

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Využití symetrie pro geodata

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Štěpánka KRUTINOVÁ**
Osobní číslo: **A17B0263P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Využití symetrie pro geodata**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s metodou hledání rovin symetrie geometrických objektů zadaných množinou bodů a její implementací.
2. Seznamte se s metodou vizualizace a klasifikace geodat zadaných množinou bodů a její implementací.
3. Spolu s vedoucí práce navrhnete využití rovin symetrie pro zlepšení klasifikace geodat a implementujte potřebné programové vybavení.
4. Metodu důkladně otestujte, výsledky testů shrňte a formulujte doporučení týkající se dané metody.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Prof. Dr. Ing. Ivana Kolingerová**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **5. října 2020**
Termín odevzdání bakalářské práce: **6. května 2021**

L.S.

Doc. Dr. Ing. Vlasta Radová
děkanka

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2021

Štěpánka Krutinová

Poděkování

Děkuji Prof. Dr. Ing. Ivaně Kolingerové za odborné vedení a pomoc při zpracování práce. Zároveň bych chtěla poděkovat Ing. Lukáši Hrudovi za vstřícnost a cenné rady týkající se softwaru.

Abstract

This bachelor thesis „The Use of Symmetry for Geodata“ deals with searching symmetries in real data. The aim was to propose and implement few methods into existing software for reflexive symmetry detection and then tests it's behaviour on the input datasets from the Earth observation. The evaluation was based on experiments always performed on all datasets with various pre-processed data. The tests were made to detect one plane as well as several planes of symmetry. The results show that the method is able to determine the symmetry planes of geo data, if the input data will be appropriately preprocessed.

Abstrakt

Tato bakalářská práce „Využití symetrie pro geodata“ se zabývá nalezením symetrií u reálných dat. Cílem bylo navrhnout a implementovat metody do již existujícího softwaru pro detekci zrcadlové symetrie a následně otestovat jeho chování na vstupní množině datasetů, obsahující data dálkového průzkumu Země. Vyhodnocování probíhalo na základě experimentů prováděných vždy na všech datasetech s různě předzpracovanými daty. Testována byla jak detekce jedné roviny symetrie, tak i více rovin symetrie. Výsledky ukazují že zkoumaná metoda je schopná určit rovinu symetrie i u geo dat, budou-li vstupní data vhodně předzpracována.

Obsah

1	Úvod	8
2	Analytická část	9
2.1	Symetrie	9
2.2	Definice symetrie	10
2.2.1	Grupa	10
2.2.2	Podgrupa	11
2.2.3	Zobrazení	11
2.2.4	Transformace systému	12
2.2.5	Stavová ekvivalence	13
2.2.6	Grupa symetrie	14
2.2.7	Symetrie v informatice	14
2.3	GEO data	15
2.3.1	LiDAR	15
2.3.2	Testovací dataset	16
2.3.3	Využití symetrie pro geo data	16
2.4	Znamé metody pro detekci	17
2.4.1	HRUDA et al.	18
2.5	Laplaceův operátor	19
3	Návrh řešení, experimenty a výsledky	24
3.1	Postup řešení	24
3.2	Experimenty	25
3.2.1	Detekce rovin symetrie	26
3.2.2	Generalizace dat	27
3.2.3	2D	30
3.2.4	Změna měřítka	31
3.2.5	Výšková pásma	32
3.2.6	Ruční výběr množiny bodů	33
3.2.7	Laplacián	36
3.2.8	RemoveSymetricPlanes	37
3.3	Zhodnocení výsledků	38
4	Závěr	39
	Literatura	40

1 Úvod

Detekce symetrií a ekvivalencí geometrických dat je obzvláště zajímavá nejen v oblastech počítačové grafiky, počítačového vidění a geometrie, ale také v mnoha technických oborech. Zjištěná symetrie je užitečná při různých problémech, jako je zarovnání objektu, komprese, symetrické úpravy nebo rekonstrukce neúplných objektů. V oblastech, jako je rozpoznávání vzorů nebo počítačové vidění, problém detekce podobnosti je zásadní, protože objekty musí být rozpoznány bez ohledu na jejich pozici a měřítko. V geometrickém modelování je symetrie důležitá sama o sobě.[13]

Studium symetrií a ekvivalence geometrických objektů přilákala mnoho badatelů a lze zaznamenat rostoucí počet příspěvků věnovaný tomuto krásnému tématu a odhalující další otevřené a náročné úkoly.

Tato bakalářská práce se zabývá hledáním symetrie v geodatech. Pro svoji práci využívám software pro detekci symetrie u 3D objektů, jehož autorem je Ing. Lukáš Hruďa. Pro převedení a vizualizaci dat využívám nástroje libLAS [2], plas.io [4] a meshLab [6].

V teoretické části práce je představena formální definice symetrie a seznámíme se s používanou metodou pro detekci symetrie. Současně se seznámíme s daty z dálkového průzkumu Země, na kterých budou prováděny experimenty.

Práce je z velké části věnována experimentům, při kterých je pozorováno chování softwaru pro detekci symetrie na vstupních geodatech. Na základě těchto experimentů a chování softwaru byly implementovány metody, které upravovaly vstupní data tak, aby výsledná rovina symetrie, detekovaná programem, byla co nejpřesnější. V závěru jsou prezentovány a zhodnoceny dosažené výsledky.

Přestože využití programové vybavení pro detekci symetrie bylo již úspěšně otestované na bodových povrchových modelech, pro jeho úspěšné využití pro geodata bylo zapotřebí navrhnout, implementovat a otestovat mnoho variant předzpracovaných dat. Proto se souhlasem vedoucí práce bylo upuštěno od částí zadání týkajících se klasifikace a práce se soustředila jen na nalezení rovin symetrie.

2 Analytická část

Tato část se zabývá symetrií, jejím popisem, její definicí pomocí teorie grup a je zde přehled již známých metod pro určení symetrie u 3D objektů. V průběhu práce na zadaném problému se ukázala potřeba využít také Laplaceův operátor, proto je součástí této kapitoly také jeho popis.

2.1 Symetrie

Symetrie je základní a všudypřítomný koncept, jedna z mnoha užitečných vlastností, zpočátku se nacházející v přírodních objektech, jež natolik inspirovala člověka, že je využit i v objektech umělých a dnes ji lze nalézt téměř všude kolem nás. Hraje také významnou roli ve vizuálním vnímání a estetice. Se začleňováním symetrie - ať už do lidské tvorby, vědy, nebo umění - souvisí i její rozpoznávání.

Zajímavým faktem je, že tak dávný koncept, jako je symetrie, dosud nenabyl obecnou definici, která by byla přijímaná napříč obory. V literatuře a na webu lze najít více definic. Zvolila jsem definici dle [20].

Pro detekci roviny symetrie dnes existuje mnoho metod viz kapitola 2.4.

Objekt je symetrický, pokud po provedení nějaké operace zůstane nezměněn. Operace je myšlený nebo skutečný pohyb objektu, při kterém se objekt dostává z pozice výchozího stavu (charakterizována určitým tvarem) do stavu, který je zcela shodný se stavem na začátku pohybu.

Na základě typu operace můžeme symetrii rozdělit:

- středová symetrie → střed souměrnosti
- rovinová, zrcadlová symetrie → rovina souměrnosti
- rotační, osová symetrie → osy souměrnosti

Symetrie může být dokonalá (silná) nebo přibližná (slabá). Dále se symetrie může týkat celého objektu anebo pouze jeho části. Podle toho pak

rozlišujeme symetrii globální (celý objekt) nebo lokální (část objektu).

Poněkud akademičtěji vyjádřeno: **symetrie je neměnnost (invariance) formy daného objektu při nějaké transformaci.** [19]

2.2 Definice symetrie

Se symetrií úzce souvisí matematická disciplína, nazývaná se *teorie grup*, spadající do oblasti algebry. Velká část následujících definic a textu byla čerpána z [20] a [11].

2.2.1 Grupa

Grupu lze chápat jako algebraickou strukturu s jednou binární operací, která má jisté vlastnosti. Binární operací může být buď aditivní operace - \oplus (sčítání) nebo multiplikativní operace - \otimes (násobení). Podle toho, zda binární operaci chápeme aditivně nebo multiplikativně, má definice příslušnou podobu.

Pokud tedy jako binární operaci uvažujeme násobení, tedy \otimes , pak množina G je grupou, splňuje-li následující vlastnosti:

1. **Uzavřenost:**

$$\forall a, b \in G : a \otimes b, b \otimes a \in G$$

2. **Asociativita:**

$$\forall a, b, c \in G : a \otimes (b \otimes c) = (a \otimes b) \otimes c$$

3. **Neutrální prvek:** Mějme prvek e značící prvek neutrální, pak platí:

$$a, e \in G : a \otimes e = e \otimes a = a$$

Lze dokázat, že prvek e je vždy unikátní.

4. **Inverzní prvek:** Mějme prvek a^{-1} značící prvek inverzní, pak platí:

$$\forall a \in G \exists a^{-1} : a \otimes a^{-1} = a^{-1} \otimes a = e$$

2.2.2 Podgrupa

Nechť G je multiplikatívni grupa. Podmnožina G' grupy G se nazývá podgrupa grupy G , má-li tyto vlastnosti:

- a) $1 \in G'$
- b) jestliže $a, b \in G$, pak i $a, b \in G'$ a $a^{-1} \in G'$

Každá grupa G má alespoň dvě podgrupy (mimo triviální jednoprvkové, která má jen jednu podgrupu), a to :

- a) celou grupu G
- b) grupu tvořenou pouze prvkem neutrálním.

2.2.3 Zobrazení

Zobrazením rozumíme jakýkoli předpis, který každému prvku z jedné množiny přiřadí jednoznačným způsobem nějaký prvek z druhé množiny (nebo z té samé). Tedy zobrazení M z množiny A do množiny B zapíšeme jako $A \rightarrow B$ nebo $M : A \rightarrow B$.

Prvek z B , který je v souladu s nějakým prvkem z A , se nazývá obraz prvku A . Prvek A se pak nazývá objekt prvku B . B tedy může být obrazem více než jednoho prvku (many-to-one) - surjekce. Pokud je ale každý obraz B obrazem jedinečného objektu v A (one-to-one), pak toto zobrazení nazýváme injektivní (prosté).

Pokud $A = B$, mluvíme o zobrazení na množině. Je-li zobrazení prosté a na, pak se jedná o tzv. *bijekci*. Pro zobrazení lze nalézt inverzní zobrazení právě tehdy, když je bijekcí. Takové zobrazení pak nazveme invertibilním.

2.2.4 Transformace systému

Jelikož jsme v kapitole 2.1 symetrii definovali jako invariaci formy objektu při nějaké transformaci, je nutné si pojem transformace systému nadefinovat, čemuž je věnována tato podsekce.

Systémem rozumíme to, co zkoumáme - v našem případě tedy všechny body vstupních dat. Stavový prostor systému je množina všech možných stavů systému, kde stavem rozumíme určitou formu, ve které se naše množina vstupních bodů může nacházet - např. po škálování či posunutí.

Transformace systému je zobrazení stavového prostoru systému do sebe, tzn. každý stav má obraz, který je rovněž stavem. Více stavů může mít stejný obraz, ale ne každý stav musí být obrazem.

Transformaci T značíme:

$$u \xrightarrow{T} v \text{ nebo } v = T(u)$$

kde u a v reprezentují stavy systému. Stav v je obraz stavu u po transformaci T .

Složení transformací je smysluplné, pokud transformace působí ve stejném stavovém prostoru, což předpokládáme. Kompozicí transformací rozumíme zobrazení stavového prostoru do sebe, tudíž se jedná o transformaci, a lze tedy říci, že množina transformací systému je uzavřená.

Kompozici získáme postupnou aplikací jednotlivých transformací - tedy pro transformace T_1 a T_2 je jedna z možných kompozic T_2T_1 , značící výsledek aplikace nejdříve transformace T_1 a poté T_2 :

$$\begin{array}{ccccc} u & \xrightarrow{T_1} & v & \xrightarrow{T_2} & w \\ & \searrow & & \nearrow & \\ & & T_2T_1 & & \end{array}$$

Jinak řečeno - kompozici dvou transformací lze nadefinovat jako aplikaci výsledku jedné transformace na výsledek jiné.

Pro kompozici transformací postupnou aplikací dále platí:

$$T_3(T_2T_1) = (T_3T_2)T_1$$

tzn. asociativita, kterou lze dokázat.

Transformace I taková, pro kterou platí:

$$u \xrightarrow{I} u \text{ nebo } u = I(u)$$

pro každý stav stavového prostoru, se nazývá neutrální transformace. Pouze pro transformace T , které jsou bijektivní, mohou být definovány inverzní transformace, které se značí T^{-1} , a platí pro ně:

$$u \xrightarrow{T} v \text{ nebo } u = T(u)$$

$$v \xrightarrow{T^{-1}} u \text{ nebo } u = T^{-1}(v)$$

$$T^{-1}T = TT^{-1} = I$$

Vidíme, že se jedná o grupu. Množina všech bijektivních transformací stavového prostoru systému tvoří grupu při kompozici transformací postupnou aplikací, která se nazývá *transformační grupa*.

2.2.5 Stavová ekvivalence

Nadefinujme si množinu E , jejíž prvky mohou být symboly (bity, číslice, písmena), body, či jakýkoli jiný název, v závislosti na praktické studii symetrie, kterou budeme provádět. V této množině poté můžeme nadefinovat objekt jako funkci, jež má svůj vstupní argument v E . Objekty jsou funkce definované na E způsobem, který nyní není nutné specifikovat, za předpokladu, že jsme schopni deklarovat, kdy objekt je totožný s jiným objektem. Tento druh identity má smysl za následujících tří předpokladů:

- a) Objekt je sám se sebou totožný.
- b) Pokud je objekt totožný s druhým objektem, pak je i druhý totožný s prvním.
- c) Pokud je objekt totožný s druhým objektem a tento druhý objekt je totožný s třetím, pak i první objekt je totožný s třetím.

[19]

To lze vyjádřit matematicky, a to následovně: relace ekvivalence pro stavový prostor systému nastane mezi dvěma stavy při platnosti následujících podmínek:

a) **Reflexivita** - každý stav je v relaci sám se sebou, tedy

$$a \equiv a$$

b) **Symetrie** - je-li první stav v relaci s druhým, tak i druhý stav musí být v relaci s prvním

$$a \equiv b \Leftrightarrow b \equiv a$$

c) **Transitivita** - je-li první stav v relaci s druhým a druhý se třetím, pak i první je v relaci se třetím

$$a \equiv b, b \equiv c \Rightarrow a \equiv c$$

Uvedené podmínky platí pro všechny stavy množiny.

2.2.6 Grupa symetrie

Grupou symetrie nazveme podgrupu transformační grupy, která je tvořena množinou všech invertibilních transformací symetrie stavového prostoru systému pro relaci ekvivalence.

2.2.7 Symetrie v informatice

Zatímco matematika popisuje objekty především rovnicemi, počítačová grafika obvykle pracuje s množinou bodů nebo s trojúhelníkovou sítí. V počítačové grafice má informace o symetrii v objektech různé aplikace, jako je zarovnání objektu do souřadného systému, komprese, symetrické úpravy nebo rekonstrukce neúplných objektů [13].

V posledních letech se výzkum symetrie v oblasti počítačové grafiky zaměřil na symetrie trojúhelníkových povrchových modelů nebo velké sady bodů generované 3D skenováním, což nás přivádí k další kapitole (2.3), kde si tyto sady bodů představíme.

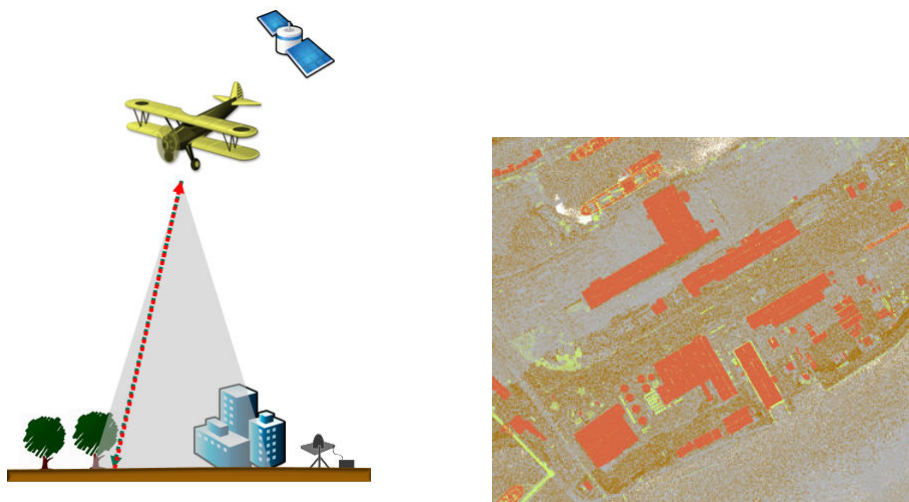
2.3 GEO data

Protože symetrii máme hledat u dat získaných z výškového průzkumu Země, je tato kapitola věnována seznámení se s těmito daty a současně zde budou představeny datasety, které byly součástí zadání práce a s nimiž budeme dále v práci pracovat.

2.3.1 LiDAR

LiDAR je zkratka vzniklá z kombinace anglického *light* and *radar*, tedy světla a radaru. Jedná se o optickou techniku dálkového průzkumu Země, která využívá laserové světlo k hustému vzorkování povrchu Země a produkuje vysoce přesná měření.

Post-zpracovaná prostorově organizovaná lidar data jsou pak známá jako mračna bodů (z anglického point cloud). Počáteční mračna bodů jsou velké sbírky 3D výškových bodů, které zahrnují x , y a z souřadnice spolu s dalšími atributy, jako jsou např. časová razítka GPS. Laser se setkává se specifickými povrchovými prvky, které jsou s následným zpracováním mračna bodů klasifikovány. Specifickým prvkem může být jak nadmořská výška, budovy, lesy, tak i dálniční nadjezdy, mosty a cokoli dalšího. Pro tuto práci ale není důležité to, jak jsou taková data získávána, proto je zde pro představu pouze názorný obrázek 2.1, v jehož zdroji [3] se lze dočíst více.



(a) Ukázka získávání dat pomocí techniky LiDAR [3]

(b) Post zpracovaná LiDAR data, tzv. point cloud [3]

Obrázek 2.1: Ukázka získávání dat a jejich post zpracování

Důležité pro nás je to, v jakém formátu jsou získavaná vzdušná data ukládána. Původně byly dodávány pouze ve formátu ASCII, ale vzhledem k obrovské velikosti nasbíraných LiDAR dat byl brzy přijat formát nazvaný LAS, což je průmyslový standardní binární formát, který spravuje a standardizuje způsob, jakým byla data lidarů organizována a šířena. Tento formát používá čtyři typy záznamu. Prvním záznamem je povinný blok, popisující soubor a základní metadata, včetně počtu bodů a x , y , z souřadnic jednotlivých bodů. Další záznamy jsou volitelné a mohou obsahovat různé typy dat, jako jsou barevné kanály, intenzita návratu impulzu, čas GPS, klasifikace, úhel skenování a mnoho dalšího [1].

Současné platformy LiDAR zahrnují pozemní, automobilové, vzdušné a satelitní platformy. Takto získané údaje jsou široce používána nejen ve 3D rekonstrukcích, ale aplikace nalezneme i v lesnictví a v hydrologických studiích.

2.3.2 Testovací dataset

Dataset, jež budeme využívat, obsahuje 6 LAS souborů. Zobrazíme-li tyto soubory programem `plas.io` [4] (viz Příloha 1), uvidíme, že se jedná o data zachycující nejen přírodu, ale i zastavěné městské části (budovy, ulice s chodníky) či vícepruhové silnice včetně mostů, železnic, kruhových objezdů a křižovatek. Tyto zachycené objekty už na první pohled vykazují známky symetrie.

2.3.3 Využití symetrie pro geo data

Jak již bylo zmíněno v úvodu a v kapitole 2.1, symetrie v přírodních objektech natolik inspirovala člověka v jeho tvorbě, že se dnes nachází všude kolem nás. Ve skutečném světě mnoho objektů s extrémně komplikovanými geometrickými tvary vykazují prvky symetrie. Studie [5] se zaměřuje na využití vnitřní charakteristiky symetrie za účelem usnadnění registrace mračna bodů s čímž souvisí i využití pro kompresi.

Další oblastí, kde by symetrie mohla být nápomocná je samotné zpracování datasetů z dálkového průzkumu Země (= EO datasetů). Typické úlohy řešené při zpracování EO dat se týkají sémantické segmentace a rozpoznávání objektů. Sémantická segmentace spočívá v přiřazení sémantických štítků (tříd) bodům, jako je země, budovy nebo vegetace. Je součástí procesu fúze dat, který srovnává data z různých zdrojů s jejich geografickými

polohami, a následně používá tyto prvky jako vstupy do sémantické segmentace a na základě této segmentace provádí rozpoznávání jednotlivých objektů (např. botanických stromů). [16]

V následujících odstavcích této kapitoly jsou zmíněny metody pro detekci a využití symetrie pro geo data. Jelikož nebylo nutné důkladně pochopit jejich popis a implementaci, jedná se pouze o stručný výběr existujících metod.

Kerber et al. [15] představili metodu detekce parciálních symetrií u velmi velkých mračen bodů z 3D skenování města.

Wang H. a kol. [21] navrhli automatický rozklad a modelování složených budov z LiDAR dat na sémantická primitiva s parametrickými tvary využitím symetrie obsažené ve struktuře budovy.

Clode a kol. [7] představovali jeden z prvních přístupů k detekci silnic z údajů LiDAR, využívajících jak údaje o intenzitě, tak o dosahu. Symetrie je implicitně detekována a použita při výpočtech os, šířky a směru. Metoda dále poskytuje vektorizaci.

2.4 Známé metody pro detekci

Jelikož vstupními soubory jsou mračna bodů získaná z geodat, hledáme metodu, která by našla symetrii těchto mračen. Přestože cílem je použití a testování metody popsané v kapitole 2.4.1, v literatuře [13] lze vyčíst hned několik metod, které lze použít na mračnu bodů. Takové metody nevyžadují kromě souřadnic bodů žádné další informace. Protože metody jsou poměrně obtížné a jejich pochopení nebylo předmětem této práce, vybrala jsem prameny, které se zdály pro daný problém vhodné. Mezi tyto metody patří např. Mitra et al. [18], Lipman et al. [17], Combes et al. [8] a Karakala et al. [14].

Existující metody symetrie v počítačové vědě jsou ale často vyladěny na skenování objektů vytvořených člověkem z různých obecně přístupných databází. Na jedné straně to umožňuje porovnání metod, ale na straně druhé tyto metody někdy nefungují dobře na datech z reálného života, která jsou obvykle neúplná nebo jen slabě symetrická.

2.4.1 HRUDA et al.

Metoda byla implementována v softwaru, využívaném k experimentům popsaným v kapitole 3, proto zde bude nastíněna hlavní idea této metody. Její důkladnější popis lze nalézt v [12].

Jedná se o metodu pro detekci globální roviny zrcadlové symetrie pro 3D objekty, která je používána jak pro dokonalé objekty, tak i pro detekci přibližné symetrie nedokonalých objektů, dokonce i u objektů s chybějícími částmi. Stejně jako předešlé metody, i tato je funkční na objektech reprezentovaných pouze množinou bodů, a tudíž neklade prakticky žádné požadavky na vstupní data.

Metoda je založena na maximalizaci míry symetrie účelovou funkcí (funkce 2.1), která vyhodnocuje symetrii objektu vzhledem k dané rovině. Mračno vstupních bodů je zjednodušeno na velmi nízký počet bodů a páry bodů tohoto zjednodušeného mračna bodů se používají k vytvoření řady kandidátských rovin, z nichž vybereme tu s nejvyšší mírou symetrie. Z časových důvodů je míra symetrie vypočítána na jiné zjednodušené verzi mračna vstupních bodů. Dále pomocí lokální optimalizace dojde k nalezení konečné roviny symetrie.

$$s_x(p) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} \varphi(\|r(p, x_i - x_j)\|) \quad (2.1)$$

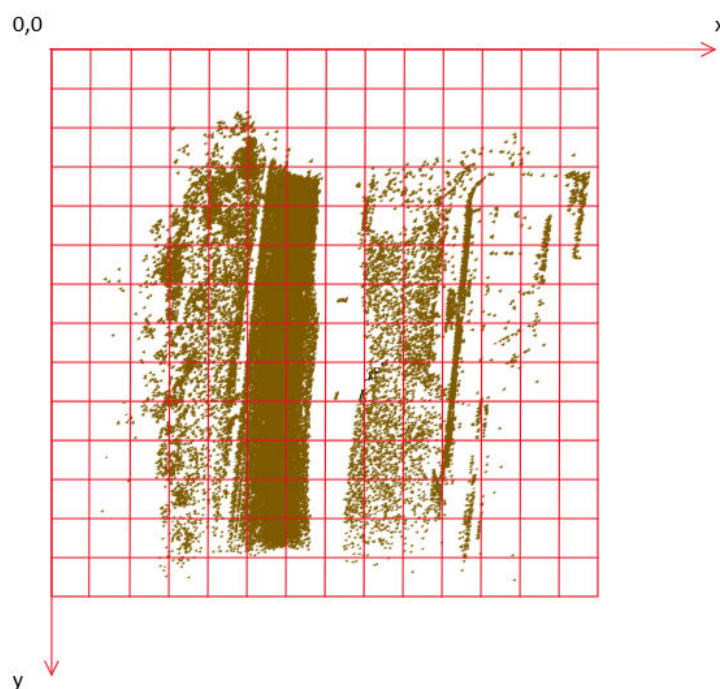
Míra symetrie také obsahuje váhy, které umožňují použít některé další informace o vstupním objektu ke zlepšení výkonu, ty ale nejsou ve výchozím nastavení používány a jejich hodnota je nastavena na 1.

2.5 Laplaceův operátor

Tato metoda vychází z metod využívaných pro zpracování digitálního obrazu. Obrazem je myšlena každá vizuální scéna, která je získána transformací předmětového prostoru do prostoru obrazového. Matematickým modelem 2D obrazu je obrazová funkce dvou proměnných $f(x, y)$, kde x a y jsou prostorové souřadnice v jednotlivých bodech obrazu a hodnota f je úměrná jasu v těchto bodech.

V našem případě se nejedná o digitální obraz, ale mračno bodů lze promítnout do dvourozměrného pole diskrétní funkce $f(i, j)$, kde indexy jsou souřadnice jednotlivých bodů a hodnoty těchto prvků neodpovídají jasu, ale výšce, která je udávána souřadnicí z .

V následujícím textu je obrazem vždy myšleno mračno bodů promítnuté do dvourozměrného pole a buňku pak považujeme za ekvivalent pixelu u digitálního obrazu. Většina textu byla převzata z [10].



Obrázek 2.2: dataset 06 - promítnutí do gridu

Jemnost dělení mřížky je důležitým aspektem. Jak vidíme na obrázku, zvolíme-li příliš řídkou mřížku, nastane to, že jedné buňce bude náležet velké

množství bodů a pro následné výpočty mezi buňkami nemusí být takové rozdíly, abychom dokázali určit body patřící hranám.

Určit body patřící hranám chceme proto, že místa odpovídající významným hranám nesou více informace než jiná místa v obraze. Hranu lze definovat jako oblast, kde se výrazně mění hodnota výškové souřadnice. Jedná se o přechod z určité hodnoty výšky na jinou hodnotu nebo oddělení stejných či podobných hodnot určitou oblastí s výrazně rozdílným parametrem. Je tedy dána vlastnostmi okolí.

Procesy prováděné se vstupním obrazem lze označit rovnicí

$$g(x, y) = T[f(x, y)] \quad (2.2)$$

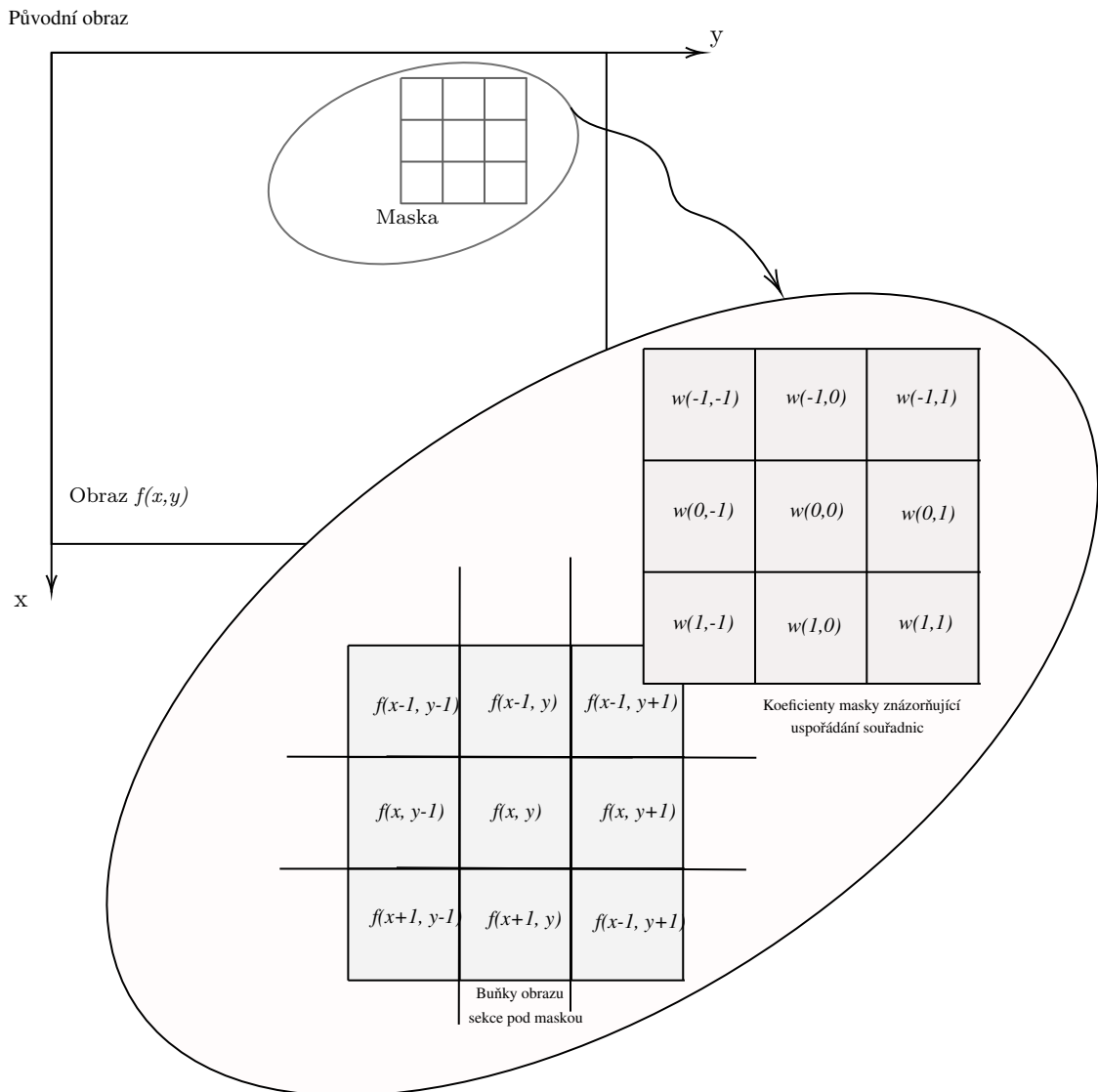
kde $f(x, y)$ značí vstupní obraz, $g(x, y)$ je zpracovaný obraz a T je operátor na f , definovaný přes nějaké okolí (x, y) . Základním přístupem při definování sousedství kolem bodu (x, y) je použití čtvercové nebo obdélníkové oblasti podobrazů se středem na (x, y) , jak ukazuje obr. 2.3. Střed podobrazu je přesunut z buňky do buňky počínaje v levém horním rohu. Operátor T je aplikován na každé místo (x, y) , čímž na tomto místě získáme výstup g .

Nejjednodušší formou T je situace, kdy okolí má velikost $1 * 1$ (tj. pouze 1 buňka). V tomto případě g závisí pouze na hodnotě $f(x, y)$. S větším okolím roste i flexibilita. Obecný přístup je určení hodnoty $g(x, y)$ použitím funkce f v předdefinovaném okolí (x, y) . Jedním z hlavních přístupů je použití tzv. masek, neboli filtrů. Masky je v zásadě malé (nejčastěji 3×3) 2D pole, jejíž hodnoty nazýváme koeficienty.

Mechanika prostorové filtrace je znázorněna na obr. 2.3. Proces jednoduše spočívá v pohybu masky z buňky do buňky v mřížce. V každé buňce (x, y) se odezva filtru vypočítá pomocí předdefinovaného vztahu. Pro masku $3 * 3$ zobrazenou na obr. 2.3 je výsledek R lineárního filtrování s maskou filtru v bodě (x, y) na obrázku součet součinů koeficientů masky s odpovídajícími buňkami přímo pod maskou. Matematicky zapsáno rovnicí 2.3.

$$R = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) \quad (2.3)$$

kde $w(i, j)$ jsou koeficienty masky a $f(x + i, y + j)$ značí hodnoty buňek obrazu pod maskou.



Obrázek 2.3: Mechanika prostorové filtrace. Zvětšený výkres znázorňuje masku 3*3 a část obrázku přímo pod ní. Dostupné z [10].

Filtry lze rozdělit na lineární a nelineární. Lineární filtr je takový filtr, jehož výstup je získán lineární kombinací vstupu. Na rozdíl od lineárních filtrů nestanovuje nelineární filtr výstupní hodnotu jako lineární kombinaci vstupních hodnot obrazu, ale podle jiného algoritmu vybírá některou z hodnot ve stanoveném okolí. Příkladem nelineárních filtrů je minimální, resp.

maximální hodnota z okolí u filtru typu minimum, resp. maximum. Protože chceme určit hodnotu buňky na základě kombinace jejího okolí, využijeme filtr lineární.

Obecně je lineární filtrování obrazu f o velikosti $M * N$ s maskou filtru o velikosti $m * n$ dáno výrazem

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (2.4)$$

kde $a = (m - 1)/2$ a $b = (n - 1)/2$. Pro kompletní filtrování obrazu musí rovnice být použita pro $x = 0, 1, 2, \dots, M - 1$ a $y = 0, 1, 2, \dots, N - 1$. Tímto způsobem zajistíme, že maska zpracuje všechny buňky v obraze.

Důležitým hlediskem, které musíme při implementaci zvážit, je otázka, co se stane, když se střed masky přiblíží na okraj obrazu. Tuto otázku lze vyřešit dvěma způsoby.

1. Pokud je požadováno, aby výsledný filtrovaný obraz byl stejné velikosti jako obraz původní, použijeme při filtrování na okrajích pouze pomocí částečnou masku.
2. Není-li požadována shodná velikost původního a vyfiltrovaného obrazu, omezíme vzdálenost středu masky od okraje na $(n - 1)/2$.

V této části si představíme filtrování pomocí Laplaceova operátoru. Laplacian je pro funkci (obraz) $f(x, y)$ dvou proměnných definován jako

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.5)$$

Protože derivace jakéhokoli řádu jsou lineární operace, je Laplacian lineární operátor. Digitální implementace dvourozměrného Laplacianu se získá sečtením částečné derivace druhého řádu ve směru x s částečnou derivací druhého řádu ve směru y .

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y) \quad (2.6)$$

Rovnice 2.6 může být implementována pomocí jedné z masek znázorněných na obrázku 2.4.

0	1	0
1	-4	1
0	1	0

a) 4okolí

1	1	1
1	-8	1
1	1	1

b) 8okolí

0	-1	0
-1	4	-1
0	-1	0

**c) 4okolí
inverzní**

-1	-1	-1
-1	8	-1
-1	-1	-1

**d) 8okolí
inverzní**

Obrázek 2.4: Laplacian operátory

Chceme-li do definice Laplacianu začlenit i úhlopříčné směry, je nutné přidat k rovnici 2.6 další dva výrazy (pro každý ze dvou diagonálních směrů). Protože každý diagonální člen také obsahuje člen $-2f(x, y)$, součet odečtený od rozdílových členů bude nyní $-8f(x, y)$. Maska použitá k implementaci této nové definice je uvedena na obr. 2.4(b).

V praxi se často používají i další dvě masky zobrazené na 2.4(c) a (d). Jsou založeny na definici Laplacianu a nabývají pouze opačných (záporných) hodnot oproti těm, které jsme použili.

3 Návrh řešení, experimenty a výsledky

V následující části je popsán postup řešení a experimenty, provedené v softwaru pro detekci symetrie na níže popsaných vstupních datech, včetně výsledků.

Všechny pokusy byly prováděny na počítači s čtyřjádrovým procesorem Intel® Core™ i78550U (8th Gen) CPU s frekvencí 1.8GHz. Vyrovnávací paměť 8 MB Intel® Smart Cache (= L3, L1 - 256 kB, L2 - 1 MB) a 16GB paměti RAM.

3.1 Postup řešení

Software (popsaný v kapitole 2.4.1) je funkční a již byl vyzkoušen pro povrchové geometrické modely. Byl funkční jak pro bodové, tak i triangulované geometrické modely, ale také pro body ležící na křivkách. Software však nebyl testován pro geo data.

Prvním krokem řešení je tedy vyzkoušet software na vstupních souborech obsahujících geo data a na základě výsledků navrhnout a implementovat doplňky.

Jak je zřejmé z prvního kroku řešení, další návrhy jsou zcela závislé na výsledcích získaných z prvního experimentu a z tohoto důvodu nelze nejdříve udělat celý návrh řešení, které by bylo možné implementovat a poté testovat. Namísto popsání celku řešení bude tedy popsán sled experimentů postupně tak, jak byly navrhovány na základě výsledků předchozích pokusů.

U každého experimentu bude popsán naimplementovaný doplněk a budou zhodnoceny výsledky na základě nalezených symetrií znázorněných na obrázcích.

3.2 Experimenty

Popis vstupních dat

Vstupem je soubor množiny bodů. Každý z bodů je popsán souřadnicemi x , y a z . Formát vstupního souboru pak vypadá následovně:

$$\begin{array}{c} N \\ x_1 \ y_1 \ z_1 \\ x_2 \ y_2 \ z_2 \\ \vdots \\ x_N \ y_N \ z_N \end{array}$$

kde N je celkový počet bodů v souboru a x_i y_i z_i jsou souřadnice i -tého bodu.

Protože původní testovací data byla ve formátu LAS, bylo nezbytné si tato data převést na data přijímaná zmíněným programem, tedy do výše uvedeného formátu, k čemuž posloužila knihovna libLAS, respektive LAS-tools [2].

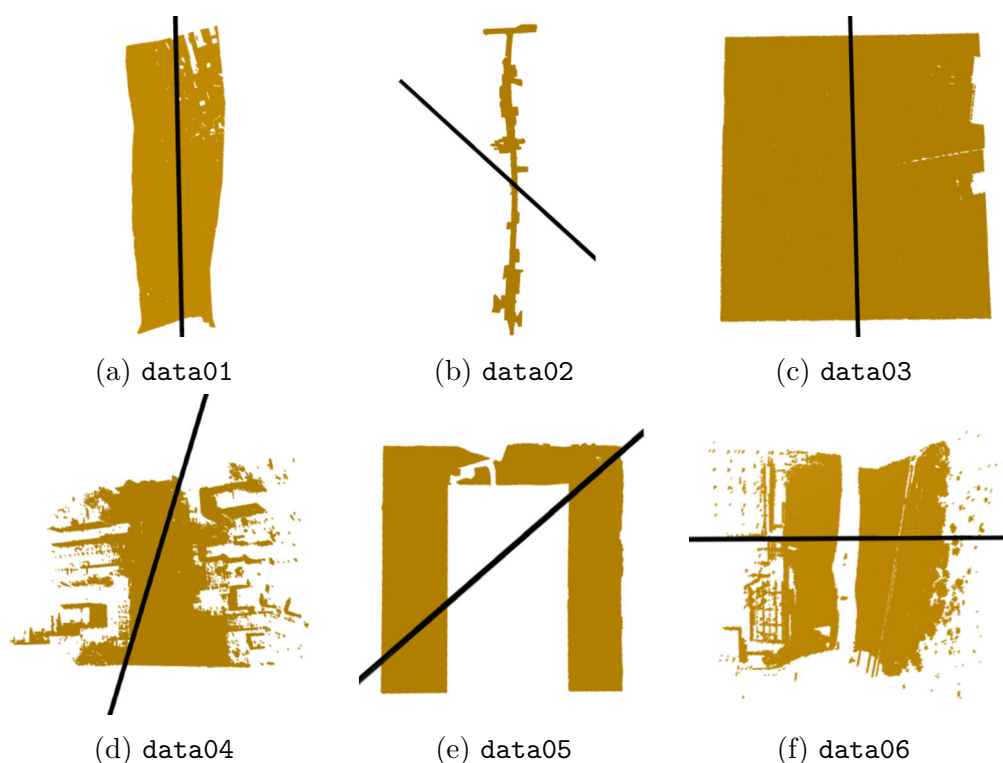
V tabulce 3.1 je uveden seznam všech souborů, na kterých experimenty probíhaly. Všechny tyto soubory jsou zvizualizovány v příloze 1. Příloha obsahuje jak pohled z půdorysu, tak i pohled perspektivní.

Seznam vstupních souborů	
Název souboru	Počet bodů v souboru
data01.pc	4.408.762
data02.pc	3 525 170
data03.pc	0 559 802
data04.pc	6 541 983
data05.pc	6 394 402
data06.pc	4 748 908

Tabulka 3.1: Seznam vstupních souborů, včetně jejich počtu bodů.

3.2.1 Detekce rovin symetrie

Program byl vyzkoušen na všech datech, výsledky ovšem nebyly uspokojivé, ačkoli detekce roviny symetrie proběhla u všech datasetů. Výsledky ukázaly, že metoda výslednou rovinu symetrie určuje na základě celkového tvaru objektu. Detekované roviny všech vstupních datasetů jsou zobrazeny na obrázku 3.1.



Obrázek 3.1: Detekce jedné roviny symetrie u dat původních.

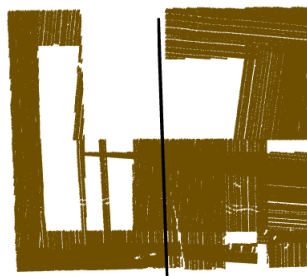
V následujících kapitolách (3.2.2, 3.2.3, 3.2.4 a 3.2.5) jsou provedeny pokusy s upravenými datasety, které by mohly vést ke zlepšení nalezení os symetrie a jejich následné vizualizace.

Protože vstupní data obsahují velké množství bodů (řádově miliony) a práce s rozsáhlejší vstupní množinou bodů zbytečně zatěžuje PC a zpomaluje práci, nabízí se úprava dat redukcí vstupní množiny bodů. Je vhodné množství bodů zredukovat tak, aby model objektu minimálně ztratil na přesnosti. Nalezení roviny symetrie ze zredukované množiny bodů je popsáno v následující kapitole 3.2.2.

3.2.2 Generalizace dat

Generalizace je výběr a zjednodušení detailů zobrazovaných objektů s ohledem na měřítko a účel mapy. Při generalizaci je cílem stanovit co je zásadní a které body naopak můžeme zcela vypustit [9]. Body lze odstraňovat pravidelně, a to tak, že je ponechán pouze každý x -tý bod, kde x jsme si rozumně, vzhledem k počtu bodů v mračnu, zvolili.

Možná je ale také náhodná eliminace některých bodů. Uživatel programu určí, jaké % bodů chce ponechat, a na základě náhodně vygenerovaného čísla se u každého z bodů rozhodne, zda bude do výsledné množiny zahrnut, či nikoliv. Chceme tím dosáhnout „rozbití“ celkového tvaru objektu. Jak je vidět na obrázku 3.2, při nevhodně zvoleném poměru počtu ponechaných a odstraněných bodů může dojít i k odstranění důležitých částí objektu.

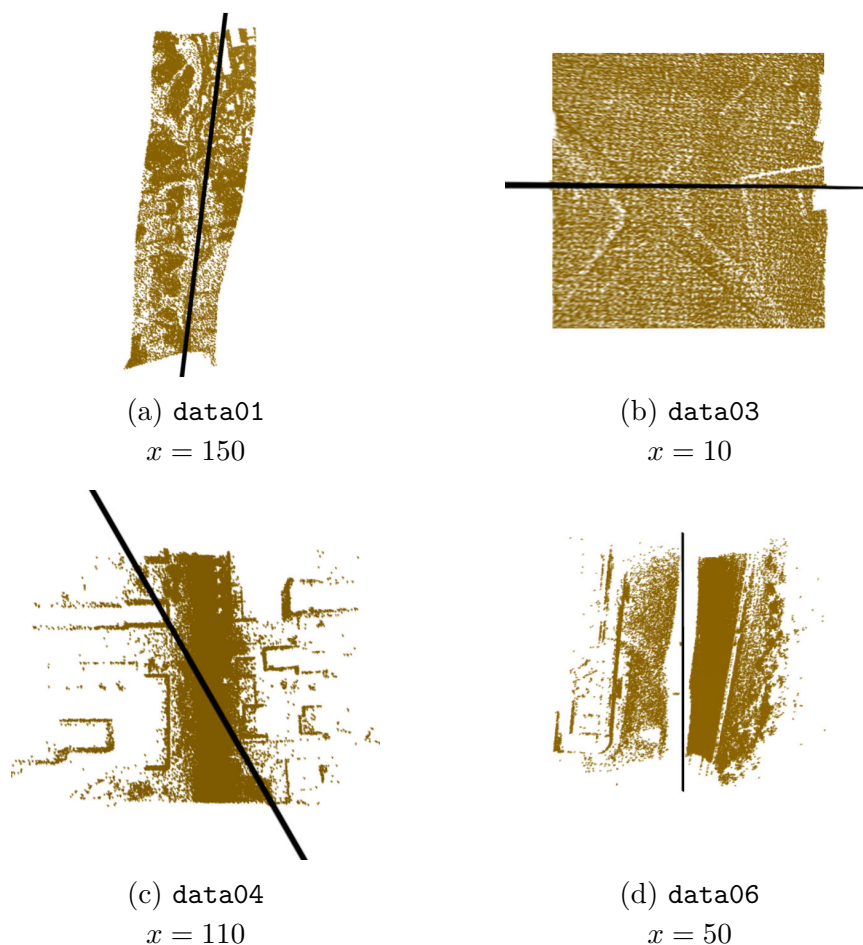


Obrázek 3.2: Náhodná redukce datasetu `data03`, kde došlo k odstranění zajímavých částí kvůli špatně zvolenému poměru.

Množství eliminovaných bodů se liší v závislosti na celkovém množství bodů jednotlivých datasetů. Například u datasetu `data03`, který je tvořen cca 560 tisíci body, je ponechán každý desátý bod. Naopak u datasetu `data04`, který tvoří více než 6,5 miliónu bodů, se hodnota x rovná 110, tedy pouze každý sto desátý bod byl ponechán. Neexistuje přesné měřítko, jaké x zvolit. V experimentech se osvědčilo zredukovat data tak, aby ve výsledku měly kolem 35 tisíc až 55 tisíc bodů, je-li někde hodnota nižší nebo naopak vyšší, výrazná změna při nalezení výsledné symetrie nenastane. Takhle velkou redukci si můžeme dovolit, protože body jsou poměrně hustě rozmístěny. V následujících sekcích kapitoly 3.2.2 na obrázcích 3.3 a 3.5 můžeme vidět, jak redukce dat pozměnila výpočet roviny symetrie. U každého obrázku je uvedena hodnota x , která byla použita a bude používána i ve všech dalších pokusech, není-li uvedeno jinak.

Detekce jedné roviny symetrie

U datasetu `data01` (obrázek 3.3a) je zřejmé, že rovina symetrie se posunula blíže ke kruhovým objezdům, což se zdá být žádoucí.



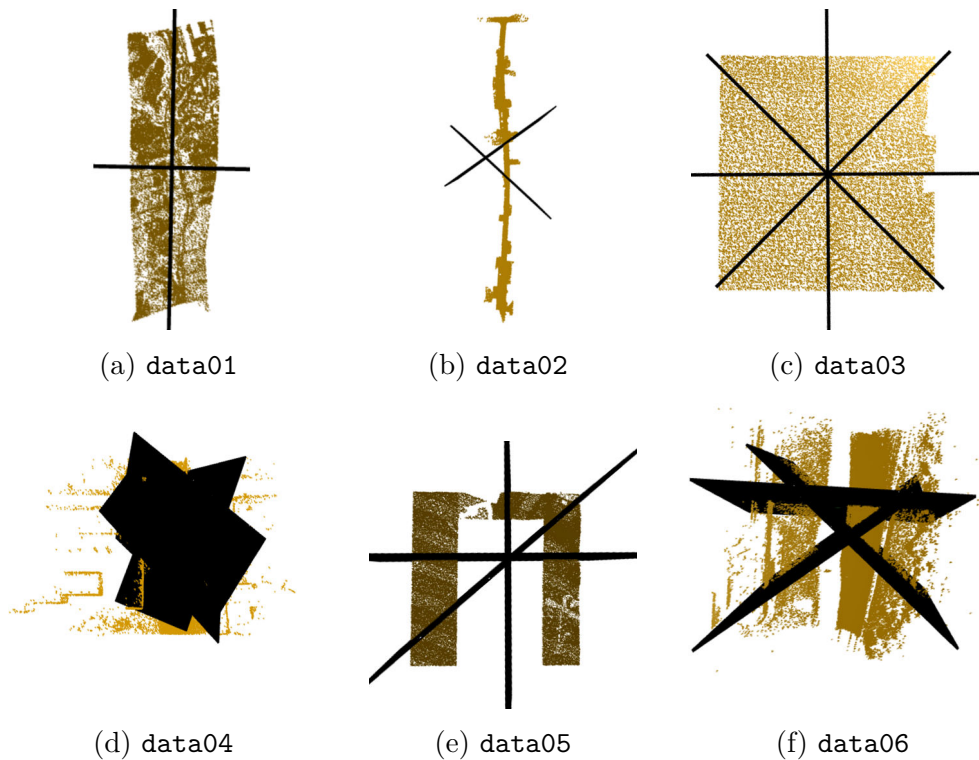
Obrázek 3.3: Detekce jedné roviny symetrie s výpočtem z dat redukováných, x udává hodnotu, jak moc byly data redukována.

Redukce ovšem nepomohla u všech datasetů. Např. u datasetu `data03`, kde se nachází křižovatka, jsme předpokládali, že zjednodušením dat by se rovina symetrie mohla posunout blíže křižovatce - v ideálním případě do jejího středu. Nestalo se tak. Stejně tak ani u dalších datasetů (`data02`, `data04`) k nalezení „lepší“ roviny symetrie nedošlo. Rovina byla nalezena pouze z jiného úhlu. Viz obrázky 3.3b, 3.3c a 3.3d.

Redukce vstupních dat měla větší přínos v samotné vizualizaci než v hledání roviny symetrie. I přes to budeme redukována data používat i v následujících experimentech, protože výpočet, který provádíme již z dat redukováných, je rychlejší.

Detekce více rovin symetrie

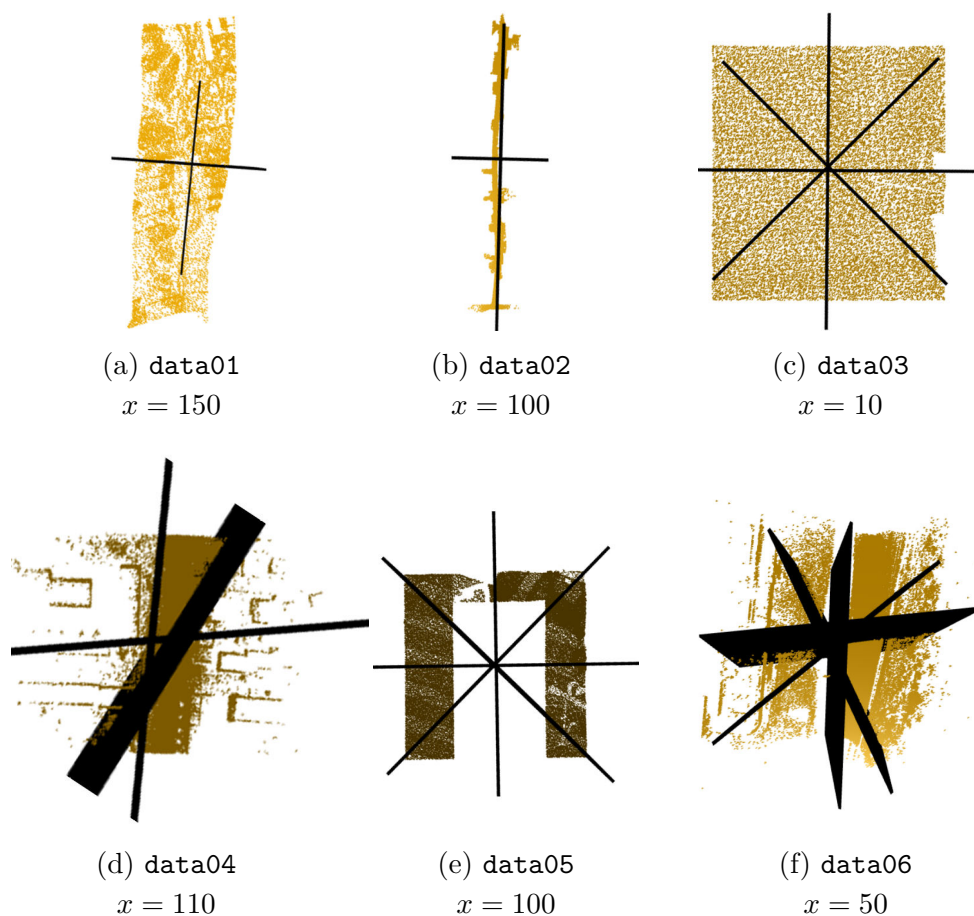
Pro detekci více rovin symetrie byla také využita data zredukováná. Nejdříve jen pro účel vizualizace a následně i pro výpočet rovin symetrie. Rovin bylo nalezeno hned několik, ovšem zkosená rovina zabraňuje pěknému náhledu. Nachází-li se zkosených rovin více, je téměř nemožné vidět body objektu, to lze vidět u datasetu data04 na obrázku 3.4d.



Obrázek 3.4: Výsledky nalezených rovin symetrie s výpočtem z dat původních.

Obrázek 3.4 demonstruje výsledky nalezení rovin symetrie s výpočtem z dat původních. Pro účel vizualizace byla použita data redukováná.

Porovnáním obrázků 3.4 a 3.5 zjistíme, že stejně jako u detekce jedné roviny symetrie, ani zde redukce nějak výrazně nepomohla k nalezení „lepší“ rovin symetrie. K jedinému, ale za to výraznému, zlepšení došlo u datasetu data02 (viz obrázek 3.5b).



Obrázek 3.5: Znázornění výsledků nalezení více rovin symetrie s výpočtem z dat redukovaných.

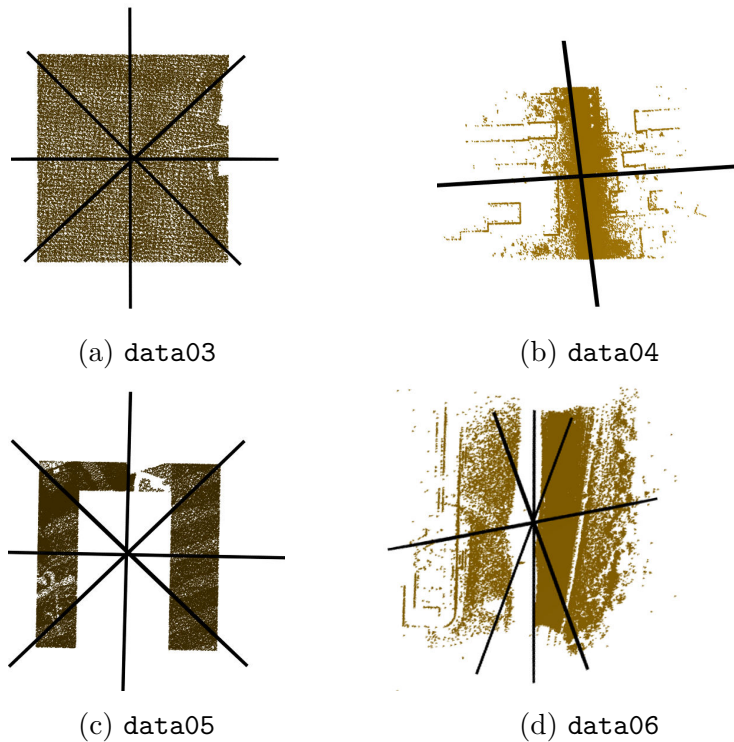
Jelikož testovací data neobsahují žádné převisy či jiné objekty, pro které bychom zkosené roviny nutně potřebovali, v následující kapitole 3.2.3 se pokusíme výškové body vynechat, a tím tyto roviny eliminovat.

3.2.3 2D

Za výškový bod lze považovat jakýkoli bod, který nemá z souřadnici nulovou. Vezmou se proto všechny body a budeme uvažovat $z_i = 0$ (kde i je index bodu z množiny bodů), čímž dosáhneme projekce původního 3D objektu do roviny XY , tedy vytvoříme 2D objekt a budeme hledat pouze jeho osy symetrie.

Dosažené výsledky jsou zobrazeny na obrázku 3.6, Metoda nepřinesla žádné velké změny. Nalezené osy symetrie jsou hledány vůči objektu jako

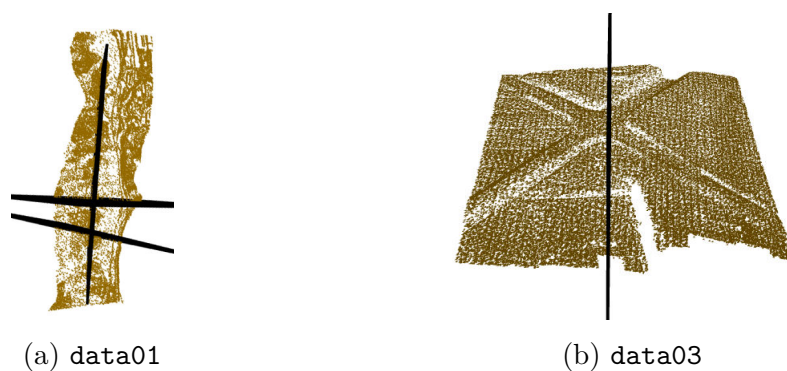
celku, tudíž u data03 (3.6a) a data05 (3.6c) jsou to osy čtverce.



Obrázek 3.6: Nalezení rovin po promítnutí 3D objektu do 2D.

3.2.4 Změna měřítka

Protože zjednodušení objektu pouze na 2D plochu nepomohlo, nabízí se opačná možnost, jíž je zvětšení hodnot na ose z .

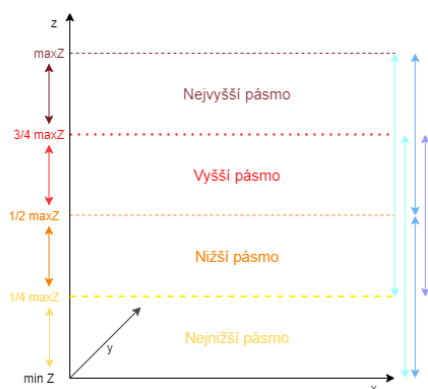


Obrázek 3.7: Vizualizace nalezení rovin při změně měřítka osy z .

Vlevo je zástupce dat, u kterých došlo ke zhoršení, a vpravo je ukázáno, že metoda našla lepší rovinu symetrie. Došlo k výraznému zlepšení u `data03`, kde jsme se přiblížili středu křížovanky (3.7b). Naopak ke zhoršení došlo při nalezení více rovin v datasetu `data01`, kde nalezené roviny jsou zcela na opačné straně, než bychom chtěli.

3.2.5 Výšková pásma

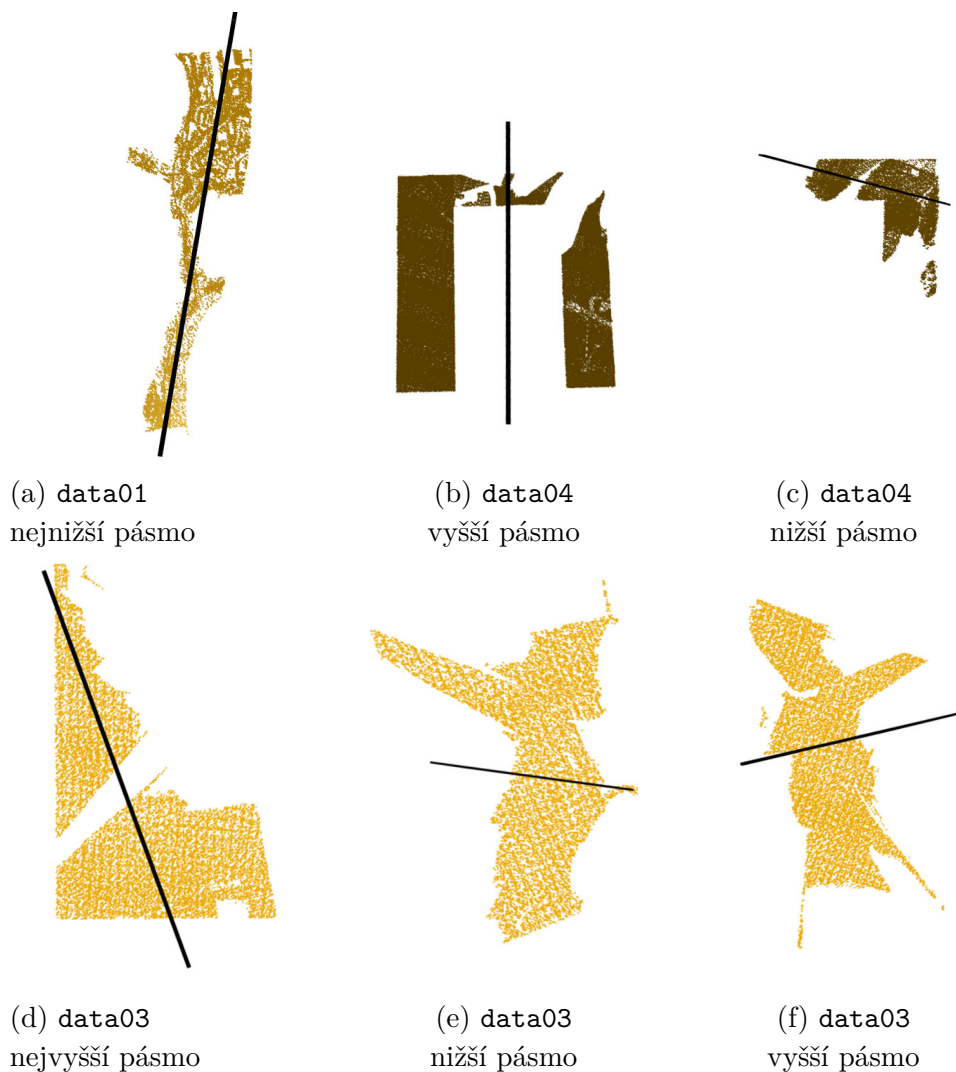
Data byla rozdělena do čtyř pásem. Pásma jsou rozdělována rovnoměrně od nejnižší po nejvyšší hodnotu souřadnice z daného datasetu - viz obrázek 3.8.



Obrázek 3.8: Ukázka rozdělení datasetu do pásem

Původní pásma jsou sice jen čtyři, ale jejich kombinací získáme samozřejmě více možností, např. vizualizaci pouze bez nejvyšších hodnot nebo naopak pouze bez nejnižších, atd. - to lze také vidět na zmiňovaném obrázku jsou znázorněny jen některé možnosti (modrými šipkami vpravo). Na obrázcích 3.9 jsou výsledky některých datasetů v různých pásmech.

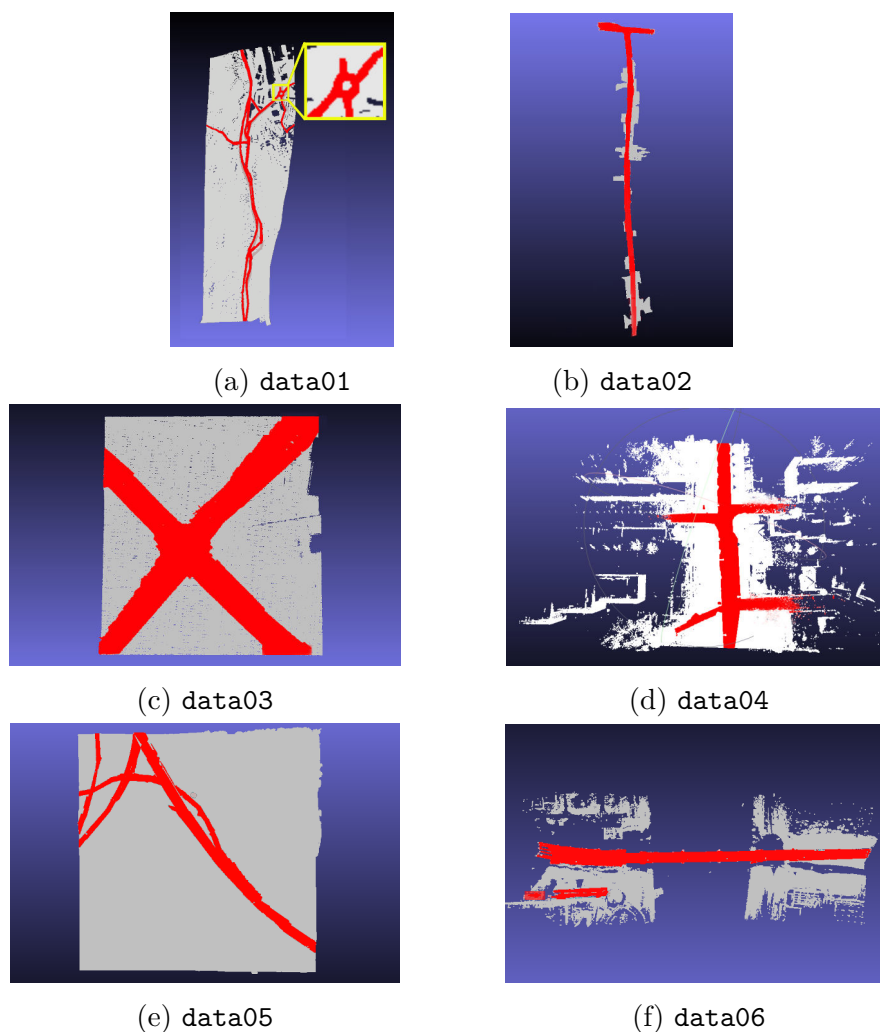
Jak lze vidět, u datasetu `data01` pomohlo zobrazení jen nejnižšího pásma, které obsahuje pro nás zajímavá data. U datasetu `data05` naopak nejnižší pásmo nebylo tak zajímavé, jako pásma nad ním (obrázky 3.9c a 3.9b). U dalších datasetů zobrazení jen nějakých pásem nepomohlo k vybrání zajímavých dat (obrázek 3.9 d-f). Pro vybrání zajímavé množiny dat nelze tedy určit obecné pravidlo, které by určilo výškové pásmo.



Obrázek 3.9: Nalezení rovin po zobrazení pouze některých výškových pásem.

3.2.6 Ruční výběr množiny bodů

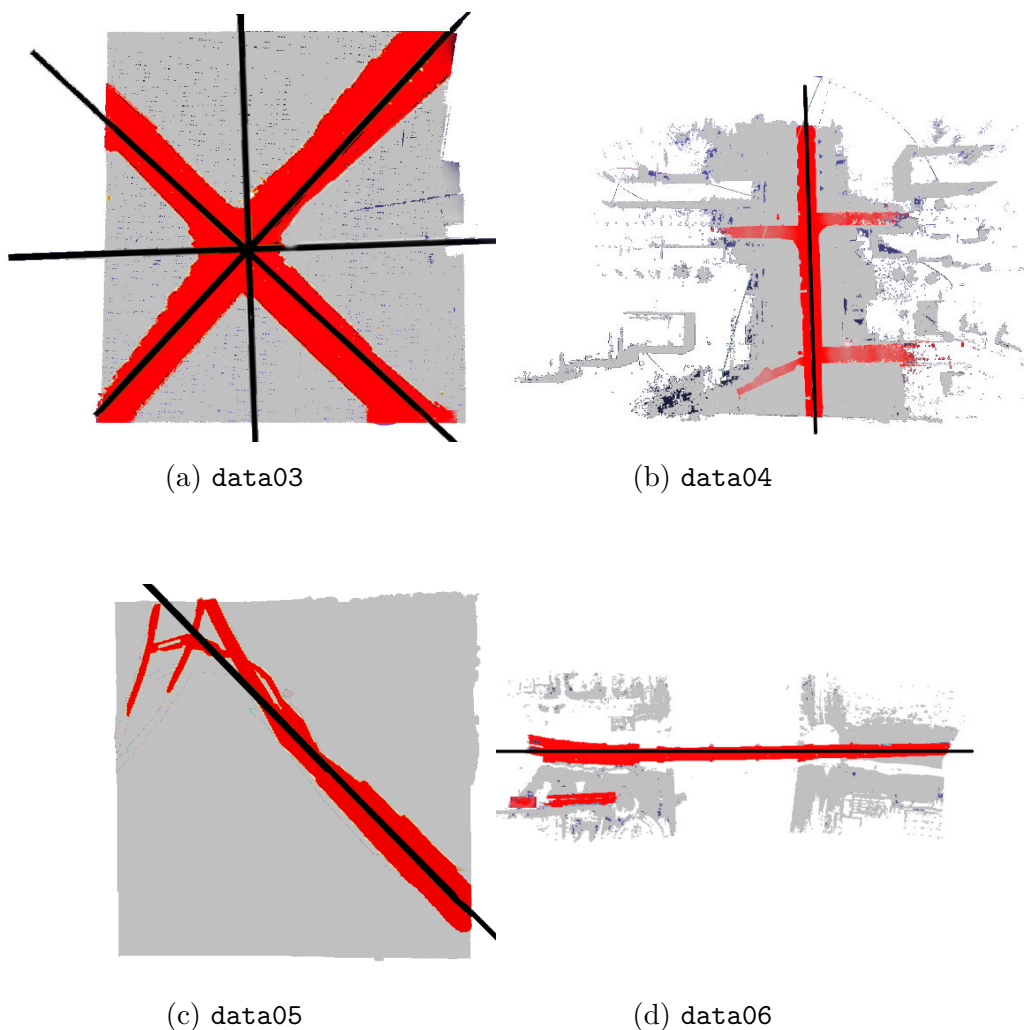
Přestože žádná z uvedených úprav vstupních dat neprokázala zlepšení nalezené roviny symetrie u všech datasetů, nelze říci, že samotná metoda pro určení roviny symetrie nefunguje na tento typ vstupních dat. Jak již bylo několikrát zmíněno, metoda spíše určuje symetrii celkového tvaru, který žádná z úprav nedokázala vhodně „rozbít“ či eliminovat nezajímavé body natolik, že by výsledná symetrie byla určována jen z pro nás zajímavých bodů. Z toho důvodu bude tato kapitola věnována experimentům jen se zajímavými body, které ručně vybereme.



Obrázek 3.10: Přehled všech datasetů s červeně zvýrazněnou množinou důležitých bodů.

Výběr byl prováděn za pomoci open source nástroje - Meshlab [6], který je volně dostupný z <https://www.meshlab.net/>. Obrázek 3.10 je přehledem všech datasetů, na nichž je červeně znázorněn výběr zajímavých bodů. Obrázek 3.11 zobrazuje nalezené roviny pouze u vybrané množiny bodů.

Jak je patrné, metoda dokáže poměrně přesně určit roviny symetrie, pokud je jí předán správný výběr množiny bodů. Před detekcí symetrie je tedy nutné předběžné zpracování mračna bodů, aby se odstranila nežádoucí data z mračna bodů a výsledná symetrie byla detekována pouze u zajímavých částí (ulice, křižovatky,... nikoliv rozsáhlá krajina a lesy - viz obrázek 3.10).

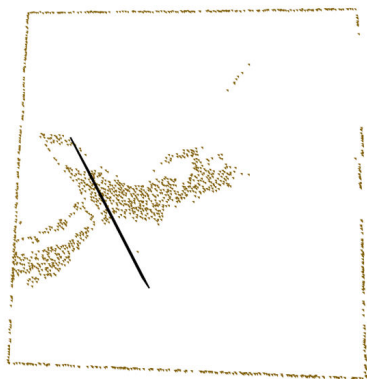


Obrázek 3.11: Nalezené roviny symetrie u vybraných množin bodů ze vstupních datasetů.

Ačkoliv v předešlých metodách byla snaha vybrat správnou množinu bodů, ne vždy metoda zafungovala a když ano, tak nikdy ne na všech testovacích datasetech. Proto se v další metodě (popsána v sekci 2.5) pokusíme předzpracovat data tak, aby byly detekovány pouze body hran ze kterých by byly důležité části zřetelné a následná detekce symetrie by nebyla tolik ovlivňována nezajímavými body.

3.2.7 Laplacián

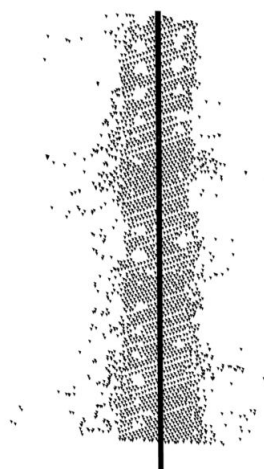
Pro svoji práci jsem zvolila a naimplementovala filtrování pomocí Laplaceova operátoru (představen v kapitole 2.5) a byl využit filtr z obrázku 2.4(a). Pokud se v jedné buňce vytvořené mřížky nacházelo více bodů, byly tyto body zprůměrovány a pro další účely bylo zacházeno s tímto průměrem. Výsledky je možné vidět na obrázcích 3.12.



(a) data03



(b) data05



(c) data04



(d) data01



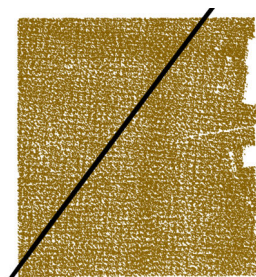
(e) data02

Obrázek 3.12: Výsledky aplikace Laplaceova operátoru.

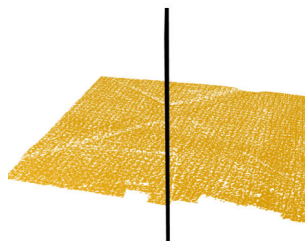
Porovnáme-li výsledky této metody s ručním výběrem zajímavých dat, uvidíme, že metoda poměrně dobře zafungovala. U datasetu data01 došlo k výběru zajímavých dat (obr. 3.12d), stejně tak u datasetů data04 a data05 (obr. 3.12c a 3.12b) byly detekovány obrysové hrany. U datasetu data03 (3.12a je vidět nalezená část křížovanky.

3.2.8 RemoveSymetricPlanes

Poslední metoda, kterou naimplementujeme, je metoda, kdy iteračně budeme odstraňovat body, které odpovídají nalezené symetrii, a výslednou rovinu symetrie budeme dále počítat z bodů zbývajících.



(a) data03 půdorys



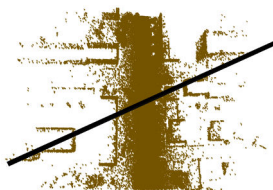
(b) data03 pohled perspective



(c) data02



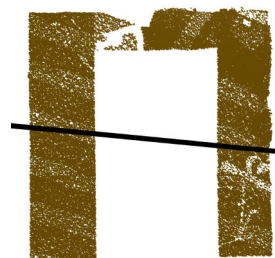
(d) data06



(e) data04



(f) data05



(g) data05

Obrázek 3.13: Výsledky metody iteračního odstraňování symetrických bodů

Jak pokusy prokázaly, tato metoda bude fungovat pouze v případě, že první nalezená rovina symetrie je dobrá (obr. 3.13 a-d). V opačném případě

metoda k ničemu nevede (obr. 3.13 e-g).

3.3 Zhodnocení výsledků

Do existujícího softwaru bylo doimplementováno několik doplňkových metod. První z nich byla generalizace vstupních dat, která přispěla hlavně k provedení rychlejšího výpočtu. Protože při detekci více rovin zakrývaly zkosené roviny samotná data, druhá metoda se snažila tyto roviny eliminovat převedením původních dat do 2D. U této metody došlo ke zjištění, že software detekuje roviny na základě celkového tvaru objektu a tudíž při detekci osy symetrie u 2D nebylo dosaženo velkých změn. Nabízela se tedy opačná možnost - zvětšit měřítko v ose z a tím dát větší váhu výškovým bodům. Z experimentů bylo zřejmé, že tato úprava má rozhodně vliv na výsledné roviny, i když v některých případech byl výsledek spíše negativní. Toto zjištění přineslo myšlenku na další možné úpravy.

Jestliže některé výškové body dokážou ovlivnit detekovanou rovinu symetrie, zkusíme všechny body rozdělit do výškových pásem a detekovat rovinu jen u některých z nich. Cílem tohoto experimentu bylo vybrat jen některá pásma, kde se zdají být zajímavé části původního objektu, a původní objekt tím „rozbít“. „Rozbít“ celkový tvar objektu se podařilo u všech datasetů, ale bohužel jen v jednom případě se podařilo zachovat pro nás zajímavá data.

Do této chvíle všechny implementované metody měly nějaký přínos, ale stále ne takový, že by došlo ke zlepšení detekce roviny symetrie u všech datasetů. Cílem bylo najít metodu, která by eliminovala nepodstatné celky objektu a zanechala jen zajímavou část, ovšem nebylo jisté, zda poté výsledná rovina bude detekována správně. Odpověď na otázku, zda má cenu hledat metodu, která by výběr zajímavé množiny bodů udělala, jsme získali dalším pokusem, který spočíval v ručním výběru zajímavých dat.

Poslední pokusy se snažily udělat výběr podmnožiny bodů, ze které by rovina symetrie byla počítána. Metoda zahrnující Laplaceův operátor našla poměrně pěknou podmnožinu bodů a tím bylo dosaženo i lepší detekce výsledné roviny symetrie.

4 Závěr

V teoretické části byla nejprve představena definice symetrie. Poté jsme se seznámili se vstupními daty a známými metodami pro nalezení a využití symetrie v geodatech.

V rámci praktické části práce bylo navrženo a naimplementováno několik možností, jak předzpracovat vstupní data. Z dosažených výsledků bylo ukázáno, že testovaná metoda pro detekci symetrie funguje i na reálných datech, ovšem za předpokladu, že vstupní množina je vhodně předzpracovaná a vstupem jsou „zajímavá“ data. Po tomto zjištění byla naimplementovaná metoda na bázi detekce hran pomocí Laplaceova operátoru. Tato metoda dosáhla překvapivých výsledků, jež nás utvrdily ve zjištění, že metodu lze na geo data využít.

Vhodným námětem pro další postup v této oblasti může být nalezení lepší metody, která by vybrala takové množiny bodů, jejichž detekovaná rovina symetrie by mohla pomoci ve výzkumu souvisejícím např. s rekonstrukcí silnic či rozpoznávání botanických objektů.

Literatura

- [1] Sustainability of Digital Formats: Planning for Library of Congress Collections, Jan 2020. Dostupné z: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000418.shtml>.
- [2] LAS 1.0/1.1/1.2 ASPRS LiDAR data translation toolset. Dostupné z: <https://liblas.org/>. (Accessed on 10/02/2020).
- [3] What is lidar data? | ArcGIS Desktop. <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>, 2019. (Accessed on 04/18/2021).
- [4] plas.io - Free Online Data Viz to Explore LiDAR Data. Dostupné z: <https://plas.io/>. (Accessed on 04/11/2021).
- [5] CHENG, L. et al. A Symmetry-Based Method for LiDAR Point Registration. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 10 2017, 11, s. 1–16. doi: 10.1109/JSTARS.2017.2752765.
- [6] CIGNONI, P. et al. MeshLab: an Open-Source Mesh Processing Tool. In SCARANO, V. – CHIARA, R. D. – ERRA, U. (Ed.) *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. doi: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136. ISBN 978-3-905673-68-5.
- [7] CLODE, S. et al. Detection and Vectorisation of Roads from Lidar Data. *Photogrammetric Engineering and Remote Sensing*. 01 2006.
- [8] COMBÈS, B. et al. Automatic symmetry plane estimation of bilateral objects in point clouds. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 06 2008. doi: 10.1109/CVPR.2008.4587605.
- [9] ČERBA, O. Generalizace, Sep 2004. Dostupné z: <http://old.gis.zcu.cz/studium/tka/Slides/generalizace.pdf>. (Accessed on 03/11/2021).
- [10] GONZALEZ, R. C. – WOODS, R. E. *Digital Image Processing, 2/e*. Pearson Education, 2008. ISBN 9788177581683.
- [11] HOLENDA, J. – RYJÁČEK, Z. – KATEDRA MATEMATIKY, Z. *Lineární algebra II: úvod do diskrétní matematiky*. Západočeská univerzita, 2000.

- [12] HRUDA, L. Hledání symetrie v geometrických modelech. Diplomová práce, Západočeská univerzita v Plzni, Plzeň, Jun 2018. Dostupné z <http://hdl.handle.net/11025/31788>.
- [13] HRUDA, L. – KOLINGEROVÁ, I. – VÁŠA, L. Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure. *The Visual Computer*. Jan 2021. doi: 10.1007/s00371-020-02034-w. Dostupné z: <https://doi.org/10.1007/s00371-020-02034-w>.
- [14] KAKARALA, R. – KALIAMOORTHY, P. – PREMACHANDRAN, V. Three-Dimensional Bilateral Symmetry Plane Estimation in the Phase Domain. *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, s. 249–256.
- [15] KERBER, J. et al. Scalable Symmetry Detection for Urban Scenes. *Computer Graphics Forum*. 02 2013, 32. doi: 10.1111/j.1467-8659.2012.03226.x.
- [16] KOLINGEROVÁ, I. Generalized Symmetries and Equivalences of Geometric Data. *Návrh GAČR projektu*. 2021.
- [17] LIPMAN, Y. et al. Symmetry factored embedding and distance. In *SIGGRAPH 2010*, 2010.
- [18] MITRA, N. J. – GUIBAS, L. – PAULY, M. Partial and Approximate Symmetry Detection for 3D Geometry. *ACM Transactions on Graphics (SIGGRAPH)*. 2006, 25, 3, s. 560–568.
- [19] PETITJEAN, M. A definition of symmetry. *Culture and Science*. 01 2007, 18, s. 99–119.
- [20] ROSEN, J. *Symmetry in Science: An Introduction to the General Theory*. Springer, 1995.
- [21] WANG, H. – HUANG, H. Group Representation of Global Intrinsic Symmetries. *Computer Graphics Forum*. 2017, 36, 7, s. 51–61.

Příloha 1

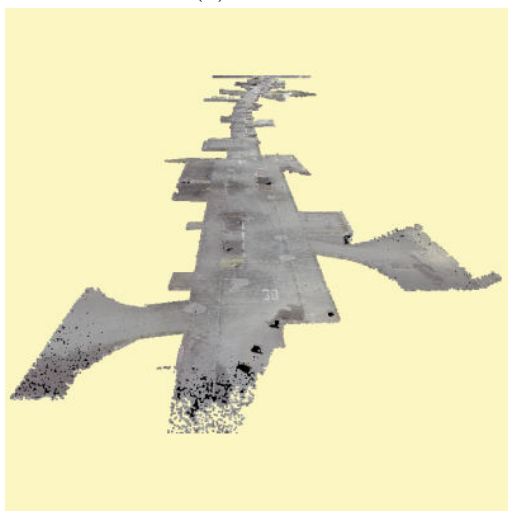
Obrázek 1: Datasetsy 01-03: perspektivní pohled (vlevo) a půdorys (vpravo)



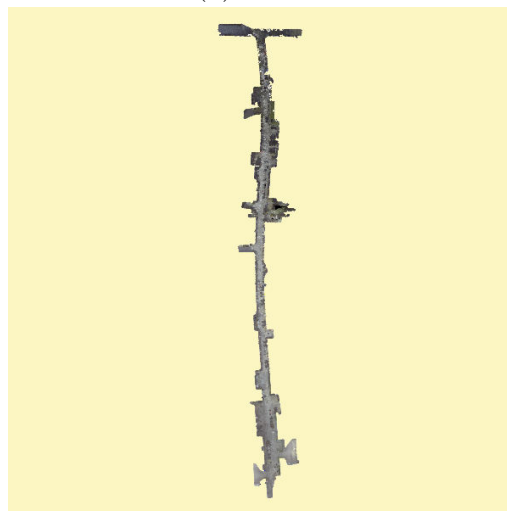
(a) data01



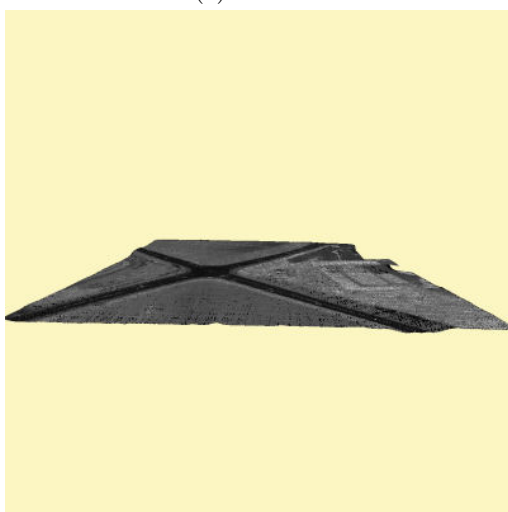
(b) data01



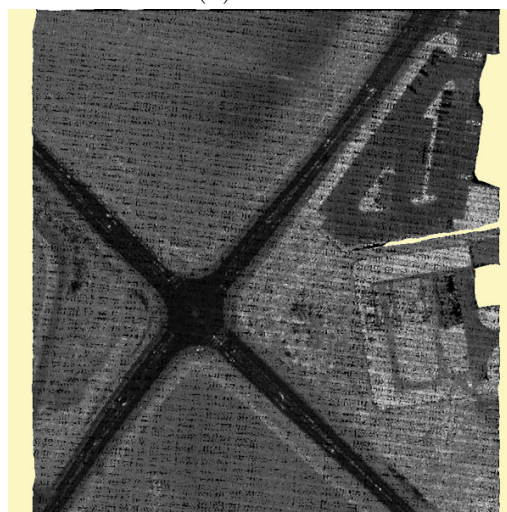
(c) data02



(d) data02

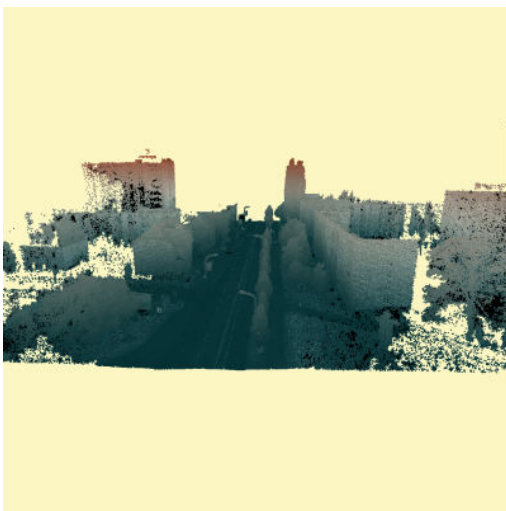


(e) data03

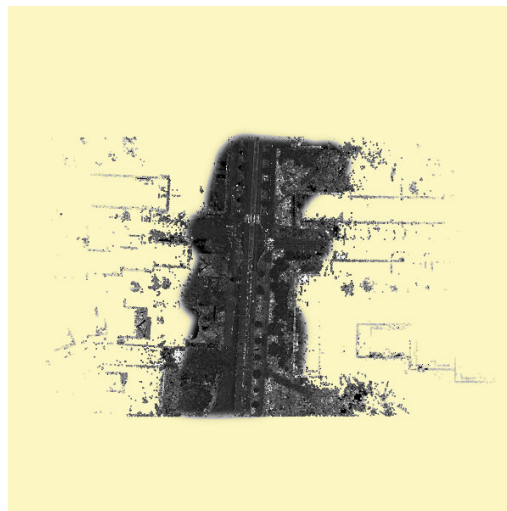


(f) data03

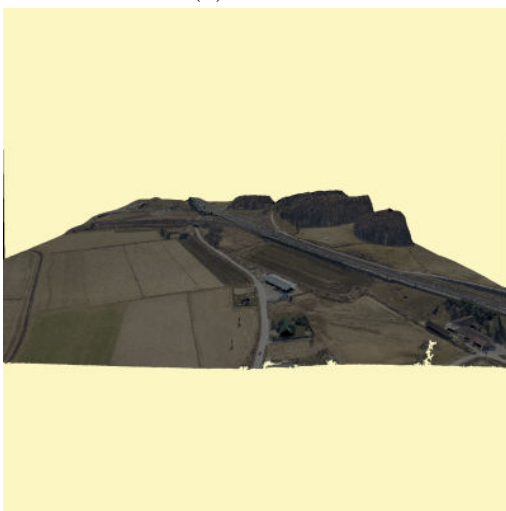
Obrázek 2: Datasetsy 04-06: perspektivní pohled (vlevo) a půdorys (vpravo)



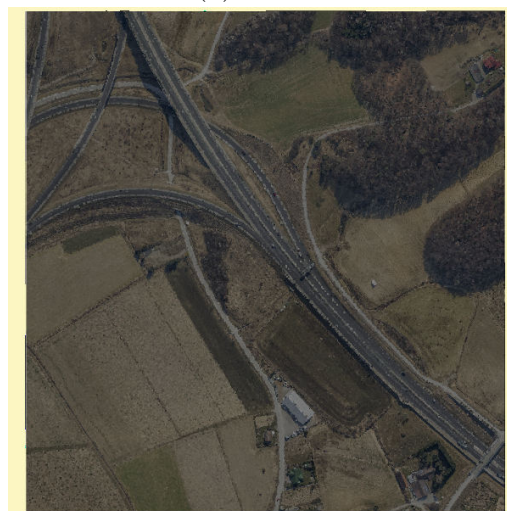
(a) data04



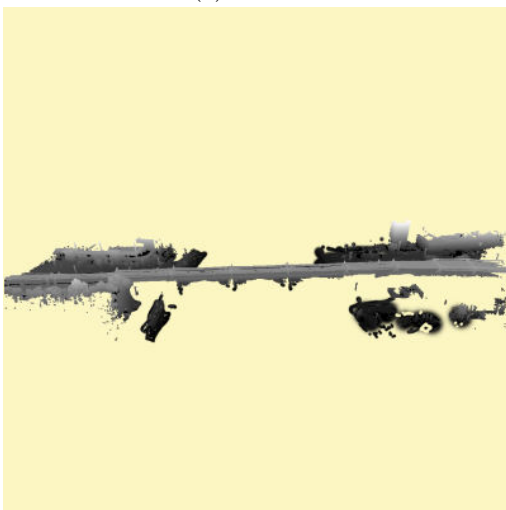
(b) data04



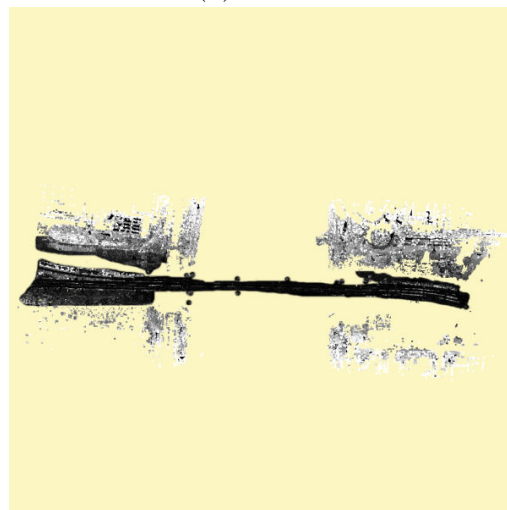
(c) data05



(d) data05



(e) data06



(f) data06

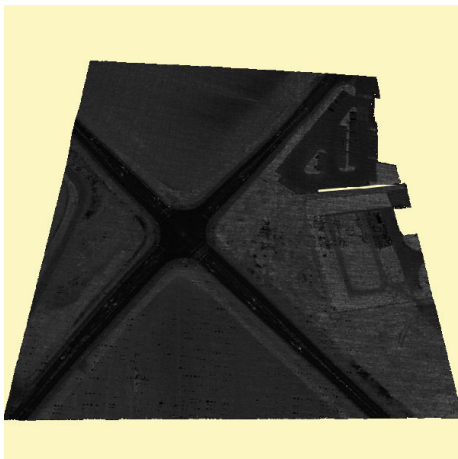
Obrázek 3: Datasetsy 01-06: perspektivní pohled používaný pro vizualizaci v experimentech



(a) data01



(b) data02



(c) data03



(d) data04



(e) data05



(f) data06