

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

Bakalářská práce

Zobecněná zrcadlová symetrie množiny bodů v rovině

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petr VONDROVIC**
Osobní číslo: **A20B0273P**
Studijní program: **B0613A140015 Informatika a výpočetní technika**
Specializace: **Informatika**
Téma práce: **Zobecněná zrcadlová symetrie množiny bodů v rovině**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s hlavními metodami hledání zrcadlové symetrie, zvláště s metodou a programovým vybavením vytvořeným na KIV.
2. Navrhněte vhodnou metodu pro nalezení osy zobecněné zrcadlové symetrie množiny bodů navzorkovaných z hranice nějakého rovinného objektu, hledanou osu aproximujte lomenou čarou. Předpokládejte, že osa nemá žádné sebeprůsečíky.
3. Navrženou metodu implementujte a vyzkoušejte na sadě netriviálních případů.
4. Dosažené výsledky zhodnoťte.

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

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce

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

Datum zadání bakalářské práce: **3. října 2022**
Termín odevzdání bakalářské práce: **4. května 2023**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

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

V Plzni dne 25. října 2022

Prohlášení

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

V Plzni dne 3. května 2023

Petr Vondrovic

Poděkování

Děkuji Prof. Dr. Ing. Kolingerové za odborné vedení, její čas, možnost mnoha osobních konzultací a všeobecnou pomoc při zpracování bakalářské práce. Taky bych chtěl poděkovat své rodině za celkovou podporu při mém studiu.

Abstract

Symmetry is a common property of geometric models. Its most common type is mirror symmetry, i.e. the division of an object, a shape, into halves that reflect each other. This paper deals with generalized mirror symmetry in 2D. The generalization in this work is the mirror symmetry of a set of points with the form of connected segments formed by parallel lines. In this work, the partition of the input point set into individual segments is solved, over which the symmetry axis is then found. The found axes are then related to each other and displayed together with the original data in an undivided form. The evaluation of the results was based on experiments performed with different input point sets in different forms, point densities and data inconsistencies. The achieved results are at a very good level even with a reasonable level of distortion of the investigated data.

Abstrakt

Symetrie je častou vlastností geometrických modelů. Jejím nejobvyklejším druhem je zrcadlová symetrie, tj. rozdělení objektu, tvaru na poloviny, které jsou vzájemně zrcadlovým odrazem. Tato práce se zabývá zobecněnou zrcadlovou symetrií v 2D. Zobecnění v této práci je zrcadlová symetrie množiny bodů s podobou navazujících úseků tvořených rovnoběžkami. V práci je řešeno rozdělení vstupní množiny bodů na jednotlivé úseky, nad nimiž je poté nalezena osa symetrie. Nalezené osy jsou poté navázány na sebe a společně s původními daty v nerozdělené podobě zobrazeny. Vyhodnocení výsledků proběhlo na základě experimentů prováděných s různými vstupními množinami bodů v různých formách, hustotou bodů a porušením dat. Dosažené výsledky jsou na velmi dobré úrovni i při mírné míře deformace zkoumaných dat.

Obsah

1	Úvod	8
2	Analytická část	9
2.1	Symetrie	9
2.1.1	Symetrické objekty	9
2.1.2	Transformace	9
2.1.3	Transformace symetrie	10
2.1.4	Druhy symetrie	10
2.2	Alternativy metody detekce osové symetrie	12
2.3	Použitá metoda detekce osové symetrie	14
3	Navržené řešení	17
3.1	Rozdělení bodů	18
3.2	Detekce os symetrie	20
3.2.1	Užití programu detekce symetrie	20
3.2.2	Přímé spočtení osy symetrie	21
3.3	Spojení os	21
3.4	Generování množin bodů	23
3.5	Technické řešení	23
4	Experimenty a výsledky	25
4.1	Limity programu detekce symetrie	25
4.1.1	Limity délky úseček	26
4.1.2	Experimenty s lomenými čarami	29
4.1.3	Pokusy o úpravu dat	34
4.2	Výsledky metody rozdělení bodů	34
4.2.1	Vliv formy dat	35
4.2.2	Vliv počtu bodů	37
4.2.3	Nerovnoměrné rozdělení bodů	41
4.2.4	Odolnost proti šumu	42
4.3	Výsledky přímého spočtení os symetrie	44
5	Závěr	50
	Literatura	51
6	Přílohy	57

1 Úvod

Symetrie je základním pojmem v přírodních vědách i mimo ně. Každé odvětví ji ale zkoumá na základě odlišných dat. V rámci počítačové grafiky se obvykle zpracovávají množiny bodů nebo trojúhelníkové sítě a hledá se symetrie konkrétního geometrického objektu.

Objekt je symetrický, existuje-li určitá operace, po jejímž provedení je výsledný objekt totožný sám se sebou v určitém smyslu. Zrcadlovou symetrii objekt splňuje, pokud existuje alespoň jedna přímka, respektivě rovina, která dělí jev či objekt na dvě poloviny tak, že jedna polovina je zrcadlovým obrazem druhé poloviny.

Tato práce zabývá hledáním zobecněné zrcadlové symetrie objektu reprezentovaného množinou bodů v rovině. Prezentována množina bodů má přitom podobu několika úseků rovnoběžek spojených do podoby lomeného objektu nebo stromu. Výsledná osa odpovídá danému objektu. Práce zkoumala možnosti využití metody detekce symetrie vyvinuté na KIV v rámci společného výzkumu s KMA pro nalezení takové osy.

V teoretické části se nachází potřebná teorie symetrie, stručný popis alternativních metod a popis použité metody detekce symetrie. V další části práce se seznámíme s detailním popisem navrženého řešení a popisem programu v němž bylo řešení naimplementováno. Kapitola experimentů se zabývá experimenty prováděnými během vývoje a experimenty testující výsledné řešení. Závěr shrnuje obsah práce a dosažené výsledky.

2 Analytická část

2.1 Symetrie

V této sekci je probrána teorie symetrie. Většina informací je převzata z [20], [18] a [21]. Teorie vychází z definice tvrdící, že objekt je symetrický, existuje-li transformace, při níž se daný objekt nemění.

2.1.1 Symetrické objekty

V této sekci je nadefinován objekt. V našem případě se jedná o množinu bodů. Aby objekt mohl být symetrický, musí být identický s jiným. Takový vztah objektů se nazývá **ekvivalence** a značí se \equiv . Ekvivalence nastává, jsou-li splněny následující tři podmínky:

- *Reflexivita* - Objekt je identický sám se sebou.

$$u \equiv u$$

- *Symetrie* - Je-li objekt totožný s druhým, pak je druhý objekt totožný s prvním

$$u \equiv v \Leftrightarrow v \equiv u$$

- *Tranzitivita* - Pokud je první objekt totožný s druhým, který je totožný s třetím, pak je i první objekt totožný se třetím

$$u \equiv v, v \equiv w \Rightarrow u \equiv w$$

Máme-li množinu objektů, pak pojmem **třídy ekvivalence** označujeme podmnožiny, kde každé dva prvky jsou navzájem ekvivalentní.

2.1.2 Transformace

Za transformaci se dá považovat za speciální funkce, například otočení, osová souměrnost, zvětšování nebo zmenšování. Transformaci T zapisujeme:

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

kde v je objekt u po transformaci.

Pojmem *stavový prostor* se označuje množina všech možných stavů tzv. systému, kterým jsou v našem případě body funkcí. Transformace se dá také popsat jako mapování stavového prostoru do sebe. Jinými slovy, obraz objektu je rovněž objektem.

Skládá-li se výchozí množina E z objektů s vlastnostmi nadefinovanými v 2.1.1, jsou všechny transformace bijektivní. To znamená, že všechny prvky množiny E mají vzájemně různé obrazy a každý obraz má právě jeden vzor.

Složení transformací

Máme dvě transformace T_1 a T_2 . **Složená transformace** vznikne provedením transformace T_1 následovanou T_2 . Obvyklé značení je:

$$u \xrightarrow{T_1} v \xrightarrow{T_2} w \text{ nebo } w = T(v) = S(T(u)) = (S \circ T)(u)$$

U složení transformací, které jsou bijektivní, lze definovat inverzní transformaci označovanou T^{-1} , pro kterou platí:

$$\begin{aligned} u &\xrightarrow{T} v \text{ nebo } v = T(u), \\ v &\xrightarrow{T^{-1}} u \text{ nebo } u = T^{-1}(v), \\ T^{-1}T &= TT^{-1} = Iu \end{aligned}$$

2.1.3 Transformace symetrie

Transformace, která nezmění třídy ekvivalence, se nazývá **transformace symetrie**. Necht S je transformace symetrie, pak pro všechny objekty v platí:

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

2.1.4 Druhy symetrie

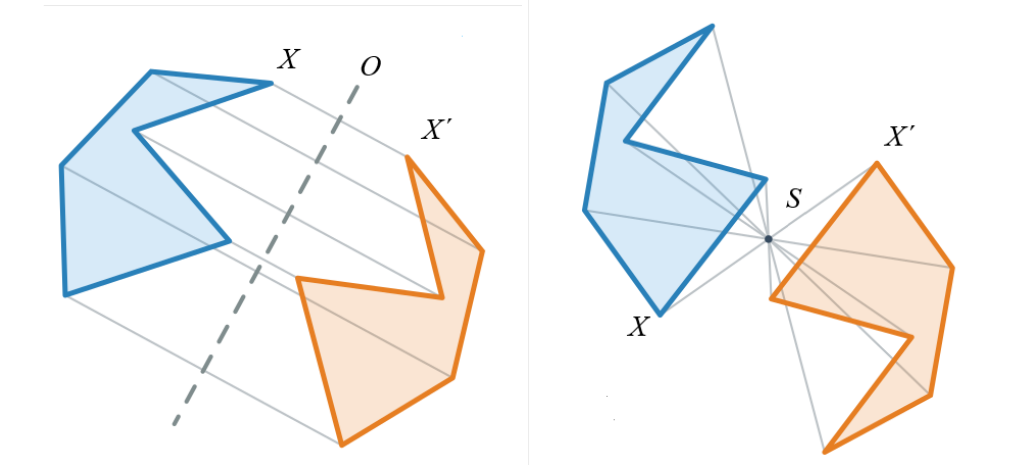
Symetrie lze dle typu transformace rozdělit na *osové*, *rotační* a *středové*. Všechny tyto transformace patří mezi shodná zobrazení. Společnou vlastností, která shodná zobrazení definuje, je zachování vzdálenosti v prostoru a rovině.

Osová symetrie

U osové souměrnosti bereme v potaz osu symetrie o . Body ležící na ose o se zobrazují samy na sebe a každý bod X ležící mimo osu se zobrazuje na obraz X' . Bod X a jeho obraz X' mají od osy stejnou vzdálenost a jejich spojnice je kolmá na osu.

Středová symetrie

Středová symetrie neboli souměrnost má bod zvaný střed souměrnosti S . Jedná se o zobrazení, které zobrazuje střed S sám na sebe a všechny ostatní body X různé od S se zobrazují na body X' . Ty leží na opačné polopřímce k SX a ve stejné vzdálenosti ($|SX| = |SX'|$).



Obrázek 2.1: Osová symetrie [1]

Obrázek 2.2: Středová symetrie [2]

Rotační symetrie

Rotační symetrie spočívá v transformaci okolo bodu, kde se protínají osy symetrie. Logicky jsou potřeba nejméně dvě osy. Transformace je aplikovaná na všechny body a sleduje se, zda i jejich obrazy náležejí stejnému objektu.

Jinými slovy, při otočení o určité úhly vypadá výchozí objekt stejně. Počet takových úhlů se označuje jako stupeň rotační symetrie.

Příkladem obrazců, nad nimiž lze rotační symetrii provést, jsou například regulární polygony o n stranách jako je čtverec, trojúhelník, pětiúhelník a další. Platí přitom, že regulární polygony mají stupeň rotační symetrie n . Dalším příkladem je kruh, který má stupňů rotační symetrie nekonečně mnoho.

2.2 Alternativy metody detekce osové symetrie

Pro detekci osové symetrie byla pro své vlastnosti i z praktických důvodů použita metoda Ing. Lukáše Hrudy [13], která je vysvětlena v sekci 2.3. Pro úplnost se zde nachází stručný popis několika dalších principů a metod používaných pro detekci symetrie. Vybrané zmíněné informace byly vybrány z [9], [13] a [7].

Jedním z algoritmů vyvinutých přímo pro detekci zrcadlové symetrie ve 2D obrazech je algoritmus autorů Michaelson a kol. [10]. Jejich metoda je založena na transformačním algoritmu invariantních charakteristik měřítka. Invariantní charakteristika se označuje i jako soběpodobnost či škálová invariance a znamená, že libovolná část objektu má podobný tvar jako celý daný objekt. Část objektu má logicky jiné rozlišení (zvětšené nebo zmenšené).

Dalším principem, který několik metod používá pro detekci symetrie ve 2D digitálním obrázku, jsou řetězové kódy. Jde o bezztrátovou kompresní techniku založenou na sledování obrysů objektu. Souřadnice objektu jsou reprezentovány jako řetězce čísel, kde každé číslo představuje určitý směr. Například autoři [15] používají k nalezení lokální symetrie obrysů izolovaných objektů Freemanův řetězový kód osmi směrů [11]. Autoři [16] užívají k nalezení lokální i globální symetrie řetězový kód směrnice (Slope Chain Code (SCC)) [4], protože je invariantní k posunu, rotaci a změně měřítka. SSC vytváří řetězec použitím přímkových segmentů stejné délky, jejichž koncové body se vždy dotýkají, a výpočtem změn sklonu mezi sousedními přínými segmenty.

Existují metody využívající rozšířený Gaussovský obraz (extended Gaussian image - EGI). Gaussovský obraz je mapování všech normál objektu na kouli s jednotným poloměrem. Všechny konce vektorů jsou ve středu koule, zatímco počátky leží na povrchu. Rozšířený Gaussovský obraz (EGI) navíc zahrnuje přidruženou plochu, kde každý bod na kouli má hmotnost úměrnou ploše. Jedna z metod využívající EGI pochází od Sun a Sherraha [6]. Metoda analyzuje orientační histogram normalizovaných jednotkových vektorů, který detekuje pouze roviny, které procházejí počátkem. K tomu metoda zkoumá mapy EGI kolem hlavních os setrvačnosti. Tato technika není robustní proti zašuměným datům.

Mezi metodami detekce symetrie se často vyskytuje použití algoritmu iterací nejbližšího bodu (Iterative closest point - ICP). Tento algoritmus je

používán k minimalizaci rozdílu mezi dvěma množinami bodů. Metoda začíná s odhadem vzájemné polohy modelů a následně tuto polohu opakovaně upřesňuje hledáním korespondujících bodů a minimalizací chyby, dokud tato chyba není dostatečně malá. ICP je třeba používána metodou pro detekci obličejů [22]. Díky použití ICP je metoda necitlivá na asymetrická data, ale zároveň ICP přináší větší výpočetní náklady.

Některé techniky určují rovinu symetrie aplikací PCA metody. PCA (Principal component analysis) je technika redukce dimenzionality dat. Tohoto cíle je dosaženo lineární transformací dat do nového souřadnicového systému, v němž lze většinu odchylek v datech popsat s menším počtem dimenzí než u původních dat. Jedná se o algoritmus učení bez učitele, používaný pro analýzu velkých souborových dat, obsahující vysoký počet dimenzí/prvků na pozorování.

Mezi techniky používající PCA patří například metoda Colbyho a Storckmana [8]. Metoda je primárně určena pro hledání roviny symetrie lidského obličejů. Nevýhodou této techniky je tendence produkovat nespolehlivé výsledky při asymetrických vstupních datech. Další ukázka metody používající PCA k odhadu roviny symetrie pochází od Di Angela a Di Stefana [3]. Tato metoda je náročná na výpočet, ale její výhoda spočívá v odolnosti proti asymetrickým datům.

Poměrně běžným přístupem k detekci symetrie je vyhledávání módů v transformačním prostoru. Je tím myšleno vytvoření řady kandidátských transformací porovnáním různých bodů nebo částí vstupního objektu a následným nalezením těch transformací, které se v transformačním prostoru vyskytují nejčastěji [14]. Takové metody mohou pracovat s transformací rotace s odrazem nebo bez odrazu, posunutím a jednotným měřítkem (stejná změna měřítka celého objektu). Například Mitra a kol. [17] používá shlukování k detekci lokální symetrie. Autoři ovšem připouštějí, že výsledky závisí na použitém poloměru pro odhad křivosti, užívaný metodou pro porovnávání bodů a filtrování kandidátů. Tento přístup byl dále upraven Caillerem a kol. [5] k hledání globální roviny symetrie. Metoda nahrazuje krok shlukování Houghovou transformací pro dosažení lepších výsledků a zlepšení výpočetní efektivity. Houghova transformace je technika shlukování, kdy vzorky dat "hlasují" pro nejrepresentativnější hodnoty vlastností v kvantovém prostoru vlastností. Klasická Houghova transformace se zabývá indentifikací čar v obraze, ale dnes je rozšířená na identifikaci pozic libovolných objektů [19].

2.3 Použitá metoda detekce osové symetrie

Osa, respektivě rovina symetrie, je detekována pomocí metody Ing. Lukáše Hrudý [13]. Technika je založena na maximalizaci diferencované míry symetrie uvádějící míru podobnosti objektu přes danou rovinu. To, že je míra symetrie diferencovaná, znamená, že funkce míry symetrie je spojitá, nemá ostré hrany a v žádném směru neroste nekonečně rychle.

Výhodou metody je její rychlost a robustnost. Funguje na objektech, které jsou symetrické jenom přibližně, a i na objektech s chybějícími částmi. Navíc je metoda odolná vůči šumu. Jednou z největších předností je to, že funguje i na objektech, které jsou definované pouze množinou bodů. Díky tomu neklade metoda prakticky žádná omezení na vstupní data.

Metoda hledá rovinu symetrie pro 3D objekty. V našem případě jsou 2D data upravena na 3D tak, že se přidá třetí souřadnice, která se rovná 0. Z nalezené roviny $ax + by + cz + d = 0$ díky tomu vypadne třetí rozměr a získáme tvar přímky $ax + by + d = 0$. Jedná se tedy o projekci této roviny do roviny xy .

Technika zhruba následuje takzvané RANSAC schéma. To znamená, že klíčovou myšlenkou je generování velkého množství kandidátů řešení. Kandidáti jsou ohodnoceni a je vybrán ten nejlepší možný. Použité řešení tento princip upravuje tím, že není třeba hledat nejlepší řešení, ale pouze jedno ve velkém konvergenčním regionu.

Stručný popis metody je následující. Je vytvořeno několik kandidátských rovin symetrie, z nichž je vybrán pouze malý počet těch, které mají největší potenciál být v oblasti konvergence řešení. Následně je zahájena optimalizace z vybraných rovin a ve výsledku je zvolena ta s největší konečnou mírou symetrie. Ostatní roviny, které po optimalizaci konvergovaly k řešení, lze použít jako sekundární roviny u objektů, které mají více významných symetrií. Podrobnější popis metody je vylíčen dále v textu.

Před samotnou detekcí symetrie je vstupní množina dat upravena tak, aby její těžiště leželo v počátku. Na konci je na výslednou rovinu/rovinu symetrie aplikovaná inverzní translace.

Těžiště množiny bodů je upraveno, protože k nalezení lokálního maxima míry symetrie se používá kavazinevtonovská optimalizační metoda L-BFGS. Tato metoda potřebuje, aby množina bodů byla blízko počátku, protože ve velké vzdálenosti od počátku i malá změna směru normálového vektoru roviny může negativně ovlivnit konvergenci metody. Metoda L-BFGS používá gradient míry symetrie, přičemž využívá jeho diferencovatelnosti, a ve většině případů konverguje s dostatečnou přesností pouze v několika iteracích.

Míra symetrie

Metoda maximalizuje míru symetrie účelovou funkcí vyhodnocující symetrii objektu vzhledem k dané rovině. Daná funkce je definována následovně:

$$s_x(\mathbf{p}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} \varphi(\|\mathbf{r}(\mathbf{p}, \mathbf{x}_i) - \mathbf{x}_j\|)$$

w_{ij} jsou váhy dvojice bodů, ukazující důležitost páru bodů. Váhy se ve výchozím nastavení neužívají, a mají tedy hodnotu 1. Jsou-li k dispozici další informace, je možné je použít a výrazně tím zvýšit flexibilitu metody. Funkce $\varphi(l)$ je radiální funkce, o níž platí, že $\varphi(0) = 1$. Se zvětšující se hodnotou l funkce klesá a blíží se k hodnotě 0. Vektor $p = [a, b, c, d]^T$ představuje libovolnou rovinu definovanou jako $ax + by + cz + d = 0$. Množina bodů $x_1 \dots x_n$ patří vstupnímu objektu.

$\mathbf{r}(\mathbf{p}, \mathbf{x}_i) \in E^3$ je vektorová funkce, která zrcadlí bod x přes rovinu \mathbf{p} . V jejím vztahu 2.1 je \mathbf{n}_p normálový vektor roviny \mathbf{p} . Složky funkce $\mathbf{r}(\mathbf{p}, \mathbf{x}_i)$ jsou spojené a diferencované s výjimkou neplatné roviny $\mathbf{p} = [0, 0, 0, d]$.

$$\mathbf{r}(\mathbf{p}, \mathbf{x}_i) = \mathbf{x} - 2 \frac{\mathbf{n}_p^T \mathbf{x} + d}{\mathbf{n}_p^T \mathbf{n}_p} \mathbf{n}_p \quad (2.1)$$

Podrobný popis metody

Při výběru kandidátů rovin je nejprve zjednodušena vstupní množina bodů X , která po úpravě obsahuje pouze pár desítek bodů. Nově vytvořená množina je v textu značena X_{cand} .

Ke zjednodušení je používán poměrně jednoduchý a rychlý algoritmus. Ten pro sadu vstupních bodů vytvoří 3D mřížku, kde z každé obsazené buňky pochází jeden bod ze zjednodušené bodové sady zprůměrováním všech bodů obsažených v buňce. Velikost buňky je definována jako $\frac{lavrg}{k} \times \frac{lavrg}{k} \times \frac{lavrg}{k}$, kde $lavrg$ je hodnota průměrné vzdálenosti bodů v sadě od jejich těžiště a k je manipulační hodnota, která je postupně zvětšována, dokud počet bodů přibližně neodpovídá požadovanému počtu bodů m .

Při vytváření množiny X_{cand} je použit algoritmus zjednodušování s hodnotou $m = 100$. Nad zjednodušenou množinou bodů X_{cand} jsou vytvořené kandidátské roviny tím, že vezmeme každou dvojici vzájemně různých bodů a vytvoříme rovinu symetrie těchto dvou bodů.

Vznikne pár tisíc kandidátských rovin, z nichž jsou mnohé velmi podobné, a není tedy nutné hodnotit všechny. Počet kandidátských rovin je

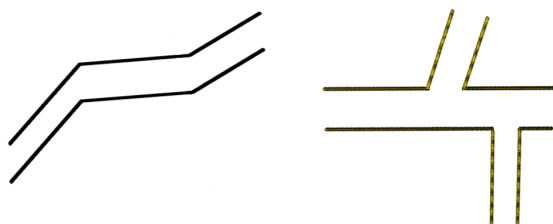
zredukován již při jejich vytváření, kdy se nová rovina nepřidá do výsledných kandidátů na základě její blízkosti k předchozí vytvořené.

Po vytvoření kandidátských rovin se nad množinou X znovu provede výše popsané zjednodušení, tentokrát s hodnotou $m = 1000$. Pro vytvořenou množinu bodů je poté vypočtena míra symetrie se všemi kandidátskými rovinami. Vybere se S kandidátů s největší mírou symetrie a zahájí se jejich optimalizace. Díky prořezání kandidátů je nepravděpodobné, že by po optimalizaci konvergovali všichni kandidáti do jedné roviny. Ve výsledku se získá S lokálních maxim míry symetrie, z nichž je vybráno to s největší mírou symetrie a je prohlášeno za výslednou rovinu symetrie. Defaultně je používána hodnota $S = 5$.

Díky tomu, že daný přístup poskytuje S lokálních maxim míry symetrie, není problém metodu upravit pro nalezení více rovin symetrie vstupního objektu. Jelikož je pravděpodobné, že dvě nebo více rovin budou velmi podobné, provede se následující. Vytvořené roviny S reprezentující lokální maxima se dle jejich míry symetrie seřadí v sestupném pořadí a iterují se od první po poslední. První rovina je vždy akceptována a jakákoliv další je vybrána pouze v případě, že její vzdálenost od další nejbližší akceptované roviny je větší než 0,25. Jelikož lze tímto způsobem najít maximálně S symetrií, je při použití detekce více rovin hodnota S zvýšena z 5 na 20 za cenu větší časové náročnosti.

3 Navržené řešení

Řešeným problémem je hledání zobecněné osy symetrie v rovině, kdy osa může mít podobu lomené čáry nebo větveného stromu. Jde tedy nalezení osy 2D “lomeného objektu“. Takový objekt si lze představit jako řeku nebo silnici. Příklad objektů je vidět na obrázku 3.1. Objekt je na vstupu očekáván ve formě množiny bodů.

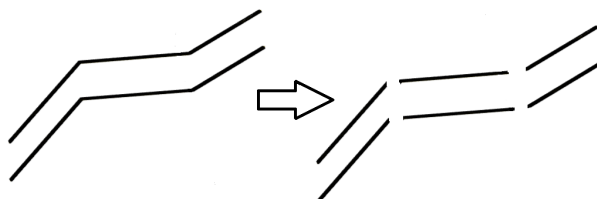


Obrázek 3.1: Ukázka vstupních objektů

Vstupní data jsou očekávána v souboru s příponou `.pc`. Na prvním řádku je celkový počet bodů v souboru. Na každém dalším řádku jsou souřadnice bodu, skládající se ze tří čísel, kdy třetí číslice je vždy nula. Celá čísla jsou od desetinných oddělená tečkou.

Po domluvě s vedoucí práce je výstup ve formě textového souboru, v němž je na prvním řádku počet os, z nichž se výsledná osa skládá. Následují dvojice bodů značící začátky a konce os. Kromě získání textového souboru je možnost si výstup graficky zobrazit.

Navržené řešení předpokládá, že body byly navzorkované z 2D objektu složeného z úseků symetrických podle úsečkové osy, viz příklad na obrázku 3.2. Data lze tímto způsobem navzorkovat algoritmem popsáním v sekci 3.1, jsou-li splněny podmínky popsané v dané sekci.



Obrázek 3.2: Rozdělení lomené čáry

Jednotlivé kroky celého řešení jsou následující.

1. Rozdělení bodů výchozího objektu na jednotlivé úseky.
2. Nalezení os symetrie jednotlivých úseků.
3. Spojení os symetrie, aby výsledná osa odpovídala tvaru výchozího objektu.

Po popsání kroků řešení se v textu této kapitoly navíc nachází popis generování metody množin bodů, kterou lze použít pro získání dat využitelných v navrženém algoritmu. Na konci sekce je popsána implementace.

3.1 Rozdělení bodů

Princip metody je provádění lineární regrese nad segmenty vstupní množiny bodů a na základě výsledků jsou segmenty spojovány. Lineární regrese představuje aproximaci (přiblížení, odhad) daných hodnot přímkou pomocí metody nejmenších čtverců. Při vyjádření přímky rovnicí $y = b_1 + b_2x$ hledáme optimální koeficienty b_1 a b_2 . V textu je koeficient b_2 označen jako sklon a b_1 jako posun.

V algoritmu se pracuje s několika konstantami, jejichž hodnoty byly zvoleny na základě experimentů s různými hodnotami při vývoji. Konstanty související s lineární regresí zahrnují hodnoty sklonu $b_2s = 0.2$, $b_2l = 1$, počty bodů $P_A = 100$ a hodnoty posunu $b_1s = 10$, $b_1l = 25$. Poslední konstanta $D_L = 1$ určuje minimální možnou vzdálenost rovnoběžek.

Z konstant a jednotlivých kroků algoritmu vyplývá několik předpokladů, při jejichž splnění metoda funguje:

- Sklony sousedících úseků se liší o více než $b_2s = 0.2$.
- Každý úsek se skládá nejméně ze 400 bodů a každý úsečka má více než 100 bodů.
- Vzdálenost mezi rovnoběžkami musí být větší než $D_L = 1$.

Popis algoritmu

Vstup: množina bodů, skládají se z dat jednotlivých úseků ležících za sebou

Výstup: seznam jednotlivých úseků

Kroky řešení:

1. Rozdělení množiny bodů na segmenty po P_A bodech. Nad segmenty je provedena lineární regrese a ty s rozdílem sklonu menším než b_2s jsou spojeny.
2. Segmenty s P_A a méně body jsou sjednoceny se sousedy při rozdílu sklonu menším než je dvojnásobek b_2s . V návaznosti na změny jsou sjednoceny i ostatní segmenty s rozdílem sklonu b_2s .
3. Je zkontrolováno, že nedošlo ke spojení dat, která k sobě nepatří. Jde hlavně o rozdělení většího množství bodů, která byla spojena přes společnou hranu sousedících úseků.

Kontrola je provedena pomocí segmentace dosud vytvořených segmentů stejně jako v prvním kroce algoritmu. Tentokrát jsou, ale rovnou spojeny segmenty s rozdílem sklonu b_2l . Jsou-li data segmentu rozdělena na jednu nebo více než tři části, je původní segment ponechán beze změn. Při rozdělení na dvě nebo tři části jsou data segmentu přerozdělena v závislosti na počtu bodů a pořadí segmentu.

4. V této fázi rozdělování je možné, že některé úseky jsou ještě rozděleny na více segmentů, které je třeba spojit. Při vývoji se ukázalo, že nejlepší je neprve vyřešit propojení segmentů s malým počtem bodů. Prvním krokem je tedy přiřazení segmentů s 300 ($3 * P_A$) a méně body k sousedovi s bližším sklonem. V následujícím kroce při splnění kritérií spojeny další segmenty.
5. Spojování dalších segmentů je prováděno ve smyčce. Nejsou přitom spojovány jenom dvojice, ale i trojice segmentů. Může totiž nastat situace, kdy data patřící k sobě jsou rozdělena na tři části a prostřední část svojí hodnotou sklonu a posunu brání jejich spojení.

Dvojice jsou spojeny, je-li rozdíl sklonu b_2s a posunu b_1l . Trojice se spojí při rozdílu sklonu menším než b_2s a pokud rozdíl posunů není větší než b_1s . Obsahuje-li prostřední segment 300 a méně bodů, počítá se s dvojnásobným rozdílem sklonů b_2s .

6. V tomto kroce dojde u segmentů k odstranění dat, která k úseku nepatří a spojí se rozdělené rovnoběžky. Odstranění se týká hlavně bodů na hranicích úseků.

Nejprve je rozhodnuto, zda je lepší seřadit body podle x -ových nebo y -ových souřadnic bodů. Následně dochází k rozdělení segmentů na všechny možné samostatné úsečky. Na základě rozdělení jsou segmenty profiltrovány, případně spojeny.

Hledání úseček se provádí na základě vzdálenosti bodů s hranicí určenou pomocí hodnoty D_T . Úsečky s méně než s P_A body jsou odstraněny.

Segmenty, které byly rozdělené na jedinou úsečku, jsou přiřazeny k sousednímu segmentu s bližší hodnotou sklonu.

3.2 Detekce os symetrie

Máme-li data ve formě množin bodů rozdělená na jednotlivé rovnoběžné úseky, viz obrázek 3.2, je třeba ke každému úseku nalézt osu symetrie. Osa nalezená programem pro detekci symetrie je na výstupu ve formě vektoru o čtyřech složkách $[a, b, c, d]$, kde (a, b, c, d) jsou koeficienty rovnice roviny symetrie. Protože se pracujeme ve 2D, je třetí koeficient roven nule ($c = 0$).

K nalezení osy lze použít jeden ze dvou připravených způsobů:

- Jednotlivé úseky jsou postupně vloženy do programu na detekci symetrie, vysvětleného v sekci 2.3.
- Osa symetrie je spočítána přímo z bodů.

3.2.1 Užítí programu detekce symetrie

Při použití programu pro detekci symetrie se s daty před samotnou detekcí nijak nemanipuluje. Na data se nekladou žádné nároky, protože program na detekci symetrie žádné nemá, jak je již zmíněno v sekci 2.3.

Nevýhoda spočívá v tom, že program často nalézá osu symetrie, která je k rovnoběžkám kolmá a nikoliv s nimi rovnoběžná, jak je blíže popsáno v sekci 4.1. Z experimentů plyne, že délka rovnoběžek musí alespoň desetinásobek násobek poloměru vzdálenosti mezi nimi, leží-li konce rovnoběžek proti sobě. Jsou-li posunuté, je tento násobek ještě větší. Navíc v některých případech se požadovaná osa nenajde vůbec.

3.2.2 Přímé spočtení osy symetrie

Pro přímé spočtení osy symetrie potřebujeme po dvou bodech z obou úseček. Dané body jsou získané spočítáním průměrného bodu z polovin rovnoběžek, aby se zabránilo ovlivnění výsledku případným šumem. Oba body z jedné rovnoběžky jsou spárovány s odlišnými body rovnoběžky druhé. Mezi těmito páry je uprostřed vygenerován bod, kterým prochází hledaná osa symetrie. Protože takové body máme dva, můžeme je použít ke spočtení rovnice přímky osy.

3.3 Spojení os

Rovnice přímek os symetrie jednotlivých úseků jsou při jejich spojování očekávány ve formě vektoru o čtyřech složkách, kdy třetí složka je nulová. K jejich spojení je kromě os třeba mít k dispozici i příslušné rozdělené úseky.

Při spojování os se rozhoduje, zda začátky a konce úseček os budou odpovídat jednomu z průsečíků dvojice os nebo tzv. původnímu kandidátovi.

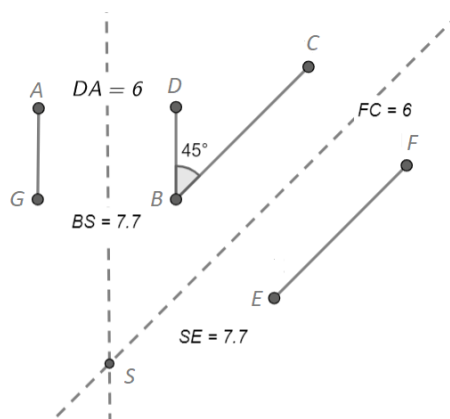
Původní kandidáti se zjišťují pomocí x -ových nebo y -ových (svislé úsečky) souřadnic extrémů rovnoběžek, ke kterým se spočte druhá souřadnice, aby vznikl bod ležící na ose. Extrémy jsou nalezené tak, že se rozdělí rovnoběžky, vezmou se jejich minimální a maximální x -ové (respektivě y -ové) souřadnice a u obou páru (minima a maxima) se spočte průměr.

Určené průsečíky se počítají mezi každým párem os a rozhoduje se o jejich validitě. Průsečíky os (x_i, y_i) se počítají pomocí rovnic 3.1.

$$x_i = \frac{b_1 \cdot d_2 - d_1 \cdot b_2}{a_1 \cdot b_2 - a_2 \cdot b_1} \quad y_i = \frac{a_2 \cdot d_1 - d_2 \cdot a_1}{a_1 \cdot b_2 - a_2 \cdot b_1} \quad (3.1)$$

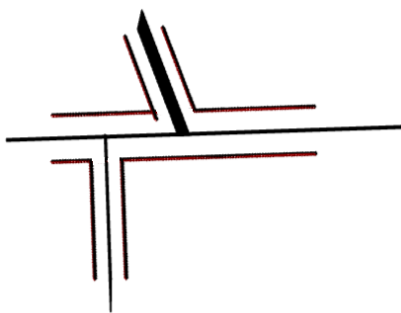
Nejprve se zkontroluje, zda nejsou osy rovnoběžné, tj. $a_1 \cdot b_2 - a_2 \cdot b_1 \neq 0$. V ostatních případech se zkontroluje, zda je průsečík v limitní vzdálenosti od bodů obou úseků příslušných os. Limitní vzdálenost dvou rovnoběžek je nastavena na 1.5 delší vzdálenosti mezi rovnoběžkami.

Limitní vzdálenost byla určena tak, aby průsečík os dvou úseků, které sdílejí jediný bod a svírají mezi sebou úhel větší než 45° , byl prohlášen za platný. Popsanou situaci znázorňuje obrázek 3.3. Zde je B společný bod rovnoběžek, jeho vzdálenost od průsečíků os S je $|BS| = 7.7$ a rovnoběžky mají stejnou vzdálenost, tedy $|DA| = 6$ a $|FC| = 6$. Hodnota limitu byla stanovena pomocí experimentů s různými rovnoběžkami a úhly v programu GeoGebra [12].

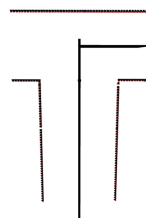


Obrázek 3.3: Zdůvodnění limitu

Po určení validních průsečíků se rozhoduje, zda budou hraničními body os. Rozhodování se provádí porovnáním, zda se průsečík nachází v limitní vzdálenosti od hraničních bodů osy, určených dle extrémů úseků, jak je popsáno výše. Rozhodování probíhá nezávisle od úseku druhého, protože hranice osy jednoho úseku nemusí být nutně hranicí i toho druhého, jak je vidět na obrázku 3.4. Při určování, zda je průsečík hraničním bodem osy, se také kontroluje, že je od opačné hranice dále než tři čtvrtiny vzdálenosti původních kandidátů. Tato kontrola se dělá pro zabránění situaci zobrazené na obrázku 3.5. Dojde-li ke střetu více průsečíků o stejný hraniční bod, zvolí se ten, který byl k původní hranici osy blíže.



Obrázek 3.4: Spojení os



Obrázek 3.5: Spojení os bez kontroly vzdálenosti od opačného konce

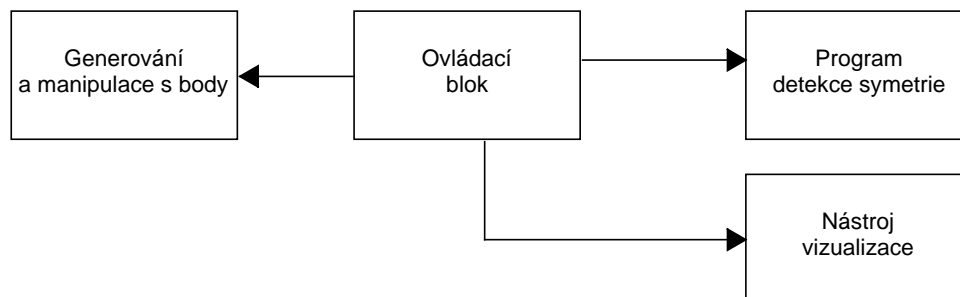
3.4 Generování množin bodů

Pro otestování bylo třeba mít možnost si generovat testovací data. Byly vytvořeny dvě verze generování dat. V první verzi mají data podobu bodů ležících střídavě na obou rovnoběžkách a druhý způsob nejprve vygeneruje body jedné z úseček, za kterou následují data úsečky druhé. Lze si zvolit možnost zamíchat data úseku. Generovaná data lze také upravit přidáním šumu.

3.5 Technické řešení

Řešení je naprogramované v jazyce .NET/C#, verze .NET 4.6.1. Je složené z několika projektů. Většina byla poskytnuta Ing. Hrudou a týká se programu pro detekci symetrie a vybavení využívané pro vizualizaci. V rámci práce se pracovalo hlavně s dvěma projekty, které byly původně vytvořeny Eliškou Mourycovou, a v rámci této práce byly dané moduly rozšířené o nové třídy a funkce.

Projekty lze rozdělit podle jejich funkce do skupin. Skupiny a jejich vzájemné propojení znázorňuje obrázek 3.6.



Obrázek 3.6: Přehled projektu

Jak je z obrázku 3.6 vidět, klíčové jsou projekty skupiny Ovládací blok. Mezi tyto projekty patří GUI (v práci nepoužitý), Client a GUIFormDataGenerator. Přes jedinou třídu Program.cs projektu Client se přistupuje k programu pro detekci symetrie a programu pro vizualizace. V tomto projektu přibyly metody, které umožňují použití těchto funkcí pro potřeby vzniklého programu. Projekt GUIFormDataGenerator obsahuje grafické rozhraní vytvořené aplikace. Dochází v něm k používání metod z projektů Client a FunctionDataGenerator.

FunctionDataGenerator patří do skupiny Generování a manipulace s body z obrázku 3.6. Společně s projektem GUIFormDataGenerator jde o projekt vytvořený Eliškou Mourycovou, do nichž byly naimplementované nové funkce programu. Do projektu FunctionDataGenerator byly vytvořeny následující třídy:

- DataGeneratorAxis - generování množin bodů, popsané v sekci 3.4
- AxisFunction - přepravka (návrhový vzor messenger) obsahující funkci osy, kolem které se generují body
- DefinedAxisFunctions - obsahuje seznam nadefinovaných instancí AxisFunction
- ConnectAxis - spojení os symetrie dle sekce 3.3.
- AxisFuncProcessing - obsahuje rozdělení množiny bodů ze sekce 3.1 a přímé spočtení osy symetrie 3.2.2

Ze tříd vytvořených E. Mourycovou byla rozšířena o nové funkce třída Point, která reprezentuje bod. Při testování byla využita metoda třídy DataGenerator, která do dat zanáší šum. Důležitá je také třída Functions, ze které byla použita metoda umožňující získání hodnoty z řetězcově zapsané funkce a byla přidána metoda kontrolující uživatelem napsanou funkci.

Zbývající projekty jsou:

- FindSymmetry, Framework -> Program detekce symetrie
- SlimDXRenderer, SlimdDXRenderSystem, Visicounter -> Nástroj vizualizace množiny bodů a rovin symetrií.

Pozn. V projektech skupiny Nástroj vizualizace množiny bodů a rovin symetrií byla upravena likvidace vytvořených instancí, aby se předešlo nedostatku paměti. (před úpravou zůstávala v paměti veškerá data vizualizačního okna).

Uživatelská příručka s obsahem informací pro uživatele se nachází v příloze A a jako samostatný soubor je přiložená u programu.

4 Experimenty a výsledky

Tato kapitola obsahuje ukázky různých provedených experimentů. Jednotkou délky všech experimentů jsou centrimetry. Experimenty byli uskutečněny na zařízení s procesorem *Intel(R) Core(TM) i5-8300 CPU 2.30 Ghz (8 CPUs)* a 8 GB paměti na operačním systému *Windows 11*.

Experimenty jsou podle jejich obsahu rozdělené na tři sekce v následujícím pořadí.

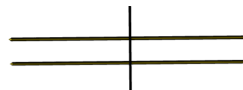
1. Sekce 4.1 zkoumá možnosti použití programu pro detekci symetrie. Nejprve jsou nalezeny parametry dat, při kterých má metoda největší šanci pro nalezení požadované osy symetrie. Následně jsou provedeny experimenty úspěšnosti nalezení osy a je prozkoumán vliv rotace dat na výsledek. V závěru sekce jsou zmíněné pokusy o úpravu dat.
2. Sekce 4.2 obsahuje experimenty ověřující limity a funkčnost metody rozdělující data. Je zkoumán vliv uspořádání a počtu bodů. Dále je otestována funkčnost metody na nerovnoměrných a zašumělých datech.
3. V sekci 4.3 jsou výsledky přímého spočtení os symetrie. Nejprve je metoda otestována na rovnoměrně rozložených datech bez šumu, poté na nerovnoměrných datech a na konci je ověřena funkčnost řešení při přidání šumu.

4.1 Limity programu detekce symetrie

Tato sekce se zabývá experimenty, kterými bylo zkoumáno, při kterých datech najde program pro detekci osy symetrie požadovanou osu. Nástroj totiž často místo hledané osy symetrie, tedy osy rovnoběžné s objektem, viz obrázek 4.1, najde osu kolmou na rovnoběžky viz obrázek 4.2.



Obrázek 4.1: Požadovaný výsledek



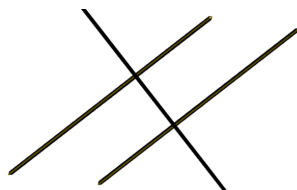
Obrázek 4.2: Chybný výsledek

4.1.1 Limity délky úseček

Během tohoto experimentu byl zkoumán vliv délky čar úseků při různé vzdálenosti rovnoběžek od osy a počtu bodů na jednotlivých úsečkách.

Experimentovalo se s případy, kdy je délka čar větší než vzdálenost rovnoběžek od osy.

K daným kritériím se testoval také vliv případného vzájemného posunutí rovnoběžek, jak je zobrazeno na obrázku 4.3. Při pokusu se konkrétně generovaly úsečky, jejichž směrnice měly úhel 45° vzhledem k ose x .



Obrázek 4.3: Vzájemně posunuté rovnoběžky

Experiment byl prováděn generováním rovnoběžných úseček s měnícími se parametry. Konkrétně se zkoumaly počty bodů 100, 250 a 500. K těm se volily vzdálenosti úseček od osy v rozsahu od 1 do 5. Pro tyto hodnoty se hledaly délky úseček, při kterých by byla nalezena osa rovnoběžná k úsečkám.

Výsledky experimentu

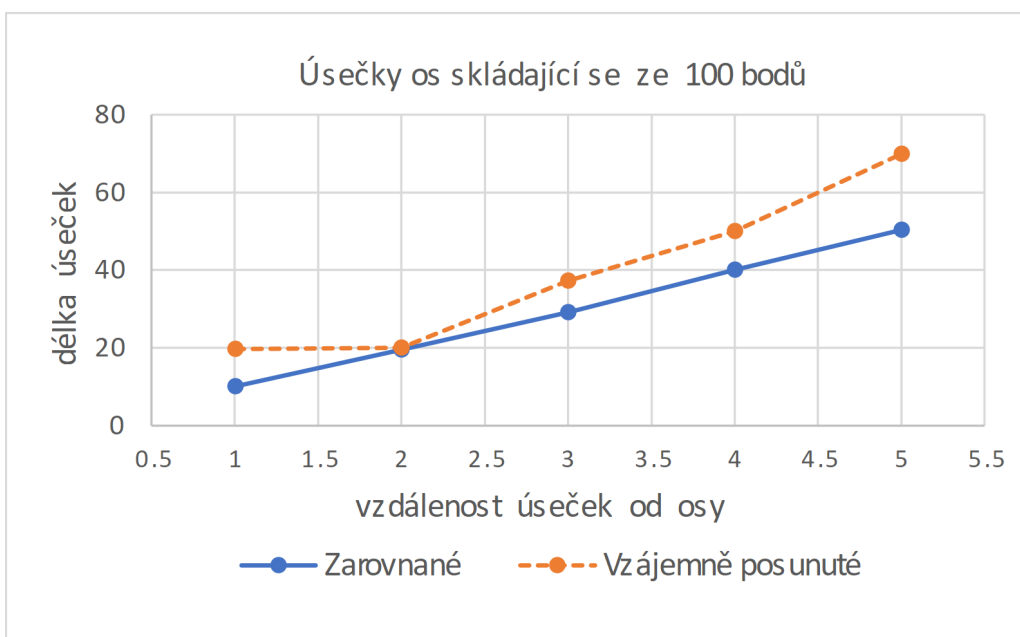
Výsledky experimentu jsou uvedeny v tabulkách 4.1, 4.2, 4.3. Tabulky uvádí nalezené minimální délky úseček rovnoběžek, při kterých se našla požadovaná osa. Data tabulek jsou zobrazená i v grafech 4.4, 4.5, 4.6 vytvořených pomocí tabulkového procesoru Excel.

r	vzájemně zarovnané	vzájemně posunuté
1	10,1	19,7
2	19,5	20,0
3	29,2	37,3
4	40,0	50,0
5	50,3	70,0

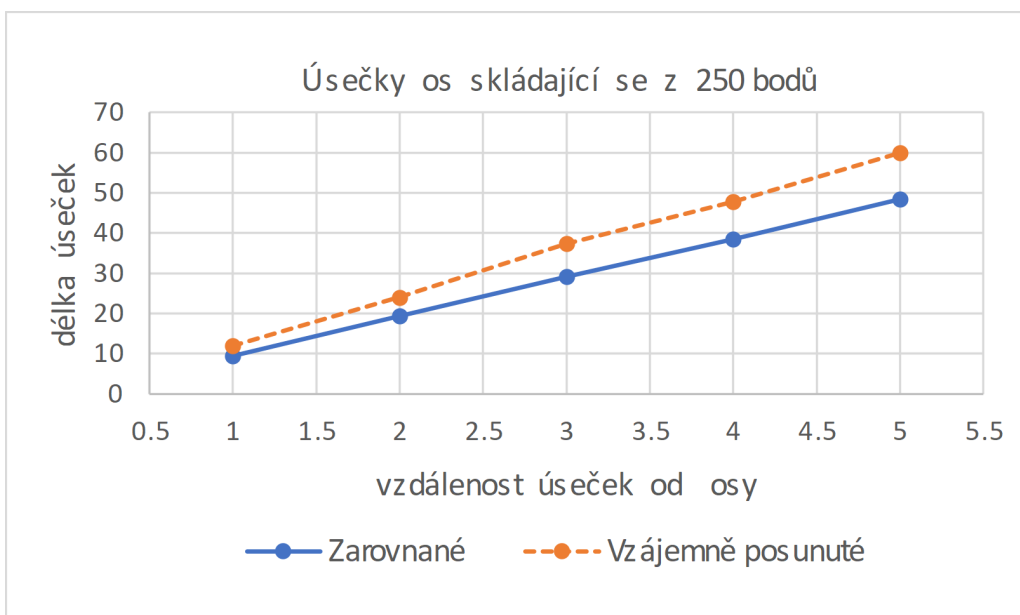
Tabulka 4.1: Délky úseček ze 100 bodů (r = vzdálenost osy od rovnoběžek)

r	vzájemně zarovnané	vzájemně posunuté
1	9,4	12,0
2	19,4	24,0
3	29,2	37,4
4	38,5	47,8
5	48,4	60,0

Tabulka 4.2: Délky úseček z 250 bodů (r = vzdálenost osy od rovnoběžek)



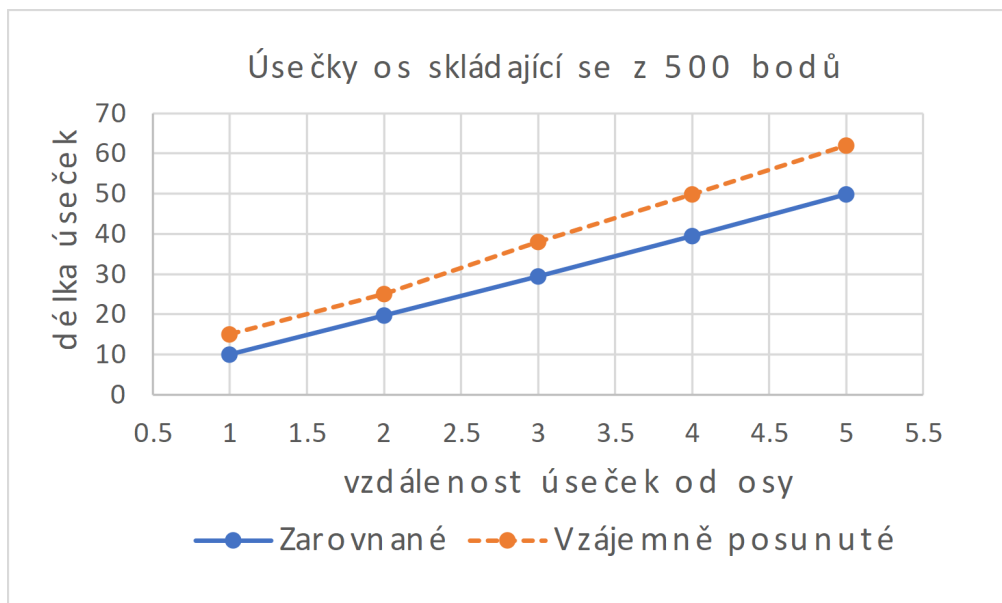
Obrázek 4.4: Graf 100 bodů



Obrázek 4.5: Graf 250 bodů

r	vzájemně zarovnané	vzájemně posunutě
1	10,0	15,0
2	19,7	25,0
3	29,5	38,0
4	39,5	49,9
5	49,9	62,0

Tabulka 4.3: Délky úseček z 500 bodů (r = vzdálenost osy od rovnoběžek)



Obrázek 4.6: Graf 500 bodů

Z tabulek 4.1, 4.2, 4.3 a jejich příslušných grafů 4.4, 4.5, 4.6 je vidět, že délka úseček s vzrůstající vzdáleností od osy v případě zarovnaných i vzájemně posunutých rovnoběžek roste zhruba lineárně, a to ve všech zkoumaných případech. U zarovnaných rovnoběžek byla nutná délka úseček pro požadovaný výsledek přibližně desetinásobek poloměru. Zároveň se ukázalo, že použité nízké počty bodů mají na výsledek zanedbatelný vliv.

Závěr

Na základě experimentu bylo rozhodnuto, že při hledání symetrie je nejvhodnější pracovat s rovnoběžnými úsečkami, které jsou zarovnané. Navíc by úsečky měly mít délku, která je nejméně desetinásobek vzdálenosti úseček od osy. Pro počet bodů bylo rozhodnuto používat nejméně 500 bodů na úsečku.

Krátké úsečky


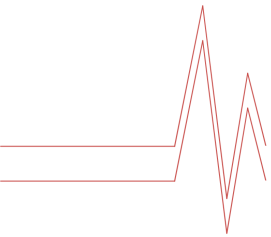
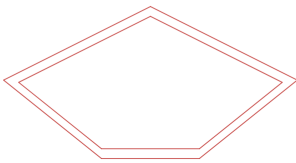
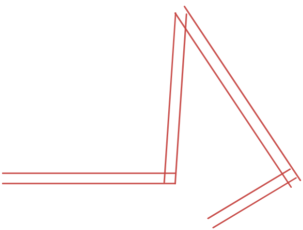
V tomto dodatečném experimentu šlo o zkoumání, v jakém poměru jsou délky čar ku poloměru vzdálenosti od osy. Hledaný poměr se týkal případů, kdy délka čar je menší než vzdálenost od osy. Byly zkoumány délky čar o velikosti 1, 2, 3, 4, 5 a 10.

Největší fungující nalezený poměr byl 0,198. Hodnotu nelze ovšem považovat za bezpečnou, jelikož u délky čar 1 docházelo při menším poměru k nekonzistentním výsledkům, kdy v různých případech byla osa správně rovnoběžná a jindy kolmá na vygenerovanou čáru. Za bezpečný byl tedy stanoven poměr 0,179.

Experiment byl proveden v rámci hledání možných řešení problému. Nakonec se z důvodu praktičnosti rozhodlo zůstat u délek čar větších než poloměry.

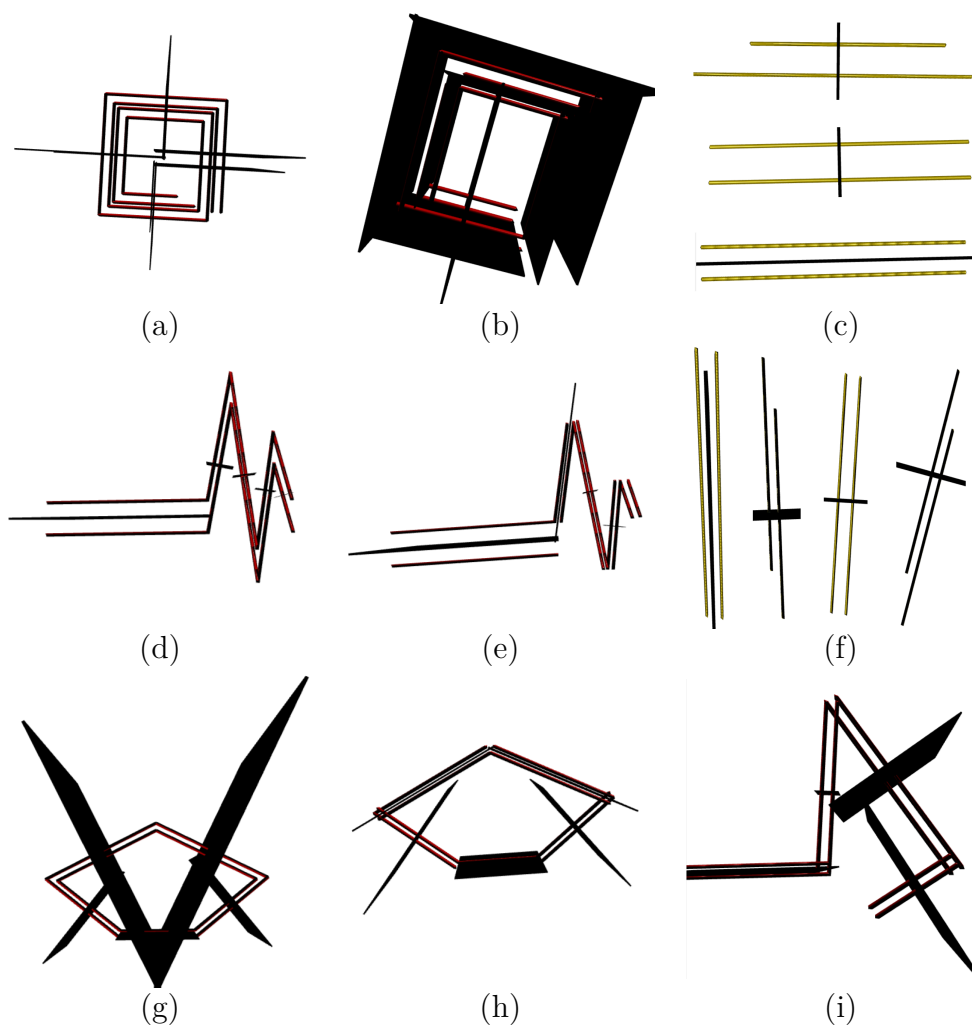
4.1.2 Experimenty s lomenými čarami

Tato sekce se zabývá experimenty s vybranými lomenými čarami a softwarem pro detekci symetrie. V tabulce 4.4 jsou vidět lomené funkce a jejich rovnice použité v experimentech.

Vizualizace	Rovnice	r	Popis
	$y = x \{0 < x < 54\}$ $y = -x + 100 \{0 < x < 54\}$ $y = x + 100 \{-54 < x < 0\}$ $y = -x \{-54 < x < -8\}$ $y = x + 16 \{-8 < x < 38\}$ $y = -x + 84 \{0 < x < 38\}$ $y = x + 84 \{-38 < x < 0\}$ $y = -x + 16 \{-38 < x < -3\}$	4	spirála
	$y = 0 \{-50 < x < 0\}$ $y = 5x \{0 < x < 8.077\}$ $y = -8x + 105 \{8.077 < x < 15\}$ $y = 6x - 105 \{15 < x < 21\}$ $y = -4x + 105 \{21 < x < 26.25\}$	5	tep
	$y = -45 \{-50 < x < 50\}$ $y = 0.8 * x - 85 \{50 < x < 150\}$ $y = -0.5 * x + 100 \{0 < x < 150\}$ $y = 0.5 * x + 100 \{-150 < x < 0\}$ $y = -0.8 * x - 85$ $\{-150 < x < -50\}$	5	pětiúhelník
	$y = 0.6 * x \{-2.5 < x < 42.9\}$ $y = -1.5 * x + 84 \{-16 < x < 45\}$ $y = 15 * x + 300$ $\{-21.3 < x < 15.8\}$ $y = 22.5 \{-16 < x < 100\}$	3 5 40 2.5	lomenice

Tabulka 4.4: Testované lomené objekty (r = posun hodnoty bodu Y od osy (+-))

Aby výsledky neovlivnily případné odchylky při rozdělávání bodů metodou z 3.1, byly lomené čáry rozdělené méně obecnou metodou založenou na směrnici a seřazených datech. Této pomocné metodě byly navíc pro dosažení požadovaného rozdělení generované množiny dány v pozměněném pořadí.



Obrázek 4.7: Výsledky testovaných lomených objektů

Obrázek 4.7 obsahuje výsledky při použití metody detekce symetrie Ing. Hrudy. Jak je vidět, u spirály (a) nebyla ani u jedné části nalezena požadovaná osa. Jedna samotná část spirály zobrazená nahoře v obrázku (c) ukazuje, že je to způsobeno tím, že kratší úsečka je zkrácena z obou stran. Je tedy nalezena správná osa symetrie, ale nikoliv ta, kterou potřebujeme. Při zarovnání rovnoběžek částí spirály (b) bylo detekování os úspěšnější. Požadovaná osa nebyla nalezena pouze u nejkratšího úseku, jeho délka po oříznutí byla 41. Obrázek (c) znázorňuje uprostřed tuto nejkratší část a dole se nachází ukázka nalezení požadované osy u jednoho z ostatních úseků.

U tepu se povedlo najít požadovanou osu symetrie v prvním úseku a v případě seříznutí ještě v úseku následujícím, jak zobrazují obrázky (d) a (e). Obrázek (f) zobrazuje neseříznuté a seříznuté výsledky druhého a předposledního úseku tepu. Všechny úseky kromě prvního měly vzdálenost mezi

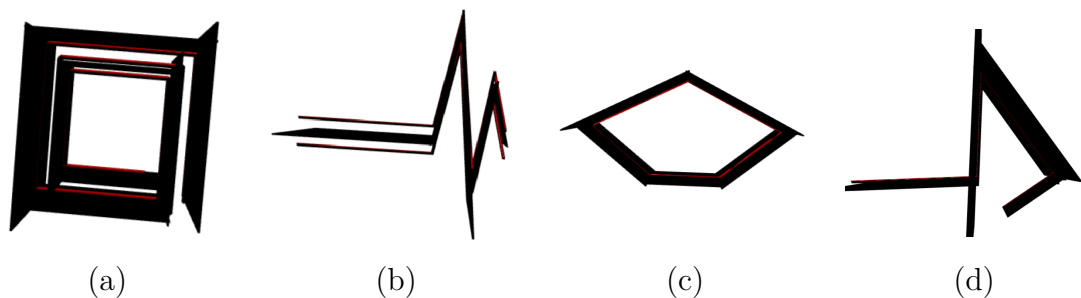
rovnoběžkami v rozmezí od 1.24 do 2.4 a délku větší než deseti násobek vzdálenosti úseček od osy. Úsek s rovnicí $y = -8x$ byl zvlášť zkoumán s různými délkami a vzdálenostmi od osy, ale nepovedlo se najít rozměry, při nichž by se jako osa symetrie vyhodnotila požadovaná osa. V případě rovnice $y = 6x$ se povedlo najít rozměry, při nichž se našla požadovaná osa, ale výsledky nebyly konzistentní.

U zarovnaného pětiúhelníku (h) se nepovedlo najít požadovanou osu u rovnic zahrnujících sklon $\pm 0.8x$. Podobně jako u tepu se ani zde nepovedlo při změnách délky a vzdálenosti od osy nalézt parametry, při kterých by se našla požadovaná osa. Narozdíl od rovnic tepu se ale podařilo dosáhnout požadovaného výsledku při změně počtu bodů z 1000 na 500.

V případě lomenice se rovnou pracovalo se zarovnanými úseky, které navíc byly sestaveny tak, aby všechny měly zhruba stejnou vzdálenost rovnoběžek pohybující se mezi hodnotami 5 a 5.5, od požadované osy byly tedy úsečky rovnoběžek vzdáleny v rozmezí od 2.5 do 2.75. Jak ukazuje obrázek (i) pouze u zcela horizontálního úseku se povedlo nalézt požadovanou osu symetrie.

Z experimentu vychází závěr, že program pro detekci symetrie není pro hledání zobecněné osy symetrie vhodný, protože se často stává, že nalezená osa symetrie, není ta, kterou potřebujeme. Zároveň z experimentu vyplynula hypotéza, že na výsledek detekce osy symetrie má vliv rotace rovnoběžek, což je zkoumáno dalším experimentem.

Při použití přímého výpočtu se požadovanou osu nalézt podařilo, jak je vidět na obrázku 4.8.



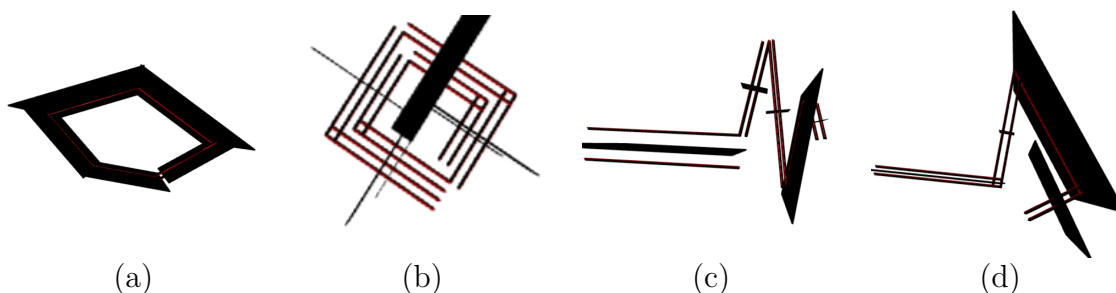
Obrázek 4.8: Výsledky použití přímého spočtení osy na lomených objektech

Zkoumání vlivu rotace

V tomto experimentu šlo o prozkoumání vlivu rotace na výsledky metody pro detekci symetrie.

Při rotaci se pomocí lineární regrese určily koeficienty rovnice přímky a následně se najdou dva body ležící na přímce. Pomocí těchto bodů se určí směrnice ve stupních a spočítá se rozdíl s cílovým úhlem. O hodnotu rozdílu jsou následně změněny pozice všech bodů množiny. Nové souřadnice bodů se počítají dle vztahu v 4.1, kde *angle* je úhel rotace ve stupních, *x* a *y* jsou původní souřadnice a *x_r*, *y_r* jsou souřadnice bodů po rotaci. Úhel, o který jsou body otočeny, je zaznamenán a nalezená osa je otočená o jeho zápornou hodnotu.

$$a = \text{angle} * \frac{\pi}{180} \quad x_r = x * \cos(a) - y * \sin(a) \quad y_r = x * \sin(a) + y * \cos(a) \quad (4.1)$$



Obrázek 4.9: Výsledky rotace na 180°

Jak je vidět na obrázku 4.9, nejlepší výsledky při otočení o 180° měl objekt pentagonu (a), u něhož se povedlo najít všechny požadované osy symetrie. Nejhorší výsledek se vyskytl u labyrintu (b), kde se i přes pozitivní výsledek před otočením, nepovedlo při rotaci najít ani jednu požadovanou osu. U tepu (c) a lomenice (d) se našlo o jednu požadovanou osu více než před rotací.

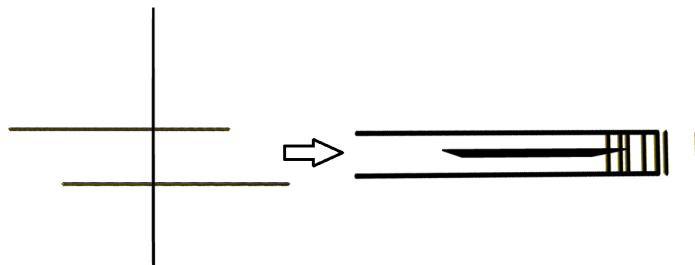
Kromě úhlu 180° byly zkoumány rotace o úhly 45° a 90°. Výsledky byly podobné výsledkům z tabulky 4.9 nebo z hlediska nalezení požadované osy horší.

Experiment tedy nevedl k nalezení úhlu, při kterém by metoda spolehlivěji našla požadovanou osu symetrie, ale zároveň ověřil hypotézu, že na výsledek detekce osy symetrie je ovlivněn rotací množiny bodů.

Ohledně vlivu rotace proběhla komunikaci s autorem metody ing. L. Hrudou. Podle něj je výběr finální osy symetrie z výběru několika stejně dobrých symetrií věc náhody. Na výsledek ale může mít vliv i zarovnání se souřadnicovými osami, protože metoda provádí mřížkovou simplifikaci a rozházení bodů do buněk mřížky závisí na orientaci i pozici objektu, přičemž svislá orientace může způsobit, že symetrie podél rovnoběžek může být o něco silnější. L. Hruďa navrhl, že donutit metodu pro preferenci podélné osy delšího rozměru objektu by nejspíš šlo s využitím váhy bodů. To by ovšem mohlo vést k dalším problémům a po konzultaci s vedoucí bylo rozhodnuto tímto směrem nejít.

4.1.3 Pokusy o úpravu dat

Během vývoje byli prováděny experimenty upravující data jednotlivých úseků tak, aby se co nejvíce zvýšila šance na nalezení požadované osy programem pro detekci osy symetrie. Úpravy dat zahrnovaly prodloužení úseček, jejich zarovnání a přidání pomocných kolmic. Stav dat před úpravami a po úpravách znázorňuje obrázek 4.10.

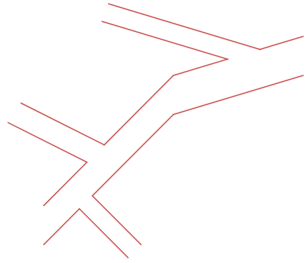
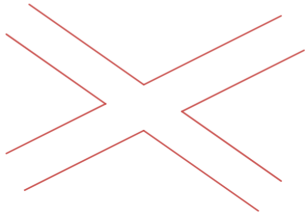
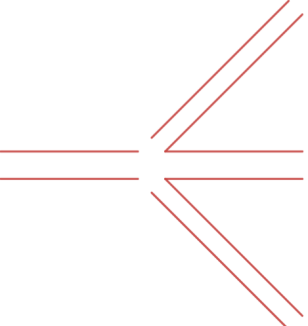


Obrázek 4.10: Data s nalezenou osou před a po úpravách

Provedené změny v datech zvýšily pravděpodobnost nalezení požadované osy, ale za cenu velké manipulace s množinou bodů. Tyto změny byly nakonec vyhodnoceny jako nadbytečné, protože byly tak rozsáhlé, že bylo praktičtější a logičtější určit osu symetrie rovnou přímým spočítáním.

4.2 Výsledky metody rozdělení bodů

V této sekci jsou uvedeny experimenty prováděné s vytvořenou metodou rozdělování bodů. Kromě lomených objektů z tabulky 4.4 se pracovalo s objekty stromů nadefinovanými v tabulce 4.5. Hlavním rozdílem mezi množinami dat z lomených čar a stromovými objekty je přítomnost mezer v úsecích. Při generování množiny bodů těchto úseků se vždy generují nejprve data jedné úsečky a poté data druhé.

Vizualizace	Rovnice	r	Popis
	$y = x \quad x_+ \in (0, 6.667) \cup (9.3, 20)$ $x_- \in (0, 5.5) \cup (7.5, 20)$ $y = 0.3 * x + 14 \quad x_- \in (20, 40)$ $x_+ \in (20, 28.4) \cup (33.3, 40)$ <hr/> $y = -0.3 * x + 35$ $x_+ \in (10.333, 40) \quad x_- \in (9, 28.4)$ $y = -x + 10 \quad x \in (5.5, 15)$ $y = -0.5 * x + 15$ $x \in (-5.5, 9.333)$	3 2	základní strom
	$y = 0.5 * x$ $x_+ \in (-30, -8.3) \cup (0, 30)$ $x_- \in (-26, 0.0) \cup (8.3, 35.0)$ <hr/> $y = -0.7 * x$ $x_+ \in (-25, 0) \cup (8.3, 30)$ $x_- \in (-30, -8.3) \cup (0, 25.0)$	5	křižovatka
	$y = 0 \quad x \in (0, 50) \cup (60, 110)$ $y = x - 50 \quad x \in (55, 110)$ $y = -x + 50 \quad x \in (55, 110)$	5	vidlice

Tabulka 4.5: Testované stromové objekty (r = posun hodnoty bodu Y od osy (+-))

Při vizualizaci výsledků se generuje i osa programem pro detekci symetrie, aby se zobrazená osa mohla srovnat s výsledky ze sekce 4.1. Osa navíc pomáhá k odlišení jednotlivých obrázků.

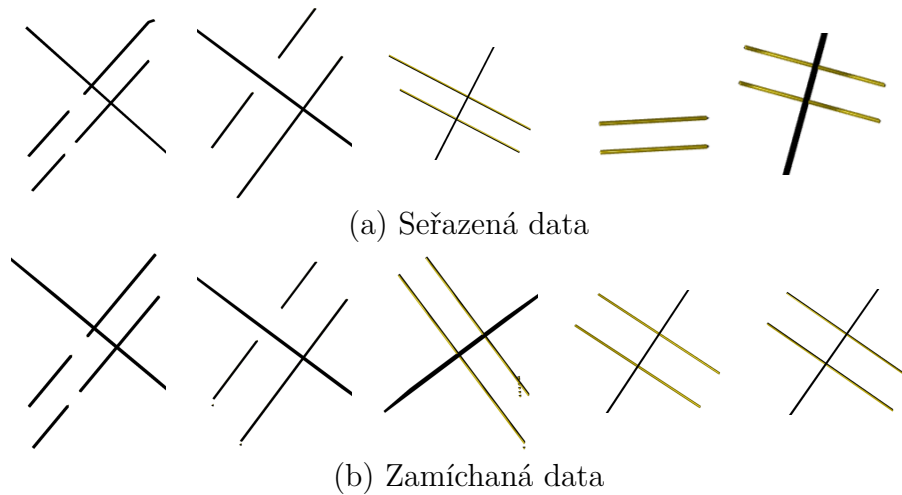
4.2.1 Vliv formy dat

Tento experiment zkoumal vliv seřazení množiny bodů na rozdělení dat.

Tabulka 4.11 zobrazuje výsledky rozdělení množiny bodů základního stromu. V základní podobě mají první tři úseky základního stromu data ve formě jedné úsečky následované daty druhé (a). Jak je vidět, data se

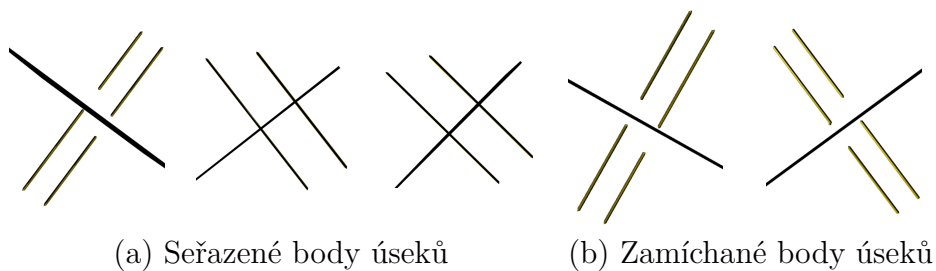
povedlo správně rozdělit. Výjimku tvoří poslední dva úseky, které byly metodou rozdělení dány dohromady. Oba tyto úseky měly body ležící střídavě na obou rovnoběžkách. I přes dostatečně odlišné sklony byly úseky dány dohromady.

Při zamíchání vygerovaných dat každého úseku (b) došlo v případě prvních tří úseků opět ke správnému rozdělení. Poslední dva úseky se, ale narozdíl od nezamíchaných bodů povedlo rozdělit správně.



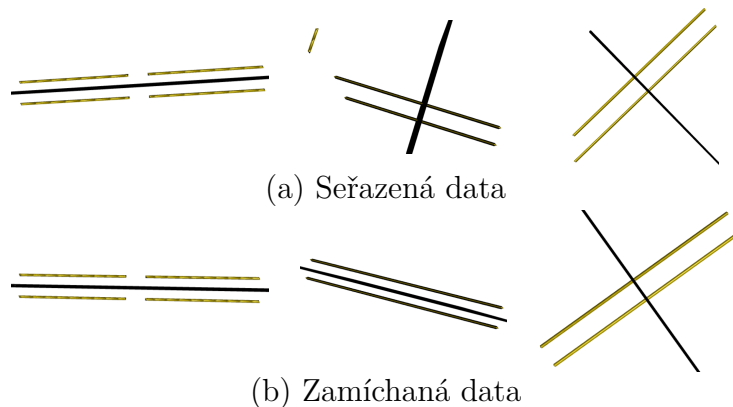
Obrázek 4.11: Rozdělení dat základního stromu

V případě dat křižovatky jsou data obou úseků defaultně v podobě jedné úsečky následovány body úsečky druhé. Rozdělení takových seřazených dat zobrazuje na obrázku 4.12 část (a). Data druhého úseku se následkem rozpuštění rozdělily na dvě části. Při zamíchání dat k tomuto jevu nedošlo, jak zobrazuje druhý obrázek (b).



Obrázek 4.12: Rozdělení dat křižovatky

Nejlepších výsledků se dosáhlo u dat vidlice 4.13. Jak při seřazení (a), tak při zamíchání (b) se data rozdělila na úseky správně. U seřazených dat došlo pouze u druhému úseku k přiřazení části dat patřících úseku dalšímu.



Obrázek 4.13: Rozdělení dat vidlice

Prozkoumání dat ukazuje, že metoda rozdělení množiny bodů má lepší výsledky u zamíchaných dat, a to jak z hlediska rizika spojení úseků, tak i v případě nesprávného rozdělení. Tento jev je vzhledem ke způsobu použití lineární regrese při rozdělení bodů metodou z 3.1 logický. Výsledky u seřazených bodů jsou ovšem z větší míry též správné.

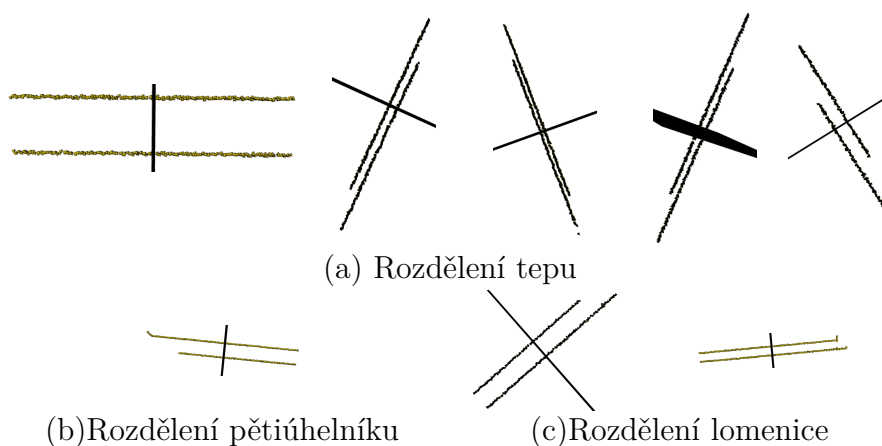
4.2.2 Vliv počtu bodů

V tomto experimentu se zkoumalo, jak je rozdělení bodů na jednotlivé úseky úspěšné v závislosti na počtu bodů. Body částí v ukázkách byly při jejich generování zamíchány.

Minimální počet bodů

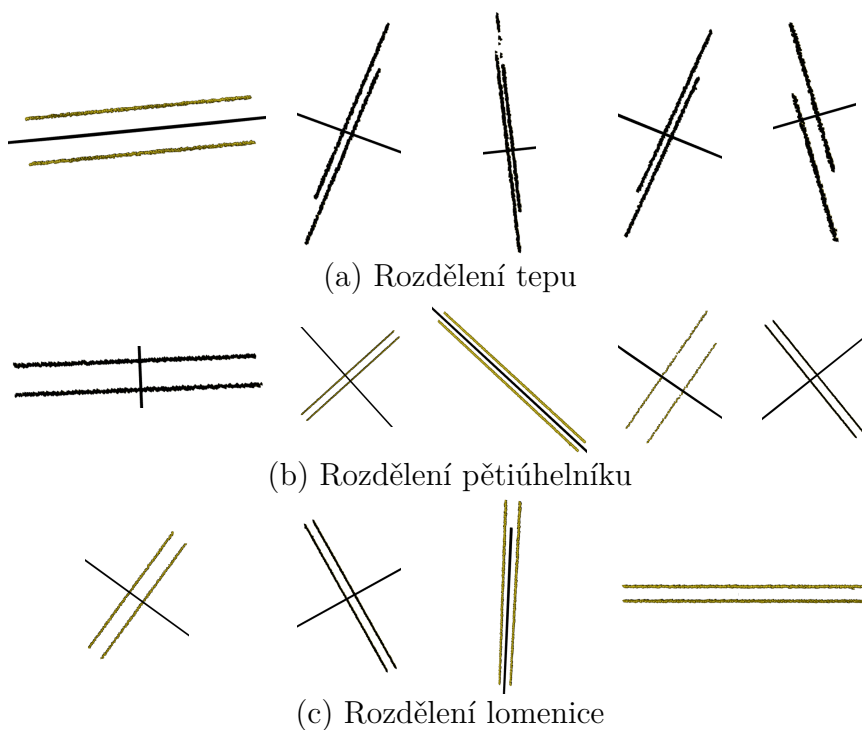
Z algoritmu rozdělení 3.1 vyplývá, že by mělo jít rozdělit data, jejichž úsečky se skládají z více než 100 bodů a úsek má dohromady alespoň 400 bodů. Byly proto ověřeny množiny bodů s přibližně 200 body na úsečku úseku. Data úseků byla zamíchána a byl přidán mírný šum, aby se našla optimální hranice minima bodů.

Jak ukazuje výsledek rozdělení tepu (a) na obrázku 4.14, při daném počtu bodů může metoda fungovat. V několika případech jsou ovšem výsledky více než znepokojivé, jak ukazuje případ pětiúhelníku (b) a lomenice (c), kdy došlo k odstranění dat většiny úseků. Výsledkem je, že máme-li na výběr, je lepší metodu použít s více než 200 body na úsečku a 400 body na úsek.



Obrázek 4.14: Výsledek rozdělení zamíchaných úseků s 200 body na úsečku

Po navýšení počtu bodů úsečky na 500 se dosáhlo o poznání lepších výsledků. Ukázky v obrázku 4.15 zahrnují rozdělení tepu (a), pětiúhelníku (b) a lomenice (c). Ve všech případech se objekty rozdělily správně. U posledního úseku lomenice došlo k zajímavému výsledku nalezení osy symetrie, jejíž detail zobrazuje obrázek 4.16. Hranice 500 bodů se dá z hlediska rozdělení na úseky považovat za bezpečnou hranici minima bodů na samostatnou úsečku úseku.



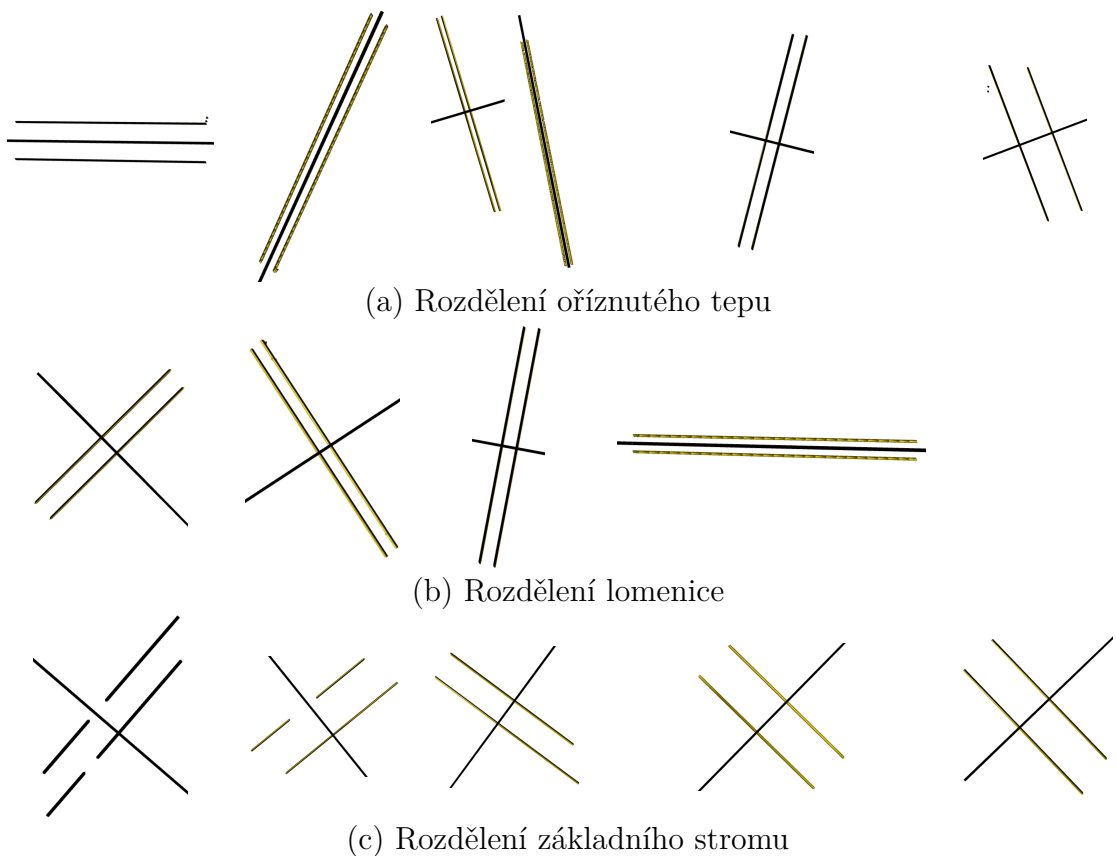
Obrázek 4.15: Výsledek rozdělení zamíchaných úseků s 500 body na úsečku



Obrázek 4.16: Detail programem nalezené osy v posledním úseku lomenice

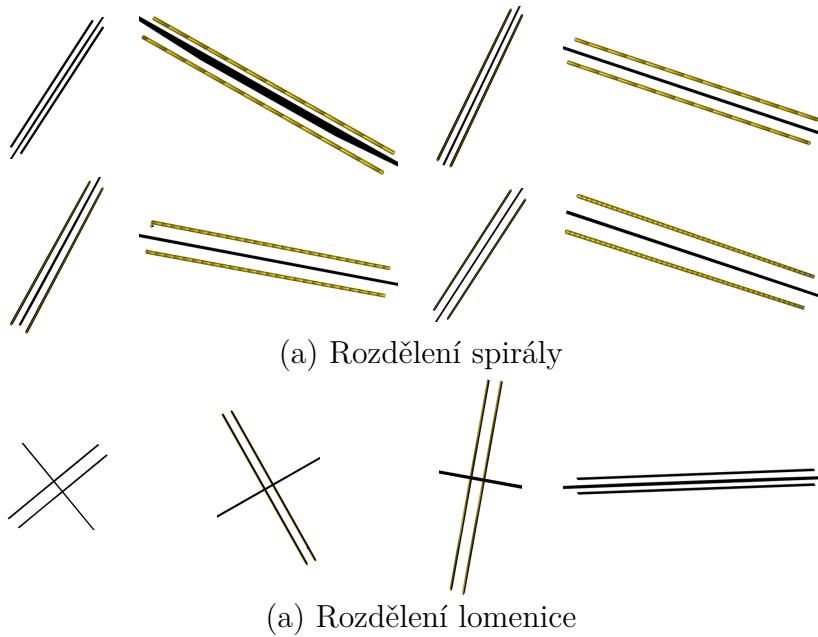
Obecné množství bodů

Na obrázku 4.17 jsou ukázky rozdělení dat, jejichž jednotlivé úsečky se skládají z 2333 bodů. Jde tedy o počet bodů nedělitelný 100. V případě lomenice (b) a základního stromu (c) se data rozdělila na své úseky přesně. U oříznutého tepu (a) došlo u jednoho z úseků k rozdělení dat. Nejedná se o problém, jenž by souvisel s obecným počtem bodů, ale o občasný výskyt rozdělení jednoho úseku na dvě množiny bodů. Tato chyba je brána za přístupnou, protože při generování osy symetrie se v závislosti na metodě osy buď překryjí, nebo vzájemně napojí.



Obrázek 4.17: Výsledek rozdělení zamíchaných úseků s 2333 body na úsečku

Při použití vysokého počtu obecného množství bodů přináší rozdělení také dobré výsledky, jak je znázorněno na obrázku 4.18, kde se nachází rozdělení spirály (a) a lomenice (b), jejichž úsečky jednotlivých úseků se skládají z přibližně 43247 bodů. V případě obou objektů došlo ke správnému rozdělení. Zajímavé je, že programem nalezené osy úseků většinou odpovídají nalezeným osám z výsledků ze sekce 4.1.2, ale v případě spirály je výsledek lepší, protože se podařilo nalézt požadovanou osu u všech částí objektu.



Obrázek 4.18: Výsledek rozdělení dat zamíchaných při 43247 bodech na úsek

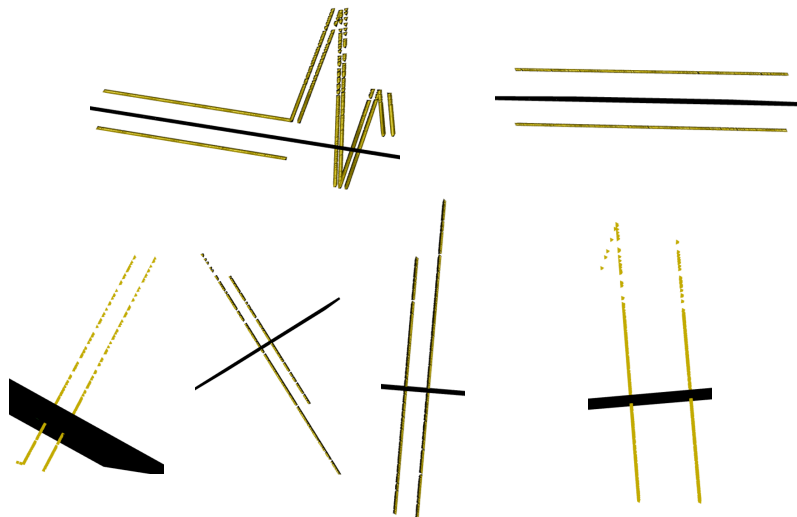
Shrnutí

Experiment ukázal, že rozdělení nezdeformovaných bodů funguje bezpečně při 500 bodech na úsečku úseku. Došlo také k potvrzení, že metoda dobře funguje při obecném počtu bodů nedělitelných 100 a v případě velkého množství bodů.

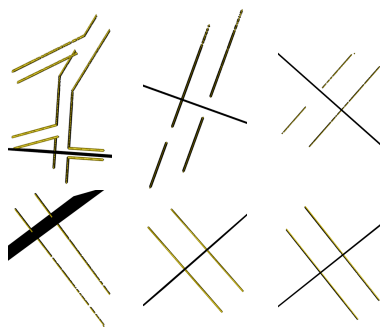
4.2.3 Nerovnoměrné rozdělení bodů

V tomto experimentu se zkoumal výsledek rozdělení bodů v případě nerovnoměrného rozložení bodů. Takového rozložení je dosaženo odstraněním náhodného počtu bodů z různých částí vygenerované množiny bodů.

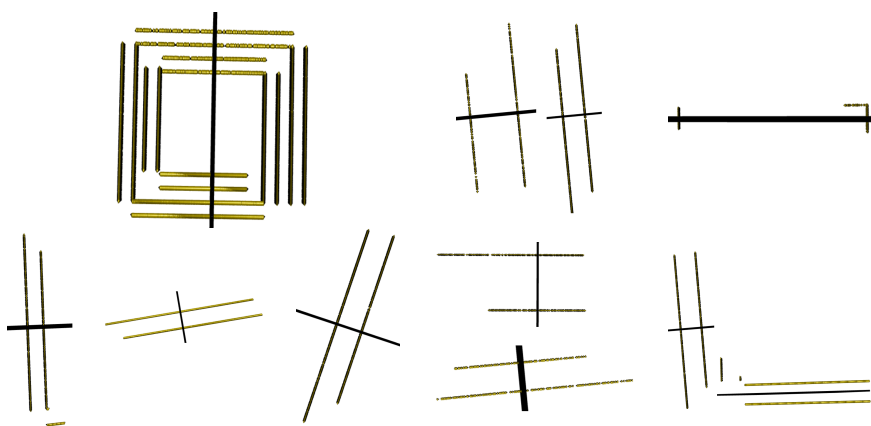
Ukázkové obrázky 4.19, 4.20 a 4.21 obsahují stav ořezaného tepu, základního stromu a ořezané spirály po úpravě, spolu s jejich rozdělením. Tep a základní strom se rozdělily přesně. U tepu přitom došlo k odlitrování částí úseček. V případě ořezané spirály došlo u dvou úseků k rozdělení na dvě části. Což vzniklo pravděpodobně příčinou seřazenosti dat, takové rozdělení je, ale považováno za přijatelné, protože při spojování os, se osy takhle rozdělených úseků prostě napojí. Výrazný problém nastal u třetího úseku, kde došlo k odfiltrování většiny bodů. Ve většině případů je tedy metoda u nerovnoměrně rozdělených bodů stejně úspěšná, jako v případě rovnoměrného rozložení. V některých případech, může ale dojít vlivem nerovnoměrnosti ke ztrátě části bodů.



Obrázek 4.19: Výsledky rozdělení nepravidelné množiny bodů oříznutého tepe, 1000 bodů na úsečku, seřazené



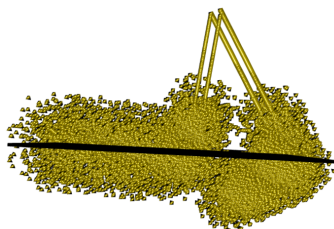
Obrázek 4.20: Výsledky rozdělení nepravidelné množiny bodů základního stromu - původně 2500 bodů na úsečku, zamíchané



Obrázek 4.21: Výsledky rozdělení nepravidelné množiny bodů spirály - původně 2333 bodů na úsečku, seřazené

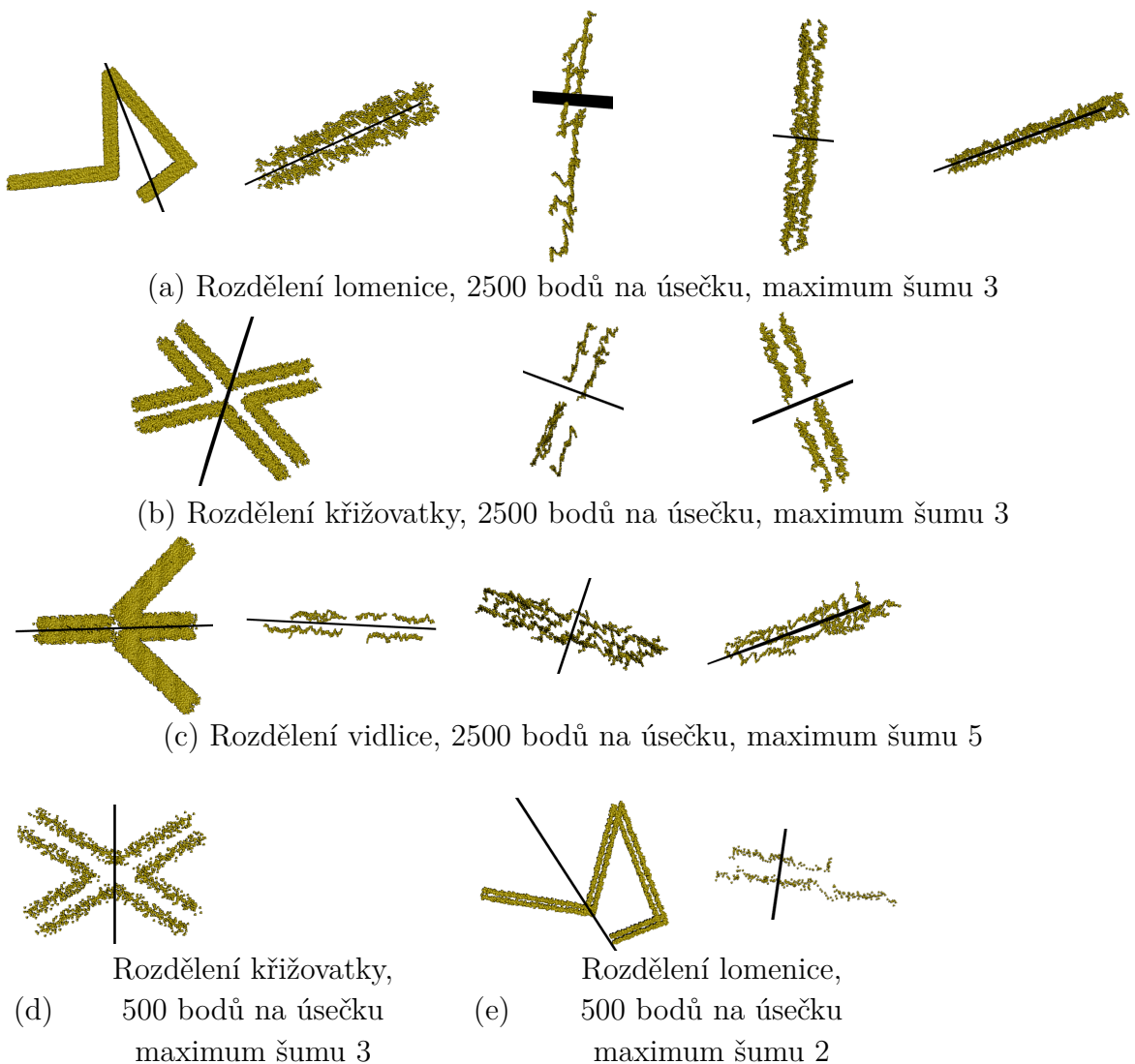
4.2.4 Odolnost proti šumu

Zde se nachází ukázky a zhodnocení rozdělení zašumělých dat. Zavedení šumu se provádí posunutím každého bodu v náhodném směru náhodně daleko, ale ne dál, než je nastavena maximální odchylka. Nastavení maxima je důležité, protože při zcela náhodném šumu, mohou vzniknout data, kde mohou kompletně zaniknout rovnoběžky, mezi nimiž chceme ve výsledku vygenerovat osu. Takový případ dat znázorňuje obrázek 4.22.



Obrázek 4.22: Ukázka výsledku s náhodným šumem bez hranic

Obrázek 4.23 zobrazuje objekt lomenice, křižovatky a vidlice spolu s výsledky jejich rozdělení. U lomenice (a) a křižovatky (b) byli množiny bodů vygenerované s 2500 body na úsečku a s hranicí šumu 3 jednotek, v případě vidlice (c) s 5. Ve všech třech případech bylo rozdělení úspěšné a to i přesto, že v případě lomenice (b) a vidlice (c) splývaly rovnoběžky úseků. Také, ale došlo k výraznému profiltrování bodů. Při snížení počtu bodů na úsečku z 2500 na 500 došlo u křižovatky (d) k odflitrování všech bodů a z lomenice (e) zbyla jen malá část. Je třeba poukázat, že při malé míře šumu se množiny bodů s 500 body na úsek v tabulce 4.15 správně rozdělily.



Obrázek 4.23: Výsledky rozdělení zašumělých dat

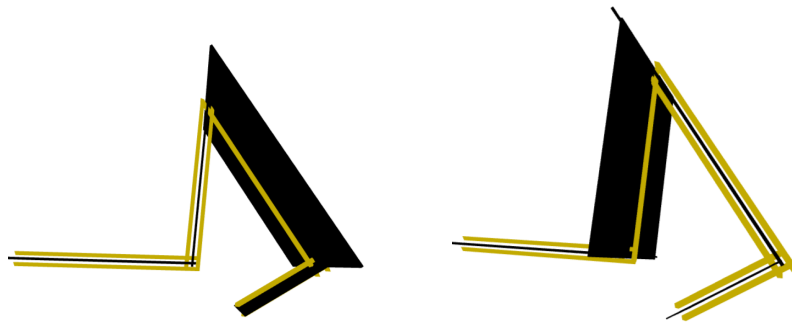
Z výsledků experimentu plyne, že úspěšnost rozdělení zašumělých dat závisí více na hustotě bodů než na velikosti šumu. I při poměrně velkém šumu je metoda rozdělení úspěšná, jsou-li data dostatečně hustá. V opačném případě dochází bohužel nejen k neúspěšnému rozdělení, ale dokonce i až k odstranění všech bodů.

4.3 Výsledky přímého spočtení os symetrie

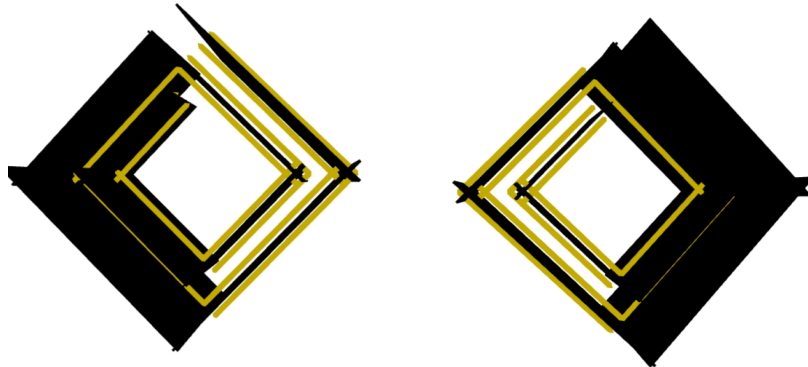
V této sekci se nachází ukázky výsledků spojení spočítaných os jednotlivých úseků množiny bodů. Úseky byly získány metodou rozdělení ze sekce 3.1. Většina výsledků se týká množin bodů, jejichž rozdělení bylo ukázáno v sekci 4.2. Při generování ukázek byl zesílen jas pro lepší odlišení množin bodů a os symetrií. Většina ukázek je zobrazena z více úhlů pohledu, pro lepší reprezentaci výsledků.

Rovnoměrná data bez šumu

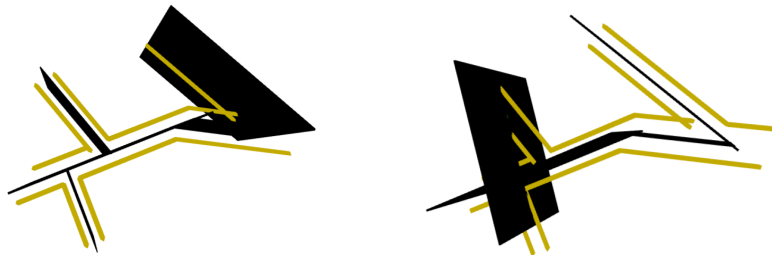
Na obrázku 4.24 jsou ukázky spojení spočtených os jednolitých úseků množin bodů. Body jsou v některých případech seřazené a v jiných zamíchané. V případě lomenice (a), spirály (b) a základního stromu se zamíchanými body (c) jsou výsledky výborné. V případě základního stromu se seřazenými body (d) se dva úseky při rozdělení spojily viz obrázek 4.11 sekce 4.2.1. Vlivem tohoto rozdělení chybí osa jednoho úseku. V případě křížovanky též nastal výborný výsledek jak při zamíchání bodů úseků (f), tak i při seřazení bodů (e). U seřazených bodů (e) přitom došlo k rozdělení druhého úseku na dva, jak bylo ukázáno na obrázku 4.12 sekce 4.2.1.



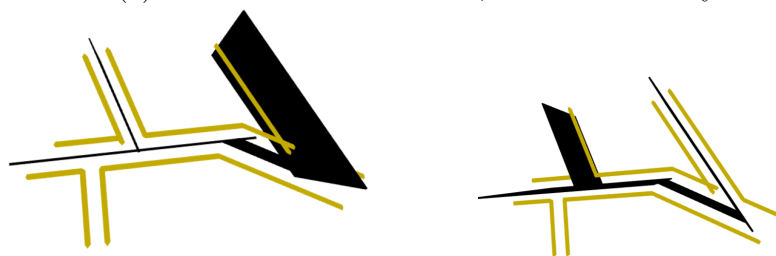
(a) Osa lomenice, zamíchané body



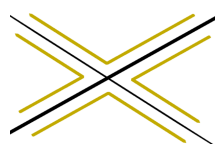
(b) Osa spirály, zamíchané body



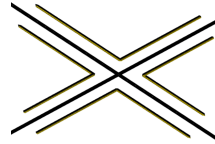
(c) Osa základního stromu, zamíchané body



(d) Osa základního stromu, seřazené body



(e) Křižovatka seřazené body

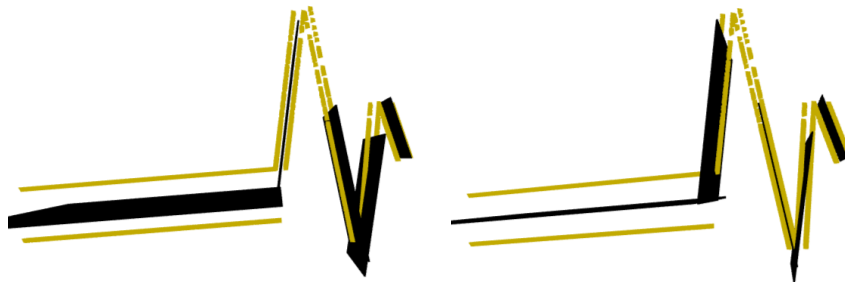


(f) Křižovatka zamíchané body

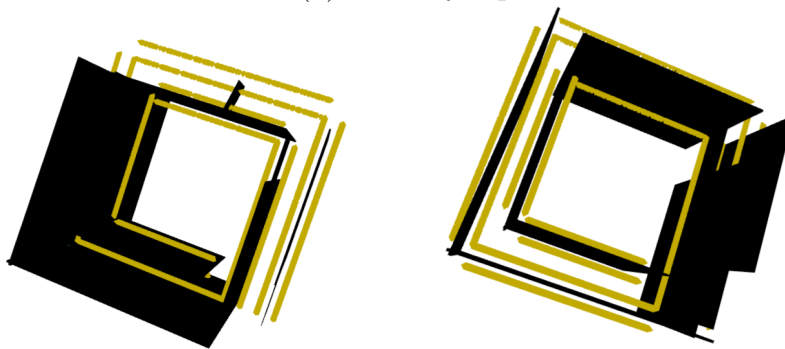
Obrázek 4.24: Výsledky přímého spočtení osy na rovnoměrných datech bez šumu

Nerovnoměrná data

Na obrázku 4.25 jsou vidět ukázky spojení přímo spočítaných os na nerovnoměrných datech. U ořezaného tepu (a) došlo vlivem nerovnoměrnosti k špatnému určení extrémů třetího a čtvrtého úseku, což vedlo ke zkrácení jejich os, které byly ovšem určeny správně. Většina os spirály (b) se spočetla přesně a správně se spojily, s výjimkou jediné, která se naklonila a špatně spojila. Osa základní stromu (c) se našla přesně.



(a) Ořezaný tep



(b) Ořezaná spirála



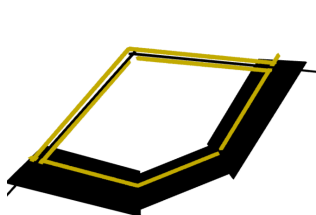
(c) Základní strom

Obrázek 4.25: Výsledky přímého spočtení osy na nerovnoměrných datech

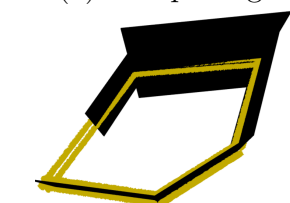
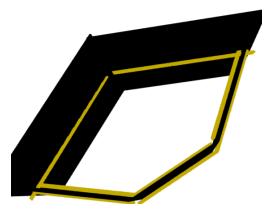
Zašumělá data

Na obrázcích 4.26 a 4.27 jsou zobrazeny ukázky spojení přímo spočítaných os na množinách bodů, na které byl aplikován šum s nastavenou maximální odchylkou, stejně jako v sekci 4.2.4. Je-li zašumění pouze mírné, jsou výsledky výborné, jak je vidět na ukázce pentagonu (a). V případě silněji zašumělých dat jsou výsledky celkem různorodé. Například osa silněji zašumělého pentagonu (b), byla až na zkrácení jednoho úseku nalezena správně. U křižovatky (c) se šumem s maximem 3 došlo k naklonění jedné osy a k jejich špatnému spojení. U vidlice (d) se šumem s maximem 5 došlo k naklonění jedné osy a k jejich špatnému spojení.

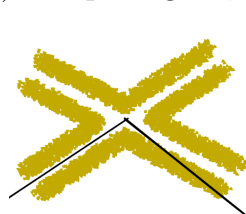
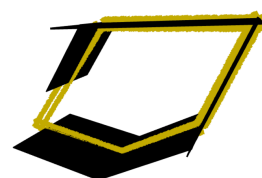
Problémy mohou především nastat, je-li maximum šumu větší než vzdálenost úseček od os. To je dobře vidět na vidlici (d). Ale i přes takový šum se můžeme dostat k dobrému výsledku. To znázorňuje lomenice (e), která má průměrnou vzdálenost úseček od osy 2.5 a při šumu s maximem 3 došlo u většiny úseků jen k mírnému naklonění částí os. Obecně platí, že je-li maximální odchylka šumu stejná nebo trochu menší než vzdálenost od osy, je spočtení os úspěšné, jak znázorňují příklady lomenice (f) a tepu (g).



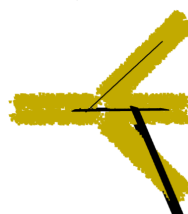
(a) Osa pentagonu, 1250 bodů na úsečku, mírný šum



(b) Osa pentagonu, 2000 bodů na úsečku, maximum šumu 2

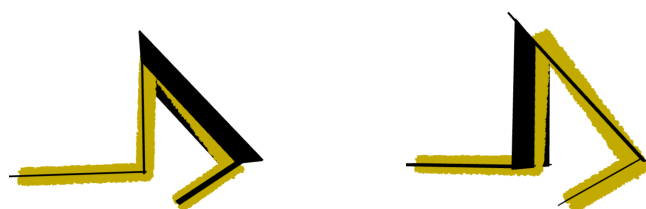


(c) Křižovatka,
(c) 2500 bodů na úsečku,
maximum šumu 3

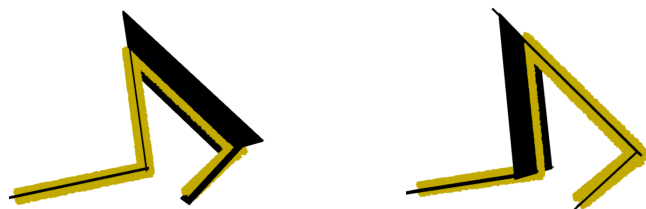


(d) Vidlice,
2500 bodů na úsečku,
maximum šumu 5

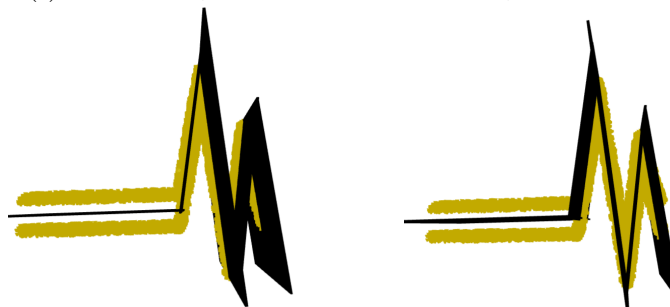
Obrázek 4.26: Výsledky přímého spočtení osy zašumělých dat ((a) až (d))



(e) Lomenice, 2500 bodů na úsečku, maximum šumu 3



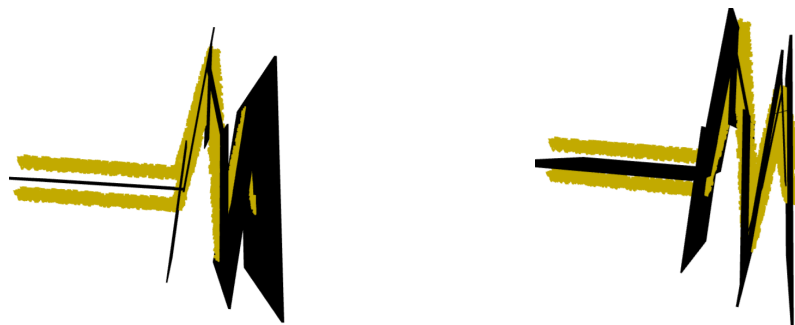
(f) Lomenice, 3000 bodů na úsečku, maximum šumu 2



(g) Osa tepu, 3000 bodů na úsečku, maximum šumu 2

Obrázek 4.27: Výsledky přímého spočtení osy zašumělých dat((e) až (g))

Obrázek 4.28 obsahuje nepovedené výsledky. V případě dat seřazeného tepu, došlo vlivem šumu k rozdělení na větší množství úseků, z nichž následně vznikne hrbolatá osa. Za problémem nestojí jenom seřazenost dat, ale i jejich téměř vertikální uspořádání. To dokazuje případ seřazeného stromu (i), jenž měl dobré výsledky dokonce i při šumu, kdy jednotlivé rovnoběžky splývají (j). Problém s vertikálními úseky znázorňuje lomenice na obrázcích (k) a (l), kdy k nalezení požadované osy symetrie byl třeba větší počet bodů na úsečku.



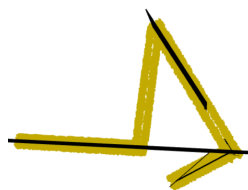
(h) Osa tepu, 2000 seřazených bodů na úsečku, maximum šumu 2



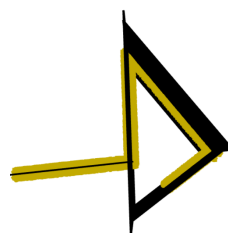
Osa stromu,
(i) 2000 seřazených bodů na úsečku,
maximum šumu 1



Osa stromu,
(j) 2000 seřazených bodů na úsečku,
maximum šumu 2



Osa lomenice,
(k) 2000 bodů na úsečku,
maximum šumu 2



Osa lomenice,
(l) 6000 bodů na úsečku,
maximum šumu 2

Obrázek 4.28: Výsledky přímého spočtení osy zašumělých dat ((h) až (j))

Shrnutí

U rovnoměrných dat bez šumu má spočítání a spojení os velmi dobré výsledky, které může poškodit pouze chybné rozdělení bodů na úseky. V případě nerovnoměrných bodů jsou výsledky přibližně na stejné úrovni. Případné zhoršení vede ke zkrácení os úseků nebo jejich špatnému sklonu. U zašumělých dat závisí výsledek na maximální odchylce šumu a hustotě bodů. Je-li daná odchylka menší nebo rovna vzdálenosti úsečky od osy, jsou výsledky většinou podobné jako u nezašumělých dat, případně dojde ke zhoršení z hlediska spojení os. U zašumělých dat jsou také výsledky horší v případě vertikálních úseků.

5 Závěr

V teoretické části byly popsány a vysvětleny základní pojmy a definice týkající se symetrie. Též bylo představeno několika metod sloužících k detekci osové symetrie. Na konci této části je popis použité metody detekce symetrie vyvinuté na KIV.

V rámci praktické části práce byla naimplementována metoda sloužící k rozdělení množin bodů, které mají podobu dvojic rovnoběžných úseček spojených do lomených čar nebo do stromového větvení. Tato metoda je dostačující pro testovaná vstupní data, ale klade na data několik požadavků souvisejících s počtem bodů, sklonem sousedních úseků a vzdáleností rovnoběžek. Dále bylo naimplementováno zjištění osy symetrie jednotlivých úseků, a to buď pomocí programu pro detekci symetrie nebo přímým spočítáním. Vygenerované osy jsou následně spojeny, aby na sebe co možná nejlépe navazovaly.

Také byla naprogramovaná prostá metoda pro generování množin bodů, na kterých byl program zkoušen.

Užití programu pro detekci symetrie se ukázalo jako nepříliš vhodné, protože nalezená osa symetrie často není ta, kterou potřebujeme. V rámci experimentů se našly parametry, při jejichž splnění má metoda největší šanci pro nalezení požadované osy, ale i při splnění těchto parametrů se očekávaného výsledku většinou nedočkáme. Proto byla naimplementována metoda, která osu symetrie na základě práce s daty spočítá přímo.

Funkčnost vytvořeného rozdělení dat, následného spočítání os jednotlivých úseků a jejich spojení byla otestována v experimentech. Experimenty došly k výsledku, že při velké hustotě bodů a i deformaci dat nebrání rozpoznání jednotlivých úseček funguje metoda dobře. Nevhodné výsledky mohou nastat v případě silně zdeformovaných téměř vertikálních dat.

Vhodným námětem pro další postup by byla možnost rozdělení na části skládající se z křivek. Problematika této práce by se dala také převést na množiny bodů ve 3D.

Literatura

- [1] *Osová souměrnost – Procvičování online – Umíme matiku. Umíme matiku – Zábavné procvičování matematiky online* [online]. Dostupné z: <https://www.umimematiku.cz/cviceni-osova-soumernost>.
- [2] *Středová souměrnost – Procvičování online – Umíme matiku. Umíme matiku – Zábavné procvičování matematiky online* [online]. Dostupné z: <https://www.umimematiku.cz/cviceni-stredova-soumernost>.
- [3] PAOLO DI STEFANO, L. D. A. Bilateral symmetry estimation of human face. In *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 7, s. 217–225, 2013. doi: 10.1007/s12008-012-0174-8.
- [4] BRIBIESCA, E. A measure of tortuosity based on chain coding. In *Pattern Recognit*, 46, s. 716–724, 2013.
- [5] CAILLIÈRE, D. et al. 3D mirror symmetry detection using Hough transform. In *15th IEEE International Conference on Image Processing, ICIP 2008*, San Diego, CA, United States, 2008. doi: 10.1109/ICIP.2008.4712119. Dostupné z: <https://hal.science/hal-01548463>.
- [6] CHANGMING SUN, S. J. 3D symmetry detection using the extended Gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2 1997, 19, 2, s. 164–168. doi: 10.1109/34.574800. Dostupné z: https://www.researchgate.net/publication/3192650_3D_symmetry_detection_using_the_extended_Gaussian_image.
- [7] CHIARA BARTALUCCI, L. G. a. Y. V. R. F. A survey of methods for symmetry detection on 3D high point density models in biomedicine. 2018. Dostupné z: <https://doi.org/10.3390/sym10070263>.
- [8] COLBRY, D. a. G. S. Canonical Face Depth Map: A Robust 3D Representation for Face Verification. *IEEE Conference on Computer Vision and Pattern Recognition*. 2007, s. 1–7. doi: 10.1109/CVPR.2007.383108. Dostupné z: https://www.researchgate.net/publication/224716250_Canonical_Face_Depth_Map_A_Robust_3D_Representation_for_Face_Verification.
- [9] CÉSAR OMAR JIMÉNEZ-IBARRA, H. S.-C. a. M. V.-M. D. C. Axial Symmetry Detection Using AF8 Code. In ARABNIA, L. D. H. S. F. G. T. a. Q.-N. T. H. R. (Ed.) *Advances in Computer Vision and Computational Biology [online]*, s. 143–157, 2021-08-06. Dostupné z: doi:10.1007/978-3-030-71051-4_11.

- [10] E. MICHAELSEN, M. A. D. M. Recognition of symmetry structure by use of Gestalt algebra. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, s. 206–210, 2013. doi: 10.1109/CVPRW.2013.37.
- [11] FREEMAN, H. Techniques for the digital computer analysis of chain-Encoded arbitrary plane curves. In *Proc. Natl. Electronics. Conf.*, 17, s. 421–432, 1961.
- [12] HOHENWARTER, M. M. a. B. et al. *GeoGebra* [online]. Dostupné z: <http://www.geogebra.org>.
- [13] HRUDA, K. I. . V. L. L. Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure. 2020. Dostupné z: <https://doi.org/10.1007/s00371-020-02034-w>.
- [14] HRUDA, K. I. . V. L. L. Plane Space Representation in Context of Mode-Based Symmetry Plane Detection. 2020. doi: https://doi.org/10.1007/978-3-030-50426-7_38.
- [15] J. INESTA, M. S. M. B. Local symmetries of digital contours from their chain codes. In *Pattern Recognition*, 29, s. 1737–1749, 1996.
- [16] M. ALVARADO-GONZALEZ, E. G. C. V. E. B. V. M.-B. W. A. Mirror symmetry detection in curves represented by means of the Slope Chain Code. In *Pattern Recognition*, 87, s. 67–79, 2019. doi: <https://doi.org/10.1016/j.patcog.2018.10.002>.
- [17] 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.
- [18] MOURYCOVÁ, E. Hledání symetrie funkcí s využitím symetrie množiny bodů. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Vedoucí práce Prof. Dr. Ing. Ivana Kolingerová, 2020.
- [19] NIXON, M. S. – AGUADO, A. S. 5 - High-level feature extraction: fixed shape matching. In NIXON, M. S. – AGUADO, A. S. (Ed.) *Feature Extraction and Image Processing for Computer Vision (Fourth Edition)*. Academic Press, fourth edition edition, 2020. s. 223–290. doi: <https://doi.org/10.1016/B978-0-12-814976-8.00005-1>. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9780128149768000051>. ISBN 978-0-12-814976-8.
- [20] PETITJEAN, M. A DEFINITION OF SYMMETRY. *Symmetry: Culture and Science*. 2007, 18 (2-3), s. 99–119. Dostupné z: <https://hal.science/hal-01552499/>.

- [21] ROSEN, J. *Symmetry in science: an introduction to the general theory*. 1995.
- [22] TANG, X. M. – CHEN, J. S. – MOON, Y. S. Accurate 3D face registration based on the symmetry plane analysis on nose regions. In *2008 16th European Signal Processing Conference*, s. 1–5, 2008. doi: 10.3390/sym10070263. Dostupné z: <https://ieeexplore.ieee.org/document/7080771>.

Seznam obrázků

2.1	Osová symetrie [1]	11
2.2	Středová symetrie [2]	11
3.1	Ukázka vstupních objektů	17
3.2	Rozdělení lomené čáry	17
3.3	Zdůvodnění limitu	22
3.4	Spojení os	22
3.5	Spojení os bez kontroly vzdálenosti od opačného konce	22
3.6	Přehled projektu	23
4.1	Požadovaný výsledek	25
4.2	Chybný výsledek	25
4.3	Vzájemně posunuté rovnoběžky	26
4.4	Graf 100 bodů	27
4.5	Graf 250 bodů	27
4.6	Graf 500 bodů	28
4.7	Výsledky testovaných lomených objektů	31
4.8	Výsledky použití přímého spočtení osy na lomených objektech	32
4.9	Výsledky rotace na 180°	33
4.10	Data s nalezenou osou před a po úpravách	34
4.11	Rozdělení dat základního stromu	36
4.12	Rozdělení dat křížovatky	36
4.13	Rozdělení dat vidlice	37
4.14	Výsledek rozdělení zamíchaných úseků s 200 body na úsečku	38
4.15	Výsledek rozdělení zamíchaných úseků s 500 body na úsečku	38
4.16	Detail programem nalezené osy v posledním úseku lomenice	39
4.17	Výsledek rozdělení zamíchaných úseků s 2333 body na úsečku	39
4.18	Výsledek rozdělení dat zamíchaných při 43247 bodech na úsek	40
4.19	Výsledky rozdělení nepravidelné množiny bodů oříznutého tepe, 1000 bodů na úsečku, seřazené	41
4.20	Výsledky rozdělení nepravidelné množiny bodů základního stromu - původně 2500 bodů na úsečku, zamíchané	42
4.21	Výsledky rozdělení nepravidelné množiny bodů spirály - původně 2333 bodů na úsečku, seřazené	42
4.22	Ukázka výsledku s náhodným šumem bez hranic	42
4.23	Výsledky rozdělení zašumělých dat	43
4.24	Výsledky přímého spočtení osy na rovnoměrných datech bez šumu	45
4.25	Výsledky přímého spočtení osy na nerovnoměrných datech	46
4.26	Výsledky přímého spočtení osy zašumělých dat ((a) až (d))	47

4.27	Výsledky přímého spočtení osy zašumělých dat ((e) až (g))	48
4.28	Výsledky přímého spočtení osy zašumělých dat ((h) až (j))	49
A.1	Záložka Axis Function (při použití programu jsou nezaškrtnuté bloky zablokovány, zde povoleny pro lepší čitelnost)	61
A.2	Záložka Noise	62
A.3	Záložka Finish	64
A.4	Okno s vizualizací množiny bodů a os symetrií	65
A.5	Ukázka vyplnění Several Parts Axis Function	66

Seznam tabulek

4.1	Délky úseček ze 100 bodů (r = vzdálenost osy od rovnoběžek) . . .	26
4.2	Délky úseček z 250 bodů (r = vzdálenost osy od rovnoběžek) . . .	26
4.3	Délky úseček z 500 bodů (r = vzdálenost osy od rovnoběžek) . . .	28
4.4	Testované lomené objekty (r = posun hodnoty bodu Y od osy (+-))	30
4.5	Testované stromové objekty (r = posun hodnoty bodu Y od osy (+-))	35

6 Přílohy

Příloha A: Uživatelská příručka

Příloha B: Struktura přiloženého souboru

Kapitola A

Uživatelská příručka

Překlad a spuštění

Program spustíme dvojitým kliknutím na soubor `GUIFormDataGenerator.exe`. Ten se nachází v adresáři `symmetry-detection/GUIFormDataGenerator/bin/x86/Release/`.

Program se dá v adresáři `symmetry-detection/GUIFormDataGenerator` spustit z konzole příkazem `make run`.

Poznámka: první spuštění může trvat dlouho, kvůli kontrole programu antivirem

Obsluha programu

Po spuštění programu se zobrazí okno přepnuté na záložku *Axis Functions* (viz obrázek A.1). Zároveň se otevře systémová konzole, do které budou po spuštění generování nebo detekci symetrie vypisovány informace o běhu programu.

Mezi jednotlivými záložkami se přepíná pomocí myši nebo s klávesami šipek doprava \rightarrow a doleva \leftarrow .

Při generování bodů nebo detekování os symetrií se v levém dolním rohu objevují zprávy informující o průběhu prováděné akce. Ukázka takového výpisu, konkrétně úspěšného vygenerování množiny bodů, se nachází na obrázku A.3 (záložka *Finish*).

Aplikace se skládá celkem z 5 záložek - *Axis Functions*, *Functions*, *Noise*, *Symmetry Detector* a *Finish*. Záložky *Functions* a *Symmetry Detector* pocházejí z práce Elišky Mourycové a v rámci této práce se neupravovaly a nejsou používány. Jsou ovšem funkční a jejich detailní popis se nachází v práci E. Mourycové viz [18]. Stručně je lze popsat následovně:

- *Functions* - umožňuje generování obecných funkcí
- *Symmetry Detector* - lze konfigurovat, kolik os (respektivě ploch) má software pro detekci symetrie detekovat

Záložka Axis Function

V záložce *Axis Functions* (viz obrázek A.1) si lze zvolit jednu ze tří možností vygenerování množiny bodů skládající se z úseků rovnoběžek.

Možnosti zahrnují:

- *Select Axis Function* - nabídka připravených množin bodů, volí se počet bodů na úsečku
- *One Part Axis* - možnost si textově zvolit jednu funkci, kolem které se budou generovat body

Je potřeba nadefinovat minimální a maximální hodnotu x -ových souřadnic, počet bodů na úsečku a posunutí y -ových souřadnic od generované osy. Lze si zvolit možnost zarovnání vygenerovaných rovnoběžek.

- *Several Parts Axis Function* - možnost si nadefinovat až pět vlastních funkcí. Vygenerovaná množina bodů se skládá ze zadaných x -ových souřadnic, y -ové souřadnice jsou posunuty o hodnotu $\pm Y shift$.

Forma zadání definičního oboru x : seznam dvojic reálných čísel (začátky a konce intervalů). Pro psaní desetinných čísel se používá tečka, například 5.5.

Při generování výsledného objektu x -ové souřadnice rovnoběžek jsou stejné jako souřadnice osy (zadané funkce), y -ové souřadnice jsou od původní osy posunuty podle hodnoty $\pm Y shift$.

U možnosti *Several Parts Axis Function* si lze u každé funkce zvolit jednu ze dvou možností na základě zaškrtnutí políčka *define separately bottom and top line*.

– zaškrtnuté:

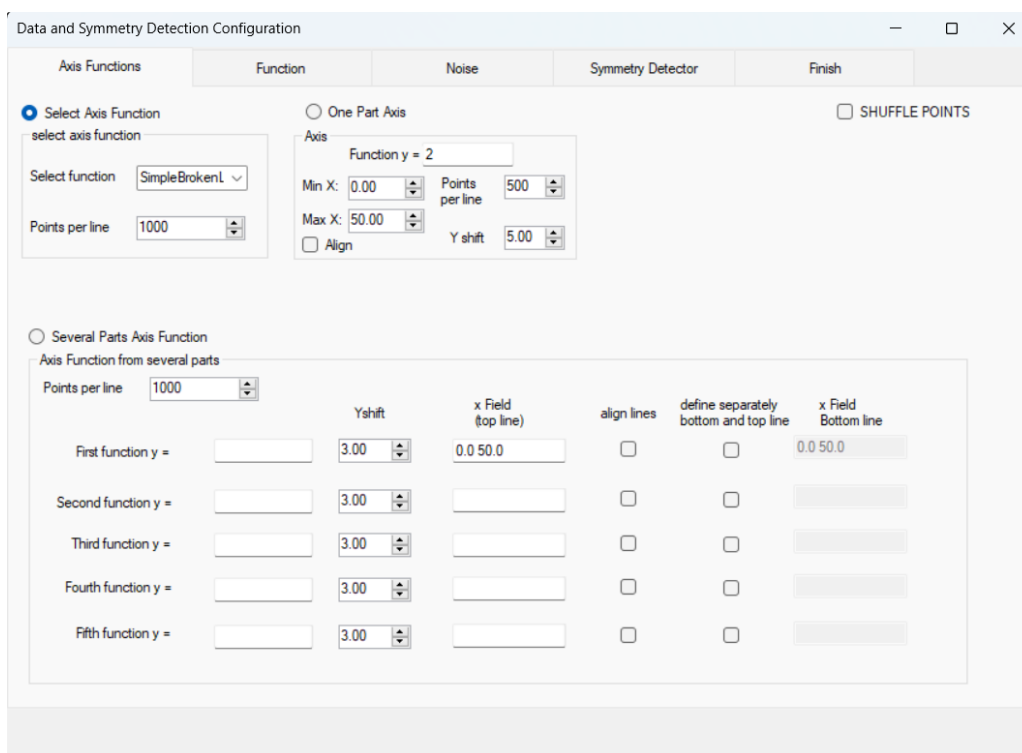
* *x Field Bottom line* - definiční obor úsečky x -ové souřadnice, y -ové souřadnice jsou od osy posunuty o zápornou hodnotu $-Y shift$

* *x Field Top line* - definiční obor úsečky x -ových souřadnice, y -ové souřadnice jsou od osy posunuty o kladnou hodnotu $+Y shift$

– nezaškrtnuté:

* *x Field Bottom line* - definiční obor úsečky x -ových souřadnice, y -ové souřadnice jsou od osy posunuty o zápornou i kladnou hodnotu $\pm Y shift$

* *align lines* - zarovnat/nezarovnat rovnoběžky



Obrázek A.1: Záložka Axis Function (při použití programu jsou nezaškrtnuté bloky zablokovány, zde povoleny pro lepší čitelnost)

Poznámka: při zadávání funkce si je třeba dát pozor na následující chyby:

- nezapomínat psát znak násobení $*$ - místo $0.5x$ je třeba napsat $0.5 * x$
- na místě desetinných míst psát tečku $.$ a ne čárku $,$ - například 5.5

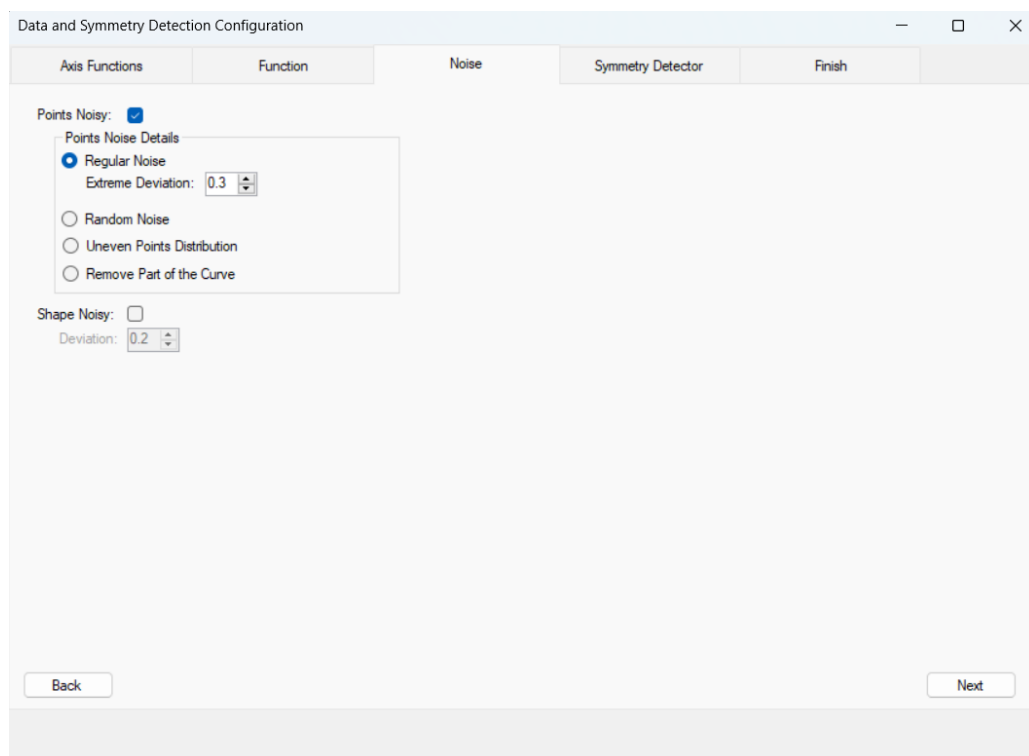
V případě nedodržení formátu je uživatel upozorněn.

Záložka Noise

V záložce *Noise* (viz obrázek A.2) si lze zvolit, zda se do vygenerovaných bodů zanesou šum. První možnost výběru se týká šumu zaneseného do vygenerovaných bodů (*Points Noisy*). Při zaškrtnutí této možnosti je potřeba si následně zvolit jeden z typů šumu.

1. *Regular Noise* - šum s nastavenou maximální odchylkou
2. *Random Noise* - zcela náhodný šum
3. *Uneven Points Distribution* - odstranění náhodného počtu bodů z různých částí vygenerované množiny bodů
4. *Remove Part of the Curve* - z jednoho vybraného intervalu se odstraní všechny body

Druhou možností je zanesení šumu do tvaru křivky (*Shape Noisy*). Tato možnost funguje pouze na funkce z nabídky *Functions*.



Obrázek A.2: Záložka Noise

Záložka Finish

Záložka *Finish* (viz obrázek A.3) obsahuje čtyři bloky.

- *Generate New Point Cloud*

Pomocí tlačítka *Browse* volíme místo k uložení souboru s vygenerované množiny bodů. Tento soubor má příponu `.pc`.

Pomocí tlačítka *Generate!* data pouze vygenerujeme. Tlačítko *Generate, Detect Symmetry and Visualize* k datům navíc detekuje symetrii, která je zapsána do textového souboru. Data a plocha symetrie jsou následně zobrazeny. Symetrie je detekována podle nastavení v záložce *Symmetry Detector*.

- *Select Existing Point Cloud*

Zvolíme soubor s příponou `.pc` obsahující body. K nim se detekuje symetrie, podle nastavení v *Symmetry Detector* a zobrazí se množina bodů spolu s osou. Vygenerovaná osa je zapsána do textového souboru.

- *Divide the Axis Function of Point Cloud*

Data ze souboru s příponou `.pc` se rozdělí na jednotlivé úseky. Jsou očekávaná data skládající se z úseků rovnoběžek.

Rozdělené úseky jsou zapsány do souboru se jménem původního souboru k němuž je přidáno `_split_i.pc`, kde `i` je číslo úseku.

- *Find Generalized Mirror Symmetry of Axis Function*

Data ze souboru s navazujícími rovnoběžnými úseky (přípona `.pc`) se rozdělí na úseky. Ke každé úseku se detekuje/spočítá osa a ty jsou následně spojeny.

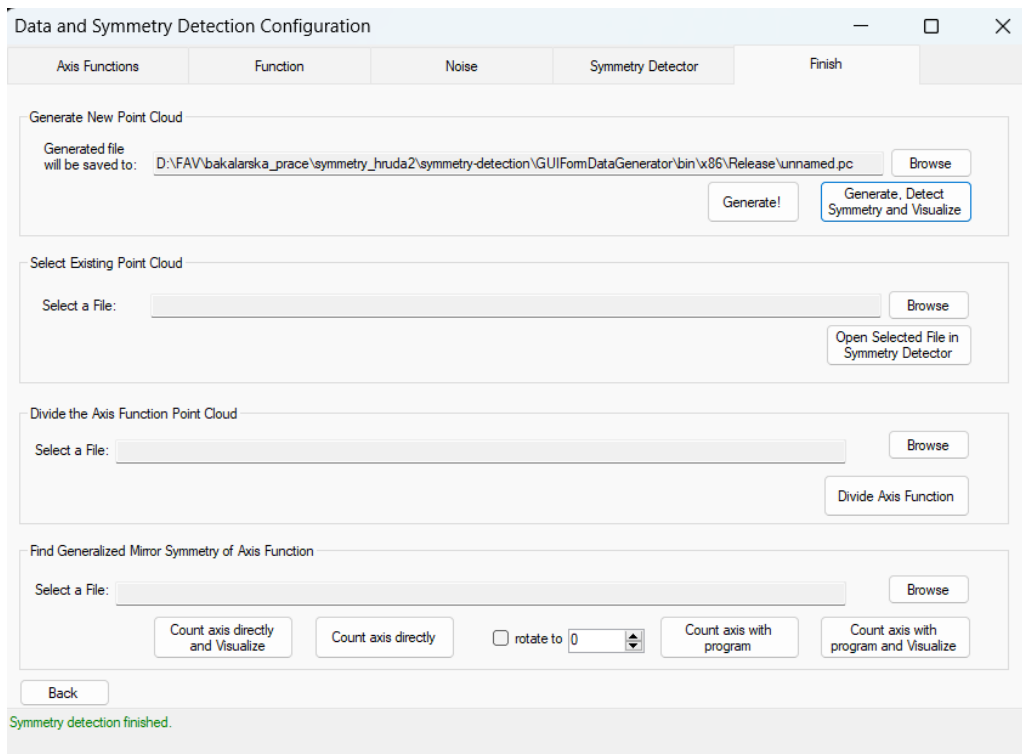
Můžeme si zvolit najít osy buď přímo nebo s užitím programu pro detekci symetrie. V případě užití programu si můžeme zvolit úhel, na který budou úseky při detekci symetrie natočeny.

Výsledné osy jsou zapsány do souboru, k jehož jménu je přidáno `_axis-directlyCount` nebo `_axis_programDetect`. Přípona souboru je `.txt`

Zvolený výsledek si můžeme zvizualizovat.

K zobrazování úseků se používá vizualizační knihovna SlimDX.

Pozn.: Tato knihovna někdy způsobí pád programu, obvykle při opakovaném použití. Problém není konzistentní a jeho příčinu se nepovedlo odhalit. Proto problém nebyl odstraněn.



Obrázek A.3: Záložka Finish

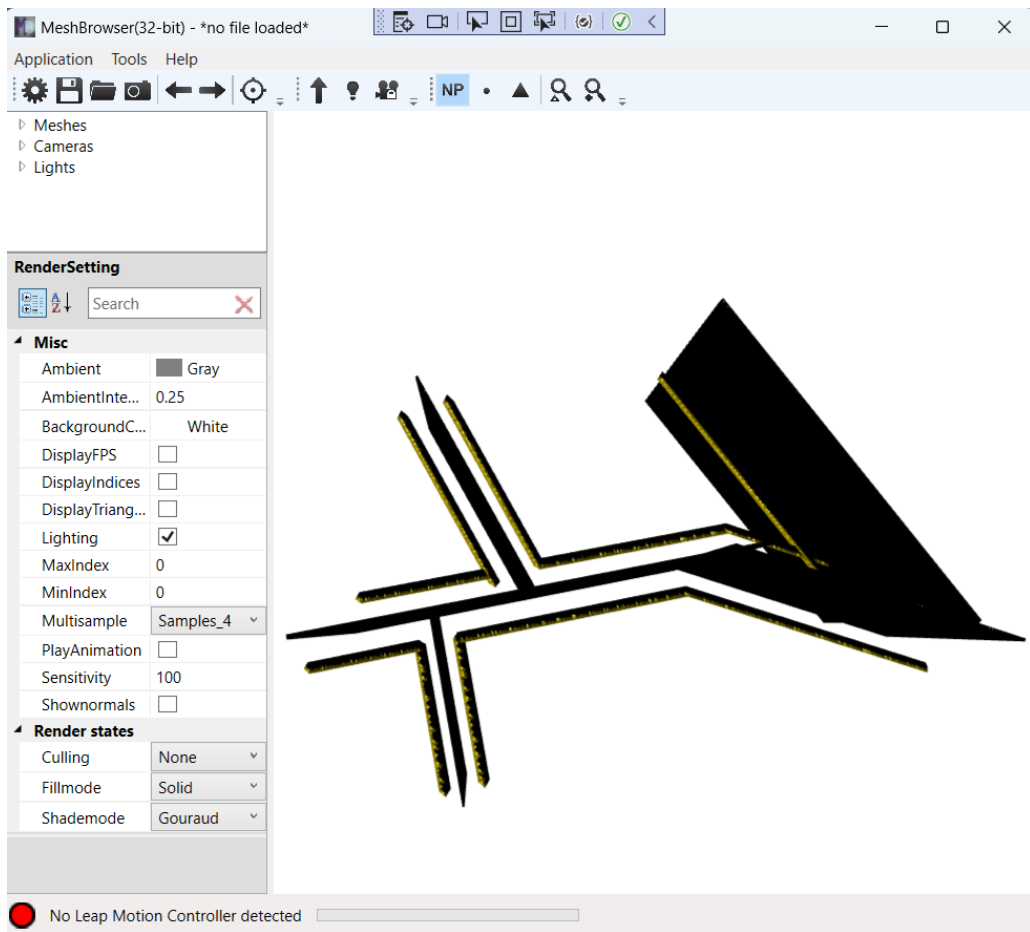
Ovládání vizualizačního nástroje

Při požadavku zobrazení množiny bodů a rovin symetrií se otevře nové okno (viz obrázek A.4).

Držením levého tlačítka myši a tažením lze objekt otáčet a držením pravého tlačítka myši a tažením se objektem posouvá.

Stisknutím klávesy X se přepíná mezi osvětlením se stínem a bez stínu (zcela jasné barvy objektu). Pomocí klávesnice se dá ovládat několik funkcí vizualizačního okna. Ty neužitečnější jsou klávesy ovládající kameru (pohled):

- S - oddálení
- D - posun doprava
- T - rychlé oddálení
- A - posun doleva
- W - přiblížení
- Q - posun dolů
- R - rychlé přiblížení
- E - posun nahoru



Obrázek A.4: Okno s vizualizací množiny bodů a os symetrií

Příklady použití

Vygenerování a zobrazení nadefinovaného objektu

1. V záložce *AxisFunction* zaškrtneme políčko *Select Axis Function*
2. Z nabídky *Select function* se zvolíme jednu možnost - například *Spiral*
3. Přepneme program do záložky *Finish*
4. V boxu *Generate New Point Cloud* stiskneme tlačítko *Browse*
5. Zvolíme kam chceme soubor umístit a jeho pojmenování
6. Stiskneme tlačítko: *Generate, Detect Symmetry and Visualize* – detekovaná osa se zapíše do textového souboru a objekt se spolu s osou symetrie zobrazí ve vizualizačním nástroji

Vygenerování vlastního osového objektu se šumem

1. V záložce *AxisFunction* zaškrtneme políčko *Several Parts Axis Function*
2. Vyplníme předpisy funkce a jejich definiční obory – nadefinujeme svůj objekt. Příklad vyplnění je obrázek A.5:

Points per line	Yshift	x Field (top line)	align lines	define separately bottom and top line	x Field Bottom line
1000	3.00	0.0 6.667 9.3 20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0.0 5.5 7.5 20
	3.00	20 28.4 33.3 40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	20 40
	2.00	10 33.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	9 28.4
	2.00	5.5 15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	3.00		<input type="checkbox"/>	<input type="checkbox"/>	

Obrázek A.5: Ukázka vyplnění *Several Parts Axis Function*

3. Přejdeme do záložky *Noise*
4. Zaškrtneme políčko *Points Noisy*
5. Uvnitř boxu *Points Noise Details* zaškrtneme políčko *Regular Noise* a nastavíme hodnotu *Extreme Deviation* například na hodnotu 1.00

6. Přejdeme do záložky *Finish*
7. V boxu *Generate New Point Cloud* stiskneme tlačítko *Browse*
8. Zvolíme kam chceme soubor umístit a jeho pojmenování
9. Stisknutím tlačítka *Generate* se soubor vygeneruje

Zobrazení již vygenerovaného souboru

1. Přepneme program do záložky *Finish*
2. V boxu *Generate New Point Cloud* stiskneme tlačítko *Browse*
3. Zvolíme si soubor s příponou *.pc*, který chceme zobrazit
4. Stiskneme tlačítko *Open Selected File in a Symmetry Detector* a chvíli počkáme na detekování osy a její zobrazení s objektem ve vizualizačním nástroji. Koeficienty detekované osy symetrie se zapíší do textového souboru.

Rozdělení osového objektu

1. Přepneme program do záložky *Finish*
2. V boxu *Divide the Axis Function Point Cloud* stiskneme tlačítko *Browse*
3. Zvolíme *.pc* soubor s obecným objektem, který chceme rozdělit na části
4. Stisknutím tlačítka *Divide Axis Function* se rozdělí vstupní množina bodů a každá nalezená část je zapsána do vlastního souboru

Spočtení zobecněné osy zrcadlové symetrie

1. Přepneme program do záložky *Finish*
2. V boxu *Find Generalized Mirror Symmetry of Axis Function* stiskneme tlačítko *Browse*
3. Zvolíme *.pc* soubor s osovým objektem, jehož obecnou zrcadlovou osu symetrie chceme najít
4. Použijeme tlačítko *Count axis directly and Visualize* – spočítá se osa symetrie, která se spolu se vstupní množinou bodů zobrazí ve vizualizačním nástroji. Spočtená osa (její části) se zapíší do textového souboru.

Kapitola B

Struktura přiloženého souboru

- Aplikace_a_knihovny:
 - symmetry-detection - samostatná aplikace, obsahující zdrojové kódy a potřebné knihovny, popsané v sekci Technické řešení v textu práce
 - * adresáře/projekty: GUIFormDataGenerator a FunctionsDataGenerator - zdrojové kódy obsahující vytvořenou práci
 - * adresáře/projekty: Client, FindSymmetry, Framework, GUI, SlimDXRenderer, SlimDXRendererSystem, Visicounter - projekty vizualizačního nástroje a programu pro detekci symetrie
 - * adresáře: Objects, packages, symmetry-coloring - balíčky/knihovny používané vizualizačním nástrojem
 - testovací_data - data, která je možné použít k otestování aplikace
 - uzivatelska_prirucka.pdf - příručka s popisem ovládání aplikace
- Text_prace:
 - Vondrovic_BP.pdf - text bakalářské práce
 - tex - obsahuje LaTeX soubory sloužící k vygenerování textu práce
- Readme.txt - stručný popis obsahu souboru