

UNIVERZITA KARLOVA V PRAZE

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie

Studijní program: Geografie (magisterské studium)

Studijní obor: Kartografie a geoinformatika



Bc. Matěj SOUKUP

**AUTOMATIZOVANÁ ANALÝZA PODOBNOSTI MAP
S VYUŽITÍM INVARIANTŮ**

**AUTOMATIZED MAP SIMILARITY ANALYSIS USING SHAPE
INVARIANTS**

Diplomová práce

Vedoucí diplomové práce: Ing. Tomáš Bayer, Ph.D.

Praha 2014

Zadání diplomové práce

Téma:	Automatizovaná analýza podobnosti map s využitím invariantů
Název:	Automatizovaná analýza podobnosti map s využitím invariantů
Název v českém jazyce:	Automatizovaná analýza podobnosti map s využitím invariantů
Akademický rok vypsání:	2012/2013
Typ práce:	diplomová práce
Jazyk práce:	
Ústav:	Katedra apl. geoinformatiky a kartografie (31-370)
Vedoucí / školitel:	Ing. Tomáš Bayer, Ph.D.
Řešitel:	Bc. Matěj Soukup - zadáno vedoucím/školitelem
Datum přihlášení:	27. 01. 2013
Datum zadání:	26. 02. 2013

Předběžná náplň práce

Diplomová práce je zaměřena na automatickou analýzu podobnosti vektorových map, využívající tvarové a atributové podobnosti vzájemně odpovídajících si objektů, popsaných s využitím vhodné parametrizace. Cílem práce je navrhnout, implementovat a analyzovat algoritmus „měřící“ podobnost map na základě automatického rozpoznání a parametrizace odpovídajících si 1D / 2D objektů dvou mapových vrstev znázorňujících stejné území. Mapové vrstvy se mohou lišit např. vztahným měřítkem, souřadnicovým systémem, natočením, mírou generalizace, obsahem či přesností zákresu jednotlivých objektů. Navrhovaný algoritmus vylepší stávající přístupy hledání vzájemně odpovídajících si prvků, založené převážně na posuzování tvaru objektů, o možnost analýzy vzájemné polohy, sousednosti objektů a vztahu analyzovaného objektu k těmto objektům. Výsledné kritérium bude určeno jako lineární kombinace jednotlivých kritérií a poslouží k výpočtu míry podobnosti obou map. V praktické části bude provedena implementace algoritmu ve zvoleném programovacím jazyce a ověření funkcionality na různých datových sadách jak z infromatického, tak z kartografického pohledu.

Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze dne 25. 8. 2014

.....

podpis

Poděkování

Rád bych zde na prvním místě poděkoval vedoucímu mé práce Ing. Tomáši Bayerovi, Ph.D. za čas i ochotu při konzultacích, cenné rady a připomínky k práci. Velký dík patří rovněž mé rodině za veškerou podporu při zpracovávání této práce.

Automatizovaná analýza podobnosti map s využitím invariantů

Abstrakt

Práce je zaměřena na využití technik pro rozpoznávání tvaru k automatizovanému porovnání dvou různých mapových vrstev. Mapové vrstvy se mohou lišit například vztažným měřítkem, souřadnicovým systémem, natočením, mírou generalizace, obsahem či přesností zakresu jednotlivých objektů. Hlavní část práce se zabývá hledáním shodných objektů na obou mapových vrstvách a jejich přiřazováním k sobě. K tomuto přiřazení je používán shape invariant turning function a další metody založené na poloze objektů. Jednotlivé metody jsou testovány na objektech konkrétních mapových vrstev a navzájem porovnávány. Na základě výsledků testů je navržena vhodná kombinace metod sloužící k přiřazení odpovídajících objektů. Po přiřazení jsou obě mapové vrstvy porovnávány z několika různých hledisek. Navržený algoritmus je implementován v programovacím jazyku Python.

Klíčová slova: podobnost map, podobnost tvarů, míry podobnosti, turning function

Automatized Map Similarity Analysis Using Shape Invariants

Abstract

This diploma thesis is focused on using shape similarity measures for automatized map comparison. Maps can vary e.g. in the reference scale, coordinate system, rotation, degree of generalization, map content or accuracy of depiction of individual objects. The main part of the thesis deals with finding identical objects on both map layers and their matching using shape invariant turning function and other methods based on object location. These methods are tested on specific maps and compared with each other. The specific combination of methods is proposed and used for matching objects. After that the maps are compared to each other from several different perspectives. The proposed algorithm is implemented in Python programming language.

Keywords: map similarity, shape similarity, similarity measures, turning function

OBSAH

PŘEHLED POUŽITÝCH ZKRATEK	8
SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK	10
SEZNAM PŘÍLOH.....	10
1 ÚVOD	11
2 UVEDENÍ DO PROBLEMATIKY A LITERÁRNÍ REŠERŠE	12
2.1 Podobnost tvarů	12
2.2 Specifika použití technik shape matching v geoinformaticce	13
2.3 Popis jednotlivých metod	13
2.4 Srovnání technik	21
2.5 Dosavadní použité metody v geoinformaticce	22
2.6 Analýza podobnosti map	26
3 MAPY A JEJICH PODOBNOST	27
3.1 Mapa	27
3.2 Podobnost map	28
3.3 Podobnost objektů	29
3.4 Charakter porovnávaných dat	30
3.5 Vliv rozdílných parametrů na určení podobnosti mapových vrstev	30
3.5.1 Vliv změny měřítka.....	31
3.5.2 Vliv generalizace.....	31
3.5.3 Vliv kartografického zobrazení.....	34
4 ZÁKLADNÍ GEOMETRICKÉ STRUKTURY, POJMY A ALGORITMY	35
4.1 Polygon a centroid polygonu	35
4.2 Metriky a měření podobnosti.....	37
4.3 Voroného diagram	38
4.4 Přiřazovací problém a maďarská metoda	39
5 ANALÝZA PODOBNOSTI TVARŮ	41
5.1 Turning function	41
5.2 Vzdálenost turning funkcí.....	43
5.3 Invariance vůči translaci a rotaci	47
6 ANALÝZA PODOBNOSTI POLOHY.....	49
6.1 Absolutní určení polohy	49

6.2	Relativní určení polohy.....	49
6.2.1	Vymezení okolí pomocí euklidovské vzdálenosti	50
6.2.2	Vymezení okolí prostřednictvím Voroného diagramů.....	52
7	NÁVRH ALGORITMU PRO ANALÝZU PODOBNOSTI MAP	54
7.1	Dílčí metriky a jejich kombinace.....	55
7.1.1	Analýza tvaru	56
7.1.2	Analýza absolutní polohy	66
7.1.3	Analýza relativní polohy	66
7.1.4	Analýza velikosti objektu	68
7.2	Nalezení odpovídacích si objektů.....	69
7.2.1	Matice podobnosti a nastavení vah	69
7.2.2	Přiřazení odpovídajících objektů	70
7.3	Procházení datasetu	71
7.4	Porovnání map	73
7.4.1	Individuální hledisko	74
7.4.2	Lokální hledisko.....	74
7.4.3	Globální hledisko	76
8	IMPLEMENTACE	77
8.1	Datová reprezentace polygonů	78
8.2	Zpracování objektů v dlaždicích.....	80
8.2.1	Turning funkce a jejich porovnání	82
8.2.2	Porovnání okolí	83
8.2.3	Testování vah	85
8.3	Porovnání mapových vrstev	85
9	HODNOCENÍ VÝSLEDKŮ	88
9.1	Testovací data	88
9.2	Vybraná testovací území.....	90
9.3	Testování dílčích metrik	93
9.4	Nalezení vah pro kombinovanou metriku.....	94
9.5	Podobnost testovacích vrstev	97
9.6	Zhodnocení navrženého algoritmu	98
10	DISKUZE A ZÁVĚR	103
11	REFERENCE.....	105

PŘEHLED POUŽITÝCH ZKRATEK

CSS	Curvature Scale Space
DTW	Dynamic Time Warping distance
GIS	Geoinformační systém
ICP	Iterative Closest Point
IDSC	Inner Distance Shape Context
MCC	Multiscale Convexity Concavity Representation
TAR	Triangle Area Representation
TF	Turning function

SEZNAM OBRÁZKŮ

Obr. 1: Vzdálenost centroidů.....	14
Obr. 2: Symmetric Difference, verze bez i se ztotožněním centroidů.....	14
Obr. 3: Kompaktnost	15
Obr. 4: Hausdorffova míra.....	16
Obr. 5: Angle Histogram	17
Obr. 6: Polygon a jeho turning function	17
Obr. 7: Polygon a jeho reprezentace metodou TAR.....	18
Obr. 8: Výpočet log-polárních histogramů.....	19
Obr. 9: Vnitřní vzdálenost	19
Obr. 10: Reprezentace polygonů metodou CSS	20
Obr. 11: Objekty v mapových vrstvách s kardinalitou $N : 1$ a $M : N$	23
Obr. 12: Možný postup při problémech s kardinalitou.....	24
Obr. 13: Generalizace plošných prvků	32
Obr. 14: Generalizace skupin plošných prvků.....	33
Obr. 15: Simple polygon a self-intersecting polygon.....	36
Obr. 16: Donut polygon.....	36
Obr. 17: Voroného diagram nad množinou bodů v rovině.....	39
Obr. 18: Polygon a jeho turning funkce	43
Obr. 19: Přidání přístavku a prodloužení obvodu.....	48
Obr. 20: Okolí objektu P rozdělené do sektorů podle polárních souřadnic.....	51
Obr. 21: Vynechání objektu a změna Voroného diagramu	53
Obr. 22: Rozdělení TF na obdélníkové pásy	60
Obr. 23: Posun TF o t	62
Obr. 24: Výpočet vzdálenosti dvou TF metodou obdélníkových pásů	63
Obr. 25: Tvorba matice M součtem obsahů ploch v sektorech	68
Obr. 26: Testování napříč dlaždicemi.....	73
Obr. 27: Výběr společných sousedů	75
Obr. 28: Testované mapové vrstvy	89
Obr. 29: Testovací území Ořechovka (O) na vrstvě KN	91
Obr. 30: Testovací území Břevnov (B) na vrstvě KN	91
Obr. 31: Testovací území Karlovo náměstí (K) na vrstvě KN	92
Obr. 32: Testovací území Braník (R) na vrstvě KN	92

Obr. 33: Vynechání částí objektu	100
Obr. 34: Agregace objektů.....	101

SEZNAM TABULEK

Tab. 1: Výsledky některých metod na MPEG-7.....	22
Tab. 2: Úspěšnost přiřazení při použití metrik samostatně	93
Tab. 3: Optimální nastavení vah.....	95
Tab. 4: Optimální nastavení vah bez absolutní polohy	96
Tab. 5: Optimální nastavení vah bez absolutní i relativní polohy	97
Tab. 6: Výsledky porovnání mapových vrstev	98

SEZNAM PŘÍLOH

Příloha 1: DVD obsahující text práce, zdrojové kódy aplikace, testované mapové vrstvy

1 ÚVOD

Geoinformační systémy (GIS) pracují s velkým množstvím kvalitativně odlišných prostorových dat. Data uložená ve formě mapových vrstev se dostávají do systémů různými způsoby, jako je digitalizace starších map, zaměření v terénu či export z existujících databází prostorových dat. V závislosti na svém původu, způsobu pořízení či účelu dat se mapové vrstvy mohou lišit vztažným měřítkem, souřadnicovým systémem, mírou generalizace, obsahem, přesností zákresu jednotlivých objektů nebo aktuálností.

Pro různé aplikace může být důležité porovnání dvou obsahově podobných mapových vrstev znázorňujících stejné území. Dvě obsahově podobné mapy se mohou lišit počtem, polohou i tvarem zobrazených objektů. Cílem této práce je navrhnout, implementovat a analyzovat algoritmus, který dokáže zhodnotit podobnost dvou vektorových map. Tento algoritmus bude pracovat na základě parametrizace a automatického přiřazení odpovídajících si 2D objektů obou mapových vrstev. K nalezení a přiřazení vzájemně odpovídajících si objektů navrhovaný algoritmus použije jak tvar objektů, tak i další charakteristiky, jakými jsou vzájemná poloha, sousednost nebo velikost objektů. V praktické části bude provedena implementace algoritmu a ověření funkcionality na různých datových sadách.

Práce je členěna do několika kapitol. Úvod práce je věnován přehledu různých metod určování podobnosti a přiřazování objektů, a to jak z obecného hlediska, tak i z hlediska využití v geoinformaticce. Ve třetí kapitole jsou shrnuty různé odlišnosti a problémy, které mohou nastat při porovnání dvou rozdílných mapových vrstev. Kapitola 4 se zabývá vysvětlením některých pojmů a pomocných algoritmů, které jsou v dále využívány. Konkrétní provedení určení vzájemné podobnosti dvou tvarů za pomoci metody turning function je popsáno v páté kapitole. Šestá kapitola se zabývá různými způsoby určení podobnosti polohy dvou objektů na mapách. V kapitole 7 je navržen kompletní algoritmus pro porovnávání mapových vrstev a v další kapitole je popsána jeho implementace. Na závěr je provedeno zhodnocení výsledků a efektivity navrženého algoritmu.

2 UVEDENÍ DO PROBLEMATIKY A LITERÁRNÍ REŠERŠE

V následující části jsou shrnuty a popsány některé často používané techniky rozpoznávání a přiřazování tvarů (shape matching) nalezené v literatuře ve formě rešerše. Jsou zde popsána specifika jejich použití v geoinformatice a je diskutována vhodnost použití různých technik na tvary objektů v mapových vrstvách. Dále se rešerše zabývá způsoby určování podobnosti celých mapových sad.

2.1 Podobnost tvarů

Rozpoznávání tvarů, jejich přiřazování k sobě a určování jejich podobnosti se věnuje část počítačové grafiky zaměřená na tzv. shape matching. Metody používané k přiřazování tvarů mají velké využití v řadě oborů. Používají se nejen v informatice (např. v počítačovém vidění), ale i v kartografii, molekulární biologii či počítačové animaci (Alt, Guibas, 1999).

Při porovnávání dvou obecných tvarů je nutné tvar nějakým způsobem popsat a následně určit způsob, kterým lze zhodnotit, zda jsou si objekty podobné či nikoliv. K popsání tvaru za účelem jejich dalšího porovnání se používají tzv. **shape deskriptory** ve spojení s tzv. **shape similarity measures** (míry podobnosti tvarů), které dokáží vyjádřit podobnost dvou tvarů. Tyto metody se liší např. podle prostorové dimenze objektů, na které jsou použitelné. Většina těchto deskriptorů je použitelná pro 2D objekty, některé je možné použít i pro objekty nižších prostorových dimenzí nebo 3D objekty. Liší se také svou výpočetní složitostí a v neposlední řadě citlivostí vůči různým změnám tvaru.

Objekty mohou být vzájemně otočené, mohou mít odlišnou polohu či měřítko, mohou být různě zkosené, protažené nebo jinak lineárně i nelineárně deformované. U 3D objektů se v závislosti na úhlu pohledu projeví afinní transformace objektu. U kartografických objektů se jejich tvar mění v závislosti na použitém kartografickém zobrazení. V některých případech je třeba, aby použitý shape deskriptor zhodnotil objekt nezávisle na některých deformacích. Příkladem může být např. potřeba rozpoznat tvar objektu v prostoru nezávisle na úhlu pohledu. To je využíváno třeba při snímání registračních značek vozidel, kdy každá registrační značka může být jinak natočena ke kameře. Jiným případem je rozpoznávání osob. Jestliže je rozpoznáván obličej osoby, je třeba, aby podobnost se vzorem byla posuzována nezávisle na natočení a sklonu hlavy či vzdálenosti snímaného člověka od

kamery. Z těchto důvodů bývají míry podobnosti tvarů navrženy tak, aby nebyly citlivé na určité typy změn tvaru objektu.

Pokud není míra podobnosti tvaru citlivá na určitý typ transformace, nazýváme ji **invariantní** vůči dané transformaci. Například u míry invariantní vůči rotaci nezáleží na tom, jak je objekt otočen, vždy bude hodnocen stejně, tedy nezávisle na rotaci. Mezi nejčastější požadované invariance patří invariance vůči translaci a rotaci objektu. Obě tyto změny objektu patří do skupiny lineárních transformací.

2.2 Specifika použití technik shape matching v geoinformaticce

Využití metod shape matchingu v geoinformaticce a digitální kartografii má svá specifika. V mapových vrstvách porovnáváme zpravidla 2D objekty, nejčastěji jde o linie a polygony. Tvar totožného objektu ve dvou různých vrstvách se může lišit z mnoha různých důvodů. Může to být způsobeno nižší přesností jedné z vrstev nebo rozdílnou dobou vzniku obou vrstev. Jednou z nejčastějších příčin je i rozdílná generalizace obou vrstev. Tyto odlišnosti budou podrobně popsány v kapitole 3.1.

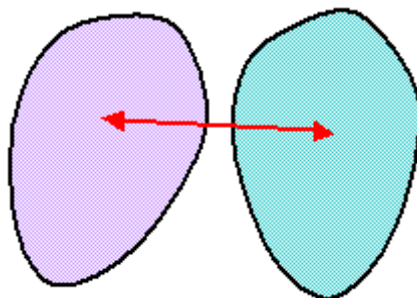
Pro použití technik shape matching v geoinformaticce je podstatná přítomnost a důležitost informace o poloze analyzovaných objektů. V některých případech je poloha důležitější než tvar. Konkrétní město může být na mapě zobrazeno jako složitý polygon nebo jako jeden bod (tj. plošným či bodovým kartografickým znakem), avšak jeho poloha převedená z mapy do zeměpisných souřadnic bude v obou případech podobná.

Při porovnávání objektů v různých mapových vrstvách není nutné se omezovat pouze na tvar objektu, ale je možné pracovat i s polohou objektu na zemském povrchu. Z tohoto důvodu je škála technik využívajících tvar rozšířena o ty, které tvar kombinují s polohou objektů.

2.3 Popis jednotlivých metod

Následuje přehled některých, v literatuře často zmiňovaných metod, podle kterých se zjišťuje podobnost či shodnost objektů. Jsou zde uvedeny jak metody pracující pouze s tvarem objektu, tak i metody využívající zároveň i polohu objektů. U každé metody je popsána její použitelnost pro prvky různých prostorových dimenzí v geoinformaticce.

Vzdálenost centroidů. Vzdálenost centroidů (Centroid distance) je jednoduchá míra využívající pouze polohu objektů. Centroid je geometrickým středem objektu, jeho výpočet bude podrobně řešen v kapitole 4.1. Tato míra měří vzdálenost centroidů obou posuzovaných objektů (Obr. 1). Čím jsou centroidy blíže, tím jsou objekty považovány za podobnější. Je použitelná pro body, linie i polygony.

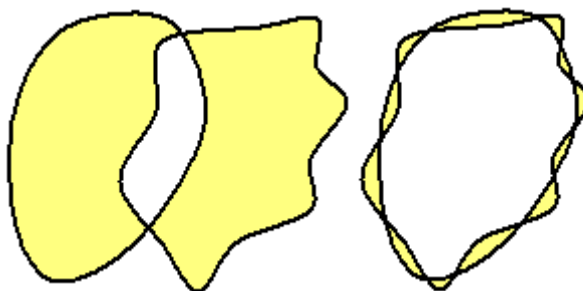


Obr. 1: Vzdálenost centroidů, zdroj: Vivid Solutions 2003

Symmetric Difference. Jednoduchá míra podobnosti objektu využívající polohu i tvar posuzovaných objektů. Vyjadřuje obsah plochy, kterou posuzované objekty nemají společnou. Je počítána jako sjednocení ploch obou objektů mínus plocha jejich průniku. Symbolicky ji lze pro plochy objektů A a B zapsat jako:

$$A \Delta B = (A \cup B) \setminus (A \cap B)$$

Čím je symmetric difference menší, tím více jsou si objekty podobné. Je použitelná pouze pro polygony.



Obr. 2: Symmetric Difference, verze bez (vlevo) i se ztotožněním centroidů (vpravo), zdroj: Vivid Solutions, 2003

Podobně jako vzdálenost centroidů je i tato metoda přímo závislá na poloze objektů v souřadnicovém systému. Metodu je však možné upravit tak, by byla invariantní vůči poloze. Před samotným výpočtem se jeden objekt přesune nad druhý objekt tak, aby centroidy obou splynuly a teprve poté je vypočítán rozdíl jejich sjednocení a průniku. Obě varianty jsou znázorněny na Obr. 2.

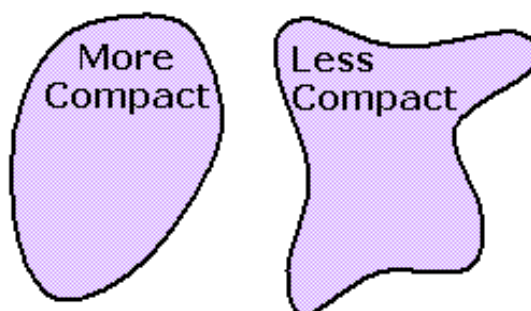
Podíl překrytí. Podíl překrytí P (Percentage of Overlap) je podobná míra jako symmetric difference. V tomto případě je počítán podíl společné části tvarů, tj. jejich průniku, vůči jejich sjednocení. Čím větší je překrytí objektů, tím větší podobnost. Na rozdíl od symmetric difference je míra relativní:

$$P = \frac{(A \cup B)}{(A \cap B)}$$

Kompaktnost. Kompaktnost (Compactness) C je jednoduchý atribut vypočítaný z plochy S a obvodu O polygonu:

$$C = \frac{S}{O}$$

Hodnoty kompaktnosti mezi oběma objekty můžeme porovnávat (Obr. 3). Podobné objekty by měly mít podobnou hodnotu kompaktnosti. Míra není vhodná pro výrazně nekonvexní objekty, protože nedostatečně zachycuje skutečný tvar objektu.



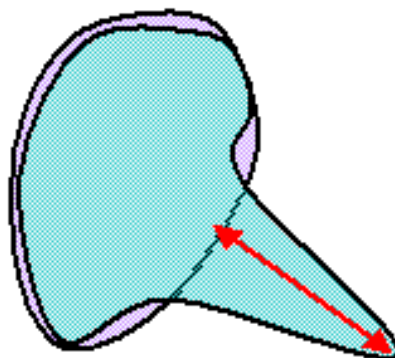
Obr. 3: Kompaktnost, zdroj: Vivid Solutions, 2003

Hausdorffova míra. Hausdorffova míra (Hausdorff Distance) d_H je jednou z nejpoužívanějších měr pro analýzu tvaru objektu. V rámci dvou porovnávaných objektů je to maximum ze všech minimálních vzdáleností mezi dvěma tvary (Obr. 4). Pro polygony P a Q je to nejmenší δ , kdy v δ -okolí polygonu P je obsažen celý polygon Q a naopak (Alt et al., 2004).

Hausdorffova míra je definována jako:

$$d_H(P, Q) = \max \left\{ \max_{p \in P} \min_{q \in Q} d(p, q), \max_{q \in Q} \min_{p \in P} d(p, q) \right\},$$

kde $d(p, q)$ je vzdálenost bodů p a q , které jsou libovolnými body polygonů P a Q . Čím menší je vzdálenost $d_H(P, Q)$, tím jsou si křivky podobnější. Je použitelná jak pro linie, tak pro polygony.



Obr. 4: Hausdorffova míra, zdroj: Vivid Solutions 2003

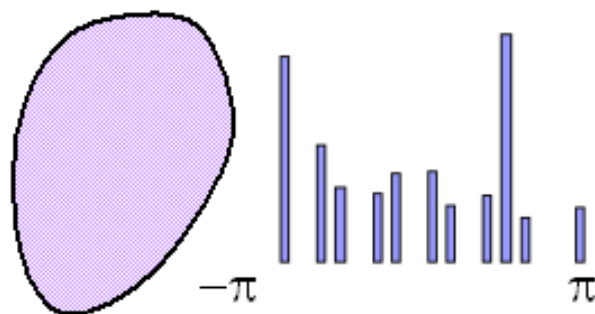
Na rozdíl od předcházejících měř je zde posuzován skutečný tvar objektu, ne pouze překrytí ploch. V některých případech může vycházet tato vzdálenost poměrně malá i přesto, že jsou křivky velice odlišné (Alt et al., 2004). Stává se to tehdy, když jsou tvary podobně velké a mají podobné konvexní obálky.

Obdobně jako u symmetric difference lze tuto míru modifikovat jako invariantní vůči translaci tím, že dojde ke ztotožnění centroidů tvarů.

Fréchet distance. Tato míra je velice podobná Hausdorffově míře. Lze si ji pro zjednodušení představit takto: Dva porovnávané polygony si představíme jako trajektorie, po kterých se pohybují dva objekty spojené spolu provazem. Fréchet distance je pak minimální délka provazu, který bude pro celou trasu potřeba, přičemž oba objekty mohou měnit rychlost a zastavovat se na libovolně dlouhou dobu, ale nesmí couvat. Existuje i slabá forma této míry, kdy je přípustné couvat.

V některých případech je vhodnější než Hausdorffova míra, ale její výpočet je podstatně obtížnější (Alt et al., 2004), proto je v praxi málo používaná. Pro konvexní polygony vycházejí obě stejně.

Angle Histogram. Pro každou hranu polygonu je zjištěn úhel, který svírá s osou x. Podle velikosti tohoto úhlu jsou poté hrany rozděleny do několika tříd intervalu úhlů. V každé třídě je sečtena celková délka hran. Tím je vytvořen tzv. histogramu úhlů (Obr. 5). Tyto histogramy mohou být mezi sebou porovnány. Tvary jsou tím podobnější, čím jsou podobnější jejich histogramy úhlů.

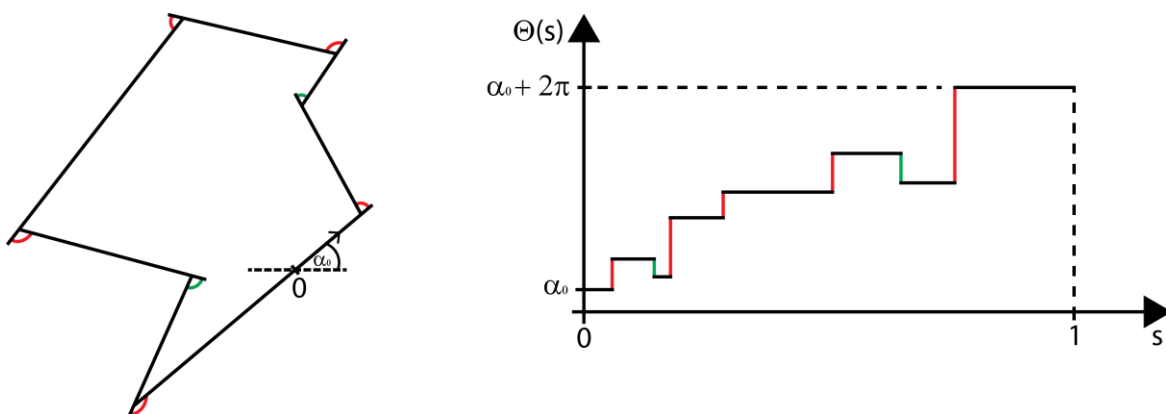


Obr. 5: Angle Histogram, zdroj: Vivid Solutions 2003

Turning Function. Turning function je metrika založená na parametrizaci objektu. Tato metrika pro každý bod měří jeho kumulativní vzdálenost po křivce (arc length) od vztažného bodu a úhel sevřený tečnou v tomto bodě a osou x (proti směru hodinových ručiček). Při každém zatočení doleva funkce stoupá, při zatočení doprava klesá (Veltkamp, 2001). Takto zaznamenané body tvoří po částech konstantní funkci pro každý tvar (Obr. 6). Funkce je možné mezi sebou porovnat pomocí jejich L_p vzdálenosti. Podobné objekty budou mít podobnou turning function (TF).

Ze své podstaty je invariantní vůči posunu (translaci). Funkce není invariantní vůči měřítku, tudíž před jejím použitím je nutné, aby objekty byly normovány. Z definice funkce také jasně vyplývá, že je velmi citlivá na velikost úhlů.

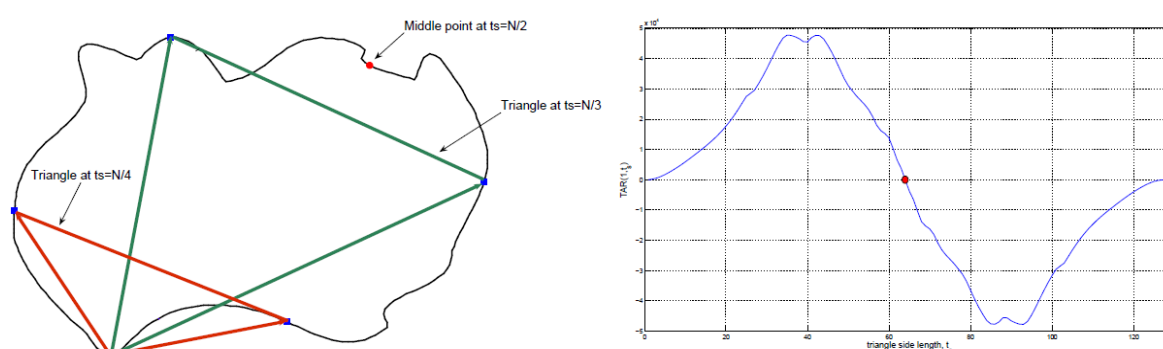
Algoritmus lze modifikovat tak, že není nutné porovnávat celé objekty, ale pouze jejich části (Cohen, Guibas, 1997). To může být v digitální kartografii výhodné zejména tehdy, když by došlo k agregaci dvou objektů z jedné vrstvy do jednoho objektu v druhé vrstvě. V agregovaném objektu by tak bylo možno rozpoznat části tvarů původních objektů.



Obr. 6: Polygon a jeho turning function, zdroj: vlastní

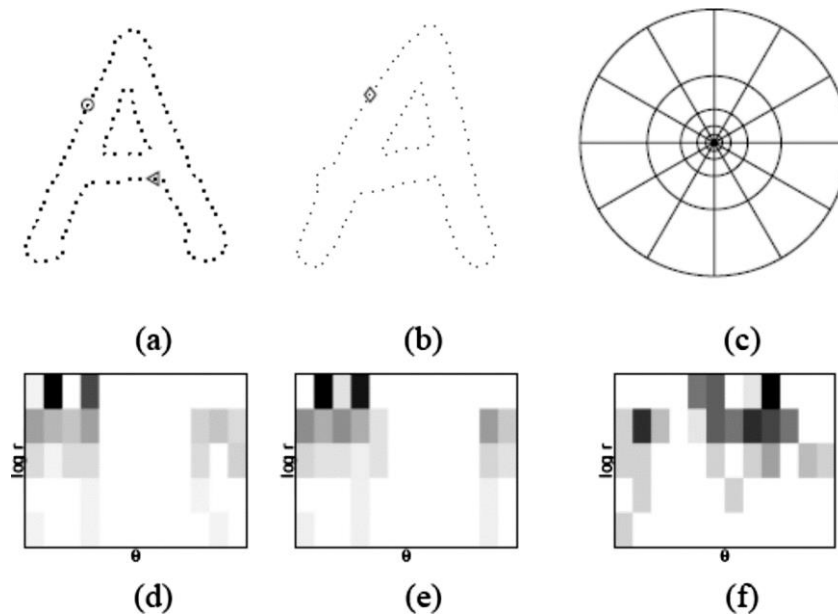
TAR. Triangle Area Representation (Alajlan et al., 2007) představuje charakteristiku objektu, která je normalizována vůči danému měřítku. Nejprve je objekt převzorkován na

daný počet bodů. Od každého bodu je sestaven rovnoramenný trojúhelník o délce ramena t , kde hlavní vrchol je v posuzovaném bodě a zbylé dva vrcholy leží někde na obrysu objektu v místech, kde kružnice o poloměru t protíná obrys polygonu. V tomto trojúhelníku měříme míru konkávnosti či konvexnosti úhlu u hlavního vrcholu (Obr. 7). Délka ramena t nám reprezentuje měřítko posuzovaných detailů. Při malých t se odrazí i drobné detaily na křivce, zatímco velká t zachytí spíše globální charakteristiky křivky. Jedná se tedy o charakteristiku křivosti křivky v daném bodě. Charakteristiky zjišťujeme pro velký počet bodů pro různá t . Vynesením charakteristik za všechny body za různá t vzniká graf. Takto popsané tvary objektů je možno porovnat mezi sebou.



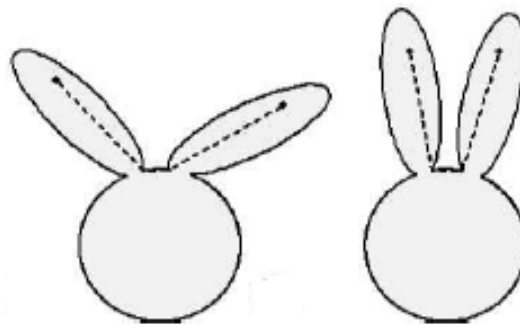
Obr. 7: Polygon a jeho reprezentace metodou TAR při vzrůstající délce ramena trojúhelníku, zdroj: Alajlan et al., 2007

IDSC. Přístup Inner Distance Shape Context spojuje deskriptor Shape Context (SC) s Inner Distance (Ling, Jacobs, 2005). Shape Context pracuje s body ležícími na hranici objektu, nejčastěji vrcholy polygonu. Pro každý bod zjišťuje vzdálenost a směr ke všem ostatním bodům (tedy jejich relativní pozici). Každý bod je zaznamenáván do log-polárního histogramu. Jak je naznačeno na Obr. 8, je zaznamenáván na ose x je azimut θ , na ose y logaritmovaná vzdálenost ($\log r$) a četnost je značena intenzitou barvy. Na základě podobnosti těchto histogramů se každému bodu jednoho objektu přiřadí odpovídající bod v druhém objektu. Na závěr se vypočte suma rozdílů histogramů odpovídajících si bodů vypovídající o podobnosti obou objektů. Shape Context je invariantní vůči posunutí (je relativní), a po normalizování vzdáleností průměrnou vzdáleností mezi body rovněž i vůči změně měřítka. Invariantnosti vůči natočení může být dosaženo měřením úhlů k tečně v daném bodě (body jsou na hranách objektů) namísto azimutů (Belongie et al., 2002).



Obr. 8: Výpočet log-polárních histogramů konkrétního bodu v obou objektech a jejich rozdílu, zdroj: Belongie et al., 2002

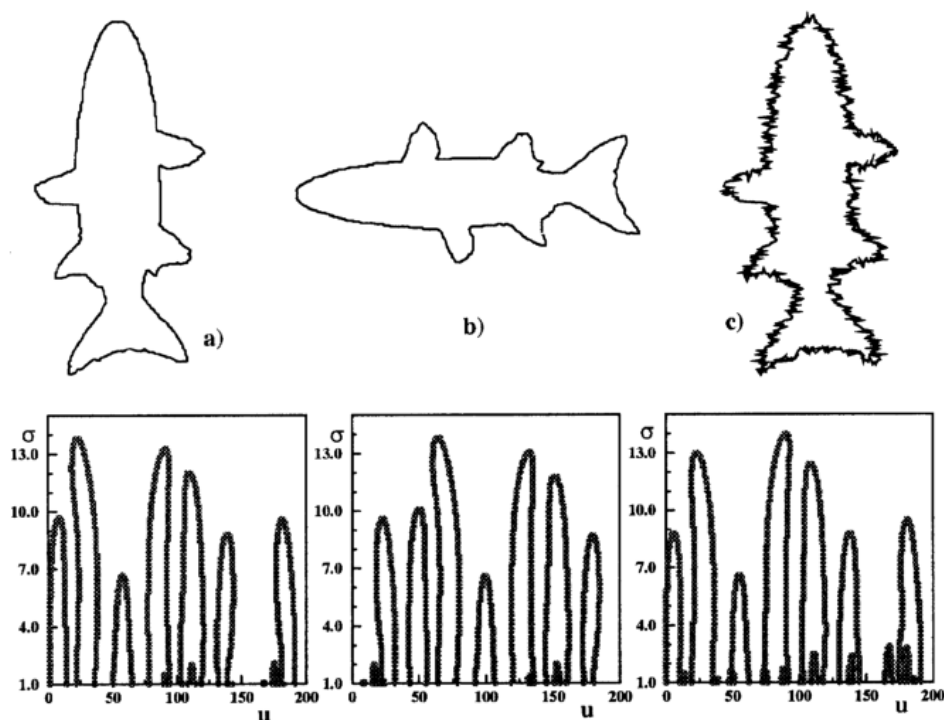
Shape Context ve své původní definici používá Euklidovskou vzdálenost. Ta ovšem nedává dobré výsledky při ohnutí objektu (v kloubu). Ling a Jacobs jako řešení navrhnou použití tzv. Inner Distance (vnitřní vzdálenost). **Inner Distance** je definována jako délka nejkratší cesty vnitřkem objektu (Obr. 9). Při rozpoznávání například obrazů lidí zůstává vzdálenost od ramena k dlani stejná, ať máme ruku napnutou anebo ohnutou v lokti. Takto modifikovaný deskriptor označují autoři jako Inner Distance Shape Context (IDSC).



Obr. 9: Vnitřní vzdálenost zůstává při ohnutí stejná, zatímco euklidovská se mění, zdroj: Ling, Jacobs 2005

CSS. Curvature Scale Space využívá inflexních bodů (Abbasi et al., 1999). Objekt se postupně zjednodušuje a shlazuje použitím Gaussova filtru postupně se zvětšující šířky. V každé použité šířce Gaussova filtru jsou zaznamenávány pozice inflexních bodů na polygonu. Takto vzniká graf, v němž na ose x jsou pozice inflexních bodů křivky a na ose y různé šířky Gaussova filtru. Při určité šířce Gaussova filtru dochází přiblížení dvou

inflexních bodů a jejich zániku, jak je vidět na Obr. 10. Body, v kterých tento zánik nastává, se v grafech projeví jako lokální maxima. Pozice lokálních maxim jsou porovnávány mezi objekty. Podobné tvary mají podobnou pozici lokálních maxim. Tato charakteristika je invariantní vůči změně měřítka a pootočení. Metoda rovněž není příliš citlivá na drobný šum. Optimálním přiřazením lokálních maxim k sobě lze zabezpečit invariantnost vůči volbě počátečního bodu.



Obr. 10: Repräsentace polygonů metodou CSS, zdroj: Abbasi et al., 1999

WARP. WARP je jedna z technik používajících Fourierovu transformaci (Bartolini et al., 2005). Objekty jsou převedeny do frekvenční domény a je použita jejich fázová složka. Nízkofrekvenční koeficienty jsou upraveny tak, aby byla charakteristika invariantní vůči přesunu, měřítku, rotaci a volbě počátečního bodu. Tak jsou normalizovány a poté převedeny zpět inverzní Fourierovou transformací. Poté je provedeno porovnání obou objektů použitím dynamického programování (konkrétně Dynamic Time Warping distance DTW). Metoda podle autorů však přináší horší výsledky než CSS a není invariantní vůči obecné afinní transformaci (Alajlan et al., 2007).

MCC. Multiscale Convexity Concavity Representation (Adamek, O'Connor, 2004) využívá podobných principů jako CSS. Tvar je postupně vyhlazován pomocí Gaussova filtru vzrůstající šířky. V případě konvexních oblastí se body při vyhlazování přesouvají směrem do středu objektu, v případě konkávních oblastí naopak od středu. Změřením relativního

posunu bodu dvou různých úrovní vyhlazení tedy jednoduše zjistíme vlastnosti křivky v okolí bodu. Tyto posuny jsou zaznamenány do trojrozměrného grafu, který zaznamenává posun (kladný, pokud je směrem do původní kontury objektu, záporný mimo konturu objektu), normalizovanou vzdálenost po křivce a šířku Gaussova filtru. Grafy dvou objektů jsou pak porovnány prostřednictvím DTW. Je invariantní vůči lineárním transformacím (translaci, rotaci i změně měřítka) i volbě počátečního bodu. Zároveň není citlivý k některým drobným změnám. Není však invariantní vůči obecným afinním transformacím.

2.4 Srovnání technik

V předchozí kapitole byly popsány některé, nejčastěji používané, míry podobnosti objektů, pomocí kterých se testuje stejnost či podobnost objektů. Zevrubný a jasně strukturovaný přehled přináší Zhang a Lu (2004).

Jednotlivé metody jsou často testovány na databázi MPEG7 CE-Shape-1, která obsahuje 1400 siluet, které spadají do 70 tříd po 20 podobných obrazcích (Latecki et al., 2000). Každý obrazec je porovnán s každým a 40 nejlepších shod je bráno do úvahy. Ideálně by v těchto 40 shodách mělo být všech 20 siluet z dané třídy (měly by být nejpodobnější). Skóre udává součet správných přiřazení vzhledem k nejlepšímu možnému výsledku (20×1400). Výsledky jednotlivých metod na souboru MPEG-7 shrnuje Bai et al. (2010), podle kterých má nejlepší výsledky z výše představených metod MCC, TAR a IDSC, jejichž skóre se pohybuje mezi 85 – 92 %. U turning function uvádějí Tanase a Veltkamp (2005) skóre 54 %.

V následující tabulce (Tab. 1) jsou porovnány dříve představené metody včetně jejich invariancí.

Tab. 1: Výsledky některých metod na MPEG-7 – ✓ano, (✓) volitelně/modifikace, zdroj: vlastní

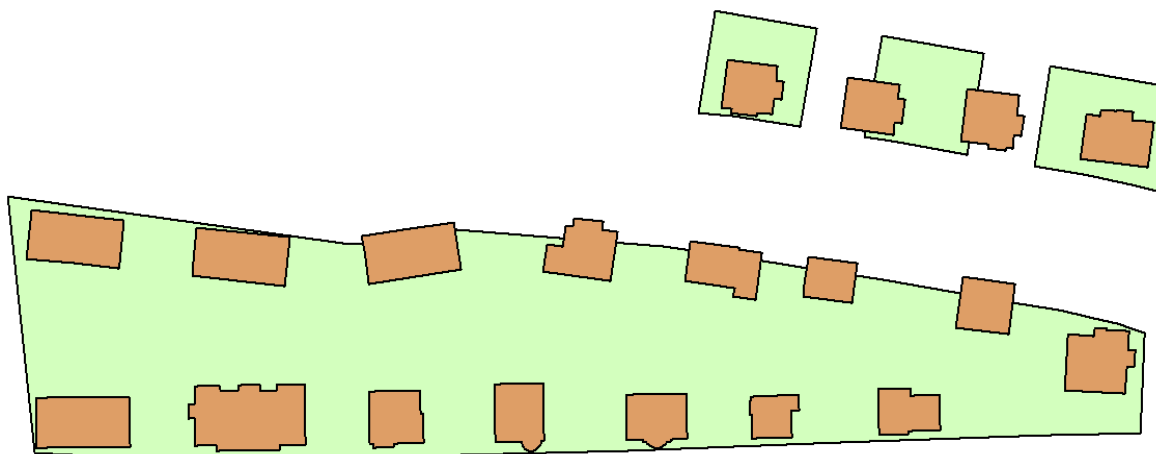
Metoda	Využívá		Invariantní vůči			Použitelný pro		Skóre MPEG-7
	Tvar	Polohu	Translaci	Rotaci	Změně měřítka	Linie	Polygony	
Centroid Distance		✓		✓	✓	✓	✓	
Symmetric Difference	✓	(✓)	(✓)				✓	
Percentage of Overlap	✓	(✓)	(✓)				✓	
Compactness	✓		✓	✓	✓		✓	
Hausdorff Distance	✓	(✓)	(✓)			✓	✓	
Fréchet Distance	✓	(✓)	(✓)			✓	✓	
Angle Histogram	✓		✓	(✓)	(✓)	✓	✓	
Turning Function	✓		✓	(✓)	✓	✓	✓	54 %
Triangle Area Representation	✓		✓	✓	✓	✓	✓	87 %
Inner Distance Shape Context	✓		✓	(✓)	(✓)	✓	✓	85 %
Curvature Scale Space	✓		✓	✓	✓	✓	✓	75 %
WARP	✓		✓	✓	✓	✓	✓	
Multiscale Convexity Concavity Representation	✓		✓	✓	✓	✓	✓	85 %

2.5 Dosavadní použité metody v geoinformatice

Kardinalita. Metody využívané v geoinformatice k přiřazování objektů často naráží na problémy s tzv. kardinalitou, způsobenou rozdílným počtem objektů v obou vrstvách. Kardinalitou se rozumí mohutnost vztahu objektů obou zpracovávaných vrstev. Pokud jeden objekt jedné vrstvy odpovídá jednomu objektu druhé vrstvy, jedná se o kardinalitu 1 : 1. Takový případ je ideální. Může se však stát, že daný objekt z první vrstvy v druhé vrstvě nebude. Pak je tento vztah možno označit kardinalitou 1 : 0. Pokud je více objektů z jedné vrstvy zobrazeno jako jeden objekt v druhé vrstvě, jedná se o kardinalitu N : 1. Může nastat

i situace, kdy M objektů jedné vrstvy bude odpovídat N objektům druhé vrstvy, ale u žádného z nich nepůjde o vztah $1 : 1$. Pak vztah obou množin označujeme jako $M : N$.

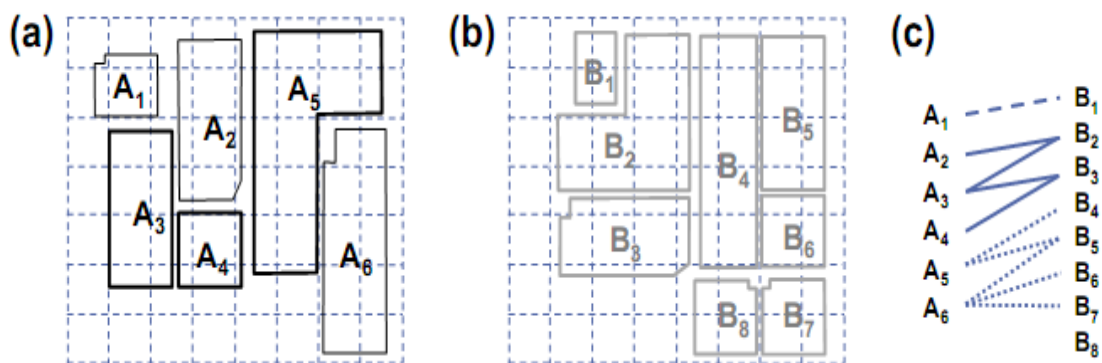
Problémy s kardinalitou mohou být způsobeny agregací při generalizaci mapové vrstvy. Na Obr. 11 je znázorněn příklad takové generalizace. Spodní skupina budov byla agregována do jednoho velkého objektu (kardinalita $N : 1$). Vrchní skupina čtyř podobně velkých budov byla změněna na skupinu tří větších budov (kardinalita $M : N$).



Obr. 11: Objekty v mapových vrstvách s kardinalitou $N : 1$ a $M : N$, zdroj: vlastní

Také Gösseln a Sester (2004) popisují problémy s kardinalitou, které jsou způsobeny odlišnou mírou generalizace vrstev. Například skupina vodních ploch může být v jedné vrstvě reprezentována skupinou polygonů. V druhé, generalizovanější vrstvě, pak může být reprezentována pouze jedním objektem. Pro dosažení co nejlepších výsledků přiřazení, autoři doporučují pomocí testu překrytí vytvoření všech možných agregovaných skupin z objektů v pevně daném okolí v obou zkoumaných vrstvách. Tyto agregované objekty je pak třeba prověřit nějakou z metod shape matchingu, a to každý smysluplně agregovaný objekt z jedné vrstvy s každým smysluplně agregovaným objektem z druhé vrstvy. Nakonec pár agregovaných objektů, který má nejvyšší podobnost, bude označen jako korespondující.

Problémy s kardinalitou řeší i Huh at al. (2011). Jejich postup je znázorněn na Obr. 12. Polygon A1 se překrývá pouze s polygonem B1, a naopak B1 se překrývá pouze s A1, tudíž jsou označeny jako sobě odpovídající. Polygon A2 se překrývá s B2. Ten se však překrývá i s A3 a to se překrývá i s B3, které zase zasahuje do A4. Celé skupiny budou tedy označeny jako odpovídající si (je zde kardinalita $3 : 2$). Aby se předešlo špatnému přiřazení, jsou zde ještě limity týkající se úrovně překryvu objektů. Tedy objekty, které se překrývají jen z malé části, k sobě přiřazeny nebudou.



Obr. 12: Možný postup při problémech s kardinalitou, zdroj: Huh et al., 2011

Seznamme se nyní s některými přístupy detekce tvarů používanými v rámci geoinformatiky. Dobrý přehled o dosavadních metodách používaných v geoinformaticke k nalezení shodných objektů v různých mapových vrstvách přináší Volz (2006). Podle typu dat dělí přístupy na bodové, liniové, plošné a smíšené.

Bodové metody. V těchto přístupech se obecně pro přiřazování odpovídajících si bodů používá vzdálenost mezi nimi a porovnání jejich atributů. Časté je hledání optimálního transformačního klíče mezi dvěma množinami bodů. Jelikož zde není možné porovnat tvar, nemá smysl se nimi v této práci zabývat.

Liniové metody. Liniové prvky velice často tvoří sítě (sít' silnic, říční sít'). V těchto sítích se liniové prvky kříží v místech křížení (křižovatky, soutoky). Lze předpokládat, že u velké části těchto křížení bude možné nalézt odpovídající křížení i ve druhé mapové vrstvě. Podobnost linií je tak převedena na podobnost mračna bodů, představujících křížení (Doytsher et al., 2000).

Walter a Fritsch (1999) používají buffer k nalezení vhodných kandidátů k přiřazení. Linie jsou nejprve spojeny tak, aby segment sítě mezi dvěma body křížení tvořil vždy jednu linii. Ke každé linii je vytvořen buffer určité vzdálenosti. Každý objekt z druhé vrstvy, jež leží alespoň částí ve stanoveném bufferu, je zařazen do seznamu kandidátů k přiřazení.

Z těchto kandidátů jsou na základě jejich atributů (například tvar, velikost) vyřazeni nepravděpodobní kandidáti. Stále však zůstává seznam kandidátů, často s kardinalitou $M : N$. Z těch jsou vytvořeny možné kombinace a statisticky je vybrána kombinace s nejlepší vypočtenou mírou podobnosti. V jejím výpočtu se hodnotí podobnost azimutu, pozice, délky, zakřivení linie a topologického spojení s dalšími prvky. Devogele (2002) naopak v podobné úloze používá k přiřazení kandidátů Fréchet distance.

Plošné metody. Při hledání odpovídajících si ploch je možné použít podobný princip jako u linií. Z listu kandidátů určených bufferem o dané vzdálenosti je vybrán k přiřazení takový objekt, který vykazuje nejlepší shodu na základě určitých parametrů. Používá se plocha objektu (Volz, 2006), vzdálenost centroidů (Volz, 2006, Vivid Solutions, 2003), počet hran (Volz, 2006), podíl překrytí (Volz, 2006, Gösseln a Sester, 2004), symmetric difference (Butenuth, et al., 2007, Vivid Solutions, 2003), Angle Histogram (Butenuth, et al., 2007, Vivid Solutions, 2003), kompaktnost (Gösseln a Sester, 2004, Vivid Solutions, 2003), Fréchet distance (Devogele, 2002) a Hausdorffova míra (Gösseln a Sester, 2004, Vivid Solutions, 2003).

Přiřazení objektů. Některé práce se zabývají také problematikou postupu přiřazování objektů k sobě navzájem. Li a Goodchild (2012) poukazují na to, že klasický greedy algoritmus, který je zpravidla používán, z kandidátského seznamu v každém kroku vždy spojí a odebere jeden pár. Tento postup má nevýhodu v tom, že takto vybraný pár je definitivní, nelze se k němu vracet a případně ho měnit. Proto autoři navrhují použít optimalizaci. Autoři vychází ze snahy minimalizovat sumu nepodobností všech objektů a snaží se každému jednomu objektu přiřadit jeden nebo žádný objekt z druhého datasetu. Tento přístup je však vhodný pouze pro datasety, kde se nevyskytují kardinality 1 : N nebo dokonce M : N.

Lokální transformace. Mezi tvary objektů v kandidátském seznamu z obou mapových vrstev mohou existovat lokální rozdíly vzniklé řadou důvodů. Mohou být posunuté, pootočené či jinak lokálně transformované. Aby se předešlo ovlivnění shape deskriptorů těmito odlišnostmi, někteří autoři (např. Gösseln a Sester, 2004) používají algoritmus, který tyto transformace nejprve odstraní a až poté porovnává oba tvary.

Jedním z nejznámějších algoritmů použitelných k tomuto účelu je algoritmus ICP (Iterative Closest Point) představený Beslem a McKayem (1992). Algoritmus slouží k přiblížení dvou odpovídajících si mračen bodů. Funguje pro dva dostatečně blízké sety bodů, kdy používá translace a rotace k přiblížení jednoho mračna co nejbližšímu druhému modelovému mračnu. Skládá se z několika fází (Rusinkiewicz, Levoy, 2001):

První fází může být výběr bodů. Ten nastává v momentě, kdy jsou mračna bodů velká či porovnáváme datasety neskládající se z bodů (např. křivky). Další fází je přiřazení bodů. Algoritmus každému z vybraných bodů z jednoho datasetu přiřadí nejbližší vybraný bod z druhého datasetu. Zde je zpravidla stanovena nějaká vzdálenost, které se má vyhledávat

nejbližší bod. Pokud v zadané vzdálenosti neexistuje žádný bod z druhého datasetu, pak se tento bod nebere v úvahu.

Poté se odhadnou transformační parametry na základě metody nejmenších čtverců pro rezidua vzdáleností přiřazených bodů. Všechny body se transformují podle zadaných parametrů. Jde o iterativní algoritmus, čili se opakuje, dokud nenastane zadaný počet operací, nebo součet vzdáleností mezi body v přiřazených párech nepoklesne pod stanovenou mez.

2.6 Analýza podobnosti map

Porovnáním celých map se ve své práci věnují Frank a Ester (2006). Mapy a objekty na nich porovnávají ze třech různých hledisek – individuálního, lokálního a globálního.

Individuální hledisko je reprezentováno samotnou podobností tvarů jednotlivých objektů na mapách. To je řešeno pomocí podobnosti turning function (TF) pro každé dva odpovídající si objekty. Hodnota porovnání (nepodobnost) dvou TF je normalizována a posléze je vypočítána průměrná hodnota napříč všemi odpovídajícími si objekty v celé mapě.

V *lokálním hledisku* se zkoumá poloha posuzovaného objektu vůči objektům v jeho okolí, jeho sousedům. K identifikaci sousedů se využívá Voroného diagramů, které jsou podrobně popsány v kapitole 4.3. Sousedé jsou všechny objekty, jejichž Voroného buňky mají společnou hranici s buňkou posuzovaného objektu. Je měřena vzdálenost nejbližších bodů obou objektů u všech sousedů a porovnána se stejnou vzdáleností ve druhé mapě. Po normalizaci a vydělení počtem sousedů se získává pro každý objekt na mapě průměrná změna vzdáleností mezi tímto objektem a jeho sousedy. Tuto hodnotu je možno napříč všemi objekty zprůměrovat a získat tak jednu hodnotu charakterizující celou mapu.

V individuálním i lokálním hledisku je možné brát v úvahu pouze objekty s kardinalitou 1 : 1. Objekty s jinou kardinalitou jsou zachyceny v třetím, *globálním hledisku*. To pracuje s objekty rozdělenými do pevně daných tříd. Je zde předpoklad, že rozdělení objektů do tříd je shodné pro obě vrstvy. Důležitost každé třídy je posuzována podle součtu plochy Voroného buněk jejich objektů. Autoři definují metriku zvanou Voroného entropie, která pracuje s počty segmentů v každé třídě vážené velikostí Voroného buněk. Odlišná hodnota Voroného entropie mezi dvěma mapami je vzniklá odlišnou kardinalitou některých objektů. Vypočteným odlišnostem v třech různých hlediscích je možno dávat různou váhu v konečném výpočtu v závislosti na požadované důležitosti jednotlivých hledisek.

3 MAPY A JEJICH PODOBNOST

Předmětem této práce je porovnání a určení podobnosti map, proto je důležité shrnout možné odlišnosti v mapách. V této kapitole budou stručně popsány různé aspekty ovlivňující tvorbu map. Budou zde diskutovány, jak tyto různé aspekty ovlivňují nebo omezují určování podobnosti různě odlišných map.

3.1 Mapa

Cílem diplomové práce má být porovnání map, vyjdeme proto z její definice. Existuje mnoho různých definic pojmu mapa, pro příklad lze uvést definici z ČSN 730402:

Definice. *Mapa je zmenšený generalizovaný konvenční obraz Země, nebeských těles, kosmu či jejich částí, převedený do roviny pomocí matematicky definovaných vztahů (kartografickým zobrazením), ukazující podle zvolených hledisek polohu, stav a vztahy přírodních, socioekonomických a technických objektů a jevů.*

Uvedená definice naznačuje, jakým způsobem se od sebe různé mapy mohou odlišovat. Chápeme-li mapu jako „zmenšený generalizovaný konvenční obraz...“, je patrné, že se mapy se od sebe mohou odlišovat mírou zmenšení, tj. **měřítkem** i mírou **generalizace**. Z definice vyplývá, že mapa je obrazem „Země, nebeských těles, kosmu či jejich částí“, jednotlivé mapy se od sebe liší také **zobrazovaným územím**. K převodu zobrazovací plochy do roviny se používá „matematicky definovaných vztahů“ vyjádřených formou **kartografického zobrazení**. Různé mapy mohou používat různé typy zobrazení, v závislosti na jejich měřítku, účelu, požadovaném zkrácení či rozsahu zobrazovaného území. Mapa „ukazuje podle zvolených hledisek polohu, stav a vztahy přírodních, socioekonomických a technických objektů a jevů“. Záleží tedy na tvůrci mapy, jaká kritéria pro **výběr objektů a jevů na zemském povrchu** zobrazených do mapy použije.

Odlišnost map. Kromě výše jmenovaných existuje mnoho dalších faktorů, kterými se mapy od sebe odlišují navzájem, a podle kterých je třídíme. Jmenujme ty nejdůležitější z nich:

- a) územní rozsah,
- b) měřítko a míra generalizace,
- c) obsah mapy – výběr prvků,
- d) kartografické zobrazení,

- e) kartografický jazyk, kompozice, barevnost mapy,
- f) forma mapy – mapy analogové, obrazové, digitální,
- g) vznik mapy – mapy původní a odvozené,
- h) metoda vyjádření skutečnosti – mapy analytické, syntetické, komplexní.

Ne všechny tyto faktory jsou kvantifikovatelné a měřitelné, proto nelze v práci obsáhnout analýzu všech výše uvedených kritérií. V dalším textu se tedy budeme zabývat vlivem pouze některých faktorů, a to zejména územním rozsahem, měřítkem, mírou generalizace, obsahem mapy a kartografickým zobrazením. Z hlediska formy se bude jednat výhradně o mapy v digitální podobě.

3.2 Podobnost map

Na podobnost map je možné nahlížet z mnoha různých hledisek. Za podobné je možné označit mapy využívající podobný mapový klíč, mapy podobného měřítko či účelu, mapy znázorňující podobné území či mapy vzniklé podobnou technikou. V této práci se budeme zabývat podobností map na základě geometrické podobnosti jednotlivých objektů v nich zakreslených. Tento přístup vychází z myšlenky, že každý objekt či jev zobrazený v mapě by měl mít svůj předobraz v reálném světě. Pokud je tento objekt zobrazený i na jiné mapě, je možné analyzovat rozdíly ve znázornění objektu. Tento přístup je možné dále abstrahovat na podobnost celých obsahově souvisejících skupin prvků. Lze tak porovnat celé tematické mapové vrstvy (např. vrstvu budov) zobrazené na dvou mapách.

Podmínky určení podobnosti. Aby bylo možno porovnávat mapy tímto způsobem, je nutné, aby existovaly objekty zobrazené na obou mapách současně. Je tedy nutné, aby posuzované mapy splňovaly tyto podmínky:

- a) území zobrazené na obou mapách se alespoň částečně překrývalo,
- b) mapy se alespoň částečně obsahově překrývaly.

Podobnost tematických vrstev. Tematickou vrstvu mapy lze definovat jako množinu logicky či tematicky příbuzných objektů či jevů zobrazených v mapě. Vzájemnou podobnost lze určit pouze u odpovídajících si vrstev – tedy těch zachycujících stejné téma (např. vrstva budov) a shodné území. Podobnost dvou vrstev určují vzájemné vztahy mezi množinami

objektů z obou vrstev, tj. shodný výběr objektů a vzájemná podobnost odpovídajících si objektů.

3.3 Podobnost objektů

Prvotním úkolem při porovnávání vzájemné podobnosti map je nalezení odpovídajících si objektů. Každý objekt zaznamenaný v mapě si v sobě nese různé informace. Na základě porovnání těchto informací mezi různými objekty obou map lze stanovit podobnost dvou objektů z určitého hlediska. Předpokládejme, že podobnost korespondujících si objektů z dvou různých map bude větší než podobnost objektů nekorespondujících. Podobnost (či totožnost) objektů můžeme posuzovat na základě podobnosti jejich:

- a) tvaru,
- b) polohy,
- c) dalších atributů.

Podobnost tvaru. Tvar objektu může být popsán některým z tzv. shape deskriptorů. Tyto deskriptory popisují tvar takovým způsobem, aby je bylo možné mezi sebou snadno porovnat. Různé shape deskriptory byly popsány v kapitole 2.3, konkrétní shape deskriptor turning function bude popsán kapitole 5.

Podobnost polohy. Nejběžnějším vyjádřením polohy objektu je absolutní souřadnice objektu v určitém souřadnicovém systému. Takto vyjádřená poloha objektů udává přesné místo na Zemi, na kterém se objekt nachází. Objekty lze porovnávat podle vzdálenosti polohy na mapě, například prostřednictvím vzdálenosti jejich centroidů. Takový způsob bude řešen v kapitole 6.1.

Alternativně lze použít polohu relativní vůči jiným (zpravidla blízkým) objektům v mapě. To je výhodné v případech, kdy je celá mapa nebo její část posunuta oproti druhé mapě, nebo není možné přímo porovnávat souřadnice z obou mapových vrstev. Další podrobnosti budou uvedeny v kapitole 6.2.

Některé metody používají jak tvar objektu, tak i jeho polohu. Mezi takové metody patří například Hausdorffova míra nebo symmetric difference. Některé zase tvar zjednodušují na úroveň atributu. Do této skupiny lze zařadit podobnost ploch či kompaktnosti objektů.

Podobnost atributů. Porovnání dvou objektů pomocí jejich atributů lze použít pouze tehdy, pokud jsou v obou mapách dostupné stejné typy atributů. Například totožnost budov z dvou map lze snadno určit, jsou-li známá v obou mapách jejich čísla popisná. Atributy mohou být kvalitativní, kvantitativní, mohou to být i kartografické vlastnosti. V případě mapových vrstev nemusí vždy existovat společné atributy objektů, které by posloužily k porovnání. Pozice objektu a jeho tvar v mapě můžeme analyzovat vždy, proto se dále zaměříme právě na podobnost tvaru a polohy.

3.4 Charakter porovnávaných dat

Objekty či jevy jsou na mapách vyjádřeny pomocí bodových, liniových či plošných znaků. Jejich ekvivalentem v rámci vektorových digitálních mapových vrstev je uložení prostorových dat podle prostorové dimenze v bodech, polylineích a polygonech.

Z rešerše vyplynulo, že k prvkům různé dimenze je nutné při analýze jejich podobnosti přistupovat odlišným přístupem. Práce je zaměřena na techniky porovnání tvaru plošných prvků reprezentovaných v mapě polygony. To je výhodné, protože velká část technik shape matchingu byla vytvořena právě pro rozpoznávání 2D tvarů. V případě bodů lze tyto techniky použít pouze na celá mračna bodů, nikoliv na samostatné body, protože nemají tvar, mají nulovou délku i šířku. U liniových prvků lze často efektivně použít zjednodušenou variantu analýzy tvaru a tento přístup je odlišný od přístupu k polygonům. Proto tato problematika v práci není řešena.

Porovnávanými mapovými vrstvami tedy budou digitálně uložené vektorové vrstvy polygonů, zobrazující stejné nebo alespoň částečně se překrývající území, které představují tematicky stejnou skupinu prvků, tedy je možné mezi nimi nalézt navzájem si odpovídající objekty.

3.5 Vliv rozdílných parametrů na určení podobnosti mapových vrstev

V této kapitole bude popsán vliv některých faktorů popsaných v kapitole 3.1 na tvar, polohu a počet zobrazovaných prvků a také to, jak tyto faktory ovlivňují průběh určování podobnosti objektů.

3.5.1 Vliv změny měřítka

Vzhledem k faktu, že objekty jsou v digitálních mapách uloženy bez vazby ke konkrétnímu měřítku, jeho přímá změna při analýze podobnosti nepůsobí větší komplikace. Změna měřítka je však často doprovázena generalizací obsahu mapy. Mapy různých měřítek budou v praxi postiženy různou mírou generalizace tvarů a kvalitativních i kvantitativních charakteristik jednotlivých prvků. Tento fakt bude ztěžovat detekci jejich podobnosti, v praxi se budou jevit jako méně podobné. Čím větší bude rozdíl měřítek, tím větší budou rozdíly v generalizaci obou map. Porovnávané mapy by z tohoto důvodu neměly mít příliš odlišná měřítka.

3.5.2 Vliv generalizace

Dalším faktorem, jehož vliv budeme zkoumat, představuje kartografická generalizace. Z kartografického pohledu existuje jejich několik různých definic, uveďme normu ČSN 730402:

Definice. *Kartografická generalizace spočívá ve výběru, geometrickém zjednodušení a zevšeobecnění objektů, jevů a jejich vzájemných vztahů pro jejich grafické vyjádření v mapě, ovlivněné účelem, měřítkem mapy a vlastním předmětem kartografického zobrazování.*

Kartografická generalizace zahrnuje řadu metod sloužících k výběru, geometrickému zjednodušení či ke kartografické harmonizaci mapových prvků. Následuje přehled nejčastějších metod použitelných na polygony spolu s jejich případným dopadem na tvar, pozici a počet objektů v mapě. Metody použitelné pro jednotlivé polygonové prvky jsou ilustrovány na Obr. 13.

Zhroucení (collapse). Spočívá ve změně prostorové dimenze prvků. Plošný prvek se stává liniovým nebo bodovým. Mění se tvar objektu, poloha se nemění. Jelikož se tato práce zabývá pouze polygonovými vrstvami, není možné tuto změnu postihnout. Objekt v polygonové vrstvě bude po generalizaci chybět.

Posun (displacement). Prvek je posunut na novou pozici, aby se zabránilo konfliktům s jinými prvky. Je změněna poloha objektu.

Kresba přes míru (exaggregation). Prvek je zvětšen, aby byl na mapě zobrazitelný. Poloha zůstává, tvar se zachovává, ale je plošně zvětšen – vzroste plocha i obvod objektu.

Odstranění (elimination). Prvek je odstraněn, protože je zhodnocen jako nevýznamný, či jsou jeho rozměry vzhledem k měřítku mapy zanedbatelné. Objekt v polygonové vrstvě bude chybět.

Zjednodušení tvaru (simplification). Zjednodušení polygonu způsobí zánik detailů v tvaru objektu. Tvar objektu se zjednoduší, obvod objektu se zpravidla zkrátí.

Rotace (rotation). Spočívá v otočení objektu za účelem jeho harmonizace s ostatními změněnými prvky. Je zachována poloha, tvar objektu je potočen.

Rozdělení (split). Způsobí rozdělení plošného prvky na dva prvky. Mění se počet objektů, i jejich tvar.

Operation		Large-scale	Photo-reduced	Small-scale
Collapse	Area-to-point			
	Area-to-line			
	Partial			
Displacement				
Exaggeration	Directional thickening			
	Enlargement			
	Widening			
Elimination				
(Shape) Simplification				
Split				

Obr. 13: Generalizace plošných prvků, zdroj: Zhilin, 2007

Metody použitelné pro skupiny plošných prvků (Obr. 14). Mezi metody použitelné pro celé skupiny plošných prvků patří seskupení (agglomeration), agregace (aggregation), sloučení (amalgamation), rozpouštění (dissolving), spojení (merging). Tyto metody spočívají ve změně počtu i tvaru zobrazených prvků.

Operation	Large-scale	Photo-reduced	Small-scale
Aggregation			
Agglomeration			
Amalgamation			
Dissolving			
Merging			
Relocation			
(Structural) Simplification			
Typification			

Obr. 14: Generalizace skupin plošných prvků, zdroj: Zhilin, 2007

Důsledky generalizace. Některé metody kartografické generalizace mění počet objektů v mapě. Důsledky těchto metod nelze postihnout přístupem naznačeným v kapitole 3.3, tedy porovnáváním jednotlivých objektů mezi sebou. K ošetření těchto metod generalizace by bylo nutné objekty v mapách vhodně agregovat tak, abychom dostali objekty navzájem si odpovídající. To je možné učinit testováním polohy, plochy, přesahu objektů, rozdělením do skupin a následnou agregací, jak navrhuji Huh et al. (2011). Takováto agregace však není předmětem této práce.

Metody, které neovlivňují počet prvků, mohou ovlivňovat polohovou, tvarovou i atributovou podobnost objektů. Vlivům těchto metod generalizace lze čelit vhodným návrhem metody k určování podobnosti tvaru objektů.

Požadované invariance. Použitá metoda by měla být co možná nejméně citlivá vůči jevům způsobených generalizací. Generalizace sebou často přináší změnu tvaru (např. zjednodušení), proto by použitý shape deskriptor neměl být citlivý na detaily tvaru, které mohou být generalizací vyhlazeny. Z výše uvedeného přehledu dopadů generalizace rovněž

vyplývá, že může dojít k translaci, rotaci i změně velikosti objektu. Proto by použitý shape deskriptor měl být invariantní vůči těmto transformacím.

3.5.3 Vliv kartografického zobrazení

Na základě použitého kartografického zobrazení u analyzovaných map lze rozlišit následující situace:

- a) Kartografická zobrazení obou map jsou shodná.
- b) Kartografická zobrazení obou map jsou různá, ale známá.
- c) Kartografické zobrazení alespoň jedné z map není známé.

Shodná zobrazení. Ideální situace nastává, pokud jsou obě mapy ve shodném zobrazení, nejsou vzájemně posunuty, mají stejné měřítko a stočení. Může nastat i situace, kde jsou obě mapy ve stejném zobrazení, ale jedna mapa je vůči druhé posunuta o nějakou adiční konstantu. Pokud je tato konstanta známá, je nutné ji od souřadnic objektů dané vrstvy odečíst. Pokud známá není, je třeba postupovat jako v případě neznámého zobrazení či obě analyzované množiny ztotožnit v těžišti.

Různá zobrazení. Mají-li obě mapy různá zobrazení, jejich souřadnicové systémy nejsou homogenní. Mapy je před započítáním analýz nutné převést do jednotného zobrazení. Použijeme inverzní zobrazovací rovnice pro převod původních pravoúhlých souřadnic na souřadnice sférické, na které v dalším kroku aplikujeme zobrazovací rovnice cílového zobrazení. Tím jsou obě mapy převedeny do shodného zobrazení.

Neznámé zobrazení. Pokud nelze převést zobrazení map na zobrazení společné (např. není známo použité zobrazení), je porovnání podobnosti komplikovanější. U obou map se uplatní vliv kartografického zobrazení, jejich souřadnicové systémy budou heterogenní. V tomto případě odpadá možnost použít k nalezení odpovídajících si objektů jejich absolutní polohu. Pokud je známo alespoň měřítko mapy, je možné použít alespoň relativní polohu vůči objektům v téže mapové vrstvě, jak bude popsáno v kapitole 6.2. Z hlediska tvaru objektů může obecné zobrazení změnit tvar jakýmkoliv způsobem. Běžně používaná zobrazení však mají malá zkreslení, čili jejich vliv na tvar objektů u map velkých měřítek bude minimální. Mapy však mohou mít různé stočení, proto by použitý shape deskriptor měl být invariantní vůči rotaci.

4 ZÁKLADNÍ GEOMETRICKÉ STRUKTURY, POJMY A ALGORITMY

V následující části jsou popsány některé netriviální pomocné struktury a algoritmy, jichž bude využito v dalším textu. Jedná se zejména o definici polygonu, který je základní geometrickou strukturou, kterou se tato práce zabývá. Dále bude popsán pojem metrika, důležitý pro definici funkce stanovující podobnost objektů. V neposlední řadě bude představen Voroného diagram, použitelný pro charakteristiku mapové vrstvy, a Hungarian algorithm, použitelný k optimálnímu přiřazení objektů.

4.1 Polygon a centroid polygonu

Polygony jsou v geoinformatice základní strukturou, pomocí níž jsou zaznamenány plošné prostorové jevy v mapových vektorových vrstvách. O'Rourke (1998) polygon definuje jako část roviny ohraničenou konečným počtem segmentů tvořících dohromady jednoduchou uzavřenou křivku. Segmenty polygonu nazvěme **hranami polygonu**. Hrany označujeme e_i , kde $i \in \{1, 2, \dots, n-1, n\}$, kde n je počet hran polygonu. Každá hrana je úsečka spojující dva body, které jsou nazývány **vertexy** (vrcholy) v_i , kde $i \in \{1, 2, \dots, n-1, n\}$. Pak platí: $e_1 = v_1v_2, e_2 = v_2v_3, \dots, e_i = v_iv_{i+1}, \dots, e_{n-1} = v_{n-1}v_n, e_n = v_nv_1$. Hrany i vertexy jsou číslovány v cyklickém pořadí, po prvku s indexem n následuje prvek s indexem 1.

Jednoduché (simple) polygony musí splňovat tyto podmínky:

1. Průsečíkem každých dvou po sobě následujících hran je právě jeden bod – společný vertex:

$$e_i \cap e_{i+1} = v_{i+1}, \text{ pro všechna } i = 1, 2, \dots, n-1, n$$

2. Hrany, které po sobě nenásledují, nemají žádný společný bod:

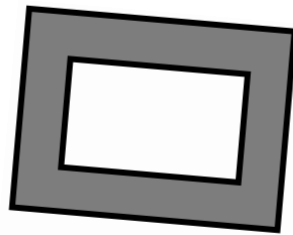
$$e_i \cap e_j = \emptyset, \text{ pro všechna } j \neq i + 1$$

Pokud není splněna podmínka č. 2, jedná se o takzvaný **self-intersecting polygon**, tedy protínající samy sebe (Obr. 15). Pro účely této práce si stanovme podmínku, že posuzované polygony musí být simple, tedy splňovat obě dvě výše popsané podmínky.



Obr. 15: Simple polygon (vlevo) a self-intersecting polygon (vpravo), zdroj: vlastní

Donut polygony. V rámci geoinformatiky je možno se setkat s tzv. donut polygony (polygony s dírou). Ty mohou být definovány pomocí dvou a více tvarů (jednoduchých polygonů), přičemž první polygon definuje vnější hranici objektu a další polygony definují díry (vnitřní hranice) v tomto objektu (Obr. 16). Druhý a další polygony musí ležet kompletně uvnitř prvního polygonu, a aby takto definovaný objekt byl simple, nesmí se žádné dvě hrany navzájem protínat.



Obr. 16: Donut polygon, polygonů s dírou se využívá např. pro znázornění městských bloků s vnitroblokem, zdroj: vlastní

Kvůli faktu, že v některých mapových vrstvách nejsou díry v polygonech (například vnitrobloky budov) znázorněny, některé shape deskriptory s nimi neumí pracovat, a také kvůli zjednodušení výpočtů, jsou donut polygony v mapových vrstvách v rámci této práce zjednodušeny na jednoduché polygony, takže jejich díry nejsou brány v úvahu.

Uložení polygonů ve vektorových vrstvách. Ve vektorových 2D mapových vrstvách bývají polygony uloženy jako seznam bodů, kde první vertex v_0 a poslední vertex v_n jsou totožné. Každý bod je uložen jako uspořádaná dvojice souřadnic. Vertexy jsou číslovány po směru hodinových ručiček. Totožný polygon může být popsán n různými způsoby v závislosti na volbě počátečního vertexu v_0 .

Centroid polygonu. Centroidem polygon (česky geometrické těžiště) se rozumí střední bod polygonu. Použitím centroidu lze pozici polygonu v rovině zjednodušeně vyjádřit prostřednictvím jediného bodu.

Výpočet centroidu. Pro jednoduché (simple) polygony o n vertexech $v_1 = [x_1; y_1]$, $v_2 = [x_2; y_2], \dots, v_{n-1} = [x_{n-1}; y_{n-1}]$, $v_n = [x_n; y_n]$ je jejich centroidem bod $C = [x_C; y_C]$, kde

$$x_C = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$y_C = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

Takto stanovený centroid může v případě nekonvexních polygonů ležet i mimo plochu polygonu.

4.2 Metriky a měření podobnosti

Metrika je abstraktní forma vzdálenosti definovaná pro určitou množinu \mathcal{M} . Je to funkce d , která dokáže určit vzdálenost či nepodobnost dvou libovolných prvků množiny \mathcal{M} . Funkce d společně s množinou \mathcal{M} tvoří tzv. metrický prostor.

Definice. Metrický prostor je dvojice (\mathcal{M}, d) , kde \mathcal{M} je množina a d je reálná funkce na dvojicích prvků z \mathcal{M} , tj. $d: \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$, která pro libovolné $A, B, C \in \mathcal{M}$ splňuje následující axiomy:

1. Axiom nezápornosti: $d(A, B) \geq 0$
2. Axiom totožnosti: $d(A, B) = 0 \Leftrightarrow A = B$
3. Axiom symetrie: $d(A, B) = d(B, A)$
4. Trojúhelníková nerovnost: $d(A, B) + d(B, C) \geq d(A, C)$

V průběhu práce bude určována podobnost či nepodobnost dvou objektů z různých hledisek, na základě různých vlastností objektů. Funkci měřící podobnost dvou objektů můžeme obecně nazvat **mírou podobnosti**. Pokud navíc splňují výše zmíněné axiomy, jedná se o **metriku**.

Z obecného hlediska mají výše zmíněné axiomy pro míru podobnosti plošných objektů své opodstatnění. Místo podobnosti objektů zpravidla měříme míru nepodobnosti d , která pro shodné objekty A, B bude $d(A, B) = 0$. Žádný objekt není více podobný než shodný objekt, proto nepodobnost všech neshodných objektů bude větší než 0. To popisuje první a druhý axiom. Třetí axiom zabezpečuje to, že nezáleží na pořadí analyzovaných objektů. Objekt A je tak podobný objektu B , jako je objekt B podobný objektu A . Čtvrtý axiom (trojúhelníková nerovnost) lze vysvětlit takto: Pokud jsou si podobné objekty A a B a zároveň jsou si podobné objekty B a C , pak by nepodobnost objektů A a C měla být maximálně tak velká jako je součet nepodobností A, B a B, C .

Normalizace. Protože bude v průběhu práce použito více metrik a budou porovnávány mezi sebou, kromě výše uvedených podmínek je třeba, aby výsledek každé metriky byl vždy ze stejného intervalu. Naprostá shoda objektů tak bude označena nulou, zatímco naprostá nepodobnost jedničkou:

$$d(A, B) \in \langle 0; 1 \rangle$$

pro všechny objekty A a B , kde 0 značí shodnost a 1 je nepodobnost.

Canberrská metrika. Výše zmíněné požadavky splňuje Canberrská metrika. Ta je normovanou variantou L_1 Manhattanské metriky:

$$d(A, B) = \sum_{i=1}^n \frac{|A_i - B_i|}{|A_i| + |B_i|}$$

kde $A = (A_1, A_2, \dots, A_n)$ a $B = (B_1, B_2, \dots, B_n)$ jsou vektory.

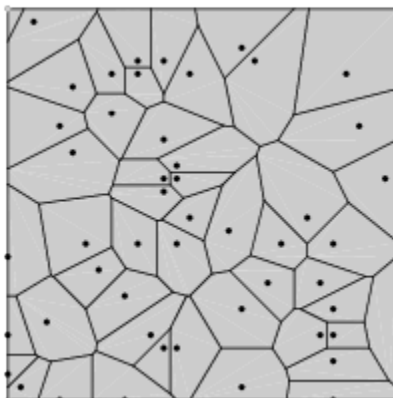
Tato metrika byla zvolena, protože má jednoduchý výpočet a zároveň splňuje požadavek, aby její výsledky byly v intervalu $\langle 0; 1 \rangle$.

4.3 Voroného diagram

Voroného diagram rozděluje rovinu do regionů příslušejících určitým bodům. Necht' B je množina těchto určitých bodů v rovině. Voroného diagram je rozdělení roviny, které přidělí každému bodu $b_i \in B$ oblast $V(b_i)$, tak aby všechny body oblasti $V(b_i)$ byly blíže k bodu b_i než k jakémukoliv jinému bodu z množiny B (Obr. 17). Oblast $V(b_i)$ se nazývá **Voroného buňka**:

$$V(b_i) = \{x; |b_i - x| \leq |b_j - x| \forall j \neq i\}, b_i, b_j \in B$$

Pokud jsou vzdálenosti měřeny pomocí Euklidovské vzdálenosti, pak jsou Voroného buňky v rovině konvexními polygony.



Obr. 17: Voroného diagram nad množinou bodů v rovině, zdroj: Weisstein, 2014

4.4 Přiřazovací problém a maďarská metoda

Přiřazovací problém (**assignment problem**) spočívá v nalezení biparitního přiřazení dvou množin minimalizující jeho cenu. Typická reprezentace tohoto problému je následující:

Mějme n dodavatelů a n úkolů. Každý dodavatel může plnit každý úkol, avšak cena za tuto práci se liší u každého dodavatele i úkolu. Je nutné provést všechny úkoly, každý úkol smí plnit právě jeden dodavatel a každý dodavatel plní právě jeden úkol. Jaké je optimální přiřazení, aby celková cena byla co nejnižší.

Matematicky lze problém definovat takto:

Mějme dvě množiny D a U o stejné velikosti n a jejich váhovou funkci $V: D \times U \rightarrow \mathbb{R}$. Váhová funkce $V(D, U)$ představuje cenu za to, když dodavatel D splní úkol U . Hledáme bijektivní zobrazení $f: D \rightarrow U$ tak, aby celkové náklady $\sum_{d \in D} V(d, f(d))$ byly minimální.

Váhovou funkci V je možné nahradit čtvercovou maticí o velikosti $n \times n$ popisující náklady na přiřazení všech $d \in D$ ke všem $u \in U$. Připomeňme, že obě množiny mají n prvků:

$$V = \begin{bmatrix} V(d_1; u_1) & \cdots & V(d_1; u_n) \\ \vdots & \ddots & \vdots \\ V(d_n; u_1) & \cdots & V(d_n; u_n) \end{bmatrix}$$

Hungarian algorithm. Hungarian algorithm (známý také jako maďarská metoda, Kuhn-Munkres algorithm nebo Munkres assignment algorithm) dokáže vyřešit přiřazovací

problém v polynomiálním čase (Munkres, 1957) Popis fungování algoritmu není triviální, a proto zde není uveden. Po modifikacích je algoritmus schopen fungovat se složitostí $O(n^3)$.

Algoritmus je možné modifikovat pro použití nestejně velký množin D a U o velikostech m a n ($m > n$). V tomto případě bude váhová matice V obdélníková a po přiřazení zůstanou některé prvky z množiny D nepřirazené. V praxi je tato modifikace řešena rozšířením váhové matice V na čtvercovou o velikosti $m \times m$ s tím, že hodnota přidaných prvků bude 0. Stále tak platí podmínka, že celková cena přiřazení $\sum_{a \in A} V(d, f(d))$ musí být minimální.

5 ANALÝZA PODOBNOSTI TVARŮ

Tato kapitola bude zaměřena na určování podobnosti tvarů plošných objektů v mapových vrstvách. Díky vhodným vlastnostem, tj. zejména spolehlivosti a snadnosti výpočtu, bude blíže popsán shape invariant založený na turning function. Dále se budeme věnovat způsobu, jakým se s jeho pomocí dají porovnávat tvary dvou polygonů.

Lidské oko je citlivé na tvar objektu a jeho změny. Toho faktu se využívá při návrhu některých shape deskriptorů, pomocí kterých lze tvary objektů popisovat a porovnávat mezi sebou. Z rešerše vyplývá, že v rámci geoinformatiky jsou používány převážně jednoduché metody určování podobnosti tvarů, a to zejména metody založené na poloze centroidu objektu (vzdálenost centroidů), překrytí dvou objektů (podíl překrytí, symmetric difference), na ploše a obvodu tvaru (kompaktnost) či jednoduché míry podobnosti tvaru, jakou je například Hausdorffova míra. Tyto techniky však nedokáží v některých případech vystihnout přesný tvar objektu. Protože popisují tvar značně zjednodušeně, nejsou tyto metody vhodné pro popis složitějších nekonvexních tvarů, se kterými se můžeme na mapách setkat.

Cílem práce je mimo jiné ověřit vhodnost použití složitějších shape deskriptorů v rámci geoinformatiky. V literární rešerši je popsána řada složitějších shape deskriptorů, u kterých lze zajistit potřebnou invarianci vůči translaci, rotaci a změně měřítka a zároveň vracejí na umělých databázích tvarů dobré výsledky. Vzhledem k rozsahu práce jsem se rozhodl důkladněji zaměřit na invariant *turning function*, který je výpočetně i implementačně poměrně jednoduchý.

5.1 Turning function

Turning function (Arkin et al., 1990) je jedním z jednodušších shape deskriptorů. Při jeho navrhování bylo cílem vytvořit takovou funkci $d(A,B)$, která dokáže vyčíslit míru nepodobnosti mezi polygony A a B za těchto podmínek:

- Míra musí splňovat obecné požadavky kladené na metriku, tedy splňovat axiomy uvedené v kapitole 4.2.
- Musí být invariantní vůči translaci, rotaci a změně měřítka.
- Musí mít jednoduchý výpočet.

- Popis tvaru musí odpovídat intuitivnímu lidskému vnímání tvaru.

S ohledem na tyto podmínky byla navržena tzv. turning function (TF), která popisuje tvar polygonu na základě délek hran a změn směrnic vždy dvou po sobě jdoucích hran polygonu.

Definice. Mějme polygon A reprezentovaný n vertexy $v_i = [x_i; y_i]$. Připomeňme, že hrany i vertexy jsou v polygonu číslovány cyklicky, tedy že $v_{n+x} = v_x$. Uzavřenou křivku, reprezentující vnější hranici polygonu A, nazveme h_A . Na této vnější hranici stanovme počáteční bod $O \in h_A$. Pro libovolný bod $X \in h_A$ definujme s jako délku trajektorie, po které musíme projít, chceme-li se dostat z bodu O do bodu X, přičemž se můžeme pohybovat pouze po křivce h_A proti směru hodinových ručiček.

Každému vertexu v_i odpovídá určitá délka s_i , kterou lze vypočítat jako:

$$s_i = s_{i-1} + \|v_{i-1} - v_i\|_2$$

kde $\|v_{i-1} - v_i\|_2$ je euklidovská vzdálenost dvou po sobě následujících vertexů, tedy délka hrany polygonu. Délka s_1 odpovídající prvnímu vertexu v pořadí je rovna jeho vzdálenost od počátečního bodu O:

$$s_1 = \|O - v_1\|_2$$

Turning function $\Theta_A(s)$ je funkcí, která každému s přiřadí hodnotu α . Pro každou hodnotu s_i náležící libovolnému vertexu platí, že

$$\Theta_A(s_i) = \alpha_i$$

kde α_i je:

$$\alpha_i = \alpha_{i-1} + (\sigma_{j-1} - \sigma_j)$$

kde σ_j je orientovaný úhel hrany $v_i v_{i+1}$ a referenčního směru (např. osy x). Přírůstek úhlů $(\sigma_{j-1} - \sigma_j)$ je kladný, pokud v_{i+1} leží nalevo od hrany $v_i v_{i+1}$ a naopak.

Funkce je nespojitá, po částech konstatní (Obr. 18). Body nespojitosti odpovídají pozici vertexů polygonu. Pro libovolná s platí, že funkční hodnota turning funkce $\Theta_A(s)$ je rovna hodnotě v předcházejícím bodě s_i :

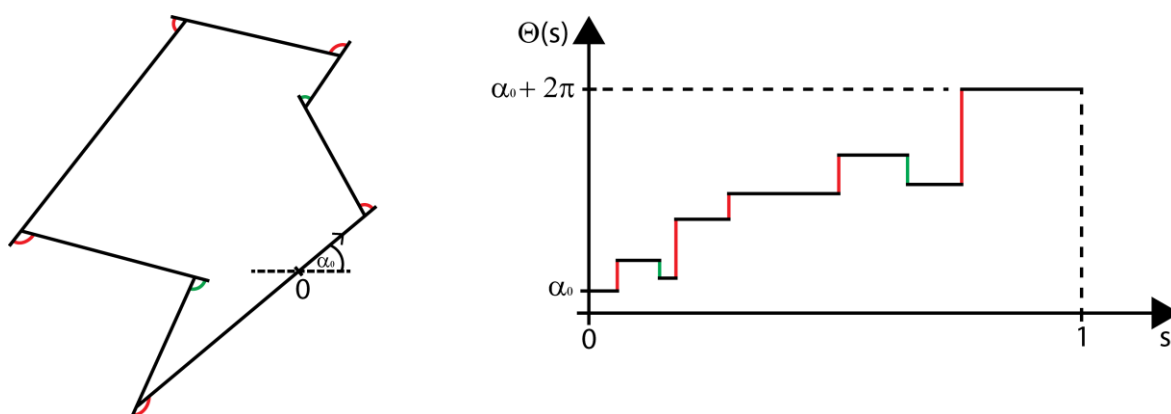
$$s \in (s_i; s_{i+1}): \Theta_A(s) = \Theta_A(s_i) = \alpha_i$$

Velikost polygonu změněna, tak aby jeho obvod byl roven 1, pak tedy $s \in \langle 0; 1 \rangle$. Počáteční úhel je roven orientovanému úhlu mezi počáteční (nultou) hranou a referenčním směrem:

$$\alpha_0 = \sigma_0$$

Z vlastností polygonu vyplývá, že:

$$\alpha_n = \alpha_0 + 2\pi$$



Obr. 18: Polygon a jeho turning funkce, zdroj: vlastní

5.2 Vzdálenost turning funkcí

Podobnost polygonů A, B lze vyjádřit s využitím L^p vzdálenosti jejich turning funkcí jako:

$$\delta_p(A, B) = \|\theta_A - \theta_B\|_p = \left(\int_0^1 |\theta_A(s) - \theta_B(s)|^p ds \right)^{\frac{1}{p}}$$

Takto definovaná vzdálenost turning funkcí je citlivá vůči rotaci polygonů a zároveň vůči volbě počátečního bodu. Pokud dojde ke změně počátečního bodu u jedné z turning funkcí (např. u θ_A), změní se i $\delta_p(A, B)$.

Označme si posun počátečního bodu po hranách polygonu jako t , $t \in \langle 0; 1 \rangle$, pak TF polygonu A lze zapsat jako $\theta_A(s + t)$. Jako Φ si označme úhel rotace mezi oběma objekty. Pokud má být vzdálenost funkcí invariantní vůči rotaci a volbě počátečního bodu, je třeba najít minimum mezi všemi turning funkcemi danými všemi možnými rotacemi (vertikální posun funkce Φ) a počátečními body (horizontální posun t):

$$d_p(A, B) = \left(\min_{\Phi \in \mathbb{R}, t \in \langle 0; 1 \rangle} D_p^{A, B}(t, \Phi) \right)^{\frac{1}{p}},$$

kde:

$$D_p^{A,B}(t, \Phi) = \int_0^1 |\theta_A(s+t) - \theta_B(s) + \Phi|^p ds.$$

Hledáme t a Φ taková, pro která je $D_p^{A,B}(t, \Phi)$ minimální. Tento dvou dimenzionální optimalizační problém lze převést na problém jedno dimenzionální (Arkin et al., 1990, Bayer 2014). Pro zjednodušení použijeme L^2 metriku a označíme:

$$f(s) = \theta_A(s)$$

$$g(s) = \theta_B(s)$$

$$h(t, \Phi) = D_2^{A,B}(t, \Phi)$$

Pak:

$$h(t, \Phi) = \int_0^1 (f(s+t) - g(s) + \Phi)^2 ds$$

$$h(t, \Phi) = \int_0^1 (f^2(s+t) + g^2(s) + \Phi^2 - 2f(s+t)g(s) + 2f(s+t)\Phi - 2g(s)\Phi) ds$$

Výpočet úhlu rotace Φ^* . Lokální minimum Φ^* lze nalézt za podmínky:

$$\frac{\partial h(t, \Phi)}{\partial \Phi} = 0$$

$$\frac{\partial h(t, \Phi)}{\partial \Phi} = \int_0^1 (2\Phi + 2f(s+t) - 2g(s)) ds$$

$$\frac{\partial h(t, \Phi)}{\partial \Phi} = 2\Phi \int_0^1 ds + 2 \int_0^1 (f(s+t) - g(s)) ds$$

$$\frac{\partial h(t, \Phi)}{\partial \Phi} = 2\Phi + 2 \int_0^1 (f(s+t) - g(s)) ds$$

Minimum Φ^* je tedy možno vyjádřit jako:

$$\Phi^* = - \int_0^1 (f(s+t) - g(s)) ds$$

$$\Phi^* = \int_0^1 (g(s) - f(s+t)) ds$$

$$\Phi^* = \int_0^1 g(s) ds - \int_0^1 f(s+t) ds$$

Z vlastností polygonu vyplývá, že:

$$f(s) = f(s-1) + 2\pi$$

Výraz $\int_0^1 f(s+t) ds$ lze pak převést:

$$\int_0^1 f(s+t) ds = \int_t^{1+t} f(s) ds$$

$$\int_0^1 f(s+t) ds = \int_t^1 f(s) ds + \int_1^{1+t} f(s) ds$$

$$\int_0^1 f(s+t) ds = \int_t^1 f(s) ds + \int_1^{1+t} (f(s-1) + 2\pi) ds$$

$$\int_0^1 f(s+t) ds = \int_t^1 f(s) ds + \int_0^t f(s) ds + 2\pi \int_1^{1+t} ds$$

$$\int_0^1 f(s+t) ds = \int_0^1 f(s) ds + 2\pi t$$

Převedení zpátky získáváme:

$$\Phi^* = \int_0^1 g(s) ds - \int_0^1 f(s) ds - 2\pi t$$

Tím se problém zjednodušil pouze na jedno dimenzionální optimalizační problém s proměnnou t :

$$d_2(A, B) = \left(\min_{t \in (0;1)} h(t, \Phi^*) \right)^{\frac{1}{2}}$$

kde:

$$h(t, \Phi^*) = \int_0^1 (f(s+t) - g(s) + \Phi^*)^2 ds$$

$$h(t, \Phi^*) = \int_0^1 \left[(f(s+t) - g(s))^2 + \Phi^{*2} - 2f(s+t)\Phi^* + 2g(s)\Phi^* \right] ds$$

$$h(t, \Phi^*) = \int_0^1 (f(s+t) - g(s))^2 ds + \int_0^1 \Phi^{*2} ds + 2\Phi^* \int_0^1 f(s+t) ds - 2\Phi^* \int_0^1 g(s) ds$$

$$h(t, \Phi^*) = \int_0^1 (f(s+t) - g(s))^2 ds + \Phi^{*2} - 2\Phi^* \left(\int_0^1 g(s) ds - \int_0^1 f(s+t) ds \right)$$

$$h(t, \Phi^*) = \int_0^1 (f(s+t) - g(s))^2 ds + \Phi^{*2} - 2\Phi^* \Phi^*$$

$$h(t, \Phi^*) = \int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2}$$

Vzdálenost turning funkcí tak může být vypočtena pomocí L^2 metriky jako:

$$d_2(A, B) = \left(\min_{t \in \langle 0, 1 \rangle} \int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2} \right)^{\frac{1}{2}}$$

Dále je potřeba nalézt takový posun t ($t \in \langle 0, 1 \rangle$) parametrizující funkci f , který minimalizuje hodnotu funkce $h(t, \Phi^*)$.

Posun t . Turning funkce f a g jsou funkce po částech konstantní, jejich hodnota se mění pouze v místech odpovídajícím vertexům polygon. Počet těchto míst je stejný jako počet vertexů polygonu – m pro funkci f a n pro funkci g . Je zjevné, že $h(t, \Phi^*)$ bude nejmenší v posunu t takovém, při kterém bude koincidovat nějaké místo nespojitosti funkce f s některým místem nespojitosti funkce g . Takových míst bude nejvýše mn . V rámci výpočtu musí být vypočteno $h(t, \Phi^*)$ pro všechna tato t a výsledkem $d_2(A, B)$ bude jejich minimum.

Výpočet $h(t, \Phi^*)$. Samotný výpočet $\int_0^1 (f(s+t) - g(s))^2 ds$ pro konkrétní t je proveden tak, že jsou obě funkce překryty přes sebe a interval $\langle 0, 1 \rangle$ je rozdělen na $m + n$ intervalů, které jsou ohraničeny body, kde dochází k nespojitosti jedné či druhé funkce. V každém z těchto pruhů je vypočtena druhá mocnina rozdílu funkčních hodnot f a g a ta je vynásobena šířkou pruhu. Sečtením těchto násobků za všechny pruhy získáváme $\int_0^1 (f(s+t) - g(s))^2 ds$. Jelikož sčítáme hodnoty $m + n$ pruhů, má tento výpočet složitost $O(m + n)$.

Výpočet optimální rotace $\Phi^* = \int_0^1 g(s) ds - \int_0^1 f(s) ds - 2\pi t$ je jednodušší. Jelikož se $\int_0^1 g(s) ds$ ani $\int_0^1 f(s) ds$ pro konkrétní t neměnní, je možné vypočítat pouze jednou pro celý výpočet $d_2(A, B)$. Pro konkrétní t je tak třeba vypočítat pouze $2\pi t$, což výpočetní složitost nezvyšuje.

Jelikož výpočet $h(t, \Phi^*)$ má složitost $O(m + n)$ a tento výpočet musíme provést pro mn různých posunů t , bude mít celý výpočet podobnosti dvou polygonů $d_2(A, B)$ složitost $O(mn(m + n))$. Celý algoritmus výpočtu bude detailně uveden v kapitole 7.1.1.

Arkin et al. (1991) uvádějí alternativní algoritmus výpočtu $d_2(A, B)$, který má složitost pouze $O(mn \log mn)$. Tento způsob vychází ze zjištění, že pokud t měníme postupně od 0 do 1, mezi dvěma výpočty $h(t, \Phi^*)$ se nemění plocha všech pruhů, ale jenom malé části z nich. Proto pro každý posun t není potřeba počítat plochy všech pruhů. Tento algoritmus je však implementačně náročnější, hodí se tudíž do aplikací, kde je časová náročnost výpočtu kritickým faktorem.

5.3 Invariance vůči translaci a rotaci

Hodnota $d_2(A, B)$ definovaná v předchozí podkapitole je invariantní vůči rotaci. U objektů na mapách však zpravidla jejich orientaci známe. Při posuzování podobnosti objektů nás tak může zajímat i to, zdali jsou objekty stejně natočeny. Pokud jsou objekty na obou mapových vrstvách stejně natočeny, je výhodnější použít variantu, který není vůči rotaci invariantní. Ze vzorce vypustíme člen Φ^* , který značí optimální rotaci:

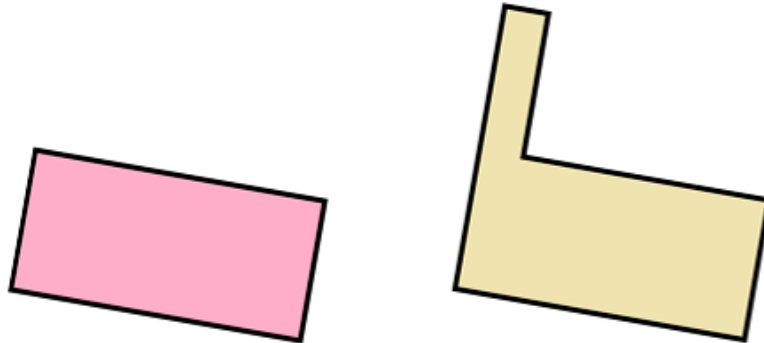
$$d_2^N(A, B) = \left(\min_{t \in (0;1)} \int_0^1 (f(s+t) - g(s))^2 ds \right)^{\frac{1}{2}}$$

a řešíme výše uvedený optimalizační problém.

Další invariance. Protože turning function bere v úvahu pouze tvar objektu, nikoliv jeho pozici, je invariantní vůči translaci. U vertexů polygonu je podstatná jejich relativní, nikoliv absolutní pozice.

Invariance vůči změně měřítka je zajištěna normalizací délky křivky, v opačném případě by byl shape deskriptor naopak velmi citlivý na změnu měřítka. I malá změna měřítka objektu by významně změnila polohu ohybů na nenormalizované křivce a zvětšila tak vzdálenost obou funkcí.

Nevýhoda použití turning function jako shape deskriptoru spočívá v tom, že i plošně malá změna tvaru polygonu může výrazně změnit jeho obvod (Obr. 19). Jelikož je TF normalizována obvodem, výraznější změna obvodu TF zkresluje.



Obr. 19: Přidání přístavku (vpravo) výrazně prodlouží obvod oproti variantě bez něho (vlevo), zdroj: vlastní

Použití L^2 metriky. Jak zmiňují Arkin et al. (1991) lze pro výpočet vzdálenosti turning funkcí použít i jinou metriku. Jako příklad zmiňují L_1 metriku. Při jejím výpočtu však optimální rotaci Φ^* nelze vypočítat přímo pro každou hodnotu t , což zvětšuje složitost algoritmu na $O(n^3 \log n)$ oproti $O(n^2 \log n)$ u L^2 metriky. V některých případech je rovněž L^1 citlivá vůči různé orientaci hran a její použití může přinášet horší výsledky.

6 ANALÝZA PODOBNOSTI POLOHY

Předchozí kapitola pojednává o možnostech porovnávání objektů podle jejich tvaru. Tato kapitola představí některé způsoby, kterými je možné porovnávat objekty na základě jejich polohy, tzv. location similarity. Podobnost na základě polohy lze určovat podle absolutní polohy, tj. porovnáním souřadnic objektů, či na základě relativní polohy, tedy vzájemné polohy vzhledem k jiným objektům v mapové vrstvě. Při určování polohy objekty bude pro zjednodušení objekt nahrazen svým centroidem.

6.1 Absolutní určení polohy

Polohu objektu je možno chápat v absolutním smyslu, tj. poloha objektu vztažená k Zemi určená souřadnicemi. Polohu dvou objektů je možno porovnat prostým porovnáním jejich souřadnic. Funkci pro výpočet polohové podobnosti tak lze realizovat jako Euklidovskou vzdálenost centroidů dvou objektů:

$$d_p(A, B) = \sqrt{(x_{C_A} - x_{C_B})^2 + (y_{C_A} - y_{C_B})^2}$$

kde $C_A = [x_{C_A}, y_{C_A}]$ a $C_B = [x_{C_B}, y_{C_B}]$ představují centroidy dvou porovnávaných objektů A a B. Takovéto určení polohové podobnosti není invariantní vůči posunu objektu, je tedy náchylné vůči nepřesnostem v poloze objektů. Tyto nepřesnosti mohou vzniknout například při převodu souřadnic mezi dvěma souřadnicovými systémy či zobrazeními. V takovém případě je výhodnější použít relativní určení polohy, které bude popsáno dále.

6.2 Relativní určení polohy

Při relativní určení polohy není poloha objektu v souřadnicovém systému určena absolutně formou souřadnic, ale spočívá v určení vztahů mezi posuzovaným objektem a objekty v jeho okolí. Je stanoveno okolí objektu a v tomto okolí jsou zaznamenávány objekty a jejich relativní pozici vůči posuzovanému objektu. Objekty nacházející se v definovaném okolí lze popsat za pomoci jejich relativní polohy od posuzovaného objektu (tj. vzdálenosti, směru), tvaru, plochy či jiných dostupných atributů. Takto popsaná okolí je možné porovnávat mezi sebou napříč objekty a určovat vzájemnou podobnost okolí. Odpovídající objekty dvou mapových vrstev by měly mít podobná okolí.

Vymezení okolí. Důležitou roli hraje stanovení okolí objektu. To lze vymežit několika různými způsoby. Okolí objektu lze vymežit například euklidovskou vzdáleností či pomocí Voroného diagramů. První varianta vytváří okolí konstantní velikosti, druhá varianta okolí s proměnnou velikostí.

6.2.1 Vymezení okolí pomocí euklidovské vzdálenosti

Jednoduché vymezení okolí spočívá v použití euklidovské vzdálenosti. Jako okolí lze charakterizovat objekty nacházející se od zkoumaného objektu do určité mezní vzdálenosti ε . Jako referenční bod každého objektu použijeme jeho centroid. Označme $B(C_P, \varepsilon)$ jako ε -okolí objektu P, definovaného jeho centroidem C_P . Pak do okolí tohoto objektu P spadají všechny objekty Q, jejichž centroidy C_Q leží v ε okolí centroidu C_P .

$$Q \in B(C_P, \varepsilon) \Leftrightarrow \|C_P - C_Q\|_2 < \varepsilon$$

Polární souřadnice. Abychom mohli okolí objektu P nějak charakterizovat, budou všechny okolní objekty Q rozříděny do košů podle jejich vzdálenosti a směru od zkoumaného objektu P. Souřadnice okolních objektů budou transformovány do polárního prostoru souřadnic, v jehož středu se nachází centroid posuzovaného objektu C_P . Pro centroid C_Q každého objektu okolí ($Q \in B(C_P, \varepsilon)$) bude vypočtena vzdálenost r a úhel φ (azimut) od bodu C_P :

$$\forall C_Q \in B(C_P, \varepsilon):$$

$$r_Q = \sqrt{(x_{C_P} - x_{C_Q})^2 + (y_{C_P} - y_{C_Q})^2}$$

$$\varphi_Q = \arctg2(y_{C_P} - y_{C_Q}, x_{C_P} - x_{C_Q}),$$

kde centroidy C_P a C_Q jsou definovány uspořádanou dvojicí pravoúhlých souřadnic:

$$C_P = [x_{C_P}, y_{C_P}], C_Q = [x_{C_Q}, y_{C_Q}]$$

Tímto postupem získáme transformované polární souřadnice pro všechny objekty spadající do okolí posuzovaného objektu P. Pomocí charakteristik těchto objektů dále popíšeme okolí objektu P.

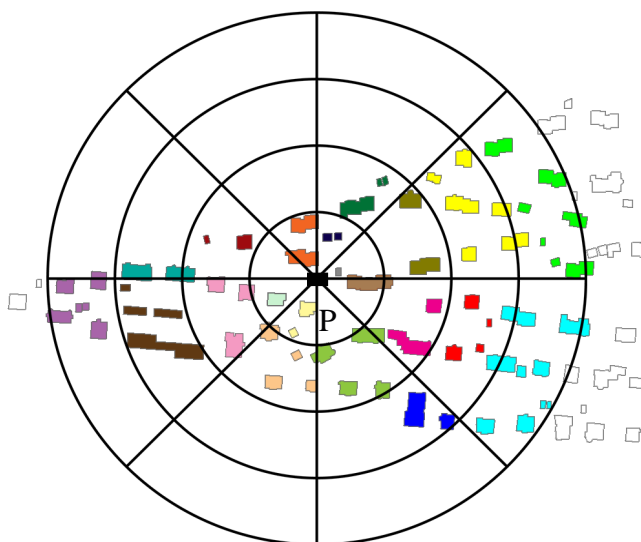
Rozdělení prostoru. Abychom mohli vhodně popsat okolí objektu P, rozdělíme ho na menší sektory. V závislosti na polárních souřadnicích všech objektů Q v okolí objektu P budou tyto

objekty zařazeny do sektorů. Sektor bude definován jako část roviny daná intervaly polárních souřadnic r a φ . Počet sektorů nesmí být příliš malý, aby dokázal dostatečně popsat okolí. Přílišný počet sektorů by naopak celý postup činil náchylným vůči drobným odchylkám v poloze objektů. Jako optimální se jeví 8 výsečí podle azimutu a 4 oblasti podle vzdálenosti, tedy celkem 32 sektorů, viz Obr. 20. Objekt Q patří do sektoru $S_{k,l}$ právě tehdy, když se polární souřadnice jeho centroidu nachází v intervalech, které sektor definují:

$$Q \in S_{k,l} \Leftrightarrow \left(\varphi_Q \in \left((k-1) \cdot \frac{\pi}{4}; k \cdot \frac{\pi}{4} \right) \wedge r_Q \in \left((l-1) \cdot \frac{\varepsilon}{4}; l \cdot \frac{\varepsilon}{4} \right) \right)$$

$$k \in \{1; 2; 3; 4; 5; 6; 7; 8\}, l \in \{1; 2; 3; 4\},$$

kde $S_{k,l}$ je sektor, odpovídající k -tému intervalu azimutu a l -tému intervalu vzdálenosti.



Obr. 20: Okolí objektu P rozdělené do sektorů podle polárních souřadnic, zdroj: vlastní

Tvorba matice okolí objektu. Pro každý sektor bude vypočítána suma všech ploch objektů Q , které do něj spadají. Tak získáme celkem 32 údajů, ze kterých bude sestrojena matice popisující okolí posuzovaného objektu P . Matice M bude mít 8 sloupců, odpovídajícím výsečím podle azimutu, a 4 řádky, odpovídající vzdálenosti od posuzovaného objektu. Každý prvek matice M odpovídá součtu obsahu ploch objektů spadajících do odpovídajícího sektoru.

$$M_{k,l} = \sum_{Q \in S_{k,l}} A_Q,$$

kde A_Q je obsah plochy objektu Q . To platí pro všechny objekty $Q \in S_{k,l}$.

Takto stanovená matice popisuje okolí posuzovaného objektu. Vyjadřuje plochu objektů nacházejících se v určitém směru a vzdálenosti od centroidu. Porovnání dvou objektů může být provedeno prostřednictvím porovnání matic okolí obou objektů. Porovnání těchto matic může být provedeno Canberrskou metrikou, popsanou v kapitole 4.2.

Při posuzování totožného objektu ve dvou různých mapových vrstvách se může stát, že některé okolní objekty nebudou obsaženy v obou těchto vrstvách. Lze předpokládat, že z velké části se bude jednat o malé objekty, které byly v jedné z vrstev vypuštěny kvůli jejich malé velikosti. Protože jsou plošně malé, neovlivní výrazně výše zmíněný postup popisu okolí objektu, který je založen na sumě obsahu ploch.

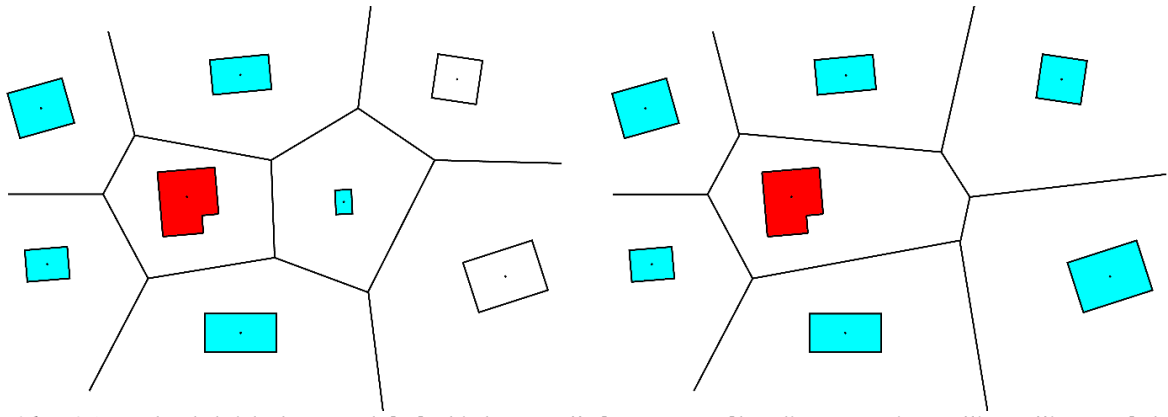
6.2.2 Vymezení okolí prostřednictvím Voroného diagramů

Jinou možností definice okolí objektu je jeho vymezení s použitím Voroného diagramů pomocí přirozených sousedů. Nad centroidy objektů je vygenerován Voroného diagram. Za přirozeného souseda objektu A je poté považován každý objekt B, jehož Voroného buňka má společnou hranu s buňkou objektu A:

$$B \in N_A \Leftrightarrow V(C_A) \cap V(C_B) \neq \emptyset$$

kde $V(C_A)$ a $V(C_B)$ jsou Voroného buňky objektů A a B vzniklé nad jejich centroidy (viz kapitola 4.3) a N_A je množina přirozených sousedů objektu A.

Tento přístup není vhodný pro nalezení navzájem si odpovídajících objektů ve dvou mapových vrstvách. Pokud obě vrstvy totiž nemají stejný počet objektů, budou některé objekty na jedné z mapových vrstev chybět. Nepřítomnost objektu způsobí rozdílné vymezení Voroného diagramů okolo pozice chybějícího objektu (Obr. 21). Při posuzování podobnosti okolních objektů by tudíž byly vybrány do okolí jiné objekty. Totožné objekty z obou vrstev by byly hodnoceny jako nepodobné, protože by jejich okolí bylo významně odlišné.



Obr. 21: Pokud dojde k vynechání objektu, změní se Voroného diagramy i množina přirozených sousedů (modře), zdroj: vlastní

Frank a Sester (2006) však využívají tohoto přístupu pro porovnání mapových vrstev za předpokladů, že již jsou k dispozici nalezené odpovídající si objekty. Z přirozených sousedů jsou vybrány ty, jež se vyskytují v obou mapových vrstvách. U nich je v rámci porovnání celých mapových vrstev hodnocena průměrná změna polohy.

7 NÁVRH ALGORITMU PRO ANALÝZU PODOBNOSTI MAP

V předchozích kapitolách byly popsány některé způsoby stanovení podobnosti objektu, kde byl používán tvar objektu či jeho poloha. Shape deskriptory, jakým je např. turning function, dokáží dobře popsat tvar objektu. Avšak charakteristika objektů v mapách pouze na základě tvaru je v tomto případě nedostatečná, protože v mapách mají objekty často podobné tvary (například obdélníkové půdorysy budov). Popis objektu na základě jejich polohy zase nezohledňuje jejich tvar ani další atributy.

Při vnímání objektů lidským okem se nezaměřujeme pouze na jedno hledisko, ale vnímáme objekt komplexně. Vnímáme jeho tvar, barvu (atributy), pozici v prostoru i jeho vztah k okolním objektům. Podobně při porovnávání objektů v mapách může zvolený přístup kombinovat více hledisek.

Z rešerše vyplývá, že autoři často pro identifikaci objektů používají jejich polohu kombinovanou s jednoduchými charakteristikami tvaru (Volz, 2006, Devogele, 2002, Gösseln a Sester, 2004). Tyto jednoduché shape deskriptory však nemusí dostatečně vystihnout tvar například u složitějších, nekonvexních objektů. Z tohoto důvodu je v této práci zkoumána možnost využití složitějších deskriptorů, konkrétně turning function (TF), k popisu tvaru. Z výše popsaných důvodů bude v práci použita turning function v kombinaci s dalšími metrikami takovým způsobem, aby výsledná metoda byla co nejvíce robustní pro porovnání různých mapových vrstev. Výsledná metrika bude navržena jako kombinace následujících metod.

Podobnost tvaru. Podobnost tvaru bude řešena pomocí podobnosti TF dvou objektů, tak jak byla popsána v kapitole 5. Tvar patří mezi základní charakteristiky objektů.

Podobnost absolutní polohy. U objektů v mapových vrstvách je podstatnou informací jejich poloha. Protože je turning function invariantní vůči translaci, polohu objektů nezohledňuje. Jak bylo uvedeno v kapitole 6.1, nejjednodušší možností porovnání poloh dvou objektů je prosté porovnání souřadnic jejich centroidů. Ač tento způsob může být nevhodný pro některé kombinace mapových vrstev (vrstvy o různých kartografických zobrazeních nebo vrstvy posunuté), v případě přesných mapových vrstev stejného zobrazení se jedná o efektivní a jednoduchý způsob zhodnocení podobnosti objektů. Pozice centroidů dvou odpovídajících si objektů ve dvou přesných mapových vrstvách stejného zobrazení

budou od sebe jen velice málo vzdálené. Je třeba mít na paměti, že centroid objektu žádný způsobem necharakterizuje jeho tvar, proto je samostatné použití podobnosti absolutní polohy nedostatečné. Z těchto důvodů bude tato metrika doplňovat posouzení tvaru objektu pomocí turning function.

Podobnost relativní polohy. Protože podobnost absolutní polohy má svá výše uvedená omezení, bude doplněna podobností relativní polohy podle kapitoly 6.2.1. Na rozdíl od absolutní polohy je možno takovou podobnost použít i na vrstvách s různým kartografickým zobrazením, případně na vrstvách posunutých, kde není možné přímo porovnávat souřadnice objektů.

Podobnost velikosti objektu. Turning function sice dokáže velice dobře popsat tvar objektu, avšak protože je normalizovaná, nezohledňuje žádným způsobem velikost objektu. Malá objekt o obdélníkového půdorysu tak může mít stejnou turning function jako velká kancelářská budova s obdélníkovým půdorysem. Z tohoto důvodu je vhodné podobnost tvaru doplnit rovněž podobností velikosti objektu. Nejjednodušším způsobem popsání velikosti objektu je obsah plochy jeho půdorysu.

Bude navržen vhodný způsob porovnávání podobnosti objektů ze všech těchto hledisek. Finální celková podobnost bude určena kombinací všech hledisek. Celková podobnost bude vypočtena jako vážený součet dílčích metrik podobnosti. Váhy jednotlivých hledisek budou určeny empiricky na základě nejlepších výsledků přiřazení v testovací sadě. Tato celková podobnost bude pak používána k přiřazení objektů k sobě navzájem. Objekty budou přiřazeny k sobě tak, aby celková nepodobnost přiřazených objektů byla minimální.

V následujících podkapitolách bude detailně popsán návrh jednotlivých dílčích metrik a jejich kombinace a způsob přiřazení odpovídajících si objektů pomocí těchto metrik. V závěrečné části návrhu pak bude řešeno porovnání mapových vrstev po nalezení odpovídajících si objektů. Porovnání map bude řešeno jako porovnání dílčích kritérií podobnosti na všech vzájemně odpovídajících si objektech z obou vrstev podle postupu vycházející z práce Frank a Ester (2003), popsané v kapitole 2.6. Před samotným porovnáním je tedy nezbytné nalezení navzájem si odpovídajících objektů v obou vrstvách.

7.1 Dílčí metriky a jejich kombinace

Požadavky na měření podobnosti. Pro každou ze jmenovaných metod (podobnost tvaru, absolutní a relativní polohy a velikosti objektu) musí být stanoven způsob výpočtu míry

podobnosti dvou objektů. Necht' $d(A, B)$ je funkce měřící nepodobnost objektů A a B určitou metodou. Funkce $d(A, B)$ musí splňovat dva základní požadavky. Musí být metrikou a zároveň musí být normalizovaná tak, aby její obor hodnot byl v intervalu $\langle 0; 1 \rangle$, kde 0 značí shodnost a 1 je nepodobnost. Tyto podmínky byly popsány v kapitole 4.2. Budou implementovány tyto podobnostní metriky splňující dané podmínky:

- a) podobnost tvaru $d_s(A, B)$
- b) podobnost absolutní polohy $d_p(A, B)$
- c) podobnost relativní polohy $d_n(A, B)$
- d) podobnost velikosti $d_a(A, B)$

Protože předpokládáme různý vliv dílčích metrik, bude celková podobnost objektu stanovena jako vážený průměr dílčích metrik podobnosti:

$$D(A, B) = \frac{w_s d_s(A, B) + w_p d_p(A, B) + w_n d_n(A, B) + w_a d_a(A, B)}{w_s + w_p + w_n + w_a}$$

Nezáporné váhy w_s, w_p, w_n, w_a jednotlivých dílčích metrik budou určeny na základě provedených analýz účinnosti jednotlivých metrik na testovacích datech. Připomeňme, že jednotlivé metriky jsou mapovány na interval $\langle 0; 1 \rangle$, proto i celková podobnost $D(A, B)$ bude vycházet v tomto intervalu. V následujících podkapitolách bude podrobně popsán návrh všech uvažovaných dílčí metriky.

7.1.1 Analýza tvaru

Metrika posuzující podobnost tvarů dvou polygonů bude reprezentována L^2 vzdáleností jejich turning funkcí, která byla představena v kapitole 5.2. Pro každý polygon bude nejprve třeba vypočítat jeho turning function (TF).

Výpočet úhlů a délek stran pro konstrukci TF. Každé dva sousední vertexy polygonu definují hranu polygonu. Při konstrukci TF jsou důležité jednak normalizované délky hran polygonu, jednak změna směrových vektorů mezi dvěma sousedními hranami. Délky hran jsou normalizované proto, aby všechny TF měly ve směru osy x stejnou délku, jak je diskutováno v kapitole 5.1.

Na základě souřadnic vertexů v_i polygonu tak budou vypočítány směrové vektory a délky pro všechny hrany polygonu:

$$v_i = [x_i, y_i]$$

$$\mathbf{e}_i = (x_{i+1} - x_i; y_{i+1} - y_i)$$

$$i \in \{1; 2; \dots; n - 1; n\}$$

Počátečním bodem O nemusí být shodný s žádným vertexem, může se jednat o libovolný bod ležící na libovolné hraně polygonu. Úsečku mezi počátečním bodem O a prvním vertexem v_1 si označíme jako počáteční hranu e_0 :

$$O = [x_0, y_0]$$

$$\mathbf{e}_0 = (x_1 - x_0; y_1 - y_0)$$

Aby byla zajištěna správná funkce algoritmu a jeho invariance vůči změně měřítka, je nutné délky hran normalizovat tak, aby jejich součet byl u všech objektů roven 1. Směrové vektory hran si označíme \mathbf{e}_i , jejich normalizované délky pak l_i :

$$l_i = \frac{\|\mathbf{e}_i\|_2}{\sum_{i=1}^n \|\mathbf{e}_i\|_2}$$

$$i \in \{0; 1; 2; \dots; n - 1; n\}$$

Vertexy i hrany jsou indexovány cyklicky proti směru hodinových ručiček polygonu. Po vertexu v_n tak následuje vertex v_1 . Pro každé dvě sousední hrany je vypočtena změna úhlu mezi sousedními hranami, která je i změnou funkční hodnoty TF v daném vertexu v_i . K tomu je využíváno skalárního součinu vektorů.

$$\varphi_i = \arccos\left(\frac{\mathbf{e}_{i-1} \cdot \mathbf{e}_i}{\|\mathbf{e}_{i-1}\|_2 \cdot \|\mathbf{e}_i\|_2}\right)$$

V počátečním bodě TF nabývá hodnoty, která je rovna orientovanému úhlu mezi počáteční hranou a referenčním směrem (osa x).

$$\varphi_0 = \arccos\left(\frac{(\mathbf{1}; \mathbf{0}) \cdot \mathbf{e}_0}{\|\mathbf{e}_0\|_2}\right), \text{pokud } (\mathbf{1}; \mathbf{0}) \times \mathbf{e}_0 \geq 0$$

$$\varphi_0 = 2\pi - \arccos\left(\frac{(\mathbf{1}; \mathbf{0}) \cdot \mathbf{e}_0}{\|\mathbf{e}_0\|_2}\right), \text{pokud } (\mathbf{1}; \mathbf{0}) \times \mathbf{e}_0 < 0$$

Konstrukce turning function. Hodnota turning funkce je dána změnou úhlů mezi hranami polygonu. Důležitý je směr otočení, který je určen orientací hrany vůči předchozí hraně. Pokud je hrana nalevo od předchozí, orientaci posuzujeme jako kladnou a hodnotu turning funkce zvětšujeme o daný úhel. V opačném případě je hodnota TF o daný úhel zmenšena. Směr otočení zjistíme pomocí vektorového součinu $\mathbf{e}_{i-1} \times \mathbf{e}_i$. Pokud je kladný, funkční hodnotu zvětšujeme a naopak. Jak bylo uvedeno v kapitole 5.1, TF bude uložena jako lineární seznam bodů nespojitosti odpovídající vertexům polygonu. Každý vertex bude převeden na dvojici hodnot $[s_i; \alpha_i]$, kde s_i odpovídá normalizované vzdálenosti vertexu v_i od počátečního bodu měřené po hranách polygonu proti směru hodinových ručiček a α_i je hodnota TF počítaná jako:

$$\alpha_i = \alpha_{i-1} + \varphi_i, \text{ pokud } (\mathbf{e}_{i-1} \times \mathbf{e}_i) \geq 0$$

$$\alpha_i = \alpha_{i-1} - \varphi_i, \text{ pokud } (\mathbf{e}_{i-1} \times \mathbf{e}_i) < 0$$

$$i \in \{1; 2; \dots; n - 1; n\}$$

Počáteční hodnota α_0 představuje orientovaný úhel první hrany od směru osy x, tj. směrnici první hrany:

$$\alpha_0 = \varphi_0$$

Výše zmíněným postupem jsou vypočteny funkční hodnoty TF ve všech jejich segmentech. Normalizované vzdálenosti vertexů od počátečního bodu měřené po hranách polygonu si označíme s_i a vypočítáme je jako součet délek hran od počátečního bodu po směru hodinových ručiček. Označením l_i rozumíme normalizovanou délku i -té hrany:

$$s_i = s_{i-1} + l_{i-1}$$

Při konstrukci grafu funkce jsou na osu x vynášeny vzdálenosti vertexů s_i a na osu y pak funkční hodnoty α_i . V intervalu mezi dvěma vertexy je TF $\theta(s)$ konstantní:

$$s_i \leq s < s_{i+1}: \theta(s) = \theta(s_i) = \alpha_i$$

Vzdálenost turning funkcí. Výpočet L^2 vzdáleností dvou TF označených $f(s)$ a $g(s)$ pro dva různé objekty bude implementován podle kapitoly 5.2. Bude implementována jak verze invariantní vůči rotaci:

$$d_2(A, B) = \left(\min_{t \in \langle 0; 1 \rangle} \int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2} \right)^{\frac{1}{2}},$$

kde

$$\Phi^* = \int_0^1 g(s) ds - \int_0^1 f(s) ds - 2\pi t,$$

tak i ve verze citlivá na rotaci:

$$d'_2(A, B) = \left(\min_{t \in \langle 0; 1 \rangle} \int_0^1 (f(s+t) - g(s))^2 ds \right)^{\frac{1}{2}}.$$

Výhody a nevýhody obou verzí byly popsány v kapitole 5.3. Připomeňme, že t označuje optimální posun funkce f . Jeho výpočet je nutný, protože není zabezpečena stejná volba počátečního bodu u obou TF. Obdobně Φ^* označuje optimální stočení objektu, tak aby byla míra invariantní vůči rotaci.

Popišme nejprve výpočet optimálního stočení Φ^* , který je důležitý pro verzi invariantní vůči rotaci. Ve verzi závislé na rotaci bude tento člen roven nule. TF je funkce po částech konstantní a je uložena jako seznam bodů, ve kterých mění svoji funkční hodnotu. Tyto body budeme označovat jako body nespojitosti a jsou určeny uspořádanou dvojicí

$$[s_i, \alpha_i], i \in \{0, 1, \dots, n-1, n\}$$

kde n je počet vertexů polygonu. Dvojice $[s_0, \alpha_0]$ odpovídá počátečnímu bodu a počáteční hraně (tj. hrana od počátečního bodu k prvnímu vertexu). Platí, že $s_0 = 0, \alpha_n = \alpha_0 + 2\pi$. Hodnoty s_i jsou uspořádány vzestupně:

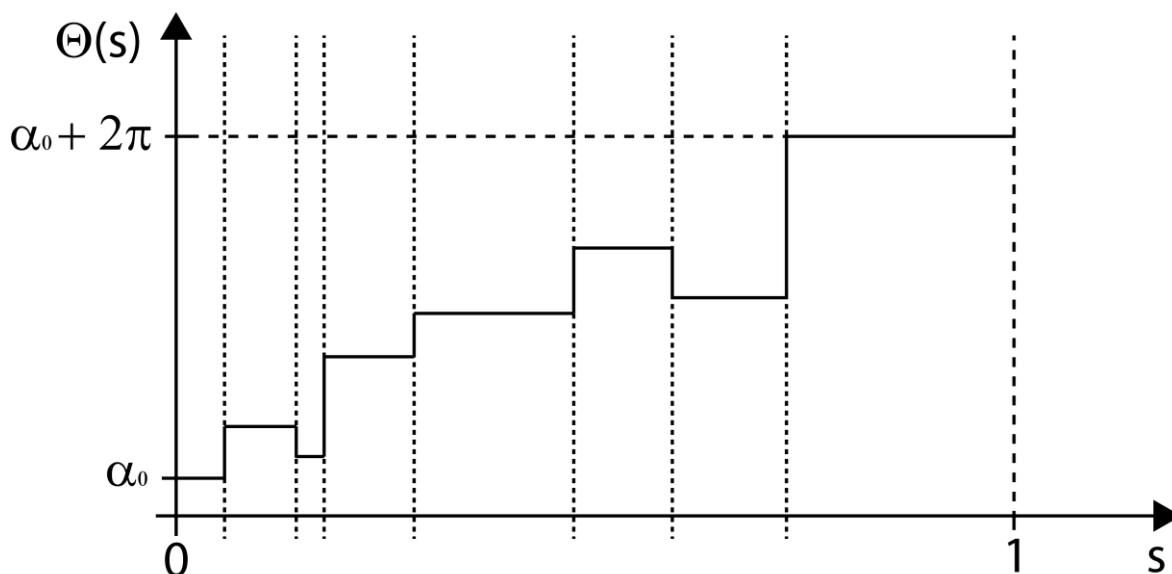
$$0 \leq s_i < s_{i+1} \leq 1$$

Každý α_i tak představuje hodnotu funkce $f(s)$, kterou nabývá na intervalu $s \in \langle s_i; s_{i+1} \rangle$ mezi dvěma sousedními vrcholy. Pro účely výpočtu můžeme vycházet z jednoduché geometrické představy, kdy TF uměle rozdělíme na pásy, která mají tvar obdélníků (viz Obr. 22).

Sečtením plochy těchto obdélníků získáme $\int_0^1 f(s) ds$:

$$\int_0^1 f(s) ds = \sum_{i=0}^n (s_{i+1} - s_i) \alpha_i$$

kde $s_{n+1} = 1$. Jedná se výpočet plochy pod křivkou lichoběžníkovou metodou. Protože jsou v tomto případě sčítány obdélníky, jsou výsledky přesné.



Obr. 22: Rozdělení TF na obdélníkové pásy, zdroj: vlastní

Obdobně je vypočtena i hodnota $\int_0^1 g(s) ds$. Protože hodnoty $\int_0^1 g(s) ds$, $\int_0^1 f(s) ds$ i jejich rozdíl δ jsou pro různá t konstantní, stačí je určit pro každou dvojici objektů pouze jednou. Pro výpočet Φ^* , tak postačí vypočítat pro každé jen hodnotu $\delta - 2\pi t$:

$$\delta = \int_0^1 g(s) ds - \int_0^1 f(s) ds$$

$$\Phi^*(t) = \delta - 2\pi t$$

Tím je zabezpečen výpočet optimálního stočení Φ^* pro verzi invariantní vůči rotaci. Dále bude popsán výpočet integrálu $\int_0^1 (f(s+t) - g(s))^2 ds$ využitý v obou verzích.

Výpočet $\int_0^1 (f(s+t) - g(s))^2 ds$. Výpočet tohoto integrálu pro dané t vychází z metody výpočtu $\int_0^1 f(s) ds$. Opět je použit výpočet plochy pod křivkou lichoběžníkovou metodou. Protože jsou při výpočtu používány body nespojitosti dvou funkcí současně, je nutné tyto body nespojitosti setřídít. K setřídění je využit jednocestný třídící algoritmus merge sort.

Posun funkce. Prvním krokem výpočtu $\int_0^1 (f(s+t) - g(s))^2 ds$ je získání bodů nespojitosti funkce $f(s+t)$, tedy bodů posunutých o t ve směru osy x . Připomeňme, že funkce $f(s)$ je uložena jako seznam uspořádaných dvojic $[s_i, \alpha_i]$ označujících body nespojitosti. Tyto body je třeba posunout o t . Posunuté body nespojitosti si označíme jako dvojice $[s_i^t, \alpha_i^t]$. Při posunu TF o t však nedochází pouze k prostému posunu funkce v horizontálním směru. Jelikož i posunutá TF musí mít stejný definiční obor, tedy $s_i^t \in \langle 0,1 \rangle$, je nutné takové body nespojitosti, které by do tohoto intervalu po posunu nespádaly, do něho převést. Proto x souřadnice bodů nespojitosti budou převedeny takto:

$$s_i^t = s_i - t, \text{ pokud } s_i \geq t$$

$$s_i^t = s_i - t + 1, \text{ pokud } s_i < t$$

Protože si první a poslední bod TF navzájem odpovídají a jejich funkční hodnota se liší o 2π , je při převedení bodu přes nulovou hodnotu nutné navýšit hodnotu tohoto bodu o 2π . Nové funkční hodnoty bodů nespojitosti $[s_i^t, \alpha_i^t]$ budou určeny takto:

$$\alpha_i^t = \alpha_i, \text{ pokud } s_i \geq t$$

$$\alpha_i^t = \alpha_i + 2\pi, \text{ pokud } s_i < t$$

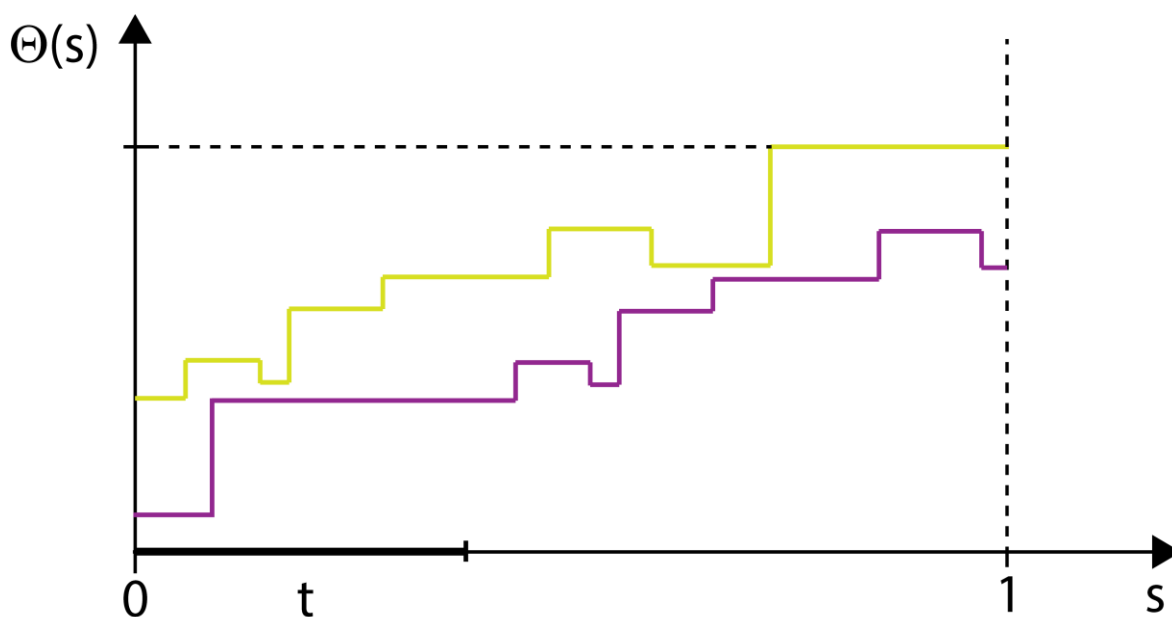
Všechny body nespojitosti budou převedeny podle těchto vzorců. Výsledkem bude opět seznam párů $[s_i^t, \alpha_i^t]$. Jelikož některé body nespojitosti budou převedeny ze začátku funkce na její konec, je vhodné již při výpočtu dbát na nové pořadí bodů a vkládat je buď na začátek, nebo na konec seznamu, aby výsledný seznam byl opět seříděný. Celý výpočet posunutých bodů je znázorněn v Alg.7.1 a na Obr. 23.

Alg. 7.1: Posun funkce

```

1. function shift(f(s),t):
2.     tf_t_points=[] //prázdný seznam bodů posunutě TF
3.     j=0
4.     for i=0 to m:
5.         if si < t:
6.             sit = si-t+1
7.             ait = ai+2*pi
8.             tf_t_points.add([sit, ait]) //přidej bod na konec
9.         else:
10.            sit = si-t
11.            ait = ai
12.            tf_t_points.insert(i, [sit, ait]) //přidej bod
                                                    na pozici i
13.            i = i+1
14.     return tf_t_points

```



Obr. 23: Posun TF o t , zdroj: vlastní

Výše uvedeným způsobem jsme určili body nespojitosti TF $f(s + t)$. Tuto posunutou TF si označíme jako $f'(s)$:

$$f(s + t) = f'(s)$$

$$\int_0^1 (f(s + t) - g(s))^2 ds = \int_0^1 (f'(s) - g(s))^2 ds$$

Výpočet se tak zjednodušil na určení L^2 vzdálenosti mezi dvěma funkcemi. V dalším zápisu si body nespojitosti funkce $f'(s)$ označíme $[s_{1,i}, \alpha_{1,i}]$, zatímco pro body nespojitosti funkce $g(s)$ budeme používat označení $[s_{2,j}, \alpha_{2,j}]$, kde $i \in \{0, 1, \dots, m\}$ a $j \in \{0, 1, \dots, n\}$, m a n je počet vertexů zkoumaných polygonů.

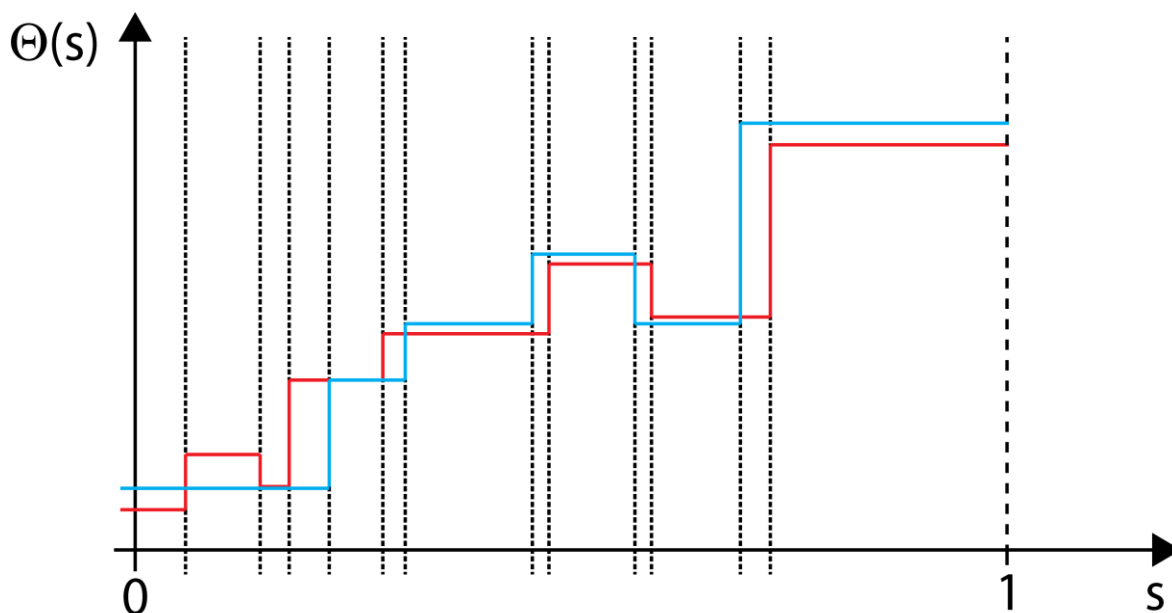
K výpočtu $\int_0^1 (f'(s) - g(s))^2 ds$ bude opět použita *lichoběžníková metoda*. Hranice obdélníkových pásů budou v tomto případě určeny množinou všech bodů nespojitosti obou funkcí, jak ukazuje Obr. 24. Těchto bodů nespojitosti bude dohromady $m + n$. Každý pás je ohraničen dvojicí bodů $s_k s_{k+1}$, které jsou body nespojitosti jedné či druhé funkce:

$$s_k \in \left(\bigcup_{i=0}^m s_{1,i} \cup \bigcup_{j=0}^n s_{2,j} \right)$$

$$k \in \{0; 1; \dots; m + n\}$$

Uvnitř každého pásu je rozdíl funkčních hodnot $f'(s) - g(s)$ konstantní. Pro každý pás vypočteme hodnotu integrálu:

$$\int_{s_k}^{s_{k+1}} (f'(s) - g(s))^2 ds = (s_{k+1} - s_k)(f'(s_k) - g(s_k))^2$$



Obr. 24: Výpočet vzdálenosti dvou TF metodou obdélníkových pásů, zdroj: vlastní

Sečtením získáme hodnotu pro celou funkci:

$$\int_0^1 (f'(s) - g(s))^2 ds = \sum_{k=0}^{q-1} (s_{k+1} - s_k)(f'(s_k) - g(s_k))^2$$

kde q je počet pásů, $q \leq m + n$. Konkrétní provedení tohoto výpočtu je analogií procedury pro jednocestný třídící algoritmus merge sort:

Dvě setříděné množiny bodů nespojitosti jsou spolu spojovány, tak aby výsledkem byla zase setříděná množina. Od nejmenšího $s = 0$ po největší $s = 1$ jsou postupně zpracovávány body nespojitosti v obou množinách. Aktuální body jsou porovnány, je vybrán bod s menším s a je počítána hodnota pásu od předchozího bodu k nově vybranému. Nový bod je označen jako předchozí a zpracování množiny, do které náležel, se posune k následujícímu bodu. Celý postup se opakuje, dokud není dosažen konec na obou množinách. Tento postup je popsán algoritmem 7.2.

Alg. 7.2: Výpočet $\int_0^1 (f(s+t) - g(s))^2 ds$

```
1. function norm2(f(s+t),g(s)):  
2.     s, A, i, j = 0  
3.     f'(s) = shift(f(s),t) //získání posunuté funkce  
4.     while (s<1): //dokud nedojdeš na konec  
5.         if i>=m-1: //pokud na první funkci dojdeš nakonec,  
6.             s1 = 1 //nastav další bod na 1  
7.         else:  
8.             s1 = s1,i // jinak na další bod  
9.             if j>=n-1: //analogicky pro druhou funkci  
10.                s2 = 1  
11.         else:  
12.             s2 = s2,j  
13.         s_old = s //dosavadní s považuj za předchozí  
14.         if s1<s2: //z s1, s2 najdi menší  
15.             s = s1 //bude stanoveno jako nové s  
16.             Δθ = f'(s1,i) - g(s2,j) //vypočten rozdíl hodnot TF  
17.             i=i+1 //posuň na dané funkci o 1  
18.         else: //analogicky pro druhou funkci  
19.             s=s2  
20.             Δθ = f'(s1,i) - g(s2,j)  
21.             j=j+1  
22.         A=A+(s-s_old) * Δθ2 //vypočti (sk+1 - sk)(f'(sk) - g(sk))2 a přičti  
23.     return A
```

Zjištění kritických posunů t . V předchozích odstavcích byl popsán způsob výpočtu výrazu $\int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2}$ pro konkrétní posun t , který je závislý na volbě počátečního bodu, jak bylo popsáno v kapitole 5.2. Protože neexistuje způsob, jak zaručit stejnou volbu počátečního bodu na obou porovnávaných TF, je nutné vypočítat hodnotu $\int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2}$ pro všechna možná t a z nich vybrat minimální. Je zřejmé, že minimum tohoto výrazu bude nastávat při takovém t , při kterém koinciduje některý bod nespojitosti funkce f s některým bodem nespojitosti funkce g . Tato t , při kterých dochází ke koincidenci bodů nespojitosti obou funkcí, budou nazvány jako kritické posuny t_c . Jelikož funkce mají m a n bodů, kde se mění jejich hodnota, bude kritických posunů nejvýše mn . Kritické posuny t_c vznikají při posunutí funkce f o hodnotu rovnou rozdílu x souřadnic dvou libovolných bodů nespojitosti obou funkcí, tj. $(s_{f,i} - s_{g,j})$ pro libovolná i a j .

$$t_c \in \bigcup_{i=0}^{m-1} \bigcup_{j=0}^{n-1} (s_{f,i} - s_{g,j})$$

Konkrétní výpočet množiny všech kritických posunů je řešen jednoduchým algoritmem 7.3.

Alg. 7.3: Výpočet kritických posunů t_c

```
1. function getCritical(f(s),g(s)):  
2.     Tc=[]  
3.     for i=0 to m: //pro všechny body nespojitosti f(s)  
4.         for j=0 to n: //pro všechny body nespojitosti g(s)  
5.             dx=(sf,i - sg,j) //vypočti rozdíl  
6.             if dx<0: //udržuj interval <0;1>  
7.                 dx=dx+1  
8.             Tc.add(dx) // přidej ke kritickým posunům  
9.     return Tc
```


Kompletní výpočet vzdálenosti dvou TF spočívá ve spojení výše popsaných dílčích funkcí. Pro dvě TF budou nejprve vypočteny kritické posuny t_c a zjištěna hodnota $\int_0^1 f(s) ds$ a $\int_0^1 g(s) ds$. Minimální dosažená vzdálenost TF bude inicializována na vysokou hodnotu. Poté bude pro všechny kritické posuny t_c počítán výraz $\int_0^1 (f(s + t_c) - g(s))^2 ds - \Phi^{*2}$. Pokud bude jeho hodnota menší než dosavadní minimum, bude nová hodnota stanovena minimem. Tento postup ukazuje algoritmus 7.4.

Alg. 7.4: Výpočet $d_2(A,B)$ s rotační invariancí

```

1. function TF_distance_rotInvar(f(s),g(s)):
2.     Tc = getCritical(f(s),g(s)) //výpočet kritických posunů
3.     f_Area = getFunctionArea(f(s)) //výpočet  $\int_0^1 f(s) ds$ 
4.     g_Area = getFunctionArea(g(s)) //výpočet  $\int_0^1 g(s) ds$ 
5.     d_min =  $\infty$ 
6.     for t in Tc: //pro každý kritický posun
7.         d=norm2(f(s+t),g(s)) //vypočtena  $\int_0^1 (f(s + t_c) - g(s))^2 ds$ 
8.          $\Phi^* = g\_Area - f\_Area - 2\pi t$  //výpočet optimální rotace
9.         d=d- $\Phi^*$ 
10.        if d<d_min: //nahrazení minima
11.            d_min = d
12.    return d_min

```

Výše popsaný postup je platný pro variantu s rotační invariancí. Pokud chceme variantu závislou na rotaci, není třeba vypočítávat $\int_0^1 f(s) ds$, $\int_0^1 g(s) ds$ ani Φ^* , čímž je postup zjednodušen. Tuto variantu popisuje algoritmus 7.5.

Alg. 7.5: Výpočet $d_2(A,B)$ závislý na rotaci

```

1. function TF_distance_rotVar(f(s),g(s)):
2.     Tc = getCritical(f(s),g(s)) //výpočet kritických posunů
3.     d_min =  $\infty$ 
4.     for t in Tc: //pro každý kritický posun
5.         d=norm2(f(s+t),g(s)) //vypočtena  $\int_0^1 (f(s + t_c) - g(s))^2 ds$ 
6.         if d<d_min: //nahrazení minima
7.             d_min = d
8.    return d_min

```

Takto je nalezena minimální L^2 vzdálenost $d_2(A, B)$ mezi polygony A a B. Ta představuje míru podobnosti mezi těmito polygony. Takto vypočtená L^2 vzdálenost může dosahovat hodnot v intervalu $\langle 0; \infty \rangle$, kde 0 značí shodnost a vysoké hodnoty značí nepodobnost obou tvarů. To nesplňuje požadavek na výsledek v intervalu $\langle 0,1 \rangle$, proto je nutné tento výsledek ještě normalizovat. L^2 vzdálenost $d_2(A, B)$ nedosahuje u běžných tvarů příliš vysokých hodnot, proto je možné použít jednoduchou normalizaci:

$$d_s(A, B) = \frac{d_2(A, B)}{1 + d_2(A, B)}$$

Tím je stanovena míra podobnosti tvaru, která splňuje podmínky určené v kapitole 4.2, tedy je metrikou a zároveň je její výsledek v intervalu $\langle 0,1 \rangle$.

7.1.2 Analýza absolutní polohy

Další metrikou navrženou za účelem nalezení odpovídajících si objektů dvou mapových vrstev je metrika založená na absolutní poloze objektů. Funkci pro výpočet absolutní polohové podobnosti d_p lze nejnázne realizovat jako euklidovskou vzdálenost centroidů dvou objektů, jak bylo popsáno v kapitole 6.1. Tento způsob bude velmi účinný na přesných mapových vrstvách, je však nepoužitelný na vrstvách posunutých či v případech, že souřadnice objektů nepůjdou přímo porovnat mezi sebou. Tento způsob není ze své podstaty invariantní vůči translaci.

Jednou z možností, jak euklidovskou vzdálenost mapovat na interval $\langle 0; 1 \rangle$, je stanovení maximální vzdálenosti d_{max} , při které budou objekty hodnoceny už jako absolutně nepodobné. Euklidovská vzdálenost objektů bude touto mezní vzdáleností vydělena a tím převedena na interval $\langle 0; 1 \rangle$. Při jejím překročení bude d_p stanovena na 1, tedy absolutní nepodobnost. Metrika bude stanovena takto:

$$d_p(A, B) = \frac{\|C_A - C_B\|_2}{d_{max}}, \text{ pokud } \|C_A - C_B\|_2 < d_{max}$$

$$d_p(A, B) = 1, \text{ pokud } \|C_A - C_B\|_2 \geq d_{max},$$

kde $\|C_A - C_B\|_2$ je euklidovská vzdálenost centroidů objektů A a B.

Funkce $d_p(A, B)$ je metrikou, splňuje všechny požadavky stanovené v kapitole 4.2. Určení maximální vzdálenosti d_{max} je závislé hlavně na polohové přesnosti posuzovaných mapových vrstev. Protože v této práci budou testovány mapové vrstvy budov o referenčním měřítku 1 : 10 000 a větším, bude d_{max} stanovena na 50 m. Posunutí objektu o 50 m v mapových vrstvách o tomto měřítku odpovídá posunutí o 5 mm na mapě, což lze považovat za podstatnou chybu.

7.1.3 Analýza relativní polohy

Analýza relativní polohy spočívá v určení vztahů mezi posuzovaným objektem a objekty v jeho blízkém okolí. Důležitou roli hraje stanovení okolí objektu, které lze vymezit několika různými způsoby. Dva z nich byly řešeny v kapitole 6.2. První metodou bylo stanovení fixní vzdálenosti od posuzovaného objektu, dalším způsobem bylo vymezení okolí pomocí

přirozených sousedů vycházejících z Voroného diagramů. Objekty nacházející se v definovaném okolí lze za pomoci jejich relativní polohy, tvaru, plochy či jiných dostupných atributů popsat a navzájem porovnávat.

Cílem této podkapitoly je navrhnout metriku zakládající se na relativní poloze objektů, která bude použitelná pro stanovení podobnosti dvou objektů za účelem nalezení odpovídajících si objektů ve dvou mapových vrstvách. Jak bylo popsáno v kapitole 6.2.2, vymezení okolí pomocí přirozených sousedů není pro takovou aplikaci vhodné. Voroného diagram se může u obou vrstev výrazně lišit, což způsobuje výběr odlišného okolí. To by pak šlo porovnávat jen velmi omezeně. Proto okolí stanovíme metodou fixní vzdálenosti.

Budeme postupovat způsobem popsaným v kapitole 6.2.1. Prvním krokem je určení vzdálenosti ε určující okolí objektu. Při určení vzdálenosti ε by měla být zohledněna hustota objektů v mapové vrstvě, průměrná velikost objektů a očekávaná polohová chyba. Jelikož budou testovány mapové vrstvy obsahující budovy v městské a příměstské zástavbě, velikost okolí ε bude stanovena na 200 m. Takto velké okolí obsahuje dostatek budov k dobré charakteristice okolí a zároveň je dostatečně malé, aby rozdíly v ploše objektů v jednotlivých sektorech byly signifikantní.

Objekty v okolí budou roztrženy do 32 sektorů podle vzdálenosti a azimutu jejich centroidu vzhledem k centroidu posuzovaného objektu. Vzorce pro výpočet azimutu a vzdálenosti byly uvedeny v kapitole 6.2.1. Rovněž v této kapitole byly uvedeny vzorce pro rozdělení objektů do sektorů, ze kterých budeme vycházet. V takto definovaných sektorech budou sečteny obsahy ploch příslušných objektů a uloženy do matice M . Výslednou matici a způsob její tvorby ilustruje

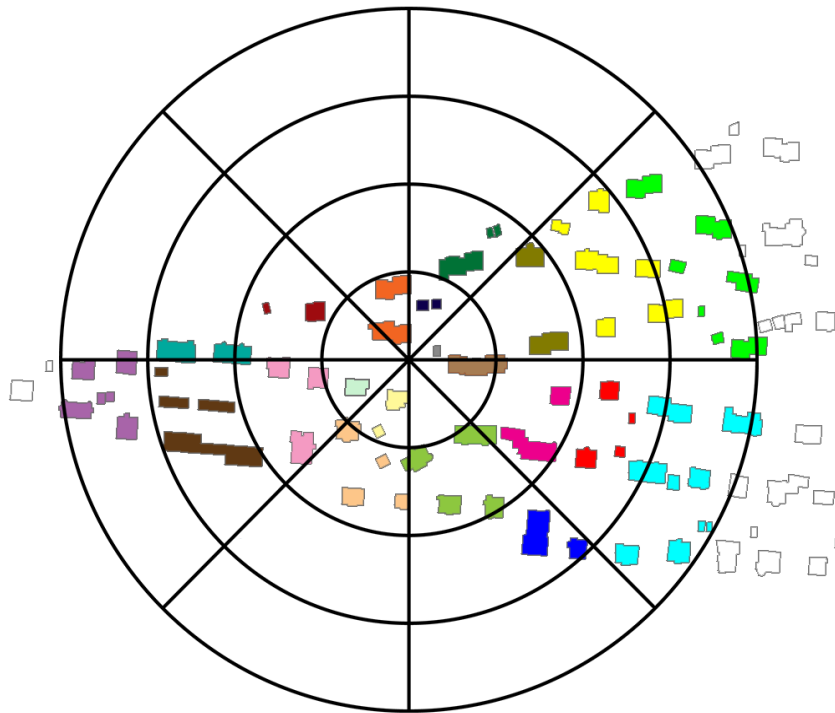
Obr. 25.

Pro každý objekt tak bude získána matice M . Při posuzování podobnosti konkrétních objektů A a B pak budou získané matice M_A a M_B typu 4×8 mezi sebou porovnány.

Porovnání matic. Podobnost matic M_A a M_B bude počítána Canberrskou metrikou, zmíněnou v kapitole 4.2:

$$d_n(A, B) = \frac{1}{32} \sum_{k=1}^8 \sum_{l=1}^4 \frac{|M_{A_{k,l}} - M_{B_{k,l}}|}{M_{A_{k,l}} + M_{B_{k,l}}}$$

Jelikož každý člen $\frac{|M_{A,k,l} - M_{B,k,l}|}{M_{A,k,l} + M_{B,k,l}}$ splňuje podmínky metriky a výsledku v intervalu $\langle 0; 1 \rangle$, splňuje tyto podmínky i jejich průměr.



$r_i[m] \setminus \varphi_i$	$\langle 0; \frac{\pi}{4} \rangle$	$\langle \frac{\pi}{4}; \frac{\pi}{2} \rangle$	$\langle \frac{\pi}{2}; \frac{3\pi}{4} \rangle$	$\langle \frac{3\pi}{4}; \pi \rangle$	$\langle \pi; \frac{5\pi}{4} \rangle$	$\langle \frac{5\pi}{4}; \frac{3\pi}{2} \rangle$	$\langle \frac{3\pi}{2}; \frac{7\pi}{4} \rangle$	$\langle \frac{7\pi}{4}; 2\pi \rangle$
$\langle 0; 50 \rangle$	grey	dark blue	orange		light green	yellow		brown
$\langle 50; 100 \rangle$	olive	dark green		dark red	pink	orange	light green	pink
$\langle 100; 150 \rangle$	yellow			teal	brown		blue	red
$\langle 150; 200 \rangle$	green				purple			cyan

Obr. 25: Tvorba matice M součtem obsahů ploch v sektorech, zdroj: vlastní

7.1.4 Analýza velikosti objektu

Protože turning function je navržena jako invariantní vůči měřítku, je vhodné popis tvaru doplnit popisem velikosti objektu. Velikost objektu může být popsána různými způsoby, např. jeho rozměry či plochou. Nejjednodušší variantou je použití obsahu ploch objektů. Podobnost dvou obsahů S_A , S_B objektů A , B bude opět počítána Canberrskou metrikou:

$$d_a(A, B) = \frac{|S_A - S_B|}{S_A + S_B}$$

Takto stanovená vzdálenost objektů splňuje jak požadavky na metriku, tak je zároveň mapovaná na interval $\langle 0; 1 \rangle$.

7.2 Nalezení odpovídajících si objektů

Dosavadní část návrhu se zabývala způsoby, kterými lze posoudit podobnost dvou objektů v mapové vektorové vrstvě. Jedná se o podobnost tvaru, plochy, pozice v souřadnicovém systému a podobnost okolí objektu. V další části návrhu bude těchto metod využito k nalezení odpovídajících si objektů ve dvou mapových vrstvách. Dílčí metriky budou agregovány do jedné obecné metriky podobnosti objektů. Všechny dílčí metriky budou vyzkoušeny jak samostatně, tak i ve formě jejich váženého průměru. Bude navržen způsob, jak na základě těchto metrik přiřadit objekty k sobě navzájem tak, aby celková nepodobnost byla co nejmenší. Protože zejména v případě mapových vrstev s mnoha objekty není možné z hlediska výpočetní náročnosti porovnávat každý objekt s každým, bude navržena metoda pro předvýběr objektů, které budou spolu navzájem testovány.

7.2.1 Matice podobnosti a nastavení vah

Předpokládejme, že v jedné mapové vrstvě je m objektů, které chceme přiřadit k n objektům druhé mapové vrstvy. Je zřejmé, že pokud $m \neq n$, pak ne všechny objekty budou nakonec přiřazeny. Aby objekty mohly být přiřazeny, je nutné posoudit jejich vzájemnou podobnost. Bude tedy posuzována podobnost každého z m objektů s každým z n objektů druhé vrstvy.

Podobnostní matice. Výsledky těchto porovnání budou uloženy v matici podobnosti D_z o rozměrech m, n pro každou dílčí metriku z :

$$D_z = \begin{pmatrix} d_z(1; 1) & \cdots & d_z(1; m) \\ \vdots & \ddots & \vdots \\ d_z(1; n) & \cdots & d_z(m; n) \end{pmatrix}$$

kde m je počet objektů v první vrstvě, n je počet objektů v druhé vrstvě, z je konkrétní metrika výpočtu podobnosti (tvar, pozice, ...) a $d_z(i; j)$ je z podobnost i -tého objektu z první vrstvy s j -tým objektem druhé vrstvy. D_z jsou dílčími maticemi podobnosti vzniklými na základě různých hledisek podobnosti. Tato práce obsahuje celkem 4 dílčí matice – matice podobnosti tvaru D_s , absolutní polohy D_p , relativní polohy (okolí) D_n , a velikosti D_a .

Agregace dílčích metrik. Výslednou matici podobnosti D definujeme jako vážený průměr dílčích matic D_z .

$$D = \frac{w_s D_s + w_p D_p + w_n D_n + w_a D_a}{w_s + w_p + w_n + w_a}$$

kde s značí tvar, p absolutní polohu, n relativní polohu a a velikost; D_t , D_n , D_p a D_a jsou dílčí matice podobnosti D_z a w jsou jejich váhy.

Volba vah. Optimální volba vah bude stanovena experimentálně, viz kap. 9.4. Pro testovací data budou testovány různé kombinace vah w . Po přiřazení (viz další podkapitola) bude sledován počet správně přiřazených objektů při různých kombinacích vah.

Volbu vah je nutno pro každý případ posuzovat samostatně. Například pokud jsou očekávány polohové nepřesnosti, či posunutí objektů, je dobré minimalizovat vliv podobnosti polohy. Naopak pokud mají data vysokou polohovou přesnost, význam polohové podobnosti naroste. Pokud mají vrstvy výrazně rozdílnou míru generalizace, nejde se spoléhat na podobnost tvaru objektů. Z tohoto důvodů budou testy probíhat na různých kombinacích mapových vrstev o odlišných parametrech – mapy stejných měřítek, odlišných měřítek, mapy s různou prostorovou přesností. Výsledné hodnoty však nelze považovat za univerzální, k tomu bylo by nutné provést širší testování pro různé typy map různých měřítek.

7.2.2 Přiřazení odpovídajících objektů

Matice D o velikosti m , n znázorňuje vzájemnou nepodobnost m objektů první vrstvy vůči n objektům z druhé vrstvy. Pokud si jsou k -tý objekt první vrstvy a l -tý objekt druhé vrstvy navzájem podobné, pak hodnota $D_{k,l}$ bude blízká nule. Naopak, pokud jsou si nepodobné, bude se blížit k hodnotě 1. Je pravděpodobné, že počet objektů v obou vrstvách bude odlišný, tedy $m \neq n$. K určení totožných objektů je potřeba najít takové přiřazení objektů, aby celková nepodobnost přiřazených objektů byla co nejmenší. V tomto případě se tedy jedná o bipartitní přiřazení minimalizující sumu ohodnocení vzájemně odpovídajících objektů. Toto přiřazení bude řešeno pomocí Hungarian algorithm, který je představen v kapitole 0. Nepodobnost dvou objektů $D_{k,l}$ bude chápána jako cena přiřazení těchto objektů k sobě a bude hledáno takové přiřazení s nejmenší sumou nepodobností. Algoritmus při porovnání dvou sad objektů o nesterélné velikosti přiřadí všem objektům z méně početné vrstvy některý z objektů z početnější vrstvy. Objektům z početnější vrstvy přiřadí buď objekt z méně početné vrstvy, nebo žádný objekt. Vzniklé vazby budou mít tedy charakter 1 : 1 nebo 1 : 0.

7.3 Procházení datasetu

Porovnávané mapové vrstvy mohou být značně rozsáhlé. Asymptotická složitost porovnávání objektů mezi sebou každý s každým je $O(n^2)$ pro stejně velké datasety o n objektech. V rámci jednoho porovnání musí být proveden výpočet jednotlivých metrik. Výpočetně nejnáročnější z nich je analýza tvaru, která má složitost $O(p^3)$ pro dva polygony o p vertexech, jak bylo řešeno v kapitole 5.2. Proto celková složitost určení podobnostní matice D je $O(n^2p^3)$.

Další výpočetně náročnou částí je použití Hungarian algorithm k přiřazení objektů. Ten i po úpravách má asymptotickou složitost $O(n^3)$. Vzhledem k této časové náročnosti algoritmu není vhodné porovnávat každý objekt s každým.

Datasety je vhodné rozdělit a analyzovat po částech. Protože předpokládáme, že oba datasety jsou ve stejném souřadnicovém systému, použijeme k jejich rozdělení právě polohu centroidů objektů. Nejprve je nutné zjistit rozsahy území zobrazeného na obou mapových vrstvách a na jejich základě určit území společné pro oba datasety. Celé společné území rozdělíme do čtvercových dlaždic o stejné délce hrany. Její délka je závislá hustotě na objektů v mapové vrstvě. V praxi pro svižný běh algoritmu by v jedné dlaždici nemělo být více než 50 objektů.

Výpočet délky hrany t každé dlaždice může být založen na daném přibližném průměrném počtu objektů v dlaždici. Délka hrany dlaždice t na celkovém území o rozměrech dx a dy je možno vypočítat takto:

$$t = \sqrt{\frac{dx \cdot dy}{\frac{P_{total}}{P_{avg}}}},$$

kde $dx = x_{max} - x_{min}$, $dy = y_{max} - y_{min}$, P_{total} je celkový počet objektů a P_{avg} je požadovaný průměrný počet objektů v dlaždici. Protože v rámci těchto dlaždic bude probíhat celý výpočet, je nutné, aby délka hrany dlaždice t byla větší nebo rovna velikosti okolí ε stanoveného z kapitole 7.1.3. Druhou možností je fixní stanovení velikosti dlaždice t , kde je možné použít takovou velikost dlaždice, jaká je stanovená velikost okolí ε .

Území pak rozdělíme do čtvercových dlaždic o délce hrany t , kde každá je vymezená intervalem x a y souřadnic.

$$x \in \langle x_{min} + a \cdot t; x_{min} + (a + 1) \cdot t \rangle$$

$$y \in \langle y_{min} + b \cdot t; y_{min} + (b + 1) \cdot t \rangle$$

$$a \in \langle 0; \frac{dx}{t} \rangle \cap Z_0^+$$

$$b \in \langle 0; \frac{dy}{t} \rangle \cap Z_0^+$$

Všechny objekty na společném území obou mapových vrstev pak budou přiřazeny do dlaždic v závislosti na souřadnicích jejich centroidů. Objekt A bude přiřazen do dlaždice $T_{a,b}$ právě tehdy, když se jeho centroid bude nacházet v území příslušející této dlaždici.

$$A \in T_{a,b} \Leftrightarrow$$

$$\left(\left(x_{C_A} \in \langle x_{min} + a \cdot t; x_{min} + (a + 1) \cdot t \rangle \right) \wedge \left(y_{C_A} \in \langle y_{min} + b \cdot t; y_{min} + (b + 1) \cdot t \rangle \right) \right)$$

kde x_{C_A} a y_{C_A} jsou souřadnice centroidu objektu A.

Přiřazení objektů do dlaždic bude využito k omezení množství potřebných porovnání. Pro každý objekt A spadající do dlaždice $T_{a,b}$ z první vrstvy bude testována podobnost pro všechny objekty B z druhé vrstvy, jejichž pozice je dostatečně blízká pozici objektu A. Testované objekty B musí spadat do téže, případně jedné ze sousedních dlaždic:

$$\text{Jestliže } A \in T_{a,b}, \text{ pak } B \in \bigcup_{i=a-1}^{a+1} \bigcup_{j=b-1}^{b+1} T_{i,j}$$

V rámci zpracování každé z dlaždic $T_{a,b}$ bude tak sestaven kandidátský seznam objektů. Ten bude obsahovat objekty dlaždice $T_{a,b}$ z první vrstvy a objekty dlaždic $T_{a-1,b-1}$, $T_{a-1,b}$, $T_{a-1,b+1}$, $T_{a,b-1}$, $T_{a,b}$, $T_{a,b+1}$, $T_{a+1,b-1}$, $T_{a+1,b}$ a $T_{a+1,b+1}$ z druhé vrstvy (viz Obr. 26). Nad těmito objekty bude proveden výpočet podobnostní matice D a objekty budou k sobě přiřazeny, jak bylo uvedeno v kapitole 7.2.2.

Každý objekt z druhé vrstvy tak bude přiřazován celkem 9 krát, tj. s objekty z odpovídající dlaždice a s objekty z každé z osmi sousedních dlaždic. Aby se zabránilo chybnému přiřazení, bude v případě přiřazení ukládána vzájemná nepodobnost obou přiřazených

objektů. Pokud bude v dalších porovnáních objekt znovu přiřazen, bude porovnána dosažená nepodobnost a objekt bude přiřazen k objektu s nejmenší nepodobností. Je pravděpodobné, že k vícenásobnému přiřazení nebude docházet příliš často, protože jedním z podobnostních kritérií je i vzdálenost objektů, tudíž budou pravděpodobně nejčastěji přiřazeny objekty ležící ve stejné dlaždici.



Obr. 26: Pro objekty z konkrétní dlaždice první vrstvy budou testovány vůči objektů z těžce a všech okolních dlaždic druhé vrstvy, zdroj: vlastní

7.4 Porovnání map

Výsledkem předchozích kroků bude přiřazení odpovídajících si objektů. V závislosti na rozdílu počtu objektů v dlaždici v obou mapách mohou některé objekty zůstat nepřijřazeny.

S přiřazenými objekty můžeme provést porovnání mapových vrstev. Při porovnání celých mapových vrstev lze vycházet z výzkumu, který provedli Frank a Ester (2006) a byl popsán v kapitole 2.6. Frank a Ester hodnotili podobnosti dvou map ze tří různých hledisek:

- individuálního, kde hodnotili podobnost tvaru odpovídajících si objektů,
- lokálního, kde byla hodnocena poloha k okolním objektům,
- globálního, kde byly zkoumány počty objektů ve vrstvách a jejich prostorové rozložení

Každé hledisko vypovídá o podobnosti map z jiného pohledu a pro různé aplikace mohou mít tyto hlediska různou důležitost. Proto autoři nastavují jednotlivým hlediskům váhy až ad hoc podle požadované aplikace.

7.4.1 Individuální hledisko

Individuální hledisko bude hodnoceno jako podobnost tvarů odpovídajících si objektů. K porovnání tvarů přiřazených objektů O_i a O'_i bude použita L^2 vzdálenost turning funkcí $d_s(O_i, O'_i)$ popisovaná v kapitole 7.1.1.

Celková podobnost tvaru SS při porovnání dvou mapových vrstev bude definována jako průměrná L^2 vzdálenost TF odpovídajících si objektů:

$$SS = \frac{\sum_{i=1}^n d_s(O_i, O'_i)}{n}$$

kde n je počet ztotožněných objektů v obou vrstvách, O_i je i -tý objekt první vrstvy a O'_i je k němu přiřazený objekt druhé vrstvy. Objekty, které nebyly přiřazeny, budou z výpočtu vynechány.

7.4.2 Lokální hledisko

Lokální hledisko sleduje vztah objektu k objektům v jeho okolí. Pro posouzení lokálního hlediska bude použito relativní určení polohy prostřednictvím Voroného diagramů, jak bylo popsáno v kapitole 6.2.2 a kterou rovněž používají Frank a Ester (2006). V tomto případě je její využití vhodnější než použití metody fixní vzdálenosti, popsané v kapitole 7.1.3 a používané v kapitole 7.1.3 pro hledání odpovídajících si objektů. Metoda založená na Voroného diagramech totiž sleduje vztahy ke konkrétním objektům v okolí, zatímco metoda fixní vzdálenosti pouze sčítá plochy objektů v různých sektorech okolo zkoumaného objektu.

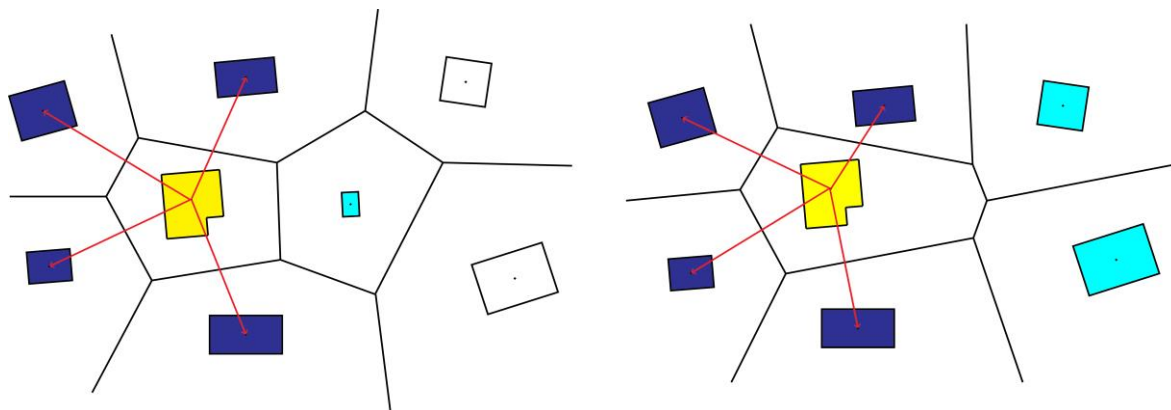
Vymezení přirozených sousedů, založené na Voroného diagramech, bylo popsáno v kapitole 6.2.2. Nad oběma mapovými vrstvami budou vymezeny Voroného diagramy a jako sused bude označen každý objekt, který má se zkoumaným objektem společnou hranu Voroného buňky. Celý postup se pro každý objekt O_i první vrstvy skládá z několika částí:

1. Pro objekt O_i první vrstvy je určena množina přirozených sousedů N_i .
2. V druhé vrstvě je nalezen takový objekt O'_i , který je přiřazen objektu O_i .
3. Pro objekt O'_i druhé vrstvy je určena množina přirozených sousedů N'_i .
4. Je vytvořena množina společných přirozených sousedů N_i^C v první vrstvě. Do té spadají ty objekty K , které se vyskytují v N_i a zároveň k nim přiřazený objekt K' patří

do množiny N'_i . Současně je vytvořena obdobná množina společných přirozených sousedů v druhé vrstvě označená N_i^C :

$$K \in N_i^C, K' \in N_i^C \Leftrightarrow K \in N_i \wedge K' \in N'_i$$

5. Jsou vypočteny euklidovské vzdálenosti v první vrstvě od objektu O_i ke všem společným sousedům K_{ij} z N_i^C , a zároveň v druhé vrstvě vzdálenosti od přiřazeného objektu O'_i ke všem společným sousedům K'_{ij} z N_i^C (Obr. 27).



Obr. 27: Ze sousedů obou odpovídajících si objektů jsou vybráni společní a jsou vypočteny vzdálenosti; žlutá: posuzovaný objekt, tmavě modrá: společní sousedé, světle modrá: sousedé jen v jedné z vrstev, zdroj: vlastní

6. Pro každého společného souseda K_{ij} , K'_{ij} je zjištěn rozdíl v jeho vzdálenosti od centrálního objektu O_i , respektive O'_i v obou vrstvách.

$$diff_{ij} = \left| \|O_i - K_{ij}\|_2 - \|O'_i - K'_{ij}\|_2 \right|,$$

kde $\|O_i - K_{ij}\|_2$ je euklidovská vzdálenost centrálního objektu O_i a jeho souseda K_{ij} .

7. Je zjištěna maximální vzdálenost některého z přirozených sousedů objektu O_i , O'_i :

$$maxdist_i = \max \left(\max \left(\|O_i - K_{ij}\|_2 \right); \max \left(\|O'_i - K'_{ij}\|_2 \right) \right),$$

pro všechna $K_{ij} \in N_i, K'_{ij} \in N'_i$.

8. Je vypočten normalizovaný rozdíl vzdáleností mezi centrálním objektem a každým jeho sousedem. Nakonec je vypočten průměr mezi všemi sousedy z N_i^c :

$$LS_i = \frac{\sum_{j=1}^p \left| \frac{diff_{ij}}{maxdist_i} \right|}{p},$$

kde p je počet společných sousedů.

Hodnotu LS_i lze interpretovat jako průměrnou změnu vzdáleností mezi konkrétním centrálním objektem O_i a jeho sousedy, kteří vyskytují v obou vrstvách. Celková hodnota podobnosti lokálního hlediska bude počítána jako průměr napříč všemi ztotožněnými objekty v celé mapové vrstvě:

$$LS = \frac{\sum_{i=1}^n LS_i}{n}$$

7.4.3 Globální hledisko

Globální hledisko se zabývá celkovými počty objektů v obou vrstvách. Toto hledisko tak stanovíme jako normalizovaný rozdíl počtu objektů obou vrstev P_1 , P_2 . Opět použijeme Canberrskou metriku:

$$CS = \frac{|P_1 - P_2|}{P_1 + P_2}$$

Individuální hledisko řeší přesnou podobnost tvarů odpovídajících si objektů a lokální hledisko řešilo změnu pozice odpovídajících si objektů na lokální úrovni. Obě tato hlediska postihovala pouze objekty, které byly ztotožněny. Naproti tomu globální hledisko sleduje objekty všechny, tedy i ty, které se nepodařilo ztotožnit.

8 IMPLEMENTACE

V této kapitole bude popsána implementace celého navrženého algoritmu. Text se zaměřuje na důležité struktury a podstatné netriviální části algoritmu. Celý kód je přiložen na DVD.

Postup uvedený v předchozí kapitole byl implementován v programovacím jazyku Python 2.7 s využitím dalších rozšiřujících modulů. Zejména byla využita knihovna `arcpy`, která zpřístupňuje funkce pro práci s prostorovými daty ze systému ArcGIS 10.2. Pro výpočty a práci s numerickými hodnotami byla použita knihovna `numpy`. Programovací jazyk Python byl zvolen zejména kvůli dobře dostupným knihovnám pro práci s prostorovými daty, snadnosti jeho použití a v neposlední řadě kvůli provázání se systémem ArcGIS.

V rámci práce byly použity tyto balíčky, knihovny a moduly pro Python:

- `Scipy` – balíček pro usnadňující použití Pythonu ve vědeckých, technických a matematických aplikacích, dostupný v rámci open-source licence BSD (Jones, et al., 2014). V rámci něho byly použity následující knihovny:
 - `Numpy` – knihovna pro práci N-dimensionálními poli a maticemi
 - `Scipy.spatial` – knihovna pro práci s prostorovými strukturami, jakou je například Voroného diagram
 - `Matplotlib` – knihovna určená pro snadné vykreslování grafů
- `Arcpy` – balíček zpřístupňující funkce pro práci s prostorovými daty systému ArcGIS v rámci jazyka Python. Balíček funguje pouze ve spojení se systémem ArcGIS, který je poskytován jako proprietární komerční software.
- `Munkres` – modul pro řešení přiřazovacího problému pomocí Hungarian algorithm, poskytovaný v rámci open-source licence BSD (Clapper, 2008).

V dalších podkapitolách bude popsána celá implementace navrženého algoritmu od načtení dat, přes jejich rozdělení do dlaždic, výpočet podobností objektů, přiřazení objektů až po závěrečné určení podobnosti mapových vrstev. S výjimkou použití výše popsaných knihoven jsou všechny dále popsané součásti programu vzniklé za účelem této práce.

8.1 Datová reprezentace polygonů

U uchovávání mapových vrstev a objektů v nich se stará modul **readshp.py**. Každý polygon na mapě je uložen v instanci třídy **MyPolygon**. Ta má následující atributy:

- `Vertices` – seznam uspořádaných dvojic souřadnic x,y vertexů polygonu,
- `Area` – vypočítaná plocha polygonu,
- `Centroid` – vypočítaný centroid,
- `OID` – Object ID, unikátní číslo objektu v dané vrstvě,
- `Tile` – dlaždice, do které polygon spadá,
- `MatchObject` – OID přiřazeného objektu,
- `MatchScore` – cena přiřazení k objektu `MatchObject`,
- `TurningFunction` – instance třídy **TurningFunction** reprezentující tvar objektu,
- `Neighbours` – atribut slouží k uložení přirozených sousedů v rámci Voroného diagramu.

Při načítání každého objektu je konstruován objekt **MyPolygon**, do kterého je ukládán tvar objektu (seznam vertexů), identifikátor objektu, jeho plochu a centroid. Zároveň při konstrukci každého objektu je na základě vertexů vytvořena instance třídy **TurningFunction** (viz dále).

Mapové vrstvy jsou uloženy v instanci třídy **Layer**. Ta se stará o uložení seznamu polygonů, ukládá si souřadnicový rozsah (`extent`) mapové vrstvy a v neposlední řadě je zodpovědná za rozdělení objektů do dlaždic. O nahrání dat se stará její metoda **readSHP** využívající `arcpy.da.SearchCursor`, který dokáže přečíst prostorová data z shapefilu i geodatabáze. Polygony jsou postupně ze shapefilu převáděny na objekty třídy **MyPolygon** a ukládány do seznamu (Alg. 8.1).

Alg. 8.1: Načtení shapefilu

```
1. def readSHP(self, input_shp):
2.     self.extent=arcpy.Describe(input_shp).extent
3.     cursor = arcpy.da.SearchCursor(input_shp, ["OID@", "SHAPE@XY", "SHAPE@AREA",
"SHAPE@"])
4.     for feature in cursor:
5.         self.objectList.append( MyPolygon( feature[3].getPart(0), feature[2],
feature[1], feature[0]))
```

Kromě toho třída **Layer** obsahuje další důležité metody pro práci s vrstvami. Metoda **calculateCommonExtent**(self, ext2) určí území společné s jinou vrstvou. To je důležité pro další kroky, protože analýza bude prováděna jen na společném území dvou mapových vrstev. Vstupní extent ext2 je porovnán s extentem dané instance třídy **Layer** a společný extent je určen jako jejich průnik.

Metoda **calculateTileEdgeLength**(avgObj) spočítá délku hrany dlaždice na základě požadovaného průměrného počtu objektů v dlaždici (avgObj) podle vzorců uvedených v kapitole 7.3. Metoda **createTiles**(tileEdge) pak na základě této vypočítané délky hran (tileEdge) vytvoří dlaždice a přiřadí objekty k dlaždicím podle polohy jejich centroidů. Metoda **computeNeighbours**() využívá knihovnu **SciPy** k vytvoření Voroného diagramu. Na základě vytvořeného Voroného diagramu pak objektům v dané vrstvě přiřadí jejich přirozené sousedy.

Pro snazší přístup k dlaždicím a do nich patřících objektů slouží třída **Tile**. Její instance reprezentují konkrétní dlaždice a uchovávají si seznam do nich náležících objektů.

Poslední třídou v modulu readshp.py je třída **LayerComparator**(layer1, layer2). Vstupními objekty do jejího konstrukturu jsou dvě instance třídy **Layer**, u nichž spočítá společný extent a rozčlení toto jejich společné území do odpovídajících si dlaždic. Metoda **match**() třídy **LayerComparator** pak slouží ke spouštění přiřazování objektů v dlaždicích takto rozčleněného území. Zpracování jedné dlaždice probíhá následujícím způsobem:

1. Jsou vybrány objekty dané dlaždice z první vrstvy.
2. Jsou vybrány objekty dané dlaždice a okolních osmi dlaždic, tedy celkem devíti dlaždic, z první vrstvy.
3. Jsou vybrány objekty odpovídající dlaždice a okolních osmi dlaždic, tedy celkem devíti dlaždic, z druhé vrstvy.

Všechny tyto seznamy objektů jsou vstupy pro konstruktore třídy **TileComparator** modulu tiles.py, která se posléze stará o samotné zhodnocení a přiřazení takto zasláných objektů.

8.2 Zpracování objektů v dlaždicích

Objekty, určené ke zpracování v rámci jedné dlaždice, jsou předávány do konstruktoru třídy **TileComparator** modulu `tiles.py` jako parametry ve formě trojice seznamů:

- `Tile1col` – obsahuje polygony spadající do právě zpracovávané dlaždice z první vrstvy
- `Tile1ol` – obsahuje objekty z právě zpracovávané dlaždice a z okolních osmi dlaždic z první vrstvy
- `Tile2ol` – obsahuje objekty z právě zpracovávané a z okolních osmi dlaždic z druhé vrstvy

Tyto seznamy jsou v rámci konstruktoru třídy **TileComparator** ukládány ve formě objektů třídy **ProcessTile**. Ta slouží k uchovávání a práci s objekty konkrétní dlaždice. Uchovává seznamy objektů a kromě jiného dokáže i popsat okolí objektu.

Třída **TileComparator** se stará o porovnání vstupujících objektů z různých hledisek a jejich následné přiřazení. Obsahuje metody pro výpočet matic podobnosti z jednotlivých hledisek popsaných v kapitole 7.1:

1. **compareEuclidean** (`dmax`) – Analýza absolutní polohy, pro každou dvojici je počítána podobnost jejich absolutní polohy podle vzorce uvedeného v kapitole 7.1.2. Parametr `dmax` odpovídá vzdálenosti centroidů, od které již budou objekty hodnoceny jako nepodobné.
2. **compareTFs** (`rotVar`) – Analýza tvaru s využitím podobnosti TF, dle kapitoly 7.1.1. Parametr `rotVar` může nabývat hodnot 0 pro rotačně invariantní algoritmus a 1 pro rotačně variantní algoritmus. Podrobná implementace bude popsána dále.
3. **compareNeighbourhood** (`dmax`) – Analýza relativní polohy, dle kapitoly 7.1.3. Parametr `dmax` označuje poloměr okolí a je standardně nastaven na 200 m. Podrobná implementace bude popsána dále.
4. **compareArea** () – Analýza velikosti objektu, pro každou dvojici je počítána podobnost obsahu jejich ploch podle vzorce uvedeného v kapitole 7.1.4.

Každá z těchto metod pracuje tak, že zhodnotí podobnost m objektů aktuální dlaždice první vrstvy (`tile1c`) s n objekty spadajícími do odpovídající nebo okolních osmi dlaždic

z druhé vrstvy (`tile2`). Protože zejména u krajních objektů se mohou odpovídající objekty vyskytovat v druhé vrstvě v sousední dlaždici, jsou testována vždy jedna dlaždice první vrstvy vůči devíti dlaždicím druhé vrstvy. Fungování těchto metod lze ukázat na metodě `compareArea()`, která porovnává plochu objektů na základě Canberrské metriky podle postupu uvedeného v kapitole 7.1.4. Jak ukazuje Alg. 8.2, je nejprve vytvořena matice $m \times n$ se všemi prvky rovny -1 . Postupně je do každého prvku matice `arsim[i, j]` uložena informace o podobnosti i -tého objektu z první vrstvy s j -tým objektem druhé vrstvy podle vzorce $d_a(A, B) = \frac{|S_A - S_B|}{S_A + S_B}$. Výsledná matice odpovídá matici D_a popsané v kapitole 7.2.1. Obdobně je řešen i výpočet dalších podobnostních matic.

Alg. 8.2: Metoda `compareArea`

```

1. def compareArea(self):
2.     arsim=-1*np.ones(shape=(self.tile1c.getLength(),self.tile2.getLength()))
3.     for oid1 in range(self.tile1c.getLength()):
4.         for oid2 in range(self.tile2.getLength()):
5.             area1=self.tile1c.getObject(oid1).getArea()
6.             area2=self.tile2.getObject(oid2).getArea()
7.             arsim[oid1,oid2]=(abs(area1-area2))/(area1+area2)
8. return arsim

```

Metoda `compareAll()` třídy `TileComparator` zabezpečuje sečtení výše zmíněných matic podobnosti na základě zadaných vah jednotlivých hledisek. K určení správného nastavení vah slouží modul `testing.py`, který bude představený v dalších kapitolách. Metoda `match(eux, nex, tfx, aex, dmax)` se pak stará o optimální přiřazení objektů na základě výsledné sečtené matice podobnosti. Parametry `eux`, `nex`, `tfx`, `aex` odpovídají vahám jednotlivých metrik (absolutní polohy, relativní polohy, tvaru a velikosti, viz kapitola 8.2.3) a `dmax` označuje maximální vzdálenost pro metriky absolutní a relativní polohy. K tomu využívá Hungarian algorithm implementovaný v rámci modulu `munkres.py`. Výsledkem je přiřazení všech m objektů `tile1col` k n objektům `tile2ol`. V případě přiřazení je u objektů uchováván odkaz na přiřazený objekt (`MyPolygon.matchObject`) a na celkovou podobnost takto přiřazených objektů (`MyPolygon.matchScore`). Vzhledem k tomu, že konkrétní dlaždice z druhé vrstvy jsou přiřazovány 9 krát, tj. s odpovídající dlaždicí a všemi okolními, mohou být novým přiřazením přiřazeny již přiřazené objekty. Z tohoto důvodu je uchováváno `matchScore`, které zaznamenává podobnost přiřazených objektů. Pokud je podobnost nového přiřazení větší než podobnost stávajícího, je přiřazení nahrazeno.

8.2.1 Turning funkce a jejich porovnání

Jednou z metrik, které vstupují do výpočtu, představuje podobnost tvaru objektů popsaná prostřednictvím turning function. Na rozdíl od výpočtu podobnosti plochy či polohy centroidů je podobnost tvaru náročnější na implementaci i výpočet. Jak bylo uvedeno v kapitole 5.2, má samotné porovnání tvarů dvou objektů složitost $O(mn(m+n))$ pro polygony s m a n vertexy. Pro porovnání turning funkcí dvou objektů slouží funkce `compute(tf1, tf2, rotVar)` modulu `turning.py` popsaná dále.

Turning funkce každého objektu je uchovávána v instanci třídy **TurningFunction** modulu `turning.py`. V souladu s kapitolou 7.1.1 je turning funkce vypočtena a uchovávána jako seznam bodů nespojitosti $[s_i; \alpha_i]$ odpovídajících vertexům polygonu. Tento seznam je uložen v atributu `TFpoints`. Třída dále obsahuje metody pro práci s turning funkcí. Metoda `getArea()` vypočítává pro TF $f(s)$ hodnotu integrálu $\int_0^1 f(s) ds$. Metoda `getXs(t)` vrací seznam všech s_i dané funkce posunutých o t a metoda `getTFpoints(t)` vrací seznam všech párů $[s_i; \alpha_i]$ posunutých o t .

Při výpočtu vzdálenosti dvou TF funkcí `compute(tf1, tf2, rotVar)` je postupováno podle postupu podrobně popsaného v kapitole 7.1.1. Na vstupu (Alg. 8.3) jsou dva objekty třídy **TurningFunction** k porovnání a parametr `rotVar` označující, zda je požadována invariance vůči rotaci. Defaultně je nastaven na 0, čili algoritmus rotačně invariantní.

Nejprve jsou počítány všechny kritické posuny t_c , tj. takové posuny t první TF, při kterých některý bod nespojitosti posunutý první TF $f(s+t)$ koinciduje s některým bodem nespojitosti druhé TF $g(s)$. Poté jsou do proměnné `tf2_p` jsou načteny body druhé TF. První TF je postupně posouvána o všechna t_c podle Alg. 7.1 a její hodnoty jsou po každém posunutí ukládány do proměnné `tf1_p`. Rozdíl mezi oběma TF je posléze vypočten jako $\int_0^1 (f(s+t) - g(s))^2 ds$ pomocí rozdělení na vertikální pásy, jak bylo podrobně popsáno v kapitole 7.1.1, a v Alg. 7.2.

Pokud je požadován rotačně invariantní algoritmus, je ještě vypočtena optimální rotace Φ^* a od hodnoty $\int_0^1 (f(s+t) - g(s))^2 ds$, uložené v proměnné A , je odečtena její druhá mocnina. Rozdíl TF je tak počítán jako $\int_0^1 (f(s+t) - g(s))^2 ds - \Phi^{*2}$. Výsledný rozdíl porovnání TF pro konkrétní posunutí t je porovnán s dosaženým minimem. Pokud je menší, je minimum nahrazeno. Takto je postupováno, dokud nejsou otestována všechna t_c . Protože

u naprosto totožných tvarů může vlivem matematických omezení programovacího jazyka dojít k záporným výsledkům blízkým nule, je před jeho výstupem použita funkce absolutní hodnota, aby výsledky byly vždy kladné.

Alg. 8.3: Výpočet L^2 vzdáleností dvou TF

```

1. def compute(tf1,tf2,rotVar=0):
2.     cp=[]
3.     for i in tf1.getXs():
4.         for j in tf2.getXs():
5.             ds=i-j
6.             if ds<0:
7.                 ds=ds+1
8.                 cp.append(ds)
9.     alfa=tf2.getArea()-tf1.getArea()
10.    tf2_p=tf2.getTFpoints(0)
11.    minA=9999
12.    for t in cp:
13.        s=0
14.        A=0
15.        tf1_p=tf1.getTFpoints(t)
16.        if tf1_p[0][0]<>0:
17.            tf1_p.insert(0,[0,tf1_p[-1][1]])
18.        i=0
19.        j=0
20.        while (s<1):
21.            if i>=len(tf1_p)-1:
22.                s1=1
23.            else:
24.                s1=tf1_p[i+1][0]
25.            if j>=len(tf2_p)-1:
26.                s2=1
27.            else:
28.                s2=tf2_p[j+1][0]
29.            s_old=s
30.            if s1<s2:
31.                s=s1
32.                da=tf1_p[i][1]-tf2_p[j][1]
33.                i=i+1
34.            else:
35.                s=s2
36.                da=tf1_p[i][1]-tf2_p[j][1]
37.                j=j+1
38.            A=A+(s-s_old)*da*da
39.            if rotVar==0:
40.                phi=alfa-2*math.pi*t
41.                A=A-phi*phi
42.            if A<minA:
43.                minA=A
44.    return abs(minA)

```

8.2.2 Porovnání okolí

Při porovnávání okolí dvou posuzovaných objektů je postupováno podle postupu uvedeného v kapitole 7.1.3. O popis okolí se stará třída **ProcessTile** modulu **tiles.py**. Ta uchovává seznamy objektů. Na rozdíl od třídy **Tile** modulu **readshp.py** však nereprezentuje pouze jedinou dlaždici ve smyslu rozdělení území do dlaždic dle kapitoly 7.3. Jsou v ní uloženy objekty z více různých dlaždic, které jsou zpracovávány současně. To je výhodné právě při posuzování okolí. Okolní objekty konkrétního objektu jedné dlaždice se totiž mohou vyskytovat nejen v oné dlaždici, ale i v dlaždicích sousedních. Z tohoto důvodu jsou v rámci

posuzování okolí uloženy v třídě **ProcesTile** objekty posuzované (střední) dlaždice i všech osmi okolních dlaždic.

Konkrétní postup získání matice M popisující okolí i -tého objektu zabezpečuje metoda **getNeighbourhood**(i, eps) třídy **ProcesTile** (Alg. 8.4). Je vytvořena matice 8×4 odpovídající osmi základním směrům a čtyřem intervalům vzdálenosti od posuzovaného objektu. U každého objektu z konkrétní instance **ProcesTile** je zjištěna vzdálenost a směr k posuzovanému objektu. Pokud se nachází v okolí definovaném vzdáleností (parametr eps), je jeho plocha přičtena do příslušného prvku matice (dle směru a vzdálenosti).

Alg. 8.4: Získání matice M popisující okolí objektu

```

1. def getNeighbourhood(self,i,eps): #relative position description of neighbourhood
2.     cPoint=self.objectList[i].getCentroid()
3.     M=np.zeros(shape=(8,4))
4.     for obj in self.objectList:
5.         centroid=obj.getCentroid()
6.         dx=centroid[0]-cPoint[0]
7.         dy=centroid[1]-cPoint[1]
8.         r=sqrt(dx**2+dy**2)
9.         if r<eps:
10.            d=3
11.            if r<(eps*0.75):
12.                d=2
13.                if r<(eps*0.5):
14.                    d=1
15.                    if r<(eps*0.25):
16.                        d=0
17.            s=0
18.            if dx<0:
19.                s=s+4
20.            if dy<0:
21.                s=s+2
22.            if fabs(dx)>fabs(dy):
23.                s=s+1
24.            M[s,d]=M[s,d]+obj.getArea()
25.     return M

```

Porovnání takto získaných matic je obstaráváno funkcí **matrixCompare**($m1, m2$) (Alg. 8.5). Ta pro dvě stejně velké matice porovnává vždy odpovídající si prvky matice prostřednictvím Canberrské metriky. Výsledkem porovnávání je průměr z porovnání všech prvků.

Alg. 8.5: Porovnání stejně velkých matic Canberrskou metrikou

```

1. def matrixCompare(m1, m2):
2.     if m1.shape==m2.shape:
3.         s=0
4.         a=m1.flatten()
5.         b=m2.flatten()
6.         for i in range(a.size):
7.             if (a[i]+b[i])<>0:
8.                 s=s+(abs(a[i]-b[i]))/(a[i]+b[i])
9.         s=s/a.size
10.    return s

```

8.2.3 Testování vah

Aby byl výsledek přiřazení, provedeného metodou `match()`, která byla popsána v předchozí kapitole, co nejlepší, je třeba správně nastavit váhy jednotlivým dílčím metrikám. Nastavení vah se může lišit podle charakteru porovnávaných mapových vrstev, jak bylo detailně popsáno v kapitole 7.2.1. Ke správnému nastavení vah slouží modul `testing.py`. Ten obsahuje jedinou třídu `TestingTileComparator`, která je potomkem třídy `TileComparator` modulu `tiles.py`.

Testování je prováděno na zvoleném testovacím území. Protože při testování není prováděno rozdělení území do dlaždic, a také pro to, že je nutné manuálně přiřadit pro každý objekt tohoto testovacího území odpovídající objekt ve všech testovaných vrstvách, nemělo by se na testovaném území nacházet více než 100 objektů.

Po načtení území formou seznamu objektů obou vrstev může být území vykresleno metodou `drawLayers()`. Jsou tak zobrazeny objekty obou vrstev do jednoho obrázku, kde každý objekt je popsán svým identifikačním číslem. Na základě tohoto vykreslení musí být do textového souboru ručně operátorem zapsány odpovídající si objekty, tj. taková přiřazení objektů, která budou považována za správná a vůči kterým bude testováno přiřazení automatické. Každé správné přiřazení musí být na samostatném řádku textového souboru ve formátu „A B“, kde A je identifikační číslo objektu první vrstvy a B je identifikační číslo odpovídajícího objektu druhé vrstvy. Metoda `loadMatching(txt)` pak slouží k načtení takto vzniklého textového souboru a uložení správných přiřazení ve formě slovníku.

K samotnému testování slouží metoda `test`, která nastavuje různé váhy jednotlivým metodám výpočtu podobnosti (podle absolutní a relativní polohy, tvary, velikosti) v intervalu $\langle 0; 4 \rangle$ a sleduje počet objektů přiřazených v souladu s manuálním přiřazením. Jsou vyzkoušeny všechny možnosti nastavení vah, tedy 5 možných vah pro 4 různé metriky. Z těchto 5^4 (= 625) možností je vybrána ta s největším počtem správných přiřazení. Výsledkem této metody je optimální nastavení vah pro daný soubor. Toto nastavení by mělo být používáno při volání metody `match(eux, nex, tfx, aex, dmax)` třídy `TileComparator`.

8.3 Porovnání mapových vrstev

Výsledkem metody `match()` třídy `TileComparator` popsané v kapitole 8.2 je přiřazení objektů jedné dlaždice k objektům druhé dlaždice. Jakmile je toto provedeno pro všechny

dlaždice, obsahují objekty obou mapových vrstev ukazatel na odpovídající si objekt z opačné vrstvy. Takto zpracované mapové vrstvy je možné porovnat mezi sebou.

Pro porovnání mapových vrstev byla vytvořena metoda `compareLayers(i, l, g)` třídy `LayerComparator` modulu `readshp.py`. Ta hodnotí vzájemnou podobnost mapových vrstev ze tří hledisek, jak bylo detailně navrženo v kapitole 7.4:

- z hlediska individuálního – změna tvaru odpovídajících si objektů,
- z hlediska lokálního – změna relativních vzdáleností přirozených sousedů odpovídajících si objektů,
- z hlediska globálního – změny v počtu objektů v obou mapových vrstvách.

Individuální hledisko. Individuální hledisko `ss` je počítáno dle kapitoly 7.4.1, tedy jako průměrná vzdálenost TF odpovídajících si objektů.

Lokální hledisko. Pro zjištění podobnosti z lokálního hlediska je nezbytné stanovit Voroného diagramy pro obě vrstvy. To je nutný krok k zjištění přirozených sousedů. Voroného diagramy a přirození sousedé jsou pro každou vrstvy vypočteny metodou `computeNeighbours()` třídy `Layer`.

Ke konstrukci Voroného diagramů je využívána metoda `Voronoi` knihovny `scipy.spatial`. Ta dokáže pro vstupní množinu bodu vygenerovat Voroného diagramy. Výstupem této metody jsou hrany Voroného diagramu (`vor.ridge_points`) reprezentované jako páry identifikátorů (pořadových čísel) obou objektů majících společnou hranu. Tato metoda je použita nad centroidy objektů v dané vrstvě (Alg. 8.6). Pro každou hranu jsou přirození sousedé navzájem doplněni do seznamu přirozených sousedů objektů.

Alg. 8.6: Doplnění seznamu sousedů

```
1. def computeNeighbours(self):
2.     centroids=self.getCentroids()
3.     vor=Voronoi(np.array(centroids))
4.     for edge in vor.ridge_points:
5.         self.objectList[edge[0]].neighbours.append(self.objectList[edge[1]])
6.         self.objectList[edge[1]].neighbours.append(self.objectList[edge[0]])
```

Po přiřazení přirozených sousedů je možno přistoupit k porovnání vrstev z lokálního hlediska podle návrhu algoritmu 7. z kapitoly 7.4.2. Podobně jako v případě tvarové (individuální) podobnosti je posuzována podobnost okolí pro každý přiřazený objekt (Alg. 8.7). Pro každý objekt jsou vybráni ti sousedé, kteří jsou stejní v obou vrstvách (`commonNbs`). Pro každého společného souseda je zjištěn rozdíl mezi vzdáleností

k posuzovanému objektu v první a ve druhé vrstvě a přidán do seznamu `diff`. Zároveň je zjišťována vzdálenost nejvzdálenějšího souseda posuzovaného objektu (proměnná `maxDist`), kterou jsou všechny rozdíly v seznamu `diff` normalizovány. Poté je vypočten průměr normalizovaných vzdáleností v rámci posuzovaného objektu (`diff_avg`). Nakonec je vypočten průměr nad všemi posuzovanými objekty (`ls`). Tento průměr určuje podobnost lokálního hlediska celých mapových vrstev a opět vychází v intervalu $\langle 0; 1 \rangle$, kde 0 značí shodnost a 1 nepodobnost. V praxi může nastat situace, kdy nebudou existovat společní sousedé. V takovém případě je rovnou stanovena podobnost okolí daného objektu jako 1, tedy nejvyšší možnou hodnotou.

Alg. 8.7: Výpočet podobnosti map z lokálního hlediska

```

1. self.layer1.computeNeighbours()
2. self.layer2.computeNeighbours()
3. sumDiff=0
4. for obj in ol:
5.     commonNbs=[nb for nb in obj.neighbours if nb.matchObject in
obj.matchObject.neighbours]
6.     maxDist=0
7.     diff=[]
8.     for cnb in commonNbs:
9.         dist1=obj.distance(cnb.getCentroid())
10.        dist2=obj.matchObject.distance(cnb.matchObject.getCentroid())
11.        if dist1>maxDist:
12.            maxDist=dist1
13.        if dist2>maxDist:
14.            maxDist=dist2
15.        diff.append(abs(dist1-dist2))
16.    if len(diff)>0:
17.        diff_avg=(sum(x/maxDist for x in diff))/len(diff)
18.    else:
19.        diff_avg=1
20.        sumDiff=sumDiff+diff_avg
21.    if (len(ol))>0:
22.        ls=sumDiff/(len(ol))
23.    else:
24.        ls=1

```

Globální hledisko. Výpočet podobnosti v globálním hledisku g_s je řešen porovnáním počtu objektů v každé z vrstev formou Canberrské metriky podle kapitoly 7.4.3.

Výstupem této metody mohou je podobnost dvou mapových vrstev ze tří různých hledisek – individuálního (tvarového) s_s , lokálního (změna polohy sousedů) l_s a globálního (počty objektů) g_s . Jak bylo uvedeno dříve, v každé aplikaci je důraz kladen na jinou složku podobnosti mapových vrstev. Proto je možné nastavit v rámci metody `compareLayers` (i, l, g) jednotlivým hlediskům váhy i, l a g . Výstupem této metody je pak číslo v intervalu $\langle 0; 1 \rangle$, které je váženým průměrem individuálního, lokálního a globálního hlediska.

9 HODNOCENÍ VÝSLEDKŮ

Předchozí kapitola se zabývala implementací navrženého algoritmu v programovacím jazyce Python. Tato kapitola se zaměří na aplikaci navrženého programu na konkrétních mapových vrstvách v rámci konkrétních území. Nejprve bude na několika různých případech otestováno přiřazování objektů. Budou postupně testovány jednotlivé dílčí techniky vždy samostatně a posléze společně s různým nastavením jejich vah. Bude sledován počet správně přiřazených objektů. Výsledkem tohoto testování bude nalezení vhodných vah pro charakterem různé mapové vrstvy. S použitím těchto ideálních vah bude provedeno finální přiřazení objektů v testovaných vrstvách a bude zhodnocena podobnost vrstev. V závěru kapitoly bude provedeno zhodnocení návržení, implementace a výsledků algoritmu.

9.1 Testovací data

Algoritmus byl testován na vektorových polygonových datech obsahujících půdorysy budov. Jelikož algoritmus neumí pracovat s objekty s různou úrovní agregace, bylo třeba pro testování použít takové vektorové mapové vrstvy, které obsahují přibližně stejný počet objektů. Pokud by objekty v jedné z vrstev byly agregovány do větších celků než ve vrstvě druhé, nebyly by mezi nimi nalézány shody. Rovněž bylo nutné zabezpečit, aby použité mapové vrstvy měly překryv území, na kterém by bylo možno je testovat. Podle těchto podmínek byly vybrány čtyři mapové vrstvy (Obr. 28), které mají společný překryv v oblasti západní části Prahy.

CEDA – PRAHA 1 : 10 000. Z databáze Praha 1 : 10 000 byla použita vrstva Budovy. Ta zobrazuje budovy poměrně generalizované, bez detailů. V některých případech zde dochází k agregaci některých objektů nebo je zde kardinalita $M : N$. Často je v této vrstvě například 5 stejných budov vedle sebe zobrazeno jako 4 budovy mírně zvětšené. Vrstva byla poskytnuta Katedrou aplikované geoinformatiky a kartografie PŘF UK.

ZABAGED. Z tzv. Základní báze geografických dat České republiky byla použita vrstva „Budova jednotlivá nebo blok budov“. Ač je referenční měřítko těchto dat také 1 : 10 000, jsou výrazně podrobnější než v případě Cedy. Je zde zobrazeno větší množství detailů i jednotlivých budov. Vrstva byla poskytnuta Katedrou aplikované geoinformatiky a kartografie PŘF UK.

KN. Vrstva „Stavební objekt“ Katastru nemovitostí byla získána přes Veřejný dálkový přístup (ČÚZK, 2014). Obsahuje větší množství objektů ve větších detailech než předcházející dvě vrstvy. Obsahuje i malé stavební objekty, jako jsou garáže, trafostanice, kůlny atd. Některé i významné stavby nebo jejich části nejsou na mapě znázorněny.

Buildings_CR. Tato vrstva poskytnutá Katedrou aplikované geoinformatiky a kartografie PřF UK má podobné charakteristiky jako vrstva KN, avšak na rozdíl od ní jsou ve vrstvě znázorněny všechny významné budovy i jejich části. Obsahuje detaily budov, avšak malé objekty jsou často agregovány.



Obr. 28: Ústřední vojenská nemocnice ve Střešovicích v mapových vrstvách CEDA (vlevo nahoře), ZABAGED (vpravo nahoře), KN (vlevo dole) a Buildings_CR (vpravo dole), zdroj: vlastní

Aby bylo dosaženo co největší homogenity dat, byly všechny vrstvy předzpracovány v softwaru ArcGIS. Nejprve byla použita funkce *Dissolve*, která způsobila spojení budov sdílejících společnou zeď do jednoho bloku. Následně použitá funkce *Multipart to singlepart* zajistila, že každá samostatně stojící budova v mapě je rovněž samostatným objektem v mapové vrstvě.

Porovnávání bude provedeno třikrát, aby byly postiženy nejběžnější kombinace podle charakteru jednotlivých vrstev. Bude testována:

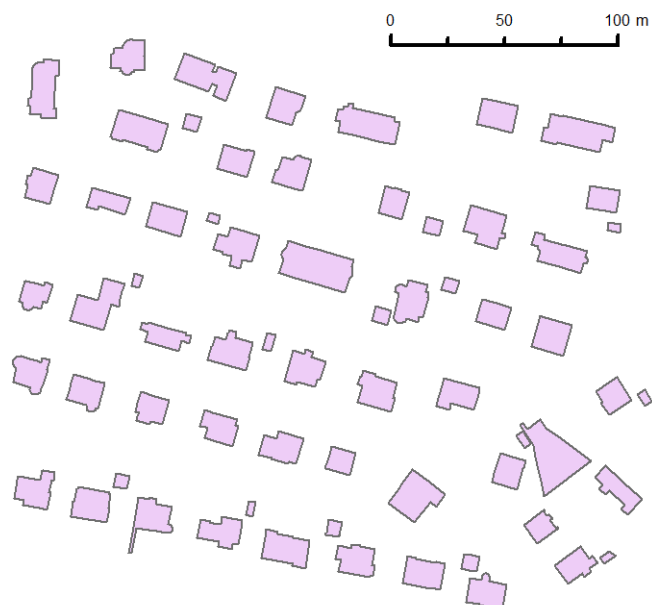
- 1) vrstva CEDA s vrstvou ZABAGED, tedy vrstva hodně generalizovaná s vrstvou se střední úrovní generalizace,
- 2) vrstva ZABAGED s vrstvou KN, tedy vrstva středně generalizovaná s vrstvou přesnou z jiného zdroje,
- 3) vrstva KN s vrstvou Buildings_CR, tedy dvě přesné vrstvy lišící se v detailech.

9.2 Vybraná testovací území

Pro testování nastavení vah byla vybrána 4 území zastupující charakteristické osídlení v českých podmínkách. Protože samotné přiřazení je časově nejnáročnější částí algoritmu a kvůli testování vhodných vah bude muset být provedeno mnohokrát, vybrané území musí mít pouze omezený počet objektů, přibližně 50–100 v každé z vrstev. Byla vybrána tato území reprezentující různé druhy zástavby:

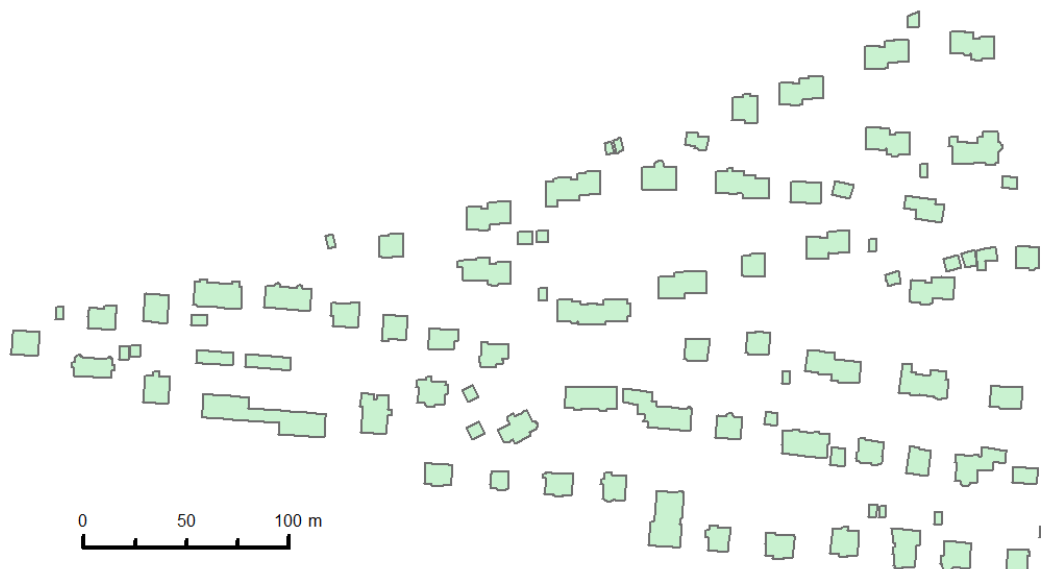
- O – Ořechovka,
- B – Břevnov,
- K – Karlovo náměstí,
- R – Braník.

Ořechovka (O). Vilová čtvrť s malými až středně velkými budovami, jednotlivé budovy se od sebe liší detaily v tvaru i velikosti (Obr. 29). Budovy jsou téměř pravidelně uspořádány.



Obr. 29: Testovací území Ořechovka (O) na vrstvě KN, zdroj: vlastní

Břevnov (B). Vilová čtvrť s menšími domy, většina domů má jednoduchý tvar blízcí se čtverci i podobnou velikost (Obr. 30). Rozmístění domů v území není pravoúhlé.



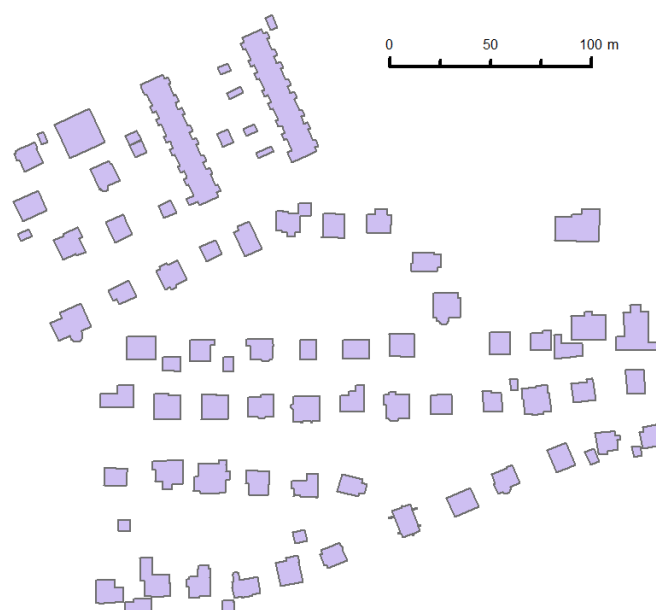
Obr. 30: Testovací území Břevnov (B) na vrstvě KN, zdroj: vlastní

Karlovo náměstí (K). Velké bloky budov se značně odlišnými tvary půdorysu i plochou (Obr. 31). Rozmístění budov v oblasti není pravidelné.



Obr. 31: Testovací území Karlovo náměstí (K) na vrstvě KN, zdroj: vlastní

Braník (R). Rodinné domky v kombinaci s většími bloky budov (Obr. 32). Rozmístění po území je vcelku pravidelné.



Obr. 32: Testovací území Braník (R) na vrstvě KN, zdroj: vlastní

9.3 Testování dílčích metrik

Jak vyplývá z předcházejících kapitol, bude při každém testování vyzkoušeno 12 různých kombinací vrstev a testovacího území – 3 různé kombinace vrstev (1, 2, 3) budou testovány na čtyřech různých územích (O, B, K, R). Nejprve bude testována schopnost přiřadit správně objekty obou vrstev pouze s použitím jedné z metrik. Pro každou kombinaci vrstev a testovacího území byl manuálně vytvořen seznam objektů, které si podle lidského úsudku odpovídají. Při každém přiřazení bude přiřazení jednotlivých objektů porovnáváno s tímto seznamem a bude sledován počet správně přiřazených objektů.

Rotační invariance analýzy tvaru. Podle kapitoly 8.2.1 byly implementovány dvě varianty hodnocení podobnosti dvou TF. Jedna je rotačně invariantní a k tvaru objektu přistupuje tak, že na jeho rotaci nezáleží. To by mohlo být výhodné, pokud u objektu došlo k rotaci například vlivem generalizace. Druhá varianta naopak není invariantní vůči natočení objektu a je možno tak použít natočení jako další vlastnost objektu. To je výhodné, pokud je v mapě hodně objektů s podobným tvarem, které se liší svým natočením. Je třeba otestovat, která varianta je vhodnější.

Tab. 2: Úspěšnost přiřazení při použití metrik samostatně, u každé metriky je uveden počet správných přiřazení (SP) a procentuální úspěšnost (%), zdroj: vlastní

Mapové vrstvy	Území	Možných správných přiřazení	TF rotačně invariantní		TF rotačně variantní		Relativní poloha		Absolutní poloha		Velikost	
			SP	%	SP	%	SP	%	SP	%	SP	%
1. CEDA - ZABAGED	O	32	0	0 %	3	9 %	9	28 %	30	94 %	0	0 %
	B	47	1	2 %	2	4 %	25	53 %	47	100 %	0	0 %
	K	42	6	14 %	9	21 %	40	95 %	42	100 %	4	10 %
	R	41	0	0 %	1	2 %	24	59 %	38	93 %	0	0 %
2. ZABAGED - KN	O	50	3	6 %	13	26 %	50	100 %	50	100 %	1	2 %
	B	59	7	12 %	8	14 %	56	95 %	58	98 %	2	3 %
	K	33	12	36 %	21	64 %	33	100 %	33	100 %	5	15 %
	R	61	3	5 %	6	10 %	59	97 %	61	100 %	3	5 %
3. KN – Buildings_CR	O	62	34	55 %	30	48 %	62	100 %	61	98 %	13	21 %
	B	83	38	46 %	48	58 %	80	96 %	81	98 %	7	8 %
	K	39	27	69 %	29	74 %	38	97 %	39	100 %	19	49 %
	R	75	44	59 %	46	61 %	75	100 %	75	100 %	21	28 %
Průměr			25 %		33 %		85 %		98 %		12 %	

Jak ukazuje Tab. 2 rotačně variantní verze porovnání dvou TF je ve většině případů lepší než verze invariantní. Avšak i tato verze porovnání objektů má průměrnou úspěšnost pouze 33 %. Nejméně úspěšná je analýza tvaru při porovnávání generalizovaných vrstev CEDA a ZABAGED. Ve vrstvě CEDA je velká část budov znázorněna pouze jako obdélník bez dalších detailů, proto zde porovnávání tvaru u většiny případů ztrácí smysl. Naopak nejlepší

úspěšnost je u přesných vrstev KN a Buildings_CR. U těchto přesných vrstev není rozdíl mezi rotačně invariantní a variantní verzí.

Nejúspěšnější metrikou je analýza absolutní pozice, která má průměrnou úspěšnost 98 %. Tato jednoduchá metrika, která využívá polohu centroidů však selhává v místech, kde se centroidy dvou objektů nacházejí velice blízko sebe.

Poměrně úspěšné je i použití relativní polohy, tedy analýza okolí objektu. V přesných vrstvách je její úspěšnost podobná jako úspěšnost absolutní polohy. Jelikož se při jejím výpočtu používá plocha okolních objektů, je slabší v případech generalizovaných vrstev. Generalizace mění tvar a tím výrazně i plochu objektů. Na rozdíl od předchozí metriky ji lze použít i v případech, kdy mapové vrstvy nejsou ve stejném souřadnicovém systému, nebo je jedna z vrstev posunutá vůči druhé, tudíž nelze přímo porovnat souřadnice centroidů.

Nejslabší metrikou v případě samostatného použití je analýza plochy objektů. Sama o sobě nedokáže dostatečně popsat objekt, což je patrné zejména na testovacích územích O a B, kde se nachází velké množství přibližně stejně velikých objektů.

Pro samostatné použití se jeví jako nejúspěšnější analýza absolutní, případně relativní polohy. Testy ukázaly, že analýza tvaru, na kterou je tato práce zaměřena, není samostatně příliš úspěšná při přiřazování objektů z různých mapových vrstev.

9.4 Nalezení vah pro kombinovanou metriku

Zatímco v předchozí kapitole bylo testováno použití dílčích metrik samostatně k přiřazení, v této kapitole bude řešeno jejich použití současně za účelem získání robustní kombinované metriky s co možná nejlepší úspěšností přiřazení objektů. Bude použita rotačně variantní verze podobnosti tvaru, podobnost absolutní polohy, podobnost relativní polohy a podobnost velikosti. Pro tyto čtyři dílčí metriky budou hledány takové váhy, aby jejich vážený průměr použitý k přiřazení objektů byl co možná nejúspěšnější v počtu správně přiřazených objektů.

Každé metrice budou postupně přiřazovány váhy z množiny $\{0, 1, 2, 3, 4\}$, tj. 625 možností různých přiřazení vah. Opět, jako při testování jednotlivých metrik, bude sledován počet správně přiřazených objektů.

Tab. 3: Optimální nastavení vah, váhy jsou uvedeny v pořadí [váha absolutní polohy, váha relativní polohy, váha tvaru, váha plochy], zdroj: vlastní

			Nejlepší nastavení			Univerzální nastavení [4,2,1,1]	
Mapové vrstvy	Území	Možných správných přiřazení	Nejlepší výsledek	Úspěšnost	Nastavení	Nejlepší výsledek	Úspěšnost
1. CEDA - ZABAGED	O	32	31	97 %	[1, 1, 0, 0]	29	91 %
	B	47	47	100 %	[1, 0, 0, 0]	46	98 %
	K	42	42	100 %	[1, 0, 0, 1]	42	100 %
	R	41	38	93 %	[1, 0, 0, 0]	37	90 %
2. ZABAGED - KN	O	50	50	100 %	[1, 0, 0, 0], [0, 1, 0, 0]	50	100 %
	B	59	59	100 %	[0, 3, 0, 1]	59	100 %
	K	33	33	100 %	[1, 0, 0, 0], [0, 1, 0, 0]	33	100 %
	R	61	61	100 %	[1, 0, 0, 0]	60	98 %
3. KN – Buildings_CR	O	62	62	100 %	[0, 1, 0, 0]	62	100 %
	B	83	83	100 %	[1, 0, 0, 1]	81	98 %
	K	39	39	100 %	[1, 0, 0, 0]	39	100 %
	R	75	75	100 %	[0, 1, 0, 0], [1, 0, 0, 0]	75	100 %
Průměr			99 %			98 %	

Tab. 3 ukazuje optimální nastavení vah vycházející z testů. V první části tabulky je zobrazen nejlepší výsledek s použitím všech možných nastavení vah. Jsou vypsána některá základní nastavení, se kterými, bylo tohoto výsledku dosaženo. Z výsledků je patrné, že pro dobrý výsledek je nezbytná vysoká váha absolutní polohy. V některých případech je však lepších výsledků dosahováno s využitím relativní polohy. Rovněž některá nejlepší přiřazení používají též podobnost velikosti objektů. To může být výhodné v těch případech, kdy je plošně velká budova ve tvaru U a v jejím středu se nachází další malá budova. Takovéto budovy mají blízké centroidy a právě metrika využívající plochu objektu je může od sebe odlišit.

Druhá část tabulky pak ukazuje výsledky s využitím univerzálního nastavení. Takto stanovené nastavení má celkově největší počet přiřazených objektů a celkově na všech testovacích souborech dosahuje průměrnou úspěšnost 98 %. Tento výsledek je téměř shodný s úspěšností samotné absolutní polohy. Z těchto výsledků vyplývá, že pokud jsou vrstvy přesné a lze mezi nimi určovat vzdálenost, je nejjednodušší variantou použití absolutní polohy doplněné nějakou jednoduchou charakteristikou tvaru, která by ošetřila případy, kdy dva tvarově odlišné objekty mají centroid blízko sebe. Toto zjištění odpovídá výsledkům dosavadních výzkumů, uvedených v kapitole 2.5.

Pokud nejsou obě mapové vrstvy v převoditelném souřadnicovém systému, není možno použít analýzu absolutní polohy. V takovém případě totiž není možné měřit přímo vzdálenost mezi centroidy objektů. Pokud však znám alespoň měřítko a stočení obou

mapových vrstev, je možné využít zbývajících tří metrik. Výsledky v takovém případě shrnuje Tab. 4.

Tab. 4: Optimální nastavení vah bez absolutní polohy, váhy jsou uvedeny v pořadí [0, váha relativní polohy, váha tvaru, váha plochy], zdroj: vlastní

			Nejlepší nastavení			Univerzální nastavení [0,4,1,2]	
Mapové vrstvy	Území	Možných správných přiřazení	Nejlepší výsledek	Úspěšnost	Nastavení	Nejlepší výsledek	Úspěšnost
1. CEDA - ZABAGED	O	32	11	34 %	[0, 1, 0, 3], [0, 1, 0, 4]	5	16 %
	B	47	26	55 %	[0, 3, 0, 2], [0, 4, 1, 0]	24	51 %
	K	42	41	98 %	[0, 3, 0, 1], [0, 4, 0, 1]	39	93 %
	R	41	24	59 %	[0, 1, 0, 0]	20	49 %
2. ZABAGED - KN	O	50	50	100 %	[0, 1, 0, 0]	50	100 %
	B	59	59	100 %	[0, 3, 0, 1], [0, 4, 0, 1]	58	98 %
	K	33	33	100 %	[0, 1, 0, 0], [0, 4, 1, 0]	32	97 %
	R	61	52	85 %	[0, 1, 0, 0], [0, 4, 0, 1]	58	95 %
3. KN – Buildings_CR	O	62	62	100 %	[0, 1, 0, 0], [0, 2, 0, 1]	62	100 %
	B	83	82	99 %	[0, 1, 0, 1], [0, 2, 0, 1]	82	99 %
	K	39	38	97 %	[0, 1, 0, 0]	38	97 %
	R	75	75	100 %	[0, 1, 0, 0]	75	100 %
Průměr			86 %			83 %	

S použitím optimálního nastavení může být i přiřazení bez absolutní polohy velice přesné. Výjimku tvoří hodně generalizované vrstvy (1. CEDA – ZABAGED) na územích, kde se nachází podobně velké budovy (O, B, R). Ve všech ostatních případech je úspěšnost přiřazení větší než 90 %.

V některých případech však nemusí být možné použít ani relativní polohu. To například nastane u mapových vrstev s výrazně odlišným výběrem prvků. Okolní prvky tak mohou být výrazně odlišné a porovnání těchto okolí mezi sebou nedává smysl. Rovněž odlišné (a neznámé) stočení obou vrstev by neumožňovalo použití podobnosti relativní polohy, protože je tato metoda závislá na měření azimutu. V takovém případě nezbyvá než použít pouze podobnost tvaru a plochy, jak to ukazuje Tab. 5.

Tab. 5: Optimální nastavení vah bez absolutní i relativní polohy, váhy jsou uvedeny v pořadí [0, 0, váha tvaru, váha plochy], zdroj: vlastní

			Nejlepší nastavení			Univerzální nastavení [0,0,1,1]	
Mapové vrstvy	Území	Možných správných přiřazení	Nejlepší výsledek	Úspěšnost	Nastavení	Nejlepší výsledek	Úspěšnost
1. CEDA - ZABAGED	O	32	3	9 %	[0, 0, 1, 0]	1	3 %
	B	47	2	4 %	[0, 0, 1, 0]	2	4 %
	K	42	21	50 %	[0, 0, 1, 1]	21	50 %
	R	41	3	7 %	[0, 0, 2, 1], [0, 0, 3, 1]	1	2 %
2. ZABAGED - KN	O	50	17	34 %	[0, 0, 4, 1]	15	30 %
	B	59	11	19 %	[0, 0, 3, 1], [0, 0, 4, 1]	9	15 %
	K	33	26	79 %	[0, 0, 3, 1], [0, 0, 4, 1]	24	73 %
	R	61	10	16 %	[0, 0, 1, 1]	10	16 %
3. KN – Buildings-CR	O	62	42	68 %	[0, 0, 1, 2], [0, 0, 1, 3]	40	65 %
	B	83	49	59 %	[0, 0, 3, 4]	48	58 %
	K	39	36	92 %	[0, 0, 3, 2]	34	87 %
	R	75	58	77 %	[0, 0, 1, 2]	57	76 %
Průměr				43 %			41 %

Přiřazení dle tvaru a plochy má nejlepší výsledky v přesných vrstvách – průměrně 70% úspěšnost. Naopak zcela selhává v generalizovaných vrstvách, obzvláště v oblastech typově podobných domů.

9.5 Podobnost testovacích vrstev

S použitím optimálních vah vzešlých z předchozí kapitoly, tedy [4, 2, 1, 1] je možné přiřadit objekty dvou libovolně velkých vrstev. K tomu je využíváno rozdělení společného prostoru mapových vrstev do dlaždic a jejich následné postupné zpracovávání, jak bylo navrženo v kapitole 7.3. Byly provedeny tři různá přiřazení různých mapových vrstev zahrnujících velkou část Prahy a okolí. Na přiřazených objektech byla dále zhodnocena podobnost obou mapových vrstev na základě postupu popsaného v kapitole 7.4. Výsledky tohoto zhodnocení shrnuje Tab. 6. Připomeňme, že i zde platí, že nepodobnost je blízká jedné, zatímco podobnost nule.

Jako celkově nejpodobnější ve všech hlediskách byly shledány podrobné vrstvy KN a Buildings_CR. Naopak celkově nejméně podobné jsou si vrstvy CEDA a ZABAGED. Ty se výrazně liší hlavně po globální a lokální stránce. Co se týče tvarové individuální podobnosti, pak nejhorší výsledky jsou při srovnání přesné vrstvy KN s vrstvou ZABAGED. Lze však předpokládat, že pokud bychom testovali KN s velmi generalizovanou vrstvou CEDA, byly by si objekty tvarově ještě nepodobnější.

Tab. 6: Výsledky porovnání mapových vrstev, zdroj: vlastní

	CEDA – ZABAGED	ZABAGED – KN	KN – Buildings_CR
Nepodobnost individuální	0,25	0,32	0,18
Nepodobnost lokální	0,13	0,07	0,03
Nepodobnost globální	0,19	0,10	0,01
Celková nepodobnost	0,19	0,17	0,07

9.6 Zhodnocení navrženého algoritmu

V této podkapitole bude diskutována kvalita dosažených výsledků a efektivita implementace algoritmu. Budou zde rovněž řešena omezení algoritmu.

Zhodnocení dosažených výsledků. Nejlepší dosažené výsledky přiřazení objektů na dvou mapových vrstvách uložených ve stejném souřadnicovém systému, kde je možné použít kombinaci všech čtyř navržených dílčích metrik, shrnuje Tab. 3. Při použití univerzálního nastavení je dosaženo 98% úspěšnosti přiřazení. Tato úspěšnost přiřazení objektů je dostačující pro další zhodnocení podobnosti mapových vrstev. Velkou měrou je takto dobré přiřazení způsobeno použitím metriky pro absolutní polohu, která měla ze všech metrik nejlepší výsledky, viz Tab. 2.

V případě, že není možné používat podobnost absolutní polohy, tj. v případech odlišných nepřevoditelných souřadnicových systémů obou vrstev, byla průměrná úspěšnost s použitím zbývajících tří metrik při univerzálním nastavení 83 %. Tento výsledek je do velké míry zkreslován špatnými výsledky při přiřazování hodně generalizované vrstvy. Na přesnějších vrstvách dosahuje i tento postup úspěšnosti 98 %. Na tomto výsledku má největší vliv použití podobnosti relativní polohy, která, jak se ukázalo v samostatných testech (viz Tab. 2), má na přesnějších vrstvách úspěšnost srovnatelnou s podobností absolutní polohy. Navíc je invariantní vůči posunu a lze ji použít na vrstvách o odlišném souřadnicovém systému, pokud nedochází k přílišnému zkreslení délek. Podmínkou však zůstává shodné natočení obou systémů.

V testech, kde nebyly použity podobnosti absolutní ani relativní polohy, byly dosažené výsledky výrazně horší. Průměrný výsledek pouze s použitím podobnosti tvaru a velikosti byl 40 %. Zde je ještě výrazněji ovlivňován rozdílnou generalizací mapových vrstev, protože

na dvou nejpodrobnějších, nejméně generalizovaných vrstvách, byla dosažena průměrná úspěšnost 70 %.

Z výše shrnutých výsledků vyplývá, že lze s pomocí navrženého algoritmu dosáhnout velice dobrých výsledků přiřazení objektů dvou mapových vrstev. Největší vliv na tomto přiřazení má použití podobnosti absolutní a relativní polohy. Potvrzuje se tak hypotéza, naznačená v kapitole 2.2, že v mapových vrstvách je poloha objektu důležitější než jeho tvar. Tvar shodných objektů v různých mapových vrstvách se často liší, zatímco poloha zůstává téměř stejná. Proto úspěšnost použití samostatné podobnosti tvaru byla mnohem menší než při použití podobností absolutní či relativní polohy. Lepší výsledky byly při použití podobnosti tvaru dosaženy pouze na přesných, velice málo generalizovaných, vrstvách.

Podstatný je rovněž fakt, že podobnost absolutní polohy dává při samostatném použití stejně dobré výsledky jako použití všech dílčích metrik při univerzálním nastavení, tj. 98 %. To ve své podstatě potvrzuje dosavadní výzkumy v geoinformatice, shrnuté v kapitole 2.5, kde ve většině z nich jsou pro přiřazení objektů používány jednoduché metody založené na absolutní poloze objektů.

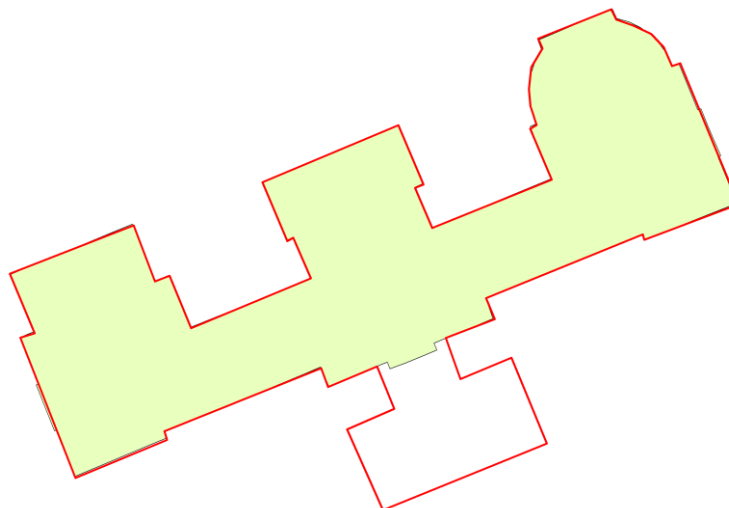
V případě, že z nějakých důvodů není možné použít metody postavené na absolutní poloze, je dobrou alternativou použití podobnosti relativní polohy objektů. Tato metoda, představená v rámci této práce, je ve své podstatě analogií na shape deskriptor Shape Context. Ten je používán na přiřazení bodů dvou objektů, zatímco mnou navržená metoda je používána k přiřazení celých objektů na základě jejich centroidů. Princip je však podobný. Shape Context popisuje pozici každého bodu ve vztahu k ostatním bodům daného tvaru. Moje podobnost relativní polohy popisuje pozici každého objektu ve vztahu k okolním objektům. V obou případech jsou okolní body (respektive objekty) popisovány na základě jejich polárních souřadnic. Jak bylo zmíněno v předchozích odstavcích, má tato metoda obdobnou účinnost jako podobnost absolutní polohy, je však robustnější a použitelná ve více případech.

Jak dokazují Tab. 2, 3 a 4, použití podobnosti tvaru a velikosti nepřineslo k metrikám podobnosti absolutní a relativní polohy žádné výrazné zlepšení úspěšnosti. Z tohoto důvodu je vhodné podobnost tvaru objektu používat pouze jako metodu doplňkovou k metodám využívajících polohu objektů. Výjimkou mohou být případy, kdy není znám souřadnicový systém, stočení mapové vrstvy nebo není možné určovat absolutní délky. Kromě podobnosti tvaru jsou všechny ostatní zkoumané dílčí metriky podobnosti totiž na měření délek závislé.

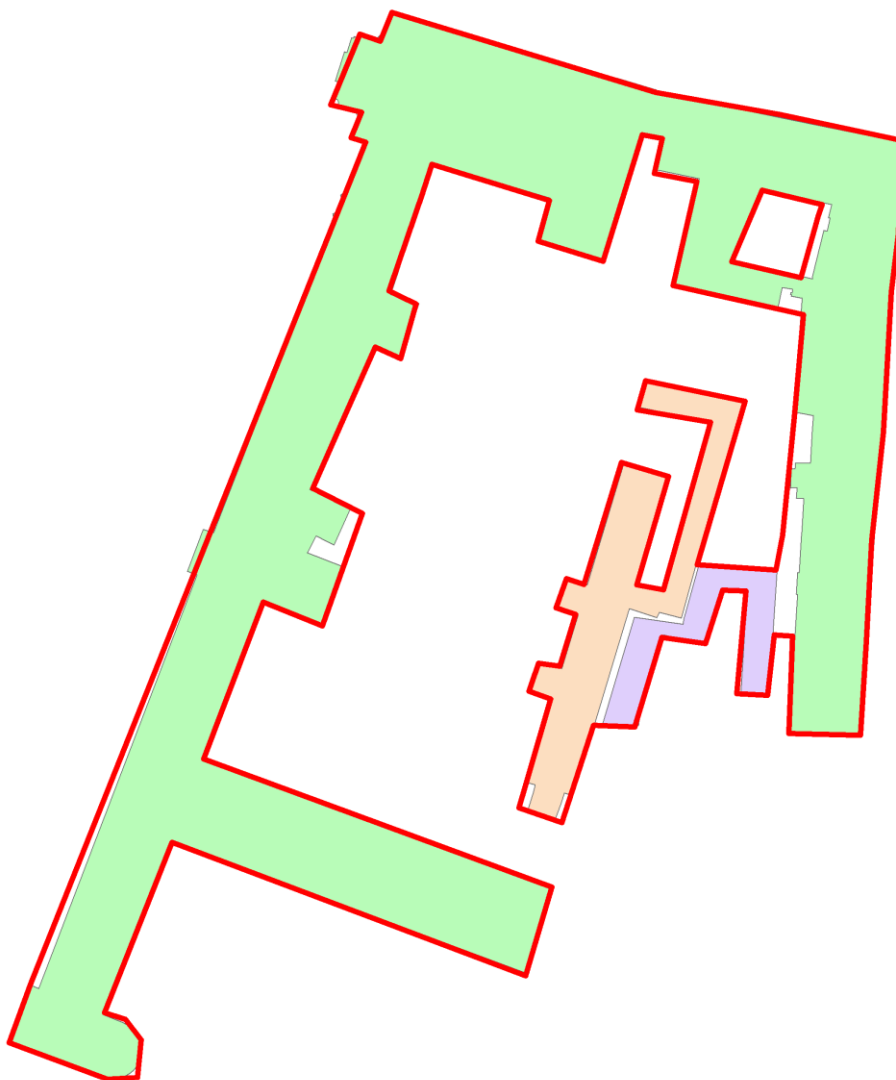
Použití podobnosti absolutní a relativní polohy by rovněž bylo nemožné, kdyby bylo třeba přiřadit samostatný objekt, jehož absolutní poloha ani jeho okolí by nebylo známé.

Použití shape deskriptoru turning function jako hlavní metody pro přiřazování objektů má opodstatnění hlavně v případech, kdy není přesně známá absolutní ani relativní poloha objektu na jedné z vrstev. Určení podobnosti polygonů prostřednictvím tohoto deskriptoru selhávalo převážně ve dvou typech případů. Jeden z nich nastává, pokud je porovnáváno větší množství tvarově podobných jednoduchých nebo hodně generalizovaných polygonů. Všechny tyto polygony mají podobný tvar, zpravidla obdélník, a jejich TF je tak velice podobná. Částečným řešením v takovém případě (kromě použití polohy objektu) je použití podobnosti velikosti objektu. Velikost objektu je však možné použít, jen pokud jsou známá alespoň měřítka obou mapových vrstev.

Druhým z případů, v kterých turning function měla neuspokojivé výsledky, byla odlišná agregace objektů (Obr. 34) nebo vynechání části objektu (Obr. 33) v jedné z vrstev. Tento problém není v navrženém algoritmu řešen, avšak bylo by ho možné řešit modifikací turning function takovým způsobem, aby dokázala posuzovat podobnost jen v části tvaru, jak to navrhuje Cohen a Guibas (1997).



Obr. 33: Vynechání částí objektu, zdroj: vlastní



Obr. 34: Agregace objektů – tři objekty (zelený, fialový, oranžový) jsou v druhé vrstvě agregovány do jednoho (červený obrys), zdroj: vlastní

Omezení algoritmu. V současné podobě má implementovaný algoritmus několik omezení vyplývajících ze způsobu jeho tvorby a výpočtu. Algoritmus vyžaduje na svém vstupu soubory dat s korektně uloženými polygony. Musí se jednat o simple polygony s korektním pořadím vertexů, tedy s vertexy uloženými po směru hodinových ručiček. Algoritmus nebere v úvahu díry v polygonech s dírou. Modifikace algoritmu tak, aby i tyto díry byly posuzovány, by nebyla příliš náročná. Tato modifikace by však mohla vyvolat komplikace v případě, že by u shodného objektu neodpovídal počet děr v obou vrstvách.

Zejména kvůli realizaci podobnosti absolutní polohy, kde je měřena vzdálenost centroidů, je pro fungování programu nutné, aby vstupní vrstvy byly uloženy ve stejném metrickém souřadnicovém systému. Pro použití jiných souřadnicových systémů by bylo nutné program

modifikovat. Problémem je zejména výpočet vzdáleností přímo z metrických souřadnic bodů. Při použití jiných souřadnicových systémů, kde jsou souřadnice bodů uloženy například ve stupních, by bylo nutné doplnit funkce pro výpočet vzdáleností v těchto systémech.

Dalším z omezení algoritmu je jeho omezená schopnost řešit problémy plynoucí z odlišné agregace objektů. Aby se těmto problémům zamezilo, jsou na vstupu požadovány takové vrstvy, kde všechny objekty, které se dotýkají, jsou agregovány v jeden, a zároveň jeden objekt nesmí mít více oddělených částí. Algoritmus by bylo možné modifikovat tak, aby zkoumal různé možné kombinace agregací objektů a použil tu s největší podobností, jak navrhuji Gösseln a Sester (2004). Takováto modifikace by však byla poměrně náročná a je nad rámec této práce.

Zhodnocení implementace algoritmu. Implementace algoritmu proběhla úspěšně v programovacím jazyku Python. Program využívá některé volně dostupné knihovny pro Python a zároveň i knihovnu `arcpy` distribuovanou jako součást komerčního softwaru ArcGIS. Tato knihovna je používána především pro načítání uložených prostorových dat. V případě potřeby použití nezávislém na softwaru ArcGIS by bylo možné tuto část upravit s využitím některé volně dostupné knihovny pro práci s prostorovými daty.

Slabinou navrženého algoritmu je jeho výpočetní náročnost. V celém návrhu algoritmu jsou dvě časově náročné části, které celý běh algoritmu výrazně zpomalují. První z nich je určení vzdálenosti mezi dvěma TF, která má pro každou dvojici porovnávaných objektů o l a m vertexech časovou složitost $O(lm(l+m))$. Tu by bylo možné částečně snížit použitím implementačně náročnějšího, avšak výkonnějšího algoritmu popsáno v kapitole 5.2, který navrhuji Arkin et al. (1990) a jenž má složitost $O(lm \log lm)$. Druhou částí je použití Hungarian algorithm k přiřazování objektů. Ta má i ve své nejefektivnější variantě časovou složitost $O(n^3)$ pro přiřazení dvou stejně velkých datasetů o n objektech. Zrychlení je možné dosáhnout porovnáváním vždy menšího počtu objektů najednou, což je umožněno rozdělením území do většího počtu menších dlaždic.

10 DISKUZE A ZÁVĚR

Diplomová práce se zabývá tématem automatizované analýzy podobnosti map. Jejím hlavním cílem bylo navrhnout postup pro automatickou analýzu podobnosti map s využitím metod na rozpoznávání tvaru s ohledem na různé odlišnosti v analyzovaných mapových vrstvách.

V první části práce byly představeny některé metody používané pro rozpoznávání tvarů společně s přehledem jejich dosavadního využití v geoinformatice. Byly také shrnuty odlišnosti, které mohou nastat při porovnání dvou map. Jednalo se zejména o možné odlišnosti v měřítku, generalizaci a kartografickém zobrazení mapy.

V dalších částech práce byla podobnost map transformována na podobnost objektů v nich zobrazených a byly hledány metody zabezpečující správné přiřazení odpovídajících si objektů na obou mapách a jejich následné porovnání. Podobnost objektů byla posuzována pomocí čtyř metod jako podobnost tvarů objektů určená prostřednictvím turning function, podobnost polohy centroidů objektů, podobnost okolí objektů a podobnost velikosti objektů. Byly diskutovány výhody a omezení jednotlivých metod. Výsledná metoda pro podobnost objektů je kombinací všech čtyř zkoumaných metod.

Jednotlivé metody byly implementovány v jazyce Python a spojeny do aplikace umožňující automatizovanou analýzu podobnosti polygonových mapových vrstev. Aplikace určí podobnost objektů ve dvou mapových vrstvách a na jejím základě přiřadí odpovídající si objekty k sobě. Po přiřazení objektů dokáže na základě několika hledisek určit podobnost obou vrstev.

Aplikace byla testována na čtyřech různých mapových vrstvách budov v rámci čtyř testovacích území. Průměrná úspěšnost přiřazení objektů na těchto územích pomocí optimální kombinace metod byla 98 %. K této úspěšnosti největší měrou přispělo použití polohové informace posuzovaných objektů. Při samostatném posouzení jednotlivých metod vykazuje nejlepší průměrnou úspěšnost 98 % podobnost absolutní polohy centroidů objektů, druhou nejlepší úspěšnost 86 % má podobnost okolí objektu. Podobnost tvaru pomocí turning function dosahuje ve spojení s podobností velikosti objektu úspěšnosti 40 %.

K přiřazování objektů se jako nejlepší jeví použití podobnosti polohy obou objektů doplněné charakteristikou tvaru a velikosti objektu. Charakteristika tvaru a velikosti může sloužit jako doplňková metoda k odlišení tvarově odlišných blízko sebe ležících objektů. Pro přesné

mapy shodných zobrazení lze použít podobnost absolutní polohy centroidů obou objektů. Pro ostatní mapy, u nichž známe alespoň měřítko a stočení mapy, je vhodné používat podobnost relativní polohy, která byla představena v této práci. Její výsledky jsou pouze nepatrně horší než u absolutní polohy, ale na rozdíl od ní je invariantní vůči posunu celé vrstvy. Mnou představená metrika relativní polohy není však invariantní vůči rotaci a změně měřítka. Takové invariance by výrazně zvýšily robustnost této metody a mohly by se stát předmětem dalšího výzkumu.

Relativně nízká úspěšnost přiřazení objektů pomocí posouzení tvaru je zapříčiněna několika hlavními aspekty. Tvary jednotlivých posuzovaných budov si jsou často velmi podobné a tvar je při generalizaci podstatně měněn. Implementovaná metoda posouzení tvaru není schopna od sebe odlišit tvarově jednoduché objekty, typicky obdélníky, i přesto, že se výrazně liší svou velikostí. Tento problém byl řešen použitím zvláštní metriky pro velikost objektu. Lepším řešením by mohla být modifikace turning function takovým způsobem, aby nebyla invariantní vůči změně měřítka. Pro použití takto modifikovaného deskriptoru by byla nutná alespoň znalost měřítka obou porovnávaných vrstev. Další problém se objevoval v případech, kdy se tvar objektu obou vrstev shodoval jenom částí. To nastávalo například při agregaci objektů nebo při vynechání části objektu v jedné z vrstev. Řešením tohoto problému by mohla být implementace turning function tak, aby dokázala nejen testovat celé tvary, ale i přiřazovat pouze části tvarů.

Obecně lepší výsledky posouzení podobnosti tvaru jsou dosahovány při posuzování tvarově složitějších objektů. Použití této metody není vhodné pro mapové vrstvy obsahující převážně objekty s malým počtem vertexů. Naopak při použití na mapových vrstvách s tvarově odlišnějšími složitějšími objekty by výsledky tohoto shape invariantu byly výrazně lepší.

Mezi hlavní přínosy této práce lze zahrnout stanovení a otestování dílčích metod určení podobnosti objektů. Nadějně se jeví zejména použití metody určení podobnosti objektů na základě jejich okolí v případech, kdy z nějakého důvodu není možné použít absolutní podobnost polohy pomocí vzdálenosti centroidů. Důležitým přínosem je i stanovení nejčastějších problémů vyskytujících se při použití turning function na polygonových mapových vrstvách. Další výzkum v této oblasti by se mohl zabývat jednak zajištěním potřebných invariancí u představené metriky relativní polohy, jednak i zajištěním lepší funkčnosti analýzy tvaru u agregovaných objektů.

11 REFERENCE

- ABBASI, S., MOKHTARIAN, F., KITTLER, J. (1999). Curvature scale space image in shape similarity retrieval. *Multimedia systems*, 7, č. 6, s. 467-476.
- ADAMEK, T., O'CONNOR, N. E. (2004). A multiscale representation method for nonrigid shapes with a single closed contour. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14, č. 5, s. 742-753.
- ALAJLAN, N., et al. (2007). Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*, 40, č. 7, s. 1911-1920.
- ALT, H., GUIBAS, L. J. (1999). Discrete Geometric Shapes: Matching, Interpolation, and Approximation: A Survey. *Handbook of Computational Geometry*, s. 121-153.
- ALT, H., KNAUER, C., WENK, C. (2004). Comparison of distance measures for planar curves. *Algorithmica*, 2004, 38, č. 1, s. 45-58.
- ARKIN, M., CHEW, L. P., HUTTENLOCHER, D. P., KEDEM, K., MITCHELL, J. S. B. (1990). An Efficiently Computable Metric for Comparing Polygonal Shapes. In: *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1990, s. 129-137.
- BARTOLINI, I., CIACCIA, P., PATELLA, M. (2005). Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27, č. 1, s. 142-147.
- BAI, X., YANG, X., LATECKI, L. J., WENYU, L., ZHUOWEN, T. (2010). Learning context-sensitive shape similarity by graph transduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32, č. 5, s. 861-874.
- BAYER T. (2014). Estimation of the unknown cartographic projection from the map. 2014, v tisku.
- BELONGIE, S., MALIK, J., PUZICHA, J. (2002). Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24, č. 4, s. 509-522.
- BESL, P. J., MCKAY, N. D. (1992). Method for registration of 3-D shapes. In: *Robotics-DL tentative*. International Society for Optics and Photonics, s. 586-606.

- BUTENUTH, M., et al. (2007). Integration of heterogeneous geospatial data in a federated database. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62, č. 5, s. 328-346.
- CLAPPER, B. M. (2008). Munkres 1.0.6. Modul pro Python, dostupný jako open-source v rámci licence BSD. [cit. 4. 8. 2014] Dostupný z WWW: <<https://pypi.python.org/pypi/munkres/>>.
- COHEN, S. D., GUIBAS, L. J. (1997) Partial matching of planar polylines under similarity transformations. In: *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, s. 777-786.
- DEVOGELE, T. (2002). A new Merging process for data integration based on the discrete Fréchet distance. In: *Advances in spatial data handling*. Springer Berlin Heidelberg, s. 167-181.
- DOYTSHER, Y., FILIN, S., EZRA, E. (2001). Transformation of datasets in a linear-based map conflation framework. *Surveying and Land Information Systems*, 61, č. 3, s. 165-176.
- FRANK, R., ESTER, M. (2006). A Quantitative Similarity Measure for Maps. *Progress in spatial data handling*. Springer Berlin Heidelberg, 2006, s. 435-450.
- GÖSSELN, G., SESTER, M. (2004). Integration of geoscientific data sets and the german digital map using a matching approach. *International Archives of Photogrammetry and Remote Sensing*, 35, s. 1249-1254.
- HUH, Y., YU, K., HEO, J. (2011). Detecting conjugate-point pairs for map alignment between two polygon datasets. *Computers, Environment and Urban Systems*, 35, č. 3, s. 250-262.
- JONES, E., OLIPHANT, E., PETERSON, P., et al. (2014): *SciPy: Open Source Scientific Tools for Python*, 2001-. [cit. 4. 8. 2014] Dostupné z WWW: <<http://www.scipy.org/>>.
- LATECKI, L. J., LAKAMPER, R., ECKHARDT, T. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. IEEE, s. 424-429.
- LI, L., GOODCHILD, M. F. (2012). Automatically and accurately matching objects in geospatial datasets. *Advances in Geo-Spatial Information Science*, 10, s. 71-79.

LING, H., JACOBS, D. W. (2005). Using the inner-distance for classification of articulated shapes. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, s. 719-726.

MUNKRES, J. (1957): Algorithms for the Assignment and Transportation Problems, Journal of the Society for Industrial and Applied Mathematics, č. 5, 1957, s. 32-38.

O'ROURKE, J. (1998): Computational Geometry in C. Cambridge University Press, 1998, 392 s., ISBN: 9780521649766.

RUSINKIEWICZ, S., LEVOY, M. (2001). Efficient variants of the ICP algorithm. In: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on. IEEE, s. 145-152.

VELTKAMP, R. C. (2001). Shape matching: Similarity measures and algorithms. In: Shape Modeling and Applications, SMI 2001 International Conference on. IEEE, s. 188-197.

Vivid Solutions (2003). Java Conflation Suite (JCS). [cit. 16. 7. 2014] Dostupné z WWW: <<http://www.vividsolutions.com/JCS/>>.

VOLZ, S. (2006). An iterative approach for matching multiple representations of street data. In: Proceedings of the JOINT ISPRS Workshop on Multiple Representations and Interoperability of Spatial Data, Hannover, s. 101-110.

WALTER, V., FRITSCH, D. (1999). Matching spatial data sets: a statistical approach. International Journal of Geographical Information Science, 13, č. 5, Taylor & Francis, London, s. 445-473.

WEISSTEIN, E. W. (2014): Voronoi Diagram. MathWorld - A Wolfram Web Resource. [cit. 8. 8. 2014]. Dostupné z WWW: <<http://mathworld.wolfram.com/VoronoiDiagram.html>>.

ZHANG, D., LU, G. (2004). Review of shape representation and description techniques. Pattern recognition, 37, č.1, s. 1-19.

ZHILIN, L. (2007). Algorithmic Foundation of Multi-scale Spatial Representation. CRC Press, Taylor & Francis Group. 280 s., ISBN: 0849390729.