

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

Bakalářská práce

Hledání symetrie funkcí s využitím symetrie množiny bodů

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Eliška MOURYCOVÁ**
Osobní číslo: **A17B0297P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informatika**
Téma práce: **Hledání symetrie funkcí s využitím symetrie množiny bodů**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte metodu hledání symetrie pro množiny bodů, vyvinutou na KIV, a seznamte se s její implementací.
2. Aplikujte tuto metodu na geometrické tvary popsané funkcí podle výběru vedoucí práce.
3. Na základě výsledků navrhnete potřebné doplňky a úpravy a implementujte je.
4. Vyzkoušejte doplněnou metodu pro různé funkce a různé parametry.
5. Zhodnoťte dosažené výsledky.

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: **7. října 2019**
Termín odevzdání bakalářské práce: **7. května 2020**



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 6. května 2020

Eliška Mourycová

Poděkování

Děkuji Prof. Dr. Ing. Kolingerové za odborné vedení a pomoc při zpracování bakalářské práce. Rovněž bych chtěla poděkovat doc. RNDr. Lávičkovi, Ph.D. a Ing. Hrudovi za vstřícnost a pomoc při získání potřebných informací a podkladů.

Abstract

This bachelor thesis „Searching of Function Symmetry Using Point Sets Symmetry“ deals with the symmetry of implicit functions. The aim was to generate a set of points from the function equation and to determine the axial, rotational and central symmetry of the given curve using the existing software for axial symmetry detection. The evaluation was based on experiments carried out with various functions on complete or somehow corrupted data. The results show better success than expected even with slight deformation of the shape of the curve.

Abstrakt

Tato bakalářská práce „Hledání symetrie funkcí s využitím symetrie množiny bodů“ se zabývá symetrií implicitně zadaných funkcí. Cílem bylo generovat množinu bodů z rovnice funkce a s využitím existujícího softwaru pro detekci osové symetrie určit osovou, rotační a středovou symetrii dané křivky. Vyhodnocování probíhalo na základě experimentů prováděnými s různými funkcemi při úplných i nějakým způsobem porušených datech. Výsledky ukazují až překvapivé úspěchy i při mírné deformaci tvaru zkoumané křivky.

Obsah

1	Úvod	8
2	Analytická část	9
2.1	Známá metoda rasterizace křivky	9
2.1.1	Funkce	9
2.1.2	Rasterizace křivky	9
2.2	Symetrie	11
2.2.1	Teorie grup	11
2.2.2	Bijekce	12
2.2.3	Transformace	12
2.2.4	Stavová ekvivalence	13
2.2.5	Transformace symetrie, grupa symetrie	14
2.2.6	Osová, rotační a středová symetrie	15
2.2.7	Symetrie v informatice	16
3	Řešení	17
3.1	Technické řešení	17
3.2	Generování množiny bodů	18
3.3	Metoda detekce osově symetrie	21
3.4	Detekce rotační a středové symetrie	21
4	Experimenty	23
4.1	Detekce jedné osy symetrie	23
4.2	Detekce více os symetrie	25
4.3	Šum a porušené křivky	26
4.4	Rotační a středová symetrie	36
4.5	Symetrie pro přibližně symetrické funkce	37
5	Závěr	40
	Literatura	
6	Přílohy	

1 Úvod

Symetrie objektů je pro celou řadu problémů potenciálně velmi užitečná vlastnost. Problém symetrie řeší výzkum v oblasti informatiky i v oblasti matematiky. V těchto dvou odvětvích existují určité rozdíly v definici konkrétních úkolů a vstupních dat. Informatika obvykle zpracovává množiny bodů nebo trojúhelníkové sítě, zatímco matematika se zajímá o objekty popsané rovnicemi. Informatika obvykle hledá symetrii konkrétního objektu, zatímco matematika se obvykle snaží najít obecnější odpovědi pro obecnější sady objektů. Výzkum v obou oblastech je oddělen, ale kombinace přístupů obou stran by mohla přinést výhody a nové metody zajímavé pro obě výzkumné komunity.[7]

Tato bakalářská práce se zabývá hledáním rotační a středové symetrie funkcí s využitím metody hledání osové symetrie pro množiny bodů, vyvinutou na KIV. Jedná se o společný výzkum s KMA především v oblasti symetrií implicitních funkcí. Implicitní zadání funkcí bylo jedním z požadavků.

V teoretické části práce je vysvětlena základní myšlenka známé metody rasterizace implicitně zadané křivky[6], dále je představena formální definice symetrie[10].

V další části práce je popsáno řešení generování množiny bodů, využitá metoda detekce osové symetrie a detekce rotační a středové symetrie.

Pomocí experimentů je ověřena funkčnost metody detekce všech výše zmíněných druhů symetrie a na závěr jsou prezentovány a zhodnoceny dosažené výsledky.

2 Analytická část

V této kapitole bude popsána známá metoda rasterizace křivky (tj. implicitně zadané funkce) v sekci 2.1 a potřebné základy teorie symetrie v sekci 2.2.

2.1 Známá metoda rasterizace křivky

Tato sekce popisuje ideu známé metody rasterizace křivky definované implicitně zadanou funkcí dvou reálných proměnných.

2.1.1 Funkce

Řekneme, že funkčním předpisem $z = f(x, y)$ je určena reálná funkce dvou reálných proměnných, jestliže:

1. Je dán obor $B \subset \mathbb{E}_2$, přípustných bodů z \mathbb{E}_2 , nazývaných definičním oborem značeným $D(f)$.
2. Každému bodu $X = [x, y] \in B$ je přiřazeno právě jedno reálné číslo $z \in E_1$ takové, že $z = f(x, y)$. [5]

Všechny funkce budou zadávány implicitně, tedy rovnice $f(x, y) = 0$ bude určovat konečnou množinu

$$M = \{[x, y] \in \mathbb{E}_2 \mid f(x, y) = 0\}.$$

Budeme předpokládat, že tato množina je křivka a ne např. kus plochy [2]. Dále budeme předpokládat, že $D(f) = \mathbb{E}_2$.

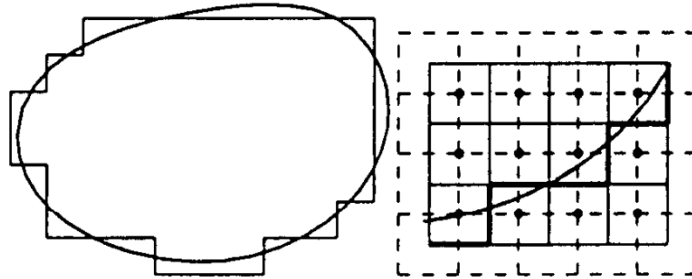
Častým typem funkcí, se kterými budeme pracovat, budou algebraické funkce.

Algebraickou funkcí budeme rozumět takovou funkci, z jejíhož vyjádření $y = f(x)$ lze odvodit rovnost typu $P(x, y) = 0$, kde $P(x, y)$ je polynomem v neurčitých x, y , jehož stupeň se nazývá stupněm funkce f . [1]

2.1.2 Rasterizace křivky

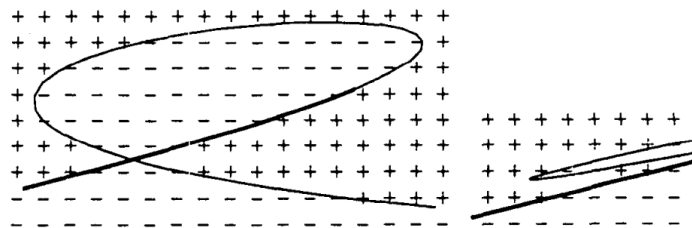
Jedním z problémů, které bylo potřeba řešit, byl problém generování množiny bodů z implicitně zadané funkce. Základní idea řešení (viz podsekcí 3.2 na str. 19) je převzata z [6].

Autor v [6] v kapitole 1 představuje algoritmus kanonické rasterizace křivky. Kanonickou rasterizaci křivky v podstatě definuje jako aproximaci dané křivky, která je omezena na čtvercovou mřížku vhodnou pro vymezení množin pixelů. Kanonická rasterizace uzavřené křivky znázorněné na obrázku 2.1 tedy ohraničuje množinu pixelů, jejichž střed leží v oblasti ohraničené křivkou.



Obrázek 2.1: Uzavřená křivka a její kanonická rasterizace

Proces kanonické rasterizace je nalezení lomené čáry, která rozděljuje množinu bodů (středů pixelů) na ty, které leží na jedné straně křivky, a na ty, které leží na druhé straně. Pro test pozice bodu se hodí sledovat znaménko funkční hodnoty po jeho dosazení do funkčního předpisu. Takový test ale není vždy dostatečný - viz obr. 2.2. Pro křivku na obr. 2.2 platí, že funkční hodnoty bodů jsou nejdříve kladné nad ní a záporné pod ní, ale tento vztah se obrátí po tom, co křivka protne sama sebe.



Obrázek 2.2: Kubické křivky a znaménka funkčních hodnot

Přestože smyslem generování křivky popsané rovnicí (pro naše účely) není hledání lomené čáry, ale pouze množiny bodů, tato základní idea a znalosti jsou využity při implementaci vlastního řešení tohoto problému.

2.2 Symetrie

Tato sekce se zabývá teorií symetrie. Prostřednictvím teorie grup je představena formální definice symetrie.

Většina informací obsažených v této kapitole je převzata z [10], kapitoly 2 a 4.

Symetrie může být slovně definována jako *odolnost vůči možné změně*. Z této věty vidíme dvě základní komponenty symetrie:

1. *Možnost změny*. Musí být možné změnu provést (ačkoliv změna nemusí být provedena).
2. *Odolnost*. Nějaká stránka situace zůstane po provedení změny stejná.

2.2.1 Teorie grup

Teorie grup je matematická disciplína, která se symetriemi úzce souvisí. Grupa G je taková množina, která splňuje následující čtyři podmínky:

1. *Uzavřenost*. Pro všechny prvky $a, b \in G$ platí

$$ab, ba \in G$$

2. *Asociativita*. Pro všechny prvky $a, b, c \in G$ platí

$$a(bc) = (ab)c$$

3. *Existence neutrálního prvku*. G obsahuje prvek e , který se nazývá *neutrální prvek*, pro který platí

$$ae = ea = a.$$

Lze dokázat, že prvek e je vždy unikátní.

4. *Existence inverzního prvku*. Pro každý prvek $a \in G$ existuje prvek a^{-1} značící *inverzní prvek*, pro který platí

$$aa^{-1} = a^{-1}a = e$$

Obecně platí, že $ab \neq ba$.

Podgrupa

Podgrupa je podmnožina grupy, která je také grupa (splňuje podmínky uvedené v podsekcí 2.2.1). Každá grupa G má alespoň dvě podgrupy:

1. grupu obsahující pouze neutrální prvek,
2. celou grupu G .

2.2.2 Bijekce

Bijekce je zobrazení (mapování), které je prosté (anglicky *one-to-one*) a na, tzn. každý prvek jedné množiny se zobrazuje na právě jeden prvek množiny obrazů. Zobrazení je invertibilní (lze pro něj nalézt inverzní zobrazení) právě tehdy, když je bijekcí - toto bude později důležité.

2.2.3 Transformace

Tato podsekcce je věnována transformacím systému. *Systém* je to, co zkoumáme (v našem případě to budou body funkcí).

Stav systému je možná forma systému.

Stavový prostor systému je množina všech možných stavů systému (v našem případě tedy množina všech možných stavů křivky, např. posunutí, změna tvaru, škálování, atp.).

Transformace systému je mapování 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

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

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

Kompozice transformací je mapování stavového prostoru do sebe, je tedy také transformace, tudíž množina transformací systému je *uzavřená*. Předpokládáme, že kompozice transformací působí ve stejném stavovém prostoru.

Kompozice transformací je provedena postupnou aplikací jednotlivých transformací, tzn. kompozice dvou transformací je definována jako výsledek aplikace jedné transformace na výsledek jiné. 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 :

$$u \xrightarrow{T_1} v \xrightarrow{T_2} w \quad \text{nebo} \quad w = T_2(v) = T_2(T_1(u)) = (T_2T_1)(u).$$

Lze ukázat, že kompozice transformací postupnou aplikací je *asociativní*, tzn.:

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

Transformace I taková, pro kterou platí

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

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:

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

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*.

2.2.4 Stavová ekvivalence

Nyní definujeme relaci ekvivalence pro stavový prostor systému. Je to taková relace, kterou značíme \equiv a která nastává mezi dvěma stavy, pokud platí následující tři podmínky:

1. *Reflexivita*. Pro všechny stavy u platí

$$u \equiv u$$

2. *Symetrie*. Pro všechny stavy u, v platí

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

3. *Tranzitivita*. Pro všechny stavy u, v, w platí

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

Podmnožina stavového prostoru systému se nazývá *podprostor*. Pokud všechny stavy tvořící tento podprostor jsou navzájem ekvivalentní a žádný jiný stav z původní množiny není ekvivalentní žádnému stavu v podprostoru, nazýváme tento podprostor *třída ekvivalence*, resp. v našem případě *podprostor ekvivalence*.

2.2.5 Transformace symetrie, grupa symetrie

V sekci 2.2 byla symetrie definována jako *odolnost vůči možné změně*. S nyní nabytými znalostmi toto můžeme rozvinout: Symetrie znamená to, že jakékoli nahrazení stavů stavu, které jsou jim ekvivalentní, ponechává nezměněné členství v příslušném podprostoru. Jinými slovy: Pokud relace ekvivalence není triviální (tj. každý stav je ekvivalentní pouze sám se sebou), je vždy možné provést netriviální změnu (změnu stavů), která stavy nahradí pouze ekvivalentními stavy. Změna se děje pouze v rámci jednotlivých podprostorů ekvivalence, tedy náležitost stavu určitému podprostoru je odolná vůči této změně. Symetrie je tedy vyjádřena jako odolnost vůči změně náležitosti třídy ekvivalence při možné změně stavů.

Dekompozice stavového prostoru do netriviálních podprostorů ekvivalence reprezentuje ten aspekt situace, který zůstává nezměněn při možných změnách stavů. Změny stavů jsou reprezentovány transformacemi. Takže symetrie je reprezentována existencí transformací, které ponechávají podprostory ekvivalence invariantní, tj. transformace, které mapují každý stav na obraz (také stav), který je ekvivalentní původnímu. Takové transformace se nazývají *transformace symetrie* pro relaci ekvivalence. Definující vlastnost transformace symetrie S je:

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

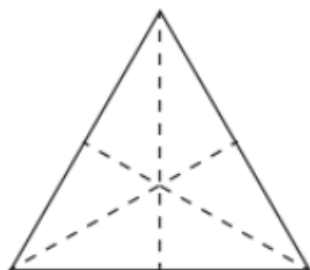
pro všechny stavy u .

Množina všech invertibilních (tj. lze k nim nalézt inverzní transformaci) transformací symetrie stavového prostoru systému pro relaci ekvivalence tvoří grupu, která je podgrupou transformační grupy (viz podsekcí 2.2.3), a nazývá se *grupa symetrie*.

Pokud žádné dva stavy stavového prostoru systému nejsou ekvivalentní, tj. každý stav je ekvivalentní pouze sám sobě, potom grupa symetrie obsahuje pouze neutrální transformaci a systém je *asymetrický*. V takovém případě, ačkoli jsou změny možné, neexistuje žádná, která by ponechala nějaký aspekt situace nezměněný.

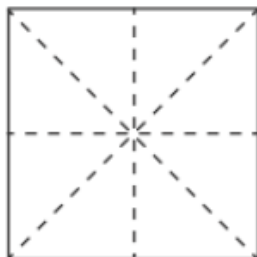
Příklad

Vezměme za příklad rovnostranný trojúhelník ve 2D (viz obr. 2.3). Grupa symetrie tohoto trojúhelníku obsahuje tyto transformace: neutrální transformaci (tj. „žádná změna“), transformace rotace okolo těžiště (na obrázku bod, ve kterém se protínají vyznačené osy) o 120° a 240° (resp. i 360°) a zrcadlení přes vyznačené osy.



Obrázek 2.3: Rovnostranný trojúhelník[3]

V případě čtverce na obr. 2.4 grupa symetrie obsahuje neutrální transformaci, transformace rotace okolo těžiště (na obrázku bod, ve kterém se protínají vyznačené osy) o 90° , 180° a 270° (resp. i 360°) a zrcadlení přes vyznačené osy.



Obrázek 2.4: Čtverec[3]

2.2.6 Osová, rotační a středová symetrie

Zde jsou popsány druhy symetrie, se kterými v rámci práce pracujeme, a jsou vztažené k definici symetrie tak, jak byla výše uvedena.

Jak už bylo zmíněno, *osová symetrie* znamená, že při zrcadlení jednotlivých bodů objektu (křivky) se každý jeho bod namapuje na bod, který také náleží objektu. Zrcadlení je tedy transformace translace všech bodů A náležících křivce takto:

$$A' = A + \vec{v},$$

kde A' je obraz zrcadleného bodu a \vec{v} je vektor vedoucí z bodu A kolmo k (podezřelé) ose symetrie. Pokud všechny takto nalezené body A' náleží křivce, potom daná transformace náleží grupě symetrie této křivky.

Rotační symetrie je nalezena pomocí transformace rotace okolo bodu, ve kterém se protínají osy symetrie, tzn. aby hledání rotační symetrie mělo smysl, je potřeba, aby daný objekt měl alespoň dvě osy symetrie (v případě, že křivka má pouze jednu nebo žádnou osu symetrie, můžeme rovnou říci, že

transformace rotace o jiný úhel než celočíselný násobek 360° nenáleží grupě symetrie křivky). Transformaci rotace opět aplikujeme na všechny body A náležící křivce a sledujeme, jestli jejich obraz také náleží křivce. Pokud ano, transformace rotace o daný úhel náleží grupě symetrie křivky.

Středová symetrie je určena podobně jako osová symetrie, tedy aplikujeme transformaci translace na všechny body A náležící křivce takto:

$$A' = A + \vec{w},$$

ovšem zde je \vec{w} vektor vedoucí z bodu A do bodu, kde se protínají osy symetrie. Opět se sleduje, jestli všechny takto nalezené body A' náleží křivce. Pokud ano, potom křivka má středovou symetrii. Středová symetrie je ekvivalentní rotaci o 180° .

Například trojúhelník na obr. 2.3 středovou symetrii nemá, ale má tři osy symetrie a rotační symetrii s nejmenším úhlem rotace o 120° .

Čtverec na obr. 2.4 má středovou symetrii, čtyři osy symetrie a rotační symetrii s nejmenším úhlem rotace o 90° .

Řád rotační symetrie

Řádem rotační symetrie[4] budeme nazývat číslo, které udává počet transformací rotace o různé úhly α náležící grupě symetrie křivky, takové, že $0 < \alpha \leq 360$.

2.2.7 Symetrie v informatice

V počítačové grafice má informace o symetrii v objektech různé aplikace, jako je zarovnání objektů do souřadného systému, komprese, symetrické úpravy nebo rekonstrukce neúplných objektů. [8]

Zarovnání objektů do souřadného systému je důležitým krokem předzpracování pro různé úkoly, např. vizualizace nebo skládání scén. K provedení tohoto zarovnání musí být pro počátek vybrán globální referenční bod a pro určení orientace musí být vybrána sada os. V zarovnání vůči osám a počátku často čekáme symetrii.[9]

Odstranění těch ploch z trojúhelníkových sítí, které se se opakují v redundantních oblastech symetrie, lze využít pro ukládání a efektivnější zpracování těchto sítí. Opačným postupem lze také rekonstruovat poškozenou síť.[11]

3 Řešení

Tato kapitola popisuje celkové řešení práce. V první části se zabývá přípravou vstupních dat (tj. získání množiny bodů z implicitně zadané funkce, popř. polygonu). V další části popisuje využívanou metodu výpočtu osy (resp. roviny) symetrie a nakonec vysvětluje metodu detekce složitějších symetrií (rotační a středové).

3.1 Technické řešení

Řešení je naprogramované v jazyce .NET/C#, verze .NET 4.6.1 a je sestavené z několika projektů, z nichž dva byly programovány v rámci této práce a ostatní byly poskytnuty Ing. Hrudou. Úplný seznam projektů je následující:

- Client,
- Framework,
- **FunctionsDataGenerator**,
- GUI,
- **GUIFormDataGenerator**,
- SlimDXRenderer,
- SlimDXRendererSystem,
- Visicounter.

Tučným písmem jsou zvýrazněny projekty, které byly vytvořeny v rámci této práce. Projekt **FunctionsDataGenerator** obsahuje třídy:

- DataGenerator,
- Functions,
- Point,
- SymmetryDetectorCR.

Třída DataGenerator se stará o generování množiny bodů (popsáno v sekci 3.2) a zanášení šumu do bodů nebo tvaru polygonu (jednotlivé typy popsané v kapitole 4). Třída Functions vrací funkční hodnoty funkcí pro zadaná x

a y , třída `Point` reprezentuje bod. Třída `SymmetryDetectorCR` obsluhuje detekci rotační a středové symetrie (popsáno v sekci 3.4).

Projekt **GUIFormDataGenerator** se stará o grafické uživatelské rozhraní.

Projekt `Framework` se stará o detekci osové symetrie (popsáno v sekci 3.3). Spouštění této metody je řešeno pomocí projektu `Client`. Tento projekt je také využitý pro integraci detekce osové symetrie do této práce. V třídě `Program` projektu `Client` je volána metoda detekce rotační a středové symetrie. Po domluvě s vedoucí práce není pro odevzdání ponechán zdrojový kód projektu `Client`, ale je použitý ve formě knihovny (DLL).

Projekty `SlimDXRenderer`, `SlimDXRendererSystem` a `Visicounter` se starají o vizualizaci zadaného objektu a jeho rovin symetrie.

3.2 Generování množiny bodů

V této sekci jsou popsány postupy generování množiny bodů z různých typů funkcí, jedná se tedy o přípravu vstupních dat, která jsou následně poskytnuta metodě pro detekci symetrie.

Generování polygonů

Při generování množiny bodů polygonu se nejdříve najdou souřadnice vrcholů. To probíhá tak, že se vypočítá úhel mezi dvěma stěnami polygonu (jeho vnitřní úhel).

Tento úhel se odečte od sta osmdesáti, čímž získáme inkrement úhlu vektoru mezi počátkem souřadného systému (SS) a vrcholy.

Vrcholy se poté naleznou přičtením tohoto patřičně rotovaného vektoru k počátku SS. Všechny vrcholy jsou ve vzdálenosti jedna od počátku SS.

Následně se generují body mezi po sobě jdoucími vrcholy. Mezi dvěma vrcholy se generuje tolik bodů, kolik udává uživatelsky zadaný parametr (viz obr. 6.1 v uživatelské příručce).

Generování rovnoběžníků

Generování množiny bodů rovnoběžníku probíhá následovně: Postupně se generují body na jeho jednotlivých stranách. Směr postupu po straně popisuje vektor. Po dokončení strany se tento vektor rotuje o úhel vypočítaný z uživatelsky zadaného parametru určujícího, jaký úhel mají svírat strany rovnoběžníku (viz obr. 6.1 v uživatelské příručce) a pokračuje se v generování bodů na dalších stranách.

Na delších stranách rovnoběžníku se generuje tolik bodů, kolik udává uživatelsky zadaný parametr (viz obr. 6.1 v uživatelské příručce), na kratších stranách je generováno méně bodů podle toho, v jakém poměru jsou délky stran.

Generování obecných funkcí

V této sekci je popsán použitý algoritmus pro hledání intervalu, na kterém se nachází zadaná funkce a následné generování množiny bodů z této funkce.

Metoda pro nalezení intervalu

Metoda pro nalezení „rozsahu“ funkce vyžaduje zadání maximálního možného rozsahu funkce, tzn. interval na osách x i y , na kterém se funkce určitě bude nacházet, zúžením tohoto intervalu lze vymezit část křivky, která bude zkoumána. Interval je implicitně nastaven na interval od -10 do 10 na osách x i y .

Pozn.: Tato práce se nezabývá hledáním části křivky, která je pro uživatele zajímavá, oblast zájmu se předpokládá u počátku souřadného systému. Pokud by tomu tak nebylo, je možné upravit intervaly na osách x i y tak, aby se omezily na část křivky, která má být zkoumána (viz obr. 6.1 v uživatelské příručce).

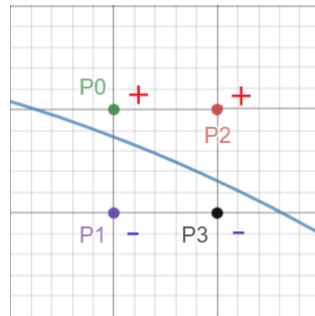
Samotný výpočet rozsahu probíhá takto: Postupuje se s relativně velkým krokem od zadaných maximálních hranic směrem k počátku souřadného systému, tzn. „shora, zdola, zleva a zprava“. Jakmile se narazí na křivku (tj. změní se znaménko výsledku výrazu funkce po dosažení daného bodu nebo se výsledek rovná nule), tento bod se uloží jako hraniční, pokud je jeho právě zkoumaná souřadnice (souřadnice x v případě postupu shora a zdola a souřadnice y v případě postupu zleva a zprava) větší či menší než naposledy nalezená.

Největší souřadnici y hledáme při postupu shora, největší x při postupu zprava, nejmenší y při postupu zdola a nejmenší x při postupu zleva.

Generování množiny bodů

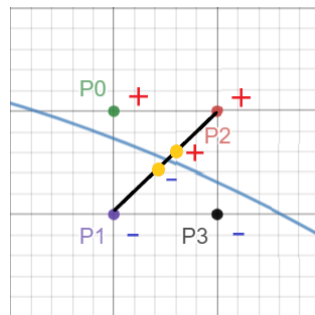
Metoda pro hledání množiny bodů z obecné funkce probíhá takto: Postupuje se s malým krokem přes celou oblast. Pro každý bod v oblasti se najdou tři jeho sousedi, tj. bod hned pod ním, bod hned vpravo a bod diagonálně vpravo dolů (viz body P0, P1, P2 a P3 na obr. 3.1).

Všechny tyto čtyři body se dosadí do výrazu funkce a následně se díváme na znaménka výsledků. Pokud všechny vrácené hodnoty pro tyto body mají stejné znaménko, tímto pomyslným obdélníkem křivka neprochází. Naopak pokud se znaménka liší (případ na obrázku 3.1), vytvoří se dvě skupiny bodů - v jedné skupině jsou body se záporným znaménkem a v té druhé ty s kladným.



Obrázek 3.1: Čtyři zkoumané body a „jejich znaménka“

Dále vytvoříme všechny možné dvojice bodů s opačnými znaménky. Mezi nimi s malým krokem postupujeme a zkoumáme jednotlivé body mezi dvojicí (viz obr. 3.2). Jakmile se změní znaménko jednoho z bodů mezi dvojicí nebo dostaneme nulový výsledek, je tento bod považován za bod ležící na křivce a je zahrnutý do výsledné množiny bodů. Pokud se ale postup mezi dvojicí dostal až za koncový bod (to se může stát např. v případě křivky na obr. 2.2), zkoumání končíme a žádný bod do množiny bodů nepřidáváme.



Obrázek 3.2: Postup mezi body P1 a P2

Body jsou ukládány ve 3D podobě, ale souřadnice z všech bodů je nastavena na 0.

Délka kroku v oblasti je vypočtena podle rozsahu funkce na ose x , resp. y . Konkrétně je to pětisetina daného rozsahu.

Délka kroku *step* mezi dvěma body v pomyslném obdélníku je vypočtena

takto:

$$step = \text{Min}(\text{Min}(stepX, stepY)/4, 0.001),$$

kde $stepX$ a $stepY$ jsou délky kroků, s kterými se postupuje přes celou oblast (ve směru x a y). Čtvrtina menšího kroku je zvolena proto, aby byl otestován dostatečný počet bodů v blízkosti křivky, čímž dostaneme přesnější výsledky. Jedná se o reálné dělení. Experimenty ukázaly, že při kroku menším nebo rovném 0.001 dosahujeme přesnějších výsledků.

3.3 Metoda detekce osové symetrie

Pro detekci os (resp. rovin) symetrie je využita metoda Ing. Lukáše Hrudý.[7] Této metodě je jako vstup poskytnuta nalezená množina bodů popisující danou křivku.

Metoda funguje na dokonale i přibližně symetrických objektech, dokonce i na objektech s chybějícími částmi a je odolná vůči šumu. Dále funguje na objektech definovaných pouze množinou bodů, a proto neklade prakticky žádná omezení na vstupní data.

Metoda je založena na maximalizaci míry symetrie účelovou funkcí, která vyhodnocuje symetrii objektu vzhledem k dané rovině a je definována takto:

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

Vektor $p = [a, b, c, d]^T$ představuje libovolnou rovinu definovanou jako $ax + by + cz + d = 0$, funkce $r(p, x)$ zrcadlí bod x přes plochu p , x_1, \dots, x_n je množina bodů vstupního objektu a $\varphi(l)$ je nějaká radiální funkce, jejíž nejvyšší hodnota je pro $l = 0$ a její hodnota klesá k 0 s růstem l . Pokud jsou kromě bodů k dispozici nějaké další informace o vstupním objektu, váhy w_{ij} mohou být nastaveny pro využití této informace ke zlepšení výkonu, ale ve výchozím nastavení se váhy nepoužívají a jejich hodnota je nastavena na 1.

Tato metoda hledá rovinu symetrie pro 3D objekty. V našem případě jí jako vstup zadáváme 2D data. Z nalezené roviny $ax + by + cz + d = 0$ získáme příslušnou přímku takto: $ax + by + d = 0$. Jedná se tedy o projekci této roviny do roviny XY .

3.4 Detekce rotační a středové symetrie

Postup určování rotační a středové symetrie je následující: Nejdříve se vyloučí osy nalezené softwarem pro detekci symetrie, které nezrcadlí křivku,

tj. nerozdělují ji na dvě stejné zrcadlově převrácené části. Jak lze vidět v provedených experimentech (viz kapitolu 4 na str. 23), ne vždy jsou osy určeny správně.

Dále je vhodné zde zmínit, že využívaný software Ing. Hrudý je primárně určen k hledání jedné osy symetrie. Jsme sice schopni jich nechat vypočítat více, ale nemáme zaručeno, že počet nalezených os bude stejný jako počet požadovaných (nalezených os může být méně než požadovaných, ale může jich být více než skutečných os symetrie).

Osy jsou vylučovány takto: Všechny vygenerované body se zrcadlí přes jednotlivé osy. Pro každý bod se hledá obraz „na druhé straně“ osy s nějakou tolerancí (tzn. obraz nemusí ležet přesně na na vypočítaném místě). Pro nevyločení osy musí alespoň 97% bodů množiny mít takto určený obraz.

Pozn.: Vzhledem k charakteru generování bodů nemáme zaručeno, že zrcadlové části křivky jsou skutečně ekvivalentní, proto je podmínka bijekce opomenuta. Tzn. nezkoumáme, jestli má každý bod (vzor) právě jeden obraz a všechny možné obrazy jsou použité.

Při translaci a rotaci bodů je potřeba uživatelsky zadat toleranci pro mapování bodů (viz obr. 6.4), protože nebyla nalezena žádná korelace mezi např. rozsahem funkce a nutnou velikostí tolerance.

Pro zjištění rotační symetrie se postupuje následovně: Nejdříve se nalezne bod, kolem kterého bude množina bodů rotována. Složky tohoto bodu se vypočítají jako průměr příslušných složek průsečíků každých dvou (nevyločených) os. Následně najdeme úhly mezi každými dvěma osami. Úhly seřadíme, zprůměrujeme „stejně“ úhly (úhly jsou považovány za stejné, pokud je jejich rozdíl menší než 5°) a nakonec zaokrouhlíme. Body rotujeme nejprve o nejmenší úhel a opět sledujeme, jestli po aplikaci transformace rotace pro každý bod nalezneme obraz. Nejmenší postupně násobíme celým kladným číslem vždy o jedna vyšším než v předchozím kole a sledujeme, jaký nejmenší úhel je úhlem rotační symetrie. Po nalezení prvního (nejmenšího) úhlu rotační symetrie proces detekce této symetrie končí.





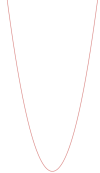

Pro určení středové symetrie jdeme přes všechny body z množiny a každému zkusíme najít obraz takovým způsobem, jak bylo popsáno v podsekcí 2.2.6 na str. 15. Opět je implementována tolerance pro obraz bodu.

4 Experimenty

Tato kapitola obsahuje ukázky experimentů prováděné s vybranými křivkami a softwarem pro detekci symetrie. Na výsledné osy nebyl kromě sekce 4.4 aplikován žádný „post-processing“ ve smyslu vylučování os, které nezrcadlí množinu vygenerovaných bodů, ale jejich správnost byla ověřována způsobem popsaným v sekci 3.4.

V tabulce 4.1 lze vidět obecné funkce a jejich rovnice použité v experimentech, v tabulce 4.2 jsou zobrazeny polygony použité v experimentech.


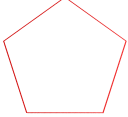
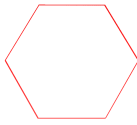

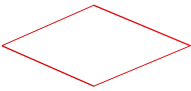
Experimenty byly prováděné na zařízení s procesorem *Intel® Core™ i5-7200U* a 8 GB paměti na operačním systému *Windows 10*.

Vizualizace	Rovnice
	$(x^2 + y^2 - 1)^3 - x^2y^3 = 0$
	$(x^2 + y^2)^2 - (x^2 - y^2) = 0$
	$(x^2 + y^2 + 6x + 2.25)^2 - 2(2x + 1.5)^3 = 0$
	$\frac{x^3}{4-x} - y^2 = 0$
	$y - x^2 = 0$
	$x(x^2 - 2.1x + 1) - 2y^2 = 0$

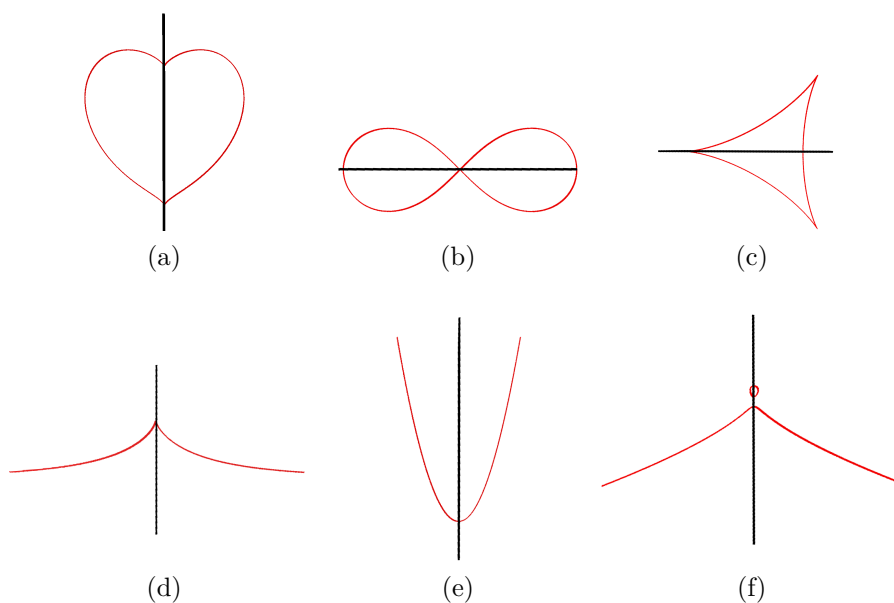
Tabulka 4.1: Obecné funkce použité v experimentech a jejich rovnice. Zobrazeno pomocí <https://www.desmos.com/calculator/>

4.1 Detekce jedné osy symetrie

Na obr. 4.1 jsou vidět ukázky detekce jedné osy symetrie v křivkách z tabulky 4.1. Lze vidět, že u všech těchto křivek byla osa detekována správně.

Vizualizace	Popis
	Čtverec
	Pětiúhelník
	Šestiúhelník
	Rovnoběžník, strany 2:1, úhel 45°
	Rovnoběžník, strany 1:1, úhel 45°

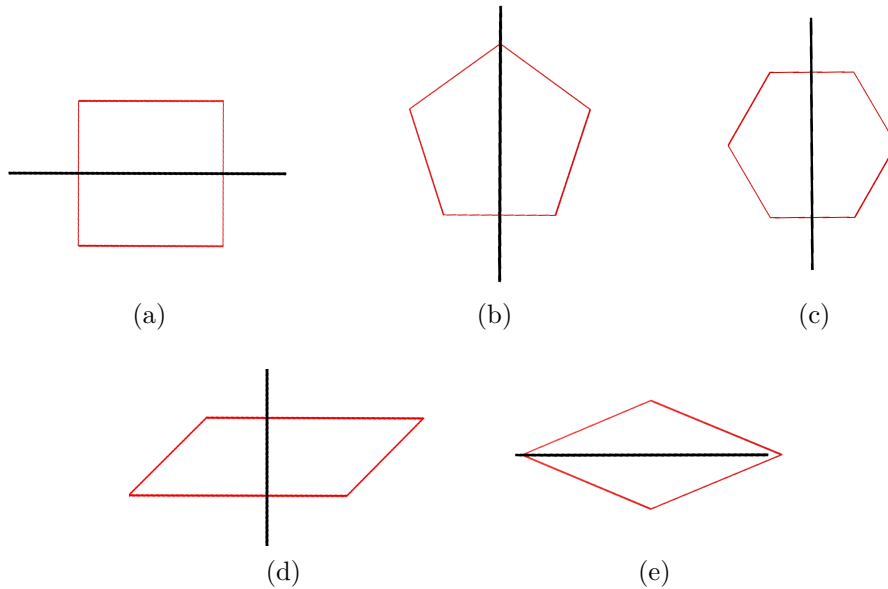
Tabulka 4.2: Polygony použité v experimentech



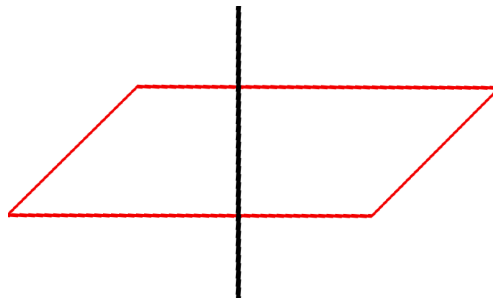
Obrázek 4.1: Detekce jedné osy symetrie v obecných funkcích

Na obr. 4.2 jsou vidět ukázky detekce jedné osy symetrie v křivkách z tabulky 4.2. Ve všech případech, kromě rovnoběžníku (d) (viz větší detail na obr. 4.3) byla osa detekována správně. Nesprávná detekce v případě rovnoběžníku (d) může být způsobena malým počtem vygenerovaných bodů. Pro vyloučení tohoto případu byl experiment opakován s 300 000 vygene-

rovanými body, stále s chybným výsledkem. Dalším možným vysvětlením je „dlouhé“ tělo objektu s malými změnami v částech rozdělených osou.



Obrázek 4.2: Detekce jedné osy symetrie v polygonech



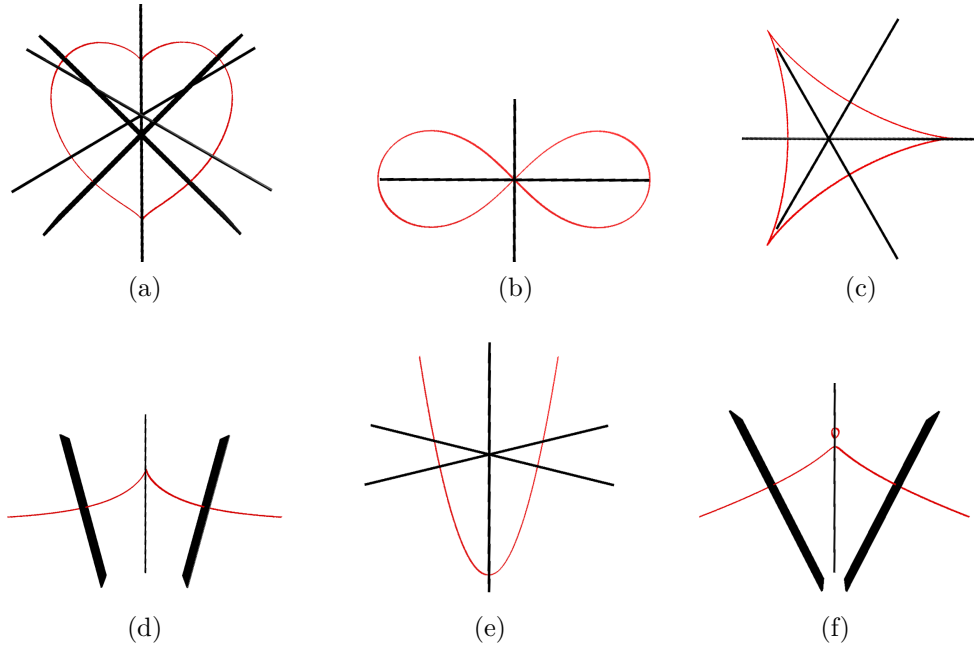
Obrázek 4.3: Rovnoběžník, detekce jedné osy

Detekce jedné osy symetrie dává v naprosté většině případů správné výsledky.

4.2 Detekce více os symetrie

Na obr. 4.4 jsou vidět ukázky detekce více os symetrie v křivkách z tabulky 4.1. Detekce os v případě křivek (b) a (c) dala správné výsledky. V případě křivky (a) bylo detekováno mnoho přebytečných, a tedy nesprávných os. Tvar této křivky je ale téměř symetrický i podle těchto os. V případě

křivek (d) a (f) lze zvolení špatných os pochopit vzhledem k dlouhým, téměř rovným částem křivek. Připomeňme, že osa symetrie je hledána programem pro detekci roviny symetrie a v případě křivek (d) a (f) nejsou některé roviny zobrazeny kolmo k rovině, ve které leží křivka. Dvě nesprávné osy v případě křivky (e) byly pravděpodobně zvoleny kvůli přibližné symetrii.



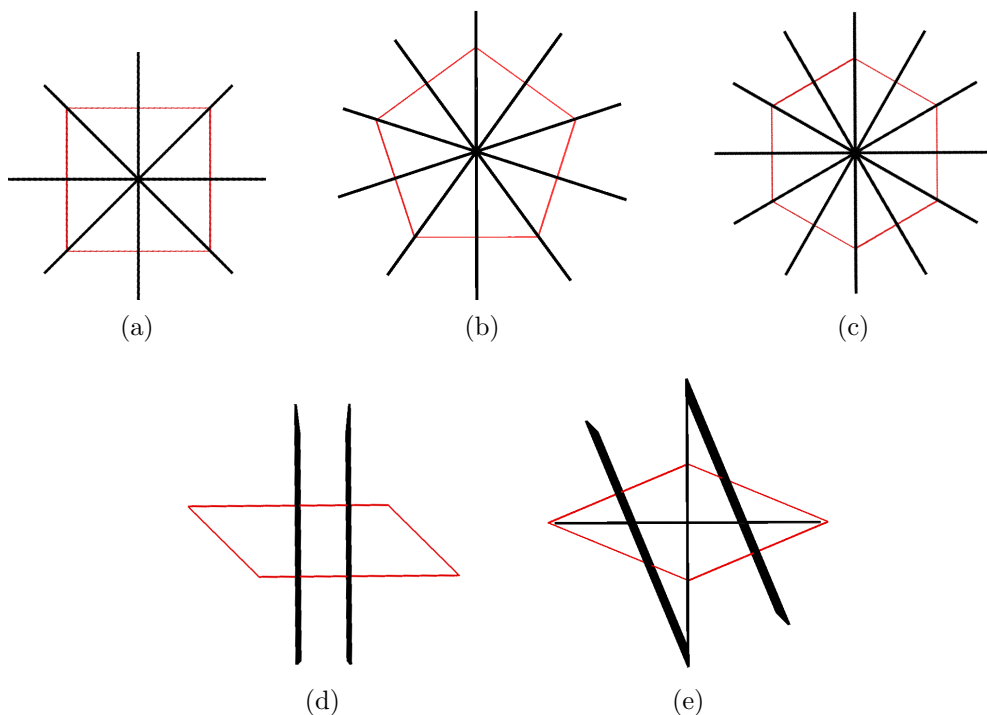
Obrázek 4.4: Detekce os symetrie v obecných funkcích

Na obr. 4.5 jsou vidět ukázky detekce více os symetrie v polygonech z tabulky 4.2. V případě pravidelných polygonů (tj. ukázky (a) - (c)) jsou výsledky výborné. Byly detekovány všechny osy symetrie daných polygonů a byly detekovány správně. Výsledek v případě (d) je podobný výsledku při detekci jedné osy symetrie. Rovnoběžníku (e) byly dvě osy detekovány správně a dvě nesprávně.

4.3 Šum a porušené křivky

Tato sekce se zabývá experimenty s křivkami, do kterých byly zaneseny různé druhy šumu. Zároveň ukazuje detekci více os symetrie na daných porušených křivkách. V následujících ukázkách zatím neprobíhala kontrola nalezených os.

Každá z následujících podkapitol obsahuje popis postupu aplikace šumu, dále obsahuje ukázky výstupu programu a stručné zhodnocení výsledků experimentů.



Obrázek 4.5: Detekce os symetrie v polygonech

Pravidelný šum

Tato sekce se zabývá experimenty s pravidelným šumem.

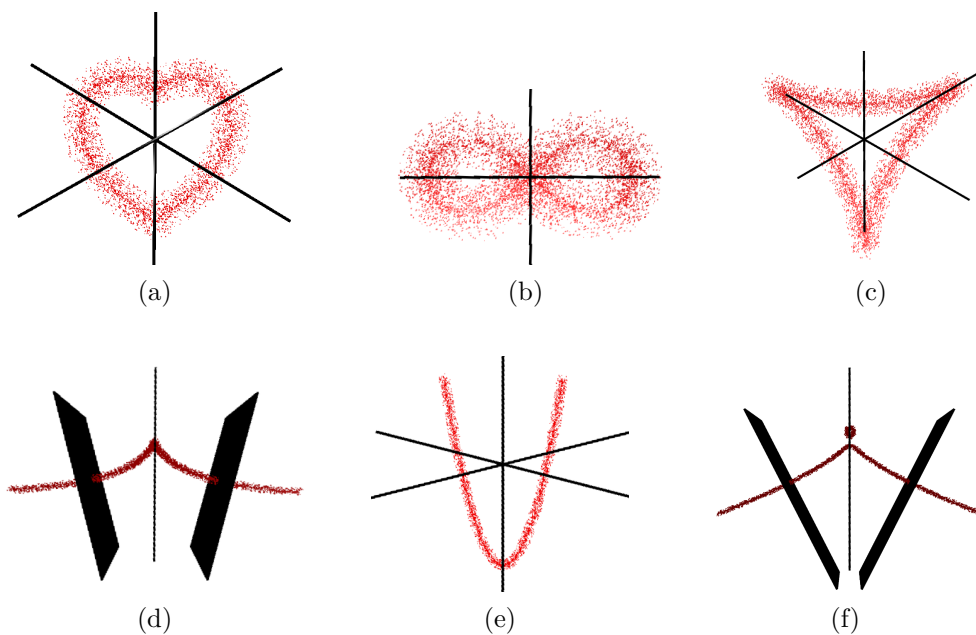
Popis algoritmu

Zanesení pravidelného šumu do vygenerované množiny bodů znamená modifikaci souřadnic jednotlivých bodů tak, že každý bod se posune v náhodném směru náhodně daleko, ne ale dál, než udává parametr maximální odchylky (viz obr. 6.3 v uživatelské příručce).

Ukázky a zhodnocení

V této podsekcí uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení. Ve všech případech byla požadována detekce šesti os symetrie.

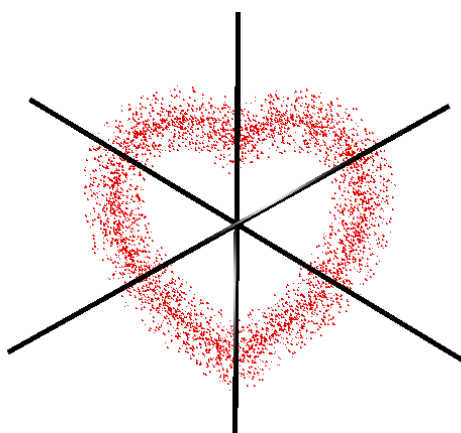
Na obr. 4.6 vidíme, že v mnoha případech pravidelný šum nebrání správné detekci os symetrie. Ve všech případech byla alespoň jedna osa detekována správně. V případě křivky (a) jsou dvě osy určeny nesprávně zřejmě proto, že křivka nemá žádné výrazné rysy, které by způsobily nezvolení těchto os



Obrázek 4.6: Detekce os symetrie v obecných funkcích - pravidelný šum

(podrobněji viz obr. 4.7). V případě křivek (d) a (f) lze zvolení špatných os pochopit vzhledem k dlouhým, téměř rovným částem křivek.

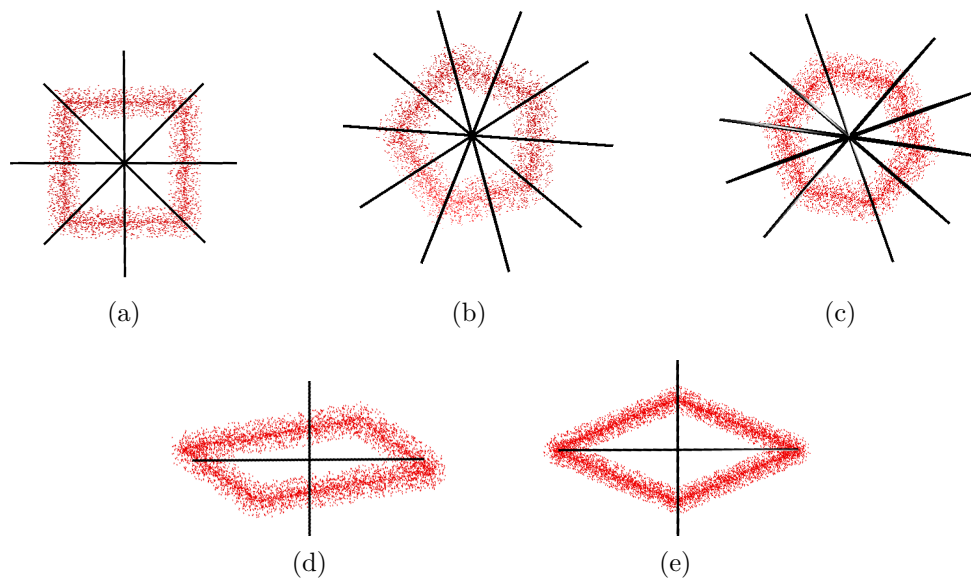
Určení os v případě 4.6 (b) dopadlo výborně i přes výrazné porušení vstupních dat.



Obrázek 4.7: Detail - pravidelný šum

Na obr. 4.8 nejsou kromě křivky (c) vidět žádné nečekané výsledky. V případě křivky (c) byla vynechána jedna osa, přestože bylo požadováno 6 os - takové chování je ale očekávatelné, protože software pro detekci osové symetrie nezaručuje výstup stejného počtu os, jaký byl požadován.

Obecně lze říci, že aplikace pravidelného šumu výsledky obvykle příliš



Obrázek 4.8: Detekce os symetrie v polygonech - pravidelný šum

nemění, maximálně při zadání velké odchylky bodů.

Nepravidelný šum

Tato sekce se zabývá experimenty s nepravidelným šumem.

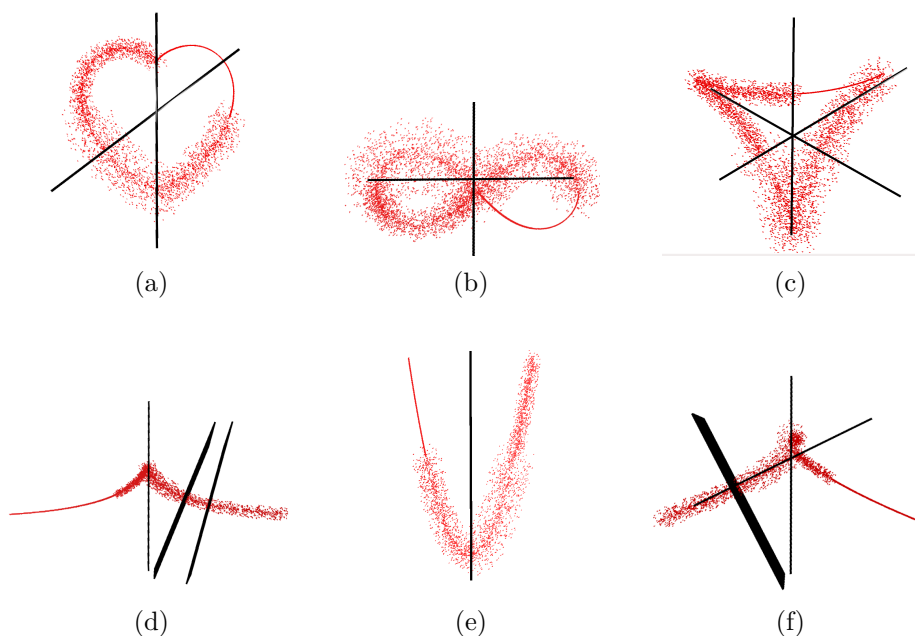
Popis algoritmu

Zanesení nepravidelného šumu do vygenerovaných bodů znamená modifikaci souřadnic jednotlivých bodů tak, že body náležící různým částem křivky se posunou v náhodném směru různě daleko. Tyto odchylky jsou určeny programově podle rozsahu jednotlivých křivek.

Ukázky a zhodnocení

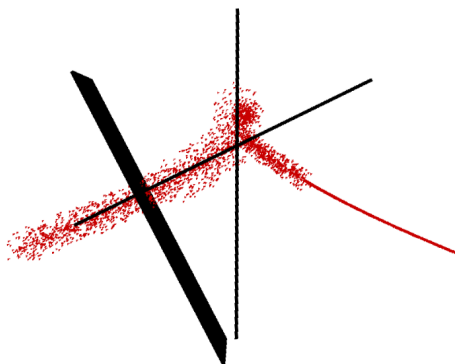
V této podsekcí uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení. Ve všech případech byla požadována detekce šesti os symetrie.

Pro křivky na obr. 4.9 byla opět alespoň jedna osa detekována správně. Ve většině případů jsou výsledky velmi dobré. Výsledek pro křivku (b) je zajímavý - průsečík os je vychýlen ze středu křivky zřejmě kvůli výraznějšímu poškození v její horní části, ale postavení os vůči sobě je správné. V případě křivky (f) je zajímavá volba osy, která podélně prochází zašuměnou částí množiny bodů. Tato osa byla nejspíše zvolena kvůli „výstupku“ v horní



Obrázek 4.9: Detekce os symetrie v obecných funkcích - nepravidelný šum

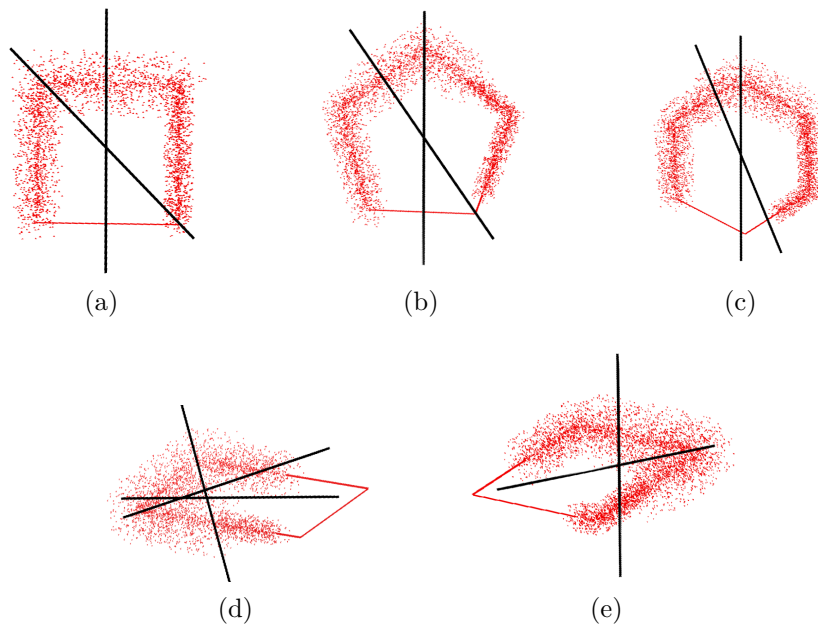
části křivky, který vypadá podobně jako kus její pravé části (podrobněji viz obr. 4.10).



Obrázek 4.10: Detail - nepravidelný šum

Na obr. 4.11 vidíme, že pro většinu polygonů byl detekován výrazně nižší počet os, než bylo požadováno, což je rozdíl oproti experimentům s nezašuměnými daty i s pravidelným šumem.

V případě nepravidelného šumu se výsledky liší případ od případu, ale vypadá to, že nepravidelný šum dělá detektoru větší problém než pravidelný šum.



Obrázek 4.11: Detekce os symetrie v polygonech - nepravidelný šum

Nerovnoměrné rozložení bodů

Tato sekce se zabývá experimenty s nerovnoměrným rozložením bodů.

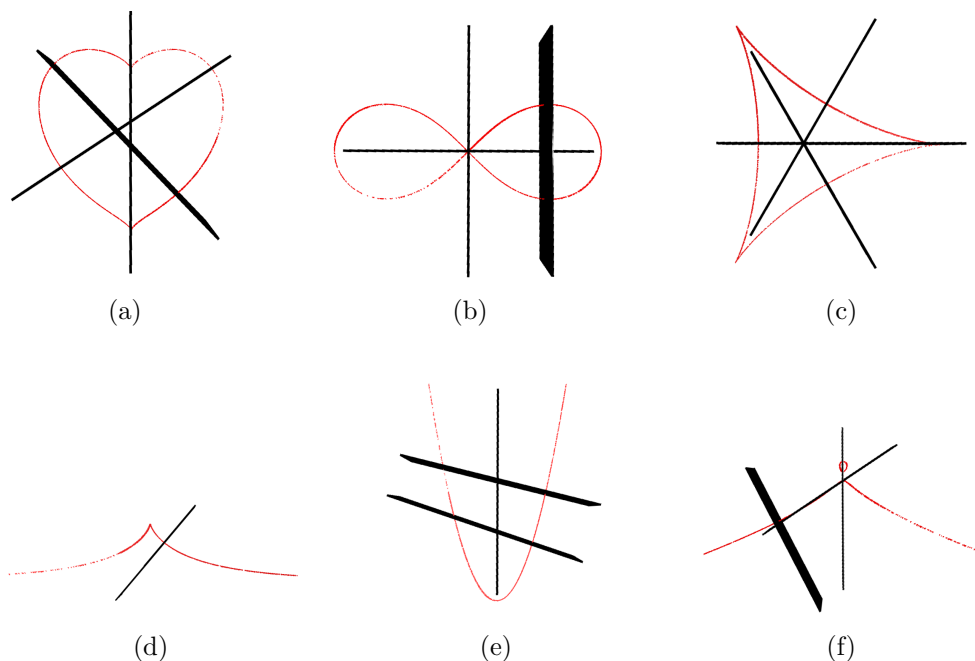
Popis algoritmu

Nerovnoměrné rozložení bodů se aplikuje tak, že z různých částí křivky se různě velká část již vygenerovaných bodů odstraní. Kolik a které konkrétní body z dané části budou odstraněny je řízeno náhodou. Intervaly jsou určeny programově podle rozsahu křivky.

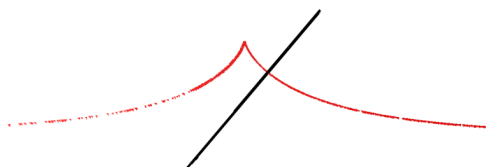
Ukázky a zhodnocení

V této podsekcí uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení. Ve všech případech byla požadována detekce šesti os symetrie.

Na obr. 4.12 vidíme, že ve většině případů nejsou výsledky špatné. Výstupy pro křivky (a), (c), (e) a (f) nejsou výrazně odlišné od předchozích experimentů. Křivce (b) byla detekována jedna osa navíc, která prochází částí s větší hustotou bodů. Pro křivku (d) byla detekována jediná osa symetrie, která není správná, ale prochází oblastí s nejvyšší koncentrací bodů (viz obr. 4.13).



Obrázek 4.12: Detekce os symetrie v obecných funkcích - nerovnoměrné rozložení bodů



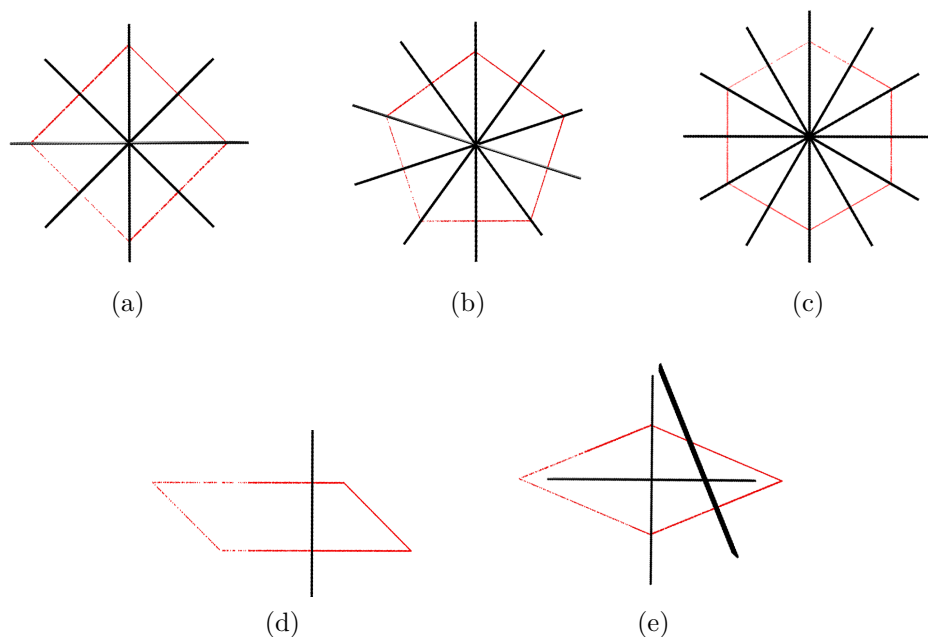
Obrázek 4.13: Detail - nerovnoměrné rozložení bodů

Výsledek pro polygony (a) - (c) na obr. 4.14 dopadl výborně, byly detekovány všechny osy symetrie a byly detekovány správně. Osa rovnoběžníku (d) není určena správně, ale opět prochází oblastí s vyšší koncentrací bodů, stejně jako jedna z os detekována pro rovnoběžník (e).

Nezdá se, že by nerovnoměrné rozložení bodů výrazně zhoršovalo výsledky oproti jiným typům šumu, naopak oproti nerovnoměrnému šumu jsou výsledky zejména v případě polygonů lepší.

Odstranění části křivky

V této podsececi uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení.



Obrázek 4.14: Detekce os symetrie v polygonech - nerovnoměrné rozložení bodů

Popis algoritmu

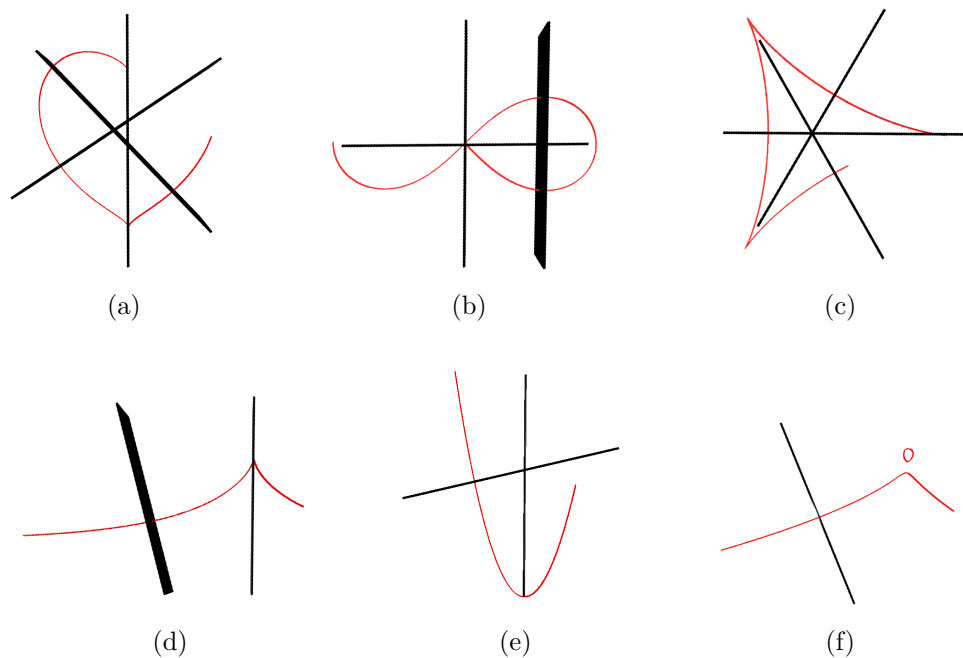
Odstranění části křivky probíhá tak, že se vybere interval (jeden kvadrant oblasti, ve které se křivka nachází), z kterého jsou následně odstraněny všechny body.

Ukázky a zhodnocení

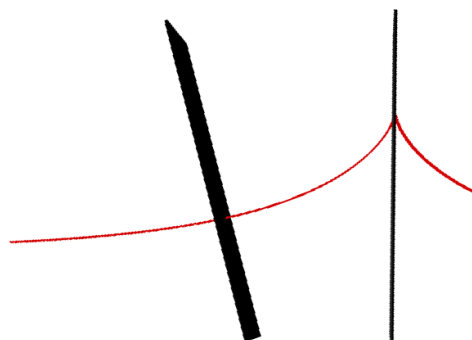
V této podsececi uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení. Ve všech případech byla požadována detekce šesti os symetrie.

Na obr. 4.15 byly pro křivku (b) detekovány stejné osy jako v případě nerovnoměrného rozložení bodů. V případě (d) byly detekovány 2 osy symetrie, obě dvě lze považovat za přijatelné, navíc osa jdoucí svisle dolů (správná) byla detekována i přesto, že velká část pravé strany křivky byla odstraněna, což je velmi dobrý výsledek (viz obr. 4.16). Je překvapivé, že pro křivku (f) správná osa detekována nebyla, i když křivka je svým tvarem podobná (d).

Výsledky pro polygony na obr. 4.17 jsou dobré, především v případě pětiúhelníku (b), pro který byly detekovány všechny osy správně a žádná nebyla vynechána. V případě čtverce (a) a šestiúhelníku (c) některé osy chybí, ale zbytek je detekován správně. Pro rovnoběžník (d) nebyla žádná



Obrázek 4.15: Detekce os symetrie v obecných funkcích - odstranění části křivky



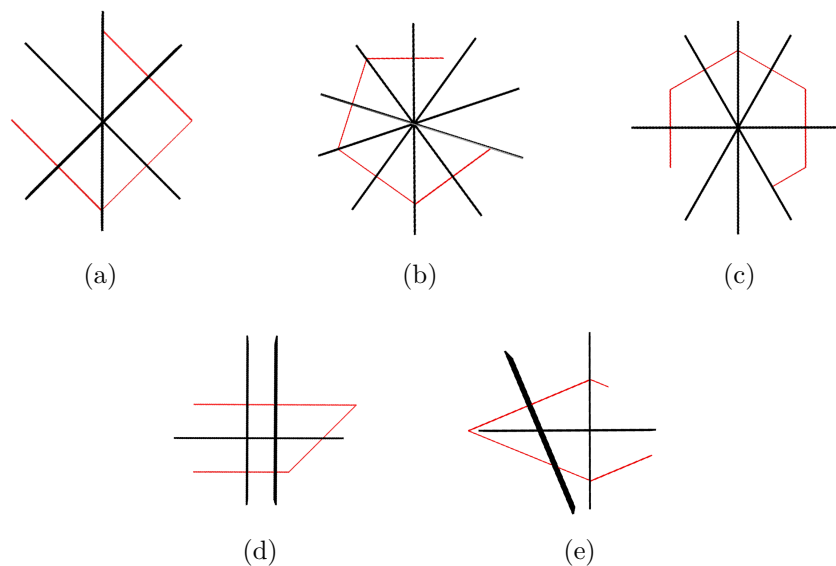
Obrázek 4.16: Detail - odstranění části křivky

osa detekována správně, ale vzhledem k tomu, která jeho část chybí, to není překvapivé. Stejně tak jedna z os nalezená rovnoběžníku (e) je chybná, to je nejspíš způsobeno chybějící částí.

Detektor osové symetrie je velmi odolný k chybějícím částem objektů.

Deformace tvaru polygonu

Tato sekce se zabývá experimenty s deformací tvaru polygonů.



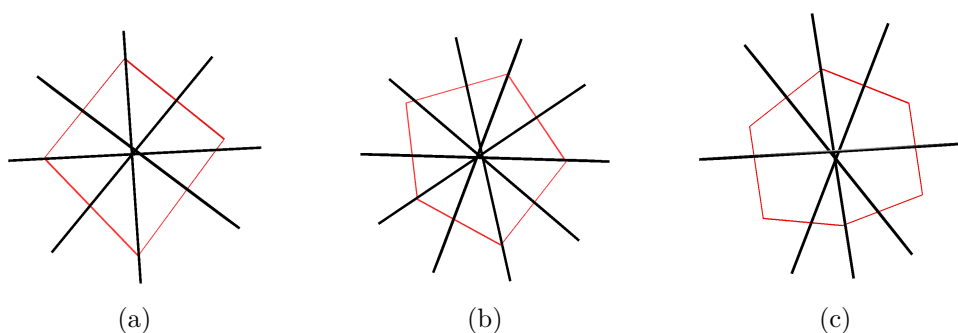
Obrázek 4.17: Detekce os symetrie v polygonech - odstranění části křivky

Popis algoritmu

Pro deformaci tvaru polygonu se náhodně vybere jeden vrchol, který se následně posune v náhodném směru o vzdálenost, kterou udává parametr odchylky (viz 6.3 v uživatelské příručce).

Ukázky a zhodnocení

V této podsekcí uvidíme výstupy softwaru pro detekci symetrie a jejich krátké zhodnocení. Ve všech případech byla požadována detekce šesti os symetrie. Možnost deformace tvaru polygonu je implementována jen pro pravidelné polygony, nikoliv pro funkce nebo rovnoběžníky.



Obrázek 4.18: Detekce os symetrie v polygonech - deformace tvaru

Na obr. 4.18 vidíme velmi dobré výsledky. V případě šestiúhelníku (c) chybí dvě osy, to je ale vzhledem k jeho výslednému tvaru pochopitelné.

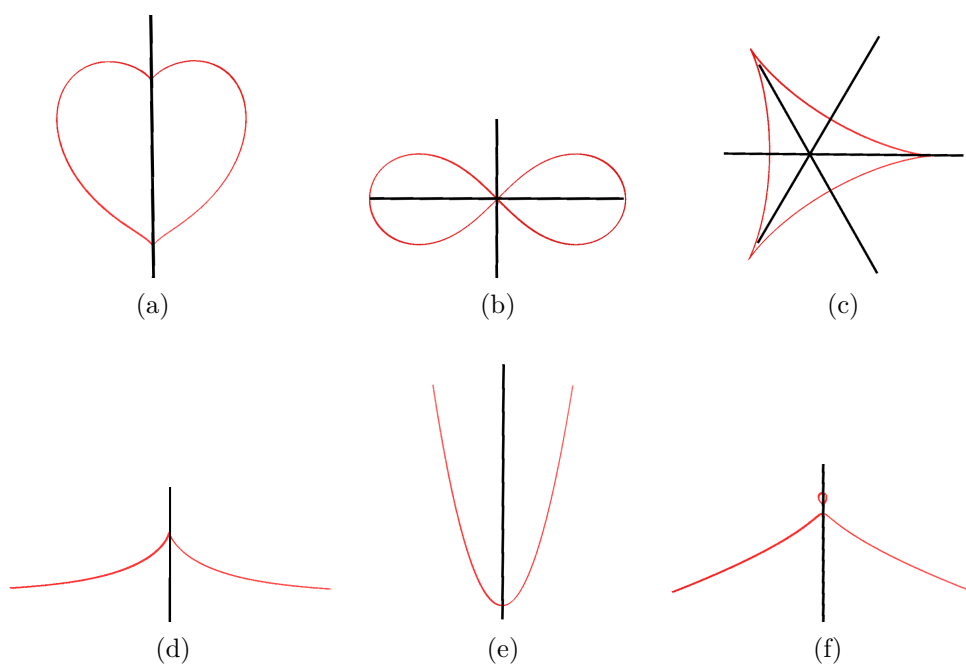
Při deformaci tvaru polygonu výsledek hodně závisí na výsledném tvaru polygonu a na velikosti odchylky, při malé deformaci jsou ale výsledky většinou správné.

4.4 Rotační a středová symetrie

Při experimentování s rotační a středovou symetrií byly uplatněny znalosti popsané v sekci 2.2 na str. 11.

Ukázky a zhodnocení

Ukázky jsou zobrazeny po vyloučení nalezených os, které křivky nezrcadlí. Ve všech případech byla požadována detekce šesti os symetrie.



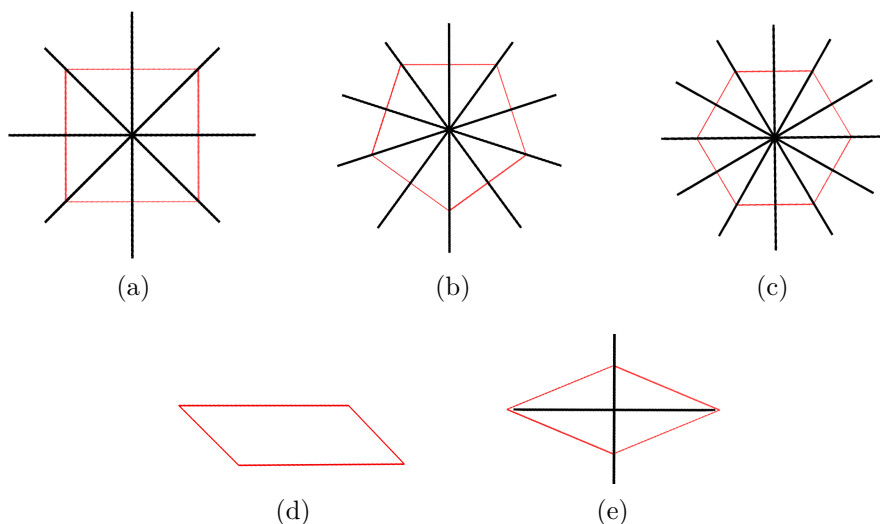
Obrázek 4.19: Detekce rotační a středové symetrie v obecných funkcích

Na obr. 4.19 vidíme výstup programu po vyloučení nevhodných os, tolerance zrcadlení i rotování bodů byla pro všechny tyto křivky 0,2. Křivkám (a) a (d) - (f) zůstala jedna (správná) osa symetrie, pro tyto funkce rotační a středová symetrie tedy zjišťována nebyla.

Funkci (b) byla detekována rotační symetrie řádu 2 (nejmenší úhel 180°) a středová symetrie. Všimněme si - rotace o 180° je ekvivalentní středové

symetrii.

Funkci (c) byla detekována rotační symetrie řádu 3 (nejmenší úhel 120°) a žádná středová symetrie.



Obrázek 4.20: Detekce os symetrie v polygonech - odstranění části křivky

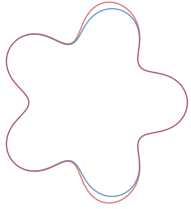
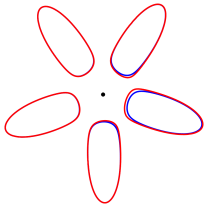
Na obr. 4.20 vidíme výstupy opět po vyloučení os, tolerance zrcadlení i rotování bodů byla pro všechny tyto polygony 0,1. Rovnoběžníku (d) nezůstala ani jedna osa (ani jedna osa nebyla určena správně), zde tedy rotační ani středová symetrie zjišťována nebyla. Pro čtverec (a) byl výstup rotační symetrie řádu 4 (nejmenší úhel 90°) a středová symetrie. Podobný výsledek platí pro šestiúhelník (c), ovšem zjistili jsme rotační symetrii řádu 6 (nejmenší úhel 60°) a opět i středovou symetrii. Pětiúhelník (b) má rotační symetrii řádu 5 (nejmenší úhel 72°) a nemá středovou symetrii. Pro rovnoběžník (e) byla zjištěna rotační symetrie řádu 2 (nejmenší úhel 180°) i středová symetrie.

Zjišťování rotační a středové symetrie funguje při dostatečném počtu správných os a při zvolení vhodné tolerance mapování bodů velmi dobře a dává správné výsledky.

4.5 Symetrie pro přibližně symetrické funkce

Díky této práci jsme schopni určit rotační a středovou symetrii i pro přibližně symetrické funkce.

V tabulce 4.3 vidíme ukázky čtyř funkcí. Barvy křivek korespondují s barvami textu rovnic. Z obrázků vidíme, že změna tvaru není velká a

Vizualizace	Rovnice
	$x^{10} + 5x^8y^2 - 5x^8 + 10x^6y^4 - 20x^6y^2 - 35x^6 - 486x^5 + 10x^4y^6 - 30x^4y^4 - 105x^4y^2 - 280x^4 + 4860x^3y^2 + 5x^2y^8 - 20x^2y^6 - 105x^2y^4 - 560x^2y^2 - 2560x^2 - 2430xy^4 + y^{10} - 5y^8 - 35y^6 - 280y^4 - 2560y^2 - 17232 = 0$ $x^{10} + 5x^8y^2 - 5x^8 + 10x^6y^4 - 20x^6y^2 - 35x^6 - 486x^5 + 10x^4y^6 - 30x^4y^4 - 105x^4y^2 - 280x^4 + 4860x^3y^2 + 5x^2y^8 - 20x^2y^6 - 105x^2y^4 - 560x^2y^2 - 2560x^2 - 2430xy^4 + 0.7y^{10} - 5y^8 - 35y^6 - 280y^4 - 2560y^2 - 17232 = 0$
	$-10141 + 31726x - 36925x^2 + 20488x^3 - 5824x^4 + 818x^5 - 45x^6 + 8122y + 5024xy - 7704x^2y + 2200x^3y - 170x^4y - 38717y^2 + 32568xy^2 - 9888x^2y^2 + 1540x^3y^2 - 135x^4y^2 + 13016y^3 - 8680xy^3 + 1420x^2y^3 - 1504y^4 + 850xy^4 - 135x^2y^4 + 182y^5 - 45y^6 = 0$ $-10140.9923 + 31726.006x - 36925.0021x^2 + 20488.0032x^3 - 5824.0005x^4 + 817.9963x^5 - 44.9938x^6 + 8122y + 5024.0034xy - 7704.0001x^2y + 2200.0059x^3y - 169.9961x^4y - 0.0017x^5y - 38717.0083y^2 + 32568.0034xy^2 - 9887.9944x^2y^2 + 1539.9929x^3y^2 - 135.0097x^4y^2 + 13016.0004y^3 - 8680.0033xy^3 + 1420.0056x^2y^3 + 0.0003x^3y^3 - 1504.0037y^4 + 849.9945xy^4 - 134.9921x^2y^4 + 181.996y^5 - 0.0014xy^5 - 44.9923y^6 = 0$

Tabulka 4.3: Obecné funkce použité v experimentech se středovou a rotační symetrií a jejich rovnice.

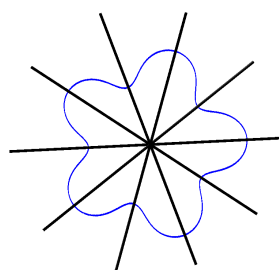
v takových případech jsme stále schopni správně určit rotační a středovou symetrii.

Pozn.: „Modré“ funkce z tabulky 4.3 budeme nazývat přesnými a „červené“ nepřesnými.

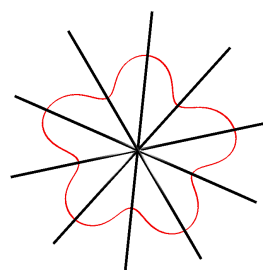
Pro funkci (a) na obr. 4.21 byla s tolerancí 0,2 detekována rotační symetrie stupně 5, tj. úhel 72° a žádná středová symetrie. Pro funkci (b) byl detekován stejný výsledek, ovšem s tolerancí 0,3.

Pro funkci (a) na obr. 4.22 byla s tolerancí 0,2 detekována rotační symetrie stupně 5, tj. úhel 72° a žádná středová symetrie. Pro funkci (b) byl detekován stejný výsledek se stejnou tolerancí.

Hodnota tolerance byla vždy nejprve nastavena na výchozí hodnotu (tj.

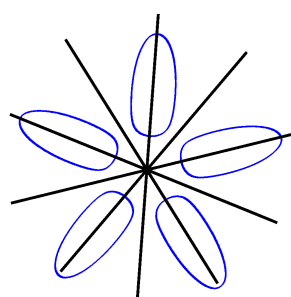


(a) Přesná funkce

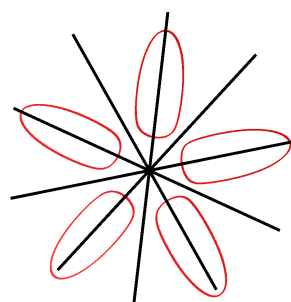


(b) Nepřesná funkce

Obrázek 4.21: Porovnání výstupu pro přesnou a nepřesnou funkci 1



(a) Přesná funkce



(b) Nepřesná funkce

Obrázek 4.22: Porovnání výstupu pro přesnou a nepřesnou funkci 2

0,2). Pokud tato hodnota nedávala správný výsledek, byla zvyšována, dokud výsledek nebyl správný.

5 Závěr

V teoretické části byla nejprve stručně popsána známá metoda rasterizace křivky popsané implicitní funkcí. Poté byla představena definice symetrie.

V rámci praktické části práce byla implementována metoda pro generování množiny bodů z implicitně zadané funkce. Tato metoda se ukázala jako dostačující pro testovaná vstupní data, avšak klade určitá omezení na zadávané funkce, protože je potřeba, aby definiční obor zadávané funkce nebyl nijak omezen. Dále byla implementována metoda pro detekci rotační a středové symetrie s využitím existujícího softwaru pro detekci osové symetrie ve 3D objektech. Funkčnost tohoto softwaru na našich vstupech byla překvapivě dobrá vzhledem k tomu, že tento software je stavěn na mnohem větší a 3D vstupní data. Experimenty ukázaly, že naše metoda detekce rotační a středové symetrie funguje při zadání vhodné tolerance pro mapování bodů velmi dobře a jsme díky ní schopni detekovat symetrii i v pouze přibližně symetrických objektech.

Vhodným námětem pro další postup v této oblasti může být spolupráce s katedrou matematiky a využití této práce jako pomocného nástroje pro řešení problémů v oblasti detekce symetrie ve funkcích.

Literatura

- [1] Algebraická funkce. Dostupné z:
<http://www.geneze.info/pojmy/subdir/funkce.htm>.
- [2] Funkce zadaná implicitně. Dostupné z:
https://mathonline.fme.vutbr.cz/download.aspx?id_file=1194.
- [3] Line and Rotational Symmetry. Dostupné z:
<http://year8maths.weebly.com/line-and-rotational-symmetry.html>.
- [4] Order of Rotational Symmetry. Dostupné z: <https://www.basic-mathematics.com/order-of-rotational-symmetry.html>.
- [5] Funkce dvou reálných proměnných. Dostupné z:
<http://rschwarz.wz.cz/fast/Mat2/FVP-I.pdf>.
- [6] HOBBY, J. Rasterization of Nonparametric Curves. *ACM Transactions on Graphics*. 12 1996, 9. doi: 10.1145/78964.78966.
- [7] HRUDA, L. – KOLINGEROVÁ, I. – VÁŠA, L. Robust, fast and flexible symmetry plane detection using differentiable symmetry measure. Submitted to Symposium on Geometry Processing, 2020.
- [8] HRUDA, L. Hledání symetrie v geometrických modelech, Jun 2018. Dostupné z: <http://hdl.handle.net/11025/31788>.
- [9] PODOLAK, J. et al. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics*. 2006, 25, 3, s. 549–559.
- [10] ROSEN, J. *Symmetry in science: an introduction to the general theory*. Springer, 1995.
- [11] SIMARI, P. – KALOGERAKIS, E. – SINGH, K. Folding Meshes: Hierarchical Mesh Segmentation Based on Planar Symmetry. In *Symposium on geometry processing*, s. 111–119, 2006.

Seznam obrázků

2.1	Uzavřená křivka a její kanonická rasterizace	10
2.2	Kubické křivky a znaménka funkčních hodnot	10
2.3	Rovnostranný trojúhelník[3]	15
2.4	Čtverec[3]	15
3.1	Čtyři zkoumané body a „jejich znaménka“	20
3.2	Postup mezi body P1 a P2	20
4.1	Detekce jedné osy symetrie v obecných funkcích	24
4.2	Detekce jedné osy symetrie v polygonech	25
4.3	Rovnoběžník, detekce jedné osy	25
4.4	Detekce os symetrie v obecných funkcích	26
4.5	Detekce os symetrie v polygonech	27
4.6	Detekce os symetrie v obecných funkcích - pravidelný šum	28
4.7	Detail - pravidelný šum	28
4.8	Detekce os symetrie v polygonech - pravidelný šum	29
4.9	Detekce os symetrie v obecných funkcích - nepravidelný šum	30
4.10	Detail - nepravidelný šum	30
4.11	Detekce os symetrie v polygonech - nepravidelný šum	31
4.12	Detekce os symetrie v obecných funkcích - nerovnoměrné rozložení bodů	32
4.13	Detail - nerovnoměrné rozložení bodů	32
4.14	Detekce os symetrie v polygonech - nerovnoměrné rozložení bodů	33
4.15	Detekce os symetrie v obecných funkcích - odstranění části křivky	34
4.16	Detail - odstranění části křivky	34
4.17	Detekce os symetrie v polygonech - odstranění části křivky	35
4.18	Detekce os symetrie v polygonech - deformace tvaru	35
4.19	Detekce rotační a středové symetrie v obecných funkcích	36
4.20	Detekce os symetrie v polygonech - odstranění části křivky	37
4.21	Porovnání výstupu pro přesnou a nepřesnou funkci 1	39
4.22	Porovnání výstupu pro přesnou a nepřesnou funkci 2	39
6.1	Okno po spuštění programu - záložka Functions	
6.2	Příklad textového zadání funkce	
6.3	Záložka Noise	

6.4	Záložka Symmetry Detector
6.5	Záložka Finish
6.6	Okno s vizualizací množiny bodů a jedné roviny symetrie . .

Seznam tabulek

4.1	Obecné funkce použité v experimentech a jejich rovnice. Zobrazeno pomocí https://www.desmos.com/calculator/ . . .	23
4.2	Polygony použité v experimentech	24
4.3	Obecné funkce použité v experimentech se středovou a rotační symetrií a jejich rovnice.	38

6 Přílohy

Příloha 1: Uživatelská příručka

Uživatelská příručka

Překlad a spuštění

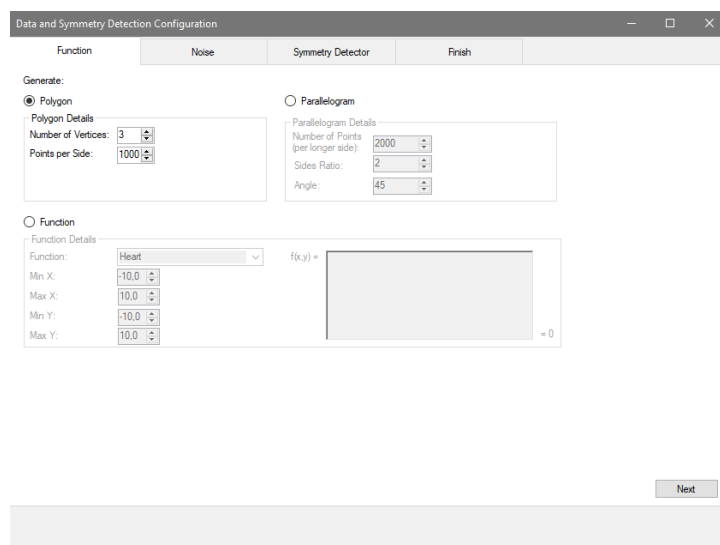
Program spustíte dvojitým kliknutím na soubor `GUIFormDataGenerator.exe`, který se nachází v adresáři `src/GUIFormDataGenerator/bin/x86/Release/`.

Překlad vyžaduje dva nástroje: Nástroj `make` (testováno na verzi 4.2.1) a nástroj `MSBuild` (testováno na verzi 15.9.21).

Pro překlad se pomocí příkazové řádky navigujte do adresáře `src/GUIFormDataGenerator`. Zde zadejte příkaz `make`. Provede se překlad. Pro spuštění můžete zadat příkaz `make run` nebo použít postup popsany výše.

Obsluha programu

Po spuštění programu se zobrazí okno přepnuté na záložku *Functions* (viz obr 6.1). Zároveň se otevře systémová konzole, do které budou po spuštění generování a/nebo detekce symetrie vypisovány podrobné informace o běhu programu.



Obrázek 6.1: Okno po spuštění programu - záložka Functions

Záložka Functions

V záložce *Functions* (viz obr 6.1) lze zvolit, jaká křivka bude generována. Na výběr je mezi třemi typy funkcí - polygonem (*Polygon*), rovnoběžníkem (*Parallelogram*) a obecnou funkcí (*Function*). Pro výběr přepnete příslušný přepínač.

Generování polygonu

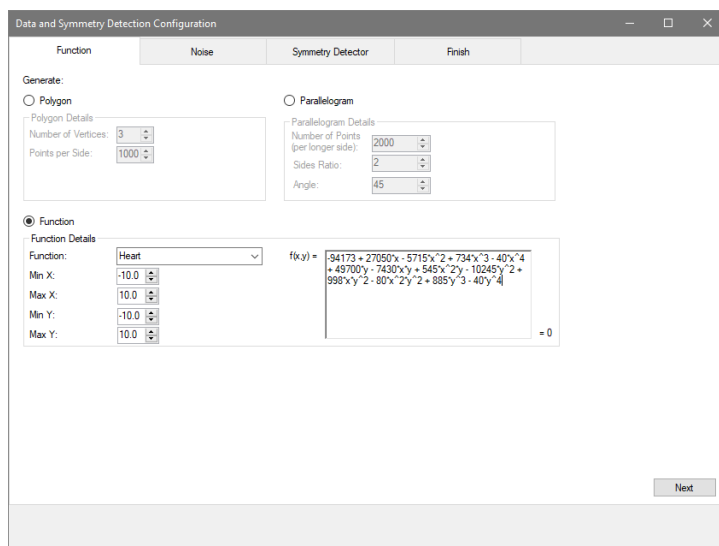
Při výběru přepínače *Polygon* se zpřístupní sekce *Polygon Details*. Zde lze konfigurovat, kolik vrcholů má generovaný polygon mít a také kolik bodů mají mít jednotlivé strany.

Generování rovnoběžníku

Při výběru přepínače *Parallelogram* se zpřístupní sekce *Parallelogram Details*. Zde lze konfigurovat, kolik bodů má mít delší strana rovnoběžníku, v jakém poměru mají strany být a jaký mají svírat úhel.

Generování obecné funkce

Při výběru přepínače *Function* se zpřístupní sekce *Function Details*. V rozbalovacím seznamu je na výběr z několika funkcí, které mohou být generovány, ale je možné zadat funkci i textově (implicitně) do textového pole vpravo. Pokud je textové pole neprázdné, generuje se funkce zadaná v textovém poli (viz obr. 6.2 pro příklad textového zadání funkce).



Obrázek 6.2: Příklad textového zadání funkce

Dále je potřeba specifikovat maximální oblast na osách x i y, ve které má být funkce generována. To se provede pomocí ovládacích prvků nacházejících se pod rozbalovacím seznamem.

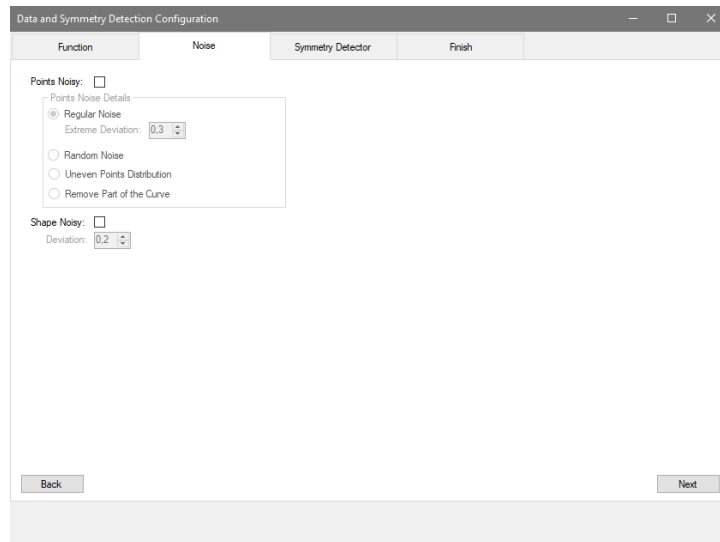
Pozn.: Vyhodnocování matematického výrazu z textu je zajištěno pomocí knihovny C# Eval Expression¹. Společnost vyvíjející tuto knihovnu

¹Viz <https://eval-expression.net/>.

pro ni neposkytuje bezplatnou licenci, proto je použita pouze její zkušební verze. Pokud během používání programu zkušební doba vyprší, knihovna při pokusu o její použití vyvolá výjimku. Potom je potřeba aktualizovat na poslední verzi knihovny. To lze provést například takto: Otevřít řešení (src/Framework.sln) v IDE Visual Studio, kliknout pravým tlačítkem myši na projekt FunctionsDataGenerator → Manage NuGet Packages... → v záložce Installed aktualizovat balík Z.Expressions.Eval.

Záložka Noise

V záložce *Functions* (viz obr 6.3) lze zvolit, jestli má být do generovaných bodů zanesen šum. Na výběr je mezi dvěma typy šumu - šum zanesený do vygenerovaných bodů (*Points Noisy*) a šum zanesený do tvaru křivky (*Shape Noisy*). Pro výběr zaškrtněte příslušné políčko.



Obrázek 6.3: Záložka Noise

Šum v bodech

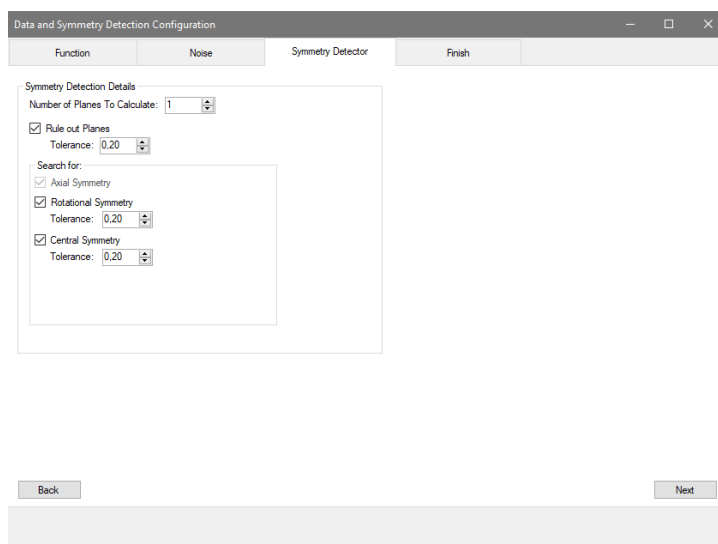
Po zaškrtnutí políčka *Points Noisy* se zpřístupní sekce *Points Noise Details*. V této sekci je na výběr ze čtyř druhů šumu, který může být do vygenerovaných bodů zanesen - pravidelný šum (*Regular Noise*), nepravidelný šum (*Random Noise*), nerovnoměrné rozložení bodů (*Uneven Points Distribution*) a odstranění části křivky (*Remove Part of the Curve*). Pro výběr přepněte příslušný přepínač. Při výběru pravidelného šumu se zpřístupní ovládací prvek, kterým lze specifikovat parametr maximální odchylky bodů.

Šum ve tvaru křivky

Po zaškrtnutí políčka *Shape Noisy* se zpřístupní ovládací prvek, kterým lze specifikovat parametr maximální odchylky vrcholu polygonu. Políčko *Shape Noisy* lze zaškrtnout pouze pokud je v záložce *Functions* vybrán Polygon.

Záložka Symmetry Detector

V záložce *Functions* (viz obr 6.4) lze konfigurovat, kolik os (resp. ploch) má software pro detekci symetrie počítat (*Number of Planes to Calculate*). Dále lze zvolit, jestli mají být kontrolováno, zda jsou nalezené osy skutečně osami symetrie (*Rule out Planes*). Nakonec je možné zvolit, jestli má být hledána rotační symetrie (*Rotational Symmetry*) nebo středová symetrie (*Central Symmetry*), u obou typů je pak potřeba pomocí zpřístupněných ovládacích specifikovat toleranci pro dané transformace bodů.



Obrázek 6.4: Záložka Symmetry Detector

Záložka Finish

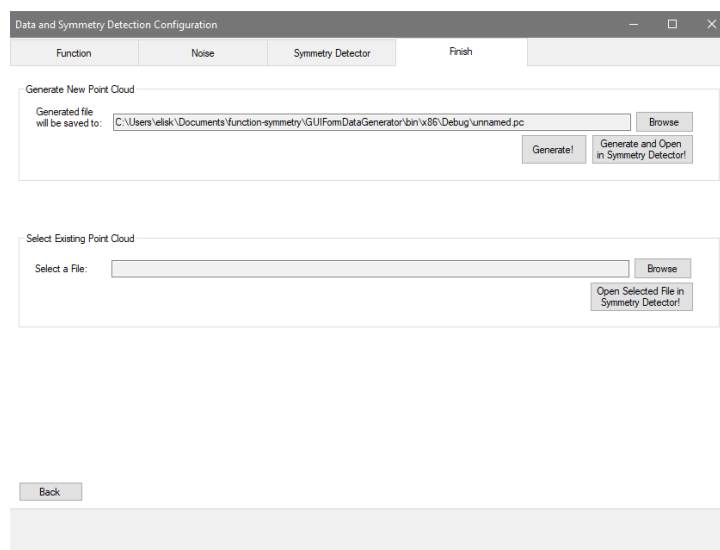
V záložce *Finish* (viz obr 6.5) v sekci *Generate New Point Cloud* lze pomocí tlačítka *Browse* zvolit cestu, kam bude uložena množina vygenerovaných bodů. Po zvolení cesty lze pomocí tlačítka *Generate!* pouze nechat vygenerovat body a uložit do souboru nebo tlačítkem *Generate and Open in Symmetry Detector!* vygenerovat body a rovnou detekovat symetrii tak,

jak bylo nastaveno v záložce *Symmetry Detector* a následně zobrazit pomocí vizualizační knihovny SlimDX.

Pozn.: Tato knihovna někdy způsobí pád programu, obvykle při pokusu o opakované generování množiny bodů se změněnými parametry. Příčina tohoto problému nebyla odhalena, proto problém nebyl odstraněn.

V sekci *Select Existing Point Cloud* lze pomocí tlačítka *Browse* zvolit cestu k dříve vygenerovanému souboru a pomocí tlačítka *Open Selected File in Symmetry Detector!* detekovat symetrii v tomto objektu a následně zobrazit pomocí vizualizační knihovny SlimDX.

Soubory s vygenerovanými body jsou ukládány s příponou *.pc* a tato přípona je také jediná možná pro načítané soubory.

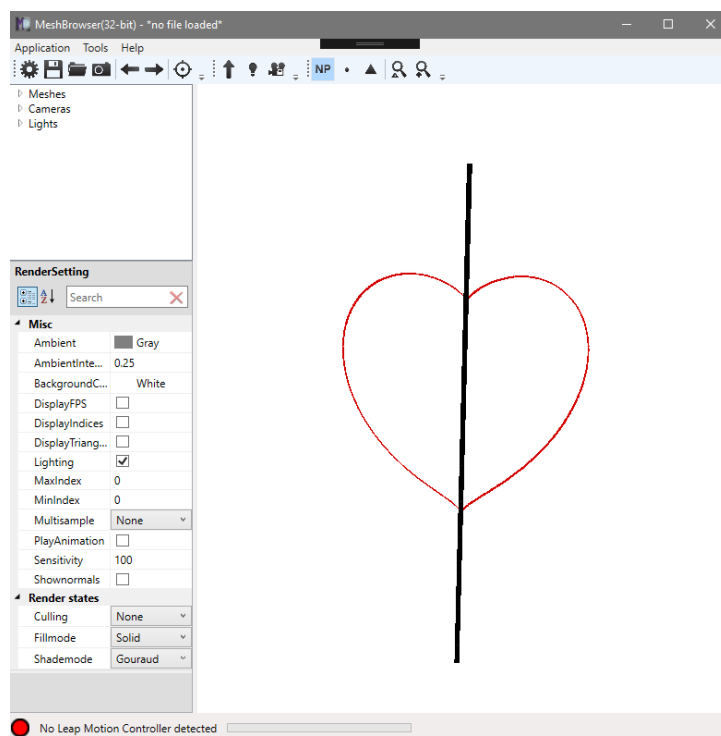


Obrázek 6.5: Záložka Finish

Ovládání vizualizačního nástroje

Při požadavku zobrazení vygenerované množiny bodů se otevře nové okno (viz obr. 6.6).

Držením levého tlačítka myši a tažením lze objekt otáčet. Pokud budete zároveň držet klávesu L, bude se nastavovat světlo. Držením pravého tlačítka myši a tažením lze objekt posouvat.



Obrázek 6.6: Okno s vizualizací množiny bodů a jedné roviny symetrie