<table>
<tr><td colspan="6">ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
KATEDRA GEOMATIKY</td></tr>
<tr><td colspan="6">Název předmětu:
Geoinformatika</td></tr>
<tr><td>Úloha:
U2</td><td colspan="5">Název úlohy:
Automatizované rozpoznávání objektů v mapách</td></tr>
<tr><td>Akademický rok:
2024/2025</td><td>Semestr:
zimní</td><td>Studijní skupina:
102C</td><td>Vypracoval:
Michal Kovář
Filip Roučka
Magdaléna Soukupová</td><td>Datum:
4. 11. 2024</td><td>Klasifikace:</td></tr>
</table>

1 Zadání:

Část A

Zadáním první části bylo implementovat algoritmus pro vyhledávání vzorů v Müllerově mapě Čech na základě obrazové korelace. Má být zvolen prvek typu obec s kostelem jako okno pro vyhledávání a na základě vhodné hodnoty korelace mají být vyhledány všechny pozice obcí s kostelem na mapovém listu. Výsledek má být odevzdán jako pixelové souřadnice kostelů.

Část B

Zadáním druhé části bylo implementovat algoritmus obrazové segmentace pomocí metody k-Means nad obrazem Topografické mapy ČSSR v měřítku 1:25 000. Výsledkem má být extrahovaná plocha lesů očištěná o vrstevnice a další překryvnou kresbu, naopak mají být z lesů odečteny lesní průseky. Otvory v lesním prostoru větší než 50 pixelů mají být zachovány. Výsledek má být odevzdán jako pole pixelových souřadnic.

2 Bonus:

Část A

- *Proředění souřadnic o vícenásobná opakování.*
- *Skript přijímající jako uživatelský vstup pět vzorů obcí s kostelem, které jsou průměrovány.*

Část B

- *Segmentovaný obraz mapy pomocí metody Graph Cut.*
- *Plocha lesa uložená jako vrstva v GeoPackage v systému UTM.*

3 Popis problému

Tento úkol se zaměřuje na dvě hlavní oblasti: vyhledávání specifických objektů na mapě a segmentaci různých oblastí na mapě.

Část A

Cílem první části je implementovat algoritmus pro vyhledávání vzorů v Müllerově mapě Čech na základě obrazové korelace. Obrazová korelace je metoda, která umožňuje nalezení specifických vzorů v obraze na základě podobnosti mezi vzorem a částí obrazu. Tento úkol zahrnuje následující kroky:

- **Výběr vzoru**: Identifikace a výběr specifického vzoru, který bude použit jako referenční obrazec pro vyhledávání v obraze. V případě že je vybráno více vzorů je vzor pro vyhledávání určen jako průměr všech vybraných vzorů.
- **Výběr vhodného barevného kanálu**: Po výběru vzoru je vhodné převést obraz i vzor do barevného prostoru, kde jsou zájmové objekty dobře rozpoznatelné. Například převod do barevného prostoru YCbCr a použití kanálu Y může zvýraznit jasové informace.
- **Úprava obrazu pomocí pohyblivého fitru (kernelu)**: Obraz je možné dále upravit například pomocí kernelu, například aplikací Gaussova filtru nebo jiných filtrů pro zvýraznění specifických rysů hledaného objektu.
- **Výpočet korelačního koeficientu mezi vzorem a částí obrazu**: Použití matematických metod pro výpočet míry podobnosti mezi vybraným vzorem a různými částmi obrazu.
- **Určení limitní hodnoty korelace pro detekci specifických objektů**: Stanovení minimální hodnoty korelačního koeficientu, která bude považována za dostatečnou pro identifikaci specifického objektu. Tato limitní hodnota se určí experimentálně.

- **Vyhledání všech pozic specifických objektů v obraze**: Prohledání celého obrazu a identifikace všech míst, kde korelační koeficient překračuje stanovenou prahovou hodnotu. Pokud je nalezen jeden objekt vícenásobně může být za nález považován pouze jeden nález tohoto objektu a to ten, který má největší korelaci.
- **Zaznamenání výsledku jako pixelové souřadnice objektů**: Výpis a vykreslení pixelových souřadnic všech nalezených objektů v obraze.

Část B

Cílem je z Topografické mapy ČSSR v měřítku 1 : 25 000 segmentovat plochy lesa pomocí metody $k - Means$. Původní mapa je z roku 1954 a je na ní vidět degradace barev. Mapa byla následně naskenována a komprimována do ztrátového formátu `*.jpg`. Jelikož se jedná o topografickou mapu, tak plochy lesa obsahují i popisy, vrstevnice cesty a další prvky které segmentaci dělají náročnější.
Při zpracování, bylo odhaleno, že zadaná mapa, má jednu část mapy zelenější, což vytváří barevný předěl. Tento úkol zahrnuje následující kroky:

- **Editace obrazových dat:** změna, kontrastu, jasu a barvených složek pro zvýraznění jednotlivých různých ploch a minimalizace vlivu zeleného předělu.
- **Převedení barevných systémů:** systém RGB převeden do systému $CIELab$
- **Segmentace:** obrazová segmentace pomocí vestavěné funkce $k - Means$.
- **Filtrace obrazu:** nově vytvořená vrstva lesa byla filtrována pomocí vyhlazovacího filtru o tvaru disku velikosti $[5x5]$.
- **Mapová algebra:** pomocí mapové algebry byla vypočítána nová vrstva z jednotlivých segmentovaných prvků.
- **Filtrace obrazu:** nově vytvořená vrstva lesa byla filtrována pomocí vyhlazovacího filtru o tvaru disku velikosti $[5x5]$.
- **Export:** export plochy segmentovaného lesa.

4 Postup

Část A

- **Výběr vzoru**: Výběr konkrétního vzorku, který bude použit jako referenční obrazec pro vyhledávání v obraze. Tento vzor může být vybrán interaktivně uživatelem nebo může být předdefinován. K výběru vzorků slouží funkce `select_samples.m`. Pro přepnutí mezi interaktivním výběrem vzorku a využitím předdefinovaného vzorku slouží část kódu:

```
interactive_selection = 'YES';
```

- **Průměrný vzorek**: Pokud je vybráno více vzorků, bude k vyhledávání použit vzorek získaný jako průměr všech vybraných vzorků.V Matlabu je průměrný vzorek vypočten jako:

```
template = zeros(template_size, 'double');
for i = 1:num_samples
    template = template + double(templates{i});
end
template = uint8(template / num_samples);
```

Matematicky lze průměrný vzorek T_{avg} vypočítat jako:

$$T_{\text{avg}}(x, y) = \frac{1}{N} \sum_{k=1}^{N} T_k(x, y)$$

kde $T_k(x, y)$ je hodnota pixelu na pozici (x, y) v k-tém vzorku a N je celkový počet vzorků. Tento průměrný vzorek pak slouží jako referenční obrazec pro vyhledávání v obraze.

- **Výběr vhodného barevného kanálu**: Po výběru vzoru pro vyhledávání v obraze je vhodné převést obraz i vzorek do barevného prostoru, kde jsou zájmové objekty dobře rozpoznatelné. Například převod do barevného prostoru YCbCr a použití kanálu Y může zvýraznit jasové informace. Matematicky lze převod z RGB do YCbCr vyjádřit jako:

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

  kde Y představuje jasovou složku, C_b a C_r jsou chrominační složky.[1].

- **Úprava obrazu pomocí pohyblivého filtru (kernelu)**: Obraz je možné dále upravit například pomocí kernelu. Pohyblivé filtry lze rozdělit na vysokofrekvenční a nízkofrekvenční. Vysokofrekvenční filtry zesilují rozdíly mezi hodnotami v obraze a nízkofrekvenční filtry potlačují rozdíly mezi hodnotami v obraze, což vede k vyhlazení obrazu.[2] Gaussův filtr je nízkofrekvenční filtr, pro jehož aplikaci lze využít vestavěnou funkci v MATLABu `imgaussfilt`.[3] Tato funkce přijímá povinný parametr `input_image`, který představuje rastr, a volitelný parametr, který značí směrodatnou odchylku. Použití funkce v MATLABu je následující:

```
output_image = imgaussfilt(input_image, 2);
```

  Matematicky lze Gaussův filtr popsat jako metodu, při které se pro jednotlivé pixely pokryté kernelem vypočtou váhové koeficienty podle Gaussovy funkce:

$$G(i,j) = \exp\left(-\frac{(i-u)^2 + (j-v)^2}{2\sigma^2}\right)$$

  kde i, j jsou libovolné pixely pod kernelem, u, v jsou střední pixely a σ je směrodatná odchylka, která se zadává jako parametr. Váhy $W(i,j)$ se určí normalizací hodnot $G(i,j)$ tak, aby jejich součet byl roven 1:

$$W(i,j) = \frac{G(i,j)}{\sum_{i=-k}^{k} \sum_{j=-k}^{j} G(i,j)}$$

  kde k je poloměr kernelu. Výsledná hodnota (stupeň šedi) pixelu se vypočítá jako vážený průměr hodnot pixelů v okně:

$$P(u,v) = \sum_{i=-k}^{k} \sum_{j=-k}^{j} W(i,j) \cdot V(i,j)$$

  kde $V(i,j)$ je původní digitální hodnota v pixelu i, j [4].

  Ukázka kernelu pro Gaussův filtr s $\sigma = 1$ a velikostí 3x3:

0.0751	0.1238	0.0751
0.1238	0.2042	0.1238
0.0751	0.1238	0.0751

- **Výpočet korelačního koeficientu mezi vzorem a částí obrazu**: Pro výpočet korelace byla využita vestavěná funkce MATLABu `normxcorr2`, která se používá následujícím způsobem:

$$c = \texttt{normxcorr2(template_sel, im_sel)}$$

  kde `template_sel` je vybraný vzor a `im_sel` je část obrazu. Korelační koeficient se vypočítá jako:

$$r = \frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \bar{A})^2 \sum_{i=1}^{n}(B_i - \bar{B})^2}}$$

  kde A_i a B_i jsou hodnoty pixelů ve vzoru a v části obrazu, a $\bar{A}$ a $\bar{B}$ jsou průměrné hodnoty těchto pixelů.

- **Vyhledání všech pozic hledaných objektů v obraze**: Prohledání celého obrazu a identifikace všech míst, kde korelační koeficient překračuje stanovenou prahovou hodnotu.
- **Odstranění všech vícenásobných opakování**: Odstranění všech nalezených korelací ve zvoleném rádiusu od místa s největší korelací. k tomu slouží funkce `find_unique_values.m` do níž vstupuje matice hodnot korelací a jejich pozic seřazená vzestupně podle hodnot korelace.

```
positions = sortrows([vals, rows, cols], -1);
unique_positions = find_unique_positions(positions, radius);
```

  Matematicky lze algoritmus pro hledání unikátních pozic popsat takto:

  Vstupem je matice $\mathbf{P}$, kde každý řádek obsahuje hodnotu korelace a souřadnice (x, y). Matice $\mathbf{P}$ je seřazena sestupně podle hodnot korelace v prvním sloupci. Rádius r je minimální vzdálenost, kterou musí mít dvě unikátní pozice od sebe, aby byly považovány za odlišné.

  - **Inicializace**: První řádek matice $\mathbf{P}$ je přidán do matice unikátních pozic $\mathbf{U}$.
  - **Iterace**: Pro každou další pozici $\mathbf{p}_i$ v matici $\mathbf{P}$:
    * Výpočet vzdálenosti $d(\mathbf{p}_i, \mathbf{u}_j)$ pro všechny $\mathbf{u}_j$ v matici $\mathbf{U}$.
    * Pokud $\min_j d(\mathbf{p}_i, \mathbf{u}_j) > r$, je přidáno $\mathbf{p}_i$ do matice $\mathbf{U}$.

  Vzdálenost mezi dvěma pozicemi $\mathbf{p}_i = (x_i, y_i)$ a $\mathbf{p}_j = (x_j, y_j)$ je dána vztahem:

$$d(\mathbf{p}_i, \mathbf{p}_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

- **Výpis a vykreslení nalezených souřadnic**: na závěr jsou nalezené obce s kostelem vykresleny pomocí funkce `plot` a `rectangle` a vypsány pomocí funkce `disp`.

Část B

- **Segmentace:** Byla vytvořena funkce `Segment_kmeans.m` která převede vstupní data z *RGB* do *CIELab* a provede segmentaci pomocí vestavěné funkce `kmeans`. Původně probíhal výpočet s 6 clustry, následně kvůli zelenějšímu zbarvení části mapy byl počet navýšen na 12 clusterů.
  Navýšení počtu clusterů prodloužilo dobu segmentace, proto pro urychlení času byly výsledky uloženy jako typ `Structure array` do souborů `*.mat` tím se výpočet pro daný počet clusterů provede pouze jednou. Tento postup funguje i na jiných zařízeních.
- **Identifikace segmentů:** Jelikož výsledky funkce `kmeans` mají při každém použití různé pořadí výsledků a pro následné použití nelze využít předem nastavené hodnoty. Proto byla vytvořena funkce `generate_bands.m`, která zobrazeným výsledkům vytvoří proměnou s číslem vrstvy na základě uživatelských vstupů. Funkce ukládá proměnné do souborů `*.mat` pro opakované použití.
- **Filtrace:** Na filtraci dat byl použit $low - pass$ vyhlazovací filter, který využíval kernel velikosti $[5x5]$:

0	0.2	0.5	0.2	0
0.2	0.5	0.8	0.5	0.2
0.5	0.8	1.0	0.8	0.5
0.2	0.5	0.8	0.5	0.2
0	0.2	0.5	0.2	0

- **Mapová algebra:** Pro odstranění vzniklých děr po textech a vrstevnicích byly k rastru lesu připočteny rastry obsahující vrstevnice a texty. Od výsledného rastru byly odečteny rastry obsahující cesty, mýtiny a vodstvo.

- **Export dat:** Výsledně byly exportovány data v podobě formátu `tif`, který obsahuje pouze segmentovanou plochu lesa. Druhý výsledek je také `tif`, který obsahuje vrstvy R,G a segmentovaný les pro následnou georeferenci.
- **Segmentovaný obraz pomocí metody Graph Cut:** Segmentace plochy lesa z obrazu mapy byla provedena v softare Matlab pomocí funkce Graph Cut v aplikaci Image Segmenter. Jedná se o poloautomatickou techniku segmentace, kterou můžeme použít k rozdělení obrazu na prvky popředí a pozadí. Aplikace rozdělí obrázek na segmenty na základě nakreslených čar, jež lze déle zpřesňovat kreslením dalších linií do obrázku, dokud nejsme s výsledkem spokojeni. Tyto inicializační body nakreslených čáry představují pevná omezení a ostatní pixely jsou rozdělena globálně pomocí vypočtených podmínek, které určují, zda určitý pixel reprezentuje popředí nebo pozadí. Je vhodné označovat pixely reprezentující popředí takovým způsobem, aby pokryl celé barevné spektrum popředí. [5]
- **Vytvoření GeoPackage v systému UTM:** Soubor ve formátu `tif`, obsahující vrstvy R,G a binaární masku lesa byl nahrán do programu ArcGis pro, vrstvy R, G a B byly ponechány kvůli georeferenci. Gereferencovaný `tif` byl transformovaný do systému UTM. Pomocí funkce `Extract Bands` byla extrahována binární maska lesa. Funkcí `Raster to Polygon` byl vytvořen polygon lesa. Pomocí `Create SQLite Database` byl vytvořen soubor `Les.gpkg` obsahující vrstvu lesa v systému UTM.

5 Vstupní data

Vstupními daty pro tuto úlohu jsou naskenované mapy ve formátu JPEG, což má negativní vliv na kvalitu obrazu kvůli ztrátové kompresi viz U1. Konkrétně se jedná o následující mapy:

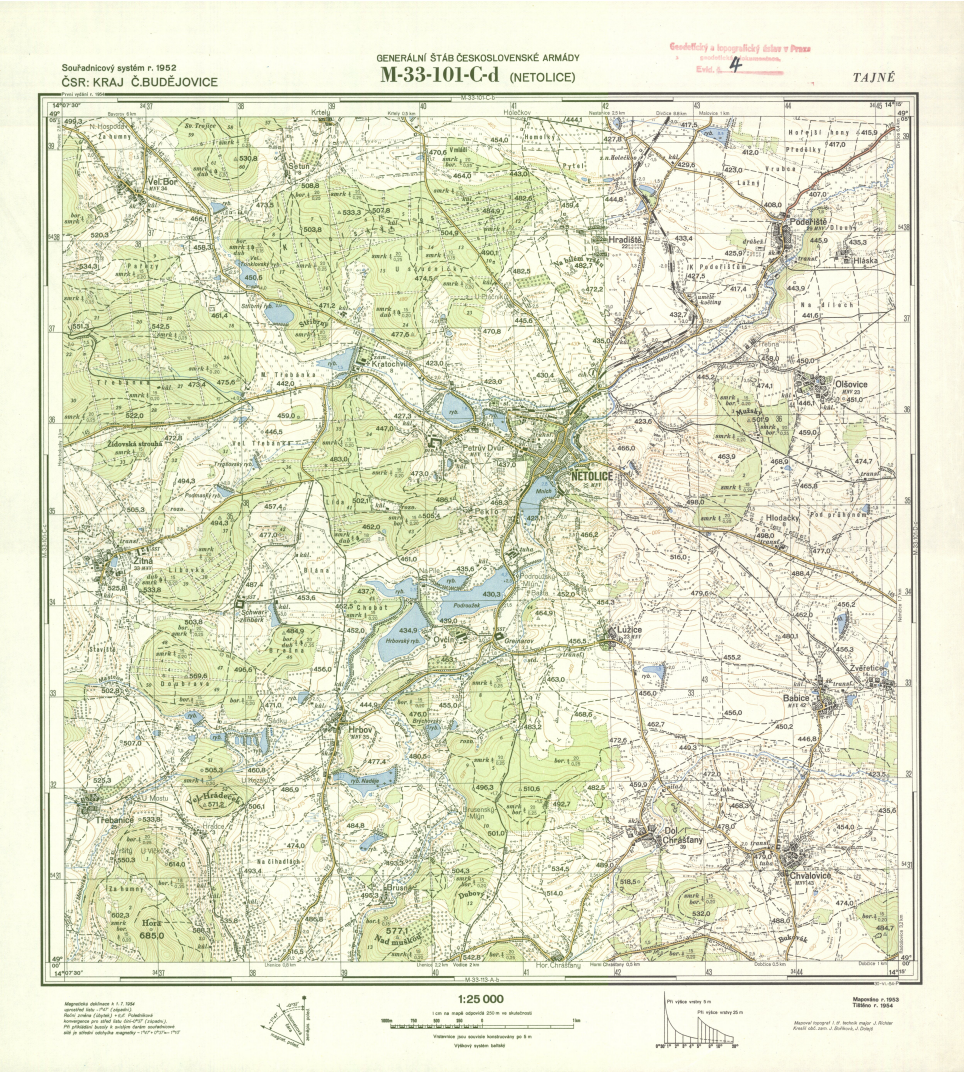
Část A

Müllerova mapa Čech: Vstupem je konkrétně mapový list 14 (*Sectio XIV*). Naskenovaný obraz mapy je uložen v adresáři `images` pod názvem `MMC14_sk3.jpg`.

Obrázek 1: Müllerova mapa Čech

Část B

Topografická mapa ČSSR: Konkrétně mapový list M-33-101-C-d (Netolice). Naskenovaný obraz mapy je uložen v adresáři `images` pod názvem `TM25_sk3.jpg`. Mapa je v sytému S-42 (3333).



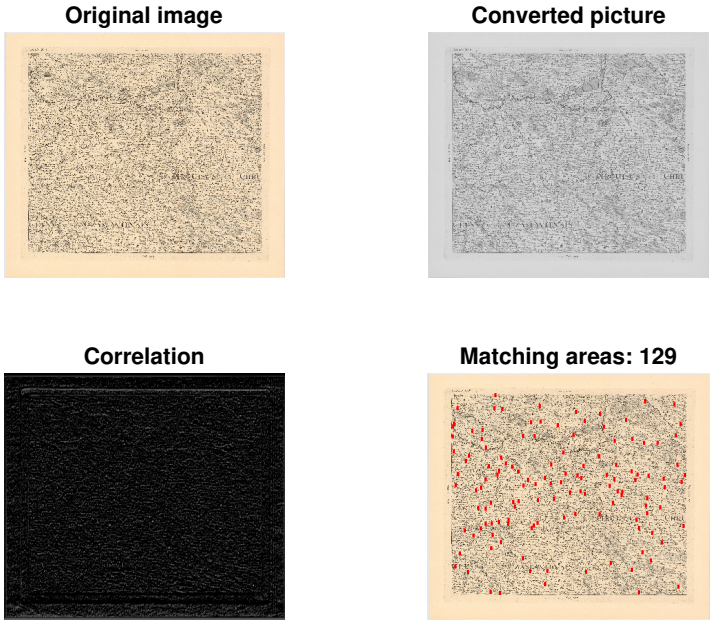
Obrázek 2: Topografická mapa ČSSR

6 Výstupní data

Část A

Graf s jednotlivými kroky

Tento subplot vykresluje napřed původní mapu, poté mapu převedenou do YCbCr s tím, že byla využita pouze složka Y, na níž byl dále aplikován Gaussův pohyblivý filtr (kernel). Dále jsou vykresleny hodnoty korelací a nakonec jsou v původní mapě červenými obdélníky vykresleny nalezené vesnice s kostelem.



Obrázek 3: Původní obrázek, konvertovaný obrázek s využitím složky Y prostoru YCBCR a následnou aplikací Gaussova filtru, hodnoty korelací a nalezené obce s kostelem označené červenými obdélníky

Detail nalezených obcí s kostelem

Obrázek 4: Detail nalezených vesnic

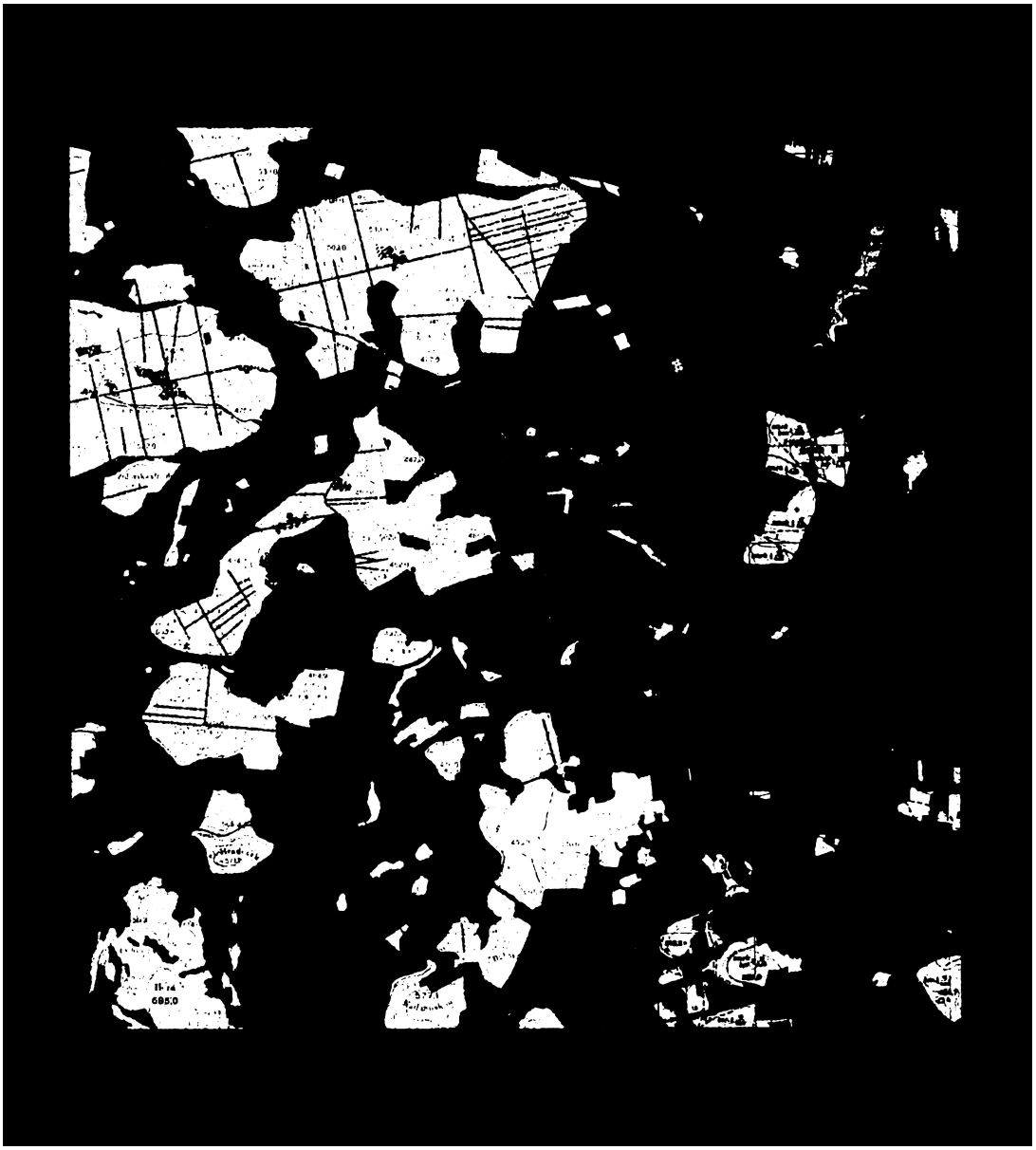
Souřadnic nalezených bodů

Tabulka 1: Souřadnice nalezených vesnic s kostelem

X	Y	X	Y	X	Y
5499	3335	4373	1029	1601	1243
3289	5291	1727	6910	1826	5698
3131	5067	4441	6253	4180	1782
3753	1409	2308	4776	5532	1122
1071	4135	2069	1622	2840	4171
1931	6403	4085	2252	2173	726
914	3120	2536	3440	971	832
1486	4908	2918	4278	863	6869
5211	6215	3912	4112	3085	3004
4253	7123	1826	1507	1364	2880
3033	3712	2543	6955	3108	780
1323	4375	5016	896	2509	2176
1342	3144	2716	1985	2825	4985
2808	840	5510	2851	2545	5250
6089	1752	5150	2650	1479	697
2203	3777	3603	5159	4076	5812
3124	2667	3385	5400	2588	1705
4322	1479	3186	1487	2818	5845
4703	2039	4186	1613	2963	2793
3316	3963	3768	1613	3603	1760
1574	1527	3257	1345	5871	3271
4869	1771	3712	2425	4189	2241
3478	6118	2852	6151	2916	3264
1412	734	1757	698	4518	1278
3025	7070	2791	3841	5473	5685
2346	2046	1740	2655	4302	2897
3298	5419	2730	5679	1295	5524
2437	1062	3283	4256	3354	4379
2293	1151	3581	2382	4311	5885
3731	6478	4984	5982	1443	3623
6097	5883	5524	4410	4497	1808
2931	6570	5939	6982	3066	3427
3926	4786	3169	6600	5376	768
2942	5198	3896	6113	6115	2017
1128	4799	4161	3059	3488	3521
3561	2533	2575	4337	974	1858
3532	3070	773	6056	4144	1982
2831	7163	1417	7042	4237	2957
5278	5097	3680	6074	993	2078
2736	2520	1935	6024	2599	2353
1878	5206	2823	1943	1711	4060
609	1886	2949	1613	4712	6523
2971	5783	4017	3798	1749	2976

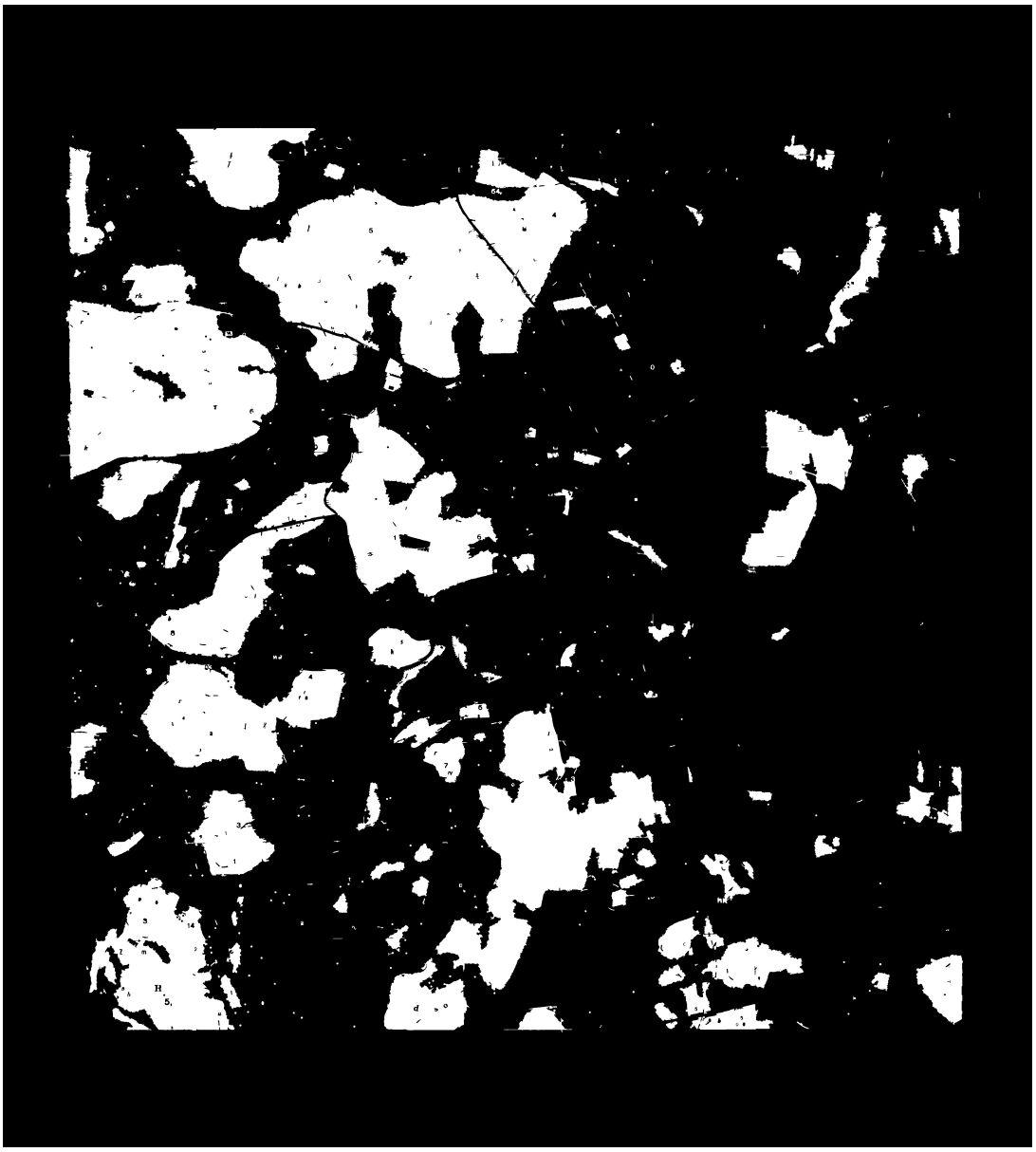
Část B

- **Hlavní úkol:**



Obrázek 5: Segmentovaný les z topografické mapy pomocí funkce $k - Means$

- **Bonusový úkol číslo 1:**



Obrázek 6: Segmentovaný les z topografické mapy pomocí Graph cut

Obrázek 7: Rozdíl výsledků

U porovnání obou výsledků je vidět že filtrací dat přišel výsledek vypočítaný pomocí funkce $k-means$ o kraje segmentovaného lesa. Segmentace pomocí Graph cut si nedokázala poradit s lesními cestami a šumem, který procesem vznikl. Funkce Graph cut si dokázala lépe poradit s problémem, kdy každá část mapy měla různý odstín zelený, způsobený degradací.

- **Bonusový úkol číslo 2:**
  Výsledkem je soubor `Les.gpkg`, který obsahuje polygon segmentovaného lesa v souřadnicovém systému UTM (32633).

7 Závěr

Část A:

V první části byl implementován algoritmus pro vyhledávání vzorů v Müllerově mapě Čech na základě obrazové korelace. Jako vzor slouží průměr pěti uživatelem zvolených nebo předdefinovaných vzorů. Pro vyhledávání byla využita jasová složka Y barevného prostoru YCbCr a data byla dále upravena pomocí

Gaussova filtru. Nalezené prvky byly očištěny o vícenásobné nálezy a byly vykresleny do původní mapy, přičemž jejich souřadnice byly zároveň vypsány.

Část B:

Byl segmentován les z topografické mapy pomocí funkce $k-Means$. Následně byl les segmentován pomocí aplikace Graph cut, výsledky byl porovnány. Výsledný rastr byl převeden do polygonové vrstvy a uložen do GeoPackage v systému UTM.

8 Náměty na vylepšení:

Část A:

- **Chybná identifikace:** Jedním z problémů, který nastal při vyhledávání vesnic s kostelem, je mylné označení městysů jako vesnic s kostelem. Tento problém vzniká, protože prostřední část značky městys se velmi podobá značce vesnice s kostelem. To vede k falešným nálezům, kdy jsou městysy nesprávně identifikovány jako vesnice s kostelem.
- **Chybné nenalezení:** Dalším problémem je, že ne všechny vesnice s kostelem jsou detekovány. Tento problém by mohl být řešen snížením limitní hodnoty korelace pro označení vesnice s kostelem. Nicméně, snížení tohoto limitu by také vedlo k většímu množství mylných nálezů.
- **Vyhledání více objektů:** Jedním z možných řešení je vyhledávat kromě vesnic s kostelem také městysy. Pokud se nalezený městys překrývá s nalezenou vesnicí s kostelem, může být tato vesnice považována za chybný nález a odstraněna ze seznamu nalezených vesnic s kostelem. Toto by zabránilo mylným nálezům a zároveň umožnilo snížení limitní hodnoty korelace, aby došlo k nálezu všech vesnic s kostelem.

Část B:

- **Manipulace s daty:** pro efektivnější segmentaci by bylo výhodné vytvořit masku, která by z dat získala pouze mapové pole.
- **Segmentace:** pro univerzálnější řešení této úlohy, by bylo lepší aby počet clusterů byl nastavitelný parametr, který vstupuje do funkce.
  Dále byl celý proces měl pro získání co nejlepších výsledků procházet data v kernelu a následně složit výsledky.
- **Identifikace segmentů:** pro usnadnění práce s výsledky vynechat zadávání výsledků skrz Command Window což je nepřehledné. Pro usnadnění by bylo lepší zadávat parametry skrz jednoduché UI.

Odkazy

[1] Tomáš Bayer. *Algoritmy pro kompresi rastrových dat.* URL: `https://web.natur.cuni.cz/~bayertom/images/courses/Apk/komprese.pdf` (cit. 21. 10. 2024).

[2] ČVUT v Praze Katedra geomatiky Fakulta stavební. *Filtrace obrazu.* URL: `https://k155cvut.github.io/dprz/cviceni/cviceni4/` (cit. 03. 11. 2024).

[3] MathWorks. *imgaussfilt.* URL: `https://www.mathworks.com/help/images/ref/imgaussfilt.html` (cit. 03. 11. 2024).

[4] ČVUT v Praze Katedra geomatiky Fakulta stavební. *153ZODH / 5. cvičení.* URL: `https://geo.fsv.cvut.cz/gwiki/153ZODH_/_5._cvi%C4%8Den%C3%AD` (cit. 03. 11. 2024).

[5] MathWorks. *Segment Image Using Graph Cut in Image Segmenter.* URL: `https://www.mathworks.com/help/images/segment-image-using-graph-cut.html` (cit. 04. 11. 2024).