

Tvarové metriky v geoinformatice

Shape Metrics in Geoinformatics

Lukáš Maloň

Bakalářská práce

Vedoucí práce: doc. Mgr. Jiří Dvorský, Ph.D.

Ostrava, 2021

Abstrakt

Bakalářská práce se zaměřuje na tvarové metriky v geoinformatice. V úvodu si představíme geoinformatiku a její souvislost s tvarovými metrikami. Představíme si způsoby využití a rozdělení tvarových metrik. Rozebereme si některé z existujících softwarových nástrojů vytvořených k výpočtům metrik. V praktické části si ukážeme implementaci knihovny metrik v programovacím jazyce C# a návrh vlastní metriky založené na želví grafice. Na závěr se budeme zabývat testováním knihovny a vyhodnocováním výsledků testování.

Klíčová slova

tvarové metriky; .NET knihovna; želví grafika

Abstract

This thesis focuses on Shape metrics in Geoinformatics. At first I will get you acquainted with geoinformatics and its relationship with shape metrics. I will introduce to you variety of ways in which metrics are used and their classifications. Another part aims on introducing some of existing software created for metric calculating. In practical part I will show you implementation of newly created shape metrics library in C# programming language and new metric based on turtle graphics. In conclusion I will show you some testing of my library and evaluation of testing results.

Keywords

shape metrics; .NET library; turtle graphics

Poděkování

Chtěl bych poděkovat panu doc. Mgr. Jiřímu Dvorskému, Ph.D. za odborné vedení, za pomoc a rady při zpracování mé bakalářské práce.

Obsah

Seznam obrázků	6
Seznam tabulek	7
Seznam výpisů zdrojového kódu	8
1 Úvod	9
2 Využití metrik	11
3 Rozdělení tvarových metrik	13
3.1 Patch level	13
3.2 Class level	14
3.3 Landscape level	14
4 Software k výpočtu metrik	15
4.1 FRAGSTATS	15
4.2 Shape Metrics Tool	19
4.3 landscapemetrics	21
4.4 Vector-based Landscape Analysis Tools (V-LATE)	21
4.5 Další nástroje	24
5 Implementace vlastní knihovny	26
5.1 Třída Map	26
5.2 Knihovna metrik	30
6 Návrh vlastní metriky	34
7 Testování knihovny	36
7.1 Rovinné obrazce	36
7.2 Iterace rovinných fraktálů	37

7.3 Metriky jádrových oblastí	39
8 Závěr	41
Literatura	42
Přílohy	44
A Třídní diagram Knihovny metrik	45
B Třídní diagram Želví metriky	47

Seznam obrázků

3.1	Znázornění tří kategorií tvarových metrik na mapě [13]	14
4.1	FRAGSTATS metriky, první část	16
4.2	FRAGSTATS metriky, druhá část	17
4.3	Radius of gyration, h_{ijr} označuje vzdálenost buňky od středu plošky, z je počet buněk plošky	18
4.4	Shape Metrics Tool metriky, první část [11]	22
4.5	Shape Metrics Tool metriky, druhá část [11]	23
5.1	Vlevo neumannovo sousedství, vpravo moorovské sousedství. Modrou barvou je označena buňka, jejíž sousedství aktuálně zkoumáme. Zelenou barvou jsou vyznačeni její sousedé. [19]	27
5.2	Komponenty grafu a mapa	27
5.3	Průchod mapou do šířky po vlnách. Postupně jsou procházeny všechny buňky stejného odstínu, poté přejdeme na buňky o odstín světlejší. Počítáme s tím, že všechny buňky na obrázku náležejí prohledávané plošce.	28
6.1	Průchod želvy ploškou s výsledným řetězcem.	35
7.1	Vybrané rovinné obrazce	36
7.2	Šest iterací dvou vybraných fraktálů	38
7.3	Porovnání metriky Area s jádrovou metrikou Core Area	40
A.1	Hierarchie knihovny metrik	46
B.1	Využití návrhového vzoru kompozit v želví metrice.	48

Seznam tabulek

7.1	Výpočet metrik na rovinných obrazcích	37
7.2	Výpočet metrik šesti iterací Kochovy vločky	38
7.3	Výpočet metrik šesti iterací Box fraktálu	39
7.4	Porovnání metrik jádrových oblastí při odebrání pásu určité šířky	40

Seznam výpisů zdrojového kódu

5.1	Algoritmus prohledávání plošky	28
5.2	Výpočet metriky <i>Gyrate</i>	31
5.3	Výpočet metriky <i>Contig</i>	32

Kapitola 1

Úvod

Od počátků vývoje se člověk pohybuje v prostoru, kterým je na světě obklopen. Nejdříve v něm pouze přežíval, ale postupem času se o něj začal více zajímat a zkoumat jeho zákonitosti. Rozvoj lidské společnosti přivedl člověka také k přímému ovlivňování prostoru, a to chtěnému i nechtěnému v pozitivním i negativním směru. Člověk si stále více uvědomuje svůj razantní vliv na reálný svět. Dnes je jeho cílem svět okolo sebe aktivně a v první řadě pozitivně ovlivňovat, jednak z důvodu nápravy již existujících negativních důsledků lidských činností, ale také kvůli zachování potenciálu světa k dalšímu rozvoji lidské společnosti. K těmto účelům se začal rozvíjet vědní obor, který se zabývá zpracováváním informací souvisejících s prostorovým uspořádáním reálného světa. Tímto oborem je geoinformatika.

Geoinformatika je věda, která se jako samostatná disciplína rozvíjí již několik desetiletí. Čerpá ze znalostí jiných vědních oborů, které se zabývají poznáváním reálného světa mezi kterými můžeme jmenovat geografii, geologii, kartografii, geodézii a další. Znalosti z těchto oborů kombinuje se znalostmi technických oborů jako jsou informatika, databázové a informační systémy nebo počítačová grafika.

Geoinformatika využívá znalosti těchto disciplín k řešení geografických problémů se zájmem o získávání geografických dat, zkoumání prostorového uspořádání objektů, jejich vizualizaci a ukládání nebo tvorbu digitálních modelů reálného světa. Prostorová data reprezentována formou digitálních map, leteckých snímků, tabulek apod. můžeme analyzovat kvalitativně nebo kvantitativně dnes již efektivně s pomocí geoinformačních systémů. Mezi kvantitativní metody analýzy spadají právě tvarové metriky.

Tvarové metriky nejsou celistvě shrnuty v žádné publikaci, která by souhrnně popisovala jejich význam. Jedná se o veličiny vyjadřující hodnotu specifické tvarové vlastnosti objektu či jevu reálného světa nebo vzájemného polohového vztahu mezi nimi. Tyto hodnoty můžeme aplikovat na objekty různých tvarů a vzájemně je porovnávat. Následně ze srovnání vyvozovat výsledky. Právě díky rozvoji geoinformatiky se stalo hodnocení objektů či jevů na základě jejich tvarových charakteristik efektivnějším.

Cílem bakalářské práce je práce s tvarovými metrikami. V první části si probereme příklady využití tvarových metrik. S tvarovými metrikami se můžeme setkat v různých odvětvích. Jedním z nejčastějších oblastí výskytů tvarových metrik je krajinná ekologie. Následně si metriky rozdělíme do tří skupin podle úrovně analýzy. V další části se zaměříme na představení některých již existujících nástrojů zabývajících se výpočty metrik. Popíšeme si, s jakými metrikami tyto nástroje pracují. Po teoretické části se zaměříme na část praktickou, která se skládá ze tří kapitol. V první kapitole se zaměříme na implementaci vlastní knihovny metrik. Popíšeme si třídu *Map*, která nám slouží k reprezentaci plochy nad níž budeme provádět výpočty metrik. Objasníme si pojem sousedství buněk, souvislosti mezi grafy a mapou, kterou využijeme k reprezentaci útvarů. Dále si ukážeme implementaci vybraných metrik a jejich objektové rozvrstvení. Ve druhé kapitole si popíšeme tvorbu vlastní tvarové metriky. Ta je založená na želví grafice a využívá návrhový vzor kompozit. Nakonec je potřeba knihovnu otestovat a zhodnotit výsledky provedených experimentů.

Kapitola 2

Využití metrik

Jak již bylo zmíněno v úvodu, metriky své uplatnění nacházejí zejména v krajinné ekologii, která iniciovala vznik některých metrik.[1] Krajinnou ekologií se podrobně zabývá mezi krajinnými ekology hojně využívaná publikace autorů Formana a Gordona[2]. Ti zavádí do zkoumání krajiny pojem *patch*. *Patch* (v českém vydání překládáno jako ploška) je základním pojmem, s nímž krajinná ekologie operuje. Označuje homogenní část povrchu země (krajiny), který se svým vzhledem nápadně liší od svého okolí (např. les, rybník). Jedná se o jednotlivé prvky, které společně tvoří mozaiku krajiny, tedy její výsledné vzezření. Dále uvádí, že zkoumání tvarů plošek přináší širší možnosti pojetí zkoumání krajiny, než jen porovnávání jejich rozlohy.

Přestože se stále spekuluje o možnostech a šíři využití tvarových metrik v této oblasti, jedná se o jeden ze způsobů hodnocení krajiny. Mnoho ukazatelů, jejichž výpočet byl odvozen z tvarů ploch, upozornilo na výskyt prostorových vzorů, které mohou mít vliv na ekologické procesy.

Skupina mexických vědců[3] zkoumala dopady změn krajiny na rozmanitost výskytu savců v oblasti roztržitého Lacandonského deštného pralesa v oblasti jihovýchodního Mexika. Ve své studii využili celkem osm tvarových metrik. Jejich práce prokázala, že s větší složitostí tvarů ploch a rozlohou se zvyšuje biodiverzita dané oblasti. Obávají se však neustálého rozšiřování pastvin a kácení lesů, což by časem mohlo zásadně narušit fungování místního ekosystému.

Tvarové metriky jsou využívány také k hodnocení šíření lesních požárů. Parisien a kol.[4] použil metriky k popisu postupu šíření, tvaru nebo například velikosti požárů vznikajících v lesích Kanady. Nabádá k detailnějšímu prozkoumání těchto vzorců s přihlédnutím k dalším okolnostem jako jsou rozmístění specifických krajinných útvarů (vodní plochy nebo již vypálené části země), role počasí a podnebí apod. Nicméně informace nabyté tímto výzkumem jsou podle Parisiena významné pro následné plánování využití krajiny.

Gökyer[5] pomocí metrik porovnával rozložení krajiny v severní části Turecka, provincii Bartın, v letech 2000 a 2010. Zpozoroval nárůst lesních ploch a úbytek ploch zemědělských. Také zde klesla průměrná velikost ploch, což značí větší fragmentaci území a narušuje život tamních živočichů. V závěru dodává, že metriky jsou důležitým nástrojem k získávání informací o stavu krajiny.

Čepová se ve své práci zabývá využitím tvarových metrik v hydrologii, konkrétně ke studiu tvarů povodí a říční sítě v oblasti povodí Moravy a Dyje. Vyzdvihuje metriky jako vhodný nástroj k hodnocení povodí, ale uvádí, že jejich interpretace je relativně složitá a často do jisté míry subjektivní. [6]

Tvarové metriky se mohou uplatnit také při hodnocení dat získaných z technologie dálkového průzkumu Země. Liu a Jiao zkoumali využití krajiny pomocí snímků z dálkového průzkumu v jižní oblasti Číny s využitím výpočtů tvarových metrik s vidinou tvorby přesnějšího a inteligentnějšího systému k autonomnímu rozpoznávání využití krajiny. [7]

Dalším odvětvím, kde nalezly tvarové metriky uplatnění je sociální geografie. Metriky jsou využívány k hodnocení sídelní kaše (anglicky *urban sprawl*), tedy prorůstání měst a obcí do okolní krajiny. Neuspořádané rozrůstání měst spolu s vysokou hustotou obyvatel a výraznou rozptýleností obydlí, které můžeme pozorovat především v Jižní Americe, střední a východní Asii nebo Africe značně zatěžuje okolní prostředí. Ať už energeticky, vodohospodářsky nebo větší závislosti na automobilové a hromadné dopravě. Torrens a Alberti se zabývali způsoby, jak popsat sídelní kaši mimo jiné i s použitím tvarových metrik. [8] Hartmannová uvádí možnost využití tvarových metrik k hodnocení (sub)urbanizace a její dopad na životní prostředí a životní úroveň obyvatel. Ve své práci využila 34 metrik k zjištění a vyhodnocení prostorových charakteristik na 47 vybraných městech. [9] Pászto ve své práci prezentuje tvarové metriky jako zdroj dodatečných kvantitativních informací o zkoumaných objektech či jevech a věří, že společně s dalšími metodami *geocomputation* probíranými v této [10] práci přináší přidanou hodnotu k běžným přístupům v geoinformatice a vhodně je doplňují.

Parent uvádí možnost využití tvarových metrik k identifikaci *gerrymanderingu*. [11] Jde o účelnou manipulaci s volebními obvody podle znalosti preferencí voličů vedoucí ke zvýhodnění určité politické strany. Jako další využití zmiňuje vyhodnocení vhodnosti části krajiny k různým záměrům při plánování výstavby.

Kapitola 3

Rozdělení tvarových metrik

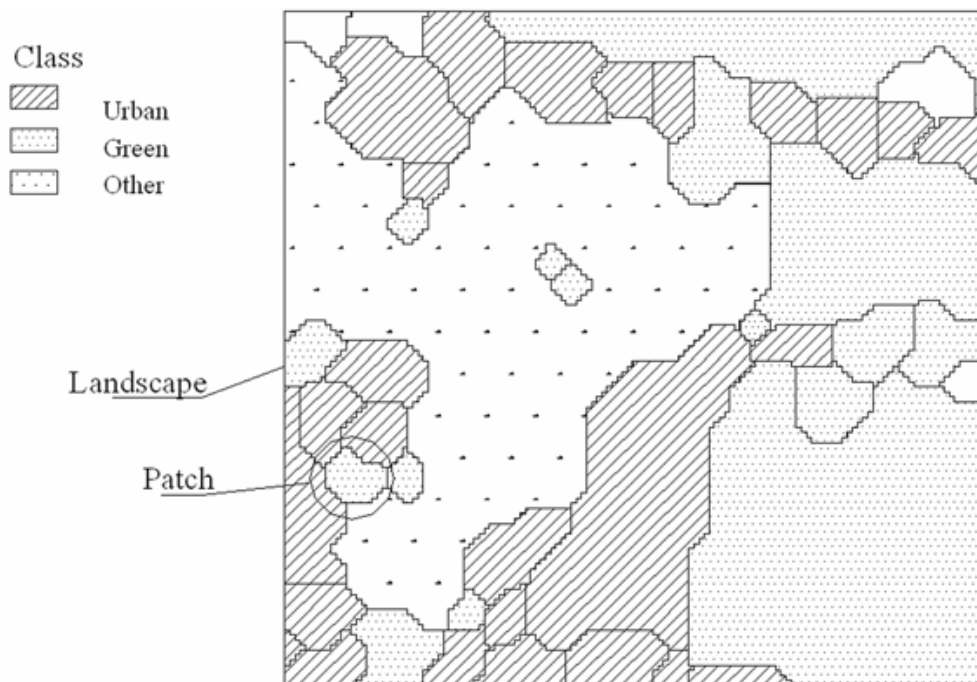
Základními tvarovými metrikami jsou obvod a rozloha. Tyto výchozí metriky jsou často používány ještě například spolu s body tvořícími obvod zkoumaného tvaru nebo body uvnitř útvaru k výpočtům metrik komplexnějších. Při analýze tvarů je důležité brát v potaz jejich měřítko. To se snaží některé metriky kompenzovat vztahem hodnot na čtverec, jedná-li se o vstupní data v podobě rastru, popřípadě na kruh při práci s vektorovými daty o stejné rozloze jakou disponuje zkoumaná plocha.[11] U tvarových metrik je klíčová především jejich korektní interpretace, která s sebou nese pochopení matematických vztahů stojících za výpočtem.[10] McGarigal také uvádí[12], že v některých případech mohou některé metriky navzájem korelovat a jevit se jako zcela nadbytečné, avšak často spolu bývají k dispozici v softwarových nástrojích, jelikož nabízí odlišný pohled, jak na shodnou informaci nahlížet. Ve výsledku pak záleží na oblasti použití metrik a požadavcích uživatele.

Metriky můžeme rozdělit podle souhrnu prostorových vlastností útvarů, které zkoumáme. McGarigal uvádí tyto kategorie: *Area a edge*, *Shape complexity*, *Core Area*, *Contrast*, *Aggregation*, *Subdivision a Isolation* Existující softwarové nástroje dělí metriky do často stejných nebo alespoň podobných skupin podle zaměření. V kapitole 4 o softwarových nástrojích jsou tato rozdělení více probrána.

Dalším způsobem je dělení podle úrovně, kterou metriky hodnotí. Zde se bavíme o třech kategoriích: *patch*, *class a landscape level* viz 3.1.

3.1 Patch level

Rastrová mapa, na kterou vykreslujeme hodnocené útvary je tvořená jednotlivými čtvercovými buňkami. Každá buňka má svou hodnotu, kterou se vyznačuje ve svém okolí (sousedství). Tato hodnota buňky určuje její příslušnost k objektu nebo jevu, který má na mapě reprezentovat (les, komunikaci, míru znečištění apod.). Pokud buňky se stejnou hodnotou spolu sousedí, je tento celistvý shluk označován jako ploška. Jestli vezmeme v úvahu vektorovou interpretaci mapy, můžeme ploškami označit jednotlivé uzavřené polygony. Tyto metriky řeší tvar, velikost, kompaktnost a další vlast-



Obrázek 3.1: Znázornění tří kategorií tvarových metrik na mapě [13]

nosti útvarů na mapě zvlášť pro každou plošku. Můžeme je využít ke vzájemnému porovnávání objektů či jevů stejného typu. Slouží také jako základní nástroje pro výpočet metrik dalších úrovní. Na tvorbu knihovny pro tyto metriky se zaměřuje tato bakalářská práce.

3.2 Class level

Jedná se o metriky, které se na jednotlivé plošky dívají souhrnně, ale dávají do kontextu pouze ty, které popisují stejný typ objektu nebo jevu na mapě (např. plošky znázorňující všechny vodní toky, plošky znázorňující oblasti se stejnou průměrnou roční teplotou), tedy plošky náležící stejné třídě. Často jsou určeny k výpočtům průměrné, maximální nebo minimální hodnoty metrik úrovně *Patch level* pro danou třídu objektů nebo jevů na mapě. Využití v krajinné ekologii získávají zejména metriky této úrovně například při analýze osídlení ploch.

3.3 Landscape level

Podobně jako *Class level* metriky se i tyto dívají na plošky souhrnně, avšak již nerozlišují, jaký typ objektu či jevu ploška reprezentuje. Hodnotí tak celkové uspořádání všech plošek na mapě, tedy celou mozaiku krajiny, kterou mapa prezentuje.

Kapitola 4

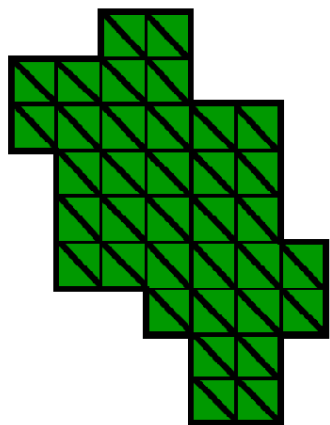
Software k výpočtu metrik

V této části bakalářské práce si představíme některé z existujících softwarových nástrojů, které byly vyvinuty k výpočtům tvarových metrik. K získávání hodnot metrik se používají nejen programy vytvořené speciálně k jejich výpočtům, ale také již zavedené komplexní geoinformační nástroje, které umožňují spouštění doprogramovatelných doplňků tzv. extenzí. Každý nástroj pracuje s metrikami různých názvů a definic, avšak mnohé z nich mají svůj ekvivalent nebo obdobu ve všech nástrojích. Také si zde uvedeme příklady konkrétních metrik a jejich popis pro ukázkou, s jakými metrikami se můžeme setkat.

4.1 FRAGSTATS

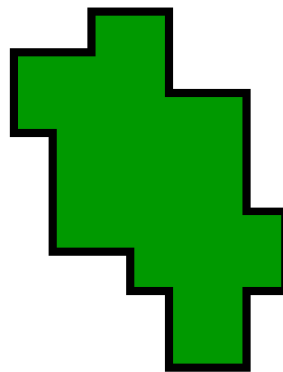
Jedním z nejčastěji používaných a nejstarších nástrojů k výpočtům krajinných metrik je program FRAGSTATS. Tento software byl vytvořen Dr. McGarigalem a Marksovou a poprvé zveřejněn ve verzi v2 v roce 1995. Obsahuje přes 70 tvarových metrik, jestliže vezmeme v potaz separátně i směrodatné odchylky a vážené průměry těchto metrik, a organizuje je do tří kategorií rozdělených podle úrovně analýzy (*patch, class a landscape level*). Vstupem jsou data v podobě rastrového obrázku, který může být reprezentován různými formáty (ASCII grid, ESRI grid, GeoTIFF grid a další). Jeho aktuální verze je v4.2, která přináší přepracované grafické rozhraní a podporu pro více rastrových formátů. FRAGSTATS je psaný v jazyce C++ pomocí nástroje Microsoft Visual C++ jako 32bitová aplikace určená pro systémy Windows. Více informací ohledně programu FRAGSTATS, včetně uvedení uživatele do problematiky analýzy krajiny, instrukcí ke zprovoznění a používání softwaru a popisu všech použitých metrik lze nalézt v obsáhlé uživatelské příručce vytvořené společně s programem.[12] Níže si popíšeme devět vybraných *patch level* metrik, které byly implementovány do knihovny:

Patch Area – Jedná se o celkovou rozlohu plošky v hektarech, viz obrázek 4.1a na stránce 16. Jak již bylo zmíněno, tato hodnota je často využívána pro výpočet v dalších metrikách.



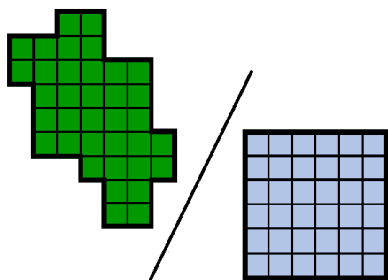
$$\text{AREA} = a_{ij} \left(\frac{1}{10,000} \right)$$

(a) Obsah plošky, a_{ij} značí rozlohu všech buněk udávanou v metrech. Rozloha plošky je následně přepočítána na hektary.



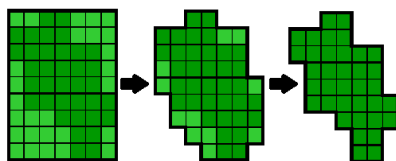
PERIM

(b) Obvod plošky počítaný v metrech, kdy hrana jedné buňky je rovna jednomu metru. Výsledkem je součet všech vnějších hran buněk.



$$\text{SHAPE} = \frac{0.25 p_{ij}}{\sqrt{a_{ij}}}$$

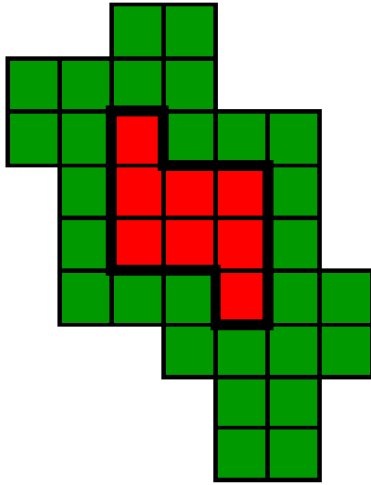
(c) Shape Index, p_{ij} označuje obvod plošky, a_{ij} značí její obsah.



$$\text{FRACT} = \frac{2 \ln (0.25 p_{ij})}{\ln a_{ij}}$$

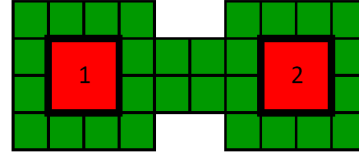
(d) Fractal dimension index, p_{ij} označuje obvod plošky, a_{ij} značí její obsah

Obrázek 4.1: FRAGSTATS metriky, první část



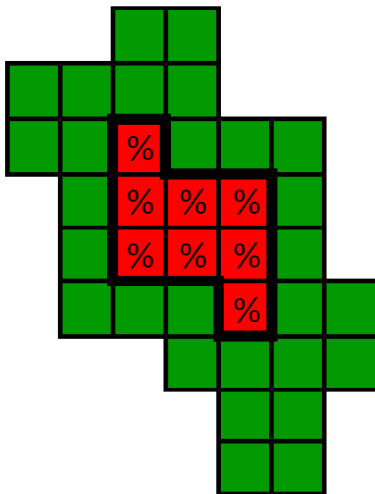
$$\text{CORE} = a_{ij}^c \left(\frac{1}{10,000} \right)$$

(a) Core area udává rozlohu jádrové oblasti v hektarech. a_{ij}^c označuje rozlohu všech jádrových buněk.



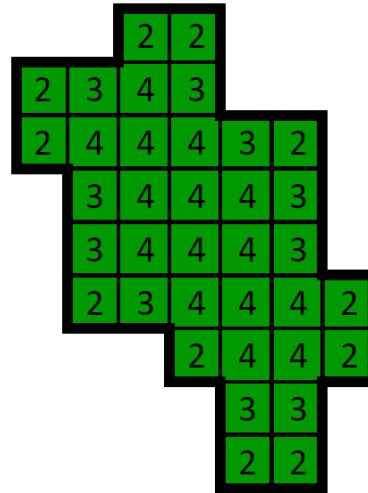
NCORE

(b) Number of core areas značí počet separovaných jádrových oblastí.



$$\text{CAI} = \frac{a_{ij}^c}{a_{ij}} (100)$$

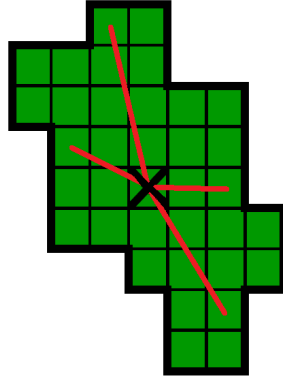
(c) Core Area Index, a_{ij}^c označuje rozlohu všech jádrových buněk a a_{ij} označuje rozlohu všech buněk plošky



$$\text{CONTIG} = \frac{\left[\frac{\sum_{r=1}^z c_{ijr}}{a_{ij}} \right] - 1}{v - 1}$$

(d) Contiguity Index, c_{ijr} je průměrná spojitost buňky s okolními sousedy, v je součet hodnot pomocné matice.

Obrázek 4.2: FRAGSTATS metriky, druhá část



$$GYRATE = \sum_{r=1}^z \frac{h_{ijr}}{z}$$

Obrázek 4.3: Radius of gyration, h_{ijr} označuje vzdálenost buňky od středu plošky, z je počet buněk plošky

Patch Perimeter – Značí obvod měřené plošky. Ve FRAGSTATS je udáván v metrech. Jde také o metriku užívanou v komplexnějších výpočtech.

Shape Index – Určuje složitost plošky vzhledem ke čtverci o stejné rozloze jakou má zkoumaná ploška. Znázorněno na obrázku 4.1c. Plošky, které se tvarem podobají čtverci, mají hodnotu 1. Čím více je ploška proměnlivá a svým tvarem se odlišuje od čtvercového tvaru, tím vyšší má hodnotu této metriky. Počítá se jako čtvrtina hodnoty obvodu plošky vydělená druhou odmocninou rozlohy plošky:

$$Shape = \frac{0.25p_{ij}}{\sqrt{a_{ij}}}$$

Fractal Dimension Index – Hodnoty tohoto indexu se pohybují mezi 1 a 2. Plochy s hodnotou 1 se vyznačují velice jednoduchou obvodovou hranicí, naopak tvary, u kterých se hodnota Fractal dimension indexu blíží 2 mají hranici obvodu komplexnější. Index vypočítáme jako dvojnásobek přirozeného logaritmu ze čtvrtiny velikosti obvodové hranice vydělený přirozeným logaritmem obsahu počítaného útvaru podle vzorce:

$$Fract = \frac{2 \ln(0.25p_{ij})}{\ln a_{ij}}$$

Pokud nastane situace, kdy se obvodová hranice rovná 4 a zároveň obsah útvaru je roven 1, index nabývá hodnoty 1.

Core Area – Odebráním pásu o určité šířce po obvodu útvaru vzniká útvar nový nazývaný jádrová oblast (*Core Area*). Tato metrika reprezentuje rozlohu nově vzniklé oblasti. Hodnota rozlohy

jádrové oblasti se snižuje s odebráním širšího pásu plochy po obvodu. Na obrázku 4.2a vidíme jádrovou oblast znázorněnou červenou barvou, zelenou jsou vykresleny hranice původní plošky, které byly pro vznik jádrové oblasti odebrány.

Number of core areas (*NCore*) – Při výpočtu metriky *Core area* může nastat situace, při které se původně celistvá plocha při vytváření jádrové oblasti rozdělí na více částí od sebe oddělených viz obrázek 4.2b. Hodnota indexu určuje počet takto oddělených jádrových oblastí. Počet vzniklých jádrových plošek je ovlivněn velikostí a tvarem původní plošky, ale také šířkou odebíraného pásu po obvodu.

Core Area Index – V procentech určuje poměr mezi obsahy jádrové oblasti a původní plošky viz obrázek 4.2c. Vypočtené indexy, které se blíží hodnotě 100 % znamenají, že vzniklé jádrové oblasti nabývají podobného obsahu jako původní ploška.

Contiguity Index (*Contig*) – Udává průměrnou souvislost plošky. Sleduje hranice každé buňky rastrové plošky a počítá kolika hranami buňka sousedí s ostatními buňkami plošky. Hodnoty všech buněk se zprůměrují a vzniká výsledek této metriky. Získáváme tak informaci o propojenosti, konektivitě, mezi buňkami zkoumané plochy. Postup mé implementace výpočtu je popsán v části 5.2.

Radius of gyration (*Gyrate*) – Představuje průměrnou vzdálenost mezi geometrickým středem plošky tzv. centroidem a všemi buňkami náležícími plošce. Pokud plošku tvoří pouze 1 buňka je hodnota metriky rovna 0. Maximální hodnoty dosáhne, pokud ploška okupuje celou zkoumanou plochu. Metrika charakterizuje rozlohu plošky a její kompaktnost. Implementace této metriky je popsána v části 5.2 této práce.

4.2 Shape Metrics Tool

Je rozšířením programu *ArcGIS for Desktop* od firmy *Esri*, které bylo vytvořeno týmem Jasona Parenta.[11] Jde o skript napsaný v programovacím jazyce Python. Do výpočtů vstupují vektorová data. Pro každou z metrik lze získat její absolutní nebo normalizovanou hodnotu. Normalizovaná hodnota vzniká vztažením hodnoty absolutní k hodnotě metriky kruhu o stejném obsahu (tzv. *Equal Area Circle*). Hodnoty se tak pohybují od 0 do 1. Parent u většiny metrik udává příklad využití dané metriky. Toto rozšíření pracuje s výpočty těchto dvanácti metrik:

Cohesion index – Index udává informaci o průměrné vzdálenosti všech dvojic bodů uvnitř plošky. Všem bodům přikládá stejnou váhu, nezáleží tedy jestli se jedná o body hraniční nebo vnitřní. Jelikož by byl výpočet u plošek s velkým množstvím bodů náročný, vybírá se jen několik náhodných bodů, mezi kterými se vzdálenosti měří. V reálném světě můžeme tuto metriku připodobnit k pohybu osob v rámci města (naší plošky). Jestliže se normalizovaná metrika

blíží více 1, předpokládá se že město je rozloženo kompaktněji, tedy jeho tvar se podobá kruhu. Naopak pokud se hodnota blíží 0, očekává se, že město je více protáhlé.

Proximity index – Popisuje průměrnou vzdálenost všech bodů plošky od jejího geometrického středu. V praxi si lze představit dojíždění osob do centra města. U normalizovaného indexu se při hodnotě 1 opět tvar zkoumané plošky podobá kruhu.

Spin index – Jde o variaci proximity indexu. Liší se použitím druhých mocnin vzdáleností. Více se tak projevují extrémy. Tato metrika se hodí ke zkoumání kompaktnosti plošky se zaměřením na tvarové výběžky a k hodnocení tzv. sídelní kaše.

Exchange index – Jedná se o rozlohu plochy, která spadá do Equal Area Circle umístěného do středu plošky.

Perimeter index – Značí obvod plošky. Porovnáváním normalizovaných indexů získáme informaci o míře vystavení plošky vnějším vlivům. Toho se využívá například při zkoumání plošek lesních porostů. Plošky s vyšší hodnotou normalizovaného indexu jsou v menší míře vystaveny vnějšímu prostředí.

Detour index – Udává délku obvodu nejmenšího konvexního mnohoúhelníku, který opisuje zkoumanou plošku. Jeho hodnotu můžeme využít k určení míry, kterou omezují dálnice pohyb divoké zvěře a měřit efektivitu výstavby ekoduktů.

Girth index – Vyjadřuje poloměr největší kružnice vepsané do tvaru plošky. Slouží ke stanovení ploch, které se vyznačují souvislou rozlohou dostatečně velkou k potencionálnímu rozvoji (např. výstavbě díla).

Viable Interiour index – Ekvivalent *Core area* z FRAGSTATS. Po odebrání souvislého pásu z hraniční oblasti získáme plošku menší. Dá se využít k určení, do jaké vzdálenosti od hranice původní plošky je ploška ovlivňována vnějšími vlivy.

Depth index – Označuje průměrnou vzdálenost všech vnitřních bodů plošky k nejbližším bodům ležícím na hranici zkoumaného tvaru. S narůstající průměrnou vzdáleností se zvyšuje izolovanost plošky od vnějších vlivů. Tuto metriku lze uplatnit při analýze vnějších vlivů (pole, městská zástavba) a jejich dopad na výskyt druhů rostlin a živočichů.

Traversal index – Jde o průměr nejkratších vzdáleností dvou hraničních bodů. Cesta mezi vybranými body musí náležet zkoumané plošce.

Range index – Index vyjadřuje průměr nejmenší kružnice opisující plošku. Jedná se o délku mezi dvěma nejvzdálenějšími body plošky.

Dispersion index – Označuje průměrnou vzdálenost mezi středem plošky a hraničními body. Určuje, zda se ploška rozšiřuje rovnoměrně do všech směrů. Případně místa uvnitř plošky, která plošce nenáleží, tato metrika nebere v úvahu.

4.3 landscapemetrics

Jedná se o open-source rozšiřující balíček programovacího jazyka R. Jako vstup využívá data v podobě rastru. Byl vytvořen týmem okolo Maximiliana H. K. Hesselbartha[14] a implementuje metriky definované McGarigalem v programu FRAGSTATS ve všech třech úrovních. Navíc obsahuje metriky zabývající se celkovou složitostí krajiny (např. *joint entrophy* nebo *conditional entrophy*) a plánuje se přidání dalších metrik v následných aktualizacích. Klasifikuje metriky celkem do šesti různých kategorií:

Area and edge metrics (metriky plochy a ohrazení) – Popisují velikost plošek, tříd a ohrazení. Ohrazení je zde definováno jako pás mezi dvěma ploškami stejné třídy v metrech. Tyto metriky jsou schopny určit převahu nebo naopak řídkost plošek na mapě a charakterizují tvarové uspořádání krajiny.

Shape metrics (tvarové metriky) – Využívají převážně hodnoty obvodu a rozlohy plošek k jejich analýze. Zaměřují se na posuzování tvaru plošky, který může být různý pro plošky se stejnou rozlohou.

Core metrics (metriky jádrové oblasti) – Analyzují plochu jádrové oblasti plošek. Tyto metriky mohou být vhodné při posuzování vzájemného ovlivňování sousedících plošek. Zahrnuje například metriky *Core Area*, *Core Area Index*, počet oddělených jádrových oblastí.

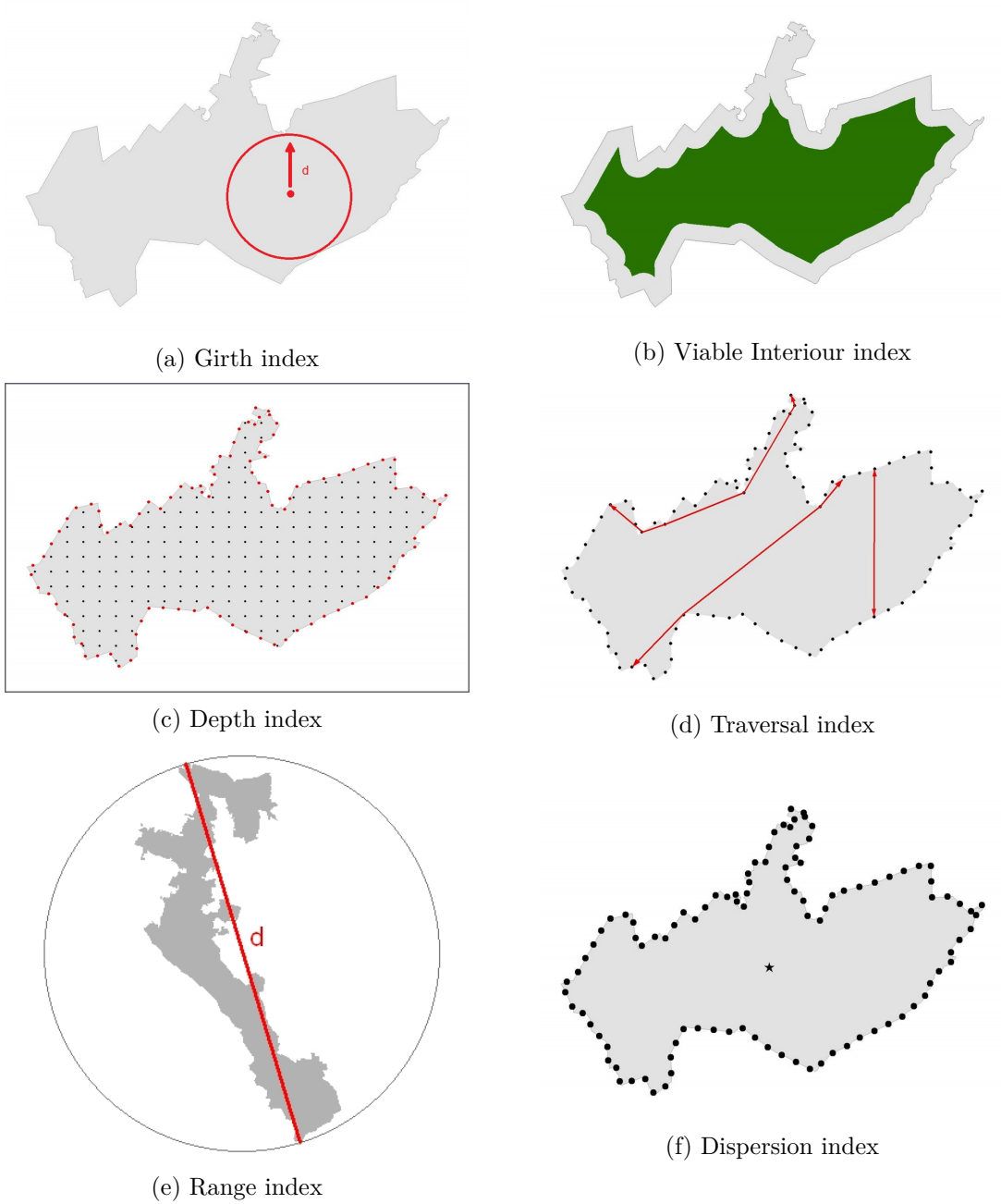
Aggregation metrics (agregační metriky) – Plošky se mohou nacházet buďto nahuštěné na určité části mapy, anebo se mohou izolovat od ostatních. Tyto vlastnosti v uspořádání krajiny zkoumají agregační metriky.

Diversity metrics (metriky rozmanitosti) – Jedná se o metriky, které se zaměřují pouze na *landscape level* analýzu a zkoumají rozložení, hojnost a rozmanitost plošek stejné třídy.

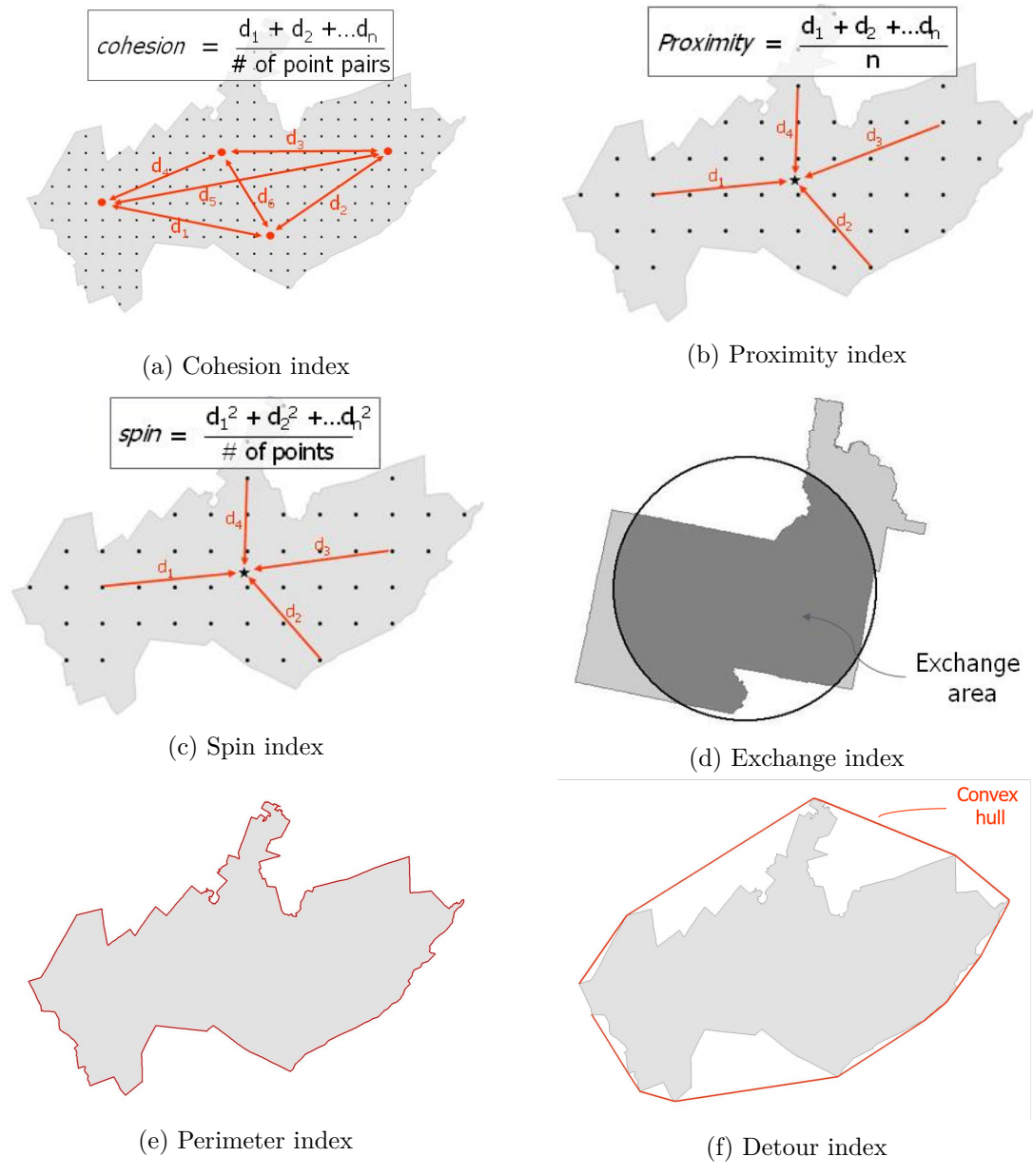
Complexity metrics (metriky složitosti) – Zabývají se celkovou složitostí krajiny podobně jako metriky rozmanitosti. Zkoumají vztahy oddělených plošek stejné třídy, především jejich polohu v rámci krajiny.

4.4 Vector-based Landscape Analysis Tools (V-LATE)

V-LATE je rozšiřujícím balíčkem programu *ArcGIS* podobně jako již zmiňovaný *Shape Metrics Tool*. Pracuje také pouze s vektorově orientovanými prostorovými daty, která jsou ukládána ve formě



Obrázek 4.4: Shape Metrics Tool metriky, první část [11]



Obrázek 4.5: Shape Metrics Tool metriky, druhá část [11]

polygonů do formátu Shapefile. Nástroj byl vytvořen na Salcburské univerzitě v rámci projektu SPIN. Sestává z metrik často používaných k analýze krajinné mozaiky. Extenze kategorizuje tyto metriky do sedmi typů analýz:

Area Analysis (analýza plochy) – Slouží k výpočtům rozlohy plošek stejné třídy (*Class Area*), popřípadě celé mapy (*Total Area*), průměrné velikosti plošek (*Mean Patch Size*), směrodatných odchylek velikostí (*Patch Size Standart-Deviation*) v m^2 a celkovému počtu plošek (*Number of Patches*).

Form Analysis (analýza tvaru) – Počítá *Shape Index*, *Fractal Dimension Index*, poměr mezi obvodem a obsahem plošky a jejich průměrné hodnoty ze všech tříd plošek.

Edge Analysis (analýza ohraničení) – Vypočítá průměrnou délku ohraničení plošek stejné třídy (*Mean Patch Edge*) nebo celé mapy (*Total Edge*) v metrech a hustotu ohraničení (*Edge Density*) v m/ha.

Core Area Analysis (analýza jádrových ploch) – Vytvoří novou vrstvu, která reprezentuje vnitřní část plošek nebo-li jádrové oblasti s volbou velikosti výběru. Výstupními metrikami analýzy jsou: *Core Area Index* v procentech, počet oddělených jádrových oblastí celková rozloha, celková rozloha jádrových oblastí (*Total Core Area*) a rozloha jádrových oblastí plošek stejné třídy (*Total Class Core Area*) v m^2 .

Proximity Analysis (analýza vzájemné vzdálenosti ploch) – Výpočtem získáme nejbližší sousední plošku (*Nearest Neighbour*). Její vzdálenost od počítané plošky (*Distance to Nearest Neighbour*), hodnotu (*Nearest Neighbour ID*) a rozlohu (*Nearest Neighbour Area*).

Diversity Analysis (analýza rozmanitosti typů ploch) – Obsahuje metriky k určení plnosti/bohatosti plošky včetně relativních a potencionálních hodnot, proporcí a Shannonův index diverzity a rovnováhy používané v biologii ke studiu druhové rozmanitosti.

Subdivision Analysis (analýza rozdělení ploch) – Operuje s metrikami, které se zabývají nahuštěním plošek nebo naopak jejich rozptýlením v rámci mapy. Mezi počítané metriky patří Index dělení krajiny (*Landscape Division Index*), index rozdělení (*Splitting Index*) nebo velikost skutečné sítě (*Effective Mesh Size*).

Před provedením výpočtu některé z těchto analýz je nejprve nutné spočítat plochu a obvod jednotlivých tvarů, až poté program zpřístupní požadovanou funkcionalitu. Většinu výpočtů provádí na úrovni *class level* případně *landscape level*.

4.5 Další nástroje

Softwarových nástrojů sloužících k výpočtům tvarových charakteristik krajiny existuje mnohem více. Můžeme zmínit ještě například *Patch Analyst*, *LecoS* nebo *StraKa*.

Patch Analyst je opět doplňkem programu *ArcGIS*, která využívá metrik definovaných v programu *FRAGSTATS*[12]. Pracuje s vektorovými i rastrovými daty.

Pro open-source geografický informační systém QGIS vytvořil Jung[15] rozšíření psané v jazyce Python pod názvem *LecoS*, ve kterém implementoval přes šestnáct tvarových metrik.

Toolbox *StraKa*[16] je nástroj využívající vzorce k výpočtu metrik definované Formanem a Gordonem[2]. Byl publikován v roce 2008 jako výstup práce Kateřiny Pavkové. Je rozdělen na dva toolsety: Indexy krajiny a Popis krajiny. Indexy zahrnují šest metrik určených k hodnocení krajiny: *Index tvaru plošky*, *index izolovanosti plošky*, *index přístupnosti plošek*, *interakce mezi ploškami*, *izolovanosti plošek (rozptyl)* a *index rozptýlenosti plošek*.

Kapitola 5

Implementace vlastní knihovny

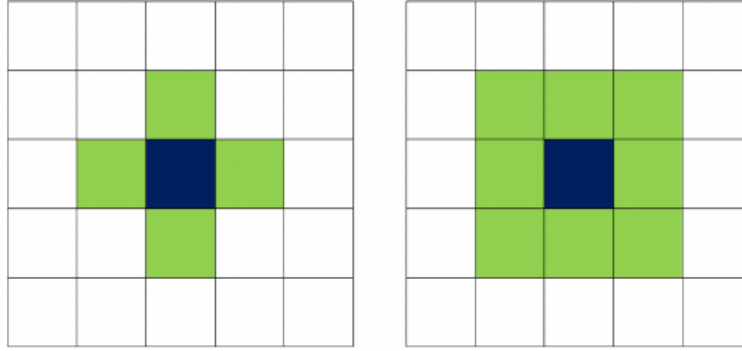
Prvním bodem praktické části této bakalářské práce je implementace knihovny poskytující výpočty vybraných tvarových metrik. Pro implementaci byla vybrána platforma .NET 5[17] (do verze 3.1 označovaná jako .NET Core) a programovací jazyk C#. .NET 5 je open-source platforma vyvíjená společností Microsoft. Jde o nástupce technologie .NET Framework. Důvodem jejího vzniku bylo vytvoření prostředí k tvorbě výkonných aplikací bez závislosti na specifickém operačním systému.

Knihovna je opatřena dokumentací, včetně popisu jednotlivých metrik, vytvořenou pomocí nástroje *Doxygen*[18]. Implementaci knihovny můžeme rozdělit na dvě části. Tou první je vytvoření třídy k uchování mapy a rozlišování jednotlivých plošek, které obsahuje. V druhé se budeme zabývat implementací samotné knihovny několika vybraných metrik.

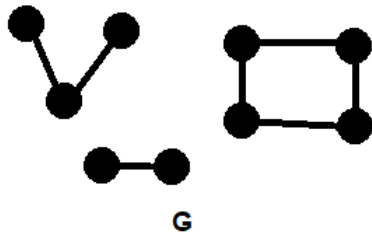
5.1 Třída Map

Třída *Map* je základní třídou celé knihovny. Obsahuje dvourozměrné pole typu *Integer*, které reprezentuje zkoumanou plochu. Pomocí dostupných konstruktorů lze mapu vytvořit z objektu typu *Bitmap* nebo například ze souboru. Třída operuje s tím, že veškeré hodnoty s číslem 0 jsou považovány za prázdnou část mapy. Naopak hodnoty v poli s číslem 1 jsou určeny k identifikaci a rozlišení na jednotlivé oddělené plošky.

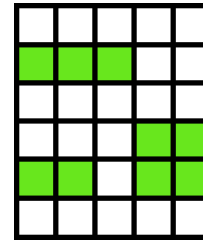
S rozlišováním plošek souvisí parametr konstruktoru s názvem *directions*, který akceptuje dvě číselné hodnoty: 4 a 8. Tímto mapě podáme informaci o tom, s kolika jinými buňkami sousedí každá zkoumaná buňka plošky. V případě hodnoty 4 se jedná o tzv. neumannovské okolí (sousedství). Anglicky taktéž označováno jako Rook's case podle možných tahů věže v šachu. V tomto případě může zkoumaná buňka sousedit až se čtyřmi dalšími buňkami, se kterými sdílí hranu, jak je znázorněno na obrázku 5.1 vlevo. Pokud se podíváme na hodnotu 8, hovoříme o moorovském okolí (sousedství) pojmenovaném po Edwardu F. Mooreovi, který se stejně jako John von Neumann zmíněný výše zabýval buněčnými automaty, kde se s pojmem okolí buňky pracuje. V anglickém jazyce se analogicky můžeme setkat s výrazy *Queen's* případně *King's case* opět vycházející z volby možných



Obrázek 5.1: Vlevo neumannovo sousedství, vpravo moorovské sousedství. Modrou barvou je označena buňka, jejíž sousedství aktuálně zkoumáme. Zelenou barvou jsou vyznačeni její sousedé. [19]



(a) Graf G skládající se ze tří komponent souvislosti.



(b) Mapa s vykreslenými ploškami, která je reprezentována grafem G z obrázku 5.2a.

Obrázek 5.2: Komponenty grafu a mapa

tahů tentokrát dámy nebo krále v šachu. V tomto případě zkoumaná buňka sousedí s osmi jinými buňkami, a to jak s těmi, se kterými sdílí hranu, tak i s buňkami se kterými sdílí pouze vrchol, viz obrázek 5.1 vpravo. V dalších vyobrazeních map budu primárně využívat neumannovské okolí, pokud nebude uvedeno jinak.

Samotné rozlišování jednotlivých plošek je zajištěno grafovým algoritmem prohledávání do šířky (*BFS – Breadth-first search*)[20]. Graf je základním předmětem oboru diskretní matematiky nazývané teorie grafů. Jedná se o matematickou strukturu tvořenou dvěma množinami: neprázdnou množinou vrcholů (popřípadě uzlů) a množinou hran, která se skládá z dvojic prvků množiny vrcholů. Tato struktura slouží k reprezentaci objektů a jejich vzájemného propojení. V našem případě můžeme za vrcholy označit jednotlivé buňky zkoumané mapy a jako hrany všechny neuspořádané dvojice buněk, které spolu sousedí v závislosti na již zmiňovaném parametru *directions*. Pokud se zaměříme pouze na vybranou část grafu, mluvíme o podgrafu. Dalším pojmem, který souvisí s naší reprezentací mapy je souvislost grafu. Graf nazýváme souvislým, pokud pro každé dva jeho vrcholy existuje sled, tedy posloupnost vrcholů, kdy mezi každými dvěma vrcholy existuje hrana.

Zkoumanou mapu si představíme jako graf, ve kterém jsou jednotlivé plošky znázorněny jako jeho maximální souvislé podgrafy, ty nazýváme komponenty souvislosti. Na obrázku 5.2a je zobrazen graf

7	6	5	4	5	6	7
6	5	4	3	4	5	6
5	4	3	2	3	4	5
4	3	2	1	2	3	4
5	4	3	2	3	4	5
6	5	4	3	4	5	6
7	6	5	4	5	6	7

Obrázek 5.3: Průchod mapou do šířky po vlnách. Postupně jsou procházeny všechny buňky stejného odstínu, poté přejdeme na buňky o odstín světlejší. Počítáme s tím, že všechny buňky na obrázku náležejí prohledávané plošce.

se třemi komponentami souvislosti, který reprezentuje mapu znázorněnou na obrázku 5.2b. Pokud mapa obsahuje pouze jednu komponentu, můžeme celou mapu znázornit jako souvislý graf.

BFS algoritmus prochází postupně vrcholy grafu, zde reprezentované jako buňky mapy, od vrcholu počátečního. Jeho způsob průchodu bývá označován jako záplavové prohledávání. Toto označení vychází z představy, kdy do počátečního vrcholu přitéká voda, která se následně šíří hranami do vrcholů sousedních. Děje se tak rovnoměrně v každém směru tzv. po vlnách jak můžeme vidět na mapě na obrázku 5.3 s počátečním vrcholem označeným číslem 1.

Pro prohledávání grafu do šířky se používá lineární datová struktura fronta. Fronta uplatňuje mechanismus přístupu FIFO (First in, First out), tedy prvním odebraným prvkem z fronty je ten, který jsme do ní jako první vložili.

```
Patch BFS(int[,] map, int row, int col, bool[,] visited, int value)
{
    //pozice sousedících buněk
    int[] neighborRow = new int[] { -1, 0, 0, 1, -1, -1, 1, 1};
    int[] neighborCol = new int[] { 0, -1, 1, 0, -1, 1, -1, 1};

    Queue<Tuple<int, int>> queue = new Queue<Tuple<int, int>>();
    //přidání počátečního vrcholu a označení nové plošky
    queue.Enqueue(new Tuple<int, int>(row, col));
    visited[row, col] = true;
    map[row, col] = value;
}
```

```

Patch patch = new Patch(col, row, col, row, value);

while (queue.Count != 0)
{
    int i = queue.Peek().Item1;
    int j = queue.Dequeue().Item2;

    //průchod všech sousedů aktuálně zkoumané buňky
    for (int k = 0; k < directions; k++)
    {
        //kontrola hranic a hodnoty buňky
        if ((i + neighborRow[k] >= 0) && (i + neighborRow[k] < maxRow) && (j +
            neighborCol[k] >= 0) && (j + neighborCol[k] < maxCol)
            && (map[i + neighborRow[k], j + neighborCol[k]] != 0 && !visited[i
                + neighborRow[k], j + neighborCol[k]]))
        {
            map[i + neighborRow[k], j + neighborCol[k]] = value;
            visited[i + neighborRow[k], j + neighborCol[k]] = true;
            queue.Enqueue(new Tuple<int, int>(i + neighborRow[k], j +
                neighborCol[k]));

            //úprava krajních hranic plošky
            if (patch.maxCol < j + neighborCol[k]) patch.maxCol = j +
                neighborCol[k];
            if (patch.maxRow < i + neighborRow[k]) patch.maxRow = i +
                neighborRow[k];
            if (patch.minCol > j + neighborCol[k]) patch.minCol = j +
                neighborCol[k];
            if (patch.minRow > i + neighborRow[k]) patch.minRow = i +
                neighborRow[k];
        }
    }
}
return patch;
}

```

Kód 5.1: Algoritmus prohledávání plošky

Prohledávání zahájíme procházením mapového pole, při kterém hledáme první buňku náležící některé z plošek. Při průchodu zaznamenáváme navštívené souřadnice do pomocného pole typu *boolean*. Po nalezení umístíme souřadnice buňky do fronty a zároveň v pomocném poli označíme polohu jako již navštívenou viz kód 5.1. Vytvoříme nový objekt *Patch* nesoucí informaci o hraničních souřadnicích plošky a identifikační hodnotě, která se s každou další ploškou zvyšuje. Odebereme souřadnice z fronty, označíme zkoumanou buňku identifikační hodnotou nově vzniklé plošky a projdeme všechny sousedící buňky v okolí. Pokud jsou okolní buňky ještě nenavštívené a zároveň se nejedná o prázdnou část mapy, přidáme souřadnice všech takových buněk do fronty. Celý proces se opakuje dokud není fronta vyprázdněná. Jakmile máme identifikovanou první plošku, pokračujeme stejným způsobem v hledání další dosud nenavštívené buňky označující útvar mapy. Prohledávání končí průchodem celého mapového pole.

5.2 Knihovna metrik

Bylo implementováno celkem devět již existujících metrik definovaných v dokumentaci programu *FRAGSTATS* popsané v kapitole 4.1. Všechny metriky vycházejí ze společné abstraktní třídy *Metric*, která obsahuje abstraktní metodu *calculate*. Tato metoda žádá jako parametr mapu, na které chceme provádět výpočet metriky a vrací datovou strukturu *List* typu *double* s výsledky metriky zvlášť pro každou plošku. V příloze A na obrázku A.1 vidíme třídní diagram knihovny zachycující i nově vytvořenou vlastní metriku založenou na želví grafice.

Tři metriky, konkrétně *CoreArea*, *CoreAreaIndex* a *NCore* mají jako svého předka abstraktní třídu *CoreMetric*, která dědí z třídy *Metric*. Tyto tři metriky ke svým výpočtům používají jádrovou oblast plošek a všechny potřebují k výpočtu informaci o velikosti těchto jádrových oblastí, kterou můžeme ovlivnit atributem *width*.

Pro ukázkou si zde probereme dvě z metrik podrobněji a ukážeme si jejich implementaci v kódu. K tomuto účelu byly vybrány metriky *Gyrate* a *Contig*. Metriku *Gyrate* vypočítáme jako průměrnou vzdálenost každé buňky od středu plošky. Nejprve si tedy musíme určit střed plošky. Ten dostaneme průchodem plošky a výpočtem průměrné hodnoty souřadnic buněk náležících plošce. Posléze projdeme plošku znovu a vypočítáme vzdálenost každé buňky od dříve vypočteného středu pomocí vztahu pro výpočet eukleidovské vzdálenosti v rovině:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2},$$

kde p a q jsou body v rovině a dolními indexy jsou označeny jejich jednotlivé souřadnice. Tuto vzdálenost vydělíme počtem buněk plošky. Postup zopakujeme pro každou buňku a součtem získáme výslednou průměrnou vzdálenost. Pokud plošku tvoří pouze jedna buňka, je hodnota metriky rovna 0.

```

List<double> gyrate = new List<double>();

foreach(Patch p in m.patches)
{
    int xCenter = 0;
    int yCenter = 0;
    int cells = 0;

    //hledání středu plošky
    for (int i = p.minRow; i <= p.maxRow; i++)
    {
        for (int j = p.minCol; j <= p.maxCol; j++)
        {
            if (p.value == m.map[i, j])
            {
                xCenter += i;
                yCenter += j;
                cells++;
            }
        }
    }
    xCenter /= cells;
    yCenter /= cells;

    //výpočet vzdálenosti
    double distance = 0;
    for (int i = p.minRow; i <= p.maxRow; i++)
    {
        for (int j = p.minCol; j <= p.maxCol; j++)
        {
            if (p.value == m.map[i, j])
            {
                distance += Math.Sqrt((xCenter - i) * (xCenter - i) + (yCenter - j)
                    * (yCenter - j)) / cells;
            }
        }
    }
}

```

```

//Pokud plošku tvoří pouze jedna buňka, je hodnota metriky rovna 0.
if (cells == 1)
    gyrate.Add(0);
else
    gyrate.Add(distance);
}
return gyrate;

```

Kód 5.2: Výpočet metriky *Gyrate*

Výsledkem metriky *Contig* je hodnota vyjadřující průměrnou souvislost plošky. K výpočtu je využita pomocná matice o velikosti 3×3 . Při průchodu plošky se u každé buňky díváme na její okolí. Pokud má buňka v okolí souseda, přičteme jí za každého hodnotu, která se nachází na jeho pozici v pomocné matici, pokud vezmeme v úvahu, že aktuálně kontrolovaná buňka se nachází uprostřed na souřadnicích [1, 1]. Jestliže zkoumaná mapa využívá moorovského okolí, jsou upřednostňováni sousedé buňky, kteří s aktuálně kontrolovanou buňkou sdílí celou hranu. Na těchto pozicích je v pomocné matici uvedena dvojnásobná hodnota, oproti pozicím, se kterými buňka sdílí pouze vrchol. Součet těchto hodnot pro všechny buňky vydělíme rozlohou celé plošky, od které odečteme 1. Výsledek poté vydělíme součtem hodnot v pomocné matici zmenšený o 1 a získáme finální hodnotu viz obrázek 4.2d. Opět je uvedeno, že pokud plošku tvoří pouze jedna buňka, je hodnota metriky rovna 0.

```

//pomocná matice
int[,] filterMatrix = new int[,] { { 1, 2, 1 },
                                     { 2, 1, 2 },
                                     { 1, 2, 1 } };

//souřadnice sousedních buněk
int[] neighborRow = new int[] {0, -1, 0, 0, 1, -1, -1, 1, 1};
int[] neighborCol = new int[] {0, 0, -1, 1, 0, -1, 1, -1, 1};

List<double> contig = new List<double>();
foreach (Patch p in m.patches)
{
    double patchArea = 0;
    double allPixelContig = 0;
    for (int i = p.minRow; i <= p.maxRow; i++)
    {
        for (int j = p.minCol; j <= p.maxCol; j++)
        {

```



```

if (m.map[i, j] == p.value)
{
    patchArea += 1;
    for (int k = 0; k <= m.directions; k++)
    {
        //kontrola hraničních buněk
        if ((i + neighborRow[k] >= p.minRow) && (i + neighborRow[k] <= p
            .maxRow) && (j + neighborCol[k] >= p.minCol) && (j +
            neighborCol[k] <= p.maxCol)
            && (m.map[i + neighborRow[k], j + neighborCol[k]] == p.value
                ))
        {
            //přidání hodnoty z 3 × 3 matice
            allPixelContig += filterMatrix[1 + neighborRow[k], 1 +
                neighborCol[k]];
        }
    }
}
}
}
//Pokud plošku tvoří pouze jedna buňka, je hodnota metriky rovna 0.
if (patchArea == 1)
    contig.Add(0);
else
    contig.Add((((allPixelContig / patchArea) - 1) / (13 - 1)));
}
return contig;

```

Kód 5.3: Výpočet metriky *Contig*

Kapitola 6

Návrh vlastní metriky

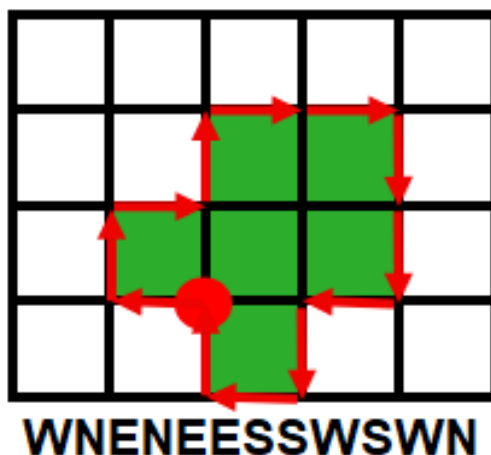
Tato část bakalářské práce se věnuje návrhu vlastní metriky založené na želví grafice. Želví grafikou se v počítačové grafice rozumí způsob vykreslování obrázků na určitou vymezenou oblast (označovanou plátno) pomocí virtuálního kurzoru, tzv. želvy, který má pouze možnost otáčet se do stran a pohybovat se dopředu (případně měnit štětec), zatímco za sebou zanechává viditelnou stopu podobně jako reálná želva plazící se v písku za sebou zanechává rozrytou trasu. Z této představy získala své pojmenování.

Ve stejném duchu byla želví grafika využita při vytváření nové metriky. Cílem je projít plošku mapy s pomocí pohybů želvy a vytvořit tak řetězec reprezentující směry, kterými se želva dala po obvodu plošky, jak můžeme vidět na obrázku 6.1. K tomuto účelu byl využit algoritmus *Square tracing*. [21] Myšlenka tohoto algoritmu je vcelku prostá. Prvním krokem je určení startovní pozice želvy a směru její orientace. Poté se řídíme dvěma základními pravidly, jak s želvou pohybovat:

1. Pokud želva aktuálně stojí na buňce plošky, posune se o jedno pole doleva.
2. Jestliže želva stojí na prázdné buňce, posune se o jedno pole doprava.

Tato dvě pravidla dodržuje do té doby, než se vrátí zpět do své startovní pozice. Nevýhodou tohoto algoritmu je ve výběru podmínky ukončení průchodu, jelikož v některých případech se želva může vrátit na startovní pozici, přestože neabsolvovala celou trasu kolem plošky. Existují dva efektivní způsoby, jak upravit původní ukončovací podmínku:

1. Ukončit prohledávání po opakovaném navštívení startovní pozice. V tomto případě musíme startovní pozici znovu navštívit alespoň dvakrát.
2. Ukončit prohledávání po navrácení želvy do původní pozice, avšak pouze v případě, kdy želva vstoupí na startovní pozici ve stejném směru jako při startu. Tento způsob ukončení se označuje anglicky *Jacob's stopping criterion* podle Jacoba Eliossoffa. [21]



Obrázek 6.1: Průchod želvy ploškou s výsledným řetězcem.

V implementaci bylo využito právě druhé zmíněné podmínky. Při průchodu si metrika ukládá seznam navštívených buněk do kolekce. Každá tato buňka může být tvořena až čtyřmi směrovými vektory. Tyto jednodušší objekty, tedy jednotlivé vektory obsažené v buňkách se ukládají do struktury složené, vycházející z návrhového vzoru kompozit. V příloze B na obrázku B.1 vidíme implementaci návrhového vzoru kompozit do želví metriky.

Výsledkem je posloupnost vektorů na sebe navazujících a trasujících obvod plošky. Každý vektor má svou textovou reprezentaci vycházející z jeho prostorové orientace. Jedná se o první písmeno anglického názvu světové strany, do které je vektor orientován. Při určování světové strany bereme v potaz, že horní hranice zkoumané mapy je orientovaná na sever. V případě vektoru orientovaného na sever, tedy zdola nahoru, odpovídá jeho textová reprezentace písmenu N (*North*). Analogicky postupujeme v ostatních třech směrech. Poskládáním těchto písmen za sebe nám vzniká textový řetězec představující obvod plošky (obrázek 6.1). Tento řetězec následně zkomprimujeme pomocí třídy *GZipStream*. Důvodem ke kompresi je odhalení opakujících se vzorů v řetězci. Když je výskyt takových podřetězců větší a více se tak projeví komprese, dá se předpokládat, že složitost tvaru plošky jež představuje řetězec je menší. Konečným výsledkem metriky je kompresní poměr. Jde o podíl velikosti komprimovaných dat k velikosti dat nekomprimovaných. Plošky s kratší obvodovou hranicí, u kterých se komprese neprojeví mají hodnotu metriky rovnou 0.

Želví metrika je uzpůsobená na mapy s neumannovským okolím uspořádání buněk. Plošky na těchto mapách se nesmí vyskytovat na hranicích mapy, poněvadž algoritmus procházení využívá buněk okolo zkoumané plošky.

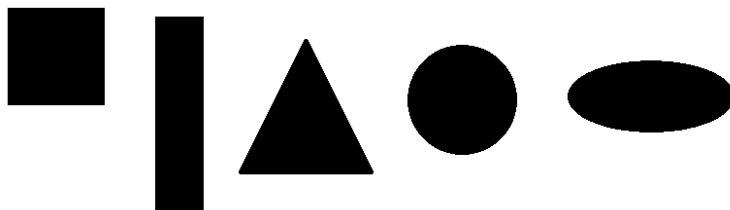
Kapitola 7

Testování knihovny

V této části si otestujeme knihovnu metrik na vybraných obrazcích. Vstupem jsou soubory v bezztrátovém formátu *PNG* sloužícímu k ukládání grafiky. Ve všech experimentech byly zkoumané plošky správně rozpoznány. První test se zabýval výpočtem a hodnocením metrik na základních geometrických útvarech přibližně stejné rozlohy. V dalším experimentu bylo zkoumáno, jak se metriky chovají na obrazcích reprezentujících postupné iterace při generování dvou typů fraktálů. V těchto dvou experimentech jsou také porovnány hodnoty želví metriky, především s metrikami *Fract* a *Shape*, které vyhodnocují složitost hranic plošky. Poslední experiment se zaměřoval na hodnocení funkčnosti metrik jádrových oblastí.

7.1 Rovinné obrazce

V tomto testu jsme vyzkoušeli vypočítat metriky základních rovinných útvarů. Srovnávali jsme obrazce s podobnou rozlohou přibližně 10 000 buněk. Tyto útvary můžete vidět na obrázku K výpočtu byly vybrány tyto útvary: čtverec, obdélník, rovnostranný trojúhelník a kruh viz obrázky 7.1. Výsledky každé metriky jsou zaznamenány v tabulce 7.1. Metrika *Area* se u všech obrazců rovná nebo značně blíží 1, jelikož počítáme s velikostí hrany buňky 1 metr a metrika *Area* je uváděna v hektarech. Jádrová oblast má rozlohu podobnou s rozlohou celkovou, protože jsme na obvodových hranicích odstranili pouze jednu buňku. Zároveň jsou zde testovány obrazce, které jsou celistvé



Obrázek 7.1: Vybrané rovinné obrazce

	Čtverec	Obdélník	Trojúhelník	Kruh	Elipsa
Area	1,0	1,0	1,0071	1,0025	1,0011
Perimeter	400,0	500,0	556,0	452,0	492,0
Core Area	0,96	0,95	0,97	0,97	0,96
Core Area Index	96,04	95,04	95,89	96,85	96,28
NCore	1,0	1,0	1,0	1,0	1,0
Contig	0,66	0,658	0,657	0,659	0,658
Gyrate	38,26	53,04	40,64	37,66	42,71
Shape	100,0	125,0	138,51	112,86	122,93
Fract	1,000	1,048	1,070	1,026	1,045
Turtle	0,048	0,042	0,049	0,113	0,112

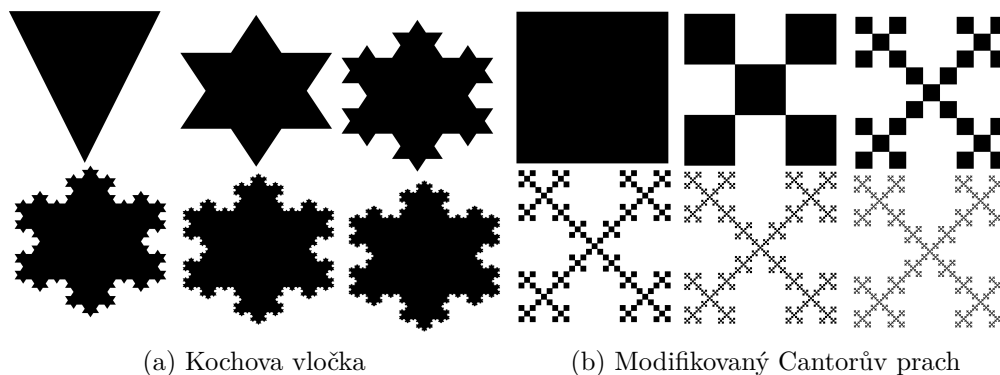
Tabulka 7.1: Výpočet metrik na rovinných obrazcích

a valná část jejich buněk se nachází mimo hraniční obvod uvnitř obrazce. Očekávané byly také hodnoty *Core Area Index*, poněvadž jádrové oblasti jsou tvořeny všemi buňkami mimo hranici obrazce. Hodnota u všech obrazců je vyšší než 95 %. Obdobně je na tom počet jádrových oblastí *Ncore*. U takto celistvých obrazců se jádrová oblast nerozdělí na více částí, proto je hodnota rovna 1. Hodnoty metriky *Contig* se drží v rozmezí 0,657 – 0,66. Podle metriky *Gyrate* má největší rozptyl buněk od středu plošky obdélník. Nabývá hodnoty 53,04 metrů.

Metriky *Fract*, *Shape* a také nová želví metrika *Turtle* by měly být vhodné k porovnávání složitostí okrajů plošek. V tabulce můžeme vidět, že metriky *Shape* i *Fract* mají pro všechny obrazce stejnou posloupnost růstu. Tedy tvarově nejjednodušší se jim jeví čtverec, následován kruhem a elipsou. Nejsložitější reliéf pak patří obdélníku a trojúhelníku. Zato nová metrika *Turtle* vyhodnotila jako obrazec s nejjednodušším tvarem obdélník. Následován trojúhelníkem a poté čtvercem, avšak s rozdílem pouze jedné tisícin. Nejsložitějšími tvary podle želví metriky nabývají elipsa s kruhem. Rozdíl mezi výsledky metrik *Fract* a *Shape* oproti metrice *Turtle* může mít souvislost s vazbou prvních dvou metrik na poslední nezmiňovanou metriku *Perimeter*, tedy celkovou velikost obvodu plošky. Jak můžeme vidět v tabulce 7.1 obě dvě metriky jsou závislé na obvodu plošky, což vyplývá také ze vzorců pro jejich výpočet uvedených v podkapitole 4.1. Naopak chování metriky *Turtle* můžeme přisuzovat změnám směru želvy při průchodu obvodovou linií. V těchto případech se jeví jednoduššími obrazce tvořené buďto rovnou obvodovou hranicí, kterou vidíme u čtverce a obdélníku, anebo obrazce, na kterých se opakují stejné sekvence vektorů, což můžeme zpozorovat u trojúhelníku. Elipsa a kruh se jeví složitější, jelikož se skládají z větších nepravidelostí v celkové posloupnosti vektorů.

7.2 Iterace rovinných fraktálů

Následující experiment se zaměřoval na vyhodnocení metrik ve dvou příkladech generování rovinných fraktálů. Fraktál můžeme chápat jako objekt, jehož geometrická struktura se neustále opakuje



Obrázek 7.2: Šest iterací dvou vybraných fraktálů

Počet iterací	1	2	3	4	5	6
Area	31,92	32,00	35,62	37,30	38,17	38,56
Perimeter	3192,0	3720,0	4952,0	6560,0	8608,0	10932,0
Core Area	31,68	31,73	35,27	36,83	37,56	37,79
Core Area Index	99,25	99,17	99,01	98,75	98,40	98,01
NCore	1,0	1,0	1,0	1,0	1,0	1,0
Contig	0,665	0,665	0,664	0,664	0,663	0,662
Gyrate	228,93	222,03	234,58	239,66	242,12	243,14
Shape	141,24	164,40	207,42	268,53	348,32	440,14
Fract	1,054	1,078	1,114	1,154	1,194	1,123
Turtle	0,013	0,018	0,021	0,030	0,051	0,071

Tabulka 7.2: Výpočet metrik šesti iterací Kochovy vločky

v něm samém. Tyto objekty, které se však běžně vyskytují v přírodě nelze jednoduše popsat pomocí eukleidovské geometrie. A právě potřeba zjednodušit popis těchto objektů vedla ke vzniku nového matematického odvětví, fraktální geometrie. V této práci se fraktály podrobněji zabírat nebudeme, ale můžu doporučit tyto publikace: [22] a [23], které se oborem fraktální geometrie zabývají.

Pro účely tohoto experimentu byl využit nástroj *OnlineFractalTools*[24] k vytvoření prvních šesti iterací tvorby fraktálu Kochovy vločky a modifikovaného Cantorova prachu tzv. *Box fractal* na rastr o velikosti 800×800 . Byla upravena barva vykreslení, šířka hranice nastavena na hodnotu 1 a odstraněn *padding* (výplň okolo obrazce). Ostatní nastavení bylo zachováno v původním stavu. Výsledkem jsou dvě sady útvarů viz obrázky 7.2a a 7.2b. Pro tyto útvary byly vypočteny tvarové metriky, jejichž hodnoty jsou uvedeny v tabulkách 7.2 resp. 7.3.

V obou případech jsou obrazce již vygenerované na určité měřítko, a proto nemusí přesně odpovídat teoretickým vlastnostem, kterých popsání fraktály nabývají. Přesto očekáváme, že se každá další iterace bude projevovat na výsledných hodnotách metrik.

Velikost rozlohy má v obou sadách opačnou tendenci růstu. U Kochovy vločky je rozloha každou iterací zvětšována, u Box fraktálu naopak snižována. Obvod se v každém cyklu zvětšuje, jelikož tvar

Počet iterací	1	2	3	4	5	6
Area	63,84	35,91	20,52	12,39	8,51	7,34
Perimeter	3196,0	5328,0	8872,0	14 784,0	24 656,0	41 112,0
Core Area	63,52	35,37	19,65	10,94	6,20	4,01
Core Area Index	99,50	98,52	95,71	88,32	72,88	54,62
NCore	1,0	1,0	1,0	1,0	1,0	1,0
Contig	0,666	0,664	0,660	0,648	0,619	0,573
Gyrate	305,70	327,78	329,83	330,06	330,42	330,66
Shape	100,0	222,28	489,53	1 050,07	2 113,14	3 794,71
Fract	1,0	1,125	1,260	1,401	1,538	1,649
Turtle	0,0116	0,0122	0,0170	0,0202	0,0268	0,0510

Tabulka 7.3: Výpočet metrik šesti iterací Box fraktálu

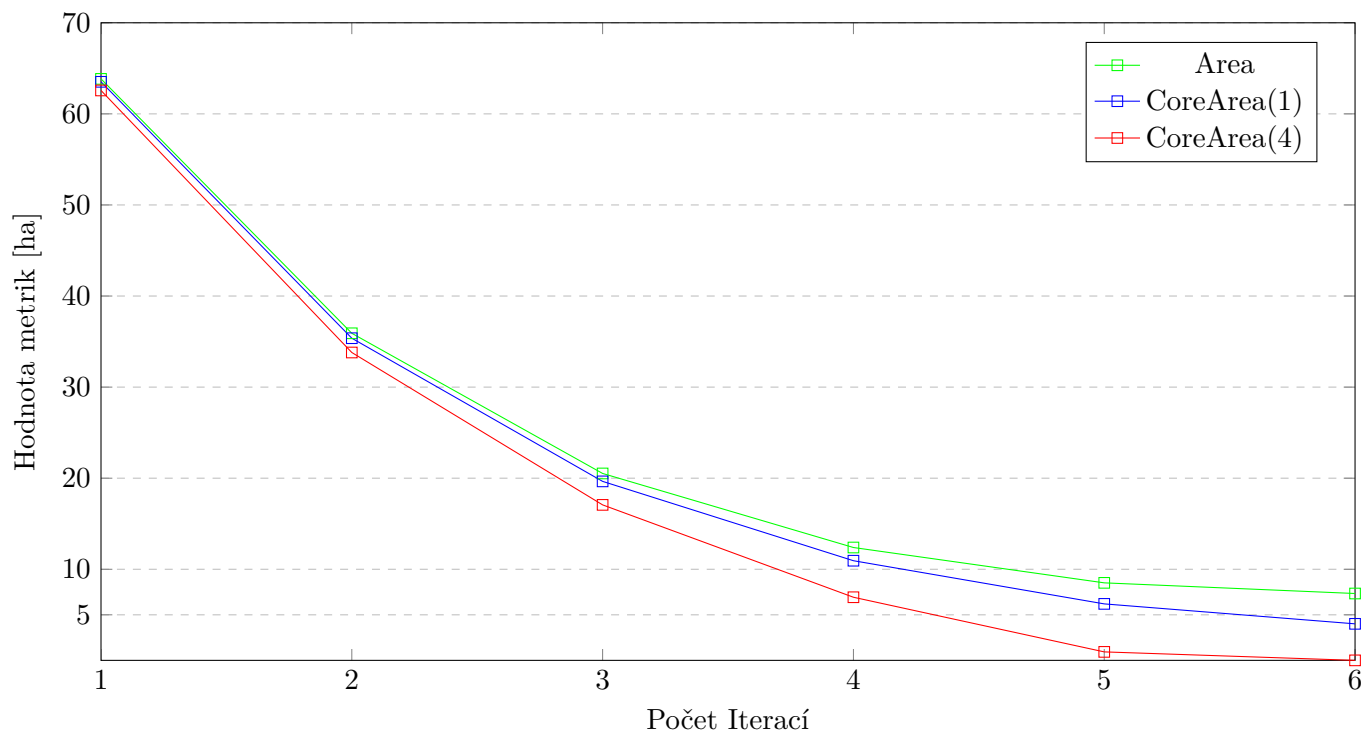
plošek je v každé iteraci komplikovanější. Pokud se podíváme na trojici metrik *Fract*, *Shape* a dokonce i *Turtle* vidíme ve všech případech na obou typech obrazců, že se výsledky s každou další iterací zvyšují, to tedy potvrzuje naše očekávání, že tvar metriky se stává složitějším. Podobnou tendenci má metrika *Contig*, která s každou vyšší iterací klesá, což znamená, že klesá celková spojitost a jednodušnost plošky.

7.3 Metriky jádrových oblastí

Obrazce Cantorova prachu (7.2b) jsou vhodnou volbou k testu jádrových metrik. Nejprve si ukážeme počet jádrových oblastí (metrika *NCore*) Při zvětšení šířky pásu (parametr *width*, který budeme odebrat, na hodnotu 4 se nám počet jádrových oblastí každé iterace fraktálu na obrázku 7.2b zvětší, jelikož se nám při takto širokém odebíraném pásu od sebe jednotlivé čtverce oddělí a vytvoří tak separovanou jádrovou oblast. Výsledky jsou uvedeny k porovnání s výsledky z tabulky 7.3 v prvních dvou řádcích tabulky 7.4. Na šesté iteraci již po odebrání pásu o šířce čtyř buněk zmizí celá ploška. Hodnota je proto rovna 0. Podobný trend můžeme očekávat také u zbylých dvou metrik jádrových oblastí uvedených v tabulce 7.4. Zmenšováním jádrové oblasti se logicky zmenšuje její rozloha a tím pádem také klesá její poměrové zastoupení k celkové rozloze plošky (metrika *Core Area Index*) s každou další iterací. Porovnání metriky *Core Area* při použití obou parametrů s metrikou *Area* je znázorněno v grafu 7.3.

Počet iterací	1	2	3	4	5	6
NCore(1)	1,0	1,0	1,0	1,0	1,0	1,0
NCore(4)	1,0	5,0	25,0	125,0	625,0	0,0
Core Area(1)	63,52	35,37	19,65	10,94	6,20	4,01
Core Area(4)	62,57	33,80	17,07	6,93	0,93	0,0
Core Area Index(1)	99,50	98,52	95,71	88,32	72,88	54,62
Core Area Index(4)	98,01	94,12	83,17	55,97	10,95	0,0

Tabulka 7.4: Porovnání metrik jádrových oblastí při odebrání pásu určité šířky



Obrázek 7.3: Porovnání metriky Area s jádrovou metrikou Core Area

Kapitola 8

Závěr

Tvarové metriky se řadí mezi kvantitativní metody vyhodnocování tvarů ploch reálného světa. Své uplatnění nacházejí především v krajinné ekologii, avšak své využití nacházejí i v dalších geografických oborech. Jedná se při nejmenším o vhodný doplněk pro vyhodnocování tvarových charakteristik ploch, přestože stále ve vědeckých kruzích panují rozpory o jejich relevantním použití. K různým potřebám je nutné vybírat správné metriky, jelikož existuje mnoho různých metrik a jejich variant, které nám v konkrétní situaci můžou pomoci s hodnocením ploch. Většina tvarových metrik je také citlivá na velikost měřítka zkoumaných map. Proto je důležité při hodnocení ploch brát na tuto skutečnost ohled. Přes všechna svá negativa jsou tvarové metriky při správné interpretaci vhodnou metodou ke zkoumání ploch reálného světa.

Hlavním přínosem práce je vytvoření knihovny metrik a návrh nové metriky využívající želví grafiku. Při jejich vytváření jsem si osvojil programování v jazyce C#. Z testování vyplynulo, že se metriky chovají korektně a vyhodnocují výsledky podle svých definic. Výsledky želví metriky se od výsledků podobných metrik při hodnocení rovinných obrazců odlišovaly svou citlivostí na pravidelnost hranic tvarů, avšak při hodnocení fraktálních obrazců nevykazovaly neočekávané výsledky. Možným rozšířením práce by byla implementace dalších metrik, kterých existuje velké množství, popřípadě vytvoření a přidání metrik nových.

Literatura

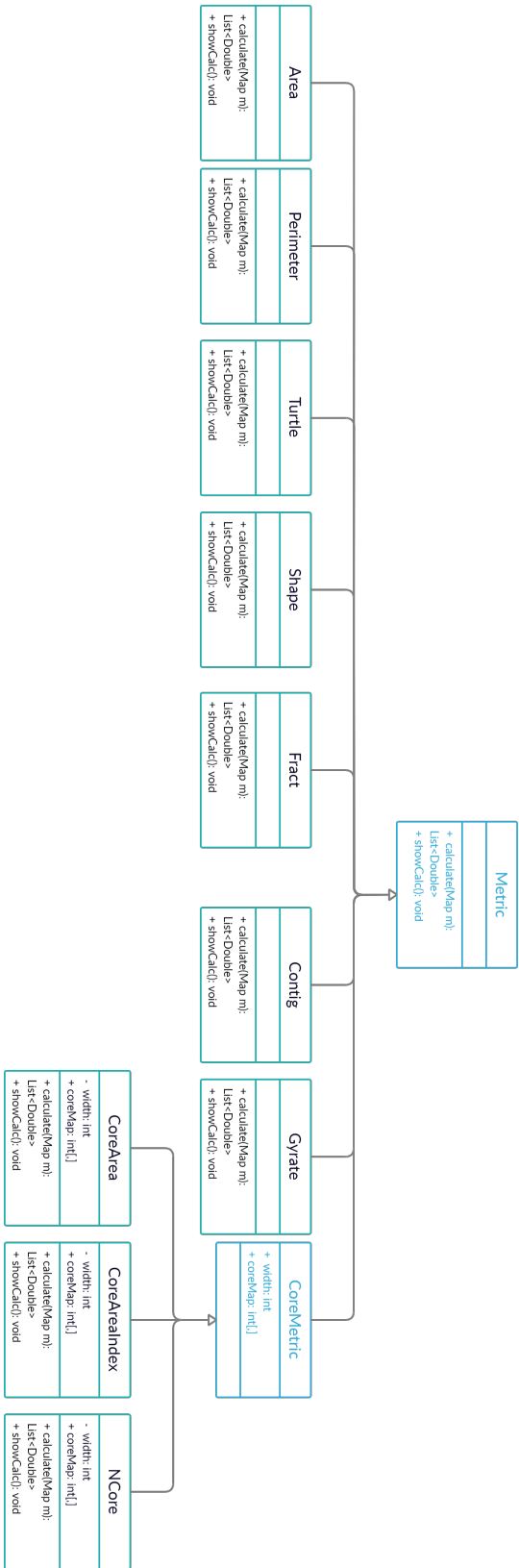
1. LAFORTEZZA, Raffaele. *Patterns and processes in forest landscapes multiple use and sustainable management*. Dordrecht: Springer, 2008. ISBN 14-020-8504-4.
2. FORMAN, Richard T. T.; GORDON, M. *Krajinná ekologie*. Praha: Academia, 1993. ISBN 80-200-0464-5.
3. GARMENDIA, Adriana; ARROYO-RODRÍGUEZ, Víctor; ESTRADA, Alejandro; NARANJO, Eduardo; STONER, Kathryn. Landscape and patch attributes impacting medium- and large-sized terrestrial mammals in a fragmented rain forest. *Journal of Tropical Ecology*. 2013-07, roč. 29. Dostupné z DOI: 10.1017/S0266467413000370.
4. PARISIEN, Marc-André; PETERS, Vernon; WANG, Yonghe; LITTLE, John; BOSCH, Erin; STOCKS, B. Spatial patterns of forest fires in Canada, 1980-1999. *International Journal of Wildland Fire*. 2006-01, roč. 15, s. 361–374. Dostupné z DOI: 10.1071/WF06009.
5. GÖKYER, Ercan. *Understanding Landscape Structure Using Landscape Metrics*. InTech, 2013. Dostupné z DOI: 10.5772/55758.
6. ČEPOVÁ, Daniela. *Stanovení vlivu geometrických vlastností povodí na odtok použitím tvarové metriky [online]*. 2015. Dostupné také z: <https://theses.cz/id/ugctxe/>. Diplomová práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta Olomouc. Vedoucí práce Ph.D. RNDR. JANA SVOBODOVÁ.
7. JIAO, Limin; LIU, Y. Analyzing the shape characteristics of land use classes in remote sensing imagery. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2012-07, roč. I-7, s. 135–140. Dostupné z DOI: 10.5194/isprsannals-I-7-135-2012.
8. TORRENS, P.M.; ALBERTI, Marina. Measuring Sprawl. *Torrens, P.M. and Alberti, M. (2000) Measuring sprawl. Working paper. CASA Working Papers (27). Centre for Advanced Spatial Analysis (UCL), London, UK*. 2001-01.
9. HARTMANNOVÁ, Sylvie. *Aplikace prostorové a tvarové metriky pro hodnocení (sub)urbanizace*. Olomouc, 2013. bakalářská práce (Bc.) UNIVERZITA PALACKÉHO V OLOMOUCI. Přírodovědecká fakulta.

10. PÁSZTO, Vít. *Prostorová informace a vybrané metody geocomputation pro její hodnocení*. Olomouc, 2015. Disertace. UNIVERZITA PALACKÉHO V OLOMOUCI. Přírodovědecká fakulta.
11. PARENT, Jason. *Shape Metrics Tool Overview*. University of Connecticut, 2009. Dostupné také z: http://clear.uconn.edu/tools/Shape_Metrics/index.htm.
12. MCGARIGAL, Dr. Kevin. *FRAGSTATS: Spatial Pattern Analysis Program for Categorical Maps*. University of Massachusetts, 2015. Dostupné také z: http://www.umass.edu/landeco/research/fragstats/documents/fragstats_documents.html.
13. MARIAS, Kostas. *The Importance of Neighborhood Scheme Selection in Agent-based Tumor Growth Modeling*. 2015. Dostupné také z: https://www.researchgate.net/figure/The-cellular-automaton-neighborhoods-von-Neumann-and-Moore-are-visually-represented-on_fig1_281581673.
14. HESSELBARTH, Maximilian H.K.; SCIAINI, Marco; WITH, Kimberly A.; WIEGAND, Kerstin; NOWOSAD, Jakub. landscapemetrics: an open-source R tool to calculate landscape metrics. *Ecography*. 2019, roč. 42, s. 1648–1657.
15. JUNG, Martin. LecoS — A python plugin for automated landscape ecology analysis. *Ecological Informatics*. 2016, roč. 31, s. 18–21. ISSN 1574-9541. Dostupné z DOI: <https://doi.org/10.1016/j.ecoinf.2015.11.006>.
16. PAVKOVÁ, Kateřina. *Toolbox pro analýzu struktury krajiny*. Olomouc, 2008. Bakalářská práce. Univerzita palackého v Olomouci.
17. *.NET documentation* [online]. Microsoft, 2021 [cit. 2021-04-12]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/>.
18. HEESCH, Dimitri van. *Doxygen* [online]. 1997 [cit. 2021-04-12]. Dostupné z: <https://www.doxygen.nl/>.
19. MALINVERNI, Eva Savina. *Change Detection Applying Landscape Metrics on High Remote Sensing Images*. 2011. Dostupné také z: https://www.researchgate.net/figure/Landscape-class-and-patch-resolution_fig5_266166276.
20. WIKIPEDIA CONTRIBUTORS. *Breadth-first search* — *Wikipedia, The Free Encyclopedia*. 2021. Dostupné také z: https://en.wikipedia.org/w/index.php?title=Breadth-first_search&oldid=1005991607.
21. GHUNEIM, Abeer George. *Contour Tracing: Square Tracing Algorithm* [online]. Montréal: McGill University, 2000 [cit. 2021-04-18]. Dostupné z: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/square.html.
22. ZELINKA, Ivan; VČELAŘ, František; ČANDÍK, Marek. *Fraktální geometrie: principy a aplikace*. 1. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-7300-191-8.

23. FALCONER, Kenneth. *Techniques in fractal geometry*. Chichester: Wiley, 1997. ISBN 04-719-5724-0.
24. *OnlineFractalTools* [online]. 2018 [cit. 2021-04-21]. Dostupné z: <https://onlinefractaltools.com/>.
25. RAPANT, Petr. *Geoinformatika a geoinformační technologie*. Ostrava: VŠB - Technická univerzita Ostrava, Hornicko-geologická fakulta, Institut geoinformatiky, 2006. ISBN 80-248-1264-9.

Příloha A

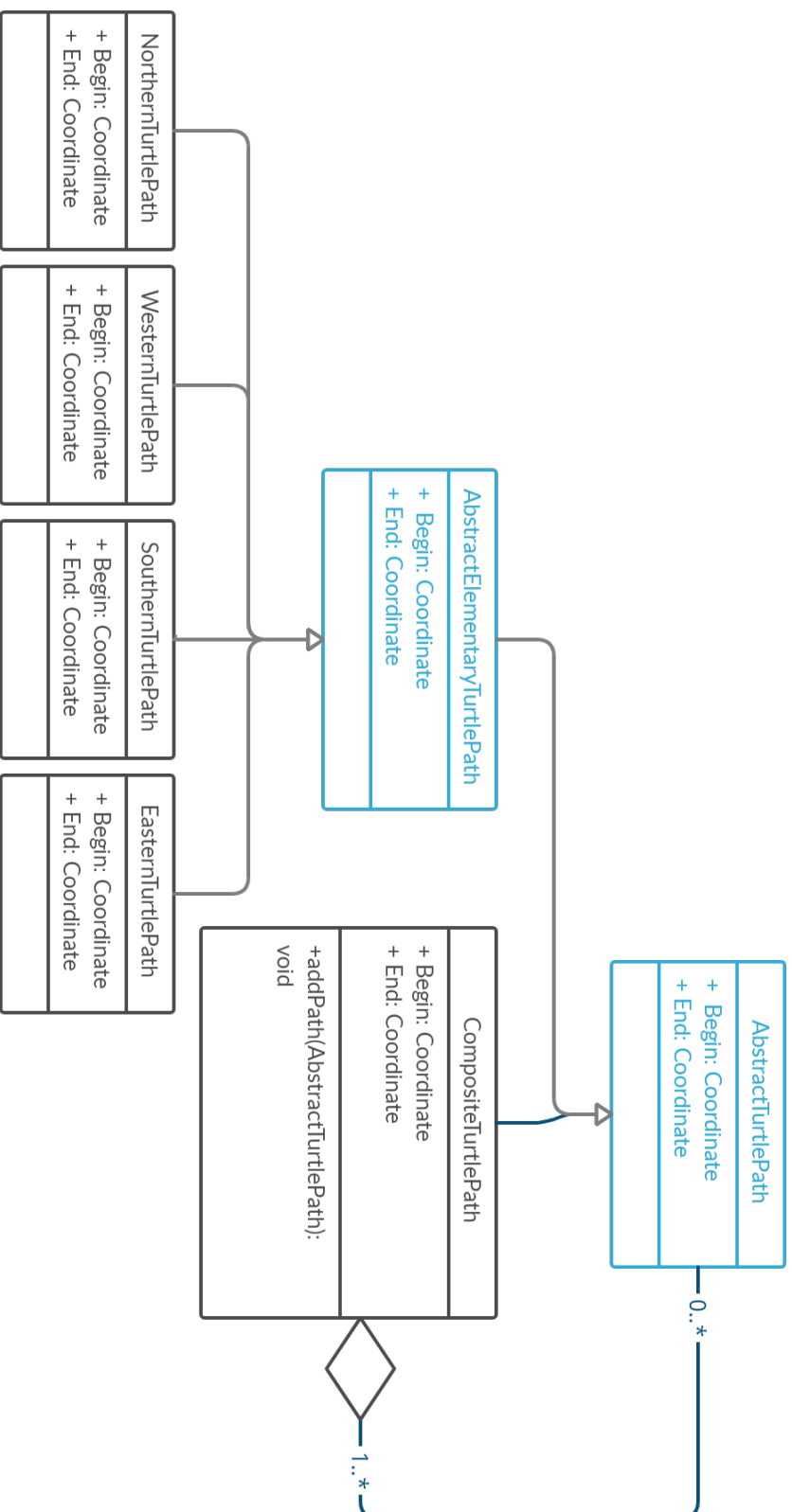
Třídní diagram Knihovny metrik



Obrázek A.1: Hierarchie knihovny metrik

Příloha B

Třídní diagram Želví metriky



Obrázek B.1: Využití návrhového vzoru kompozit v želví metrice.