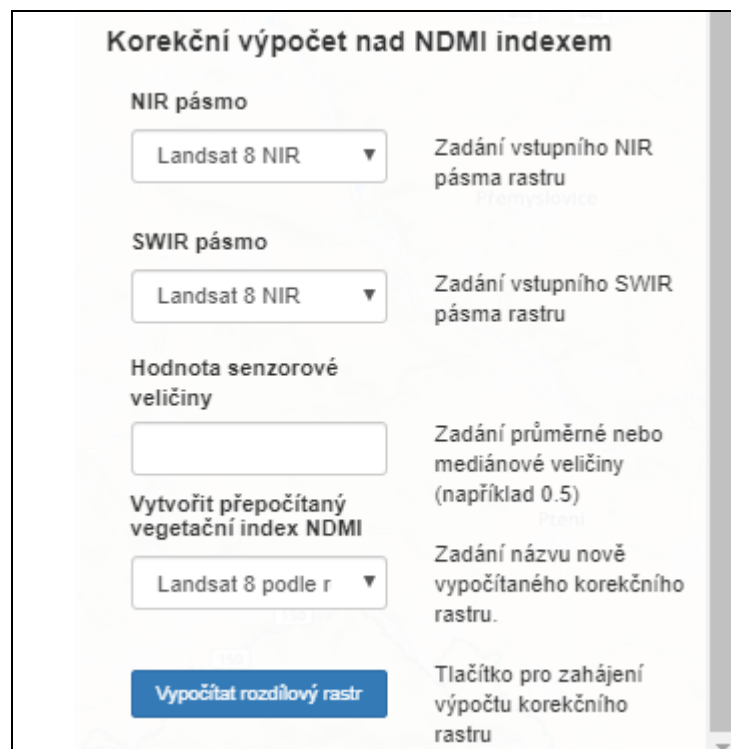
Typ senzoru
 sensor envirodata ▼ Zadání typu senzoru

Veličina
 objemová půdní v ▼ Zadání fyzikální veličiny

Denní časový extent
 2019-04-19 ▼ Zadání dne pro který je hodnota počítána

STATISTIKA
 Tlačítko pro zahájení výpočtu.

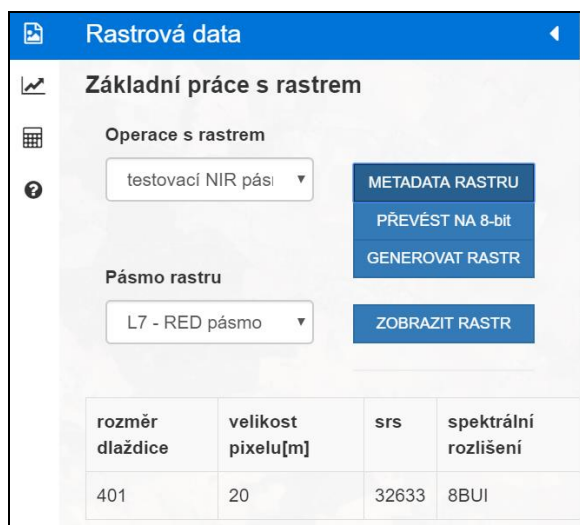
Obrázek 13a. Okno nápovědy



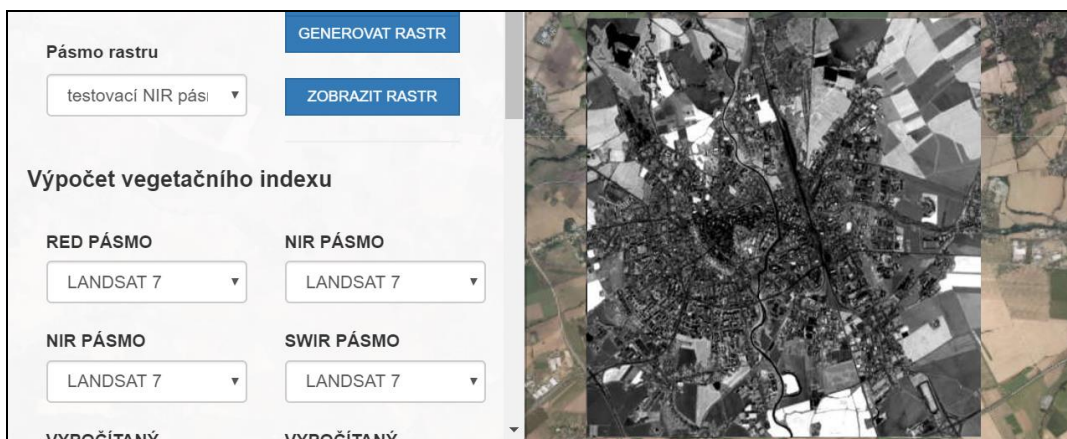
Obrázek 13b Okno nápovědy – korekční výpočet

Praktický příklad práce v aplikaci

Uživatel má podle přílohy číslo čtyři možnost vyzkoušet výpočet vegetačního indexu NDMI na připravených testovacích datech Sentinelu 2. Ty jsou již předzpracovány ve SNAP Desktop a byl proveden jejich import do databáze. První část preprocesingu do fáze nahrání rastru do databáze byla udělána autorem. Zbývá část přílohy dokumentuje práci s rastrovými daty v aplikaci, výpočet vegetačního indexu NDMI včetně korekčního výpočtu, reklasifikaci vypočítaného rastru a jeho závěrečnou vizualizaci v programu QGIS. Vstupními daty jsou pásma Sentinelu 2 s názvem „testovací NIR pásmo“ a „testovací SWIR pásmo“.



Obrázek 14. Dotaz na metadata testovacího NIR pásma



Obrázek 15a. Kontrolní zobrazení testovacího NIR pásma



Obrázek 15b. Kontrolní zobrazení testovacího SWIR pásma

NIR PÁSMO

Testovací data Se ▼

SWIR PÁSMO

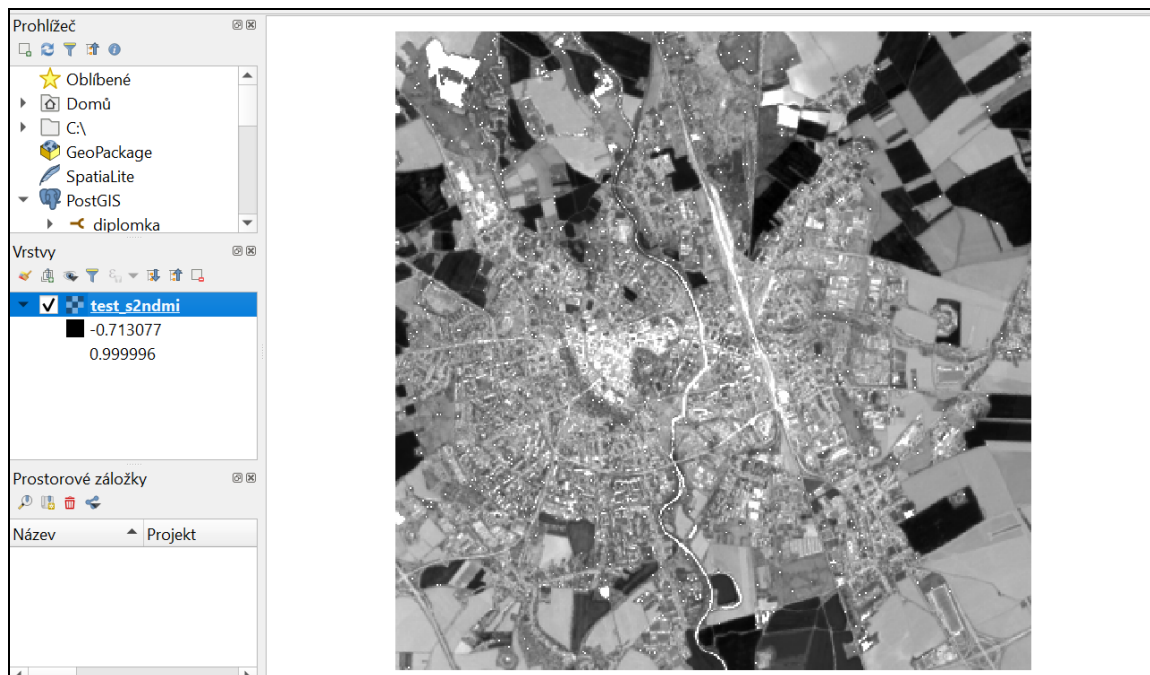
Testovací data Se ▼

VYPOČÍTANÝ VEGETAČNÍ INDEX

testovací NDMI Se ▼

VÝPOČET NDMI IDX

Obrázek 16. Zadání parametrů pro výpočet testovacího NDMI indexu ze Sentinelu 2



Obrázek 17. Výsledek vypočítaného vegetačního indexu NDMI bez korekce

📄
Senzorová data
◀

Lokalizace Cholíná

lokalita Olomouc ▼

ZOBRAZIT SENZOR V MAPĚ

Typ senzoru

senzor envirodate ▼

Veličina

objemová půdní v ▼

Měsíční časový extent

2019 srpen ▼

Denní časový extent

2019-04-21 ▼

ATRIBUTOVÁ TABULKA

STATISTIKA

medián

23.420

průměr

23,420

Obrázek 18. Zjištění hodnoty objemové půdní vlhkosti 21. dubna 2019

Výpočet NDMI s korekcí VIRRIB a RH

Výpočet indexu NDMI s korekcí

NIR pásmo
testovací NIR pásmo

SWIR pásmo
testovací SWIR pásmo

Hodnota senzorové veličiny
23.42

Vytvořit přepočítaný vegetační index NDMI
testovací Sentinel 2 podle objemové půdní vlhko:

VYTVOŘIT PŘEPOČÍTANÝ INDEX

Vstupní rastr pro reklasifikaci
testovací Sentinel 2 podle objemové půdní vlhko:

Výstupní rastr pro reklasifikaci
Reklasifikace testovací Sentinel 2 podle podle (vi)

REKLASIFIKOVAT PŘEPOČÍTANÝ INDEX

Obrázek 19. Výpočet NDMI s použitím hodnoty objemové půdní vlhkosti

Vytvořit nové připojení PostGIS

Spojení k kgi bylo úspěšné

Informace o připojení

Název: kgi
Služba:
Hostitel: 158.194.94.120
Port: 5432
Databáze: db_roubalik
SSL režim: vypnout

Ověření

Nastavení Základní

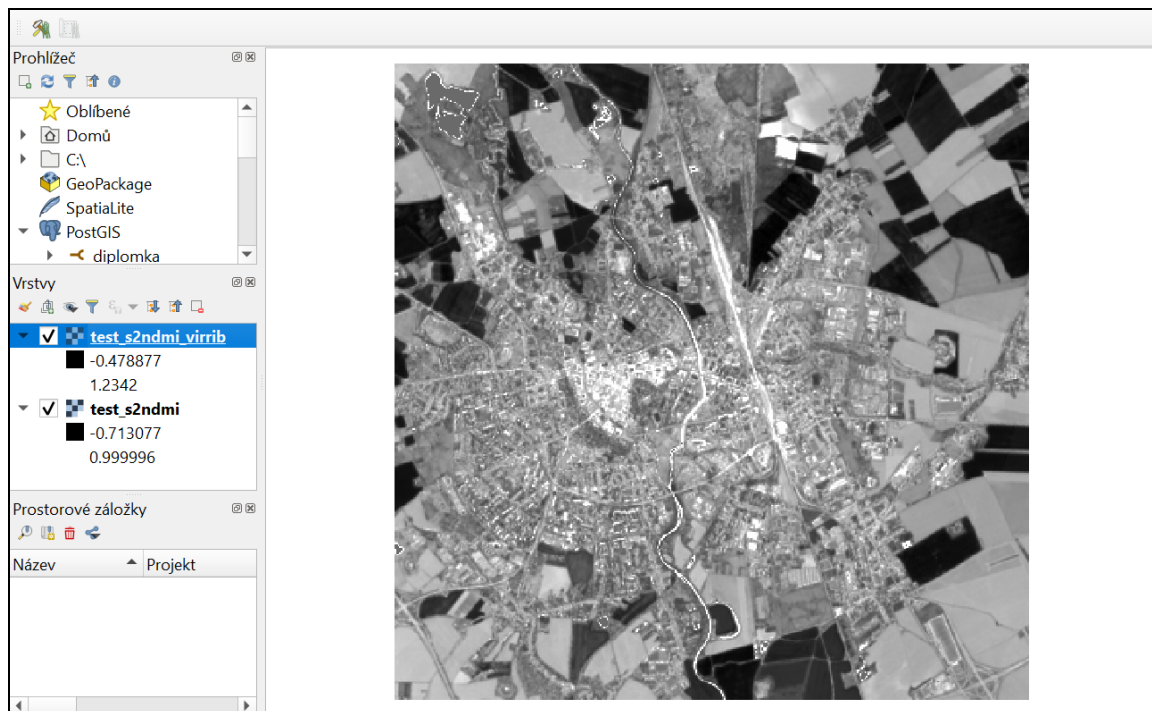
Uživatelské jméno: roubalik Uložit
Heslo: dp_roubalik2019 Uložit

Varování: pověření uložena jako prostý text v soubor projektu.

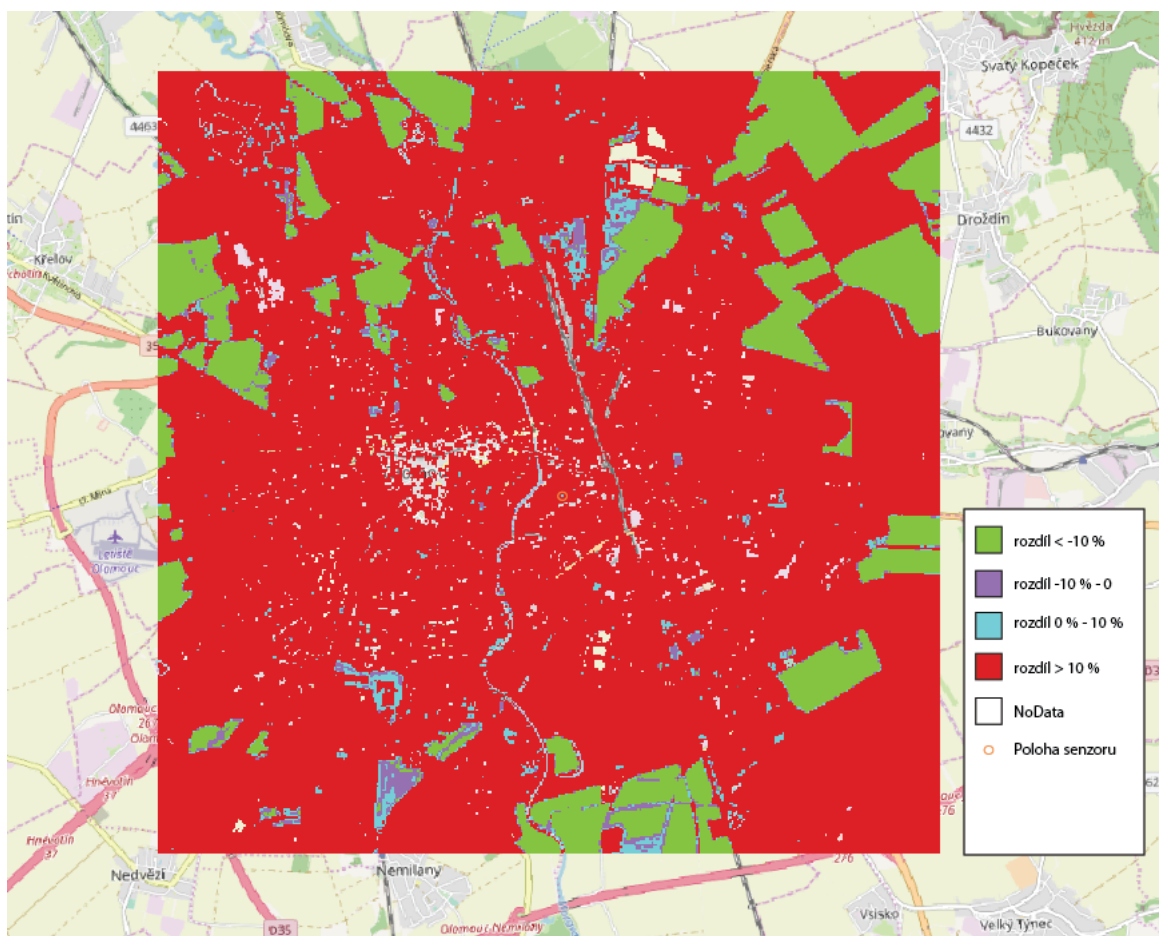
Převést na nastavení

Vyzkoušet spojení

Obrázek 20. Připojení k databázi s rastrovými daty v QGIS



Obrázek 21. Výsledek vypočítaného vegetačního indexu NDMI s korekcí podle objemové půdní vlhkosti načtený v QGISu – Sentinel 21. 4. 2019



Obrázek 22. Reklasifikovaný rastr do 4 kategorií- testovací NDMI ze Sentinelu 2 pro 21. 4. 2019