



CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

DIPLOMOVÁ PRÁCE

Filtrace diskrétního povrchu

Martina Matýsková

kostliva@cmp.felk.cvut.cz

CTU–CMP–2007–07

5. února 2007

Lze získat na
<ftp://cmp.felk.cvut.cz/pub/cmp/articles/kostliva/Matyskova-TR-2007-07.pdf>

Školitel: Jana Kostlivá

This research was supported by the Czech Academy of Sciences under project 1ET101210406.

Research Reports of CMP, Czech Technical University in Prague, No. 7, 2007

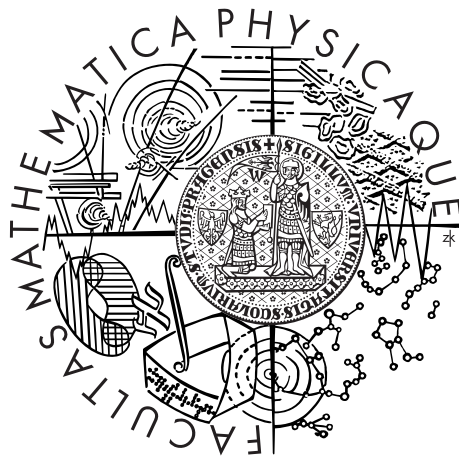
Published by

Centrum strojového vnímání, Katedra kybernetiky
Fakulta elektrotechnická ČVUT
Technická 2, 166 27 Praha 6

fax: (02) 2435 7385, tel: (02) 2435 7637, www: <http://cmp.felk.cvut.cz>

Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Martina Matýsková
Filtrace diskrétního povrchu

Katedra softwarového inženýrství
Vedoucí diplomové práce: Mgr. Jana Kostlivá
Studijní program: Informatika

Na tomto místě bych ráda poděkovala mojí vedoucí Mgr. Janě Kostlivé a panu Dr.Ing. Radimu Šárovi za jejich čas, připomínky a cenné rady.

Dále bych také ráda poděkovala svému příteli a rodičům za jejich podporu a trpělivost.

Prohlašuji, že jsem svou diplomovou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 14. prosince 2006

Martina Matýsková

Obsah

1	Úvod do problematiky	6
1.1	Motivace	6
1.2	Vyhlazování jako obecný problém	7
1.2.1	Triangulovaná síť	8
1.2.2	Vyhlázení objektu	8
1.2.3	Existující metody vyhlazování	9
1.2.4	Cíl práce	15
2	Vyhlazování 3D křivek	16
2.1	Vyhlázená křivka	16
2.2	Křivost na spojitě křivce	16
2.3	Definice křivosti	17
2.3.1	Vlastnosti vektoru křivosti \mathbf{H}_i a velikosti křivosti H_i . .	18
2.3.2	Proč pracovat s velikostí nové křivosti místo s vektorem	19
2.4	Definice neuniformity	21
2.5	Schéma vyhlazování	22
2.6	Algoritmus vyhlazování uzavřené křivky	22
2.6.1	Určení nových vrcholů křivky	23
2.7	Algoritmus vyhlazování podle kritéria křivosti	23
2.7.1	Výpočet vektoru křivosti \mathbf{H}_i^k a jeho velikosti H_i^k	23
2.7.2	Výpočet hodnoty nové křivosti H_i^{k+1}	24
2.7.3	Výpočet nového vrcholu \mathbf{a}_i^{k+1}	27
2.7.4	Speciální případy	31
2.8	Algoritmus vyhlazování podle kritéria neuniformity	32
2.8.1	Výpočet vektoru neuniformity \mathbf{U}_i^k	32
2.8.2	Výpočet nového vrcholu \mathbf{a}_i^{k+1}	32
2.9	Algoritmus pro vyhlazování neuzavřené křivky	33
2.10	Kritérium zastavení	34
2.11	Experimenty	34
2.11.1	Základní funkce	35
2.11.2	Robustnost a stabilita	38
2.11.3	Výběr lambdy pro křivost a neuniformitu	42
2.11.4	Volba iteračního schématu	48
2.11.5	Shrnutí	50

3	Vyhlazování uzavřených povrchů	52
3.1	Vyhlazený povrch	52
3.2	Křivost na spojitém povrchu	52
3.3	Definice křivosti	53
3.3.1	Vlastnosti vektoru křivosti \mathbf{H} a velikosti křivosti H	55
3.4	Definice neuniformity	55
3.5	Algoritmus pro vyhlazování uzavřeného povrchu	58
3.5.1	Určení nových vrcholů povrchu	59
3.6	Algoritmus vyhlazování podle kritéria křivosti	59
3.6.1	Výpočet vektoru křivosti \mathbf{H}^k a jeho velikosti H^k	59
3.6.2	Výpočet hodnoty nové křivosti H^{k+1}	60
3.6.3	Výpočet nového vrcholu \mathbf{a}^{k+1}	63
3.6.4	Speciální případy	65
3.7	Algoritmus vyhlazování podle kritéria neuniformity	65
3.7.1	Výpočet vektoru neuniformity \mathbf{U}^k	65
3.7.2	Výpočet nového vrcholu \mathbf{a}^{k+1}	66
3.8	Experimenty	67
3.8.1	Základní funkce	67
3.8.2	Robustnost a stabilita	70
3.8.3	Výběr lambdy pro křivost a neuniformitu	74
3.8.4	Volba iteračního schématu	79
3.8.5	Shrnutí	82
4	Vyhlazování povrchů s hranicemi	85
4.1	Rozdělení vrcholů povrchu a nalezení hranic	85
4.2	Algoritmus vyhlazení	85
4.3	Algoritmus vyhlazení hranice	86
4.4	Algoritmus vyhlazení povrchu	86
4.5	Průběh a ukončovací podmínky algoritmů	87
4.6	Experimenty	88
4.6.1	Základní funkce	89
4.6.2	Robustnost a stabilita	90
4.6.3	Reálná data	92
4.6.4	Shrnutí	95
5	Závěr	98
A	Implementace algoritmu vyhlazování uzavřené křivky	102
A.1	Vyhlazení křivky	102
A.2	Vstupní a výstupní parametry funkce	103
B	Implementace algoritmu vyhlazování povrchu	105
B.1	Vyhlazení povrchu	105
B.2	Vstupní a výstupní parametry funkce	107

Název práce: Filtrace diskrétního povrchu

Autor: Martina Matýsková

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: Mgr. Jana Kostlivá, Centrum strojového vnímání, Katedra kybernetiky, Fakulta elektrotechnická, České vysoké učení technické

e-mail vedoucího: kostliva@cmp.felk.cvut.cz

Abstrakt: Tato práce řeší úlohu vyhlazování povrchu s hranicí, který je reprezentován triangulovanou sítí. Motivací je metoda 3D rekonstrukce, jejíž výstupem je povrch reprezentovaný právě triangulovanou sítí. Takto vzniklá síť obsahuje šum, který je nutné vyfiltrovat. Nestací se však zabývat jen šumem, ale také regularitou sítě. Problém řešíme na diskrétním povrchu, kde pro odstranění šumu definujeme křivost a pro dosažení regularity sítě neuniformitu. Problém vyhlazení řešíme diskrétní difusí. V textu je popsáno vyhlazování křivek, uzavřených povrchů a povrchů s hranicemi. Text je doplněn o výsledky experimentů na datech, které jsou zpracovány navrženým algoritmem. Experimentálně bylo ověřeno, že výstupem jsou vyhlazené objekty, algoritmus je robustní, stabilní a v průběhu vyhlazování objektů nedochází k jejich smršťování, přičemž jejich základní tvar je zachován.

Klíčová slova: filtrace povrchu, vyhlazování sítě, aproximace křivosti

Title: Fairing of discrete surfaces

Author: Martina Matýsková

Department: Department of Software Engineering

Supervisor: Mgr. Jana Kostlivá, Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University

Supervisor's e-mail address: kostliva@cmp.felk.cvut.cz

Abstract: This thesis proposes a new method for fairing of surfaces, which are represented by a triangulated mesh. Motivation for our task is a 3D reconstruction, which results in a triangulated mesh. Such mesh is typically corrupted by noise (due to data and processing inaccuracy), which is required to be faired out. We solve the problem on a discrete surface, where two criteria for optimization are defined: surface curvature and non-uniformity. The curvature deals with surface smoothness, while the non-uniformity with surface regularity. The problem of fairing is solved by a discrete diffusion. Since our aim is to fair surfaces with boundaries, first we describe the fairing of curves, then fairing of closed surfaces, and finally fairing of surfaces with boundary (which is in fact a combination of the former ones). We have tested the algorithm's performance on synthetic as well as real data with good results. The experiments showed that our proposed algorithm is robust, stable and does not shrink an object, while its main surface characteristics are preserved.

Keywords: surface fairing, mesh smoothing, approximation of curvature

Kapitola 1

Úvod do problematiky

Kapitola popisuje motivaci pro tuto práci, úvod do problematiky vyhlazování objektů a stručný přehled metod zabývajících se tímto problémem.

1.1 Motivace

Motivací ke vzniku této práce je metoda, která se zabývá 3D rekonstrukcí objektů. Podrobný popis metody je v článku [4]. V této kapitole popíšeme 3D rekonstrukci velmi zjednodušeně.

Metoda 3D rekonstrukce řeší rekonstrukci modelu scény na základě množiny nasnímaných fotografií. Vstupem metody je množina nekalibrovaných fotografií objektu, který bude rekonstruován. Nekalibrovaná fotografie je taková, u které není známa pozice a nastavení fotoaparátu. Na pořadí fotografií nezáleží. Čím více je z objektu na fotografiích zachyceno a co možná pod nejvíce úhly, tím lepší kvalitu výsledku můžeme dosáhnout. Také není nutné, aby byly na vstupu jen fotografie zachycující objekt ve stejném měřítku. To je možné využít v případě, kdy objekt obsahuje složitější části, které mohou být nafoceny ve větším rozlišení. Tyto fotografie jsou pak využity pro přesnější rekonstrukci.

Prvním krokem je nalezení korespondujících oblastí mezi dvojicemi fotografií. To je provedeno pomocí porovnávání maximálně stabilních extrémálních oblastí fotografie [12]. Pro každou dvojici fotografií je současně s nalezením korespondencí spočítána i odpovídající epipolární geometrie.

Druhým krokem je výpočet konzistentních projekčních matic kamer [3]. Ke každé fotografii je hledána odpovídající projekční matice kamery. Pomocí projekční matice kamery lze spočítat přímku v prostoru, na které leží bod promítnutý do příslušného pixelu fotografie. Všechny projekční matice kamer pro zadanou sadu fotografií jsou hledány současně metodou faktorizace.

Třetím krokem je nalezení hustých korespondencí [10]. Pro každou dvojici fotografií je hledána korespondence mezi jednotlivými pixely.

Čtvrtým krokem je rekonstrukce 3D bodů z vyhledaných korespondencí. V případě přesné geometrie vznikne prostorový bod z každé korespondence

dvou pixelů jako průnik přímek, které určují, kde se může promítaný bod nacházet. V tomto případě je však geometrie nepřesná a přímky se neprotínají, proto je prostorový bod určen jako místo, ve kterém se přímky nejvíce přiblížily. Takto vzniklých bodů je velké množství, a proto je nutné reprezentovat 3D objekt jiným způsobem. Reprezentace byla zvolena pomocí šupin [14], které jsou lokálním odhadem hustoty bodů, tedy střední hodnotou a kovarianční maticí. Šupiny jsou zobrazovány jako elipsy odpovídající hlavní rovině daného kovariančního elipsoidu.

Reprezentace pomocí šupin není vhodná pro další práci s objektem (nejsou známy vztahy mezi jednotlivými šupinami, atd.). Proto je převedena na reprezentaci pomocí triangulované sítě (viz.1.2.1). Algoritmus, pomocí něhož je možné převod provést, je popsán v [1]. Vzniklá triangulovaná síť obsahuje šum a je nerovnoměrná. Proto je nutné ji před dalším použitím vyhladit. Řešení úlohy vyhlazení povrchu zadaného triangulovanou sítí je cílem této práce.

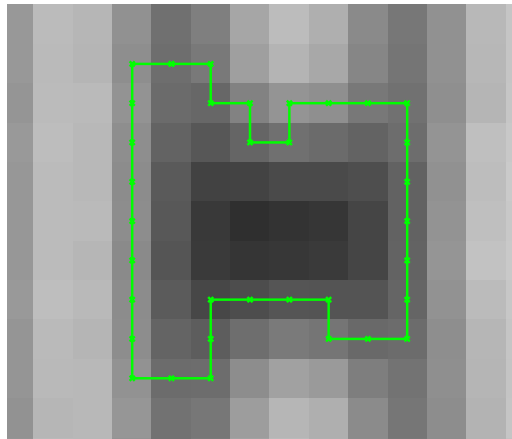
1.2 Vyhlazování jako obecný problém

Úloha vyhlazení zadané triangulované sítě není jen problémem v již zmiňovaném algoritmu 3D rekonstrukce. Existuje mnoho dalších aplikací, s nimiž je také spojena problematika vyhlazování. Pro příklad můžeme uvést 3D scannery na principu MRI (magnetické rezonance), CT (tomografu) nebo hloubkového laseru (range finder). Zde je triangulovaný povrch nejprve získán trasovacím algoritmem jako například Marching Tetrahedra [13] nebo starším Marching Cubes [13].

Všechna data, která jsou pomocí metod získána, bez ohledu na kvalitu metody, obsahují šum. Vyskytující se šum může být náhodný nebo diskretizační. Diskretizační šum vzniká například při vytváření sítě z dat reprezentovaných voxely, pomocí již zmíněných metod Marching Tetrahedra a Marching Cubes. Příklad diskretizačního šumu pro 2D případ je na obrázku 1.1. Šum obsažený v datech je nežádoucí, a proto je nutné jej před dalším zpracováním odstranit. Z toho důvodu je nezbytné zabývat se algoritmy, které síť vyhlazují.

V případě, že je takový algoritmus nalezen, je možné jej využít také k řešení úlohy takzvaného blendingu neboli natažení hladké sítě (tedy s co nejmenší energií) mezi jednotlivými objekty. To lze využít například pro napojení několika těles nebo pro vytvoření spoje v místě, kde jsou zadaná tělesa proniknuta.

V následující části textu se budeme zabývat formalizací problému.



Obrázek 1.1: Příklad diskretizačního šumu pro dvourozměrný případ. Obrázek je reprezentován pixely, jejichž hodnota je ve stupních šedi. Z části dat obrázku je vytvořena diskrétní křivka, která je kvůli pixelům „zubatá“.

1.2.1 Triangulovaná síť

Diskrétní povrch budeme reprezentovat pomocí triangulované sítě. Triangulovanou síť nazveme komplex (viz. obrázek 1.2), který je definován jako sjednocení následujících simplexů:

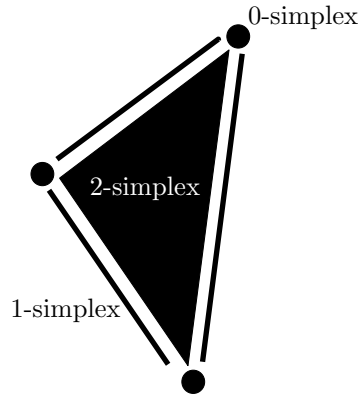
- 0-simplex – bod
- 1-simplex – úsečka bez koncových bodů
- 2-simplex – trojúhelník bez hran a bodů

Každá úsečka je ukončena bodem na obou koncích. Úsečku budeme nazývat *hranou* a bod *vrcholem*. Vrchol w je *bezprostředním sousedem* vrcholu v , právě když existuje hrana mezi vrcholy v a w . Každý trojúhelník je obklopen třemi vrcholy a třemi hranami. Předpokládáme, že každá hrana sousedí s nejvýše dvěma trojúhelníky. Pokud sousedí pouze s jedním, pak je hrana *hraniční*.

1.2.2 Vyhlazení objektu

Cílem je vytvořit algoritmus, který posunem vrcholů vytvoří lépe vyhlazený objekt. Pro diskrétně zadaný povrch neexistuje žádná obecná definice, která by říkala, zda je či není povrch vyhlazený. Proto, chceme-li definovat úlohu vyhlazení, musíme takovou definici zvolit. Volbu provedeme tak, aby odpovídala intuitivní představě o vyhlazeném povrchu. Na diskrétním triangulovaném povrchu nepostačí filtrovat normálový šum, ale také regularitu sítě. Obě složky se navzájem ovlivňují, a proto je třeba současná optimalizace.

Úkolem vyhlazovacích algoritmů je tedy modifikovat vstupní objekt tak, aby byl z objektu odstraněn převážně šum a bylo docíleno zrovnoměnění



Obrázek 1.2: Příklad sjednocení simplexů.

triangulace. Na druhou stranu od algoritmu vyžadujeme, aby v průběhu vyhlazování zachoval tvar a velikost objektu.

Šum se projevuje jako vysoká křivost, proto budeme vyhlazování provádět jako snižování křivosti v místech, kde je vysoká. V průběhu vyhlazení dochází i k odstranění ostrých hran sítě. Docílit však vyhlazení povrchu tak, aby k tomuto jevu nedocházelo, je velmi složité.

V dalších kapitolách se budeme zabývat již existujícími metodami vyhlazování, jejich rozdělením a případně i zhodnocením.

1.2.3 Existující metody vyhlazování

V poslední době bylo zveřejněno velké množství článků, které se zabývají vyhlazováním povrchu. Existující metody rozdělíme do čtyř skupin, které jsou podrobněji popsány níže:

1. Algoritmy založené na definici problému na spojitě zadaném povrchu.
2. Algoritmy založené na definici problému na diskrétně zadaném povrchu.
3. Metody vycházející z postupů pro zpracování obrazu.
4. Jiné metody se specifickým přístupem.

První dvě skupiny spojuje základní myšlenka minimalizovat povrchovou energii. Definice povrchové energie pro spojitou plochu vychází z hlavních křivostí, případně z jejich kombinací jako je Gaussova a střední křivost.

Gaussova křivost bývá označována K a definována jako

$$K = \kappa_1 \kappa_2, \quad (1.1)$$

střední křivost bývá označována jako H a definována jako

$$H = \frac{\kappa_1 + \kappa_2}{2}, \quad (1.2)$$

kde κ_1 a κ_2 je označení pro hlavní křivosti povrchu. Hlavní křivosti jsou minimem a maximem normálové křivosti κ_n (viz. [2]).

Hlavní křivosti jsou definovány jen pro spojitý povrch. Pro diskrétně zadaný povrch je tedy nutné definice energie vycházející z hlavních křivosti pro další výpočet nějakým způsobem aproximovat. Možné aproximace křivosti a energie jsou uvedeny v následujících odstavcích.

Algoritmy založené na definici problému na spojitě zadaném povrchu

Algoritmy jsou založeny na myšlence definovat řešení vyhlazování na spojitě zadaném povrchu a toto řešení zvolenou aproximací převést na řešení pro diskrétně zadaný povrch.

Řešení pomocí povrchové energie, která vychází z definic hlavních křivosti a jejich kombinací ((1.1) a (1.2)), vede na nelineární rovnici [7], jejíž řešení je komplikované a časově náročné. Častěji jsou v praxi využívány různé zjednodušené formy. Velmi často se pro minimalizaci povrchové energie vychází z rovnic:

membrane

$$E_1(X) = \frac{1}{2} \int_S X_u^2 + X_v^2 dudv, \quad (1.3)$$

thin plate

$$E_2(X) = \frac{1}{2} \int_S X_{uu}^2 + 2X_{uv}^2 + X_{vv}^2 dudv, \quad (1.4)$$

kde S je parametricky zadaný povrch a $X = X(u, v)$ je bod na povrchu S . Výraz $X_{uu} = \frac{\partial^2 X}{\partial u^2}$, $X_{vv} = \frac{\partial^2 X}{\partial v^2}$ a $X_{uv} = \frac{\partial^2 X}{\partial uv}$.

Minimalizace rovnice *membrane* minimalizuje zadaný povrch, tím ale také dochází k jeho postupnému smršťování. Toto chování není pro vyhlazování vhodné. Rovnice energie *thin plate* aproximuje křivost povrchu, ale nevýhodou je, že její hodnota závisí na parametrizaci povrchu. Viz. [7, 18].

Existuje více přístupů, které popisují jak dosáhnout minimalizace zvolené energie. Jednou z možností je aproximovat rovnici energie pro diskrétně zadaný povrch (například viz. [9]). Hledání minima takové formy vede na soustavu rovnic. Tento přístup je použit například v [16] a v [9] je součástí myšlenky celého algoritmu vyhlazování. Velikost takto vzniklé soustavy většinou odpovídá počtu vrcholů sítě a proto je řešení náročné. Z toho důvodu se soustavy často řeší za pomoci Gauss-Seidelovy metody, která v postupných krocích zlepšuje přesnost řešení.

Dalším možným řešením je využití vlastností Laplaceanu (budeme označovat jako $L(X)$) a to v případě, kdy nutná podmínka pro minimum zvolené energie je ve formě Laplaceovy rovnice

$$L(X) = 0. \quad (1.5)$$

V literatuře je z této rovnice setavena rovnice difuze

$$\frac{\partial X}{\partial t} = \lambda L(X), \quad (1.6)$$

kde ∂t je změna v čase, $X(0) = X$ a $0 < \lambda < 1$. Potom

$$\lim_{t \rightarrow \infty} \frac{\partial X}{\partial t} = 0 \quad (1.7)$$

je řešením (1.5).

Převedení problému ze spojitého na diskrétní je provedeno až ve chvíli, kdy je nutné znát hodnotu Laplaciánu. Tedy aproximována není rovnice energie, ale přímo hodnota Laplaciánu ve vrcholu povrchu. Výběr možné aproximace zaleží na autorovi algoritmu.

Příklad řešení pomocí Laplaciánu je možné nalézt v [5]. Jako výchozí rovnice je zvolena rovnice (1.3). Aproximace $L(X)$ ve vrcholu i je provedena takzvaným deštníkovým operátorem [5]:

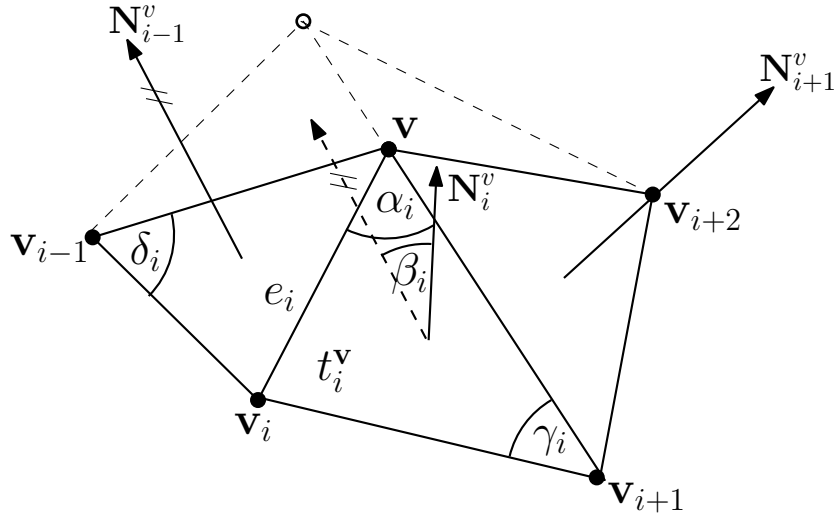
$$L(X_i) = \frac{1}{N} \sum_{j=1}^N X_j - X_i, \quad (1.8)$$

kde X_i je aktuální vrchol triangulované sítě, N je počet bezprostředních sousedů X_i a X_j je j -tý bezprostřední soused vrcholu X_i .

Nevýhodou algoritmů založených na využití Laplaciánu je fakt, že k dosažení dobrého výsledku na objektech, které jsou hodně nebo nerovnoměrně zašuměny, je nutné použití velkého množství iterací difusního kroku. Navíc u některých objektů dochází ke smršťování vyhlazovaného objektu a k přílišnému vyhlazení detailů. Na druhou stranu je nutné zdůraznit, že výhodou algoritmu je snadné řešení a implementace a velkou výhodou je také lineární složitost algoritmu.

Algoritmy založené na definici problému na diskrétně zadaném povrchu

Jak již bylo zmíněno, hlavní křivosti, ze kterých vychází Gaussova (1.1) a střední (1.2) křivost, jsou definovány jen pro spojitý povrch. Existují algoritmy, které se nezabývají vyhlazováním na spojitém povrchu, ale problém řeší přímo na povrchu diskrétním. Součástí algoritmu je definice kritéria pro diskrétní povrch, které jej popisuje. Kritérium může být aproximací některé z křivostí definovaných na spojitém povrchu nebo zvoleno nezávisle na křivosti. Nevýhodou takto definovaných křivostí je v převážné většině definic složitost jejich výpočtu. Uvedeme několik možných aproximací a pro celou podkapitulu zavedeme jednotné značení. Aktuální vrchol sítě budeme označovat \mathbf{v} a bezprostřední sousedy vrcholu \mathbf{v} jako $\{\mathbf{v}_i\}_{i=1}^n$. Trojúhelníky, které obsahují vrchol \mathbf{v} označíme jako $\{t_i^{\mathbf{v}}\}_{i=1}^n$, kde $t_i^{\mathbf{v}} =$



Obrázek 1.3: Zobrazení definovaných úhlů pro vrchol \mathbf{v} .

$(I_{\mathbf{v}_i}, I_{\mathbf{v}}, I_{\mathbf{v}_{i+1}})$ a $I_{\mathbf{v}}$ je index vrcholu \mathbf{v} v seznamu všech vrcholů. Hranu mezi vrcholy \mathbf{v} a \mathbf{v}_i označíme jako e_i a $\|e_i\|$ je velikost této hrany. Dále zavedeme některé úhly (viz. obrázek 1.3) za předpokladu, že $\mathbf{v}_i \neq \mathbf{v}_{i+1}$ pro všechna i . Potom

$$\alpha_i = \angle(\mathbf{v}_i, \mathbf{v}, \mathbf{v}_{i+1}), \quad (1.9)$$

$$\beta_i = \angle(\mathbf{N}_{i-1}^{\mathbf{v}}, \mathbf{N}_i^{\mathbf{v}}), \quad (1.10)$$

$$\gamma_i = \angle(\mathbf{v}, \mathbf{v}_{i+1}, \mathbf{v}_i), \quad (1.11)$$

$$\delta_i = \angle(\mathbf{v}, \mathbf{v}_{i-1}, \mathbf{v}_i), \quad (1.12)$$

kde $\mathbf{N}_i^{\mathbf{v}}$ je normála trojúhelníku $t_i^{\mathbf{v}}$ a lze ji spočítat jako

$$\mathbf{N}_i^{\mathbf{v}} = \frac{(\mathbf{v}_i - \mathbf{v}) \times (\mathbf{v}_{i+1} - \mathbf{v})}{\|(\mathbf{v}_i - \mathbf{v}) \times (\mathbf{v}_{i+1} - \mathbf{v})\|}. \quad (1.13)$$

Střední křivost lze aproximovat jako velikost toku křivosti [5]:

$$H = \left\| \frac{\nabla A}{2A} \right\| = \left\| \frac{1}{4A} \sum_{i=1}^n (\cot \gamma_i + \cot \delta_i)(\mathbf{v}_i - \mathbf{v}) \right\|, \quad (1.14)$$

kde A je rovno součtu ploch trojúhelníků $\{t_i^{\mathbf{v}}\}_{i=1}^n$.

Další možností je aproximace podle Gauss-Bonnetova schématu. Odvození lze nalézt v [6].

Gaussova křivost pro diskrétní povrch je rovna

$$K = \frac{2\pi - \sum_{i=1}^n \alpha_i}{\frac{1}{3}A} \quad (1.15)$$

a střední křivost je rovna

$$H = \frac{\frac{1}{4} \sum_{i=1}^n \|e_i\| \beta_i}{\frac{1}{3} A}, \quad (1.16)$$

kde A je rovno součtu ploch trojúhelníků $\{t_i^v\}_{i=1}^n$.

Jako poslední uvedeme aproximaci, která je provedena pomocí paraboloidu [17]. Aproximace vychází z myšlenky, že hlavní křivosti na spojitě zadaném povrchu jsou považovány za shodné s hlavními křivostmi oskulačního paraboloidu. V aktuálním vrcholu \mathbf{v} je spočtena normála jako

$$\mathbf{N}_v = \frac{\bar{\mathbf{N}}_v}{\|\bar{\mathbf{N}}_v\|}, \quad (1.17)$$

kde

$$\bar{\mathbf{N}}_v = \frac{1}{n} \sum_{i=1}^n \mathbf{N}_i^v \quad (1.18)$$

a \mathbf{N}_i^v je spočtena podle (1.13). Definujeme soustavu tak, že vrchol \mathbf{v} zvolíme jako počátek souřadné soustavy a vyjádříme souřadnice bezprostředních sousedů v v této souřadné soustavě. Směr osy z zvolíme shodný se směrem normály \mathbf{N}_v a osu x a y zvolíme libovolně. Rovnice oskulačního paraboloidu v takto zadané soustavě má tvar

$$z = ax^2 + bxy + cy^2. \quad (1.19)$$

Koeficienty a , b a c určíme pomocí metody nejmenších čtverců tak, aby paraboloid co nejlépe aproximoval plochu určenou vrcholem \mathbf{v} a jeho bezprostředními sousedy $\{\mathbf{v}_i\}_{i=1}^n$. Gaussova křivost lze vyjádřit jako

$$K = 4ac - b^2 \quad (1.20)$$

a střední křivost jako

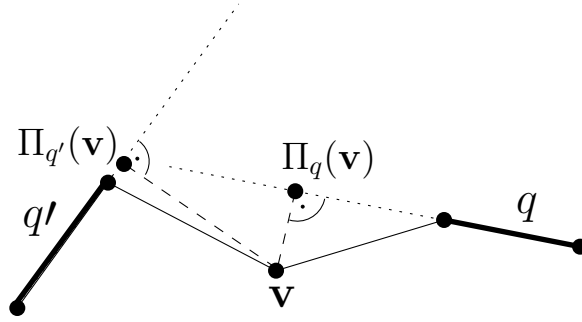
$$H = a + c. \quad (1.21)$$

Metody vycházející z postupů pro zpracování obrazu

Metody vycházejí z algoritmů vhodných pro zpracování 2D obrazu. Cílem je převedení problematiky odstranění šumu z obrazu zadaného množinou pixelů, kde každý pixel má přiřazenou určitou hodnotu, na problém vyhlazení triangulovaného povrchu.

Principem těchto algoritmů je nalezení nových normál (pro obrazová data je ekvivalentem hodnota pixelu) trojúhelníků a následná změna polohy vrcholů sítě tak, aby se normály nově vzniklých trojúhelníků rovnaly těm, které byly vypočteny. Publikovány byly například metody na principu hlavního a medián filtru [19], případně MMSE filtr [11].

Tyto metody fungují jen na velmi specifických datech. Například MMSE filtr je vhodný jen pro povrchy, které obsahují malý šum a navíc jsou kulovitěho tvaru. Kvůli těmto omezením jsou algoritmy vhodné jen pro velmi úzké spektrum použití a pro vyhlazování obecných objektů se nehodí.



Obrázek 1.4: Predikce vrcholu.

Jiné metody se specifickým přístupem

Existuje mnoho dalších metod, které nelze jednoznačně zařadit do konkrétní kategorie algoritmů. Jedna z těchto metod je uvedena v [8]. Algoritmus je založen na principu nalezení nových pozic všech vrcholů sítě. Každá pozice vrcholu je hledána jako vážená suma z predikcí, které jsou určeny pomocí sousedních vrcholů aktuálního vrcholu. V praxi je predikce založena na trojúhelníkových sítích, které reprezentují přirozené tečné roviny k povrchu (viz. obrázek 1.4). Nový vrchol sítě je určen jako

$$\mathbf{v}' = \frac{1}{k(\mathbf{v})} \sum_{q \in S} \Pi_q(\mathbf{v}) a_q f(\|c_q - \mathbf{v}\|) g(\|\Pi_q(\mathbf{v}) - \mathbf{v}\|), \quad (1.22)$$

kde

$$k(\mathbf{v}) = \sum_{q \in S} a_q f(\|c_q - \mathbf{v}\|) g(\|\Pi_q(\mathbf{v}) - \mathbf{v}\|). \quad (1.23)$$

Písmeno \mathbf{v} označuje aktuální vrchol a \mathbf{v}' je nová pozice vrcholu \mathbf{v} . Písmeno S v tomto případě označuje všechny trojúhelníky sítě. Písmeno q označuje trojúhelník, jehož jedním vrcholem je sousední vrchol vrcholu \mathbf{v} a současně tento trojúhelník vrchol \mathbf{v} neobsahuje. Určení, jak velké okolí a tedy i počet predikcí je nutné k výpočtu aktuálního vrcholu, záleží na uživateli. Označení $\Pi_q(\mathbf{v})$ určuje průmět vrcholu \mathbf{v} do roviny určené trojúhelníkem q . Označení a_q je rovno obsahu trojúhelníka q a c_q označuje těžiště trojúhelníka q . Použité označení $\|c_q - \mathbf{v}\|$ je rovno vzdálenosti c_q a \mathbf{v} . Stejně tak i $\|\Pi_q(\mathbf{v}) - \mathbf{v}\|$. Jako g a f je použita Gaussova funkce.

Výhodou algoritmu, jak je uvedeno v [8], je zachování tvaru objektu i s jemnými detaily. Na druhou stranu k dosažení dobrých výsledků je nutné okolí počítaného souseda volit dosti veliké, což způsobuje zvýšení náročnosti algoritmu. Také je nutné zdůraznit, že samotný algoritmus není dostačující k úplnému vyhlazení velkých stejnorodých povrchů. Aby bylo možné takové povrchy dostatečně vyhladit, je před samotný algoritmus zařazen průchod (zahlazení), který počítá nové hodnoty normál trojúhelníků na podobném principu jako samotný algoritmus nové vrcholy. Normály pak

určují nové plochy příslušných trojúhelníků, které jsou dále zpracovávány algoritmem.

1.2.4 Cíl práce

Práce popsaná v následujících kapitolách si kladla za cíl prozkoumat chování algoritmu vyhlazování, který vychází ze spojení dvou již dříve použitých přístupů. Prvním je zavedení úlohy vyhlazení přímo pro diskrétní povrch. Tím se vyhneme již dříve popsaným problémům aproximace spojitě definovaného problému a náročnosti výpočtu, vznikajících v případě zavedení úlohy pro spojitý povrch. Druhým přístupem je využití difuze. V našem případě je to difuze skalární funkce, čímž odstraníme problém smršťování povrchu v průběhu vyhlazování.

Přesto, že aproximací křivostí pro diskrétní povrch bylo již navrženo mnoho, zvolili jsme vlastní definici vektoru křivosti

$$\mathbf{H} = k \frac{\nabla \sum_{i=1}^n A_i^2}{\sum_{i=1}^n A_i^2}, \quad (1.24)$$

kde n je počet trojúhelníků, které obsahují vrchol, pro který je počítána křivost, k je nezáporný skalár a A_i je rovno obsahu trojúhelníka i . Požadavkem na aproximaci křivosti byla jednoduchost výrazu a snadné odvozování dalších výpočtů. Použitím druhé mocniny jsme odstranili složité vzorce, na které vede výraz, kterým je aproximace vektoru křivosti inspirována. Vycházeli jsme z výrazu

$$\mathbf{H} = \frac{\nabla \sum_{i=1}^n A_i}{2 \sum_{i=1}^n A_i}. \quad (1.25)$$

Značení je shodné jako v (1.24) a podrobný popis lze nálezt v [5].

V následujícím textu je popsáno řešení úloh vyhlazení pro křivky, uzavřené povrchy a ohraničené povrchy.

Kapitola 2

Vyhlazování 3D křivek

Tato kapitola popisuje vyhlazování křivek v 3D prostoru.

2.1 Vyhlazená křivka

Cílem algoritmu je postupnými posuny jednotlivých vrcholů dané křivky docílit takového tvaru křivky, ve kterém je *vyhlazená*. Vyhlazením rozumíme optimalizaci dvou vlastností křivky: křivosti a neuniformity. Křivost udává křivost dané křivky a neuniformita nerovnoměrnost mezi vzdálenostmi vrcholu křivky k jeho bezprostředním sousedům. Řekneme, že křivka je ideálně vyhlazená, když má křivost ve všech vrcholech rovnu průměru křivostí bezprostředních sousedů a neuniformitu rovnu nule (vzdálenosti vrcholu ke všem bezprostředním sousedům vrcholu jsou shodné).

2.2 Křivost na spojitě křivce

Nechť S je C^2 spojitá křivka, která minimalizuje

$$J = \int \left(\frac{\partial H}{\partial s} \right)^2 ds, \quad (2.1)$$

kde H je velikost křivosti a ds je element délky. Funkce $H = H(s, t)$, kde s je parametrizace křivky a t je čas. Rovnice (2.1) je Dirichletův integrál [15]. Nutná podmínka pro extrém funkce J je ve tvaru Laplaceovy rovnice

$$\Delta H = \frac{\partial^2 H}{\partial s^2} = 0. \quad (2.2)$$

Funkce H je tedy harmonická funkce. Funkci H tedy můžeme nalézt jako řešení difusní rovnice

$$\frac{\partial H}{\partial t} = \lambda \frac{\partial^2 H}{\partial s^2}, \quad 0 < \lambda < 1, \quad (2.3)$$

kde λ je koeficient difuze, a pro kterou platí

$$\lim_{t \rightarrow \infty} \frac{\partial H}{\partial t} = 0 \quad (2.4)$$

a H je harmonická funkce.

Pro funkci H platí věta o aritmetickém průměru [15]:

Nechť funkce u je v oblasti Ω spojitá. Hodnoty u ve středu každé n -rozměrné koule $\bar{K} \in \Omega$ jsou rovny aritmetickému průměru hodnot na hranici \bar{K} tehdy a jen tehdy když u je harmonická.

My použijeme difusi pro vyhlazování křivky. Potom λ je rychlost vyhlazování. Pro účely vyhlazení nepotřebujeme úplnou konvergenci, protože pak bychom z každé uzavřené Jordanovské křivky po vyhlazení vytvořili kružnici a z každé neuzavřené úsečku.

Řekneme, že diskrétní křivka je zadaná posloupností vrcholů $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$.

Diskretizací věty o aritmetickém průměru dostaneme

$$\bar{H}_i - H_i = 0, \quad (2.5)$$

kde $\bar{H}_i = \frac{1}{2}(H_{i+1} + H_{i-1})$ je průměr okolních velikostí křivosti vrcholu \mathbf{a}_i . Předpokládáme uniformní okolí o poloměru jedna. Požadavek na uniformitu je realistický, protože bude existovat člen, který jej v průběhu konvergence zaručí.

Výraz $\frac{\partial^2 H}{\partial s^2}$ ve vrcholu \mathbf{a}_i můžeme aproximovat

$$\frac{\partial^2 H}{\partial s^2} \approx \bar{H}_i - H_i. \quad (2.6)$$

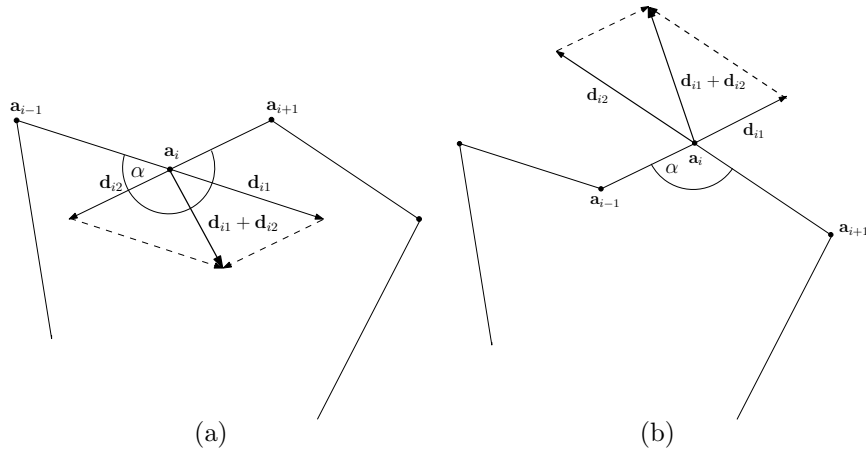
Dosazením výrazu (2.6) do výrazu (2.3) a nahrazením výrazu $\frac{\partial H}{\partial t}$ výrazem $H_i^{k+1} - H_i^k$ dostaneme definici pro výpočet nové křivosti ve vrcholu \mathbf{a}_i . Index k určuje iteraci. Potom

$$H_i^{k+1} = (1 - \lambda)H_i^k + \lambda\bar{H}_i^k, \quad 0 < \lambda < 1. \quad (2.7)$$

2.3 Definice křivosti

V průběhu vyhlazování křivky se pracuje s křivostí ve vrcholu křivky. Zavedeme vektor křivosti a uvedeme výpočet velikosti tohoto vektoru.

Řekneme, že $K = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$, kde $\mathbf{a}_i = [x_i, y_i, z_i]$, je uzavřená diskrétní křivka s n vrcholy. V následujícím textu budeme předpokládat, že křivka existuje v prostoru dimenze tři. Obecně však dimenze není omezena. Nechť vrcholy \mathbf{a}_{i-1} a \mathbf{a}_{i+1} jsou bezprostřední sousedi \mathbf{a}_i . Platí, že $\mathbf{a}_0 = \mathbf{a}_n$ a $\mathbf{a}_{n+1} = \mathbf{a}_1$. Označme $\mathbf{d}_{i1} = \mathbf{a}_i - \mathbf{a}_{i-1}$ a $\mathbf{d}_{i2} = \mathbf{a}_i - \mathbf{a}_{i+1}$.



Obrázek 2.1: Směr křivosti ve vrcholu \mathbf{a}_i . (a) Úhel α je nekonvexní. (b) Úhel α je konvexní.

Skaláry d_{i1} a d_{i2} jsou velikosti vektorů \mathbf{d}_{i1} a \mathbf{d}_{i2} a platí pro ně:

$$d_{i1} = \|\mathbf{d}_{i1}\| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2},$$

$$d_{i2} = \|\mathbf{d}_{i2}\| = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2}.$$

Pro účely této práce křivost křivky K ve vrcholu \mathbf{a}_i definujeme jako vektor $\mathbf{H}_i = [h_{i,x}, h_{i,y}, h_{i,z}]$, který spočítáme jako

$$\mathbf{H}_i = \frac{\nabla(d_{i1}^2 + d_{i2}^2)}{d_{i1}^2 + d_{i2}^2}, \quad (2.8)$$

kde $\nabla(d_{i1}^2 + d_{i2}^2) = \left(\frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial x_i}, \frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial y_i}, \frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial z_i} \right)$.

Po upravení čitatele výrazu vznikne konečná definice vektoru křivosti ve vrcholu \mathbf{a}_i křivky K . Tedy

$$\mathbf{H}_i = 2 \cdot \frac{\mathbf{d}_{i1} + \mathbf{d}_{i2}}{d_{i1}^2 + d_{i2}^2}. \quad (2.9)$$

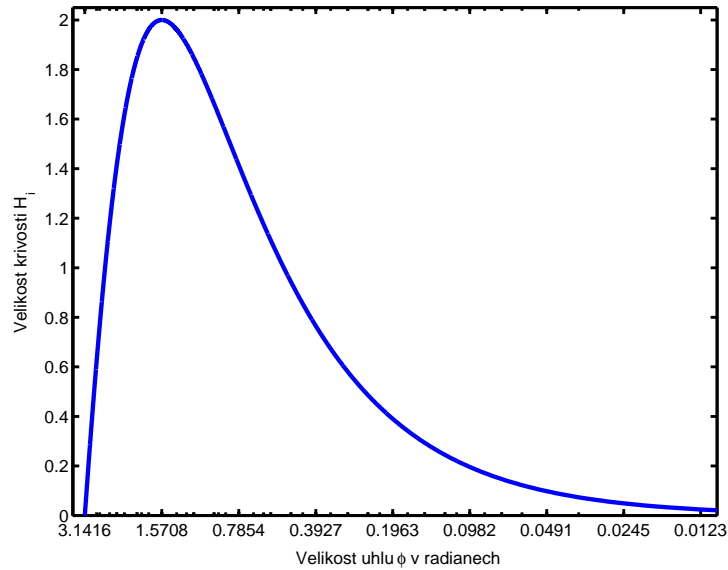
Směr vektoru křivosti je shodný se směrem vektoru $\mathbf{d}_{i1} + \mathbf{d}_{i2}$ (viz. obrázek 2.1(a) a 2.1(b)).

Velikost křivosti $H_i = \|\mathbf{H}_i\| = \sqrt{(h_{i,x})^2 + (h_{i,y})^2 + (h_{i,z})^2}$.

2.3.1 Vlastnosti vektoru křivosti \mathbf{H}_i a velikosti křivosti H_i

Zvolený vektor křivosti má vlastnosti, které jsou užitečné pro úlohu vyhlazování:

- Křivost $\mathbf{H}_i = \mathbf{0}$, právě když $\mathbf{d}_{i1} = -\mathbf{d}_{i2}$.
- Velikost křivosti H_i je invariantní k rotaci a translaci křivky.



Obrázek 2.2: Závislost velikosti křivosti H_i na úhlu φ v případě, kdy je $d_{i1} = d_{i2}$.

- Z definice \mathbf{H}_i plyne, že v případě pravidelného rovinného polygonu směřuje \mathbf{H}_i od středu polygonu a $H_i = \frac{1}{r}$, kde r je poloměr opsané kružnice polygonu.

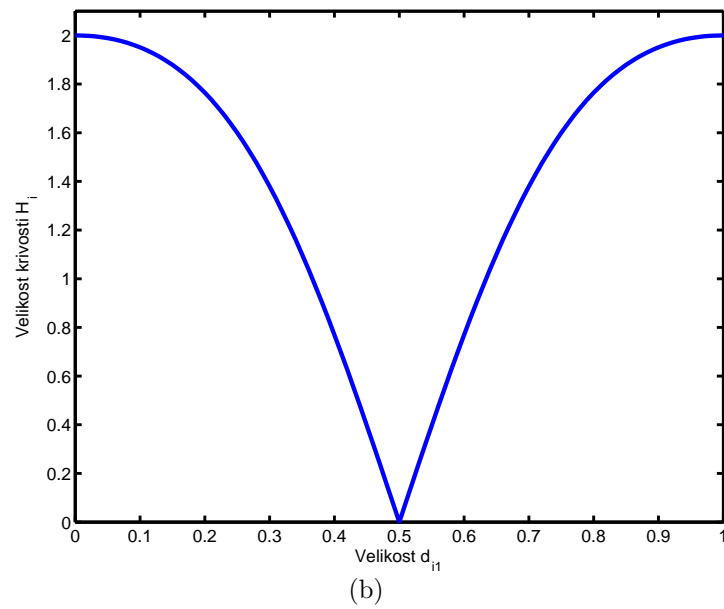
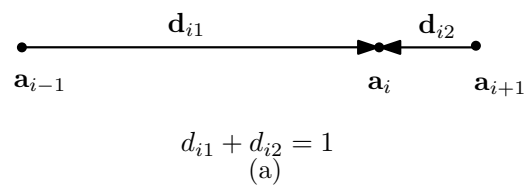
Důkaz: Odvodíme velikost hrany polygonu $d = 2r \sin(\frac{\pi}{n})$, kde n je počet vrcholů polygonu. Nechť $\mathbf{u}_1 = -\mathbf{d}_{i1} = \mathbf{a}_{i-1} - \mathbf{a}_i$ a $\mathbf{u}_2 = -\mathbf{d}_{i2} = \mathbf{a}_{i+1} - \mathbf{a}_i$. Pak $\|\mathbf{u}_1\| = d$ a $\|\mathbf{u}_2\| = d$. Zvolme $\mathbf{c} = \mathbf{u}_1 + \mathbf{u}_2$, pak $c = 2d \sin(\frac{\beta}{2})$, kde $\beta = \frac{2\pi}{n}$. Po dosazení za d dostaneme $c = 4r \sin^2(\frac{\pi}{n})$. Z definice (2.9) zapíšeme $H_i = 2 \frac{\|\mathbf{d}_{i1} + \mathbf{d}_{i2}\|}{d_{i1}^2 + d_{i2}^2} = \frac{2 \cdot \|(-\mathbf{u}_1) + (-\mathbf{u}_2)\|}{2d^2} = \frac{2 \cdot \|\mathbf{c}\|}{2d^2} = \frac{2 \cdot 4r \sin^2(\frac{\pi}{n})}{2 \cdot 4r^2 \sin^2(\frac{\pi}{n})} = \frac{1}{r}$.

□

- Velikost křivosti H_i není monotonní funkcí úhlu φ , kterým označujeme úhel, který svírají vektory \mathbf{d}_{i1} a \mathbf{d}_{i2} . Uvedeme případ pro $d_{i1} = d_{i2}$ a posun vrcholu \mathbf{a}_i ve směru $d_{i1} + d_{i2}$ (viz. obrázek 2.2).
- V případě, že \mathbf{d}_{i1} a \mathbf{d}_{i2} leží v jedné přímce, pak závislost H_i na poměru \mathbf{d}_{i1} a \mathbf{d}_{i2} je znázorněna na obrázku 2.3(a) a 2.3(b).

2.3.2 Proč pracovat s velikostí nové křivosti místo s vektorem

V případě, že bychom ve vzorci (2.7) použili místo velikosti křivosti vektor, docházelo by v průběhu vyhlazování ke smršťování křivky. Korektní chování algoritmu je zachovat velikost plochy, kterou daná křivka ohraničuje.



Obrázek 2.3: Závislost velikosti křivosti H_i na poměru velikostí vektorů \mathbf{d}_{i1} a \mathbf{d}_{i2} ležících v jedné přímce.

Příklad smršťování křivky, v případě použití vektoru křivosti, si ukážeme na vyhlazování pravidelného polygonu. Ve vrcholu pravidelného polygonu spočítáme vektor nové křivosti tak, že ve vzorci (2.7) nahradíme velikosti křivosti vektory křivosti. Nový vektor křivosti je menší než vektor staré křivosti, což má za následek posunutí vrcholu \mathbf{a}_i směrem ke středu polygonu. Takto posunutý vrchol způsobí smrštění polygonu.

V případě, že pro výpočet nové křivosti použijeme vzorec (2.7) a tedy spočteme novou křivost přímo jako velikost, je u pravidelného polygonu velikost nové křivosti H_i^{k+1} stejná jako velikost staré křivosti H_i^k . Vrchol \mathbf{a}_i křivky K se nepohne, což je korektní chování.

2.4 Definice neuniformity

V průběhu vyhlazování křivky se pracuje také s neuniformitou. S neuniformitou pracujeme analogicky jako s křivostí.

Zdefinujeme vektor neuniformity a výpočet velikosti tohoto vektoru ve vrcholu křivky.

Definice křivky a ostatních použitých veličin je uvedena v kapitole (2.3).

Označení \mathbf{P}_i označuje projekci do směru určeného vektorem $\mathbf{b}_i = \mathbf{a}_{i+1} - \mathbf{a}_{i-1}$. Projekci spočítáme:

$$\mathbf{P}_i = \frac{\mathbf{b}_i \mathbf{b}_i^\top}{\|\mathbf{b}_i\|^2}, \quad (2.10)$$

kde $\|\mathbf{b}_i\|$ je velikost vektoru \mathbf{b}_i .

Neuniformitu křivky K ve vrcholu \mathbf{a}_i definujeme jako vektor $\mathbf{U}_i = [u_{i,x}, u_{i,y}, u_{i,z}]$. Vektor neuniformity je rovnoběžný s tečným směrem ve vrcholu \mathbf{a}_i . Tečný směr ve vrcholu \mathbf{a}_i je určen vektorem \mathbf{b}_i a

$$\mathbf{U}_i = \frac{-\mathbf{P}_i(\nabla(d_{i1}^2 + d_{i2}^2))}{4}, \quad (2.11)$$

kde $\nabla(d_{i1}^2 + d_{i2}^2) = \left(\frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial x_i}, \frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial y_i}, \frac{\partial(d_{i1}^2 + d_{i2}^2)}{\partial z_i}\right)$.

Po upravení čitatele výrazu vznikne definice vektoru neuniformity ve vrcholu \mathbf{a}_i křivky K . Tedy

$$\mathbf{U}_i = \frac{-\mathbf{P}_i(\mathbf{d}_{i1} + \mathbf{d}_{i2})}{2}. \quad (2.12)$$

Velikost vektoru neuniformity určuje nepoměr vzdáleností vrcholu \mathbf{a}_i k jeho bezprostředním sousedům. Čím bude poměr vzdáleností d_{i1} a d_{i2} větší, tím bude větší i velikost vektoru neuniformity. V ideálním případě, kdy vrchol \mathbf{a}_i leží uprostřed mezi vrcholy \mathbf{a}_{i+1} a \mathbf{a}_{i-1} , je velikost vektoru neuniformity rovna nule.

Velikost vektoru neuniformity $U_i = \|\mathbf{U}_i\| = \sqrt{u_{i,x}^2 + u_{i,y}^2 + u_{i,z}^2}$.

Diskrétní difusi pro neuniformitu definujeme předpisem

$$U_i^{k+1} = (1 - \lambda_u)U_i^k, \quad (2.13)$$

kde $0 < \lambda_u < 1$, neboť chceme, aby v ekvilibriu byla neuniformita rovna nule.

2.5 Schéma vyhlazování

Pro vyhlazování křivky jsou použita dvě kritéria. První je kritérium křivosti a druhé kritérium neuniformity. Algoritmus pro vyhlazování pracuje s kritérii jednotlivě. To znamená, že kritérium, podle kterého se pracuje, je střídáno za druhé kritérium po určitém počtu kroků algoritmu. S kritérii pracujeme odděleně i přes to, že nejsou ortogonální, tedy změnou jednoho se změní i druhé. Experimentálně jsme ověřili, že je výhodnější pracovat s kritérii odděleně, než pracovat s nimi současně. Náročnost algoritmu se tím nezvyšuje. Srovnání různých typů schémat je v kapitole 2.11.4.

2.6 Algoritmus vyhlazování uzavřené křivky

Cílem algoritmu je postupným posunem vrcholů křivky dosáhnout nových konečných vrcholů křivky tak, aby křivost a neuniformita v těchto nových vrcholech byly v rovnováze (vrcholy se již nepohybují). Snahou algoritmu je také nezměnit plochu, kterou výchozí uzavřená křivka ohraničuje. Algoritmus pracuje odděleně s křivostí a neuniformitou. Zpracování kritéria křivosti a neuniformity se bude střídát po daném (určeném uživatelem) počtu iterací algoritmu.

V případě obou kritérií algoritmus určí nové hodnoty, křivosti nebo neuniformity, ve vrcholech křivky. Nová křivka je tedy popsána jen novými hodnotami, ale poloha nových vrcholů, kterým by tyto hodnoty odpovídaly, není známá.

Algoritmus musí být tedy schopen v následujícím kroku křivku zrekonstruovat neboli určit nové vrcholy křivky. Tento postup se opakuje, dokud není splněno kritérium zastavení (viz. 2.10). V každé iteraci provedeme každý krok algoritmu pro všechny vrcholy křivky. Podrobný popis jednotlivých kroků algoritmu podle kritéria křivosti je v kapitole 2.7 a podle kritéria neuniformity v kapitole 2.8.

V případě, že se v aktuální iteraci (k) pracuje s kritériem křivosti, jsou v iteraci provedeny následující kroky:

1. Výpočet vektoru křivosti \mathbf{H}_i^k a výpočet velikosti křivosti H_i^k (viz. sekce 2.7.1).
2. Výpočet velikosti nové křivosti H_i^{k+1} (viz. sekce 2.7.2).

3. Výpočet polohy nového vrcholu \mathbf{a}_i^{k+1} (viz. sekce 2.7.3).
4. Nahrazení vrcholu \mathbf{a}_i^k křivky K novým vrcholem \mathbf{a}_i^{k+1} (viz. sekce 2.6.1).

V případě, že se v aktuální iteraci (k) pracuje s kritériem neuniformity, jsou v iteraci provedeny následující kroky:

1. Výpočet vektoru \mathbf{U}_i^k (viz. sekce 2.8.1).
2. Výpočet polohy nového vrcholu \mathbf{a}_i^{k+1} (viz. sekce 2.8.2).
3. Nahrazení vrcholu \mathbf{a}_i^k křivky K novým vrcholem \mathbf{a}_i^{k+1} (viz. sekce 2.6.1).

2.6.1 Určení nových vrcholů křivky

Jelikož jsou všechny veličiny pro k -tou iteraci algoritmu počítány z výchozích vrcholů, je nutné nahrazení vrcholu provést až ve chvíli, kdy už výchozí vrchol nebudeme potřebovat. Posledním výpočtem, ve kterém je nutné znát výchozí vrcholy křivky, je určení nových vrcholů \mathbf{a}_i^{k+1} . V $(k+1)$ -ní iteraci pracujeme jako s výchozími vrcholy s těmi, které jsme spočítali na konci k -té iterace. Tedy na konci každé iterace algoritmu provedeme pro všechny vrcholy křivky $K = (\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_n^k)$ záměnu za nově spočítané vrcholy $K = (\mathbf{a}_1^{k+1}, \mathbf{a}_2^{k+1}, \dots, \mathbf{a}_n^{k+1})$.

2.7 Algoritmus vyhlazování podle kritéria křivosti

V této kapitole popíšeme podrobně jednotlivé kroky algoritmu vyhlazování podle kritéria křivosti.

2.7.1 Výpočet vektoru křivosti \mathbf{H}_i^k a jeho velikosti H_i^k

Pro výpočet difuze křivosti je nezbytné spočítat vektor křivosti a jeho velikost. Vektor křivosti je také použit pro určení nového vrcholu křivky.

Vrcholy křivky $K = \{\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_n^k\}$ jsou aktuální vrcholy křivky, se kterými se pracuje v k -té iteraci algoritmu. Vektor křivosti $\mathbf{H}_i^k = [h_{i,x}^k, h_{i,y}^k, h_{i,z}^k]$ ve vrcholu \mathbf{a}_i^k křivky K spočteme pomocí vzorce (2.9). Vektor $\mathbf{d}_{i1}^k = \mathbf{d}_{i1}$ a velikost vektoru $d_{i1}^k = \|\mathbf{d}_{i1}^k\|$. Obdobně vektor $\mathbf{d}_{i2}^k = \mathbf{d}_{i2}$ a velikost vektoru $d_{i2}^k = \|\mathbf{d}_{i2}^k\|$.

Vzorec pro výpočet vektoru křivosti pro k -tou iteraci algoritmu je

$$\mathbf{H}_i^k = 2 \cdot \frac{\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k}{(d_{i1}^k)^2 + (d_{i2}^k)^2}. \quad (2.14)$$

$$\text{Velikost křivosti } H_i^k = \|\mathbf{H}_i^k\| = \sqrt{(h_{i,x}^k)^2 + (h_{i,y}^k)^2 + (h_{i,z}^k)^2}.$$

2.7.2 Výpočet hodnoty nové křivosti H_i^{k+1}

Ve vrcholu \mathbf{a}_i^k křivky K spočítáme hodnotu nové křivosti, která bude charakterizovat nový vrchol křivky. Tato křivost je závislá na vektoru a velikosti křivosti ve vrcholu křivky a na vektorech a velikostech křivostí bezprostředních sousedů vrcholu. Novou křivost H_i^{k+1} spočítáme jako skalární veličinu (viz. sekce 2.3.2), která může nabývat kladných i záporných hodnot. V případě, že je hodnota nové křivosti záporná, pak směr vektoru křivosti v nově spočítaném vrcholu \mathbf{a}_i^{k+1} (v případě, že jsme dosud nepohybovali se sousedními vrcholy) bude opačný než směr vektoru křivosti \mathbf{H}_i^k ve vrcholu \mathbf{a}_i^k .

Velikost nové křivosti je dána rovnicí (2.7), potom

$$H_i^{k+1} = (1 - \lambda)H_i^k + \lambda\bar{H}_i^k, \quad 0 < \lambda < 1,$$

$$\bar{H}_i^k = \frac{s_1^k \cdot H_{i-1}^k + s_2^k \cdot H_{i+1}^k}{2}, \quad (2.15)$$

kde H_{i-1}^k je velikost křivosti ve vrcholu \mathbf{a}_{i-1} a H_{i+1}^k je velikost křivosti ve vrcholu \mathbf{a}_{i+1} . Definujeme s_1^k jako znaménko křivosti, kterým se přenáší vektor křivosti \mathbf{H}_{i-1}^k do vrcholu \mathbf{a}_i^k , a s_2^k jako znaménko křivosti, kterým se přenáší vektor křivosti \mathbf{H}_{i+1}^k do vrcholu \mathbf{a}_i^k . Rekonstrukce znamének je uvedena v následující části.

Rekonstrukce znamének křivostí

Ačkoli s novou křivostí pracujeme pouze jako se skalárem, v průběhu jejího výpočtu potřebujeme znát znaménko, se kterým se bude velikost křivosti ze sousedního vrcholu přenášet do vrcholu \mathbf{a}_i^k . Znaménko určuje, v jaké vzájemné poloze je vektor křivosti ve vrcholu \mathbf{a}_i^k s vektory křivosti bezprostředních sousedů vrcholu \mathbf{a}_i^k .

Řešením problému rekonstrukce znaménka by bylo také křivku explicitně zorientovat. Pak bychom byli schopni vzájemnou polohu vektorů křivostí určit právě vůči orientaci křivky. Jelikož nezůstaneme jen u filtrací křivek, ale budeme se snažit tento postup dále aplikovat na filtrace povrchů, mezi nimiž se vyskytují i povrchy nezorientovatelné, rozhodli jsme se problém rekonstrukce znaménka již pro křivky řešit bez explicitního zorientování křivky.

Podle definice (2.15) je s_1^k znaménko křivosti, kterým se přenáší vektor křivosti \mathbf{H}_{i-1}^k do vrcholu \mathbf{a}_i^k , a s_2^k znaménko křivosti, kterým se přenáší vektor křivosti \mathbf{H}_{i+1}^k do vrcholu \mathbf{a}_i^k .

Následující výpočet je určen pro vyhodnocení znaménka s_1^k . Pro výpočet znaménka s_2^k je nutné nahradit s_1^k za s_2^k , vektor křivosti \mathbf{H}_{i-1}^k vektorem křivosti \mathbf{H}_{i+1}^k a vektor \mathbf{d}_{i1}^k vektorem \mathbf{d}_{i2}^k .

Uvažujeme dva případy:

1. Vektory \mathbf{d}_{i1}^k , \mathbf{H}_i^k , \mathbf{H}_{i-1}^k jsou v obecné poloze (viz. obrázek 2.4(a) a obrázek 2.4(b)).
2. Alespoň jeden z vektorů \mathbf{H}_i^k , \mathbf{H}_{i-1}^k leží v přímce určené vektorem \mathbf{d}_{i1}^k (viz obrázky 2.4(c), 2.4(d), 2.4(e) a 2.4(f)).

V případě bodu 1 definujeme přenos znaménka tak, že promítneme vektory \mathbf{H}_i^k a \mathbf{H}_{i-1}^k do roviny kolmé k vektoru \mathbf{d}_{i1}^k a zjistíme úhel mezi nimi. Pokud je menší než $\frac{\pi}{2}$, je znaménko kladné, pokud je větší než $\frac{\pi}{2}$, je znaménko záporné, jinak je rovno nule. Takže

$$m_1 = \text{sign}\left(\mathbf{H}_{i-1}^k (\mathbf{E} - \tilde{\mathbf{d}}_{i1}^k (\tilde{\mathbf{d}}_{i1}^k)^\top) \mathbf{H}_i^k\right), \quad (2.16)$$

kde \mathbf{E} je jednotková matice 3×3 a $\tilde{\mathbf{d}}_{i1}^k = \frac{\mathbf{d}_{i1}^k}{\|\mathbf{d}_{i1}^k\|}$ je jednotkový vektor.

Tento postup selhává v případě 2., kdy $m_1 = 0$. Zde ale můžeme rozhodnout podle znaménka

$$m_2 = \text{sign}\left((\mathbf{H}_{i-1}^k)^\top \mathbf{H}_i^k\right). \quad (2.17)$$

Celkové rozhodnutí uděláme takto: jestliže $m_1 = 0$ tak $s_1^k = m_2$ jinak $s_1^k = m_1$. To lze implementovat následujícím postupem:

1. Vypočítáme čísla

$$a = (\mathbf{H}_{i-1}^k)^\top \mathbf{H}_i^k, \quad b = (\mathbf{H}_{i-1}^k)^\top \tilde{\mathbf{d}}_{i1}^k (\tilde{\mathbf{d}}_{i1}^k)^\top \mathbf{H}_i^k.$$

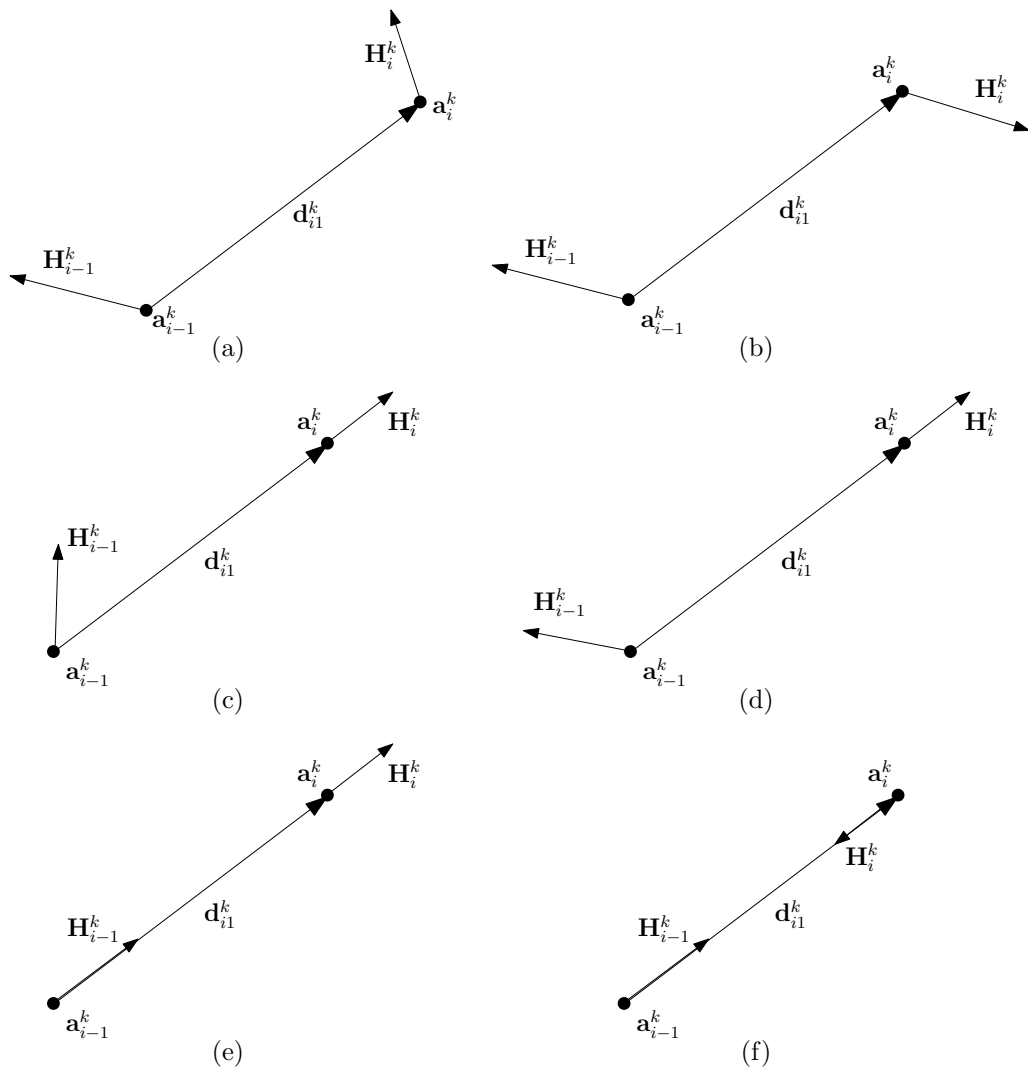
2. Znaménko s_1^k rozhodneme podle tabulky:

test	s_1^k
$a > b \vee (a = b \wedge a > 0)$	+1
$a < b \vee (a = b \wedge a < 0)$	-1
$a = b = 0$	0

Znaménko s_1^k bude rovno nule pokud:

- Vektor $\mathbf{H}_i^k = 0$.
- Vektor $\mathbf{H}_{i-1}^k = 0$.
- Vektory \mathbf{H}_{i-1}^k a \mathbf{H}_i^k jsou navzájem kolmé a alespoň jeden z nich je kolmý na spojnici \mathbf{d}_{i1}^k . V takovém případě nelze rozhodnout. Proto je správné, že znaménko přenosu je rovno nule.

Vektor \mathbf{H}_i^k je nulový, pokud \mathbf{a}_i^k a \mathbf{a}_{i-1}^k leží v jedné přímce a $\mathbf{a}_i^k - \mathbf{a}_{i-1}^k = \mathbf{a}_{i+1}^k - \mathbf{a}_i^k$. To je v ekvilibriu možné pouze, pokud všechny body křivky leží na přímce a nebo v inflexním bodu. Poloha takového bodu se při další aktualizaci zachovává ($\mathbf{H}_i^{k+1} = 0$). Pokud je \mathbf{H}_{i-1}^k nulový, nevadí, že příslušné znaménko přenosu je také nulové.



Obrázek 2.4: Případy propagace znaménka: (a) kladné, (b) záporné, (c) kladné, (d) záporné, (e) kladné, (f) záporné.

2.7.3 Výpočet nového vrcholu \mathbf{a}_i^{k+1}

Ve vrcholu křivky je spočtena velikost nové křivosti. Pro další vyhlazování nebo zpracování křivky je nutné znát polohu nového vrcholu křivky.

Nový vrchol \mathbf{a}_i^{k+1} křivky K , který bude výchozím vrcholem pro $(k+1)$ -ní iteraci algoritmu, spočítáme podle

$$\mathbf{a}_i^{k+1} = \mathbf{a}_i^k + \alpha_i^k (\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k), \quad (2.18)$$

kde \mathbf{a}_i^k je výchozí vrchol křivky pro k -tou iteraci. Výpočet hodnoty α_i^k je uveden dále.

Nový vrchol křivky bude tedy ležet na přímce, jejíž směr je určen vektorem křivosti \mathbf{H}_i^k v bodě \mathbf{a}_i^k .

Pro výpočet nového vrcholu \mathbf{a}_i^{k+1} křivky je nutné znát skalární hodnotu α_i^k . Tato hodnota určuje parametr posunu vrcholu \mathbf{a}_i^k po přímce určené vektorem křivosti \mathbf{H}_i^k tak, abychom dostali nový vrchol \mathbf{a}_i^{k+1} .

Odvození α_i^k

Ze vzorce (2.9) pro \mathbf{H}_i^{k+1} platí

$$\mathbf{H}_i^{k+1} = 2 \cdot \frac{\mathbf{d}_{i1}^{k+1} + \mathbf{d}_{i2}^{k+1}}{(d_{i1}^{k+1})^2 + (d_{i2}^{k+1})^2}, \quad (2.19)$$

kde \mathbf{H}_i^{k+1} je vektor křivosti pro novou pozici vrcholu \mathbf{a}_i^k křivky K . Polohu ostatních vrcholů křivky uvažujeme tu, která je výchozí pro výpočet v iteraci k . Tedy $\forall j \neq i : \mathbf{a}_j^k = \mathbf{a}_j^{k+1}$.

Pak z definice v kapitole 2.3 plyne:

$$\mathbf{d}_{i1}^{k+1} = \mathbf{a}_i^{k+1} - \mathbf{a}_{i-1}^k = \mathbf{a}_i^k + \alpha_i^k (\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k) - \mathbf{a}_{i-1}^k = (1 + \alpha_i^k) \mathbf{d}_{i1}^k + \alpha_i^k \mathbf{d}_{i2}^k, \quad (2.20)$$

$$\mathbf{d}_{i2}^{k+1} = \mathbf{a}_i^{k+1} - \mathbf{a}_{i+1}^k = \mathbf{a}_i^k + \alpha_i^k (\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k) - \mathbf{a}_{i+1}^k = (1 + \alpha_i^k) \mathbf{d}_{i2}^k + \alpha_i^k \mathbf{d}_{i1}^k, \quad (2.21)$$

$$\begin{aligned} (d_{i1}^{k+1})^2 + (d_{i2}^{k+1})^2 &= (\mathbf{d}_{i1}^{k+1})^\top \mathbf{d}_{i1}^{k+1} + (\mathbf{d}_{i2}^{k+1})^\top \mathbf{d}_{i2}^{k+1} = \\ &= (1 + \alpha_i^k)^2 \cdot ((d_{i1}^k)^2 + (d_{i2}^k)^2) + \\ &\quad + (4\alpha_i^k + 4(\alpha_i^k)^2) \cdot ((\mathbf{d}_{i1}^k)^\top (\mathbf{d}_{i2}^k)) + \\ &\quad + (\alpha_i^k)^2 \cdot ((d_{i1}^k)^2 + (d_{i2}^k)^2). \end{aligned} \quad (2.22)$$

Po dosazení do vzorce (2.19) dostaneme

$$\mathbf{H}_i^{k+1} = \frac{(1 + 2\alpha_i^k) \cdot \mathbf{H}_i^k \cdot ((d_{i1}^k)^2 + (d_{i2}^k)^2)}{(1 + 2\alpha_i^k + 2(\alpha_i^k)^2) \cdot ((d_{i1}^k)^2 + (d_{i2}^k)^2) + 4\alpha_i^k \cdot (1 + \alpha_i^k) \cdot ((\mathbf{d}_{i1}^k)^\top \mathbf{d}_{i2}^k)}. \quad (2.23)$$

Pokud čitatele i jmenovatele vzorce (2.23) vydělíme výrazem $(d_{i1}^k)^2 + (d_{i2}^k)^2$ dostaneme výraz

$$\mathbf{H}_i^{k+1} = \mathbf{H}_i^k \cdot \frac{1 + 2\alpha_i^k}{1 + 2\alpha_i^k + 2(\alpha_i^k)^2 + 4\alpha_i^k(1 + \alpha_i^k) \cdot P}, \quad (2.24)$$

$$\text{kde } P = \frac{(\mathbf{d}_{i1}^k)^\top \mathbf{d}_{i2}^k}{(d_{i1}^k)^2 + (d_{i2}^k)^2}. \quad (2.25)$$

Pro další odvození z rovnice je nutné, aby se v rovnici nevyskytovaly vektory křivosti ale hodnoty křivosti. Proto celou rovnici vynásobíme výrazem $\frac{\mathbf{H}_i^k}{H_i^k}$. Dostaneme rovnici:

$$\mathbf{H}_i^{k+1} \cdot \frac{\mathbf{H}_i^k}{H_i^k} = H_i^k \cdot \frac{1 + 2\alpha_i^k}{1 + 2\alpha_i^k + 2(\alpha_i^k)^2 + 4\alpha_i^k(1 + \alpha_i^k) \cdot P}. \quad (2.26)$$

Výraz $\mathbf{H}_i^{k+1} \cdot \frac{\mathbf{H}_i^k}{H_i^k}$ můžeme nahradit H_i^{k+1} , což je hodnota nové křivosti zadaná v kapitole (2.7.2). Dále jej upravíme tak, abychom z něj mohli vyjádřit řešení kvadratické rovnice (α_i^k) :

$$(\alpha_i^k)^2 \cdot (2H_i^{k+1} + 4H_i^{k+1} \cdot P) + \alpha_i^k \cdot (2H_i^{k+1} + 4H_i^{k+1} \cdot P - 2H_i^k) + H_i^{k+1} - H_i^k = 0. \quad (2.27)$$

Pro lepší přestavu v následujícím textu můžeme čitatele i jmenovatele vzorce (2.24) vynásobit výrazem $Q = \frac{d_{i1}^k}{d_{i2}^k} + \frac{d_{i2}^k}{d_{i1}^k}$. Dostaneme výraz

$$\frac{H_i^{k+1}}{H_i^k} = \frac{(1 + 2\alpha_i^k) \cdot Q}{(1 + 2\alpha_i^k + 2(\alpha_i^k)^2) \cdot Q + 4\alpha_i^k(1 + \alpha_i^k) \cdot \cos \varphi}, \quad (2.28)$$

kde $\cos \varphi = \frac{(\mathbf{d}_{i1}^k)^\top \mathbf{d}_{i2}^k}{d_{i1}^k \cdot d_{i2}^k}$ a φ je úhel, který svírá vektor \mathbf{d}_{i1}^k s vektorem \mathbf{d}_{i2}^k .

Abychom mohli dále pracovat s α_i^k , budeme se nejdříve soustředit na to, jak vypadá funkce velikosti křivosti H_i^k v případě, kdy je vrchol \mathbf{a}_i^k vytahován v jednom směru (posun jen vrcholu \mathbf{a}_i^k , sousedi jsou na místě) po přímce, která je kolmá ke spojnici sousedů vrcholu a prochází vrcholem \mathbf{a}_i^k . Funkce velikosti křivosti H_i^k v takto vytahovaném vrcholu \mathbf{a}_i^k není monotonní. Chování funkce si popíšeme na příkladu: Vytahujeme-li vrchol po přímce směrem ven a tím i zmenšujeme úhel φ , který svírají vektory \mathbf{d}_{i1}^k a \mathbf{d}_{i2}^k , pak se velikost křivosti zvětšuje. Toto chování platí na intervalu $\varphi \in [\pi, \text{mez}\varphi]$, kde $\text{mez}\varphi \in [\pi, 0]$ a funkce velikosti křivosti je tedy na tomto intervalu rostoucí. Úhel $\text{mez}\varphi$ je roven úhlu, pro který nastává maximum funkce H_i^k a je roven $\frac{\pi}{2}$. Odvození je provedeno pro funkci, která má stejný průběh jako funkce H_i^k . Vysvětlení a odvození je dále v textu. Pokud vytahujeme vrchol dále, tedy $\varphi \in [\text{mez}\varphi, 0]$, velikost křivosti se zmenšuje a tedy funkce H_i^k bude na tomto intervalu klesající. Interval, na kterém je křivka klesající, není pro

vyhlazování vhodný, protože i přes zmenšující se velikost křivosti se vrchol více „špičatí“. Chování funkce H_i^k tedy neodpovídá intuitivní představě o chování křivosti, podle které by se měla křivost se zmenšujícím se úhlem mezi \mathbf{d}_{i1}^k a \mathbf{d}_{i2}^k zvětšovat.

Výběr řešení α_i^k

Rovnice (2.27) je kvadratická. Mohou tedy nastat tři případy řešení:

1. Rovnice má jen jedno řešení:
Použijeme α_i^k , které vyšlo.

2. Rovnice má dvě různá řešení:
Musíme rozhodnout, které řešení je pro algoritmus výhodnější.

Při výběru využijeme pozorování o chování funkce H_i^k . Funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ je složená z funkce $H_i^{k+1}(\alpha_i^k)$ a konstanty H_i^k , která je ve vrcholu \mathbf{a}_i^k daná. Chování funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ je na intervalu, kde $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k) \geq 0$, stejné jako chování funkce H_i^k až na vynásobení nenulovou konstantou, což na průběh funkce nemá vliv. Na intervalu, kde $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k) \leq 0$, je chování funkce stejné jako chování funkce $-H_i^k$, také až na vynásobení nenulovou konstantou. Příklad $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ je na obrázku 2.5. Řešení α_i^k budeme volit takové, které náleží intervalu, pro který je funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ rostoucí. Z tvaru funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ plyne, že maximálně jedno řešení náleží tomuto intervalu.

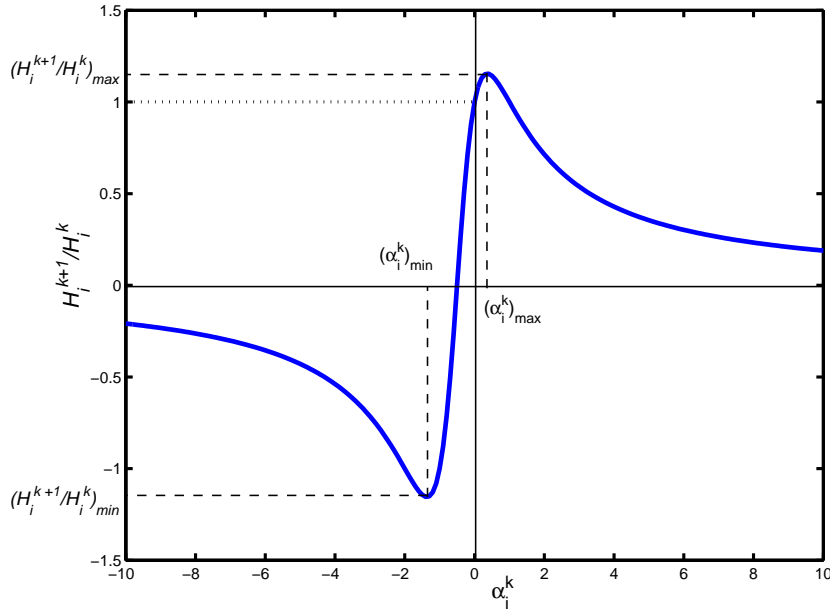
V případě dvou různých řešení α_i^k vybíráme tedy takto: Řešením rovnice (2.27) je to ze dvou řešení $(\alpha_i^k)_{1,2}$, které náleží do intervalu $((\alpha_i^k)_{\min}, (\alpha_i^k)_{\max})$, kde z výrazu (2.28) vyjádříme:

$$(\alpha_i^k)_{\min} = -\frac{1}{2} - \frac{1}{2} \sqrt{\frac{Q - 2 \cos \varphi}{Q + 2 \cos \varphi}}, \quad (2.29)$$

$$(\alpha_i^k)_{\max} = -\frac{1}{2} + \frac{1}{2} \sqrt{\frac{Q - 2 \cos \varphi}{Q + 2 \cos \varphi}}. \quad (2.30)$$

Věta: Velikost úhlu $mez\varphi$, který svírají vektory \mathbf{d}_{i1}^{k+1} a \mathbf{d}_{i2}^{k+1} (případně \mathbf{d}_{i1}^k a \mathbf{d}_{i2}^k), je pro extrémů funkce $\frac{H_i^{k+1}}{H_i^k}$ (tedy maximum i minimum) rovna $\frac{\pi}{2}$.

Důkaz: Důkaz provedeme pro maximum, pro minimum je zcela analogický. Ve výrazu (2.30) nahradíme Q a $\cos \varphi$ rozpisem, pomocí



Obrázek 2.5: Graf funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$, kde $Q = 2$ a $\varphi = \frac{4\pi}{6}$. Jiné hodnoty Q a φ nemění charakter průběhu funkce, mění se pouze pozice (vůči α_i^k) a velikost extrémů funkce.

\mathbf{d}_{i1}^k a \mathbf{d}_{i2}^k . Jak bylo již použito v 2.28, $Q = \frac{d_{i1}^k}{d_{i2}^k} + \frac{d_{i2}^k}{d_{i1}^k}$ a $\cos \varphi = \frac{(\mathbf{d}_{i1}^k)^\top \mathbf{d}_{i2}^k}{d_{i1}^k \cdot d_{i2}^k}$. Po upravení dostaneme výraz

$$(\alpha_i^k)_{\max} = -\frac{1}{2} + \frac{1}{2} \sqrt{\frac{(\mathbf{d}_{i1}^k)^2 + (\mathbf{d}_{i2}^k)^2 - 2(\mathbf{d}_{i1}^k)^\top (\mathbf{d}_{i2}^k)}{(\mathbf{d}_{i1}^k)^2 + (\mathbf{d}_{i2}^k)^2 + 2(\mathbf{d}_{i1}^k)^\top (\mathbf{d}_{i2}^k)}}. \quad (2.31)$$

Pomocí výrazů 2.20 a 2.21 vyjádříme vektory \mathbf{d}_{i1}^{k+1} a \mathbf{d}_{i2}^{k+1} , v případě, že by k výpočtu nového vrcholu \mathbf{a}_i^{k+1} byla použita hodnota $(\alpha_i^k)_{\max}$:

$$(\mathbf{d}_{i1}^{k+1})_{\max} = (1 + (\alpha_i^k)_{\max}) \mathbf{d}_{i1}^k + (\alpha_i^k)_{\max} \mathbf{d}_{i2}^k. \quad (2.32)$$

$$(\mathbf{d}_{i2}^{k+1})_{\max} = (1 + (\alpha_i^k)_{\max}) \mathbf{d}_{i2}^k + (\alpha_i^k)_{\max} \mathbf{d}_{i1}^k. \quad (2.33)$$

Vyjádříme-li a upravíme skalární součin vektorů $(\mathbf{d}_{i1}^{k+1})_{\max}$ a $(\mathbf{d}_{i2}^{k+1})_{\max}$ dostaneme

$$(\mathbf{d}_{i1}^{k+1})_{\max} \cdot (\mathbf{d}_{i2}^{k+1})_{\max} = 0, \quad (2.34)$$

z čehož plyne, že úhel $mez\varphi$, který svírají vektory, je roven $\frac{\pi}{2}$.

□

Poznámka: Pro odstranění problému dělení nulou při implementaci výpočtu extrému funkce přidáme do jmenovatele pod odmocninou ε , $\varepsilon > 0$ a $\varepsilon \rightarrow 0$.

Výbírat řešení jen způsobem $\alpha_i^k = \min(|(\alpha_i^k)_1|, |(\alpha_i^k)_2|)$ není možné. V některých případech se vrchol křivky pohybuje přesně opačným směrem, než je nutné k dosažení dobrých výsledků vyhlazování.

3. Rovnice nemá řešení:

Tento případ je ekvivalentní situaci, kdy hodnota $\frac{H_i^{k+1}}{H_i^k}$ nepatří do intervalu $((\frac{H_i^{k+1}}{H_i^k})_{\min}, (\frac{H_i^{k+1}}{H_i^k})_{\max})$, kde

$$\left(\frac{H_i^{k+1}}{H_i^k}\right)_{\min} = \frac{H_i^{k+1}}{H_i^k} \left((\alpha_i^k)_{\min}\right), \quad (2.35)$$

$$\left(\frac{H_i^{k+1}}{H_i^k}\right)_{\max} = \frac{H_i^{k+1}}{H_i^k} \left((\alpha_i^k)_{\max}\right). \quad (2.36)$$

Pro implementaci je nutné α_i^k zvolit. Hodnotu α_i^k zvolíme tak, abychom zvolili co možná nejbližší hodnotu funkce $\frac{H_i^{k+1}}{H_i^k}(\alpha_i^k)$ k hodnotě $\frac{H_i^{k+1}}{H_i^k}$. Nejbližší hodnota je vyjádřena extrémy funkce $(\frac{H_i^{k+1}}{H_i^k})_{\min}$ a $(\frac{H_i^{k+1}}{H_i^k})_{\max}$. Příslušný extrém zvolíme dle znaménka extrému a hodnoty $\frac{H_i^{k+1}}{H_i^k}$ (znaménka musejí být shodná). Podle zvoleného extrému vybereme odpovídající $(\alpha_i^k)_{\min}$ nebo $(\alpha_i^k)_{\max}$.

Tedy jestliže

$$\text{sign}\left(\frac{H_i^{k+1}}{H_i^k}\right) = \text{sign}\left(\left(\frac{H_i^{k+1}}{H_i^k}\right)_{\min}\right), \quad (2.37)$$

potom

$$\alpha_i^k = (\alpha_i^k)_{\min},$$

jinak

$$\alpha_i^k = (\alpha_i^k)_{\max}.$$

2.7.4 Speciální případy

Pokud velikost křivosti $H_i^k = 0$, pak dosazením do vzorce (2.26) dostaneme, že $H_i^{k+1} = 0$. Novou křivost ve vrcholu \mathbf{a}_i^k tedy zvolíme rovnu nule. Vzhledem k definici je volba nové křivosti H_i^{k+1} správná, protože vektor určující směr, ve kterém má být nalezen nový vrchol \mathbf{a}_i^{k+1} , je nulový.

Pokud v průběhu výpočtu k -tého kroku algoritmu byla splněna některá z následujících podmínek, hodnotu α_i^k nepočítáme a dosadíme přímo hodnotu, která je uvedena:

1. Pokud $H_i^k = 0$, pak H_i^{k+1} je také rovno nule a pozice vrcholu \mathbf{a}_i křivky K se nemění, proto z (2.18) vyjde $\alpha_i^k = 0$.
2. Pokud $H_i^k \neq 0$ a $H_i^{k+1} = 0$, pak dosazením H_i^k a H_i^{k+1} do (2.26) vyjde $\alpha_i^k = -0.5$.

2.8 Algoritmus vyhlazování podle kritéria neuniformity

V této kapitole popíšeme podrobně jednotlivé kroky algoritmu vyhlazování podle kritéria neuniformity.

2.8.1 Výpočet vektoru neuniformity \mathbf{U}_i^k

Pro výpočet difuze neuniformity a určení nového vrcholu křivky je nutné spočítat vektor neuniformity ve vrcholu křivky. Vektor neuniformity je nezbytný k výpočtu polohy nového vrcholu křivky.

Vrcholy křivky $K = \{\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_n^k\}$ jsou aktuální vrcholy křivky, se kterými se pracuje v k -té iteraci algoritmu. Vektor neuniformity $\mathbf{U}_i^k = [u_{i,x}^k, u_{i,y}^k, u_{i,z}^k]$ ve vrcholu \mathbf{a}_i^k křivky K spočteme pomocí vzorce (2.12). Vektor $\mathbf{d}_{i1}^k = \mathbf{d}_{i1}$ a $\mathbf{d}_{i2}^k = \mathbf{d}_{i2}$.

Vzorec pro výpočet vektoru neuniformity pro k -tou iteraci algoritmu je

$$\mathbf{U}_i^k = \frac{-\mathbf{P}_i^k(\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k)}{2}, \quad (2.38)$$

kde $\mathbf{P}_i^k = \mathbf{P}_i$ a je zdefinováno výrazem (2.10).

2.8.2 Výpočet nového vrcholu \mathbf{a}_i^{k+1}

Definujeme výpočet nového vrcholu \mathbf{a}_i^{k+1} křivky:

$$\mathbf{a}_i^{k+1} = \mathbf{a}_i^k + \beta \mathbf{U}_i^k, \quad (2.39)$$

kde \mathbf{a}_i^k je aktuální vrchol křivky K a hodnota β je odvozena dále.

Z definice neuniformity (2.12) odvodíme vzorec pro výpočet neuniformity, kterou by měla křivka mít v $(k+1)$ -ní iteraci algoritmu:

$$\mathbf{U}_i^{k+1} = \frac{-\mathbf{P}_i^{k+1}(\mathbf{d}_{i1}^{k+1} + \mathbf{d}_{i2}^{k+1})}{2}, \quad (2.40)$$

kde \mathbf{P}_i^{k+1} je projekce zdefinovaná výrazem (2.10) a platí, že $\mathbf{b}_i = \mathbf{b}_i^{k+1}$ a tedy $\mathbf{P}_i^{k+1} = \mathbf{P}_i^k$.

Vektor nové neuniformity ve vrcholu \mathbf{a}_i^{k+1} odpovídá vektoru \mathbf{U}_i^{k+1} zdefinovaném výrazem (2.40), pokud pohybuje jen s vrcholem \mathbf{a}_i^k a všechny ostatní vrcholy křivky K stojí. Vektor \mathbf{b}_i^k se nemění. Vektory \mathbf{U}_i^k a \mathbf{U}_i^{k+1} jsou tedy rovnoběžné.

Můžeme zapsat:

$$\begin{aligned} \mathbf{U}_i^k &= \mathbf{u}_i^k \cdot \|\mathbf{U}_i^k\| = \mathbf{u}_i^k \cdot U_i^k, \\ \mathbf{U}_i^{k+1} &= \mathbf{u}_i^k \cdot \|\mathbf{U}_i^{k+1}\| = \mathbf{u}_i^k \cdot U_i^{k+1}, \end{aligned} \quad (2.41)$$

kde \mathbf{u}_i^k je jednotkový vektor ve směru \mathbf{U}_i^k .

Vlastnosti \mathbf{U}_i^{k+1} využijeme pro odvození hodnoty β . Platí tedy:

$$\begin{aligned} \mathbf{d}_{i1}^{k+1} &= \mathbf{a}_i^{k+1} - \mathbf{a}_{i-1}^k = \mathbf{a}_i^k + \beta \mathbf{U}_i^k - \mathbf{a}_{i-1}^k = \mathbf{d}_{i1}^k + \beta \mathbf{U}_i^k, \\ \mathbf{d}_{i2}^{k+1} &= \mathbf{a}_i^{k+1} - \mathbf{a}_{i+1}^k = \mathbf{a}_i^k + \beta \mathbf{U}_i^k - \mathbf{a}_{i+1}^k = \mathbf{d}_{i2}^k + \beta \mathbf{U}_i^k. \end{aligned} \quad (2.42)$$

Potom

$$\begin{aligned} \mathbf{U}_i^{k+1} &= \frac{-\mathbf{P}_i^k(\mathbf{d}_{i1}^k + \beta \mathbf{U}_i^k + \mathbf{d}_{i2}^k + \beta \mathbf{U}_i^k)}{2} = \\ &= \frac{-\mathbf{P}_i^k(\mathbf{d}_{i1}^k + \mathbf{d}_{i2}^k)}{2} - \beta \mathbf{P}_i^k \mathbf{U}_i^k = \mathbf{U}_i^k - \beta \mathbf{U}_i^k. \end{aligned} \quad (2.43)$$

Použitím výrazu (2.41) dostáváme:

$$\mathbf{u}_i^k \cdot \mathbf{U}_i^{k+1} = \mathbf{u}_i^k \cdot \mathbf{U}_i^k - \mathbf{u}_i^k \cdot \mathbf{U}_i^k \beta. \quad (2.44)$$

Celou rovnici vynásobíme $(\mathbf{u}_i^k)^\top$ a za \mathbf{U}_i^{k+1} dosadíme z rovnice difuze (2.13).
Potom

$$U_i^k - \lambda_u U_i^k = U_i^k - \beta U_i^k, \quad (2.45)$$

z toho vyplývá, že $\beta = \lambda_u$. Vyjádříme nový vrchol \mathbf{a}_i^{k+1} :

$$\mathbf{a}_i^{k+1} = \mathbf{a}_i^k + \lambda_u \mathbf{U}_i^k, \quad (2.46)$$

kde $0 < \lambda_u < 1$.

Nový vrchol tedy leží na přímce, jejíž směr je určen vektorem neuniformity \mathbf{U}_i^k ve vrcholu \mathbf{a}_i^k .

2.9 Algoritmus pro vyhlazování neuzavřené křivky

Pokud budeme vyhlazovat neuzavřenou křivku, můžeme použít algoritmus pro uzavřené křivky pozmeněný o následující body:

1. Neuzavřená křivka obsahuje dva krajní (neplatné) vrcholy, které mají jen jednoho souseda. Pro tyto vrcholy nebudeme výpočty algoritmu provádět vůbec. Souřadnice těchto krajních vrcholů budou po celou dobu výpočtu ty, které byly zadány jako výchozí pro tyto vrcholy křivky.
2. Bezprostřední sousedi krajních (neplatných) vrcholů mají pro výpočet nové křivosti informace jen z jednoho platného souseda. Výpočet nové křivosti H_i^{k+1} budeme v těchto vrcholech provádět podle upraveného vzorce:

$$H_i^{k+1} = (1 - \lambda) H_i^k + \lambda \bar{H}_i^k, \quad 0 < \lambda < 1, \quad (2.47)$$

kde

$$\bar{H}_i^k = s_1^k \cdot H_{i-1}^k. \quad (2.48)$$

Velikost H_{i-1}^k je velikost křivosti platného souseda a s_1^k je znaménko přenosu křivosti z platného souseda do počítaného vrcholu.

3. Pro výpočet neuniformity (stejně jako pro výpočet křivosti) platí, že dva krajní vrcholy se nepočítají a jejich souřadnice se po celou dobu výpočtu nemění. S ostatními vrcholy křivky pracujeme stejně jako při práci s neuniformitou uzavřené křivky.

Všechny ostatní výpočty jsou shodné s výpočty pro algoritmus, který pracuje s uzavřenými křivkami.

2.10 Kritérium zastavení

Pro výpočet je nutné zvolit tři konstanty, které určují, kdy se má algoritmus ukončit, a dvě konstanty, které určují schéma vyhlazování:

1. Konstanta I určuje maximální počet iterací algoritmu (obou kritérií dohromady). Algoritmus je ukončen, pokud je splněna podmínka $k > I$, kde k je součet iterací vyhlazování podle kritéria křivosti a neuniformity.
2. Konstanta S_h určuje minimální přípustný průměrný posun všech vrcholů křivky K v případě práce s kritériem křivosti. Pokud je tedy vyhlazováno podle kritéria křivosti a platí, že průměrný posun všech vrcholů křivky K je menší než S_h , algoritmus je ukončen.
3. Konstanta S_u určuje minimální přípustný průměrný posun všech vrcholů křivky K v případě práce s kritériem neuniformity. Pokud je tedy vyhlazováno podle kritéria neuniformity a platí, že průměrný posun všech vrcholů křivky K je menší než S_u , algoritmus je ukončen.
4. T_h a T_u konstanty slouží k určení, po kolika iteracích se bude měnit kritérium, podle kterého se pracuje. Pokud je pracováno s kritériem křivosti a bylo takto provedeno T_h iterací, algoritmus v dalším kroku začne pracovat s kritériem neuniformity a je provedeno T_u iterací, pak se opět začne pracovat podle kritéria křivosti.

Pokud není splněna některá z ukončovacích podmínek, algoritmus pokračuje iterací $k = k + 1$. Jinak je algoritmus ukončen.

2.11 Experimenty

V této kapitole ukážeme výsledky implementace algoritmu pro vyhlazování křivek. Uvedeme experimenty, které prokazují robustnost a stabilitu algoritmu. Nakonec uvedeme experimenty, které ukazují, jak volba vstupních parametrů ovlivňuje výsledek algoritmu.

V celé kapitole budeme používat jednotné značení pro vstupní parametry algoritmu. Parametr λ určuje velikost difusního kroku pro kritérium

křivosti (viz. (2.7)). Parametr λ_u určuje velikost difusního kroku pro kritérium neuniformity (viz. (2.13)). Parametr I určuje maximální počet iterací algoritmu (kapitola 2.10), S_h a S_u určují minimální přípustný průměr posunu všech vrcholů pro kritérium křivosti a neuniformity (kapitola 2.10). Abychom předešli rozdílnému nastavení těchto parametrů pro stejnou křivku zadanou v různých měřítkách, v implementaci jsou tyto parametry vynásobeny průměrnou délkou hrany vstupní křivky. Parametry T_h a T_u určují, po kolika iteracích se mění kritérium vyhlazování (kapitola 2.10).

U popisu každého experimentu budou popsány parametry, které jsou zvoleny jinak, než je jejich předdefinovaná hodnota. Parametry λ , λ_u a I jsou uváděny vždy. Pokud parametr není v popisu experimentu uveden, jeho hodnota je:

- $S_h = 0.0001$,
- $S_u = 0.0001$,
- $T_h = 50$,
- $T_u = 50$.

2.11.1 Základní funkce

V této části jsou uvedeny experimenty, které ukazují základní funkci algoritmu. Každý experiment je proveden tak, že testovaná křivka je zašuměna a takto zašuměná křivka je vstupem pro vyhlazovací algoritmus. Provedena jsou dvě vyhlazení vstupní křivky. První vyhlazení je provedeno na deset iterací a druhé je spuštěno tak, aby vyhlazování dokonvergovalo (algoritmus byl ukončen na podmínku jedné ze sum S_h a S_u dle aktuálního kritéria). Obě vyhlazení jsou spouštěna se shodnými parametry, kromě parametru I .

Pro každý experiment jsou ukázány výsledky vyhlazování (obrázek (a)), původní nezašuměná křivka (obrázek (b)) a následující grafy:

- V obrázku (c) jsou uvedeny grafy funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{H_{\text{max}}}{H_0}$ a $\frac{H_{\text{min}}}{H_0}$ v závislosti na k -té iteraci algoritmu. Výraz H_0 označuje průměrnou velikost křivosti před začátkem vyhlazování. Výraz H_{avg} označuje průměrnou velikost křivosti v k -té iteraci algoritmu, H_{max} a H_{min} jsou maximální a minimální velikosti křivosti v k -té iteraci.
- V obrázku (d) jsou uvedeny grafy funkcí $\frac{U_{\text{avg}}}{U_0}$, $\frac{U_{\text{max}}}{U_0}$ a $\frac{U_{\text{min}}}{U_0}$ v závislosti na k -té iteraci algoritmu. Výraz U_0 označuje průměrnou velikost neuniformity před začátkem vyhlazování. Výraz U_{avg} označuje průměrnou velikost neuniformity v k -té iteraci algoritmu, U_{max} a U_{min} jsou maximální a minimální velikost neuniformity v k -té iteraci.

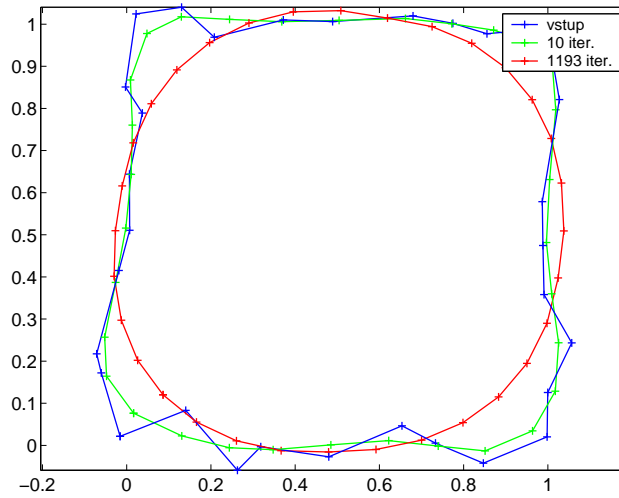
- V obrázku (e) je znázorněn graf funkce $\frac{S_{\text{avg}}}{S_0}$ v závislosti na iteraci algoritmu. Výraz S_0 označuje průměrný posun všech vrcholů křivky v první iteraci algoritmu a S_{avg} je průměrný posun v k -té iteraci algoritmu.

Velikost rozdílu mezi starou a novou křivostí je omezena parametrem λ (případně λ_u), a tím je i omezen pohyb vrcholu. Rozdíly mezi průměrným, maximálním a minimálním pohybem vrcholu nejsou v daném měřítku grafu téměř zřetelné, proto nejsou grafy funkcí $\frac{S_{\text{max}}}{S_0}$ a $\frac{S_{\text{min}}}{S_0}$ uvedeny.

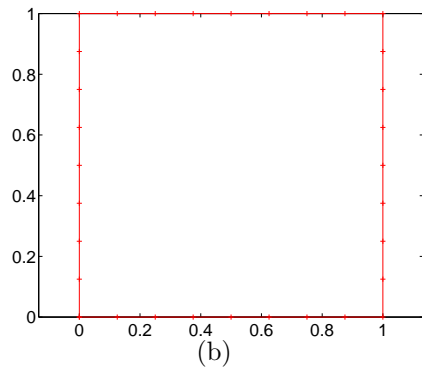
První experimenty byly provedeny na pravidelných n -úhelnících. Z definice výpočtu hodnoty nové křivosti ve vrcholu křivky (2.7) plyne, že v každém vrcholu pravidelného n -úhelníka je nová křivost rovna staré křivosti a vrcholy n -úhelníku se tedy nebudou pohybovat. Algoritmus vyhlazení by tedy na pravidelné n -úhelníky neměl mít vliv a výstupní křivka by měla být shodná s křivkou vstupní. Experimenty potvrdili, že pravidelné n -úhelníky jsou algoritmem nezměněny a algoritmus je ukončen již po první iteraci výpočtu (vyhlazováno jak podle kritéria křivosti, tak i podle kritéria neuniformity).

V dalším experimentu jsme testovali vyhlazování čtverce. Vstupní křivkou je zašuměný čtverec s dělenými hranami. Parametry jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 1193 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.6. Na obrázku 2.6(a) je vidět, že již po 10 iteracích (zeleně) jsou z křivky odstraněny velké nerovnosti a po dokonvergování po 1193 iteracích (červeně) je křivka téměř (díky velkému vstupnímu šumu) kružnicí. Na obrázku 2.6(c) lze pozorovat, že funkce $\frac{H_{\text{avg}}}{H_0}$, v průběhu vyhlazování, velmi rychle získá konstantní charakter a funkce $\frac{H_{\text{min}}}{H_0}$ a $\frac{H_{\text{max}}}{H_0}$ se k ní se vzrůstajícím počtem iterací přibližují. V průběhu vyhlazování je také velmi rychle dosaženo téměř (díky velkému vstupnímu šumu) uniformní křivky. To lze pozorovat (viz. obrázek 2.6(e)) na charakteru funkce $\frac{S_{\text{avg}}}{S_0}$: Iterace, ve kterých se pracuje s kritériem křivosti je v grafu funkce pozorovatelná změna průměrného posunu vrcholů. Pro iterace, ve kterých se pracuje s kritériem neuniformity je průměrný posun vrcholů roven téměř nule.

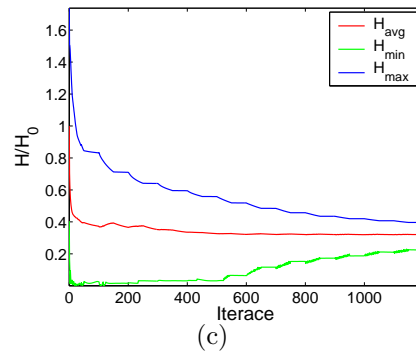
V následujícím experimentu je vstupní křivkou zašuměný šestiúhelník s dělenými hranami. Parametry jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 241 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.7. Na obrázku 2.7(a) je vidět, že již po 10 iteracích algoritmu (zeleně) je křivka téměř vyhlazena a v dalších iteracích je zmenšována převážně neuniformita křivky. To je možné vidět také na grafech: Na obrázku 2.7(c) lze pozorovat, že funkce $\frac{H_{\text{avg}}}{H_0}$ je již po velmi malém počtu kroků téměř konstantní, naopak na obrázku 2.7(d) lze pozorovat, že funkce $\frac{U_{\text{avg}}}{U_0}$, $\frac{U_{\text{min}}}{U_0}$ a $\frac{U_{\text{max}}}{U_0}$ klesají k nule. Funkce $\frac{U_{\text{avg}}}{U_0}$ nedokonvergovala, protože algoritmus byl



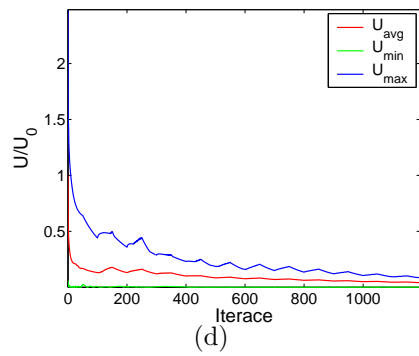
(a)



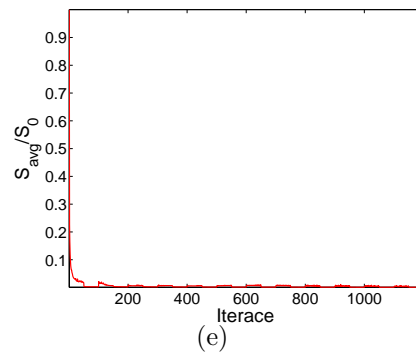
(b)



(c)



(d)



(e)

Obrázek 2.6: Vyhlazování zašuměného čtverce s dělenými hranami. Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 1193 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).

ukončen po 241 iteracích a ze schématu vyhlazování (50 iterací kritérium křivosti, 50 iterací kritérium neuniformity) plyne, že byl ukončen v průběhu pracování s kritériem křivosti. Průměrný posun všech vrcholů křivky tedy již splňoval podmínku ukončení na minimální sumu S_h .

V následujícím experimentu jsme sledovali vyhlazování složitější křivky, kde vstupem je zašuměná hvězdice. Parametry jsou $\lambda = 0.01$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 2185 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.8. Na obrázku 2.8(c) lze pozorovat, že funkce $\frac{H_{avg}}{H_0}$ a $\frac{H_{max}}{H_0}$ na začátku vyhlazování křivky oscilují: To je způsobeno vlastností funkce velikosti křivosti. Křivka obsahuje vrcholy, které jsou velmi „špičaté“ a vyhlazováním jsou postupně stahovány. V průběhu stahování vrcholu nabývá hodnota funkce velikosti křivosti svého maxima, což se projevuje ve funkcích $\frac{H_{avg}}{H_0}$ a $\frac{H_{max}}{H_0}$ jako postupný nárůst a posléze postupný pokles hodnoty funkce. Na obrázku 2.8(d) lze na funkcích $\frac{U_{avg}}{U_0}$ a $\frac{U_{max}}{U_0}$ také pozorovat oscilaci: Na intervalech, kde je vyhlazováno podle kritéria křivosti, velikost funkcí roste. Naopak na intervalech, kde je vyhlazováno podle kritéria neuniformity, velikost neuniformity klesá. Oscilace je způsobena neortogonalitou kritérií. Z grafu funkcí je však patrné, že se zvyšujícím se počtem iterací je oscilace méně výrazná a funkce konvergují k nule.

V posledním prezentovaném experimentu je vstupní křivkou zašuměná lemniskáta. Parametry jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 297 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.9. Křivka obsahuje inflexní body, které by měly být po vyhlazení zachovány. Na obrázku 2.9(a) je vidět, že inflexní body křivky jsou po vyhlazení skutečně zachovány.

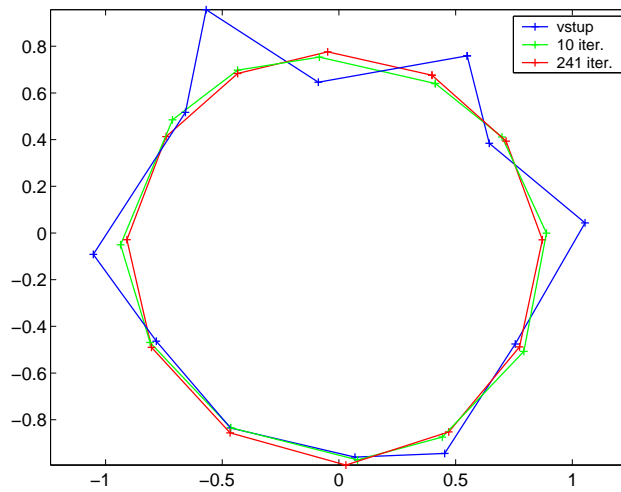
Součástí experimentů byl také test, ve kterém bylo zjišťováno, zda výsledky vyhlazování závisí na dokončování v libovolném ze dvou kritérií, nebo jen v kritériu křivosti. Experimentálně bylo prokázáno, že pokud je vynuceno dokončování jen v případě pracování s kritériem křivosti, výsledná křivka je téměř totožná s křivkou, která je výsledkem algoritmu, jehož dokončování není omezeno na kritérium křivosti, ale počet iterací je téměř dvojnásobný.

2.11.2 Robustnost a stabilita

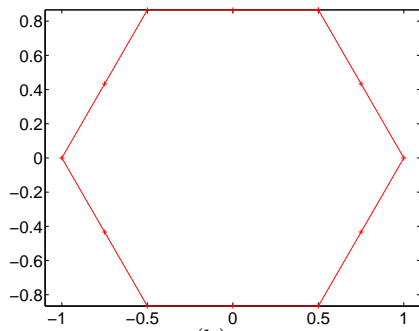
V této kapitole jsou uvedeny experimenty, které prokazují robustnost a stabilitu algoritmu. Robustnost je ukázána na vyhlazení extrémně zašuměných křivek. Stabilita je ukázána tak, že po dokončování vyhlazované křivky je k výsledku přidán šum a takto pozměněná křivka je algoritmem znovu vyhlazena.

K experimentům s robustností jsou uvedeny výsledky vyhlazování (obrázek (a)), původní nezašuměná křivka (obrázek (b)) a grafy, jejichž popis je shodný s popisem grafů v kapitole 2.11.1.

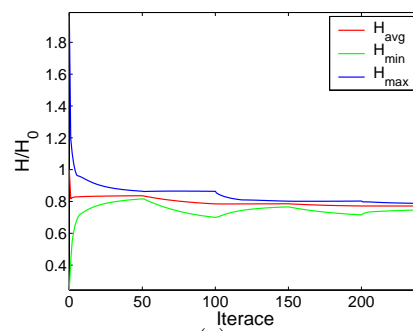
Pro ověření robustnosti algoritmu jsme cíleně zvolili složitější objekty.



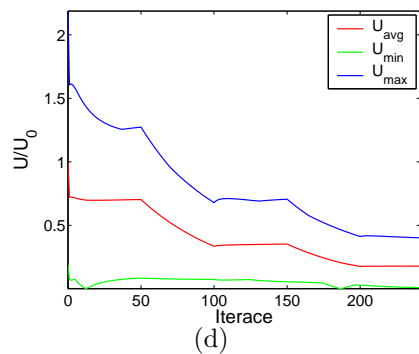
(a)



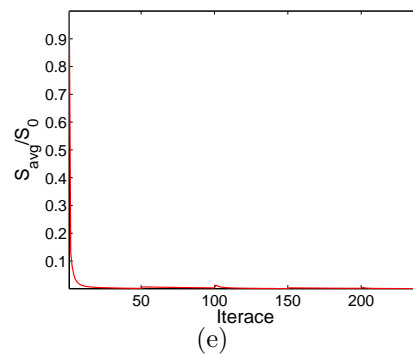
(b)



(c)

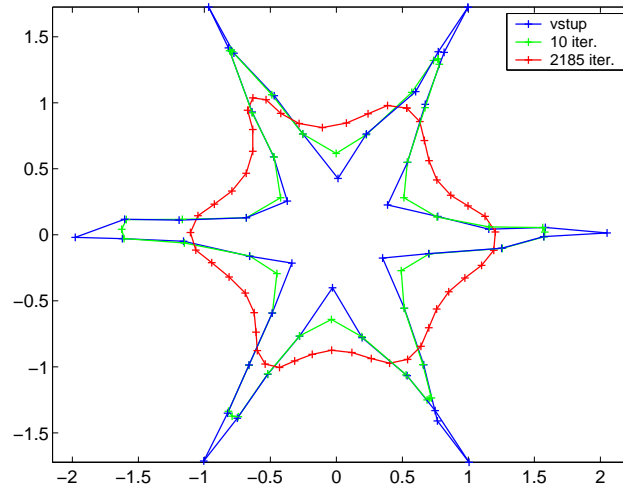


(d)

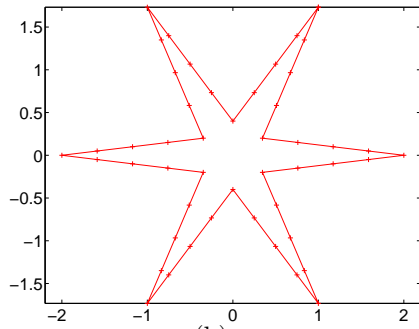


(e)

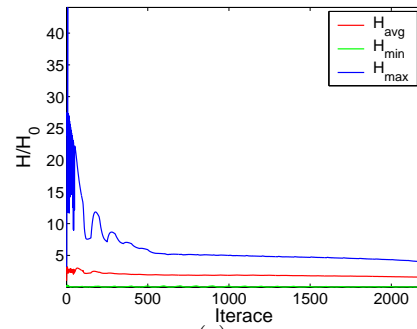
Obrázek 2.7: Vyhlazování zašuměného šestiúhelníku s dělenými hranami. Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 241 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



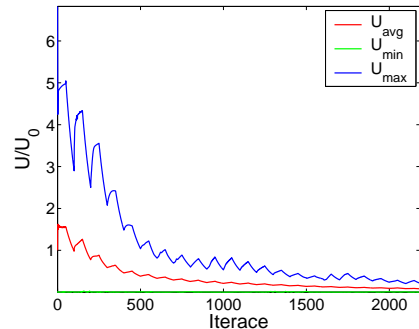
(a)



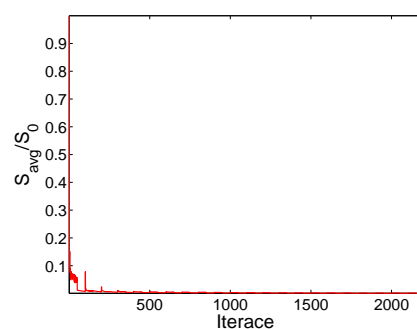
(b)



(c)

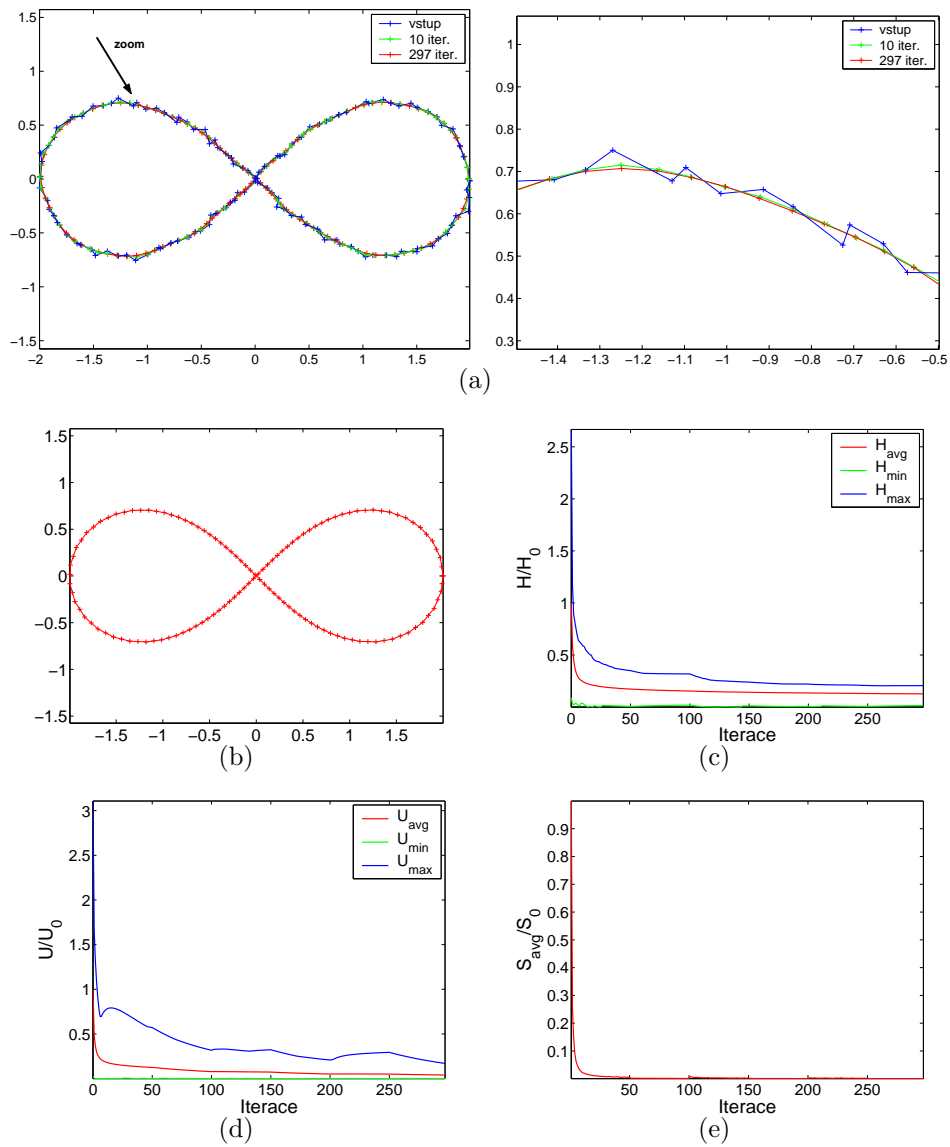


(d)



(e)

Obrázek 2.8: Vyhlazování zašuměné hvězdičky. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 2185 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



Obrázek 2.9: Vyhlazování zašuměné lemniskáty. Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 297 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).

V prvním experimentu je vstupní křivkou extrémně zašuměná hvězdice. Šum je tak velký, že vstupní křivka není Jordanovskou křivkou (vzájemně se protíná). Parametry jsou $\lambda = 0.01$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 6231 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.10. Na obrázcích 2.10(c) a 2.10(d) lze pozorovat oscilaci funkcí stejně, jako je tomu u experimentu s hvězdici v kapitole 2.11.1. I přes extrémní zašumění je vyhlazená křivka téměř souměrná!

V dalším experimentu je vstupní křivkou extrémně zašuměná lemniskáta. Parametry jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 395 iteracích. Průběh vyhlazování a příslušné grafy jsou na obrázku 2.11. Na obrázku 2.11(a) je vidět, že již po 50 iteracích je křivka téměř vyhlazena. Výsledný tvar křivky, po dokonvergování algoritmu, odpovídá tvaru křivky, která je vstupem algoritmu: Algoritmus zachovává i inflexní body přidané šumem. Na obrázku 2.11(d) lze pozorovat, že funkce $\frac{U_{\text{avg}}}{U_0}$ dokonvergovala téměř k nule.

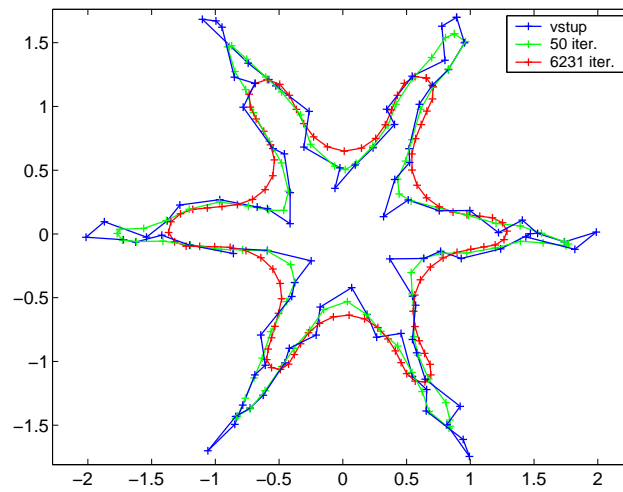
K experimentům stability jsou uvedeny výsledky vyhlazování (obrázek (a)), výsledek vyhlazování s přidaným šumem a jeho vyhlazení (obrázek (b)) a grafy, jejichž popis je shodný s popisem grafů v kapitole 2.11.1. Grafy zachycují celý průběh vyhlazování (včetně vyhlazení po přidání šumu do již vyhlazené křivky).

Nejdříve jsme testovali zašuměný šestiúhelník s dělenými hranami. Algoritmus dokonvergoval po 419 iteracích. K vyhlazené křivce byl přidán šum a křivka byla znovu vyhlazena. Algoritmus dokonvergoval po 332 iteracích. Parametry jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Průběhy vyhlazování a příslušné grafy jsou na obrázku 2.12. Na grafech funkcí lze pozorovat, že po přidání šumu do již vyhlazené křivky se velmi změnila hodnota všech funkcí. V průběhu dalších iterací algoritmu se však všechny hodnoty vracejí k hodnotám funkcí před zašuměním křivky. To znamená, že křivka s přidaným šumem je opět vyhlazena do stejného tvaru, jako před přidáním šumu.

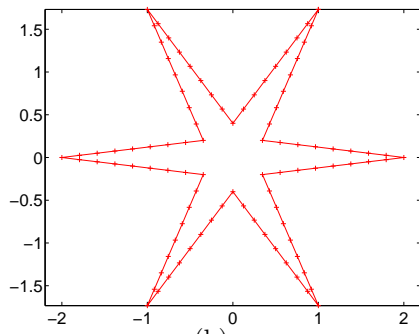
V tomto experimentu je vstupní křivkou zašuměná hvězdice. Algoritmus dokonvergoval po 1987 iteracích. K vyhlazené křivce byl přidán šum a křivka byla znovu vyhlazena. Algoritmus dokonvergoval po 1281 iteracích. Parametry jsou $\lambda = 0.01$ a $\lambda_u = 0.01$. Průběhy vyhlazování a příslušné grafy jsou na obrázku 2.13. Na obrázcích 2.13(c) a 2.13(d) lze pozorovat, že hodnoty funkcí $\frac{H_{\text{avg}}}{H_0}$ a $\frac{U_{\text{avg}}}{U_0}$ jsou po dokonvergování křivky se znovu přidaným šumem menší než hodnoty funkcí před přidáním šumu. Myslíme si, že některé vrcholy mohly být ve „falešných“ inflexních bodech a po přidání šumu a vyhlazení křivky byly tyto inflexní body odstraněny, což způsobilo zmenšení hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$ a $\frac{U_{\text{avg}}}{U_0}$.

2.11.3 Výběr lambdy pro křivost a neuniformitu

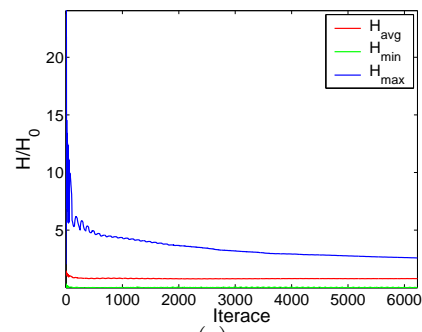
V této kapitole ukážeme, jak volba λ a λ_u ovlivňuje výsledek algoritmu.



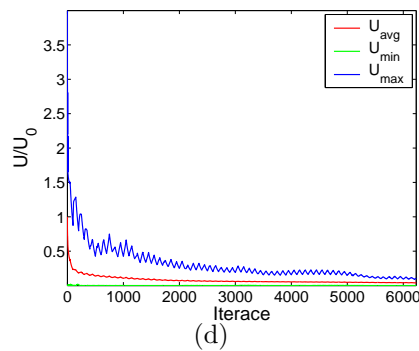
(a)



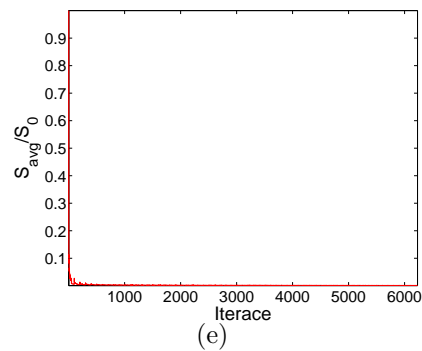
(b)



(c)

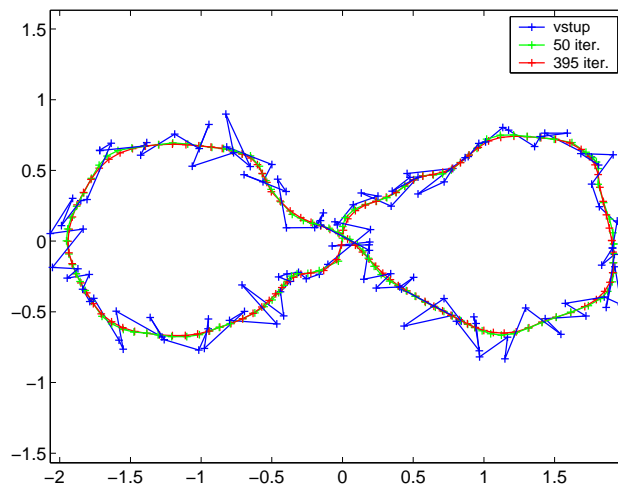


(d)

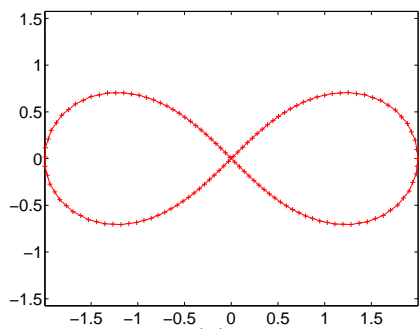


(e)

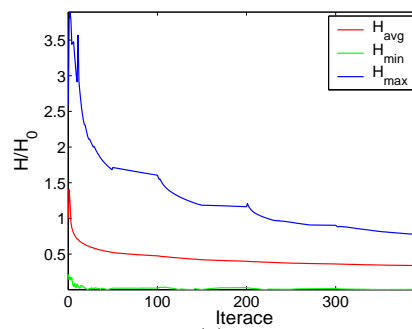
Obrázek 2.10: Vyhlazování extrémně zašuměné hvězdice. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 6231 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



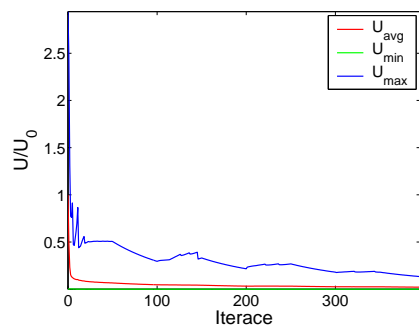
(a)



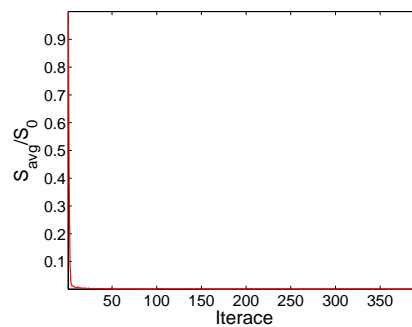
(b)



(c)

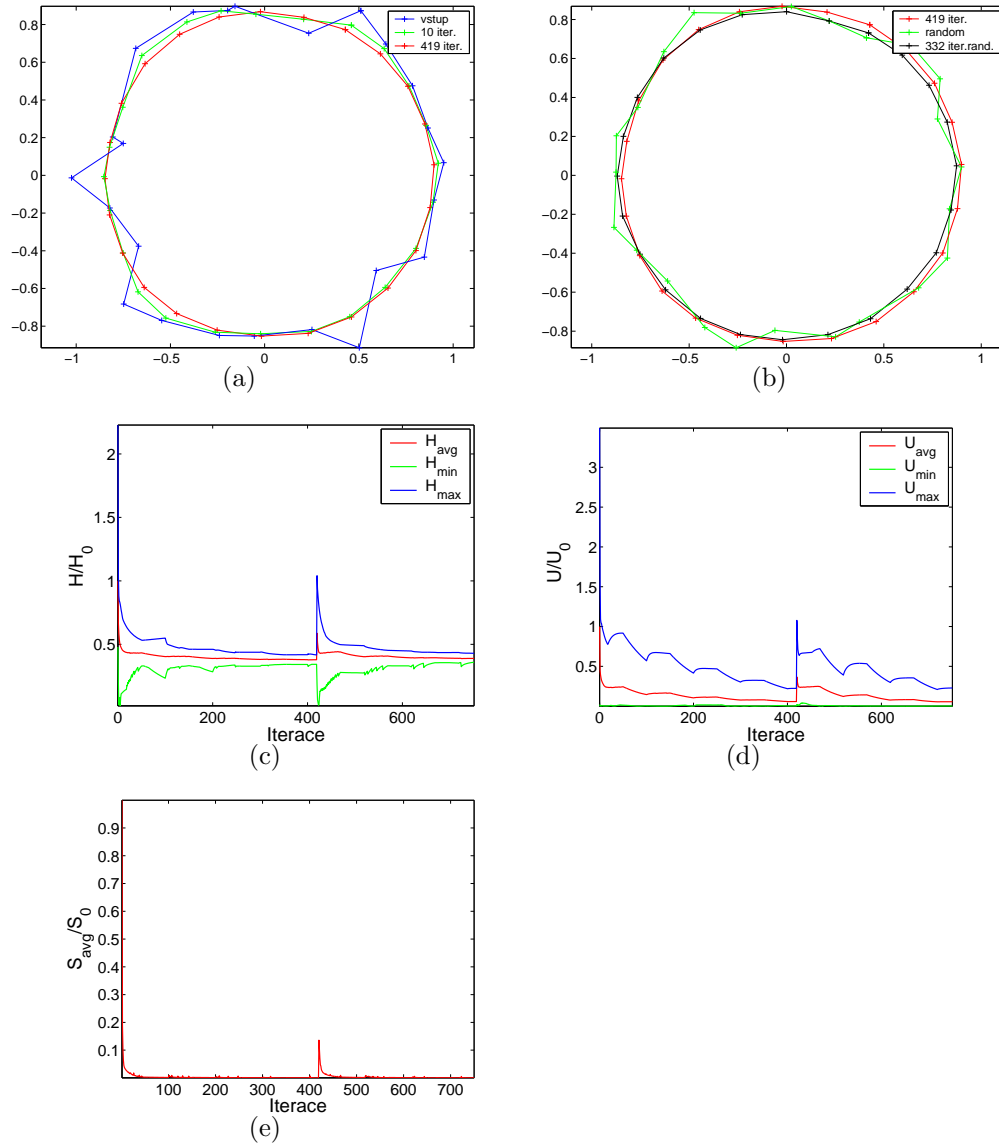


(d)

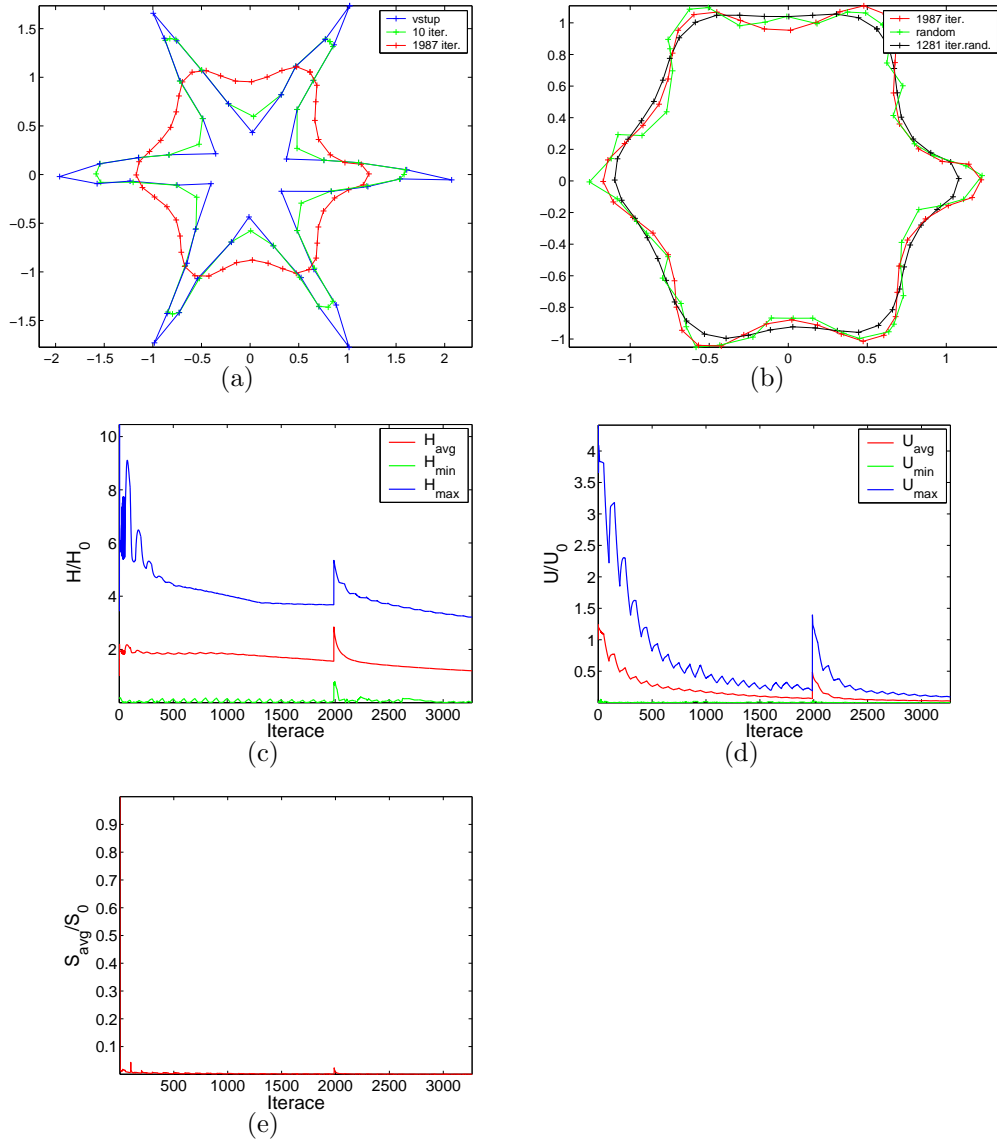


(e)

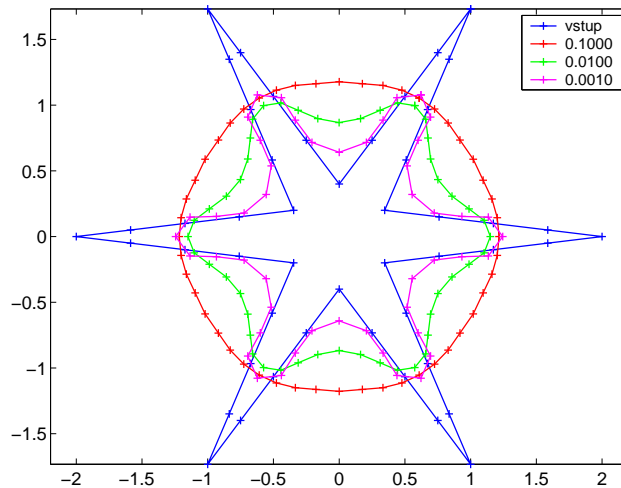
Obrázek 2.11: Vyhlažování extrémně zašuměné lemniskáty. Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 395 iteracích. (a) Průběh vyhlazování. (b) Nezašuměný vstup. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



Obrázek 2.12: Vyhlazování zašuměného šestiúhelníku s dělenými hranami. Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 419 iteracích. Po přidání šumu algoritmus dokonvergoval po 332 iteracích. (a) Průběh vyhlazování. (b) Průběh vyhlazování po přidání šumu k dokonvergované křivce. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



Obrázek 2.13: Vyhlazování zašuměné hvězdice. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. Algoritmus dokonvergoval po 1987 iteracích. Po přidání šumu algoritmus dokonvergoval po 1281 iteracích. (a) Průběh vyhlazování. (b) Průběh vyhlazování po přidání šumu k dokonvergované křivce. (c) Funkce křivosti. (d) Funkce neuniformity. (e) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (2.11.1).



Obrázek 2.14: Vyhlazování hvězdice pro různé nastavení parametru λ . Parametr $\lambda_u = 0.01$. Data k průběhu vyhlazování jsou v tabulce 2.1.

Experimenty jsou provedeny tak, že měníme hodnotu příslušné lambdy a ostatní parametry jsou nastaveny na optimální. Výsledky jsou prezentovány pomocí tabulek, které zachycují hodnoty funkcí $\frac{H_{avg}}{H_0}$, $\frac{U_{avg}}{U_0}$ a $\frac{S_{avg}}{S_0}$ pro testované lambdy. Popis funkcí lze nalézt v 2.11.1. Pokud je v tabulce za hodnotou uvedeno číslo v závorce (X), znamená to, že algoritmus se zvolenými parametry dokonvergoval po X iteracích. Pokud je v tabulce použit symbol „-“, znamená to, že algoritmus se zvolenými parametry, již pro tuto iteraci neběžel (dokonvergoval po menším počtu iterací).

Experimenty se změnou λ :

Vstupní křivkou je hvězdice. Parametr $\lambda_u = 0.01$. Výsledek vyhlazování pro nastavení parametru λ je v tabulce 2.1 a na obrázku 2.14.

Pro nastavení $\lambda = 0.1$ je křivka vyhlazena nejrychleji a nejlépe. V tabulce lze pro $\lambda = 0.1$ pozorovat mírný nárůst křivosti. Myslíme si, že je to způsobeno posunem vrcholů hvězdice, které jsou před začátkem vyhlazování nejbližší ke středu objektu, směrem od středu. V průběhu vyhlazování se vrcholy posunují a průměrná křivost klesá. V jednom okamžiku je potom velikost křivosti, v těchto vrcholech, rovna nule (leží v přímce s jejich sousedy). V průběhu dalšího vyhlazování jsou však vrcholy posouvány dále od středu a jejich křivost se opět zvětšuje, což způsobí nárůst průměrné křivosti. Pro nastavení $\lambda = 0.01$ je dosaženo stejné hodnoty neuniformity (jako pro $\lambda = 0.1$) po mnohem více krocích a křivka je smrštěná, ale zachovává charakter „hvězdice“. Nastavení $\lambda = 0.001$ není vhodné, protože vyhlazování křivky postupuje velmi pomalu. Doporučujeme nastavení parametru λ jako $\lambda = 0.1$, ale uživateli je ponechána volba dle jeho požadavků.

Experimenty se změnou λ_u :

Vstupní křivkou je hvězdice. Parametr $\lambda = 0.1$. Výsledek vyhlazování pro

Hodnota H_{avg}/H_0 po iteraci						
λ	400	800	1300	1700	2100	2535
0.1000	1.265	0.947	0.975(989)	–	–	–
0.0100	2.803	2.706	2.654	2.476	2.308(1991)	–
0.0010	2.844	2.823	2.860	2.881	2.895	2.903

Hodnota U_{avg}/U_0 po iteraci						
λ	400	800	1300	1700	2100	2535
0.1000	0.008	0.004	0.004(989)	–	–	–
0.0100	0.020	0.011	0.006	0.005	0.004(1991)	–
0.0010	0.022	0.016	0.010	0.008	0.006	0.005

Hodnota S_{avg}/S_0 po iteraci						
λ	400	800	1300	1700	2100	2535
0.1000	0.001	0.001	0.001(989)	–	–	–
0.0100	0.003	0.002	0.001	0.001	0.001(1991)	–
0.0010	0.003	0.002	0.001	0.001	0.001	0.001

Tabulka 2.1: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různé nastavení parametru λ vyhlazované hvězdice (viz. obrázek 2.14). Parametr $\lambda_u = 0.01$.

nastavení parametru λ_u je v tabulce 2.2 a na obrázku 2.15.

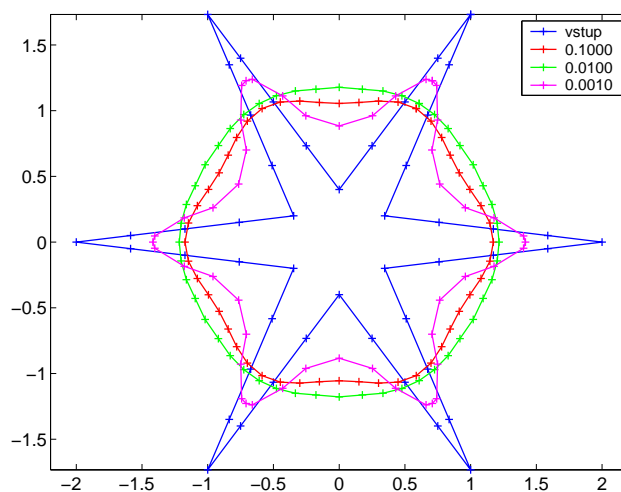
Pro nastavení $\lambda_u = 0.1$ je křivka rychleji vyhlazena, než pro nastavení $\lambda_u = 0.01$, bohužel je však více smršťená. Nastavení $\lambda_u = 0.001$ není vhodné, protože vyhlazování skončilo již po 51 iteracích: Schéma vyhlazování bylo nastaveno na vyhlazování 50 iterací kritériem křivosti a 50 iterací kritériem neuniformity. Algoritmus tedy provedl 50 iterací podle kritéria křivosti a jen jednu iteraci podle kritéria neuniformity. Nejvhodnější nastavení je tedy pro $\lambda_u = 0.01$.

2.11.4 Volba iteračního schématu

V této kapitole ukážeme, jak volba schématu ovlivňuje výsledek algoritmu.

Experiment je proveden tak, že měníme parametry T_h a T_u . Ostatní parametry jsou nastaveny na optimální. Výsledky jsou prezentovány stejně jako v předchozím případě (kapitola 2.11.3).

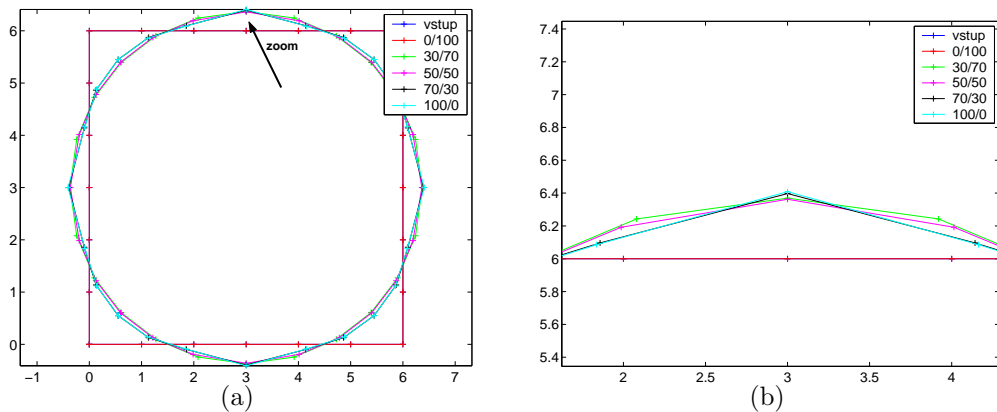
Vstupem algoritmu je čtverec s rozdělenými hranami. Parametry algoritmu jsou $\lambda = 0.1$ a $\lambda_u = 0.01$. Výsledek nastavení parametrů T_h a T_u je v tabulce 2.3. Obrázek 2.16 zobrazuje výsledky vyhlazení křivky s různě nastavenými schématy. Vstupní křivka (modře) a křivka vyhlazovaná (červeně) se schématem ($T_h = 0$ a $T_u = 100$) v obrázku splývají a je tedy vidět pouze červená křivka. Z obrázku je patrné, že vyhlazování jen s kritériem křivosti ($T_h = 100$ a $T_u = 0$) obsahuje „špičky“, které nejsou žádoucí. Chování je podobné také pro nastavení $T_h = 70$ a $T_u = 30$. Pokud bychom vycházeli z hodnot v tabulce 2.3, mohlo by se zdát, že nejvhodnější volba schématu je $T_h = 0$ a $T_u = 100$. Pro takto zvolené parametry je křivka nejrychleji vyhlazena, velikost křivosti je rovna jedné a velikost neuniformity je rovna nule. To je způsobeno tím, že většina vrcholů křivky má křivost rovnou nule a vstupem je již uniformní křivka. Takovou křivku však nechceme, protože



Obrázek 2.15: Vyhlazování hvězdice pro různé nastavení parametru λ_u . Parametr $\lambda = 0.1$. Data k průběhu vyhlazování jsou v tabulce 2.2.

λ_u	Hodnota H_{avg}/H_0 po iteraci					
	160	330	490	660	820	989
0.1000	2.132	1.228	1.055(396)	–	–	–
0.0100	2.614	1.425	1.103	1.030	1.005	0.975
0.0010	3.766(51)	–	–	–	–	–
λ_u	Hodnota U_{avg}/U_0 po iteraci					
	160	330	490	660	820	989
0.1000	0.008	0.001	0.000(396)	–	–	–
0.0100	0.017	0.009	0.007	0.006	0.005	0.004
0.0010	0.036(51)	–	–	–	–	–
λ_u	Hodnota S_{avg}/S_0 po iteraci					
	160	330	490	660	820	989
0.1000	0.010	0.005	0.001(396)	–	–	–
0.0100	0.002	0.007	0.001	0.001	0.001	0.001
0.0010	0.000(51)	–	–	–	–	–

Tabulka 2.2: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různé nastavení parametru λ_u vyhlazované hvězdice (viz. obrázek 2.15). Parametr $\lambda = 0.1$.



Obrázek 2.16: Vyhlazování čtverce s rozdělenými hranami pro různá schémata (nastavení parametrů T_h a T_u). Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$. Data k průběhu vyhlazování jsou v tabulce 2.3. (a) Průběh vyhlazování. (b) Zvětšená část křivky, ve které se pro nastavení $T_u = 0$ objevuje „špička“.

obsahuje „rohů“. Křivka je tedy nejlépe (bez „špiček“ a „rohů“) vyhlazena pro nastavení $T_h = T_u = 50$.

2.11.5 Shrnutí

V této kapitole shrneme provedené experimenty.

Experimenty prokázaly, že výstupem algoritmu jsou vyhlazené křivky a velikost plochy, kterou křivka ohraničuje, je zachována. Bylo také ukázáno, že vyhlazením jsou zachovány inflexní body, které křivka obsahovala před zašuměním. Pokud je vstupem pravidelný n -úhelník, výsledek vyhlazení je totožný se vstupem algoritmu. Algoritmus je schopný vyhladit i extrémně zašuměné křivky a pokud je vyhlazená křivka znovu zašuměna a následně opět vyhlazena, výstupem je křivka stejného tvaru, jako před zašuměním. Dále bylo ukázáno, jak vstupní parametry algoritmu ovlivňují výsledek vyhlazení křivky. Z experimentů vyplynulo, že vhodným nastavením pro většinu křivek je kombinace parametrů $\lambda = 0.1$, $\lambda_u = 0.01$, $T_h = T_u = 50$, $S_h = 0.0001$ a $S_u = 0.0001$.

T_h/T_u	Hodnota H_{avg}/H_0 po iteraci					
	140	280	410	550	690	825
0/100	1.000(1)	–	–	–	–	–
30/70	1.815	1.617	1.478	1.363	1.293	1.274
50/50	1.909	1.614	1.472	1.401(549)	–	–
70/30	1.865	1.854(263)	–	–	–	–
100/0	1.851	1.904(194)	–	–	–	–

T_h/T_u	Hodnota U_{avg}/U_0 po iteraci					
	140	280	410	550	690	825
0/100	0.000(1)	–	–	–	–	–
30/70	0.065	0.050	0.044	0.034	0.021	0.015
50/50	0.075	0.057	0.048	0.042(549)	–	–
70/30	0.077	0.077(263)	–	–	–	–
100/0	0.081	0.085(194)	–	–	–	–

T_h/T_u	Hodnota S_{avg}/S_0 po iteraci					
	140	280	410	550	690	825
0/100	0.000(1)	–	–	–	–	–
30/70	0.015	0.011	0.021	0.008	0.005	0.002
50/50	0.021	0.013	0.010	0.002(549)	–	–
70/30	0.016	0.002(263)	–	–	–	–
100/0	0.014	0.002(194)	–	–	–	–

Tabulka 2.3: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různá schémata (nastavení parametrů T_h a T_u) vyhlazovaného čtverce s rozdělenými hranami (viz. obrázek 2.16). Parametry $\lambda = 0.1$ a $\lambda_u = 0.01$.

Kapitola 3

Vyhlazování uzavřených povrchů

Tato kapitola popisuje vyhlazování uzavřených povrchů.

3.1 Vyhlazený povrch

Popis vyhlazeného povrchu je analogický s popisem vyhlazené křivky: Cílem algoritmu je postupnými posuny jednotlivých vrcholů daného povrchu docílit takového povrchu, který je *vyhlazený*. Vyhlazením rozumíme optimalizaci dvou vlastností povrchu: křivosti a neuniformity. Křivost udává křivost daného povrchu a neuniformita nerovnoměrnost mezi vzdálenostmi vrcholu povrchu k jeho bezprostředním sousedům. Řekneme, že povrch je ideálně vyhlazen, když má křivost ve všech vrcholech rovnu průměru křivostí bezprostředních sousedů a neuniformitu rovnu nule (vzdálenosti vrcholu ke všem bezprostředním sousedům vrcholu jsou shodné).

3.2 Křivost na spojitém povrchu

Nechť C je C^3 spojitý povrch, který minimalizuje

$$J = \int (\nabla H(u, v))^2 dudv, \quad (3.1)$$

kde H je velikost křivosti. Funkce $H = H(u, v, t)$, kde u, v je parametrizace povrchu a t je čas. Rovnice (3.1) je Dirichletův integrál [15]. Nutná podmínka pro extrém funkce J je ve tvaru Laplaceovy rovnice

$$\Delta H(u, v) = \frac{\partial^2 H}{\partial u^2} + \frac{\partial^2 H}{\partial v^2} = 0. \quad (3.2)$$

Funkce H je tedy harmonická funkce. Funkci H tedy můžeme nalézt jako

řešení difusní rovnice

$$\frac{\partial H}{\partial t} = \lambda \left(\frac{\partial^2 H}{\partial u^2} + \frac{\partial^2 H}{\partial v^2} \right), \quad 0 < \lambda < 1, \quad (3.3)$$

kde λ je koeficient difuze, a pro kterou platí

$$\lim_{t \rightarrow \infty} \frac{\partial H}{\partial t} = 0 \quad (3.4)$$

a H je harmonická funkce.

Pro funkci H platí věta o aritmetickém průměru [15]:

Nechť funkce u je v oblasti Ω spojitá. Hodnoty u ve středu každé n -rozměrné koule $\bar{K} \in \Omega$ jsou rovny aritmetickému průměru hodnot na hranici \bar{K} tehdy a jen tehdy když u je harmonická.

My použijeme difuzi pro vyhlazování povrchu. Potom λ je rychlost vyhlazování.

Řekneme, že diskretní povrch je zadán triangulovanou sítí, kde $V = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ je posloupnost vrcholů povrchu. Aktuální vrchol povrchu označíme \mathbf{a} .

Diskretizací věty o aritmetickém průměru dostaneme

$$\bar{H} - H = 0, \quad (3.5)$$

kde H je křivost ve vrcholu \mathbf{a} , $\bar{H} = \frac{1}{n} \sum_{i=1}^n H_i$ je průměr velikostí křivostí bezprostředních sousedů vrcholu \mathbf{a}_i a n je počet bezprostředních sousedů vrcholu. Předpokládáme uniformní okolí o poloměru jedna. Požadavek na uniformitu je realistický, protože bude existovat člen, který jej v průběhu konvergence zaručí.

Výraz $\frac{\partial^2 H}{\partial u^2} + \frac{\partial^2 H}{\partial v^2}$ ve vrcholu \mathbf{a} můžeme aproximovat

$$\left(\frac{\partial^2 H}{\partial u^2} + \frac{\partial^2 H}{\partial v^2} \right) \approx \bar{H} - H, \quad (3.6)$$

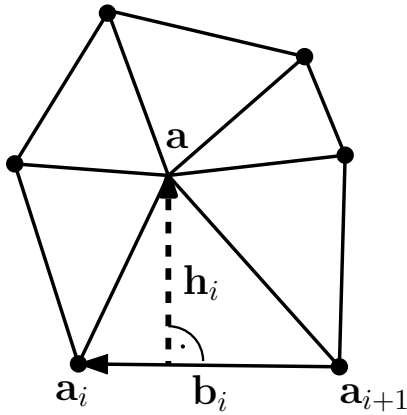
Dosazením výrazu (3.6) do výrazu (3.3) a nahrazením výrazu $\frac{\partial H}{\partial t}$ výrazem $H^{k+1} - H^k$ dostaneme definici pro výpočet nové křivosti ve vrcholu \mathbf{a} . Index k určuje iteraci. Potom

$$H^{k+1} = (1 - \lambda)H^k + \lambda\bar{H}^k, \quad 0 < \lambda < 1. \quad (3.7)$$

3.3 Definice křivosti

Stejně jako u křivek, tak i u povrchu se v průběhu vyhlazování pracuje s křivostí ve vrcholu povrchu.

Zavedeme vektor křivosti a uvedeme výpočet velikosti tohoto vektoru. Řekneme, že S je diskretně zadaný povrch reprezentovaný triangulovanou



Obrázek 3.1: Okolí vrcholu \mathbf{a} .

síti (viz. 1.2.1). Řekneme, že $V = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ jsou vrcholy povrchu S , kde $\mathbf{a}_j = [x_j, y_j, z_j]$. Dále v textu budeme aktuálně zpracovávaný vrchol \mathbf{a}_j označovat \mathbf{a} . Všechny bezprostřední sousedy vrcholu \mathbf{a} označíme $O = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$. K indexování bezprostředních sousedů budeme používat index $i = (1, 2, \dots, n)$. Tedy $\mathbf{a}_i = [x_i, y_i, z_i]$ je i -tý bezprostřední soused vrcholu \mathbf{a} .

Pro účely této práce křivost povrchu S ve vrcholu \mathbf{a} definujeme jako vektor $\mathbf{H} = [h_x, h_y, h_z]$ a

$$\mathbf{H} = \frac{3}{2} \cdot \frac{\nabla \sum_{i=1}^n A_i^2}{\sum_{i=1}^n A_i^2}, \quad (3.8)$$

kde

$$A_i = \frac{1}{2} \|\mathbf{h}_i\| \|\mathbf{b}_i\| = \frac{1}{2} h_i b_i \quad (3.9)$$

je obsah trojúhelníku, který je tvořen vrcholy \mathbf{a} , \mathbf{a}_i a \mathbf{a}_{i+1} . Dále \mathbf{h}_i je výška trojúhelníku vedoucí z vrcholu \mathbf{a} , $h_i = \|\mathbf{h}_i\|$ je velikost výšky trojúhelníku, $\mathbf{b}_i = \mathbf{a}_i - \mathbf{a}_{i+1}$ a $b_i = \|\mathbf{b}_i\|$ je velikost vektoru \mathbf{b}_i (viz. obrázek 3.1) a výraz

$$\nabla \sum_{i=1}^n A_i^2 = \sum_{i=1}^n \frac{1}{4} b_i^2 \cdot \left(\frac{\partial h_i^2}{\partial x}, \frac{\partial h_i^2}{\partial y}, \frac{\partial h_i^2}{\partial z} \right). \quad (3.10)$$

Dále si vyjádříme

$$\mathbf{h}_i = \mathbf{P}_i(\mathbf{a} - \mathbf{a}_i), \quad (3.11)$$

$$\mathbf{h}_i^2 = h_i^2 = (\mathbf{a} - \mathbf{a}_i)^\top \mathbf{P}_i(\mathbf{a} - \mathbf{a}_i), \quad (3.12)$$

kde \mathbf{P}_i je projekční matice, kterou definujeme jako $\mathbf{P}_i = \mathbf{E} - \frac{\mathbf{b}_i \mathbf{b}_i^\top}{b_i^2}$. Výraz \mathbf{E} označuje jednotkovou matici. Pro projekční matici platí $\mathbf{P}_i^2 = \mathbf{P}_i$ a $\mathbf{P}_i^\top = \mathbf{P}_i$.

Dosazením definovaných výrazů do rovnice (3.8) dostaneme konečnou definici vektoru křivosti ve vrcholu \mathbf{a} plochy S . Tedy

$$\mathbf{H} = \frac{3}{2} \cdot \frac{\sum_{i=1}^n \mathbf{B}_i(\mathbf{a} - \mathbf{a}_i)}{\sum_{i=1}^n (\mathbf{a} - \mathbf{a}_i)^\top \mathbf{B}_i(\mathbf{a} - \mathbf{a}_i)}, \quad (3.13)$$

kde $\mathbf{B}_i = b_i^2 \mathbf{P}_i = b_i^2 \mathbf{E} - \mathbf{b}_i \mathbf{b}_i^\top$.

Směr vektoru křivosti ve vrcholu \mathbf{a} je shodný se směrem vektoru $\sum_{i=1}^n \mathbf{h}_i$, kde n je počet bezprostředních sousedů vrcholu \mathbf{a} .

Velikost křivosti $H = \|\mathbf{H}\| = \sqrt{(h_x)^2 + (h_y)^2 + (h_z)^2}$.

3.3.1 Vlastnosti vektoru křivosti \mathbf{H} a velikosti křivosti H

Zvolený vektor křivosti má vlastnosti, které jsou užitečné pro úlohu vyhlazování:

- Velikost křivosti je invariantní k rotaci a translaci povrchu.
- Z definice \mathbf{H} plyne, že v případě pravidelného mnohostěnu směřuje \mathbf{H} do středu tělesa a $H = \frac{1}{r}$, kde r je poloměr opsané kulové plochy mnohostěnu.
- Velikost křivosti není monotonní funkcí výšky okolí vrcholu \mathbf{a} . Toto chování ukážeme na jehlanu s podstavou tvořenou pravidelným šestiúhelníkem a hlavním vrcholem \mathbf{a} . Vrchol \mathbf{a} budeme vytahovat a tím zvětšovat výšku tělesa. Funkce velikosti křivosti v závislosti na výšce tělesa je na obrázku 3.2.
- Leží-li vrchol \mathbf{a} a jeho bezprostřední sousedé v jedné rovině, funkce velikosti křivosti v závislosti na poloze vrcholu v této rovině je na obrázku 3.3.

Stejně jako u křivek budeme pracovat v průběhu hledání nové polohy vrcholu \mathbf{a} s velikostí křivosti, místo vektoru křivosti. Vysvětlení této volby je analogické k vysvětlení pro křivky (viz. 2.3.2).

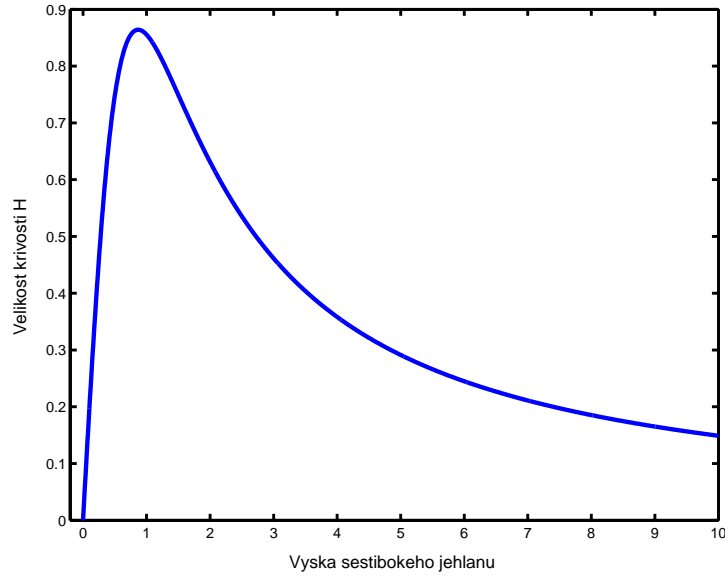
3.4 Definice neuniformity

V průběhu vyhlazování povrchů pracujeme také s neuniformitou. S neuniformitou pracujeme analogicky jako s křivostí.

Zadefinujeme vektor neuniformity a výpočet velikosti tohoto vektoru ve vrcholu. Definice povrchu a ostatních použitých veličin je uvedena v kapitole 3.3.

Bezprostřední sousedé vrcholu \mathbf{a} jsou označeni jako $O = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$ a k indexování bezprostředních sousedů je používáno indexace $i = (1, 2, \dots, n)$. Dále také platí, že $\mathbf{a}_0 = \mathbf{a}_n$ a $\mathbf{a}_{n+1} = \mathbf{a}_1$. Projekci do směru určeného vektorem $\mathbf{d}_i = \mathbf{a}_{i+1} - \mathbf{a}_{i-1}$ budeme označovat $\mathbf{P}_{\mathbf{d}_i}$. Projekci spočítáme:

$$\mathbf{P}_{\mathbf{d}_i} = \frac{\mathbf{d}_i (\mathbf{d}_i)^\top}{\|\mathbf{d}_i\|^2}, \quad (3.14)$$



Obrázek 3.2: Závislost velikosti křivosti H na výšce šestibokého jehlanu. Délka hrany podstavy je rovna jedné.

kde $\|\mathbf{d}_i\|$ je velikost vektoru \mathbf{d}_i .

Neuniformitu povrchu S ve vrcholu \mathbf{a} definujeme jako vektor $\mathbf{U} = [u_x, u_y, u_z]$. Lze-li bezprostředními sousedy vrcholu proložit rovinu, pak \mathbf{U} je s touto rovinou rovnoběžný. Vektor neuniformity definujeme

$$\mathbf{U} = \frac{1}{n} \sum_{i=1}^n \mathbf{U}_i, \quad (3.15)$$

kde n je počet bezprostředních sousedů vrcholu \mathbf{a} a \mathbf{U}_i je

$$\mathbf{U}_i = \frac{-\mathbf{P}_{\mathbf{d}_i} \left(\nabla \left(\|\mathbf{a} - \mathbf{a}_{i-1}\|^2 + \|\mathbf{a} - \mathbf{a}_{i+1}\|^2 \right) \right)}{4}. \quad (3.16)$$

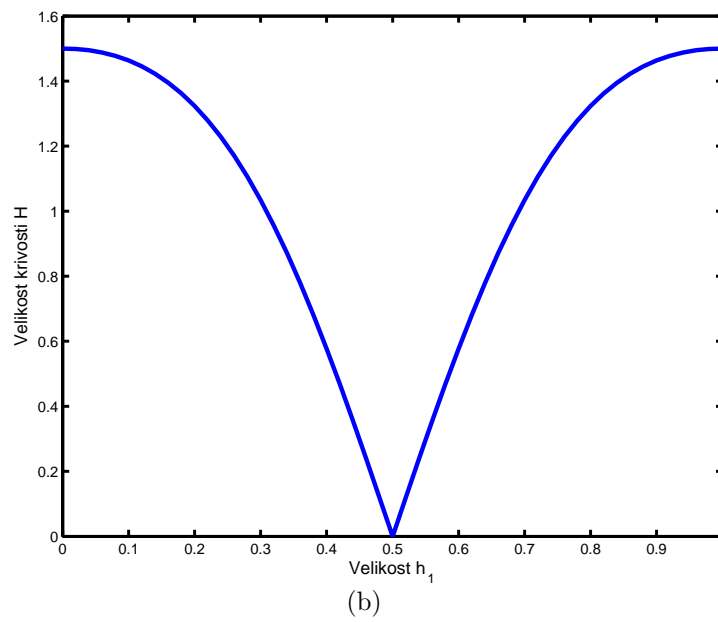
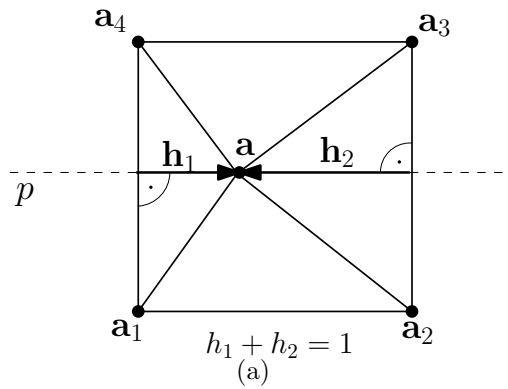
Výraz

$$\begin{aligned} & \nabla \left(\|\mathbf{a} - \mathbf{a}_{i-1}\|^2 + \|\mathbf{a} - \mathbf{a}_{i+1}\|^2 \right) = \\ & \left(\frac{\partial \left(\|\mathbf{a} - \mathbf{a}_{i-1}\|^2 + \|\mathbf{a} - \mathbf{a}_{i+1}\|^2 \right)}{\partial x}, \frac{\partial \left(\|\mathbf{a} - \mathbf{a}_{i-1}\|^2 + \|\mathbf{a} - \mathbf{a}_{i+1}\|^2 \right)}{\partial y}, \frac{\partial \left(\|\mathbf{a} - \mathbf{a}_{i-1}\|^2 + \|\mathbf{a} - \mathbf{a}_{i+1}\|^2 \right)}{\partial z} \right). \end{aligned}$$

Po upravení čitatele výrazu a dosazení do výrazu (3.15) vznikne konečná definice vektoru neuniformity ve vrcholu \mathbf{a} povrchu S . Tedy

$$\mathbf{U} = \frac{1}{n} \sum_{i=1}^n \frac{-\mathbf{P}_{\mathbf{d}_i} \left((\mathbf{a} - \mathbf{a}_{i-1}) + (\mathbf{a} - \mathbf{a}_{i+1}) \right)}{2}. \quad (3.17)$$

Velikost vektoru neuniformity určuje vzdálenost vrcholu \mathbf{a} od pomyslného středu jeho bezprostředních sousedů. V ideálním případě lze bezprostředním



Obrázek 3.3: Závislost velikosti křivosti H na pozici vrcholu \mathbf{a} . S vrcholem může být pohybováno jen ve čtverci vymezeném bezprostředními sousedy vrcholu \mathbf{a} . Pro názornost je s vrcholem pohybováno jen po přímce p .

sousedům vrcholu opsat kružnici. Velikost vektoru neuniformity v tomto vrcholu pak určuje jeho vychýlení od středu této kružnice. Čím větší je velikost vektoru, tím více je vrchol od středu vzdálen. V ideálním případě, kdy je vrchol \mathbf{a} totožný se středem opsané kružnice, je velikost vektoru neuniformity rovna nule.

Velikost vektoru neuniformity $U = \|\mathbf{U}\| = \sqrt{u_x^2 + u_y^2 + u_z^2}$.

Diskrétní difusi pro neuniformitu definujeme předpisem

$$U^{k+1} = (1 - \lambda_u)U^k, \quad (3.18)$$

kde $0 < \lambda_u < 1$, neboť chceme, aby v ekvilibriu byla neuniformita rovna nule.

3.5 Algoritmus pro vyhlazování uzavřeného povrchu

Pro vyhlazování uzavřeného povrchu jsou, shodně jako pro křivky, použita dvě kritéria: kritérium křivosti a kritérium neuniformity. Tato kritéria jsou v průběhu vyhlazování používána podle vyhlazovacího schématu, které je shodné se schématem pro vyhlazování křivek a lze jej nalézt v sekci 2.5. Srovnání různých typů schémat pro uzavřené povrchy je v kapitole 3.8.4.

Cílem algoritmu je postupným posunem vrcholů povrchu dosáhnout nových konečných vrcholů povrchu tak, aby křivost a neuniformita v těchto nových vrcholech byly v rovnováze (vrcholy se již nepohybují). Snahou algoritmu je také co nejméně zmenšit objem tělesa, který povrch ohraničuje.

Podle schématu vyhlazování je v průběhu algoritmu pracováno odděleně s křivostí a neuniformitou. Kritéria jsou střídána po daném (určeném uživatelem) počtu iterací. V obou případech je algoritmem určena nová hodnota, křivosti nebo neuniformity, ve vrcholu. Nová poloha vrcholu, kterému odpovídá spočítaná hodnota, však není známá. Určení této nové polohy vrcholu je dalším úkolem algoritmu.

Tento postup je opakován dokud není splněno kritérium zastavení. Kritéria zastavení se shodují s kritérii pro zastavení algoritmu pro vyhlazování křivek (viz. 2.10). V každé iteraci algoritmu jsou zpracovány všechny vrcholy povrchu a to tak, že pro každý vrchol jsou provedeny všechny kroky algoritmu a teprve poté se v iteraci zpracovává další vrchol povrchu. Podrobný popis jednotlivých kroků algoritmu podle kritéria křivosti je v kapitole 3.6 a podle kritéria neuniformity v kapitole 3.7.

V případě, že se v aktuální iteraci (k) pracuje s kritériem křivosti, jsou pro každý vrchol provedeny kroky:

1. Výpočet vektoru křivosti \mathbf{H}^k a výpočet velikosti křivosti H^k (viz. 3.6.1).
2. Výpočet velikosti nové křivosti H^{k+1} (viz. 3.6.2).

3. Výpočet polohy nového vrcholu \mathbf{a}^{k+1} (viz. 3.6.3).
4. Nahrazení vrcholu \mathbf{a}^k povrchu S novým vrcholem \mathbf{a}^{k+1} (viz. 3.5.1).

V případě, že se v aktuální iteraci (k) pracuje s kritériem neuniformity, jsou pro každý vrchol provedeny kroky:

1. Výpočet vektoru \mathbf{U}^k (viz. 3.7.1).
2. Výpočet polohy nového vrcholu \mathbf{a}^{k+1} (viz. 3.7.2).
3. Nahrazení vrcholu \mathbf{a}^k povrchu S novým vrcholem \mathbf{a}^{k+1} (viz.3.5.1).

3.5.1 Určení nových vrcholů povrchu

Pro k -tou iteraci provedeme výpočet nových vrcholů povrchu tak, že postupně zpracujeme všechny vrcholy. Zpracováním je myšleno provedení všech kroků algoritmu pro zadaný vrchol. V posledním kroku je spočítán nový vrchol. Nahradíme tedy starý vrchol novým a pokračujeme zpracováním dalšího vrcholu povrchu. V $(k + 1)$ -ní iteraci algoritmu pracujeme jako s výchozími vrcholy s těmi, které jsme postupně spočítali v průběhu k -té iterace. Na konci každé iterace algoritmu jsou již všechny vrcholy $V = \{\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_m^k\}$ povrchu nahrazeny novými vrcholy $V = \{\mathbf{a}_1^{k+1}, \mathbf{a}_2^{k+1}, \dots, \mathbf{a}_m^{k+1}\}$.

3.6 Algoritmus vyhlazování podle kritéria křivosti

V této kapitole popíšeme podrobně jednotlivé kroky algoritmu vyhlazování podle kritéria křivosti.

3.6.1 Výpočet vektoru křivosti \mathbf{H}^k a jeho velikosti H^k

Pro výpočet difuze křivosti je nezbytné spočítat vektor křivosti a jeho velikost. Vektor křivosti je také využit k výpočtu nového vrcholu povrchu.

Vrcholy $V = \{\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_m^k\}$ jsou aktuální vrcholy povrchu S , se kterými se pracuje v k -té iteraci algoritmu. Potom $\mathbf{a}_j^k = [x_j^k, y_j^k, z_j^k]$ a aktuálně zpracováváný vrchol budeme označovat \mathbf{a}^k . Bezprostřední sousedé vrcholu \mathbf{a}^k jsou pak $O = (\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_n^k)$. Dále uvedeme substituce, které použijeme k vyjádření vektoru křivosti $\mathbf{H}^k = [h_x^k, h_y^k, h_z^k]$. Abychom zápis zjednodušili, některé výrazy neobsahují indexaci určující (k)-tou iteraci. Neznamená to však, že se jejich hodnota v průběhu iterací nemění! Potom

$$\mathbf{B} = \sum_{i=1}^n \mathbf{B}_i ,$$

$$\begin{aligned}
\mathbf{b} &= \sum_{i=1}^n \mathbf{B}_i \mathbf{a}_i^k, \\
c &= \sum_{i=1}^n (\mathbf{a}_i^k)^\top \mathbf{B}_i \mathbf{a}_i^k, \\
\hat{\mathbf{a}}^k &= \mathbf{B} \mathbf{a}^k - \mathbf{b}, \\
d &= c - \mathbf{b}^\top \mathbf{B}^{-1} \mathbf{b},
\end{aligned}$$

kde \mathbf{B}_i je zadefinováno v (3.13). Vzorec pro výpočet vektoru křivosti \mathbf{H}^k v k -té iteraci algoritmu s využitím výše uvedených substitucí je

$$\mathbf{H}^k = \frac{3}{2} \cdot \frac{\mathbf{B} \mathbf{a}^k - \mathbf{b}}{(\mathbf{a}^k)^\top \mathbf{B} \mathbf{a}^k - 2(\mathbf{a}^k)^\top \mathbf{b} + c}, \quad (3.19)$$

nebo

$$\mathbf{H}^k = \frac{3}{2} \cdot \frac{\hat{\mathbf{a}}^k}{(\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d}. \quad (3.20)$$

Velikost křivosti $H^k = \|\mathbf{H}^k\| = \sqrt{(h_x^k)^2 + (h_y^k)^2 + (h_z^k)^2}$.

3.6.2 Výpočet hodnoty nové křivosti H^{k+1}

Ve vrcholu \mathbf{a}^k spočítáme novou hodnotu křivosti, která bude charakterizovat nový vrchol povrchu. Nová hodnota je závislá na vektoru i velikosti křivosti vrcholu \mathbf{a}^k a na vektorech i velikostech křivostí bezprostředních sousedů tohoto vrcholu. Hodnota nové křivosti může nabývat kladných i záporných hodnot. V případě, že je hodnota nové křivosti záporná, pak směr vektoru křivosti v nově spočítaném vrcholu \mathbf{a}^{k+1} (v případě, že jsme dosud nepohybovali se sousedními vrcholy) bude opačný než směr vektoru křivosti \mathbf{H}^k ve vrcholu \mathbf{a}^k .

Velikost nové křivosti spočítáme podle rovnice (3.7). Potom

$$H^{k+1} = (1 - \lambda)H^k + \lambda\bar{H}^k, \quad 0 < \lambda < 1, \quad (3.21)$$

kde

$$\bar{H}^k = \frac{\sum_{i=1}^n s_i^k \cdot H_i^k}{n}. \quad (3.22)$$

Výraz H_i^k je velikost křivosti i -tého bezprostředního souseda vrcholu \mathbf{a}^k a s_i^k je znaménko, kterým se přenáší vektor křivosti \mathbf{H}_i^k do vrcholu \mathbf{a}^k .

Rekonstrukce znamének křivostí

Při výpočtu nové hodnoty křivosti ve vrcholu \mathbf{a}^k pracujeme s křivostmi jen jako se skaláry. Přesto potřebujeme znát znaménko, se kterým se bude velikost křivosti bezprostředního souseda přenášet do aktuálně počítaného

vrcholu. Musíme tedy zjistit, v jaké vzájemné poloze vůči povrchu jsou vektory křivosti bezprostředních sousedů s vektorem křivosti ve vrcholu \mathbf{a}^k .

Řešení rekonstrukce znamének nelze provést explicitním zorientováním povrchu, protože existují povrchy, které nelze zorientovat (např. Moebiova páska, Kleinova láhev).

Rekonstrukci znaménka nebudeme řešit jako u křivek přímým porovnáváním vektoru křivosti v aktuálním vrcholu s vektorem křivosti bezprostředního souseda. Řešení budeme hledat tak, že nalezneme orientaci jednotlivých vektorů křivosti vůči lokálně zorientovanému povrchu. Lokální zorientování vždy existuje a určíme ho v průběhu výpočtu znaménka.

Pro výpočet zavedeme označení s^k , které bude udávat orientaci vektoru křivosti \mathbf{H}^k vůči lokálně zorientovanému povrchu. Znaménko s^k není ve vzorci (3.21) použito. Hodnota křivosti ve vrcholu \mathbf{a}^k je kladná, proto s^k bude určovat kladný směr. Vyjde-li záporné, orientace všech znamének sousedů se musí změnit na opačnou! Výpočet s^k je shodný s výpočtem znaménka s_i^k . Výraz s_i^k označuje znaménko vektoru křivosti \mathbf{H}_i^k ve vrcholu \mathbf{a}_i^k lokálně zorientovaného povrchu S .

Lokální orientace zaručuje korektní určení znamének pro výpočet nové hodnoty křivosti H^{k+1} . Lokálně zorientovaný povrch lze popsat jako:

- Okolí vrcholu \mathbf{a}^k musí být shodně zorientováno. To znamená, že všechny trojúhelníky, které obsahují vrchol \mathbf{a}^k , musí mít shodnou orientaci.
- Pro všechny bezprostřední sousedy \mathbf{a}_i^k a jejich okolí musí platit: Okolí vrcholu \mathbf{a}_i^k musí být shodně zorientováno a to tak, aby orientace trojúhelníků odpovídala orientaci okolí vrcholu \mathbf{a}^k .

Dále uvedeme výpočet znaménka s^k . Výpočet znaménka s_i^k je analogický.

Trojúhelníky sítě, které obsahují vrchol \mathbf{a}^k , budeme označovat $T = \{t_i\}_{i=1}^n$ a t_i je určen vrcholy \mathbf{a}^k , \mathbf{a}_i^k , \mathbf{a}_{i+1}^k . Normálu trojúhelníku t_i označíme \mathbf{N}_i a velikost normály $\|\mathbf{N}_i\|$.

Znaménko s^k určíme:

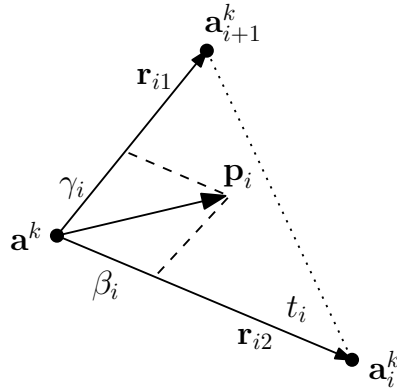
1. Pro každý trojúhelník t_i spočítáme projekci \mathbf{p}_i vektoru \mathbf{H}^k do roviny určené trojúhelníkem jako

$$\mathbf{p}_i = \mathbf{P}_{\mathbf{N}_i}(\mathbf{H}^k), \quad (3.23)$$

$$\text{kde } \mathbf{P}_{\mathbf{N}_i} = \mathbf{E} - \frac{\mathbf{N}_i(\mathbf{N}_i)^\top}{\|\mathbf{N}_i\|^2}.$$

2. Pro každý vektor projekce určíme, zda \mathbf{p}_i leží v rovinném úhlu určeném rameny $\mathbf{r}_{i1} = \mathbf{a}_{i+1}^k - \mathbf{a}^k$, $\mathbf{r}_{i2} = \mathbf{a}_i^k - \mathbf{a}^k$ trojúhelníku t_i (viz. obrázek 3.4). Pro vektor projekce platí

$$\mathbf{p}_i = \beta_i \mathbf{r}_{i1} + \gamma_i \mathbf{r}_{i2}. \quad (3.24)$$



Obrázek 3.4: Projekce \mathbf{p}_i leží v rovinném úhlu určeném \mathbf{r}_{i1} a \mathbf{r}_{i2} .

Vynásobíme-li rovnici 3.24 výrazem \mathbf{r}_{i1} a také výrazem \mathbf{r}_{i2} dostaneme dvě rovnice

$$(\mathbf{r}_{i1})^\top \mathbf{p}_i = \beta_i \|\mathbf{r}_{i1}\|^2 + \gamma_i (\mathbf{r}_{i1})^\top \mathbf{r}_{i2}, \quad (3.25)$$

$$(\mathbf{r}_{i2})^\top \mathbf{p}_i = \beta_i (\mathbf{r}_{i2})^\top \mathbf{r}_{i1} + \gamma_i \|\mathbf{r}_{i2}\|^2.$$

Řešením soustavy získáme skalární hodnoty β_i a γ_i . Možné vztahy β_i a γ_i jsou následující:

- Hodnota $\beta_i > 0$ a současně $\gamma_i > 0$. Vektor \mathbf{p}_i leží v rovinném úhlu. Budeme říkat, že vektor projekce \mathbf{p}_i leží v trojúhelníku t_i .
 - Hodnota $\beta_i = 0$ a současně $\gamma_i > 0$, nebo $\beta_i > 0$ a současně $\gamma_i = 0$. Vektor projekce je totožný s \mathbf{r}_{i1} nebo \mathbf{r}_{i2} , tedy leží na rameni trojúhelníku t_i .
 - Hodnota $\beta_i = 0$ a současně $\gamma_i = 0$. Vektor projekce leží ve vrcholu \mathbf{a}^k .
 - Hodnota $\beta_i < 0$ a současně $\gamma_i < 0$. Vektor projekce leží mimo rovinný úhel a ramena, která jej určují a neleží ve vrcholu \mathbf{a}^k .
3. Známe-li vztah vektoru projekce \mathbf{p}_i k trojúhelníku t_i pro všechny trojúhelníky T , potom můžeme nalézt trojúhelník (k němuž známe \mathbf{p}_i), podle kterého určíme orientaci vektoru křivosti \mathbf{H}^k . Mohou nastat případy:
- Existuje alespoň jeden vektor \mathbf{p}_i , který leží v jemu odpovídajícím trojúhelníku t_i . Pokud existuje jen jeden takový vektor \mathbf{p}_i , potom t_i je řešením. Jinak vybereme z těchto \mathbf{p}_i ten, který minimalizuje $\|\mathbf{H}^k - \mathbf{p}_i\|$. Řešením je odpovídající trojúhelník t_i .
 - Žádný vektor projekce \mathbf{p}_i nenáleží trojúhelníku t_i a alespoň jeden leží na rameni trojúhelníku t_i . Pokud existuje jen jeden takový

vektor \mathbf{p}_i , potom řešením je libovolný ze dvou trojúhelníků, které obsahují hranu, na níž leží vektor \mathbf{p}_i . Pokud existuje více takových projekcí, výběr a označení provedeme stejně jako v předchozím případě.

- Žádný vektor projekce \mathbf{p}_i nenáleží trojúhelníku t_i a ani neleží na rameni trojúhelníku. Může existovat vektor projekce, který leží ve vrcholu \mathbf{a}^k . Vybereme libovolný vektor projekce \mathbf{p}_i a řešením je trojúhelník t_i .

4. Znaménko přenosu s^k určíme podle vzájemné polohy vektorů \mathbf{H}^k a \mathbf{p}_i . Vektor \mathbf{p}_i odpovídá trojúhelníku t_i , který je řešením některého z předchozích bodů. Znaménko s^k spočítáme podle

$$s^k = \text{sign}((\mathbf{H}^k - \mathbf{p}_i)\mathbf{N}_i). \quad (3.26)$$

3.6.3 Výpočet nového vrcholu \mathbf{a}^{k+1}

Ve vrcholu je známá hodnota nové křivosti. K tomu, abychom mohli s tímto vrcholem dále pracovat, potřebujeme znát i jeho novou polohu. Nový vrchol \mathbf{a}^{k+1} povrchu S bude výchozí pro výpočty v $(k+1)$ -ní iteraci algoritmu. Chceme, aby nový vektor křivosti ve vrcholu \mathbf{a}^{k+1} byl rovnoběžný s vektorem křivosti ve vrcholu \mathbf{a}^k . Zapišeme tedy, že

$$\hat{\mathbf{a}}^{k+1} = (1 + \alpha^k)\hat{\mathbf{a}}^k. \quad (3.27)$$

Využitím substitucí definovaných v 3.6.1 můžeme předchozí rovnost zapsat jako

$$\mathbf{B}\mathbf{a}^{k+1} - \mathbf{b} = \hat{\mathbf{a}}^{k+1} = (1 + \alpha^k)\hat{\mathbf{a}}^k = (1 + \alpha^k)(\mathbf{B}\mathbf{a}^k - \mathbf{b}).$$

Odvodíme vzorec pro výpočet nového vrcholu \mathbf{a}^{k+1} . Tedy

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \alpha^k \mathbf{B}^{-1} \hat{\mathbf{a}}^k. \quad (3.28)$$

Poznámka:

Matice \mathbf{B} je regulární a tedy matice \mathbf{B}^{-1} existuje.

Regularita matice \mathbf{B} : Pro všechna $i = 1 \dots n$, kde n je počet bezprostředních sousedů vrcholu \mathbf{a}^k , platí, že \mathbf{B}_i je projekční matice a $\mathbf{B}_i \mathbf{b}_i = 0$. Z toho plyne, že \mathbf{B}_i je singulární a \mathbf{b}_i je nulovým prostorem této matice. Pokud existují alespoň dva vektory \mathbf{b}_i , které jsou lineárně nezávislé, matice \mathbf{B} je regulární. Tato podmínka je pro uzavřené okolí vrcholu \mathbf{a}^k vždy splněna.

K výpočtu vrcholu \mathbf{a}^{k+1} je nutné znát hodnotu α^k . Výpočet hodnoty α^k je uveden dále.

Odvození α^k

Ze vzorce (3.20) pro výpočet \mathbf{H}^k platí

$$\begin{aligned} \mathbf{H}^{k+1} &= \frac{3}{2} \cdot \frac{\hat{\mathbf{a}}^{k+1}}{(\hat{\mathbf{a}}^{k+1})^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^{k+1} + d} = \frac{3}{2} \cdot \frac{(1 + \alpha^k) \hat{\mathbf{a}}^k}{(1 + \alpha^k)^2 (\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d} = \quad (3.29) \\ &= \mathbf{H}^k \cdot \frac{(1 + \alpha^k) ((\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d)}{(1 + \alpha^k)^2 (\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d} . \end{aligned}$$

Vektor \mathbf{H}^{k+1} je vektor křivosti pro novou pozici vrcholu \mathbf{a}^{k+1} povrchu S . Polohu ostatních vrcholů povrchu uvažujeme tu, která je výchozí pro výpočet v iteraci k . Pro další odvození z rovnice je nutné, aby se v rovnici nevyskytovaly vektory křivosti, ale hodnoty křivosti. Celou rovnici vynásobíme výrazem $\frac{\mathbf{H}^k}{H^k}$ a levou stranu rovnice nahradíme H^{k+1} , potom

$$H^{k+1} = H^k \cdot \frac{(1 + \alpha^k) ((\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d)}{(1 + \alpha^k)^2 (\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d} . \quad (3.30)$$

Rovnici (3.30) upravíme tak, abychom mohli vyjádřit řešení kvadratické rovnice (α^k):

$$(\alpha^k)^2 (H^{k+1} \cdot P) + \alpha^k (2H^{k+1} \cdot P - H^k) + H^{k+1} - H^k = 0 , \quad (3.31)$$

kde výraz $P = \frac{(\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k}{(\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k + d}$.

Výběr řešení α^k

Vlastnosti a chování funkce velikosti křivosti H^k jsou shodné s vlastnostmi a chováním funkce H_i^k v případě řešení vyhlazování křivky. Odvození výběru α^k pro vyhlazování povrchu je tedy analogické k výběru α_i^k v algoritmu vyhlazování křivky. Vlastnosti funkce H_i^k a odvození výběru α_i^k lze nalézt v 2.7.3.

Rovnice (3.31) je kvadratická. Mohou tedy nastat tři případy řešení:

1. Rovnice má jen jedno řešení:
Použijeme α^k , které vyšlo.
2. Rovnice má dvě různá řešení:
Musíme rozhodnout, které řešení je pro algoritmus výhodnější. Výběr je analogický s výběrem v algoritmu vyhlazování pro křivky. Řešením rovnice (3.31) je to ze dvou řešení $(\alpha^k)_{1,2}$, které náleží do intervalu $((\alpha^k)_{\min}, (\alpha^k)_{\max})$, kde z výrazu (3.30) vyjádříme:

$$(\alpha^k)_{\min} = -1 - \sqrt{\frac{d}{(\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k}} , \quad (3.32)$$

$$(\alpha^k)_{\max} = -1 + \sqrt{\frac{d}{(\hat{\mathbf{a}}^k)^\top \mathbf{B}^{-1} \hat{\mathbf{a}}^k}} . \quad (3.33)$$

3. Rovnice nemá řešení:

Tento případ je ekvivalentní situaci, kdy hodnota $\frac{H^{k+1}}{H^k}$ nepatří do intervalu $((\frac{H^{k+1}}{H^k})_{\min}, (\frac{H^{k+1}}{H^k})_{\max})$, kde

$$\left(\frac{H^{k+1}}{H^k}\right)_{\min} = \frac{H^{k+1}}{H^k} \left((\alpha^k)_{\min}\right), \quad (3.34)$$

$$\left(\frac{H^{k+1}}{H^k}\right)_{\max} = \frac{H^{k+1}}{H^k} \left((\alpha^k)_{\max}\right). \quad (3.35)$$

Pro implementaci je nutné α^k zvolit. Hodnotu α^k zvolíme tak, abychom zvolili co možná nejbližší hodnotu funkce $\frac{H^{k+1}}{H^k}(\alpha^k)$ k hodnotě $\frac{H^{k+1}}{H^k}$. Nejbližší hodnota je vyjádřena extrémy funkce $(\frac{H^{k+1}}{H^k})_{\min}$ a $(\frac{H^{k+1}}{H^k})_{\max}$. Příslušný extrém zvolíme dle znaménka extrému a hodnoty $\frac{H^{k+1}}{H^k}$ (znaménka musejí být shodná). Podle zvoleného extrému vybereme odpovídající $(\alpha^k)_{\min}$ nebo $(\alpha^k)_{\max}$.

3.6.4 Speciální případy

Pokud velikost křivosti $H^k = 0$, pak dosazením do vzorce (3.30) dostaneme, že $H^{k+1} = 0$. Novou křivost ve vrcholu \mathbf{a}^k tedy zvolíme rovno nule. Vzhledem k definici je volba nové křivosti H^{k+1} správná, protože vektor určující směr, ve kterém má být nalezen nový vrchol \mathbf{a}^{k+1} , je nulový.

Pokud v průběhu výpočtu k -tého kroku algoritmu byla splněna některá z následujících podmínek, hodnotu α^k nepočítáme a dosadíme přímo hodnotu, která je uvedena.

1. Pokud $H^k = 0$, pak H^{k+1} je také rovno nule a pozice vrcholu \mathbf{a} se nemění, proto z (3.28) vyjde $\alpha^k = 0$.
2. Pokud $H^k \neq 0$ a $H^{k+1} = 0$, pak dosazením H^k a H^{k+1} do (3.30) vyjde $\alpha^k = -1$.

3.7 Algoritmus vyhlazování podle kritéria neuniformity

V této kapitole popíšeme podrobně jednotlivé kroky algoritmu vyhlazování podle kritéria neuniformity.

3.7.1 Výpočet vektoru neuniformity \mathbf{U}^k

Pro výpočet nového vrcholu povrchu je nutné znát vektor neuniformity.

Vrcholy $V = \{\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_m^k\}$ jsou aktuální vrcholy povrchu S , se kterými se pracuje v k -té iteraci algoritmu. Aktuálně zpracovávaný vrchol budeme

označovat \mathbf{a}^k . Bezprostřední sousedé vrcholu \mathbf{a}^k jsou $O = (\mathbf{a}_1^k, \mathbf{a}_2^k, \dots, \mathbf{a}_n^k)$. Pro bezprostřední sousedy také platí, že $\mathbf{a}_0^k = \mathbf{a}_n^k$ a $\mathbf{a}_{n+1}^k = \mathbf{a}_1^k$. Vektor neuniformity $\mathbf{U}^k = [u_x^k, u_y^k, u_z^k]$ ve vrcholu \mathbf{a}^k spočteme ze vzorce (3.17).

Vzorec pro výpočet vektoru neuniformity pro k -tou iteraci algoritmu je

$$\mathbf{U}^k = \frac{1}{n} \sum_{i=1}^n \frac{-\mathbf{P}_{\mathbf{d}_i^k}((\mathbf{a}^k - \mathbf{a}_{i-1}^k) + (\mathbf{a}^k - \mathbf{a}_{i+1}^k))}{2}, \quad (3.36)$$

kde $\mathbf{P}_{\mathbf{d}_i^k} = \mathbf{P}_{\mathbf{d}_i}$ je zadefinováno výrazem (3.14).

3.7.2 Výpočet nového vrcholu \mathbf{a}^{k+1}

Definujeme výpočet nového vrcholu \mathbf{a}^{k+1} povrchu S :

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \beta \mathbf{U}^k, \quad (3.37)$$

kde \mathbf{a}^k je aktuální vrchol povrchu a hodnota β je odvozena dále.

Z definice neuniformity (3.17) odvodíme vzorec pro výpočet neuniformity ve vrcholu \mathbf{a}^{k+1} v $(k+1)$ -ní iteraci algoritmu:

$$\mathbf{U}^{k+1} = \frac{1}{n} \sum_{i=1}^n \frac{-\mathbf{P}_{\mathbf{d}_i^{k+1}}((\mathbf{a}^{k+1} - \mathbf{a}_{i-1}^{k+1}) + (\mathbf{a}^{k+1} - \mathbf{a}_{i+1}^{k+1}))}{2}, \quad (3.38)$$

kde $\mathbf{P}_{\mathbf{d}_i^{k+1}}$ je projekce zadefinovaná (3.14) a platí, že

$$\begin{aligned} \mathbf{d}_i^{k+1} &= \mathbf{d}_i^k, \\ \mathbf{a}_{i-1}^{k+1} &= \mathbf{a}_{i-1}^k, \\ \mathbf{a}_{i+1}^{k+1} &= \mathbf{a}_{i+1}^k, \end{aligned}$$

pro všechna i , kde $i = 1 \dots n$. Potom $\mathbf{P}_{\mathbf{d}_i^{k+1}} = \mathbf{P}_{\mathbf{d}_i^k}$.

Vektor nové neuniformity \mathbf{U}^{k+1} ve vrcholu \mathbf{a}^{k+1} odpovídá vektoru \mathbf{U}^{k+1} zadefinovaném výrazem (3.38), pokud pohybujeme jen s vrcholem \mathbf{a}^k a všechny ostatní vrcholy povrchu S stojí.

Odvodíme velikost β . Dosazením (3.37) do vzorce (3.38) dostaneme

$$\begin{aligned} \mathbf{U}^{k+1} &= \frac{1}{n} \sum_{i=1}^n \frac{-\mathbf{P}_{\mathbf{d}_i^k}((\mathbf{a}^k + \beta \mathbf{U}^k - \mathbf{a}_{i-1}^k) + (\mathbf{a}^k + \beta \mathbf{U}^k - \mathbf{a}_{i+1}^k))}{2} = \\ &= \mathbf{U}^k - \frac{\beta}{n} \sum_{i=1}^n \mathbf{P}_{\mathbf{d}_i^k} \mathbf{U}^k = (\mathbf{E} - \frac{\beta}{n} \mathbf{D}) \mathbf{U}^k, \end{aligned} \quad (3.39)$$

kde $\mathbf{D} = \sum_{i=1}^n \mathbf{P}_{\mathbf{d}_i^k}$. Vzhledem k tomu, že velikost β není snadné spočítat a navíc je β závislá na datech, rozhodli jsme se pro zjednodušení výpočtu volit $\beta = \lambda_u$, kde λ_u je použita v rovnici difuze (3.18) a $0 < \lambda_u < 1$. Experimenty potvrzují, že takto zvolená β je vhodná.

3.8 Experimenty

V této kapitole ukážeme výsledky implementace algoritmu pro vyhlazování uzavřených povrchů (dále jen povrchů). Uvedeme experimenty, které prokazují robustnost a stabilitu algoritmu. Nakonec uvedeme experimenty, které ukazují, jak volba vstupních parametrů ovlivňuje výsledek algoritmu.

V celé kapitole budeme používat jednotné značení pro vstupní parametry algoritmu. Parametr λ určuje velikost difusního kroku pro kritérium křivosti (viz. (3.7)). Parametr λ_u určuje velikost difusního kroku pro kritérium neuniformity (viz. (3.18)). Následující parametry jsou zcela shodné s parametry algoritmu pro vyhlazování křivek, proto je možné se u parametrů odkazovat do kapitoly o křivkách. Parametr I určuje maximální počet iterací algoritmu (kapitola 2.10), S_h a S_u určují minimální přípustný průměr posunu všech vrcholů pro kritérium křivosti a neuniformity (kapitola 2.10). Abychom předešli rozdílnému nastavení těchto parametrů pro stejný povrch v různých měřítkách, v implementaci jsou tyto parametry vynásobeny průměrnou délkou hrany. Parametry T_h a T_u určují, po kolika iteracích se mění kritérium vyhlazování (kapitola 2.10).

U popisu každého experimentu budou popsány parametry, které jsou zvoleny jinak, než je jejich předdefinovaná hodnota. Parametry λ , λ_u a I jsou uváděny vždy. Pokud parametr není v popisu experimentu uveden, jeho hodnota je:

- $S_h = 0.00001$,
- $S_u = 0.00001$,
- $T_h = 50$,
- $T_u = 50$.

3.8.1 Základní funkce

V této kapitole jsou uvedeny experimenty, které ukazují základní funkci algoritmu. Každý experiment je proveden tak, že vstupní povrch je zašuměn. Takto zašuměný povrch je vstupem algoritmu pro vyhlazování. Ve výsledcích experimentů jsou uváděna dvě vyhlazení, která jsou spouštěna se shodnými parametry, kromě parametru I . První vyhlazení je provedeno jen na 100 iterací a druhé vyhlazení je provedeno na 5000 iterací.

Ke každému experimentu je uveden výsledek tak, že obrázek (a) zobrazuje zašuměný povrch, který je vstupem pro algoritmus vyhlazování, obrázek (b) zobrazuje první vyhlazení a obrázek (c) zobrazuje druhé vyhlazení povrchu. Dále jsou také ke každému experimentu zobrazeny grafy:

- V obrázku (d) jsou uvedeny grafy funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{H_{\text{max}}}{H_0}$ a $\frac{H_{\text{min}}}{H_0}$ v závislosti na k -té iteraci algoritmu. Výraz H_0 označuje průměrnou velikost

křivosti před začátkem vyhlazování. Výraz H_{avg} označuje průměrnou velikost křivosti v k -té iteraci algoritmu, H_{max} a H_{min} jsou maximální a minimální velikosti křivosti v k -té iteraci.

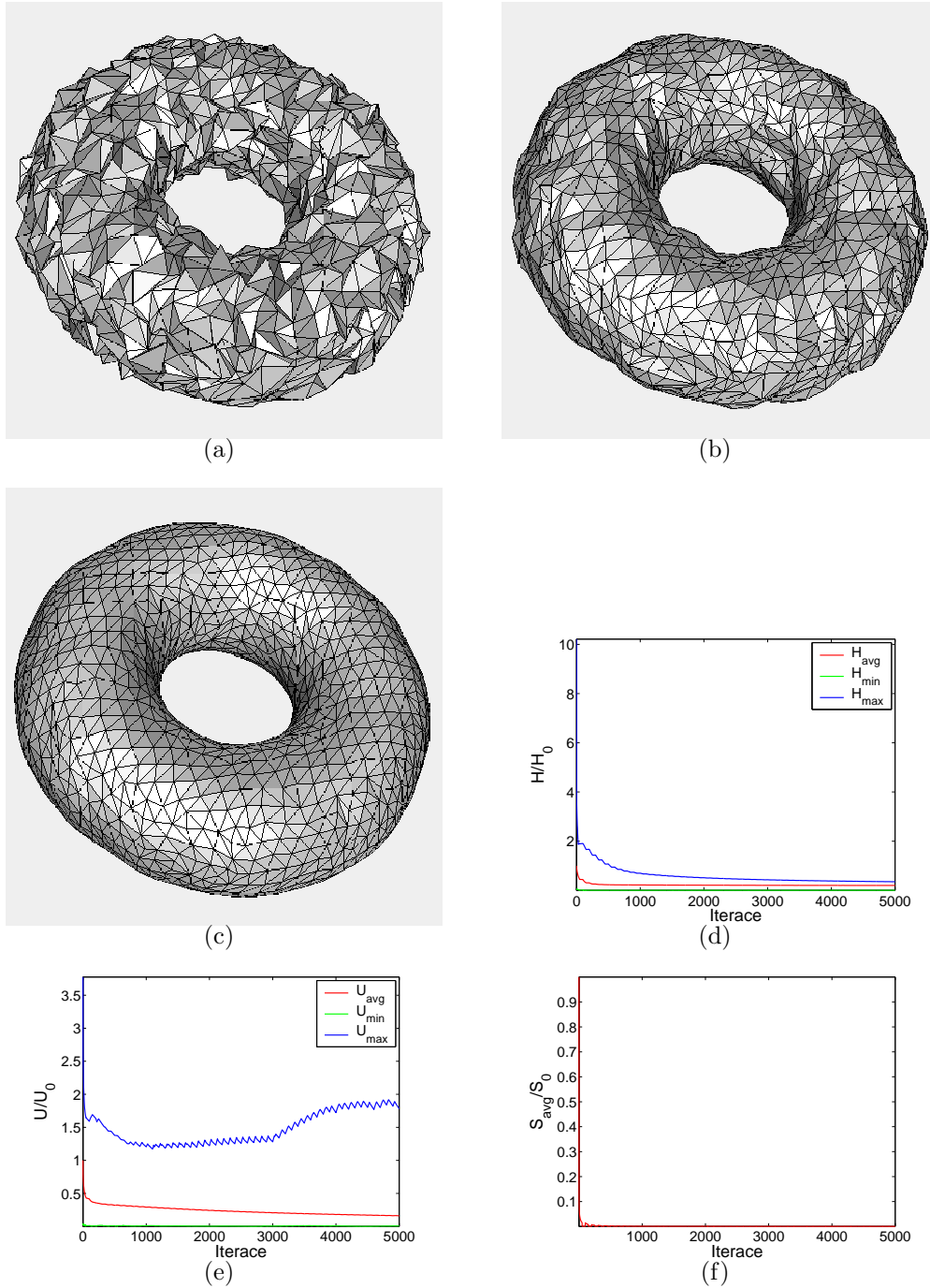
- V obrázku (e) jsou uvedeny grafy funkcí $\frac{U_{\text{avg}}}{U_0}$, $\frac{U_{\text{max}}}{U_0}$ a $\frac{U_{\text{min}}}{U_0}$ v závislosti na k -té iteraci algoritmu. Výraz U_0 označuje průměrnou velikost neuniformity před začátkem vyhlazování. Výraz U_{avg} označuje průměrnou velikost neuniformity v k -té iteraci algoritmu, U_{max} a U_{min} jsou maximální a minimální velikost neuniformity v k -té iteraci.
- V obrázku (f) je znázorněn graf funkce $\frac{S_{\text{avg}}}{S_0}$ v závislosti na iteraci algoritmu. Výraz S_0 označuje průměrný posun všech vrcholů povrchu v první iteraci algoritmu a S_{avg} je průměrný posun v k -té iteraci algoritmu.

Velikost rozdílu mezi starou a novou křivostí je omezena parametrem λ (případně λ_u), a tím je i omezen pohyb vrcholu. Rozdíly mezi průměrným, maximálním a minimálním pohybem vrcholu nejsou v daném měřítku grafu téměř zřetelné, proto nejsou grafy funkcí $\frac{S_{\text{max}}}{S_0}$ a $\frac{S_{\text{min}}}{S_0}$ uvedeny.

První experimenty byly provedeny na pravidelných mnohostěnech. Z definice výpočtu hodnoty nové křivosti ve vrcholu povrchu (3.7) plyne, že v každém vrcholu pravidelného mnohostěnu je nová křivost rovna staré křivosti a vrcholy mnohostěnu se tedy nebudou pohybovat. Algoritmus vyhlazení by tedy na pravidelné mnohostěny neměl mít vliv a výstupní povrch by měl být shodný s povrchem vstupním. Experimenty potvrdili, že pravidelné mnohostěny jsou algoritmem nezměněny a algoritmus je ukončen již po první iteraci výpočtu (vyhlazováno jak podle kritéria křivosti, tak i podle kritéria neuniformity).

V dalším experimentu je vstupem zašuměný torus. Parametry jsou $I_1 = 100$, $I_2 = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.001$. Průběh vyhlazování a příslušné grafy jsou na obrázku 3.5. Na obrázku 3.5(e) lze pozorovat na grafu funkcí $\frac{U_{\text{avg}}}{U_0}$ a $\frac{U_{\text{max}}}{U_0}$ oscilaci: Na intervalech, kde je vyhlazováno podle kritéria křivosti, velikost funkcí roste. Naopak na intervalech, kde je vyhlazováno podle kritéria neuniformity, velikost neuniformity klesá. Oscilace je způsobena neortogonalitou kritérií. Funkce $\frac{U_{\text{avg}}}{U_0}$ konverguje k nule. Funkce $\frac{U_{\text{max}}}{U_0}$ nekonverguje, dokonce se hodnota funkce po 3000 iteracích zvětšuje. Myslíme si, že je to způsobeno špatnou triangulací povrchu a také tím, že algoritmus je ukončen na počet iterací (nedokonvergoval).

V dalším experimentu je vstupem zašuměná vázička. Parametry jsou $I_1 = 100$, $I_2 = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.01$. Průběh vyhlazování a příslušné grafy jsou na obrázku 3.6. Na obrázku 3.6(c) je vyhlazená vázička. Tato vázička však již nemá vršek. To je způsobeno triangulací této oblasti, ve které je obsažen jeden vrchol, jehož stupeň je petinásobně vyšší, než stupeň ostatních



Obrázek 3.5: Vyhlazování zašuměného torusu. Parametry $\lambda = 0.01$, $\lambda_u = 0.001$. (a) Vstup algoritmu. (b) Vyhlazování po 100 iteracích. (c) Vyhlazování po 5000 iteracích. (d) Funkce křivosti. (e) Funkce neuniformity. (f) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).

vrcholů. Tento vrchol také způsobuje postupný zánik celého vršku. Tento jev je pozorovatelný také na hodnotách funkcí $\frac{H_{\max}}{H_0}$ (obrázek 3.6(d)) a $\frac{U_{\max}}{U_0}$ (obrázek 3.6(e)) na intervalu 1000 až 2500 iterací. Na tomto intervalu vrcholy, které sousedí s vrcholem s vysokým stupněm, postupně procházejí maximem velikosti křivosti (plyne z vlastností funkce velikosti křivosti). Postupným ztahováním vrcholů se také v některých vrchole poruší neuniformita. Funkce $\frac{H_{\text{avg}}}{H_0}$ (obrázek 3.6(d)) však zůstává konstantní a funkce $\frac{U_{\text{avg}}}{U_0}$ (obrázek 3.6(e)) klesá. Po stažení vršku vázičky začne mírně narůstat hodnota funkce $\frac{H_{\max}}{H_0}$, což je způsobeno postupným vychylováním vrcholu, který má vysoký stupeň, z roviny, čímž roste jeho křivost.

V dalším experimentu je vstupem zašuměná koule s různými stupni vrcholů. Parametry jsou $I_1 = 100$, $I_2 = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.01$. Průběh vyhlazování a příslušné grafy jsou na obrázku 3.7. V grafu funkce $\frac{H_{\max}}{H_0}$ (obrázek 3.7(d)) se po několika iteracích objevuje náhle velmi vysoká hodnota, která je v další iteraci opět zmenšena. To je způsobeno posunutím vrcholů povrchu tak, že vznikne vrchol obsažený v trojúhelnících, jejichž plocha je téměř stokrát menší než plocha ostatních trojúhelníků povrchu. V tomto vrcholu velmi vzroste křivost, která je v následující iteraci zmenšena. Toto chování ovlivňuje i průběh ostatních funkcí, které jsou zobrazeny.

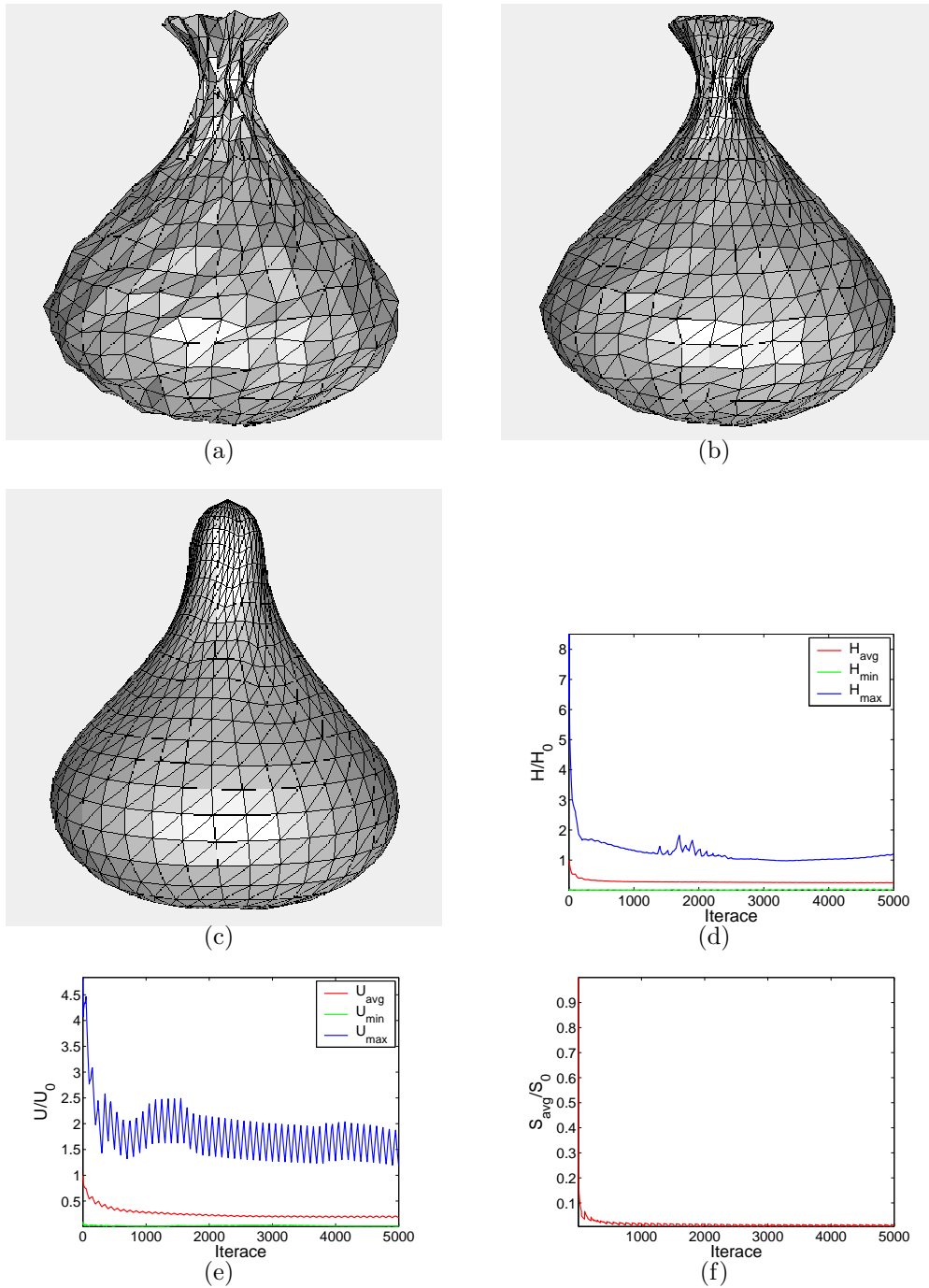
Algoritmus vyhlazení povrchu je také možné využít pro natažení hladké sítě mezi několika objekty. Provedli jsme experiment na objektu, který je sestaven ze tří válců, které jsou na sebe nepěkne napojeny a napojení obsahuje i miniaturní trojúhelníky. Toto napojení válců je nutné vyhladit. Vyhlazování je provedeno tak, že je předem určeno, se kterými vrcholy lze v průběhu vyhlazování pohybovat, a které musejí zůstat nehybné. Algoritmu je tato informace předána pomocí parametru, ve kterém se určují speciální lambdy pro jednotlivé vrcholy povrchu. Pro vrchol, který má zůstat nehybný je lambda zvolena jako nula. Pro vrchol, se kterým je možné pohybovat, je lambda zvolena jako jedna. Výsledek vyhlazení je na obrázku 3.8. Z obrázku je patrné, že napojení je vyhlazeno a síť trojúhelníků je pravidelná. Parametry algoritmu jsou $I = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.001$.

3.8.2 Robustnost a stabilita

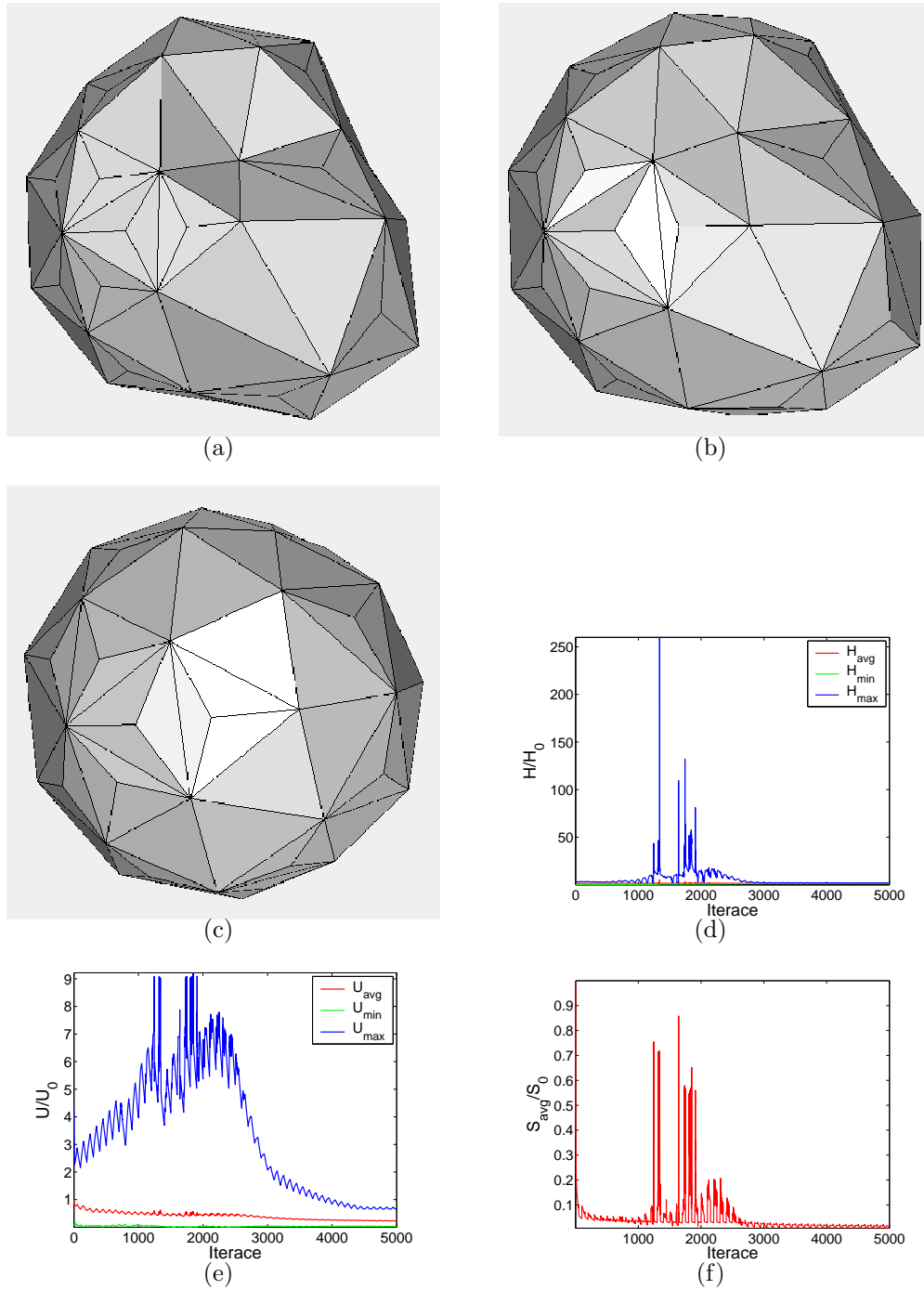
V této kapitole jsou uvedeny experimenty, které prokazují robustnost a stabilitu algoritmu. Robustnost je ukázána na vyhlazení extrémně zašuměných povrchů. Stabilita je prokázána tak, že k již vyhlazenému povrchu je přidán šum a takto pozměněný povrch je algoritmem znovu vyhlazen.

Experimenty jsou zobrazovány stejně jako experimenty v kapitole 3.8.1.

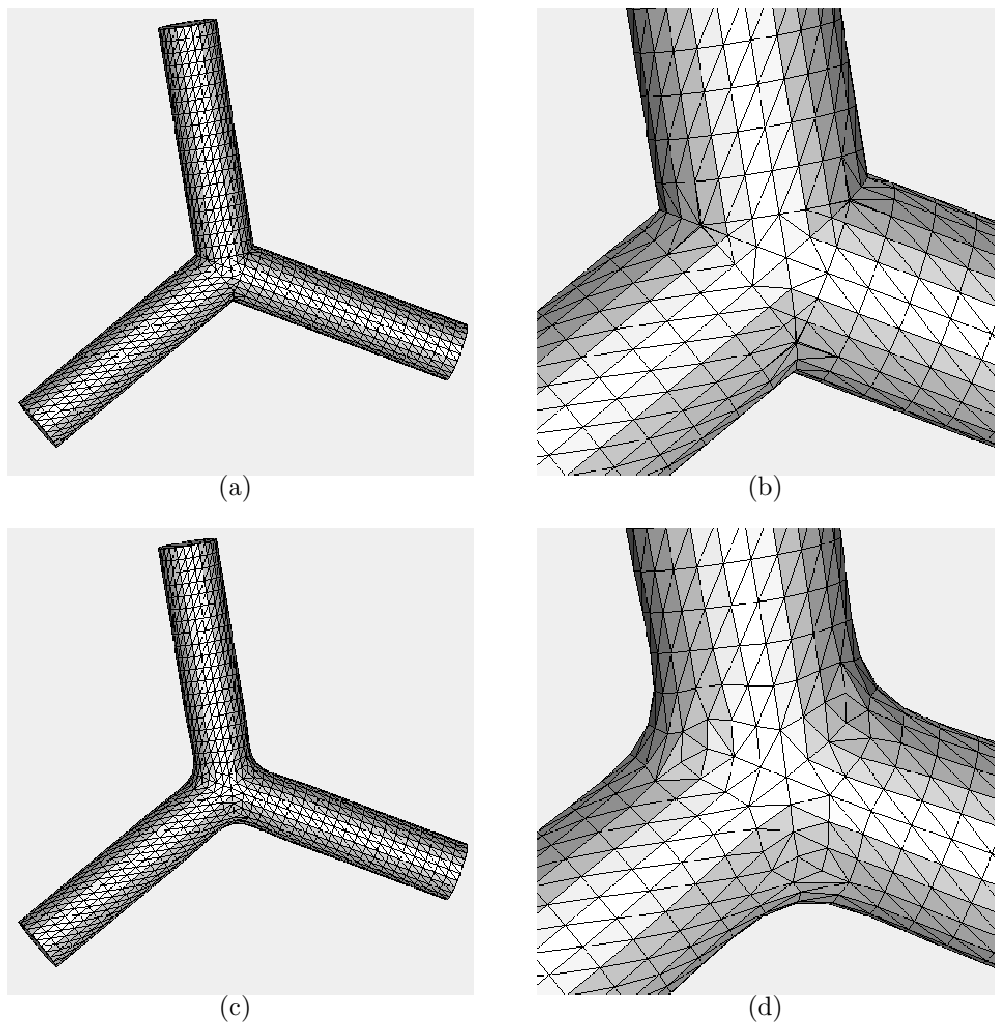
Nejprve jsme testovali robustnost na extrémně zašuměném torusu. Parametry jsou $I_1 = 500$, $I_2 = 10000$, $\lambda = 0.01$ a $\lambda_u = 0.01$. Průběh vyhlazování a příslušné grafy jsou na obrázku 3.9. Na obrázku 3.9(a) je vidět, že přidáný šum je tak velký, že se některé trojúhelníky povrchu navzájem



Obrázek 3.6: Vyhlazování zašuměné vázičky. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. (a) Vstup algoritmu. (b) Vyhlazování po 100 iteracích. (c) Vyhlazování po 5000 iteracích. (d) Funkce křivosti. (e) Funkce neuniformity. (f) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).



Obrázek 3.7: Vyhlazování zašuměné koule s různými stupni vrcholů. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. (a) Vstup algoritmu. (b) Vyhlazení po 100 iteracích. (c) Vyhlazení po 5000 iteracích. (d) Funkce křivosti. (e) Funkce neuniformity. (f) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).



Obrázek 3.8: Natažení hladké sítě mezi třemi válci. Parametry $I = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.001$. (a) Vstup algoritmu. (b) Zvětšení spoje vstupu. (c) Vyhlazení po 5000 iteracích. (d) Zvětšení spoje vyhlazeného povrchu.

protýkají. V grafu funkce $\frac{H_{\max}}{H_0}$ (obrázek 3.9(d)) se objevuje velmi vysoká hodnota pro maximum funkce. Chování funkce na okolí tohoto maxima je způsobeno vlastností funkce velikosti křivosti. Povrch obsahuje vrcholy, které jsou velmi vytaženy a vyhlazováním jsou ztahovány k ostatním vrcholům. V průběhu stahování vrcholů nabývají hodnoty funkcí velikostí křivosti maxim, což se projevuje ve funkci $\frac{H_{\max}}{H_0}$ jako postupný nárůst a posléze postupný pokles hodnoty funkce. Nicméně se vzrůstajícím počtem iterací všechna kritéria klesají a výsledný objekt je hladký.

V druhém experimentu je vstupem extrémně zašuměná vázička. Parametry jsou $I_1 = 500$, $I_2 = 10000$, $\lambda = 0.01$ a $\lambda_u = 0.01$. Průběh vyhlazování a příslušné grafy jsou na obrázku 3.10. Chování funkcí je shodné s chováním funkcí experimentu vyhlazování vázičky (kapitola 3.8.1). I přes extrémní počáteční šum je výsledný objekt hladký a svým tvarem odpovídá výsledku z předešlého experimentu (viz. obrázek 3.6).

Ke každému experimentu stability je uveden výsledek tak, že obrázek (a) zobrazuje zašuměný povrch, obrázek (b) zobrazuje vyhlazený povrch, obrázek (c) zobrazuje zašumění již vyhlazeného povrchu a obrázek (d) znovu vyhlazený povrch. Zobrazeny jsou také grafy, jejichž popis je shodný s popisem grafů v části 3.8.1 (posunuty o jednu pozici v obrázku). Grafy zachycují celý průběh vyhlazování (včetně vyhlazení po přidání šumu do již vyhlazeného povrchu).

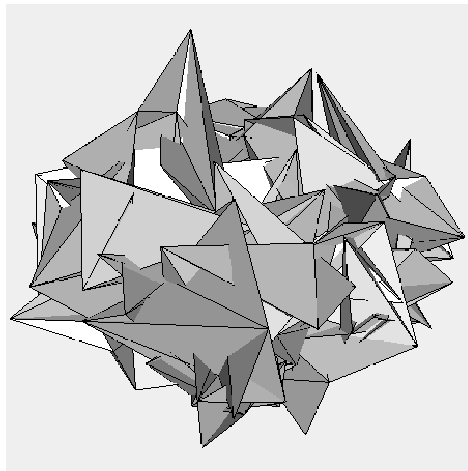
Nejprve je vstupem zašuměná krychle. Vyhlazení s parametry $I = 5000$, $\lambda = 0.01$ a $\lambda_u = 0.01$. K vyhlazenému povrchu byl přidán šum, který je větší než šum původní, a povrch byl znovu vyhlazen se shodnými parametry jako u prvního vyhlazení. Průběhy vyhlazování a příslušné grafy jsou na obrázku 3.11. Na grafech funkcí lze pozorovat, že po přidání šumu k již vyhlazenému povrchu se velmi změnila hodnota všech funkcí. V průběhu dalších iterací algoritmu se však všechny hodnoty vracejí k hodnotám funkcí před zašuměním. To znamená, že povrch s přidáním šumu, i když větším než byl šum původní, je opět vyhlazen do stejného tvaru jako před přidáním šumu.

Jako druhý objekt byl zvolen zašuměný torus. Vyhlazení s parametry $I = 10000$, $\lambda = 0.01$ a $\lambda_u = 0.001$. K vyhlazenému povrchu byl přidán šum a povrch znovu vyhlazen se shodnými parametry jako u prvního vyhlazení. Průběhy vyhlazování a příslušné grafy jsou na obrázku 3.12. Opět jsme docílili stejných výsledků jako před přidáním šumu a to jak kvantitativně (hodnoty kritérií), tak kvalitativně (tvar objektu).

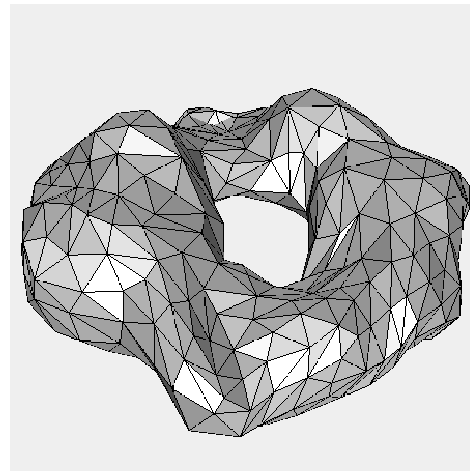
3.8.3 Výběr lambdy pro křivost a neuniformitu

V této kapitole ukážeme jak volba λ a λ_u ovlivňuje výsledek algoritmu.

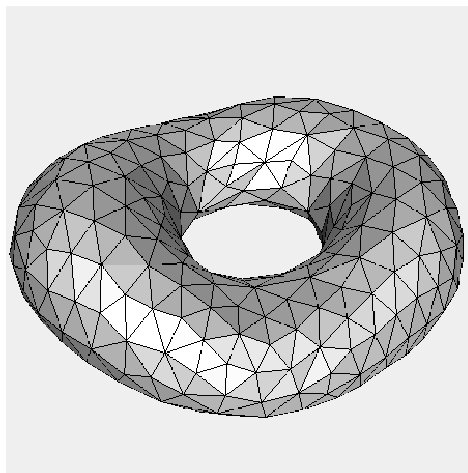
Experimenty jsou provedeny tak, že měníme hodnotu příslušné lambdy a ostatní parametry jsou nastaveny na optimální. Výsledky jsou prezentovány pomocí tabulek, které zachycují hodnoty funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro testované



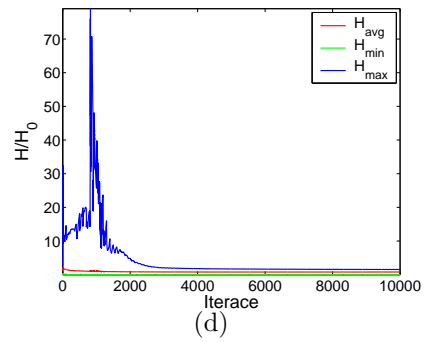
(a)



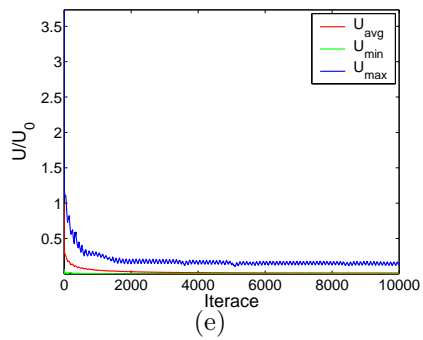
(b)



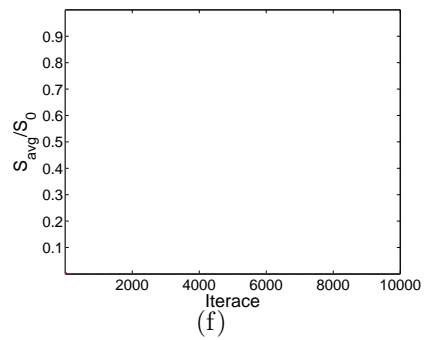
(c)



(d)

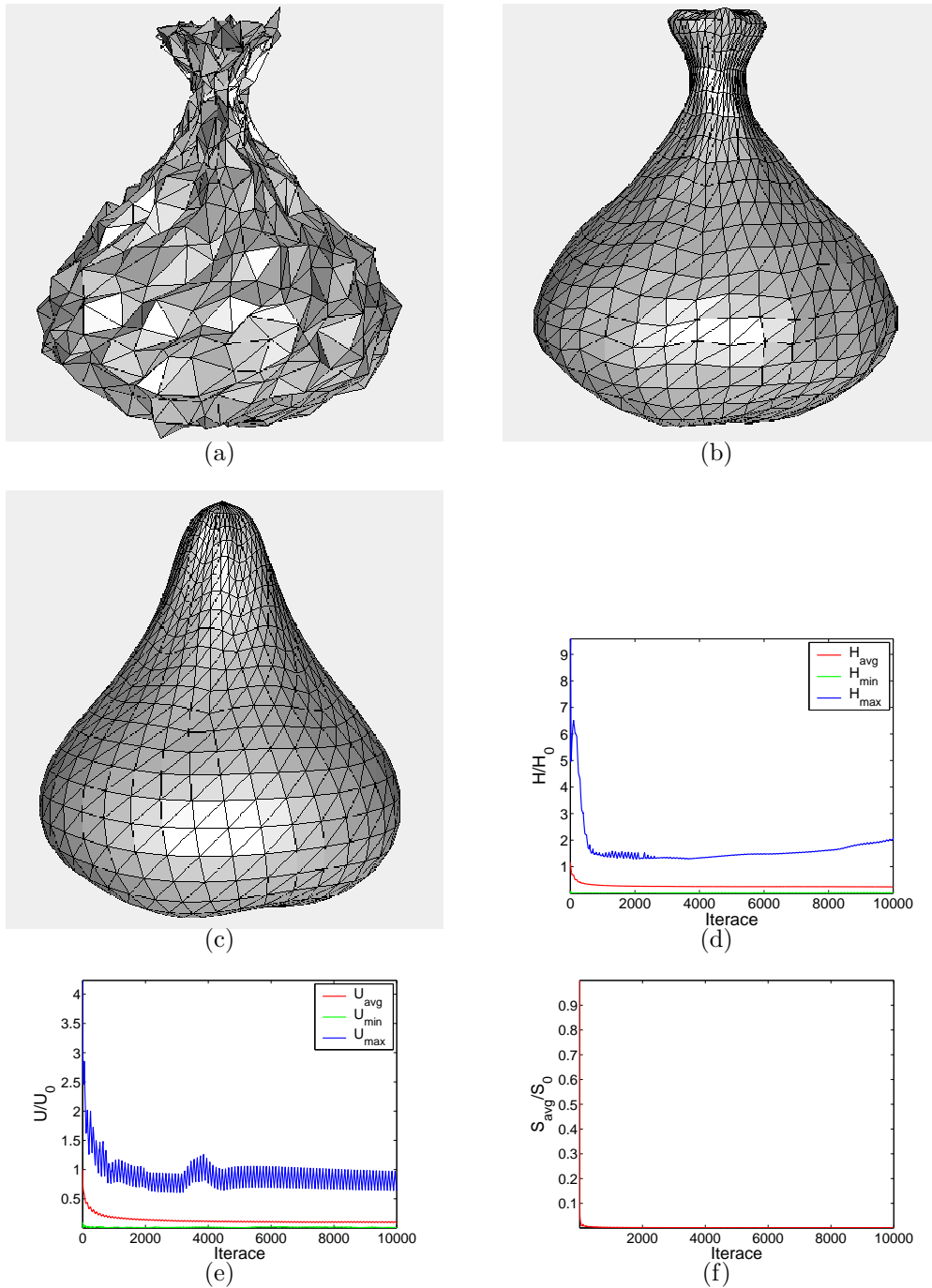


(e)

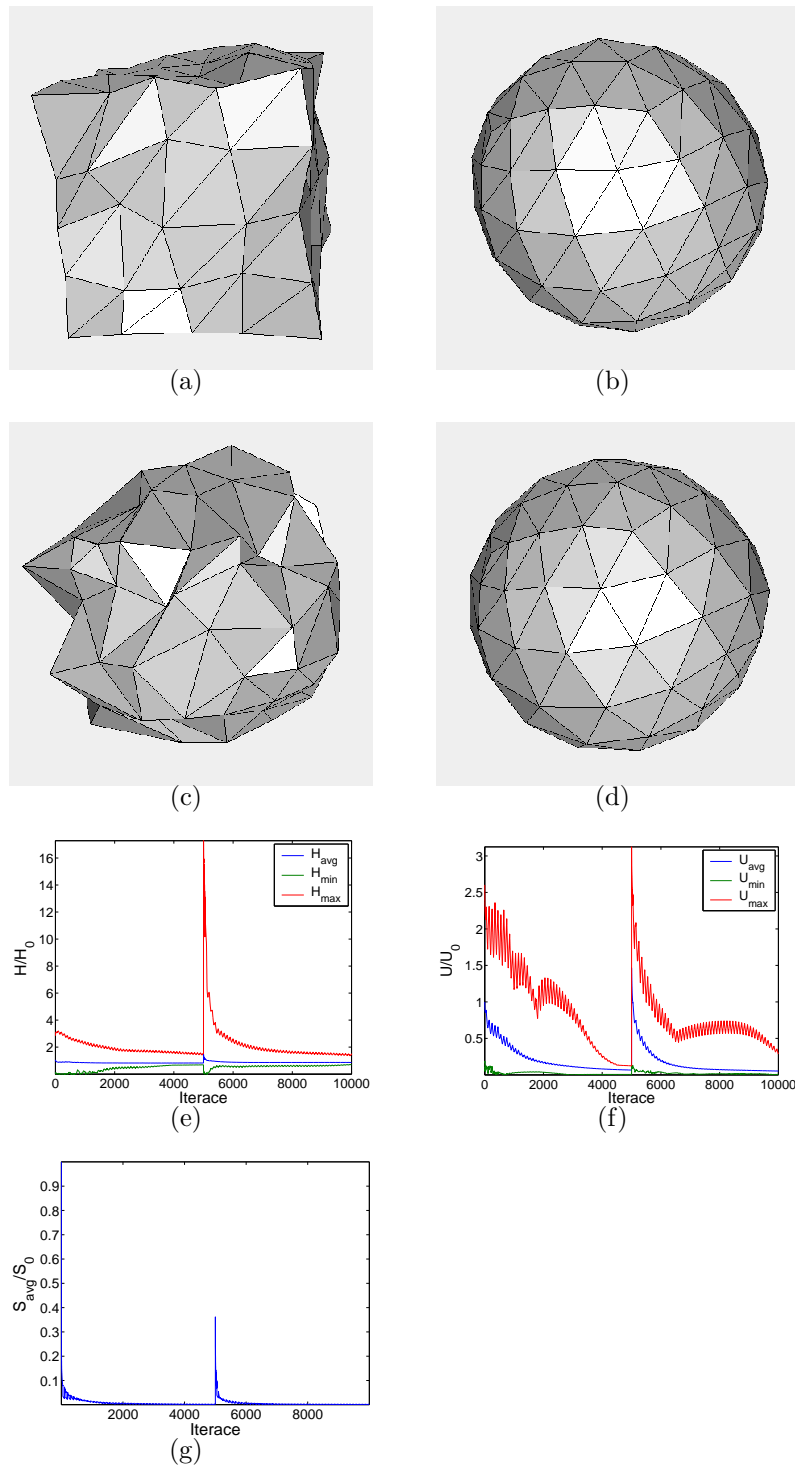


(f)

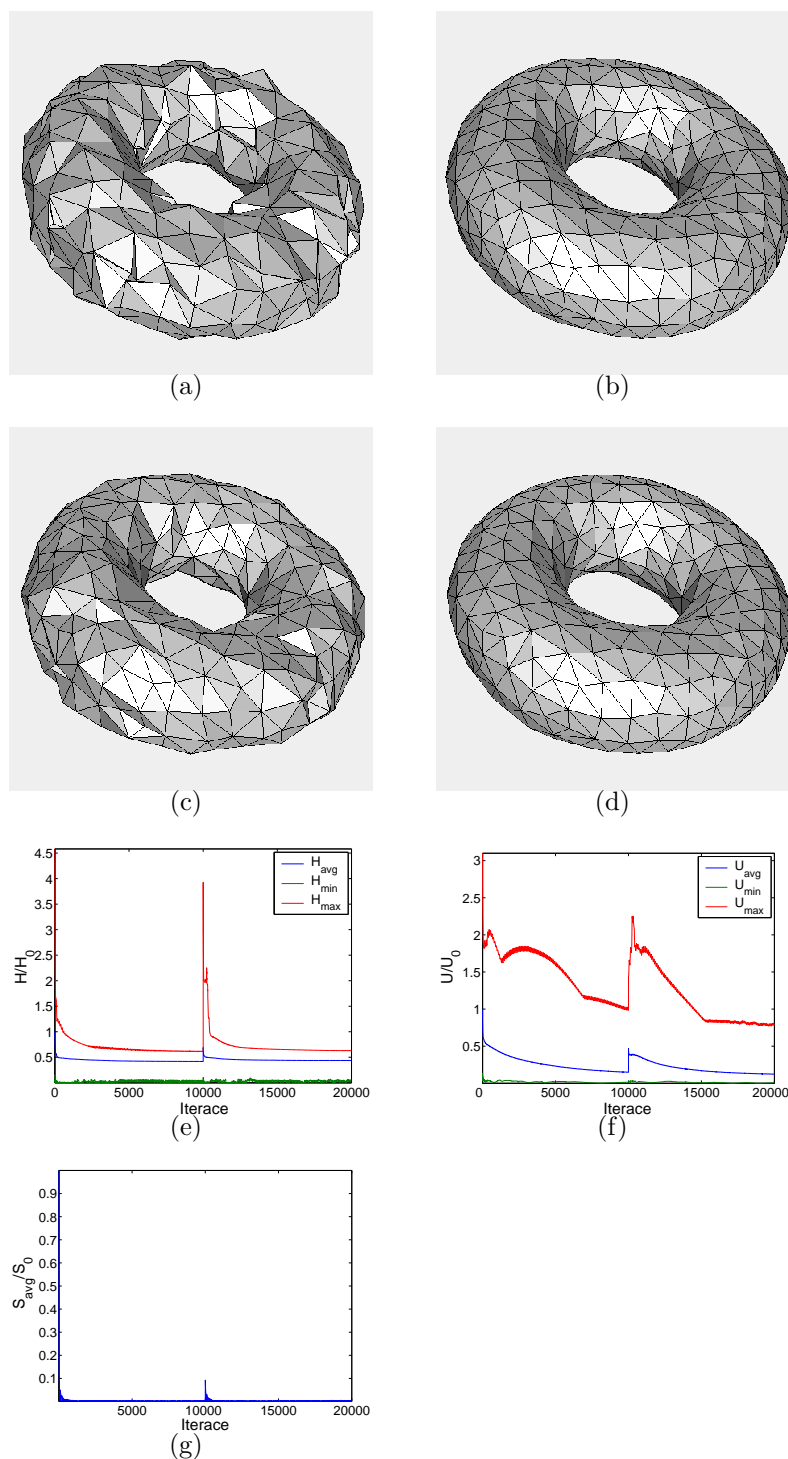
Obrázek 3.9: Vyhlazování extrémně zašuměného torusu. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. (a) Vstup algoritmu. (b) Vyhlazování po 500 iteracích. (c) Vyhlazování po 10000 iteracích. (d) Funkce křivosti. (e) Funkce neuniformity. (f) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).



Obrázek 3.10: Vyhlazování extrémně zašuměné vázičky. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. (a) Vstup algoritmu. (b) Vyhlazování po 500 iteracích. (c) Vyhlazování po 10000 iteracích. (d) Funkce křivosti. (e) Funkce neuniformity. (f) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).



Obrázek 3.11: Vyhlazování zašuměné krychle. Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. (a) Vstup algoritmu. (b) Vyhlazení po 5000 iteracích. (c) Přidán šum. (d) Vyhlazení zašuměného povrchu po 5000 iteracích. (e) Funkce křivosti. (f) Funkce neuniformity. (g) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).



Obrázek 3.12: Vyhlazování zašuměného torusu. Parametry $\lambda = 0.01$ a $\lambda_u = 0.001$. (a) Vstup algoritmu. (b) Vyhlazení po 10000 iteracích. (c) Přidání šum. (d) Vyhlazení zašuměného povrchu po 10000 iteracích. (e) Funkce křivosti. (f) Funkce neuniformity. (g) Funkce průměrného posunu vrcholů. Funkce jsou definovány v kapitole (3.8.1).

Hodnota H_{avg}/H_0 po iteraci						
λ	2000	3000	5000	7000	8000	10000
0.1000	0.428	0.431	0.445	0.459	0.467	0.487
0.0100	0.452	0.441	0.426	0.421	0.420	0.418
0.0010	0.464	0.450	0.432	0.423	0.420	0.417

Hodnota U_{avg}/U_0 po iteraci						
λ	2000	3000	5000	7000	8000	10000
0.1000	0.306	0.259	0.209	0.188	0.178	0.167
0.0100	0.353	0.298	0.230	0.185	0.169	0.148
0.0010	0.346	0.287	0.220	0.181	0.166	0.144

Hodnota S_{avg}/S_0 po iteraci						
λ	2000	3000	5000	7000	8000	10000
0.1000	0.001	0.001	0.000	0.000	0.000	0.000
0.0100	0.002	0.002	0.001	0.001	0.001	0.001
0.0010	0.003	0.003	0.002	0.002	0.002	0.001

Tabulka 3.1: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různé nastavení parametru λ vyhlazovaného torusu (viz. obrázek 3.13). Parametr $\lambda_u = 0.001$.

lambda. Popis funkcí lze nalézt v 3.8.1.

Experimenty se změnou λ :

Vstupem je zašuměný torus. Parametr $\lambda_u = 0.001$. Výsledek vyhlazování pro nastavení parametru λ je v tabulce 3.1 a na obrázku 3.13. Pro nastavení parametru $\lambda = 0.1$ je torus vyhlazen, ale jeho tvar je pozměněn a navíc dochází se zvyšujícím se počtem iterací k nárůstu průměrné křivosti. Chování pro nastavení $\lambda = 0.01$ a $\lambda = 0.001$ je téměř totožné, torus je vyhlazen bez pozměnění tvaru. Proto lze pro vyhlazení volit jedno z těchto dvou nastavení.

Experimenty se změnou λ_u :

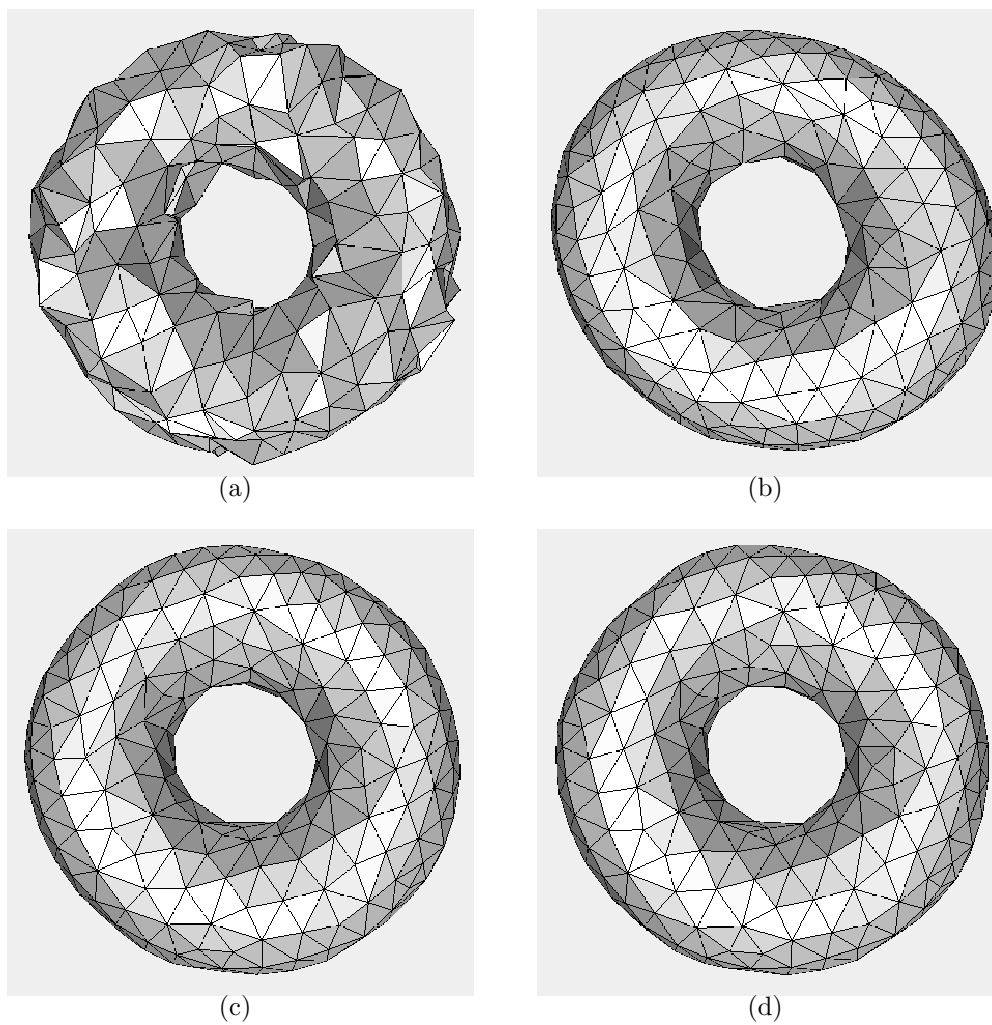
Vstupem je zašuměný torus, odlišný od torusu v předchozím experimentu. Parametr $\lambda = 0.01$. Výsledek vyhlazování pro nastavení parametru λ_u je v tabulce 3.2 a na obrázku 3.14. Pro nastavení $\lambda_u = 0.1$ není dosaženo vyhlazení. Pro nastavení $\lambda_u = 0.01$ je torus vyhlazen a ze všech nastavení je dosaženo pro maximální iteraci nejmenší křivosti a neuniformity. Pro nastavení $\lambda_u = 0.001$ je dosaženo vyhlazení, ale hodnota neuniformity je větší než pro nastavení $\lambda_u = 0.01$. Proto je pro vyhlazení nejvhodnější volit $\lambda_u = 0.01$.

3.8.4 Volba iteračního schématu

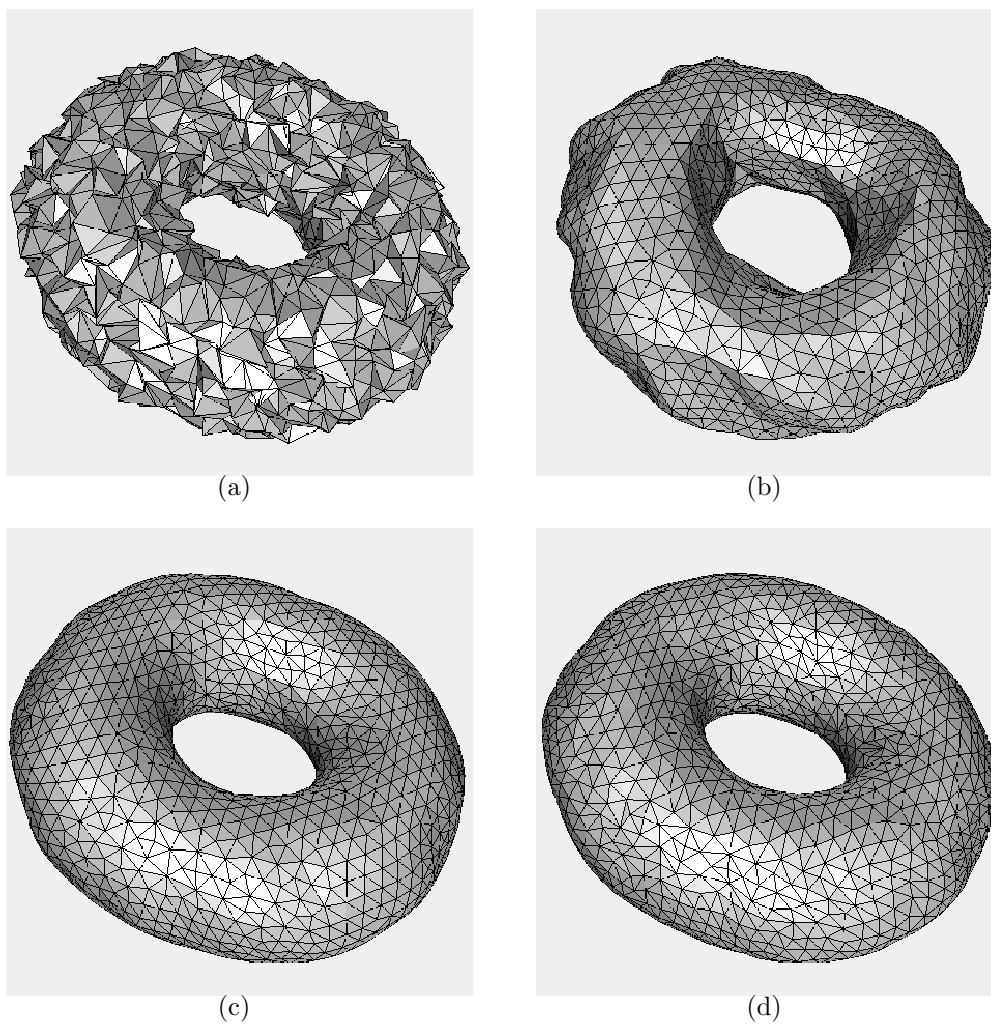
V této kapitole ukážeme, jak volba schématu ovlivňuje výsledek algoritmu.

Experiment je proveden tak, že měníme parametry T_h a T_u . Ostatní parametry jsou nastaveny na optimální. Výsledky jsou prezentovány stejně jako v předchozím experimentu (kapitola 3.8.3).

Vstupem algoritmu je zašuměná vázička. Parametry algoritmu jsou $\lambda = 0.01$ a $\lambda_u = 0.01$. Výsledek nastavení parametrů T_h a T_u je v tabulce 3.3. Obrázek 3.15 zobrazuje výsledky vyhlazení povrchu s různě nastavenými schémata. Pokud je vázička vyhlazována jen kritériem křivosti ($T_h = 100$



Obrázek 3.13: Vyhlazování torusu pro různé nastavení parametru λ . Parametr $\lambda_u = 0.001$. Data k průběhu vyhlazování jsou v tabulce 3.1. (a) Vstup algoritmu. (b) Parametr $\lambda = 0.1$. (c) Parametr $\lambda = 0.01$. (d) Parametr $\lambda = 0.001$.



Obrázek 3.14: Vyhlazování torusu pro různé nastavení parametru λ_u . Parametr $\lambda = 0.01$. Data k průběhu vyhlazování jsou v tabulce 3.2. (a) Vstup algoritmu. (b) Parametr $\lambda_u = 0.1$. (c) Parametr $\lambda_u = 0.01$. (d) Parametr $\lambda_u = 0.001$.

Hodnota H_{avg}/H_0 po iteraci						
λ_u	800	1700	2500	3300	4200	5000
0.1000	0.283	0.275	0.271	0.270	0.271	0.274
0.0100	0.205	0.193	0.190	0.189	0.188	0.188
0.0010	0.221	0.212	0.208	0.204	0.201	0.199

Hodnota U_{avg}/U_0 po iteraci						
λ_u	800	1700	2500	3300	4200	5000
0.1000	0.029	0.024	0.022	0.021	0.021	0.020
0.0100	0.104	0.067	0.058	0.053	0.051	0.049
0.0010	0.302	0.254	0.222	0.197	0.176	0.160

Hodnota S_{avg}/S_0 po iteraci						
λ_u	800	1700	2500	3300	4200	5000
0.1000	0.006	0.005	0.005	0.005	0.004	0.004
0.0100	0.002	0.001	0.001	0.001	0.001	0.001
0.0010	0.001	0.001	0.000	0.000	0.000	0.000

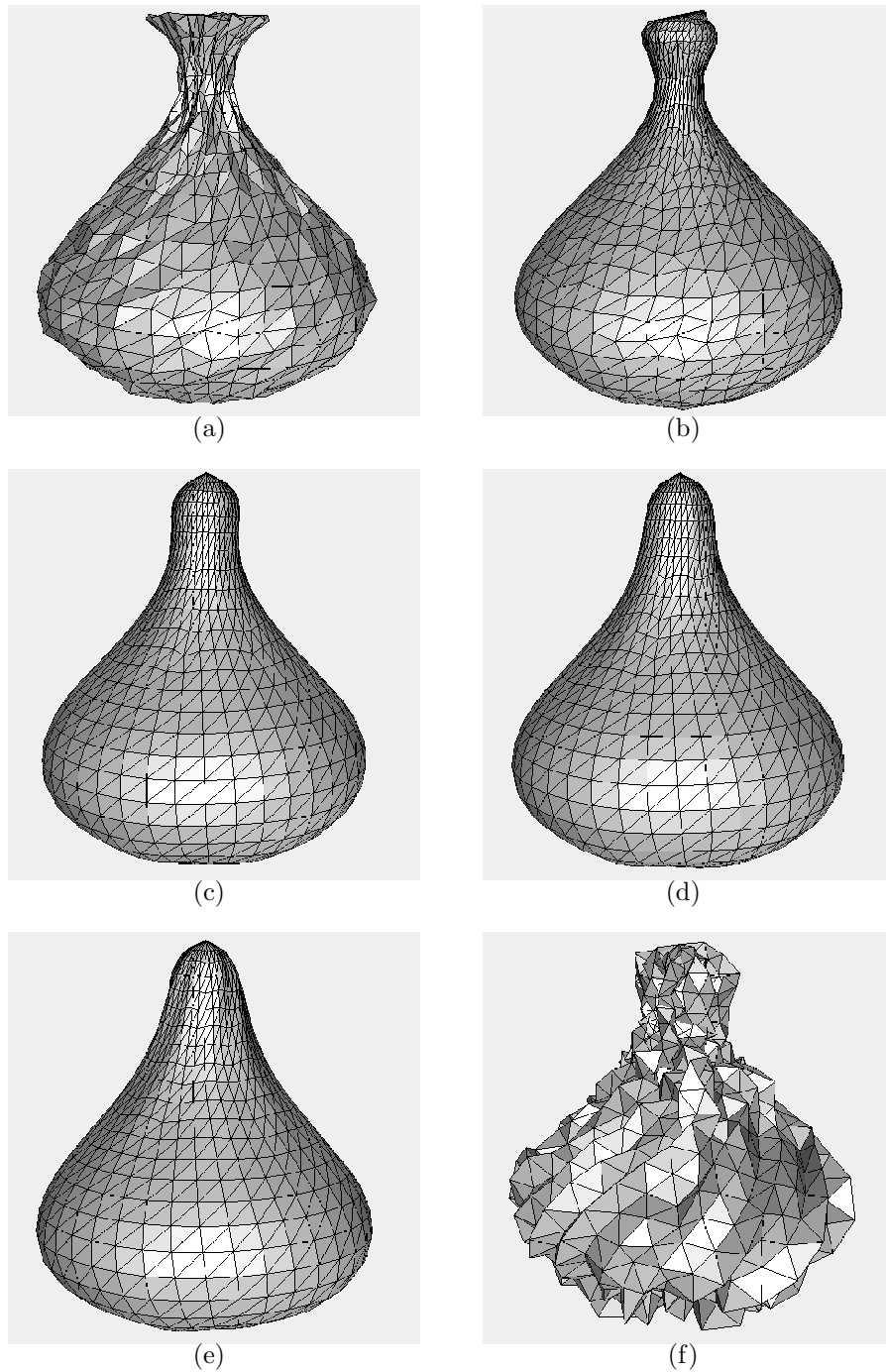
Tabulka 3.2: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různé nastavení parametru λ_u vyhlazovaného torusu (viz. obrázek 3.14). Parametr $\lambda = 0.01$.

a $T_u = 0$), triangulovaná síť je po vyhlazení velmi nerovnoměrná. Pokud je vyhlazována jen na kritérium neuniformity ($T_h = 0$ a $T_u = 100$), kolem vrcholů vázičky vzniká pravidelné okolí, ale křivost je zvětšena. Nejvhodnější nastavení, vzhledem k velikosti křivosti (po 5000 iteracích), je $T_h = T_u = 50$. Vzhledem k velikosti neuniformity, je však výhodnější nastavení $T_h = 30$ a $T_u = 70$. Toto nastavení je i vizuálně nejlépe vyhlazeno.

3.8.5 Shrnutí

V této kapitole shrneme provedené experimenty.

Výstupem algoritmu jsou vyhlazené uzavřené povrchy, které nejsou smršťené. Bylo také ukázáno, že vyhlazením jsou zachovány inflexní body, které povrch obsahoval před zašuměním. Algoritmus je schopný vyhladit i extrémně zašuměné povrchy a pokud je vyhlazený povrch zašuměn a následně opět vyhlazen, výstupem je povrch stejného tvaru, jako před zašuměním. Experimenty také prokázaly, že je možné algoritmus využít pro natažení hladké sítě mezi několika objekty. Experimenty ukázaly, že není možné (jako u křivek) přesně říci jedno vhodné nastavení parametrů algoritmu. Uvedeme ty parametry, které vedly nejčastěji k vyhlazenému povrchu: Parametry $\lambda = 0.01$, $\lambda_u = 0.01$, nebo $\lambda_u = 0.001$. Parametry pro volbu schématu jsou $T_h = 50$ a $T_u = 50$, nebo $T_h = 30$ a $T_u = 70$. Parametry posunu jsou $S_h = 0.00001$ a $S_u = 0.00001$.



Obrázek 3.15: Vyhlazování zašuměné vázičky různými schématy (nastavení parametrů T_h a T_u). Parametry $\lambda = 0.01$ a $\lambda_u = 0.01$. Data k průběhu vyhlazování jsou v tabulce 3.3. (a) Vstup algoritmu. (b) Parametry $T_h = 100$ a $T_u = 0$. (c) Parametry $T_h = 70$ a $T_u = 30$. (d) Parametry $T_h = 50$ a $T_u = 50$. (e) $T_h = 30$ a $T_u = 70$ (f) $T_h = 0$ a $T_u = 100$.

T_h/T_u	Hodnota H_{avg}/H_0 po iteraci					
	800	1700	2500	3300	4200	5000
0/100	0.910	0.926	0.904	0.847	0.786	0.737
30/70	0.311	0.278	0.272	0.270	0.271	0.275
50/50	0.292	0.276	0.267	0.258	0.254	0.254
70/30	0.300	0.289	0.280	0.274	0.268	0.262
100/0	0.319	0.322	0.327	0.327	0.328	0.329

T_h/T_u	Hodnota U_{avg}/U_0 po iteraci					
	800	1700	2500	3300	4200	5000
0/100	0.243	0.227	0.211	0.202	0.204	0.207
30/70	0.240	0.196	0.175	0.163	0.154	0.147
50/50	0.281	0.226	0.197	0.188	0.182	0.181
70/30	0.357	0.266	0.230	0.213	0.198	0.193
100/0	0.764	0.744	0.740	0.736	0.737	0.744

T_h/T_u	Hodnota S_{avg}/S_0 po iteraci					
	800	1700	2500	3300	4200	5000
0/100	0.243	0.227	0.212	0.202	0.204	0.208
30/70	0.016	0.013	0.011	0.011	0.010	0.010
50/50	0.018	0.015	0.013	0.012	0.012	0.012
70/30	0.023	0.017	0.015	0.014	0.013	0.013
100/0	0.007	0.005	0.003	0.003	0.002	0.003

Tabulka 3.3: Tabulky hodnot funkcí $\frac{H_{\text{avg}}}{H_0}$, $\frac{U_{\text{avg}}}{U_0}$ a $\frac{S_{\text{avg}}}{S_0}$ pro různá schémata (nastavení parametrů T_h a T_u) vyhlazované zašuměné vázičky (viz. obrázek 3.15). Parametry $\lambda = 0.01$, $\lambda_u = 0.01$.

Kapitola 4

Vyhlazování povrchů s hranicemi

Tato kapitola popisuje vyhlazování povrchů s hranicemi.

4.1 Rozdělení vrcholů povrchu a nalezení hranic

Nechť S je diskrétně zadaný povrch s hranicemi, dále jen povrch s hranicemi, na kterém chceme provádět vyhlazování. Definice povrchu a používaný zápis jsou shodné se zápisem pro uzavřený povrch v kapitole 3.3.

Vrcholy povrchu S rozdělíme na dvě skupiny: vrcholy, které leží uvnitř povrchu, a vrcholy hraniční, které leží na některé z hranic povrchu. Hraniční vrchol lze definovat jako vrchol, pro který existuje alespoň jedna hrana, která tento vrchol obsahuje, taková, že je obsažena právě v jednom trojúhelníku ze všech trojúhelníků povrchu.

Hraniční vrcholy tvoří t uzavřených hranic povrchu, kde $t \geq 0$. Soubor těchto hranic budeme v dalším textu označovat B .

4.2 Algoritmus vyhlazení

Myšlenkou algoritmu pro vyhlazování povrchů s hranicemi je využití již uvedených algoritmů pro vyhlazování křivek a uzavřených povrchů. Algoritmus pro vyhlazení povrchu s hranicemi popíšeme takto:

Dokud nejsou splněny ukončovací podmínky, opakuj pro každou j -tou iteraci algoritmu:

1. Pro každou hranici z B aplikuj algoritmus vyhlazení křivky (viz. 4.3).
2. Pro všechny vrcholy povrchu, které nejsou hraniční, aplikuj algoritmus vyhlazení povrchu (viz. 4.4).

4.3 Algoritmus vyhlazení hranice

Algoritmus pro vyhlazení hranice je shodný s algoritmem pro vyhlazení samostatné křivky (viz. kapitola 2). Výsledkem vyhlazování jsou nové vrcholy hranice, kterými nahradíme odpovídající vrcholy povrchu.

4.4 Algoritmus vyhlazení povrchu

Algoritmus pracuje shodně jako algoritmus pro vyhlazení uzavřeného povrchu (viz. kapitola 3) až na krok, kdy je v k -té iteraci algoritmu určována nová hodnota křivosti H^{k+1} ve vrcholu \mathbf{a}^k . Neuniformita se určí zcela shodně pro všechny vrcholy povrchu jako na uzavřeném povrchu.

Postup pro uzavřený povrch: Nová hodnota křivosti ve vrcholu \mathbf{a}^k je definována předpisem:

$$H^{k+1} = (1 - \lambda)H^k + \lambda\bar{H}^k, \quad 0 < \lambda < 1, \quad (4.1)$$

kde

$$\bar{H}^k = \frac{\sum_{i=1}^n s_i^k \cdot H_i^k}{n}. \quad (4.2)$$

Výraz H_i^k označuje velikost křivosti bezprostředního souseda vrcholu \mathbf{a}^k a s_i^k je znaménko přenosu vektoru křivosti \mathbf{H}_i^k .

Mezi vrcholy povrchu s hranicí existují takové vrcholy, jejichž množina bezprostředních sousedů obsahuje alespoň jeden vrchol, který je hraniční. V hraničním vrcholu nepočítáme \mathbf{H}^k a není tedy možné ani určit znaménko přenosu s_i^k .

Novou hodnotu křivosti ve vrcholu \mathbf{a}^k tedy určíme jen pomocí bezprostředních sousedů, kteří nejsou hraniční. Vzorec pro výpočet nové hodnoty křivosti tedy zůstane zachován, ale změny se vyhodnocení proměnné \bar{H}^k . Tedy

$$\bar{H}^k = \frac{\sum_{i=1}^n s_i^k \cdot H_i^k}{n}, \quad (4.3)$$

kde H_i^k je velikost křivosti bezprostředního nehraničního souseda vrcholu \mathbf{a}^k , n je počet nehraničních sousedů a s_i^k je znaménko přenosu vektoru křivosti \mathbf{H}_i^k .

Speciální případ:

Mezi nehraničními vrcholy povrchu mohou existovat takové, jejichž všichni bezprostřední sousedé jsou hraniční vrcholy. V tomto případě by u vzorce pro výpočet nové hodnoty křivosti docházelo k dělení nulou. Definujeme výpočet nové hodnoty křivosti jako

$$H^{k+1} = (1 - \lambda)H^k, \quad 0 < \lambda < 1. \quad (4.4)$$

4.5 Průběh a ukončovací podmínky algoritmů

Pro běh a ukončení algoritmu je nutné zvolit několik konstant. V dalším textu budeme algoritmus pro vyhlazení povrchu s hranicemi označovat jako algoritmus A , podalgoritmus algoritmu A pro vyhlazení křivek jako C a podalgoritmus algoritmu A pro vyhlazení uzavřených povrchů jako S .

1. Konstanty pro běh algoritmu vyhlazující povrch s hranicemi (algoritmus A):

- (a) Konstanta C_{switch} určuje počet iterací, které mají být provedeny podalgoritmem C v jedné iteraci j algoritmu A .
- (b) Konstanta S_{switch} určuje počet iterací, které mají být provedeny podalgoritmem S v jedné iteraci j algoritmu A .
- (c) $C_{maxIter}$ určuje maximální počet iterací, které je možné provést algoritmem C za celý běh algoritmu A . Iterace podalgoritmu C se v průběhu běhu algoritmu A sčítají. Pokud je dosaženo $C_{maxIter}$, v dalších krocích již podalgoritmus C není spouštěn.
- (d) $S_{maxIter}$ určuje maximální počet iterací, které je možné provést podalgoritmem S za celý běh algoritmu A . Iterace podalgoritmu S se v průběhu běhu algoritmu A sčítají. Pokud je dosaženo $S_{maxIter}$, v dalších krocích již podalgoritmus S není spouštěn.
- (e) C_{sum} je konstanta určující minimální přípustný průměr posunu všech vrcholů křivky v průběhu běhu podalgoritmu C . Pokud je průměr posunů vrcholů křivky menší než daná konstanta, je podalgoritmus C ukončen a v dalších krocích algoritmu A již není spouštěn.
- (f) S_{sum} je konstanta určující minimální přípustný průměr posunu všech nehraničních vrcholů povrchu v průběhu běhu podalgoritmu S . Pokud je průměr posunů vrcholů křivky menší než daná konstanta, je podalgoritmus S ukončen a v dalších krocích algoritmu A není spouštěn.

2. Konstanty pro běh podalgoritmu C a podalgoritmu S :

- (a) Konstanty λ a λ_u určují lambda pro výpočet nové hodnoty křivosti a nové hodnoty neuniformity ve vrcholu.
- (b) Konstanty $C_{H_{switch}}$ a $C_{U_{switch}}$ (analogicky $S_{H_{switch}}$ a $S_{U_{switch}}$) určují, po kolika iteracích se bude měnit kritérium, podle kterého se pracuje. Pokud je pracováno s kritériem křivosti a bylo takto provedeno $C_{H_{switch}}$ iterací, algoritmus v dalším kroku začne pracovat s kritériem neuniformity a je provedeno $C_{U_{switch}}$ iterací, pak se opět začne pracovat podle kritéria křivosti.

Ukončení algoritmu A nastává v případech:

1. Pokud podalgoritmus C již doběhl (viz. konstanty pro běh algoritmu) a poté doběhl podalgoritmus S .
2. Pokud podalgoritmus S skončil na maximální počet iterací a poté doběhl podalgoritmus C .

Poznámka: V případě, že podalgoritmus S doběhl na sumu, je možné po provedení filtrace podalgoritmem C dále pokračovat v běhu podalgoritmem S .

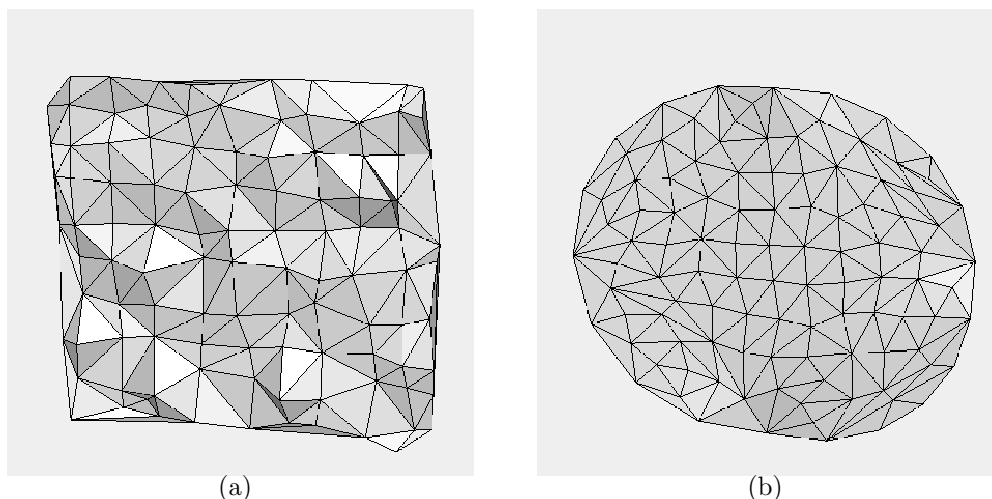
4.6 Experimenty

V této kapitole ukážeme výsledky implementace algoritmu pro vyhlazování povrchů s hranicemi. Uvedeme experimenty, které prokazují robustnost a stabilitu algoritmu. Na závěr kapitoly uvedeme výsledky na reálných datech.

V celé kapitole budeme používat jednotné značení pro vstupní parametry algoritmu. Všechny parametry začínající písmenem C náležejí k vyhlazování hranice (křivky) a všechny parametry začínající písmenem S náležejí k vyhlazování povrchu bez zadané hranice. Parametr C_λ určuje velikost difusního kroku pro kritérium křivosti (viz. (2.7)) a parametr C_{λ_u} určuje velikost difusního kroku pro kritérium neuniformity (viz. (2.13)), pro hranici. Analogicky pro parametry S_λ (viz. (3.7)) a S_{λ_u} (viz. (3.18)) pro povrch. Parametry $C_{maxIter}$ a $S_{maxIter}$ určují maximální počet iterací pro hranici a povrch. Parametry C_{sum} a S_{sum} určují minimální přípustný průměr všech posunů vrcholů pro hranici a povrch. Parametry $C_{H_{switch}}$ a $C_{U_{switch}}$ určují po kolika iteracích se bude měnit kritérium vyhlazování pro hranici. Obdobně pro parametry $S_{H_{switch}}$ a $S_{U_{switch}}$ pro povrch. Podrobnější popis parametrů lze nalézt v kapitole 4.5.

U popisu každého experimentu budou popsány parametry, které jsou zvoleny jinak, než je jejich předdefinovaná hodnota. Parametry S_λ , S_{λ_u} , C_λ a C_{λ_u} jsou uváděny vždy. Parametry $C_{maxIter}$ a $S_{maxIter}$ jsou uváděny, jen pokud algoritmus nedokonal. Jinak je uveden počet iterací, pro které algoritmus dokonal. Pokud parametr není v popisu experimentu uveden, jeho hodnota je:

- $C_{sum} = 0.0001$,
- $S_{sum} = 0.00001$,
- $C_{H_{switch}} = 50$,
- $C_{U_{switch}} = 50$,
- $S_{H_{switch}} = 50$,



Obrázek 4.1: Vyhlazení zašuměného rovinného čtverce. Parametry jsou $S_{maxIter} = 5000$, $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1404 iteracích. (a) Vstup algoritmu. (b) Vyhlazený povrch.

- $S_{U_{switch}} = 50$.

Experimentálně bylo ověřeno, že nastavení parametrů C_{switch} a S_{switch} (viz. kapitola 4.5) nemá na výsledek algoritmu vliv. Všechny experimenty jsou prováděny tak, že je nejdříve vyhlazena křivka a poté povrch.

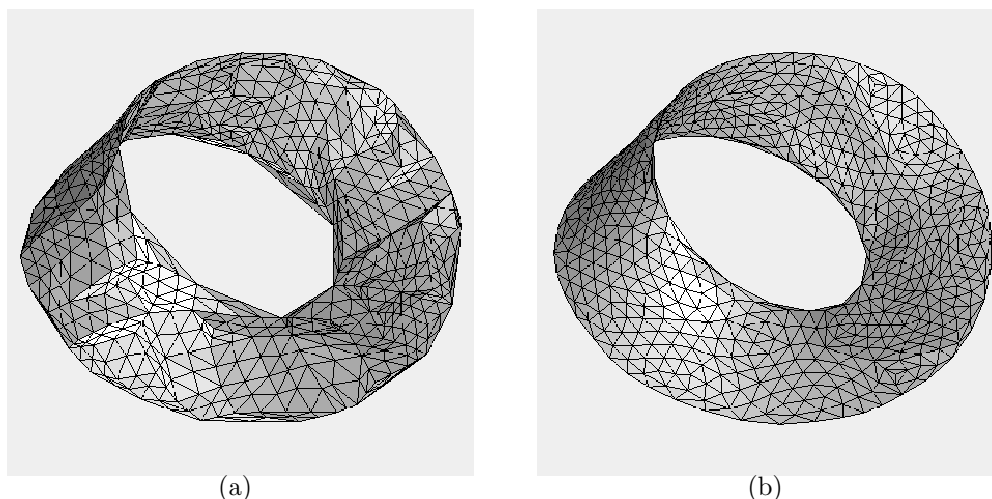
4.6.1 Základní funkce

V této kapitole jsou uvedeny experimenty, které ukazují základní funkci algoritmu.

Ke každému experimentu je uveden výsledek tak, že obrázek (a) zobrazuje vstup algoritmu, obrázek (b) zobrazuje vyhlazený povrch s hranicemi.

V prvním experimentu je vstupem zašuměný čtverec. Parametry jsou $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1404 iteracích. Vyhlazení povrchu skončilo po $S_{maxIter} = 5000$ iteracích. Průběh vyhlazování je na obrázku 4.1. Po vyhlazení je síť uvnitř objektu rovnoměrná. Na hranicích se objevují nerovnoměrnosti, které jsou však způsobeny vstupními daty. Výsledný objekt je téměř rovinný díky velikosti vstupního šumu.

V dalším experimentu je vstupem zašuměná moebiova páska. Parametry jsou $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.01$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 434 iteracích. Vyhlazení povrchu skončilo po $S_{maxIter} = 5000$ iteracích. Průběh vyhlazování je na obrázku 4.2. Vyhlazením zašuměné moebiovy pásky byly odstraněny nepěkné dlouhé trojúhelníky na okrajích povrchu a povrch pásky je po vyhlazení hladký.



Obrázek 4.2: Vyhlazení zašuměné moebiovy pásky. Parametry jsou $S_{maxIter} = 5000$, $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.01$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 434 iteracích. (a) Vstup algoritmu. (b) Vyhlazený povrch.

4.6.2 Robustnost a stabilita

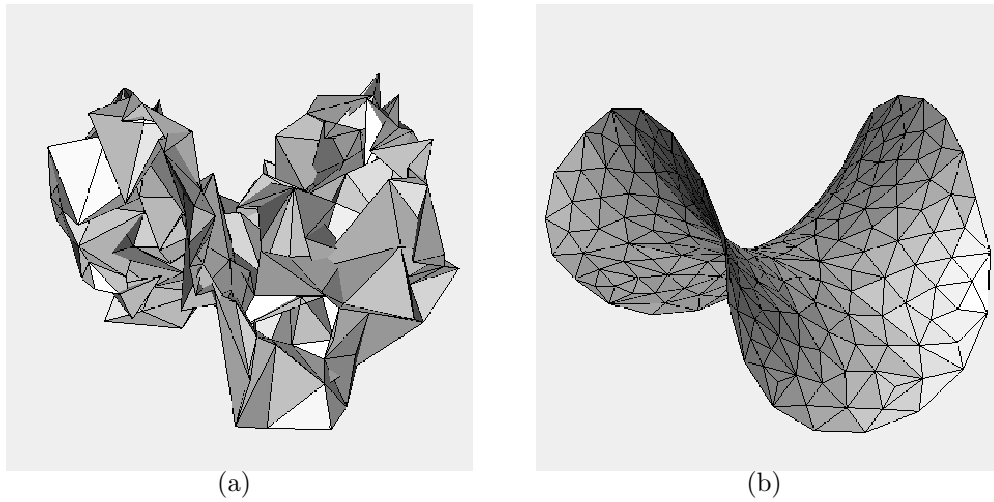
V této kapitole jsou uvedeny experimenty, které prokazují robustnost a stabilitu algoritmu. Robustnost je ukázána na vyhlazení extrémně zašuměného povrchu s hranicemi. Stabilita je prokázána tak, že k již vyhlazenému povrchu je přidán šum a takto pozměněný povrch je algoritmem znovu vyhlazen.

Experimenty jsou zobrazovány stejně jako experimenty v kapitole 4.6.1.

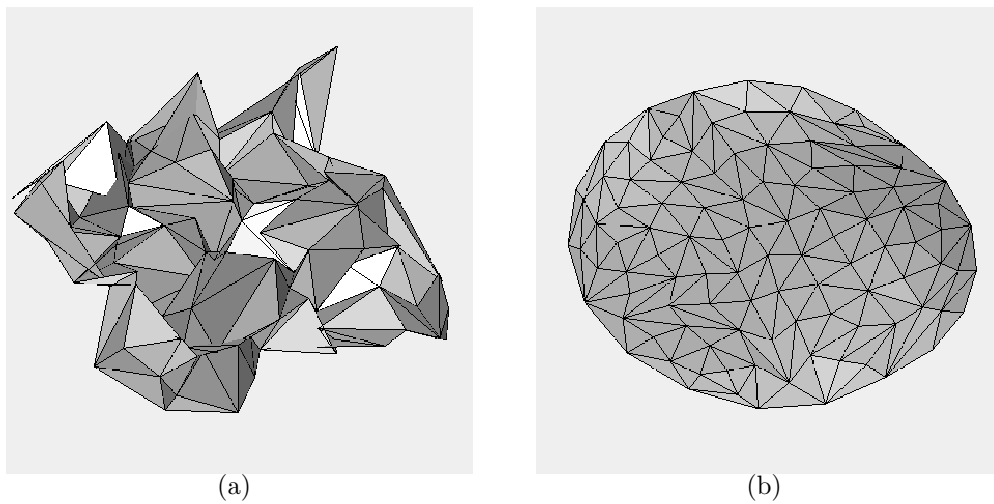
Nejprve jsme testovali robustnost na extrémně zašuměném sedle, na kterém budeme pozorovat, zda algoritmus zachovává inflexní body. Parametry jsou $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1694 iteracích. Vyhlazení povrchu skončilo po $S_{maxIter} = 10000$ iteracích. Průběh vyhlazování je na obrázku 4.3. I přes extrémní šum je povrch vyhlazen a inflexní body jsou zachovány.

V dalším experimentu je vstupem extrémně zašuměný rovinný čtverec. Parametry jsou $S_\lambda = 0.01$, $S_{\lambda_u} = 0.001$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1527 iteracích. Vyhlazení povrchu skončilo po $S_{maxIter} = 5000$ iteracích. Průběh vyhlazování je na obrázku 4.4. Vyhlazením extrémně zašuměného čtverce nebyl získán kruh, což je ale vzhledem k vstupnímu povrchu akceptovatelné. Vyhlazený povrch je, narozdíl od vstupu, téměř rovinný.

K experimentu stability je uveden výsledek tak, že obrázek (a) zobrazuje vstup algoritmu, obrázek (b) zobrazuje vyhlazený povrch, obrázek (c) zobrazuje zašumění již vyhlazeného povrchu s hranicemi a obrázek (d) znovu vyhlazený povrch.



Obrázek 4.3: Vyhlazení extrémně zašuměného sedla. Parametry jsou $S_{maxIter} = 10000$, $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1694 iteracích. (a) Vstup algoritmu. (b) Vyhlazený povrch.



Obrázek 4.4: Vyhlazení extrémně zašuměného čtverce. Parametry jsou $S_{maxIter} = 5000$, $S_\lambda = 0.01$, $S_{\lambda_u} = 0.001$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 1527 iteracích. (a) Vstup algoritmu. (b) Vyhlazený povrch.

Vstupem experimentu je zašuměné sedlo. Parametry jsou $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Algoritmus vyhlazení hranice dokonvergoval po 4076 iteracích. Vyhlazování povrchu skončilo po $S_{maxIter} = 10000$ iteracích. K vyhlazenému povrchu byl přidán šum, který je mnohem větší než první šum, a povrch byl znovu vyhlazen se shodnými parametry λ a $S_{maxIter}$ jako u prvního vyhlazení. Algoritmus vyhlazení hranice dokonvergoval po 677 iteracích. Vyhlazování povrchu skončilo po $S_{maxIter} = 10000$ iteracích. Průběh vyhlazování je na obrázku 4.5. I přes velké druhé zašumění sedla je výsledek vyhlazování, až na malé lokální rozdíly, shodný s výsledkem prvního vyhlazení.

4.6.3 Reálná data

V této kapitole uvedeme experimenty na reálných datech.

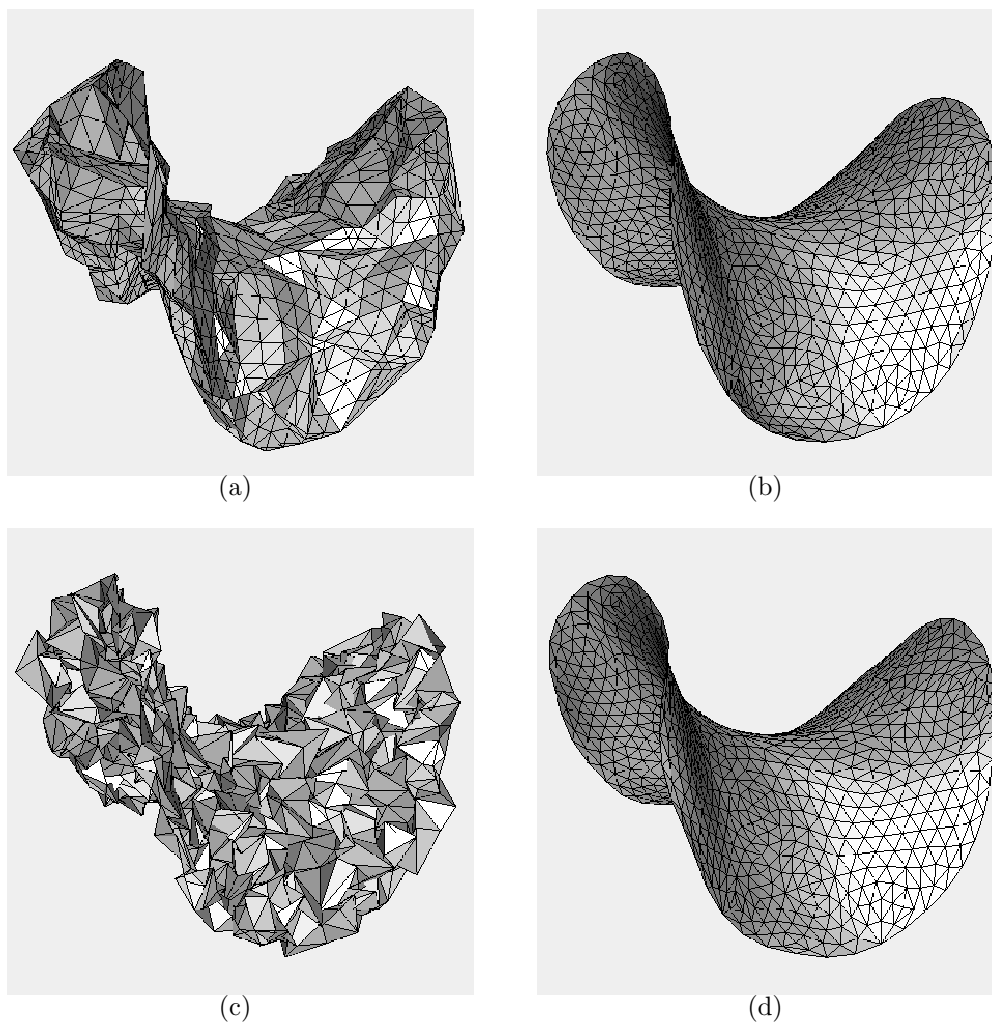
Vstupními daty pro experimenty na reálných datech jsou triangulované povrchy s hranicí, které zachycují obličej. Vzhledem k tomu, že jsou data získána 3D rekonstrukcí, obsahují velký šum. Dalším problémem je špatná triangulace povrchu. To je způsobeno za prvé převodem šupin na triangulovaný povrch a za druhé tím, že při snímání obličeje nebyla některá místa nasnímána a v těchto místech tedy chybí informace o povrchu.

Vzhledem k triangulaci jsme se rozhodli provést experiment tak, že jsme prvních 50 kroků algoritmu vyhlazovali jen podle kritéria neuniformity. Toto vyhlazení velmi zlepšilo triangulaci povrchu a tím také urychlilo a zlepšilo další vyhlazování (5000 iterací), které bylo provedeno jen podle kritéria křivosti.

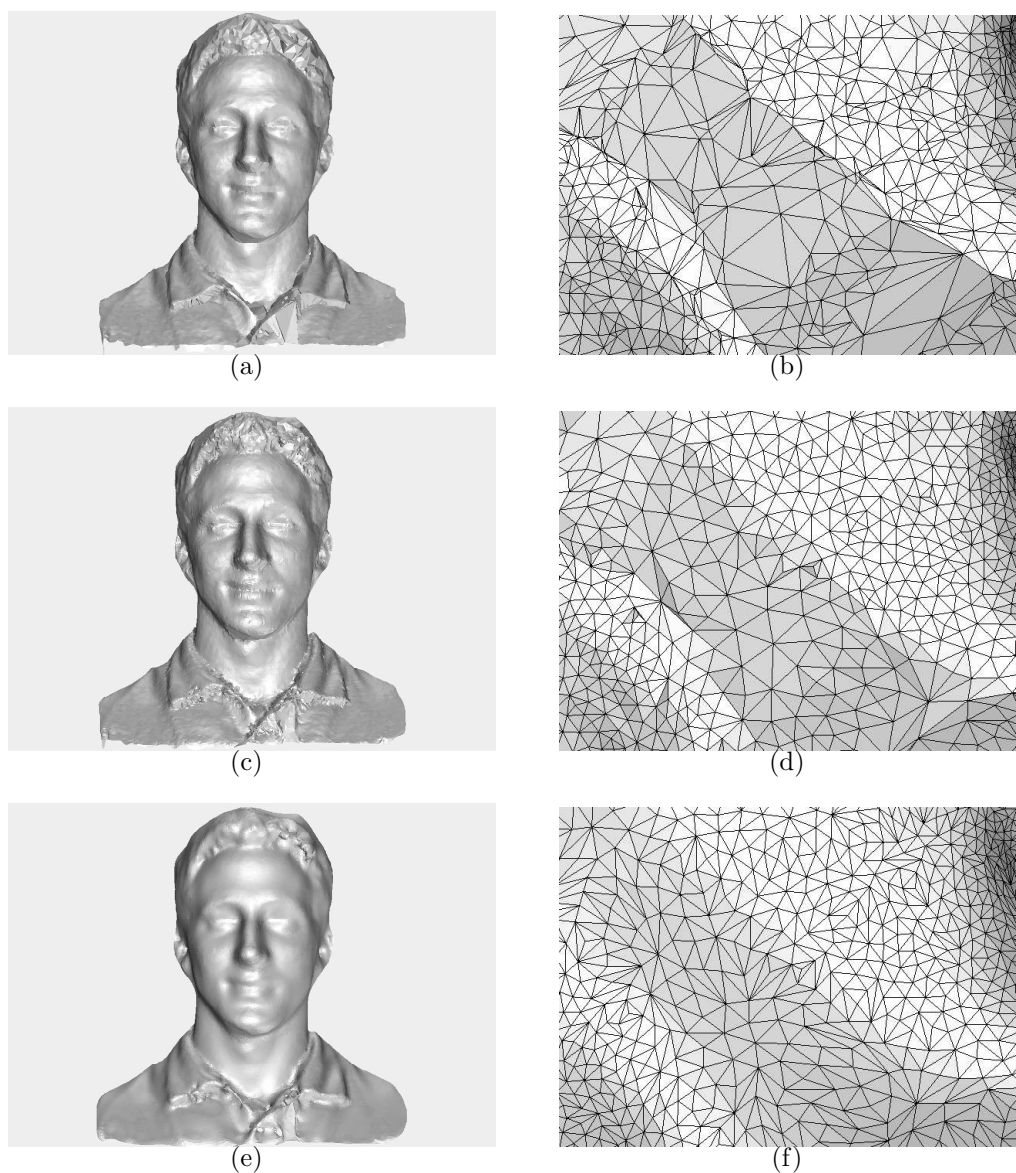
První experiment byl proveden na povrchu, který obsahuje 26194 vrcholů a 51879 trojúhelníků. Povrch je zachycen na obrázku 4.6(a). Na obrázku 4.6(b) je zvětšená část vstupního povrchu v oblasti přechodu obličeje a krku.

Parametry prvního vyhlazení (jen kritérium neuniformity) jsou $S_{maxIter} = 50$, $C_{maxIter} = 50$, $S_\lambda = 0$, $S_{\lambda_u} = 0.1$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Výsledek prvního vyhlazení je na obrázku 4.6(c). Pozorujeme, že křivost povrchu vzrostla, ale triangulace povrchu se výrazně zlepšila, což lze pozorovat na zvětšené části, která je na obrázku 4.6(d). Zvětšená část zachycuje stejnou oblast jako je na obrázku 4.6(b).

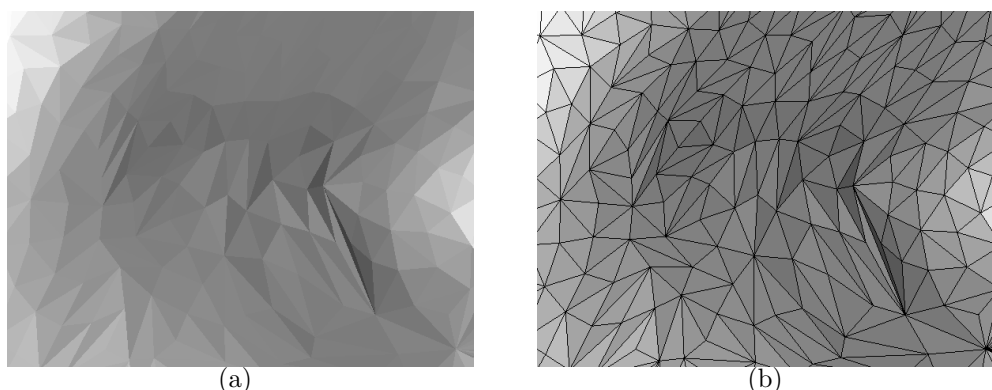
Parametry druhého vyhlazení (jen kritérium křivosti) jsou $S_{maxIter} = 5000$, $C_{maxIter} = 5000$, $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. Výsledek druhého vyhlazení je na obrázku 4.6(e). Pozorujeme výrazné zlepšení. Nejvýrazněji je změna pozorovatelná na vlasech a límečku košile. Na obrázku 4.6(f) je zvětšená část vyhlazeného povrchu. Výřez opět zachycuje stejnou oblast jako výřez vstupního povrchu. Pokud porovnáme výřezy po prvním a druhém vyhlazení, je vidět, že triangulace povrchu je rovnoměrnější po prvním vyhlazování. Pokud se ale soustředíme na jednotlivé trojúhelníky lze pozorovat, že povrch po prvním vyhlazení je méně hladký než povrch



Obrázek 4.5: Vyhlazení zašuměného sedla. Parametry $S_\lambda = 0.01$, $S_{\lambda_u} = 0.01$, $C_\lambda = 0.1$ a $C_{\lambda_u} = 0.01$. (a) Vstup algoritmu. (b) Vyhlazený povrch. Parametr $S_{maxIter} = 10000$. Algoritmus vyhlazení hranice dokonvergoval po 4076 iteracích. (c) Přidán šum. (d) Vyhlazení zašuměného povrchu. Parametr $S_{maxIter} = 10000$. Algoritmus vyhlazení hranice dokonvergoval po 677 iteracích.



Obrázek 4.6: Vyhlazení obličeje. (a) Vstupní povrch. (b) Zvětšená část povrchu. (c) Povrch po prvním vyhlazení (kritérium neuniformity). Parametry jsou $S_{maxIter} = 50$, $C_{maxIter} = 50$, $S_{\lambda} = 0$, $S_{\lambda_u} = 0.1$, $C_{\lambda} = 0.1$ a $C_{\lambda_u} = 0.01$. (d) Zvětšená část po prvním vyhlazení. (e) Povrch po druhém vyhlazení (kritérium křivosti). Parametry jsou $S_{maxIter} = 5000$, $C_{maxIter} = 5000$, $S_{\lambda} = 0.01$, $S_{\lambda_u} = 0.01$, $C_{\lambda} = 0.1$ a $C_{\lambda_u} = 0.01$. (f) Zvětšená část po druhém vyhlazení.



Obrázek 4.7: Oblast výskytu trojúhelníků, u kterých by bylo možné provést přetriangulování. (a) Oblast bez zvýrazněné triangulace. (b) Oblast se zvýrazněnou triangulací.

po druhém vyhlazení (větší rozdíly ve stupních šedi mezi sousedními trojúhelníky).

Ve výsledku druhého vyhlazování jsme pozorovali, že pokud bychom v průběhu vyhlazování změnili triangulaci tak, že bychom některé dvojice trojúhelníků triangulovali druhou diagonálou, výsledek vyhlazování by se zlepšil. Změnou diagonály by vznikly trojúhelníky, které se mohou lépe přizpůsobit tvaru povrchu v jejich okolí. Výskyt takových trojúhelníků je možné nalézt například v okolí pravé nosní dírky (obrázky 4.7(a) a 4.7(b)).

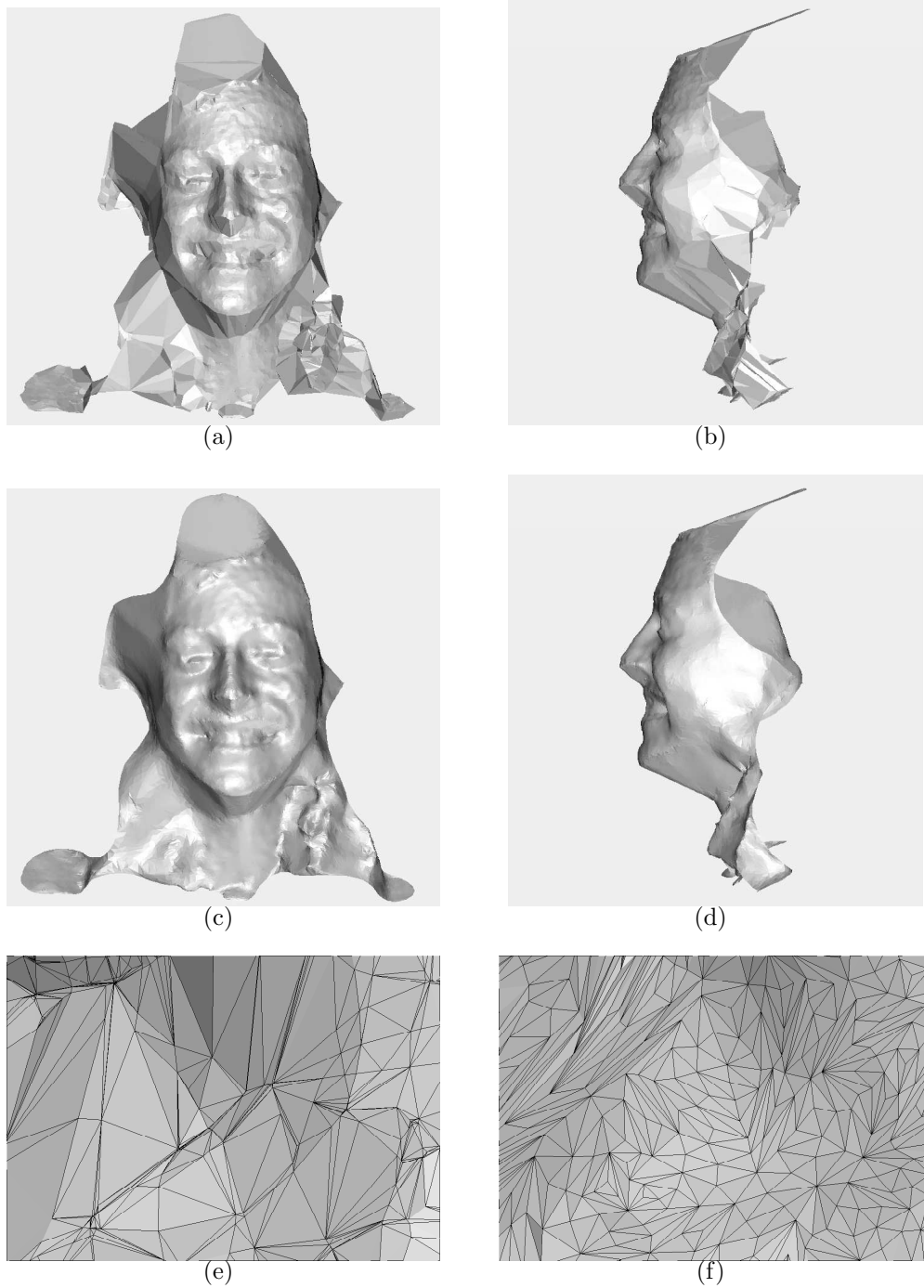
Druhý experiment byl proveden na povrchu, který obsahuje 49566 vrcholů a 98766 trojúhelníků. Povrch je zachycen na obrázku 4.8(a) a na obrázku 4.8(b). Triangulace obsahuje vrcholy se stupněm tři, ale také vrcholy se stupněm třicet a navíc jsou v triangulaci obsaženy jak velké trojúhelníky, tak i trojúhelníky, které jsou téměř degenerované. Na obrázku 4.8(e) je zvětšená část vstupního povrchu, kde je vidět triangulace povrchu.

Parametry prvního i druhého vyhlazení jsou shodné s parametry předchozího experimentu. Na obrázku 4.8(c) a 4.8(d) je vyhlazený povrch a na obrázku 4.8(f) je výřez části vyhlazeného povrchu, který se shoduje s výřezem části vstupního povrchu. Na výřezu je v levém horním rohu vidět velmi protáhlé trojúhelníky. Ty jsou způsobeny špatnou triangulací vstupu, a to převážně velkými rozdíly mezi stupni vrcholů. Celkově však lze pozorovat výrazné zlepšení proti vstupnímu povrchu.

4.6.4 Shrnutí

V této kapitole shrneme provedené experimenty.

Experimenty prokázaly, že volba vyhlazovat povrch s hranicí kombinací algoritmů pro vyhlazování křivek a uzavřených povrchů je dobrá. Spojením



Obrázek 4.8: Vyhlazení obličeje. (a) Obličej z anfasu. Vstupní povrch. (b) Obličej z profilu. Vstupní povrch. (c) Obličej z anfasu. Povrch po druhém vyhlazení (kritérium křivosti). Parametry jsou $S_{maxIter} = 5000$, $C_{maxIter} = 5000$, $S_{\lambda} = 0.01$, $S_{\lambda_u} = 0.01$, $C_{\lambda} = 0.1$ a $C_{\lambda_u} = 0.01$. (d) Obličej z profilu. Povrch po druhém vyhlazení. (e) Zvětšená část vstupního povrchu. (f) Zvětšená část po druhém vyhlazení.

algoritmů pro křivky a uzavřené povrchy tedy vznikl algoritmus, jehož výstupem jsou vyhlazené povrchy s hranicí. Navíc bylo ukázáno, že je algoritmus schopný vyhladit i extrémně zašuměné povrchy s hranicí a, pokud do již vyhlazeného povrchu přidáme šum a znovu jej vyhladíme, výstupem algoritmu je povrch, jehož tvar je shodný s tvarem povrchu před zašuměním.

Experimentálně bylo také ověřeno, že lze nejdříve vyhladit hranici a poté vyhladit vnitřní vrcholy povrchu. Není tedy nutné v průběhu algoritmu tato vyhlazování střídat. Nastavení parametrů je vhodné použít shodné jako v případě vyhlazování křivky a uzavřeného povrchu.

Algoritmus jsme vyzkoušeli také na reálných datech, která zachycují 3D modely obličejů. Experimenty prokázaly schopnost algoritmu zpracovávat i extrémně velké množiny dat (35000 vrcholů a 100000 trojúhelníků). I přes to, že data obsahovala velký šum a triangulace povrchu byla velmi nerovnoměrná, výstupem algoritmů byly vyhlazené povrchy.

Kapitola 5

Závěr

Cílem této práce bylo navrhnout algoritmus, který řeší úlohu vyhlazování diskrétně zadaného povrchu s hranicí, reprezentovaného triangulovanou sítí. Na algoritmus byl kladen požadavek, aby jeho výstupem byl vyhlazený povrch s hranicí, který není smrštěný. Algoritmus by neměl měnit triangulaci sítě a vyhlazení by tedy mělo být provedeno jen změnou polohy vrcholů sítě.

Aby bylo možné navrhnout samotný algoritmus pro vyhlazení povrchu s hranicí, nejdříve byly navrženy algoritmy pro vyhlazení křivky a uzavřeného povrchu. Z nich bylo čerpáno při vytváření konečného řešení algoritmu. Vnitřní vrcholy povrchu s hranicí jsou vyhlazovány algoritmem pro vyhlazování uzavřeného povrchu a hranice je vyhlazována algoritmem pro vyhlazování křivky.

Nad zadanou diskrétní množinou vrcholů (křivky či povrchu) definujeme dvě kritéria optimalizace (obě diskrétně): *křivost* pro získání hladkosti a *neuniformitu* pro docílení regularity. Obě kritéria jsou definována jen na základě bezprostředních sousedů vrcholu. Tato definice umožňuje lepší a rychlejší práci s vrcholy a zároveň zachovat tvarový charakter vstupní množiny. Vyhlazování provádíme nezávisle podle jednotlivých kritérií algoritmem diskrétní difuze.

Cíl práce byl splněn. Byl navržen algoritmus, který řeší úlohu vyhlazování povrchu s hranicí. Z povrchu je tedy odstraněn šum a triangulovaná síť je rovnoměrná. Experimentálně bylo ověřeno na testovacích i reálných datech, že výstupem algoritmu jsou vyhlazené objekty. Výhodou algoritmu je zachování velikosti i základního tvaru objektu a nesmrštění povrchu v průběhu vyhlazení. Nevýhodou algoritmu je nutnost velkého počtu iterací k vyhlazení rozsáhlých dat.

Jedno z možných rozšíření do budoucna je navržení algoritmu, který by v průběhu vyhlazování zlepšoval triangulaci sítě. Experimenty prokázaly, že změna triangulace by v některých případech vedla na lépe vyhlazený povrch.

Další z možných rozšíření je změna pohledu na povrch s hranicí jako na jeden celek. Hranice povrchu by již nebyla oddělována od povrchu, ale zpracována stejně jako zbytek povrchu.

Závěrem můžeme říct, že navržený algoritmus vyhlazuje mírně zašuměná data i data s extrémním šumem. Lze jej tedy použít nejen k vyhlazování synteticky vytvořených dat, ale i v praxi k vyhlazování dat získaných metodou 3D rekonstrukce, což bylo hlavní motivací k práci.

Literatura

- [1] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, New York, NY, USA, 1998. ACM Press.
- [2] B. Budinský. *Analytická a diferenciální geometrie*. Nakladatelství technické literatury, 1983.
- [3] O. Chum, J. Matas, and J. Kittler. Consistent multi-view reconstruction from epipolar geometries with outliers. In *SCIA*, pages 493–500, 2003.
- [4] H. Cornelius, R. Šára, D. Martinec, T. Pajdla, O. Chum, and J. Matas. Towards complete free-form reconstruction of complex 3d scenes from an unordered set of uncalibrated images. In *Statistical Methods in Video Processing*, pages 1–12, Berlín, Německo, 2004. Springer.
- [5] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [6] N. Dyn, K. Hormann, S. Kim, and D. Levin. Optimizing 3d triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces: Oslo 2000*, pages 135–146, Nashville, TN, USA, 2001. Vanderbilt University.
- [7] G. Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13(3):143–154, 1994.
- [8] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph.*, 22(3):943–949, 2003.
- [9] L. P. Kobbelt. Discrete fairing and variational subdivision for freeform surface design. *The Visual Computer*, 16(3-4):142–158, 2000.
- [10] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In Richard Harvey and J. Andrew Bangham, editors,

Proceedings of the British Machine Vision Conference, volume 1, pages 339–348, Londýn, UK, 2003. BMVA Press.

- [11] T. Mashiko, H. Yagou, D. Wei, Y. Ding, and G. Wu. 3d triangle mesh smoothing via adaptive mmse filtering. In *The Fourth International Conference on Computer and Information Technology (CIT'04)*, pages 734–740, 2004.
- [12] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, Londýn, 2002.
- [13] J. Žára, B. Beneš, and P. Felkel. *Moderní počítačová grafika*. Computer Press, 1998.
- [14] R. Šára and R. Bajcsy. Fish-scales: representing fuzzy manifolds. In *ICCV*, pages 811–817, 1998.
- [15] K. Rektorys and kol. *Přehled užité matematiky*. Prometheus, 1996.
- [16] R. Schneider and L. Kobbelt. Geometric fairing of irregular meshes for free-form surface design. *Computer Aided Geometric Design*, 18(4):359–379, 2001.
- [17] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin. A comparison of gaussian and mean curvatures estimation methods on triangular meshes. In *IEEE International Conference on Robotics and Automation 2003 (ICRA '03)*, pages 1021–1026, 2003.
- [18] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, 1990.
- [19] H. Yagou, Y. Ohtake, and A. Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Geometric Modeling and Processing, 2002*, pages 124–131, 2002.

Dodatek A

Implementace algoritmu vyhlazování uzavřené křivky

Kapitola popisuje implementaci algoritmu vyhlazování uzavřené křivky. Implementace algoritmu je vytvořena pro program Matlab. Funkce pro vyhlazování křivky se jmenuje `closedCurveFairing`.

A.1 Vyhlazení křivky

V této kapitole ukážeme, jak lze v Matlabu použít funkci `closedCurveFairing` k vyhlazení křivky.

Příklad použití funkce:

```
%Načtení proměnných. V tomto případě matice,  
%která obsahuje vrcholy křivky.  
curveVariables = 'dvanactiuhelnikRand';  
load(curveVariables);  
  
%Zacyklení křivky kvůli vykreslení.  
Z = [U U(:,1)];  
%Vykreslení vstupní křivky (modře).  
handle = line(Z(1,:),Z(2,:),Z(3,:));  
set(handle,'LineStyle','-', 'Marker','+', 'color',[0 0 1],  
      'LineWidth',1.0);  
axis equal  
hold on  
  
%Nastavení parametru pro funkci vyhlazování křivky.  
max_iter = 500;  
lambda = 0.1;  
unif_lambda = 0.01;
```



```

%Volání funkce pro vyhlazení křivky. Parametry sum, switch,
%unif_switch a epsilon nejsou nastaveny. Budou použity
%předdefinované hodnoty.

[U1] = closedCurveFairing(U,[max_iter, lambda, unif_lambda]);

%Zacyklení vyhlazené křivky.
Z1 = [U1 U1(:,1)];
%Vykreslení vyhlazené křivky (červeně).
handle = line(Z1(1,:),Z1(2,:),Z1(3,:));
set(handle,'LineStyle','-','Marker','+','color',[1 0 0],
      'LineWidth',1.0);

%Popis křivek v obrázku.
legend('vstup','vyhlazeno');

```

A.2 Vstupní a výstupní parametry funkce

V této kapitole popíšeme vstupní a výstupní parametry funkce `closedCurveFairing` a jejich strukturu.

Volání funkce `closedCurveFairing`:

```

[U1] = closedCurveFairing(U,
    [max_iter, lambda, unif_lambda, sum,
    switch, unif_switch, epsilon])

```

Podrobně popíšeme vstupní parametry funkce:

- U (povinný parametr)
Posloupnost vrcholů křivky, která má být vyhlazena. Vcholy jsou zadány maticí $M \times N$, kde $M = 3$ a N je počet vrcholů křivky. Jeden sloupec matice tedy obsahuje souřadnice $[x, y, z]$ vrcholu křivky. Křivka nesmí být zacyklena (první vrchol křivky nesmí být zopakován v posledním sloupci matice) !

Následující parametry jsou nepovinné. Lze tedy zapsat jen []. Parametry pak budou nastaveny na předdefinované hodnoty. Možné je také vynechat jen některé z parametrů. To však musí být provedeno tak, že jsou vynechány i parametry, které v parametrech následují.

- `max_iter` Maximální počet iterací pro vyhlazování křivky. V jedné iteraci je proveden posun všech vrcholů křivky. Celkový počet iterací je spočítán jako součet iterací obou kritérií, které jsou provedeny v průběhu vyhlazování. Předdefinovaná hodnota je 100. Parametr je podmínkou pro ukončení algoritmu.

- **lambda**
Parametr je použit k výpočtu velikosti nové křivosti. Předdefinovaná hodnota je 0.1.
- **unif_lambda**
Parametr je použit k výpočtu velikosti nové neuniformity. Předdefinovaná hodnota je 0.01. Pokud je zadán parametr **lambda** a parametr **unif_lambda** zadán není, pak je hodnota **unif_lambda** nastavena na hodnotu **lambda**.
- **sum**
Minimální průměrný posun vrcholů křivky. Tento parametr je v implementaci vynásoben průměrnou délkou hrany. Takto získaná konstanta je podmínkou pro ukončení algoritmu. Předdefinovaná hodnota je 0.0001.
- **switch** a **unif_switch**
Algoritmus vyhlazování pracuje **switch** iterací s kritériem křivosti a poté **unif_switch** iterací s kritériem neuniformity. Tento postup je opakován, dokud není splněna některá z ukončovacích podmínek. Pokud je zadán jen **switch**, parametr **unif_switch** je nastaven na hodnotu parametru **switch**. Pokud parametry zadány nejsou, hodnoty jsou nastaveny na 50.
- **epsilon**
Parametr slouží k porovnání velikosti křivosti na nulu. Předdefinovaná hodnota je 0.0001.

Výstupní parametr **U1** obsahuje vrcholy vyhlazené křivky. Struktura parametru je shodná se strukturou parametru **U** a pořadí vrcholů je shodné.

Dodatek B

Implementace algoritmu vyhlazování povrchu

Kapitola popisuje implementaci algoritmu vyhlazování povrchu. Implementace algoritmu je vytvořena pro program Matlab. Funkce pro vyhlazování uzavřeného povrchu a povrchu s hranicí se jmenuje `surfaceFairing`.

B.1 Vyhlazení povrchu

V této kapitole ukážeme, jak lze v Matlabu použít funkci `surfaceFairing` k vyhlazení povrchu.

Příklad použití funkce:

```
%Načtení proměnných. V tomto případě matice,  
%která obsahuje vrcholy povrchu a matice, která  
%popisuje trojúhelníky.  
SurfaceVariables = 'kouleRand';  
load(SurfaceVariables);  
  
%Vykreslení vstupního povrchu.  
trisurf(T',V(1,:),V(2,:),V(3,:));  
axis equal;  
hold on  
  
%Nastavení parametrů pro funkci vyhlazování povrchu.  
S_max_iter = 500;  
%Povrch nemá hranici, nebude se tedy vyhlazovat.  
C_max_iter = 0;  
S_lambda = 0.01;  
S_unif_lambda = 0.01;
```

```

%Volání funkce pro vyhlazení povrchu. Parametry S_sum, C_sum,
%S_switch, C_switch, epsilon, S_curv_switch, S_unif_switch,
%S_end_way, C_curv_switch, C_unif_switch
%a special_lambdas nejsou nastaveny.
%Budou použity předdefinované hodnoty.

```

```

V1 = surfaceFairing(V,T-1,
    [S_max_iter,C_max_iter],[S_lambda,S_unif_lambda]);

```

```

%Posunutí obrázku v souřadnicích kvůli vizualizaci.
V1(1,:) = V1(1,:) + 3;
%Vykreslení vyhlazeného povrchu.
trisurf(T',V1(1,:),V1(2,:),V1(3,:));
title('Vstup<--->Vyhlazeno','FontSize',14);
axis equal;

```

Uvedeme ještě jeden příklad nastavení parametrů a volání funkce v případě, že má být vyhlazován povrch s hranicí:

```

%Nastavení parametrů pro funkci vyhlazování povrchu s hranicí.
%Počet iterací pro vyhlazení vnitřních vrcholů povrchu.
S_max_iter = 500;
%Počet iterací pro vyhlazení hranice.
C_max_iter = 500;
%Ostatní parametry.
S_lambda = 0.01;
S_unif_lambda = 0.01;
C_lambda = 0.1;
C_unif_lambda = 0.01;
S_sum = 0.00001;
C_sum = 0.00001;

```

```

%Vyhladit nejdříve na 500 iterací hranici a poté na 500
%iterací vnitřní vrcholy povrchu.
S_switch = S_max_iter;
C_switch = C_max_iter;

```

```

fprintf('>>>> Vyhlazování ... \n');

```

```

%Volání funkce pro vyhlazení povrchu s hranicí.
%Parametry epsilon, S_curv_switch, S_unif_switch,
%S_end_way, C_curv_switch a C_unif_switch
%a special_lambdas nejsou nastaveny.

```

%Budou použity předdefinované hodnoty.

```
fprintf('>>>> Výstup algoritmu: \n');  
V1 = surfaceFairing(V,T-1,  
    [S_max_iter,C_max_iter,S_sum,C_sum,S_switch,C_switch],  
    [S_lambda,S_unif_lambda],  
    [C_lambda,C_unif_lambda]);
```

B.2 Vstupní a výstupní parametry funkce

V této kapitole popíšeme vstupní a výstupní parametry funkce `surfaceFairing` a jejich strukturu.

Volání funkce `surfaceFairing`:

```
[V1] = surfaceFairing(V,T,  
[S_max_iter,C_max_iter,S_sum,C_sum,S_switch,C_switch,epsilon],  
[S_lambda,S_unif_lambda,S_curv_switch,S_unif_switch,S_end_way],  
[C_lambda,C_unif_lambda,C_curv_switch,C_unif_switch],  
special_lambdas)
```

Vstupní parametry, které jsou označeny písmenem S náleží k vyhlazování vnitřních vrcholů povrchu a parametry označené písmenem C náleží k vyhlazování hranice (křivky).

Vstupní parametry

Podrobně popíšeme vstupní parametry funkce.

Parametry popisující povrch (povinné parametry):

- V
Vrcholy povrchu, který má být vyhlazen. Vrcholy jsou zadány maticí $M \times N$, kde $M = 3$ a N je počet vrcholů povrchu. Jeden sloupec matice tedy obsahuje souřadnice $[x, y, z]$ vrcholu povrchu.
- T
Trojúhelníky povrchu, který má být vyhlazen. Trojúhelníky jsou zadány maticí $I \times J$, kde $I = 3$ a J je počet trojúhelníku povrchu. Prvky matice jsou indexy vrcholů z parametru V. První vrchol v parametru V musí být indexován nulou!

Obecné parametry pro algoritmus jsou

```
[S_max_iter,C_max_iter,S_sum,C_sum,S_switch,C_switch].
```

Tyto parametry jsou nepovinné. Lze tedy zapsat jen []. Parametry pak budou nastaveny na předdefinované hodnoty. Možné je také vynechat jen některé z parametrů. To však musí být provedeno tak, že jsou vynechány i parametry, které v obecných parametrech následují.

- **S_max_iter**
Maximální počet iterací pro vyhlazování povrchu. V jedné iteraci je proveden posun všech vnitřních vrcholů povrchu. Celkový počet iterací je spočítán jako součet iterací obou kritérií, které jsou provedeny v průběhu vyhlazování. Předdefinovaná hodnota je 500. Parametr je podmínkou pro ukončení algoritmu.
- **C_max_iter**
Parametr je analogický k **S_max_iter**.
- **S_sum**
Minimální průměrný posun vnitřních vrcholů povrchu. Tento parametr je vynásoben průměrnou délkou hrany. Takto získaná konstanta je podmínkou pro ukončení algoritmu. Předdefinovaná hodnota je 0.00001.
- **C_sum**
Parametr je analogický k **S_sum**. Pokud není parametr **C_sum** zadán, ale parametr **S_sum** zadán je, potom je hodnota **C_sum** nastavena na hodnotu parametru **S_sum**. Pokud parametry nejsou zadány, hodnota je nastavena na 0.0001.
- **S_switch**
Pokud jsou zadány parametry **S_switch** a **C_switch** algoritmus vyhlazuje **C_switch** iterací hranici a poté **S_switch** iterací povrch (vnitřní vrcholy povrchu). Tento postup je opakován, dokud není splněna některá z ukončovacích podmínek. Pokud parametry zadány nejsou, vyhladí se nejdříve hranice a potom povrch. Pokud je zadán **S_switch** a parametr **C_switch** zadán není, hodnota **C_switch** je nastavena na hodnotu **S_switch**.
- **C_switch**
Viz. parametr **S_switch**.
- **epsilon**
Parametr slouží k porovnání velikosti křivosti na nulu. Předdefinovaná hodnota je 0.0001.

Parametry pro vyhlazování povrchu jsou:

[**S_lambda**, **S_unif_lambda**, **S_curv_switch**, **S_unif_switch**, **S_end_way**]

Tyto parametry jsou nepovinné a lze je zapsat stejně jako obecné parametry.

- **S_lambda**
Parametr je použit k výpočtu velikosti nové křivosti. Předdefinovaná hodnota je 0.01.
- **S_unif_lambda**
Parametr je použit k výpočtu velikosti nové neuniformity. Pokud není parametr **S_unif_lambda** zadán, ale parametr **S_lambda** zadán je, potom je hodnota **S_unif_lambda** nastavena na hodnotu parametru **S_lambda**. Pokud parametry nejsou zadány, hodnota je nastavena na 0.01.
- **S_curv_switch** a **S_unif_switch**
Algoritmus vyhlazování pracuje **S_curv_switch** iterací s kritériem křivosti a poté **S_unif_switch** iterací s kritériem neuniformity. Tento postup je opakován, dokud není splněna některá z ukončovacích podmínek vyhlazování povrchu (vnitřní vrcholy povrchu). Pokud je zadán jen **S_curv_switch**, parametr **S_unif_switch** je nastaven na hodnotu parametru **S_curv_switch**. Pokud parametry nejsou zadány, hodnoty jsou nastaveny na 50.
- **S_end_way**
Pro **S_end_way = 1** je algoritmus ukončen na podmínku sumy (**S_sum**), jen pokud je pracováno s kritériem křivosti. Pokud je **S_end_way = 0** (předdefinovaná hodnota), algoritmus může být ukončen na podmínku sumy při práci s libovolným kritériem.

Parametry pro vyhlazování hranice jsou:

[**C_lambda**, **C_unif_lambda**, **C_curv_switch**, **C_unif_switch**]

Tyto parametry jsou analogické k parametrům pro vyhlazování povrchu.

Parametr **special_lambdas** je vektor, jehož velikost je rovna počtu vrcholů povrchu (včetně hranice). Každá hodnota definuje speciální lambda pro vrchol, který je na stejné pozici v parametru **V** jako **lambda** v parametru **special_lambdas**. Hodnota **lambda** je vynásobena s obecně definovanou **lambda** a tento součin je použit pro difuzi v daném vrcholu.

Výstupní parametr

Popíšeme výstupní parametr funkce.

Parametr **V1** obsahuje vrcholy vyhlazeného povrchu. Struktura je shodná se strukturou parametru **V**.

Vzhledem k tomu, že algoritmus nemění triangulaci sítě, výstupní parametr popisující triangulaci není nutný. Triangulace nově vzniklého povrchu je tedy shodná s triangulací povrchu, který je vstupem algoritmu.