VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

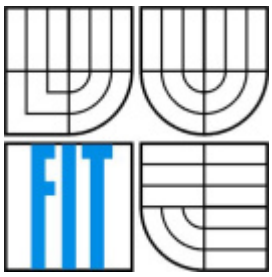
PŘEVOD TROJÚHELNÍKOVÝCH POLYGONÁLNÍCH 3D SÍTÍ NA 3D SPLINE PLOCHY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE BC. ZDENĚK JAHN
AUTHOR

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PŘEVOD TROJÚHELNÍKOVÝCH POLYGONÁLNÍCH 3D SÍTÍ NA 3D SPLINE PLOCHY

TRIANGULAR POLYGONAL 3D MESHES TO 3D SPLINE SURFACES REMESHING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. ZDENĚK JAHN

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. ING. PŘEMYSL KRŠEK, PH.D.

BRNO 2009

Zadání diplomové práce/8527/2008/xjahnz00

Vysoké učení technické v Brně – Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2008/2009

Zadání diplomové práce

Řešitel: **Jahn Zdeněk, Bc.**
Obor: Počítačová grafika a multimédia
Téma: **Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy**
Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s problematikou trojúhelníkových polygonálních 3D sítí a 3D spline plochy.
2. Analyzujte základní možnosti převodu trojúhelníkových polygonálních 3D sítí na 3D spline plochy.
3. Navrhněte algoritmus (vhodně modifikujte existující) pro převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy.
4. Implementujte vybranou část navrženého algoritmu ve vybraném jazyce (C/C++, Java, Python, C#).
5. Zhodnoťte dosažené výsledky a stanovte další vývoj projektu.

Literatura:

- Žara J., Beneš B., Felkel P.: Moderní počítačová grafika. 1. vyd. Praha, Computer press 1998, 448 s., ISBN 80-7226-049-9

Při obhajobě semestrální části diplomového projektu je požadováno:

- Provedení prvních tří bodů zadání.

Podrobné závazné pokyny vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdává v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy., úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kršek Přemysl, doc. Ing., Ph.D.,** UPGM FIT VUT
Datum zadání: 22. září 2008
Datum odevzdání: 26. května 2009

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

V počítačové grafice se můžeme setkat s nestrukturovanými trojúhelníkovými 3D sítěmi, které nejsou příliš použitelné pro zpracování kvůli své nepravidelnosti. V těchto případech může vyvstat potřeba převést danou 3D síť na vhodnější reprezentaci. Vhodnou alternativou může být určitý druh 3D spline plochy, která zavádí strukturu v podobě sítě řídících bodů a pro další zpracování je tedy mnohem vhodnější. V rámci převodu, který je popisován v této práci, se nejdříve vytvoří quadrilaterální 3D síť, jejíž struktura je pravidelná, ale především koresponduje se strukturou sítě řídících bodů výsledné 3D spline plochy. Tuto quadrilaterální 3D síť lze následně uložit a použít v určitých modelovacích aplikacích pro vytvoření 3D spline plochy, konkrétně tedy T-spline plochy.

Abstract

In computer graphics we can handle unstructured triangular 3D meshes which are not too usable for processing through their irregularity. In these situations it occurs need of conversion that 3D mesh to more suitable representation. Some kind of 3D spline surface can be proper alternative because it institutes regularity in the form of control points grid and that's why it is more suitable for next processing. During conversion, which is described in this thesis, quadrilateral 3D mesh is constructed at first. This mesh has regular structure but mainly the structure corresponds to structure of control points grid of resulting 3D spline surface. Created quadrilateral 3D mesh can be saved and consequently used in specific modeling applications for T-spline surface creation.

Klíčová slova

Vrchol, trojúhelník, quadrilaterál, polygon, 3D síť, 3D spline plocha, NURBS, T-Spline, skalární pole, vektorové pole, gradientní vektor, izoparametrický vektor, křivost, křivost polygonální sítě, plovoucí linky, distanční funkce, kostra, přesíťování, triangulace, teselace, metoda rozvodí, formát STL, formát OBJ.

Keywords

Vertex, triangle, quadrilateral, polygon, 3D mesh, 3D spline surface, NURBS, T-Spline, scalar field, vector field, gradient vector, isoparametric vector, curvature, polygonal mesh curvature, flow lines, distance function, skeleton, remeshing, triangulation, tessellation, watersheds algorithm, STL format, OBJ format.

Citace

Jahn, Z.: *Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy*. Diplomová práce, Brno, FIT VUT v Brně, 2009.

Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením docenta Krška. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Bc. Zdeněk Jahn
22. května 2009

Poděkování

Velice si vážím spolupráce s docentem Krškem, které se mi dostalo v rámci této diplomové práce a také předchozí práce bakalářské. Tato spolupráce pro mě byla značným přínosem jak po stránce profesní tak osobní.

Obsah

1 Úvod

V počítačové grafice se setkáváme s různými způsoby reprezentace 3D těles. Z těch nejběžnějších mohu jmenovat hraniční reprezentaci, konstruktivní geometrii CSG, dekompoziční modely nebo implicitní plochy. Každá z těchto forem může nalézt své uplatnění ve specifických aplikacích, ale obecně se žádná z nich nehodí na všechny účely.

Např. CSG je neocenitelné pro popis složitě strukturovaných těles v CAD aplikacích. Komplexní modely mohou být jednoduše upravovány přidáváním geometrických primitiv, přičemž kompletní model je výsledkem transformací a logických operací dílčích primitiv. Pro popis modelu se používá stromová struktura, kterou lze upravovat na libovolné úrovni tak, že je zbývající část automaticky aktualizována odpovídajícím způsobem. K tomu je možné vytvářet logické celky v podobě hladin[1] a sestav[2], což umožňuje vytvářet složité komplexní modely s uchováním detailů libovolné úrovně. CSG je ve své samotné podstatě předurčeno k technickému použití. Zobrazení modelů však není triviální a proto se obvykle převádějí na hraniční reprezentaci, která je pro tento účel vhodnější.

Dekompoziční modely jsou pravděpodobně nejvhodnějším způsobem uchování reálných 3D dat, jejichž zdrojem může být např. geologické mapování nebo třeba nějaké medicínské zařízení. Jako nositelé informace se používají tzv. voxely[3], které mohou být uloženy buďto v pravidelné mřížce nebo ve struktuře, která adaptivně reflektuje hustotu rozložení prostorových dat a redukuje tak množství potřebné paměti. Příkladem takové struktury může být oktalový strom (*angl. octree*). Dekompoziční modely jsou výborné pro zachycení kompletní informace o 3D tělesu. Jakákoliv změna tvaru tělesa je ale náročná a stejně tak jako CSG ani dekompoziční modely se příliš nepoužívají pro přímé zobrazení. Z těchto důvodů se opět používá převod na hraniční reprezentaci.

Již několikrát zmiňovaná hraniční reprezentace zachycuje informaci pouze o povrchu tělesa, zatímco vnitřek je dutý. Kromě geometrie tělesa lze definovat různé povrchové vlastnosti jako textury a materiály. Pro zobrazení existuje řada metod, z nichž některé jsou dostatečně výkonné pro práci v reálném čase, zatímco jiné umožňují dosažení vysokého stupně fotorealističnosti.

Hraniční reprezentací rozumíme buďto polygonální modely, které jsou popsány sítí vrcholů, hran a stěn (obvykle trojúhelníky, někdy čtyřúhelníky nebo polygony), nebo 3D spline plochy, které používají síť řídících bodů, a povrch tělesa lze odvodit podle určité matematické definice. Výhodou 3D spline ploch oproti polygonálním modelům je lepší aproximace povrchu tělesa a často také intuitivnější úpravy tvaru. Zatímco polygonální modely pouze zachycují geometrii tělesa bez hlubších vztahů dílčích primitiv, 3D spline plochy vytvářejí vztah mezi řídícími body a odpovídající částí povrchu. Změna polohy nebo určitého parametru řídícího bodu má vliv na jeho okolí, přičemž

[1] V CAD aplikacích se používá jako úroveň pohledu podle sledovaných objektů nebo parametrů.

[2] Sestavy slouží pro popis samostatných celků, mezi kterými je možné definovat závislosti.

[3] Voxel je 3D ekvivalent pro 2D pixel (obrazový bod).

konkrétní efekt závisí na typu 3D spline plochy. V současnosti nejčastěji narazíme na NURBS nebo T-spline plochy, z historického hlediska můžeme zmínit bikubické nebo B-spline plochy.

1.1 Motivace

Často je nutné převést model z jedné reprezentace na jinou. Např. model lebky získaný určitým medicínským zařízením bude pravděpodobně uložen v podobě dekompozičního modelu. Tato forma je vhodná pro uchování kompletní informace o tělese. Pokud však budeme chtít s tímto modelem dále pracovat, může vyvstat potřeba získat hraniční reprezentaci např. v podobě určitého druhu 3D spline plochy. Pro převod na polygonální reprezentaci lze využít algoritmus marching cubes, jehož výsledkem je nestrukturovaná trojúhelníková síť (více o algoritmu marching cubes viz [Owen99]). Tato síť je sice takřka nepoužitelná pro jakékoliv efektivní úpravy tvaru, poskytuje nám ale určitou cestu k převodu na 3D spline plochu.

Problematice převodu nestrukturovaných trojúhelníkových 3D sítí na 3D spline plochy jsem se věnoval již ve své bakalářské práci a také semestrálním projektu, viz [Jahn07] a [Jahn09]. Analyzoval jsem dvě metody vhodné pro převod trojúhelníkové 3D sítě na quadrilaterální, která představuje mezikrok při vytváření 3D spline plochy, a částečně jsem implementoval jednu z nich. Cílem této práce, která vychází z výsledků mé bakalářské práce, bylo vylepšit některé aspekty převodu na quadrilaterální 3D síť a hlavně dokončit vytváření kýžené 3D spline plochy.

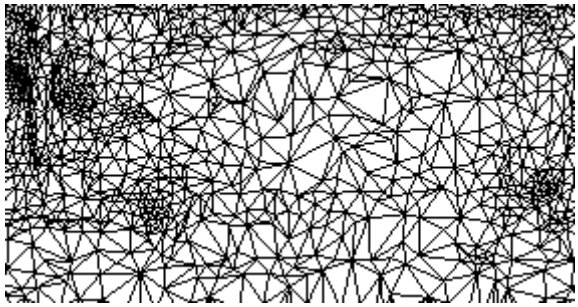
1.2 Struktura práce

V kapitole 2 – Teoretický rozbor nejdříve popíšu teoretická východiska metody pro převod trojúhelníkové sítě na quadrilaterální, následně uvedu podklady pro vytvoření kostry modelu, jejímž účelem je vylepšit kvalitu výsledného modelu a následně se budu věnovat problematice 3D spline ploch. V následující kapitole 3 – Návrh a implementace nastíním implementaci aplikace, která realizuje převod trojúhelníkové sítě do formy použitelné pro vytvoření 3D spline plochy. V kapitole 4 – Výsledky popíšu dosažené výsledky a v kapitole 5 – Závěr tyto výsledky zhodnotím.

2 Teoretický rozbor

2.1 Nestrukturovaná trojúhelníková 3D síť

Nestrukturovaná trojúhelníková síť plně postačuje pro některé účely, jako je kompletní popis povrchu tělesa nebo zobrazení pomocí grafického akcelerátoru. Pro řadu aplikací je však naprosto nepoužitelná. Hlavní nevýhodou je nerovnoměrné rozložení trojúhelníků, které nemá žádnou logickou strukturu.



Obrázek 2-1: *Výřez nestrukturované trojúhelníkové sítě.*

Převod nestrukturované trojúhelníkové 3D sítě na 3D spline plochu lze rozdělit do dvou kroků. Vstupní trojúhelníkovou síť nejdříve převedeme na síť quadrilaterální, která nám následně poslouží k vytvoření 3D spline plochy.

2.2 Quadrilaterální 3D síť

Quadrilaterální síť lze neformálně definovat jako polygonální 3D síť skládající se převážně ze čtyřúhelníkových plošek, které se tvarově blíží čtverci nebo obdélníku. Za předpokladu, že síť obepíná celé těleso a tedy nemá žádné okraje a připustíme přítomnost tzv. T-zakončení[1] *(angl. T-junctions)*, bude každá ploška na každé ze čtyř stran sousedit alespoň s jednou další ploškou. Quadrilaterální síť je asi nejpřirozenější polygonální 3D síť z hlediska lidského chápání, takže je mnohem vhodnější pro úpravy než nestrukturovaná trojúhelníková síť. Pro nás důležitou vlastností je však to, že taková síť tvoří obdélníkovou mřížku, kterou lze použít pro konstrukci 3D spline plochy.

Quadrilaterální síť, kterou potřebujeme vytvořit, musí co nejlépe zachovat tvar tělesa, pouze se bude skládat ze čtyřúhelníků narozdíl od původních trojúhelníků. Tento převod můžeme nazývat přesíťování (*angl. remeshing*[2]). Již v rámci bakalářské práce jsem zvažoval použití jedné přesíťovací metody z následujícího výběru:

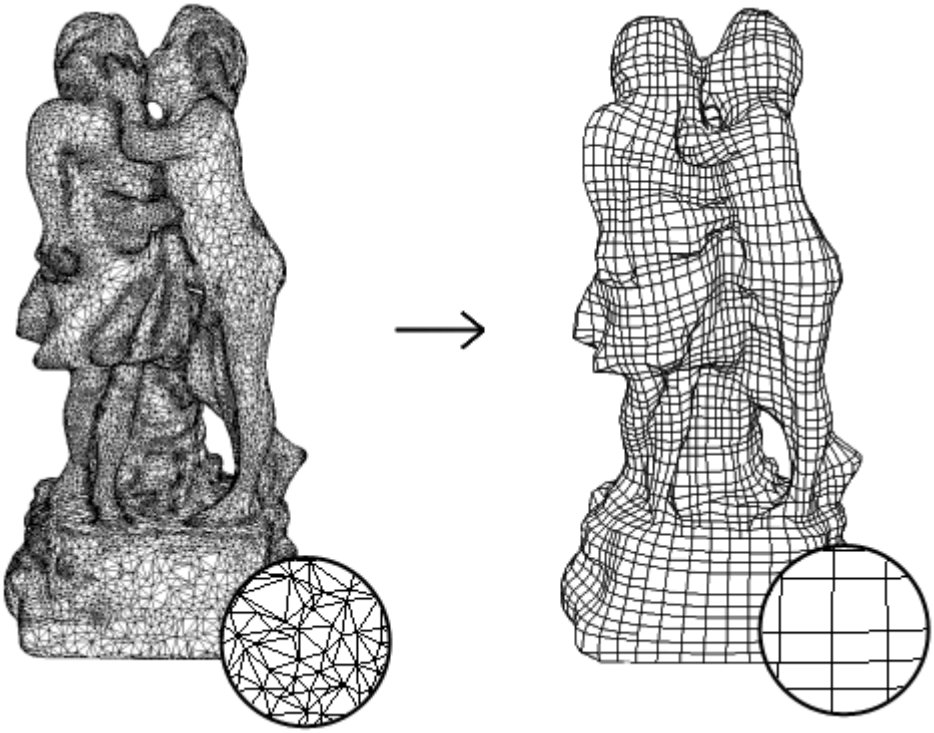
1. Metoda *Anisotropic Polygonal Remeshing* (informace jsem čerpal z [Alliez03]).

[1] T-zakončení je místo, kde hrana mezi dvěma ploškami končí uvnitř jiné hrany.

[2] Anglické slovo *remeshing* lze chápat jako přesíťování nebo přeneseně také převzorkování.

2. Metoda *Harmonic Functions for Quadrilateral Remeshing of Arbitrary Manifolds* (informace jsem čerpal z [Garland05]).

Společným rysem obou metod je vytváření sítě linek pokrývajících původní síť, ze které jsou extrahovány vrcholy, hrany a plošky nové sítě. Významně se ale liší v konstrukcí této sítě. Zatímco první metoda používá pro určení sítě linek tenzorové[1] pole počítané pro blízké okolí jednotlivých vrcholů, druhá metoda používá dvojici vektorových polí určených podle skalárního pole, které se konstruuje pro celou síť chápanou jako jeden celek.



Obrázek 2-2: *Vlevo je vidět nestrukturovaná trojúhelníková 3D síť, vpravo ekvivalentní quadrilaterální síť (obrázek jsem převzal z [Garland05]).*

V rámci bakalářské i diplomové práce jsem se věnoval druhé z uvedených dvou metod, která bude v následujících kapitolách vysvětlena podrobněji. Tato metoda pracuje s manifold sítěmi libovolných tříd *(angl. genus)*, je zcela obecná, efektivní a flexibilní (v souladu s tím co uvádí [Garland05]). To znamená, že si poradí se sítí jakékoliv topologie a vždy podá uspokojivý výsledek v dohledném čase. Převádí nestrukturovanou trojúhelníkovou na strukturovanou quadrilaterální 3D síť, přičemž umožňuje kontrolovat hustotu linek a ovlivnit tvar výsledné sítě. Převod probíhá v několika krocích. Každý krok funguje jako samostatná operace s konkrétním výsledkem závislá na výsledcích předchozích kroků. Díky tomu je možné opakovat některou fázi s různými parametry a dolaďovat výslednou síť.

[1] Tenzor je matematické zobecnění vektoru a skaláru.

2.2.1 Algoritmus

Uvádím přehled celého algoritmu přeložený z originálu [Garland05]:

Předpokládejme, že máme danou vstupní trojúhelníkovou manifold síť $M = (V, F)$ *složenou z množiny vrcholů V a množiny trojúhelníků F. Každému vrcholu i je přiřazena souřadnice* $x_i \in R^3$ *ve 3D euklidovském prostoru. Požadujeme, aby byla síť manifold, jinak může být libovolných druhů a nepředpokládáme žádná omezení, co se počtu hraničních křivek týče. Na nejvyšší úrovni se tento přesíťovací algoritmus skládá z následujících základních kroků:*

1. *Výpočet po částech lineárního skalárního pole* $u : V \to R$ *nad vrcholy M.*
2. *Na základě u odvození dvou ortogonálních po částech konstantních tangentních vektorových polí* $g1, g2 : F \to R^3$ *nad ploškami (angl. faces) M.*
3. *Sestavení sítě polygonů nad povrchem konstrukcí integrálních linek vektorových polí* g_1 *a* g_2*.*
4. *Eliminace tzv. T-zakončení.*

Výsledkem je neuniformní[1] *(angl. non uniform) převážně quadrilaterální síť pokrývající vstupní manifold M.*

2.2.2 Skalární pole

Prvním krokem je konstrukce *harmonického skalárního pole*, které splňuje podmínku

$$\Delta u = \nabla^2 u = 0, \tag{2-1}$$

kde Δ je Laplace-Beltramiho operátor, který je možné podmínit Dirichletovými mezními podmínkami, takovými, že vrcholům z množiny $C \subset V$ nazývaným omezené vrcholy, je přiřazena hodnota

$$u_i = c_i \text{ pro všechna } i \in C. \tag{2-2}$$

Takové skalární pole se mění plynule ve všech směrech a jediné extrémy se nacházejí v omezených vrcholech. Počet extrémů je libovolný, pro relevantní výsledek je však nutné definovat alespoň po jednom minimu a maximu. Vhodným rozmístěním extrémů lze ovlivnit tvar skalárního pole a tedy i výsledné sítě, což bude podrobněji rozebráno v kapitole 2.2.4.

Skalární pole je spojité, tato metoda však předkládá diskrétní způsob řešení. Laplaceho rovnost se převede do tvaru

$$\Delta u = -L\vec{u}, \tag{2-3}$$

kde $\vec{u}$ představuje vektor hodnot skalárního pole odpovídajících jednotlivým vrcholům sítě a L je následující matice:

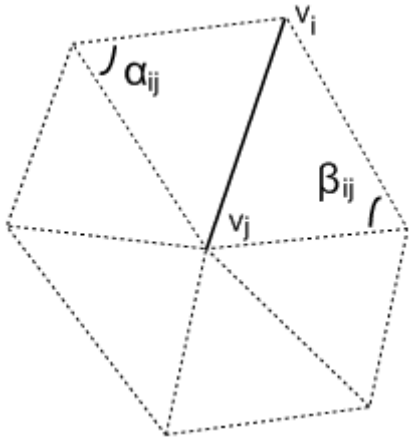
$$L_{ij} = \begin{cases} \Sigma_{<i,k>\in M} w_{ik} & i = j \\ -w_{ij} & <i, j> \in M \\ 0 & jinak \end{cases}. \tag{2-4}$$

[1] Pokud je celek neuniformní (nebo také nehomogenní), potom se skládá z rozdílných prvků.

Podle [Garland04] lze váhu hrany w_{ij} definovat různými způsoby, např. kombinatoricky v závislosti na počtu hran vedoucích k vrcholu, propagováním střední hodnoty nebo váženou hodnotou. [Garland05] uvádí následující metriku:

$$w_{ij} = -\tfrac{1}{2}\left(\cot\alpha_{ij} + \cot\beta_{ij}\right), \tag{2-5}$$

kde α_{ij} a β_{ij} jsou protilehlé úhly oproti hraně v_iv_j.



Obrázek 2-3: *Úhly pro určení váhy hrany.*

Řešením lineární soustavy 2-3 však nezískáme potřebný výsledek. Do soustavy potřebujeme vnést extrémy, ke kterým pole konverguje. Požadovanou soustavu tedy definujeme takto:

$$A\vec{u} = \vec{b}\,, \tag{2-6}$$

kde

$$A_{ij} = \begin{cases} \delta_{ij} \,\middle|\, i \in C \\ L_{ij} \,\middle|\, i \notin C \end{cases} \qquad b_i = \begin{cases} c_i \,\middle|\, i \in C \\ 0 \,\middle|\, i \notin C \end{cases}.$$

V tomto okamžiku máme lineární soustavu pro výpočet skalárního pole. Kroneckerova delta funkce δ_{ij} zařídí, že hodnoty všech extrémů zůstanou výpočtem nezměněny (více o delta funkci viz [Weisstein07]). Výsledné pole potom k těmto extrémům konverguje.

Výhoda tohoto postupu spočívá v tom, že není potřeba síť parametrizovat. Díky tomu může být metoda zcela obecná pro všechny manifold sítě a navíc se vyhneme složitým integracím. Skalární pole představuje harmonickou, po částech lineární funkci. Hodnoty ve vrcholech se dají spočítat diskrétně řešením lineární soustavy 2-6, v ostatních bodech potom lineární interpolací hodnot ve vrcholech odpovídajícího trojúhelníka.

2.2.3 Kritické body

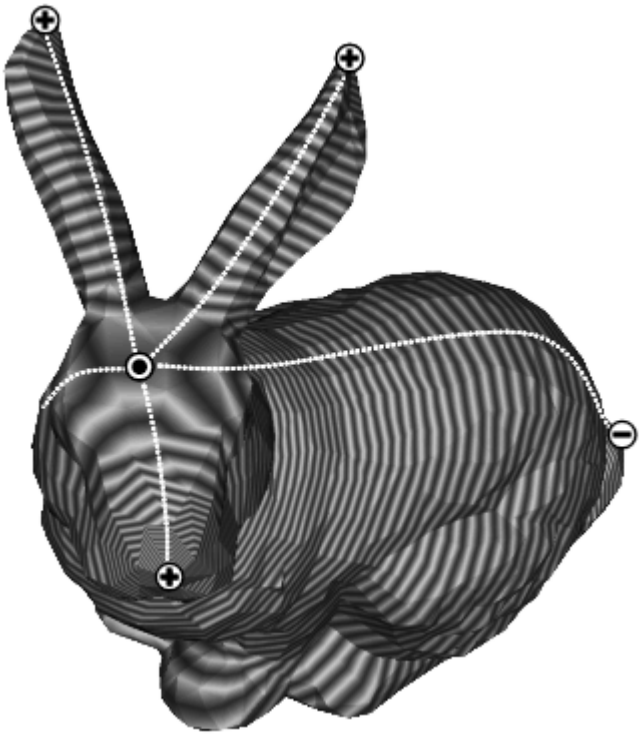
[Garland05] uvádí, že Eulerova charakteristika popisuje třídu manifoldu v závislosti na počtu jeho vrcholů, hran a plošek *(angl. vertices, edges, faces)*:

$$\chi = 2 - 2g = |V| - |E| + |F|\,. \tag{2-7}$$

Pomocí Morseho teorie lze z Eulerovy charakteristiky odvodit vztah popisující počet kritických bodů, tj. počet minim, maxim a tzv. *sedlových bodů (angl. saddle points)* skalárního pole:

$$\chi = \eta_{\min} + \eta_{\max} - \eta_{saddle}\ . \tag{2-8}$$

Sedlové body jsou stacionární, ale neextrémní body. Ze vztahu 2-8 vyplívá, že čím více extrémů bude mít skalární pole, tím více bude mít i sedlových bodů. Při vytváření sítě plovoucích linek (viz kapitola 2.2.6) jsou sedlové body problematická místa. Z toho důvodu je vhodné minimalizovat jejich počet a proto i počet extrémů. Na druhé straně určitý počet extrémů je nezbytný. Pro smysluplný výsledek alespoň po jednom minimu a maximu, vhodný počet ale závisí na konkrétním modelu.

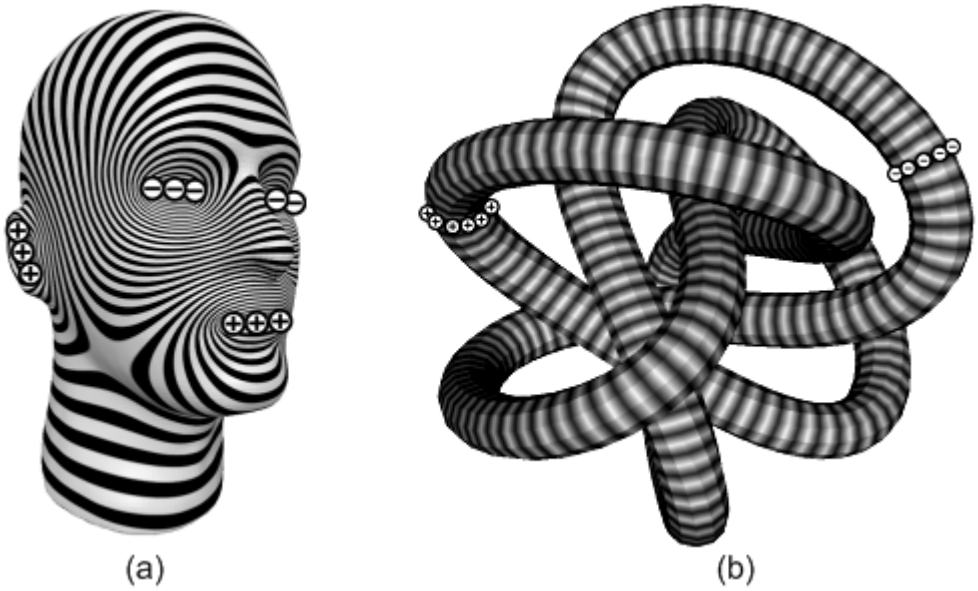


Obrázek 2-4: *Sedlový bod (kolečko s tečkou) a extrémy (kolečka s +/-) skalárního pole na modelu zajíce. Skalárního pole je v sedlovém bodě lokálně stacionární, v širším okolí v některých směrech roste, v jiných klesá.*

2.2.4 Omezené vrcholy

Omezené vrcholy odpovídají extrémům skalárního pole. Vhodným výběrem těchto vrcholů, který může být ponechán na uživateli, lze upravit tvar skalárního pole a tedy i tvar výsledné sítě. Cílem je získat rovnoměrně rozprostřené skalární pole bez výrazných změn, tak aby výsledná síť co nejlépe pokryla celé těleso. Z toho důvodu je vhodné umístit malé množství kladných extrémů na jednom konci tělesa a malé množství záporných extrémů na opačném konci.

Co se týče přesného výběru vrcholů, obvykle se vybírají špičky výrazných výčnělků, pokud jsou přítomny, čímž se zajistí jejich nejlepší pokrytí. V případě, že nelze nalézt vhodná místa, ke kterým by mělo skalární pole konvergovat, vhodné pokrytí lze zajistit výběrem skupin vrcholů dohromady tvořících křivku nebo uzavřenou smyčku. Příkladem může být obrázek 2-5, který znázorňuje takové situace. Obrázek 2-5a ukazuje podélný výběr pro lepší pokrytí očí, úst a uší na modelu lidské hlavy. Na obrázku 2-5b je potom vidět prstencový výběr vrcholů kolem povrchu smyčky, díky kterému skalární pole rovnoměrně pokrývá její celý povrch. Popisovaná opatření jsou však nezbytná pouze zřídka. U reálných těles obvykle nejsou potřeba, protože u nich lze nalézt dostatek význačných míst.



Obrázek 2-5: *Ilustrace výběru skupin omezených vrcholů (a) pro lepší pokrytí určitých linií, (b) na tělese bez význačných míst.*

2.2.5 Vektorová pole

Na základě toho, jak se mění hodnoty skalárního pole, lze spočítat vektorové pole, které bude určovat směr a velikost změny. Toto pole nazveme *gradientní vektorové pole*. Dále definujeme *izoparametrické vektorové pole*, které je ortogonální ke gradientnímu, a udává tedy směr, pro který má skalární pole neměnnou hodnotu. Tato vektorová pole budou později určovat směr *gradietních* a *izoparametrických plovoucích linek*, které slouží k vytvoření nové sítě (viz kapitola 2.2.6).

Protože je skalární pole po částech lineární a tedy v každém bodě libovolného trojúhelníka se hodnota mění lineárně s konstantním směrem, můžeme pro každý trojúhelník spočítat právě po jednom vektoru obou vektorových polí. Gradientní vektor pro trojúhelník (i, j, k), jehož vrcholy mají souřadnice $x_i, x_j, x_k \in R^3$ a $\vec{n}$ je jeho normálový vektor, určíme jako $g_1 = \nabla u$ řešením lineární soustavy:

$$\begin{bmatrix} x_j - x_i \\ x_k - x_j \\ \vec{n} \end{bmatrix} \begin{bmatrix} \\ \vec{g}_1 \\ \\ \end{bmatrix} = \begin{bmatrix} u_j - u_i \\ u_k - u_j \\ 0 \end{bmatrix}. \tag{2-9}$$

Protože je izoparametrický vektor kolmý ke gradientnímu i normálovému vektoru, spočítá se jako vektorový součin těchto vektorů:

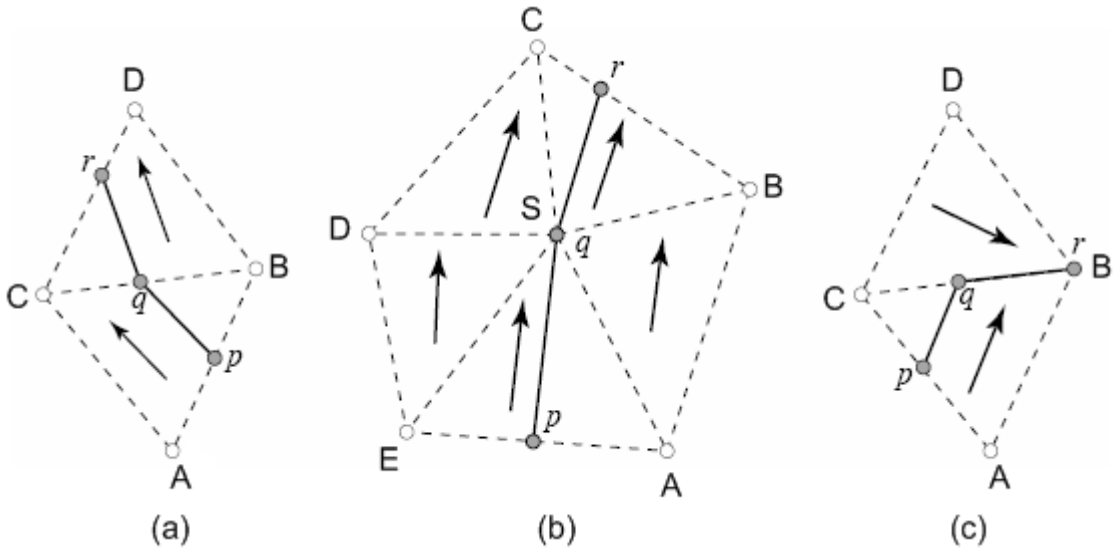
$$\vec{g}_2 = \vec{n} \times \vec{g}_1. \tag{2-10}$$

2.2.6 Plovoucí linky

Plovoucí linky jsou po částech lineární křivky, které sledují původní síť podél jednoho z vektorových polí definovaných v kapitole 2.2.5. Gradientní linky mají vždy počátek a konec, izoparametrické linky mohou také tvořit uzavřené smyčky. Cílem je vytvořit síť linek, ze které by bylo možné sestavit novou quadrilaterální síť (viz kapitola 2.2.10).

Počátek plovoucích linek určují tzv. semínka, jejichž rozmístění bude popsáno v kapitole 2.2.9. Při vytváření každé linky jsou postupně vytvářeny jednotlivé segmenty, které vždy vycházejí z posledního bodu linky, tedy semínka resp. posledního dosaženého uzlu, a pokračují podél příslušného vektorového pole k nejbližší hraně. Linka končí, pokud dosáhne extrému, přiblíží se k jiné lince nebo vytvoří smyčku. Vzdálenost, na jakou se mohou dvě linky přiblížit, určuje distanční funkce (viz kapitola 2.2.7). Během hledání cesty plovoucí linky mohou nastat tyto tři případy:

A. Obvykle počátek i konec segmentu leží na dvou hranách jednoho trojúhelníka. Pokud linka prochází přes trojúhelník *ABC* a aktuální koncový uzel leží na hraně *BC*, následující uzel bude určen jako průsečík s jednou z hran *BD*-*DC* trojúhelníka *BDC*, viz obrázek 2-6a.
B. V některých případech linka prochází vrcholem. Pro nalezení následujícího uzlu je nutné prohledat okolní trojúhelníky a vybrat ten, ve kterém linka postoupí nejdále, jestliže existuje více možností. V případě gradientního toku rozhoduje při výběru možností rozdíl hodnot skalárního pole, v případě izoparametrického toku existuje maximálně jedna možnost. Obrázek 2-6b ukazuje průchod linky přes vrchol *S*, přičemž hrana *BC* představuje jedinou alternativu pro pokračování.
C. Výjimečně nelze nalézt průsečík s žádnou hranou. V takovém případě musí linka pokračovat podél hrany k vrcholu, ke kterému směřuje vektorový tok, viz obrázek 2-6c. Toto nastává v okolí sedlových bodů, přičemž je to jediná situace, kdy linka nepokračuje přímo podél daného vektorového toku.



Obrázek 2-6: *Tři varianty průchodu plovoucí linky přes trojúhelník.*

2.2.7 Distanční funkce

Distanční funkce slouží ke dvěma účelům. Určuje jednak počáteční rozestupy plovoucích linek (viz kapitola 2.2.9) a také minimální vzdálenost, na kterou se může přiblížit jedna linka k jiné. Pokud během vytváření linky klesne vzdálenost koncového uzlu od sousedních linek pod hodnotu udanou distanční funkcí, je linka ukončena. Důležité poznamenat, že se vzdálenost v obou případech určuje podél povrchu tělesa ve směru ortogonálního vektorového pole.

V případě izotropní[1] sítě vystačíme s konstantní funkcí h, v případě anizotropní[2] sítě použijeme dvojici distančních funkci h_1 a h_2 (první pro gradientní, druhou pro izoparametrický tok):

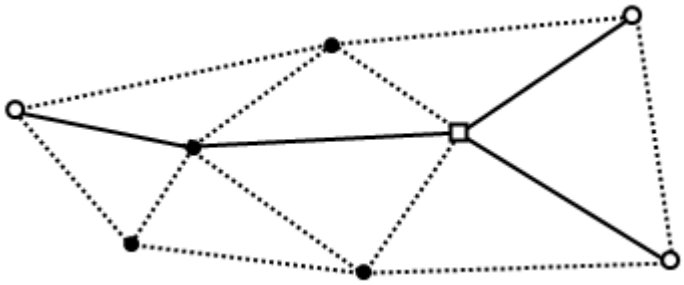
$$h_1 = \frac{h}{1+\alpha \log_{10}\left(1+\kappa_n^2\right)}, \qquad h_2 = \frac{h}{1+\alpha \log_{10}\left(1+\kappa_n^1\right)}. \tag{2-11}$$

κ^1 a κ^2 jsou normálové křivosti povrchu ve směru g_1 resp. g_2 (křivost viz kapitola 2.4.1). Hodnota α udává stupeň anizotropie a bývá obvykle menší než 20. Může být ale i mnohem větší. Pokud se $\alpha = 0$, degradují funkce h_1 a h_2 do konstantního tvaru ($h_1 = h_2 = h$) a tedy výsledná síť bude izotropní.

2.2.8 Vlastnosti

Jako vlastnosti (*angl. features*) jsou souhrnně označovány vrcholy a hrany, které by měly být uchovány. Jsou to vlastně význačné linie, které mají význam na utváření rysů tělesa. Z vlastností můžeme sestavit kostru, jejíž části označujeme:

1. Řetězy (*angl. chains*) – skupina po sobě jdoucích hran.
2. Šipky (*angl. darts*) – vrcholy s jednou hranou (ukončení řetězu).
3. Rohy (*angl. corners*) – vrcholy se třemi a více hranami.



Obrázek 2-7: *Ilustrace kostry – tučné linky jsou řetězy, kolečka šipky a čtverec je roh.*

Kostře se budu podrobněji věnovat v kapitole 2.4.

Během vytváření plovoucích linek (viz kapitola 2.2.6) může linka protínat hranu označenou jako vlastnost. V tom případě musí být uzel plovoucí linky ležící na této hraně použit při vytváření nové sítě (viz kapitola 2.2.10) jako vrchol a také spojen hranou se sousedními uzly ležícími na stejném řetězu.

2.2.9 Semínka plovoucích linek

Semínka určují začátek plovoucích linek. Počáteční semínka by měla být umístěna ve význačných vrcholech, jimiž jsou omezené vrcholy a vrcholy rohů vlastností.

Z počáteční množiny semínek vznikne první sada linek. Semínka dalších linek stejného toku jsou rozmisťována v pravidelných intervalech podél každé plovoucí linky po jejích obou stranách ve vzdálenosti udávané distanční funkcí (viz kapitola 2.2.7). Toto rozmisťování má především význam

[1] U izotropní sítě jsou rozměry plošek nezávislé na pozici v síti.

[2] U anizotropní sítě závisí velikost plošek na lokální křivosti tělesa.

pro anizotropní sítě, ale lze použít i v případě izotropních sítí. Semínka jsou udržována v prioritní frontě, ve které mají nejvyšší prioritu semínka nejvíce vzdálená od linky, vedle které byla vytvořena. Je to z toho důvodu, že linka vycházející ze vzdálenějšího semínka bude pravděpodobně delší, než linka vycházející z méně vzdáleného semínka. V případě izotropní sítě mají všechna semínka stejnou prioritu, proto může být použita fronta bez priority. Vytváření plovoucích linek každého toku končí v okamžiku vyprázdnění fronty semínek.

2.2.10 Konstrukce nové sítě

Výsledkem předchozích kroků je síť plovoucích linek, která se použije pro konstrukci nové polygonální sítě. Průsečíky gradientních a izoparametrických linek tvoří nové vrcholy. Každý vrchol má obvykle čtyři sousední vrcholy. Pouze v místě ukončení plovoucí linky, budou sousední vrcholy tři a v místě ukončení obou linek pouze dva. Vrcholy s jedním nebo žádným sousedním vrcholem by měly být odstraněny, protože nejsou použitelné pro vytvoření polygonu. Takové vrcholy mohou vzniknout pouze průnikem příliš krátkých linek, které se neprotínají s žádnou jinou, a tedy nejsou podstatné pro další zpracování.

Výpočet průsečíků

Plovoucí linky se skládají z lineárních segmentů. Uvnitř každého trojúhelníka sítě můžeme nalézt rovinnou pravoúhlou mřížku tvořenou těmito segmenty, jejíž osy odpovídají gradientnímu a izoparametrickému vektoru vektorových polí v daném trojúhelníku. Protože jak koncové body, tak průsečíky segmentů leží v jedné rovině, můžeme jejich souřadnice do této roviny transformovat. Jestliže gradientní vektor označíme jako $\vec{v}$, izoparametrický jako $\vec{u}$ a definujeme vektor $\vec{p} = P_{xyz} - O$, kde O je počátek roviny a P_{xyz} je libovolný bod v prostoru, můžeme vyjádřit souřadnice bodu P_{xyz} v dané rovině jako

$$P_{rs} = \left[\frac{\vec{u} \cdot \vec{p}}{\vec{u} \cdot \vec{u}}; \frac{\vec{v} \cdot \vec{p}}{\vec{v} \cdot \vec{v}} \right], \tag{2-12}$$

kde *r* a *s* představují osy roviny rovnoběžné s vektory $\vec{u}$ a $\vec{v}$. (Pozn.: souřadnice bodu P_{rs} ve vzorci 2-12 odpovídají vzdálenosti bodu od roviny dané bodem *O* a normálovým vektorem $\vec{u}$ resp. $\vec{v}$, přičemž výpočet lze odvodit z obecné rovnice roviny a parametrického vyjádření jí kolmé přímky.) Naopak pro libovolný bod P_{rs} ležící v rovině lze spočítat jeho souřadnice v prostoru jako

$$P_{xyz} = O + \vec{u} \cdot P_r + \vec{v} \cdot P_s . \tag{2-13}$$

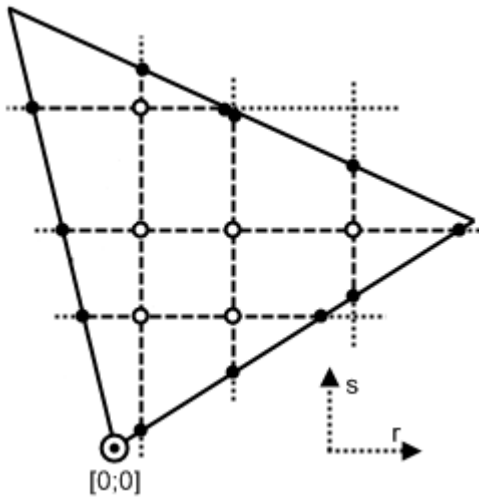
Jestliže má izoparametrický segment koncové body *A*, *B* a gradientní segment koncové body *C*, *D*, můžeme tyto segmenty zapsat parametricky:

$$\left.\begin{array}{l} AB: X = A + (B - A) \cdot p \\ CD: X = C + (D - C) \cdot q \end{array}\right\} p, q \in \langle 0;1 \rangle . \tag{2-14}$$

Protože je úsečka *AB* rovnoběžná s osou *r* a úsečka *CD* s osou *s*, lze ze vztahu 2-14 odvodit, že

$$[p;q] = \left[\frac{C_s - A_s}{B_s - A_s}; \frac{A_r - C_r}{D_r - C_r} \right]. \tag{2-15}$$

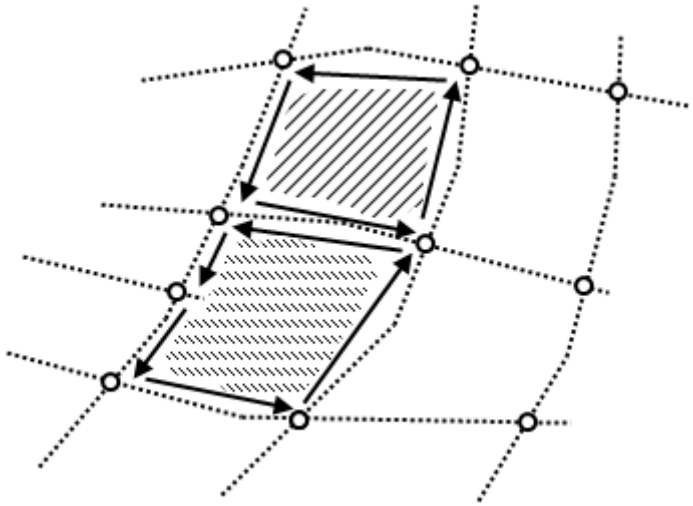
Parametry *p*, *q* nabývají hodnot v intervalu $\langle 0;1 \rangle$, jestliže došlo k průniku uvnitř dané dvojice segmentů, v opačném případě k průniku došlo mimo platnou oblast a průsečík je tedy neplatný.



Obrázek 2-8: *Pravoúhlá rovinná mřížka uvnitř trojúhelníka tvořená segmenty plovoucích linek. Koncové body a průsečíky segmentů lze vyjádřit vzhledem k rovině trojúhelníka s počátkem ve zvoleném vrcholu a osami rovnoběžnými s gradientním a izoparametrickým vektorem.*

Vytvoření polygonů

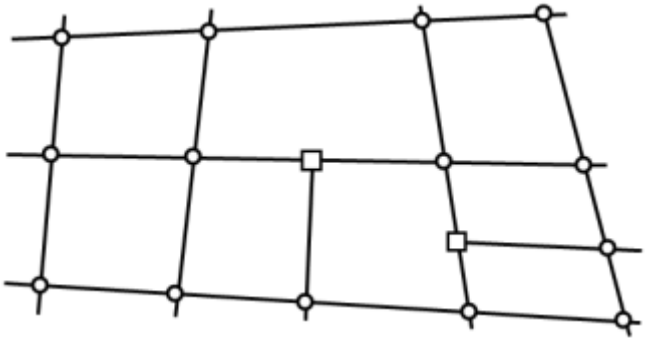
Pro správné napojení vrcholů do hran je nezbytné uchovat informaci o sousednosti podle pořadí vzniku vrcholů na plovoucích linkách. Točivým průchodem hran buďto v hodinovém či opačném smyslu vymezíme jednotlivé polygony. Směr zatočení není rozhodující, ale měl by být stejný pro všechny polygony.



Obrázek 2-9: *Vytvoření hran a polygonů z vrcholů, které leží na průniku plovoucích linek.*

2.2.11 Eliminace T-zakončení

Ve výsledné síti mohou vznikat T-zakončení – konkrétně v místech, ve kterých se plovoucí linky příliš přiblíží a jedna z nich končí. Zakončení linky má potom tvar T, protože hrana vytvořená v tomto místě končí na spojnici mezi dvěma vrcholy. V důsledku toho vznikají polygony s více jak čtyřmi hranami, což může být nežádoucí. Takové defektní polygony se odstraní rozdělením na několik dílčích trojúhelníků.



Obrázek 2-10: *Ilustrace T-zakončení (čtvereček).*

Ve skutečnosti ale tento krok není nezbytný, protože např. T-spline plocha (viz kapitola 2.5.3) umožňuje existenci T-zakončení v sítí řídících bodů a tento efekt je naopak žádoucí.

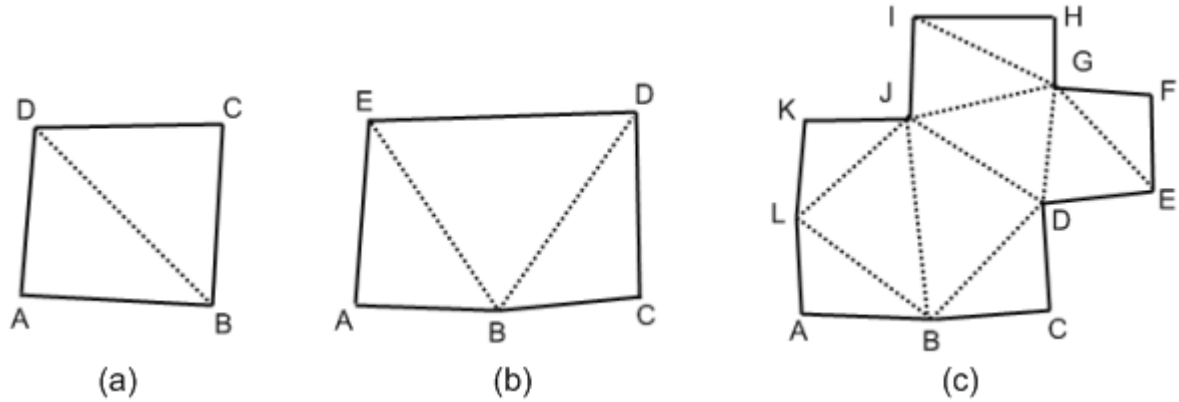
2.3 Teselace polygonů

Pro některé účely může být vhodné převést výslednou quadrilaterální síť opět na trojúhelníkovou. Tento převod lze řešit tak, že budeme každý polygon zvlášť teselovat na trojúhelníky, čímž vytvoříme onu trojúhelníkovou síť. Může se zdát, že se jedná o nesmyslný postup, převádět trojúhelníkovou síť na quadrilaterální a následně opět na trojúhelníkovou. Teselace však neporuší strukturu nové sítě, kterou jsme získali, a např. pro prezentaci quadrilaterální sítě na grafickém výstupu to může být potřeba. Samozřejmě z hlediska převodu na 3D spline plochu je tento krok zbytečný.

2.3.1 Algoritmy

Teselace polygonů je poměrně složitá záležitost a lze k tomuto účelu použít řada algoritmů. Pro výběr nebo návrh vhodného algoritmu je ale nutné uvědomit si, jak vypadají polygony, které chceme teselovat.

1. Vrcholy polygonů libovolné 3D sítě obecně neleží v jedné rovině, což trochu komplikuje situaci. Řada algoritmů funguje pouze v rovině a tím jsou zdánlivě nepoužitelné. Pokud ale nejsou polygony příliš deformované, lze je promítnout do roviny, teselaci provést na jejich průmětu a tu stejnou teselaci aplikovat na původní polygon.
2. Většina polygonů quadrilaterální sítě je podle očekávání čtyřúhelníková (obrázek 2-11a) a lze tedy jednoduše teselovat rozdělením na dva trojúhelníky. Jedinou volbou je, zda polygon *ABCD* rozdělit na dvojici trojúhelníků *ABC-ACD* nebo *ABD-BCD*. V důsledku přítomnosti T-zakončení může síť obsahovat polygony s více vrcholy. Jisté množství takových polygonů má jakoby čtyřúhelníkový tvar, ale některé strany obsahují více vrcholů (obrázek 2-11b). Pro takové polygony by se dala sestavit množina šablon, které by určovaly vhodné teselace pro konkrétní konfigurace. Malé množství polygonů však může mít mnohem komplikovanější tvar a může se skládat z libovolného množství vrcholů (obrázek 2-11c). Tyto je nutné teselovat nějakým robustnějším algoritmem.

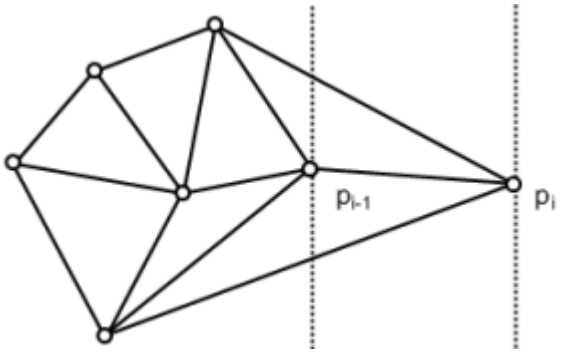


Obrázek 2-11: *Různé druhy plošek a jejich možné teselace.*

Triangulace množiny vrcholů

Pro triangulaci libovolné množiny vrcholů ležících v rovině lze použít řádkovou metodu podle [Sochor96]:

Na počátku seřadíme vrcholy podle jedné osy. Tuto množinu posléze procházíme a dělíme přímkou rovnoběžnou s druhou osou na dvě podmnožiny. V té první je triangulace již dokončená. Předpokládejme, že dělící přímka prochází bodem p_i. Propojíme tento bod s předchozím bodem p_{i-1}. V obou směrech od bodu p_{i-1} až do nalezení tečen konvexního obalu spojujeme body konvexního obalu s bodem p_i.



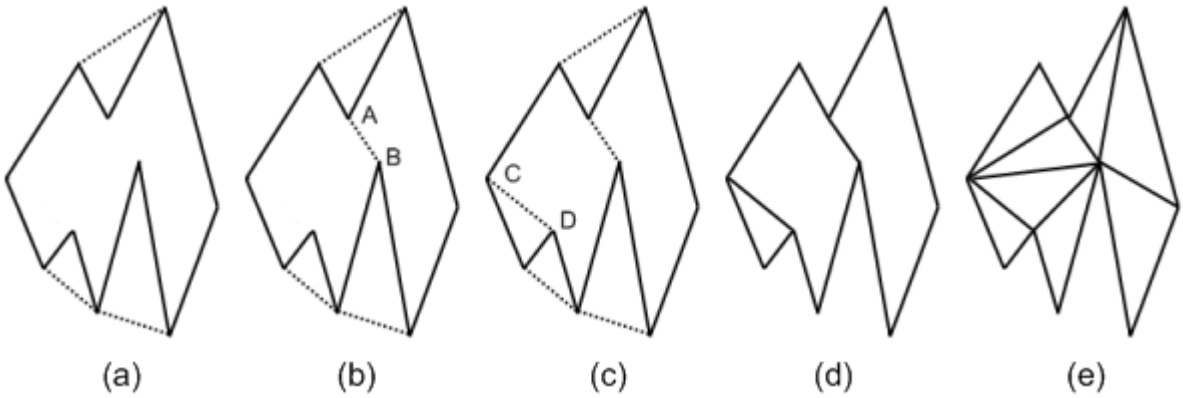
Obrázek 2-12: *Ilustrace řádkové metody pro triangulaci množiny vrcholů.*

Teselace rovinných nekonvexních polygonů

Pro teselaci rovinných nekonvexních polygonů můžeme použít algoritmus ze stejného zdroje, jako předchozí algoritmus:

Polygon rozdělíme na jednoduché subpolygony, omezené dvěma vertikálně monotónními řetězy hran. Vertikálně monotónní řetězy mají nejvýše jeden průsečík s horizontální řádkou. U těchto monotónních polygonů snadno vyřešíme triangulaci postupným krokem po obou monotónních řetězech v čase $O(n)$. Pro vytvoření monotónních polygonů nejprve vypočteme konvexní obal všech vrcholů. Jeho hrany budou zároveň patřit do triangulační sítě obecné úlohy pro množinu úseček. Pak použijeme řádkovou metodu a hledáme ve směru zdola nahoru vrcholy, které jsou lokální minima (jejichž hrany směřují nahoru). Nalezený bod propojíme hranou s nejbližším níže položeným vrcholem již prohlédnutého úseku. Po dosažení nejvyššího vrcholu obrátíme směr prohledávání,

směrem dolů hledáme vrcholy, které jsou lokální maxima. Tyto spojujeme s nejbližšími výše položenými body.



Obrázek 2-13: *Ukázka teselace nekonvexního polygonu: (a) doplnění hran konvexního obalu, (b) doplnění hrany AB při průchodu zdola nahoru, (c) doplnění hrany CD při průchodu shora dolů, (d) rozdělení polygonu na konvexní subpolygony, (e) výsledná teselace.*

2.3.2 Kvalita trojúhelníka

Pro posouzení získané teselace je vhodné mít určité kritérium. Tím může být kvalita dílčích trojúhelníků. Kvalita trojúhelníka je určité numerické ohodnocení, které umožňuje porovnat dva trojúhelníky a rozhodnout, který z nich je „lepší“. Pokud budeme pokládat rovnostranný trojúhelník za nejlepší variantu, můžeme pro výpočet kvality použít následující metriku, která je také použita v knihovně VectorEntity (viz kapitola 3.1):

$$q(t_{ABC}) = \frac{\sqrt{3} \cdot \max\{|AB|, |BC|, |CA|\} \cdot o(t_{ABC})}{12 \cdot S(t_{ABC})}, \tag{2-16}$$

kde t_{ABC} značí trojúhelník t s vrcholy A, B, C, $\max\{|AB|, |BC|, |CA|\}$ je délka nejdelší strany, $o(t_{ABC})$ je obvod a $S(t_{ABC})$ obsah trojúhelníka. Tato metrika přiřazuje rovnostrannému trojúhelníku kvalitu hodnoty 1, vyšší hodnoty znamenají menší kvalitu.

Nutno poznamenat, že je tato metrika invariantní vůči ohodnocení podobných[1] trojúhelníků. To lze jednoduše dokázat dedukcí. Pro dva podobné trojúhelníky t_{ABC} a $t_{A'B'C'}$ platí, že $|A'B'| = k \cdot |AB|$, $|B'C'| = k \cdot |BC|$ a $|C'A'| = k \cdot |CA|$, z čehož plyne:

$$q(t_{A'B'C'}) = \frac{\sqrt{3} \cdot \max\{k \cdot |AB|, k \cdot |BC|, k \cdot |CA|\} \cdot k \cdot o(t_{ABC})}{12 \cdot k^2 \cdot S(t_{ABC})} = q(t_{ABC}). \tag{2-17}$$

Většinu polygonů lze teselovat více způsoby. Za nejvhodnější teselaci můžeme potom považovat např. takovou, kde součet kvalit získaných trojúhelníků nabývá nejmenší hodnoty.

[1] Dva trojúhelníky jsou podobné, pokud se rovnají jejich vnitřní úhly.

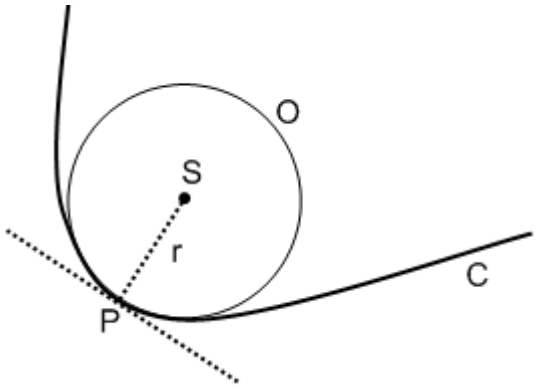
2.4 Kostra

Kostru trojúhelníkové sítě můžeme definovat jako soubor hran, které vyznačují význačné linie tělesa, jež mají význam na utváření jeho tvaru. Pokud by tyto hrany byly pozměněny nebo odstraněny, ovlivní to vzhled tělesa. Ostatní hrany tak velký význam nemají, proto mohou být při převodu trojúhelníkové sítě na quadrilaterální odstraněny a nahrazeny jinými. Obecně lze říct, že se hrany kostry nacházejí v místech s ostrými přechody. Nalezení těchto míst ale nemusí být zcela jednoduché.

2.4.1 Křivost

Dříve než přistoupíme k určování vrcholů a hran kostry, musíme nejprve vysvětlit, co je to křivost a jak s ní budeme pracovat ve spojení s polygonální sítí. Informace jsem čerpal z [Weisstein09b].

Pokud budeme mluvit o křivce v rovině, její křivost κ v bodě P bude rovna převrácené hodnotě poloměru tzv. oskulační kružnice, což je kružnice, která nejlépe aproximuje křivku v okolí bodu P. Přímka má ve všech bodech konstantní nulovou křivost, zatímco křivka může v každém bodě nabývat libovolné křivosti.



Obrázek 2-14: *Obecná křivka a její oskulační kružnice v bodě P.*

Určování křivosti plochy zakřivené v prostoru je poněkud komplikovanější. Pokud v určitém bodě plochy proložíme rovinu, která obsahuje normálový vektor n a libovolný tečný vektor t plochy v daném bodě P, získáme tzv. normálový řez. Křivost křivky odpovídající normálovému řezu se nazývá *normálová* nebo také *normální křivost* plochy podle vektoru t a značí se $k_n(t)$. Podle konvence nabývá normální křivost kladných hodnot, pokud je křivka „ohnutá“ ve směru normálového vektoru zakřivené plochy, v opačném případě záporných.

Normálových řezů je nekonečně mnoho a tedy v daném bodě roviny lze určit nekonečně mnoho normálních křivostí. Normální křivost, jež nabývá maximální hodnoty, se nazývá *maximální křivost* a značí se κ_1, analogicky k tomu normální křivost, jež nabývá minimální hodnoty, se nazývá *minimální křivost* a značí se κ_2. Souhrnně se tyto křivosti nazývají hlavní křivosti. Pro dvojici hlavních křivostí potom platí, že směrový vektor t_1 křivosti κ_1 je kolmý ke směrovému vektoru t_2 křivosti κ_2. Kromě hlavních křivostí se také často používá *střední křivost H* a *Gaussova křivost K*. Střední křivost odpovídá průměrné hodnotě hlavních křivostí, Gaussova jejich součinu, tedy:

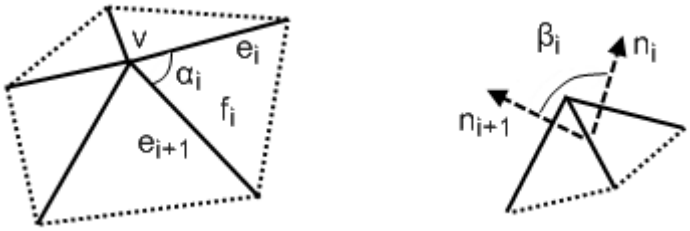
$$H = (\kappa_1 + \kappa_2)/2, \tag{2-18}$$

$$K = \kappa_1 \kappa_2. \tag{2-19}$$

2.4.2 Křivost polygonální sítě

Protože je polygonální síť po částech lineární aproximací, křivost ve skutečnosti nelze určit v žádném jejím bodě. Z toho důvodu je třeba použít určité metriky, které by vhodně aproximovaly křivost původní plochy. Takovou vhodnou aproximací křivostí může být metrika použitá v [Hulík08] a také [Kim02]:

Trojúhelníková síť M se skládá z množiny plošek F, množiny hran E a množiny vrcholů V. Mějme libovolný vrchol $v \in V$, *jehož okolní plošky označíme* f_i, *obsah plošek* S_i, *okolní hrany* e_i, *okolní vrcholy* v_i, *úhly svírané hranami* e_i *a* e_{i+1} *jako* α_i *a úhly svírané normálovými vektory plošek* f_i *a* f_{i+1} *jako* β_i.



Obrázek 2-15: *Vlevo okolí vrcholu v, vpravo dihedrální úhel.*

Gaussovu a střední křivost pro tento vrchol můžeme spočítat následovně:

$$K = \frac{2\pi - \sum_{i=1}^{n} \alpha_i}{\frac{1}{3}\sum_{i=1}^{n} S_i}, \tag{2-20}$$

$$H = \frac{\frac{1}{4}\sum_{i=1}^{n} \|e_i\| \beta_i}{\frac{1}{3}\sum_{i=1}^{n} S_i}. \tag{2-21}$$

Ze vzorců 2-18 a 2-19 lze potom odvodit vzorec pro výpočet hlavních křivostí:

$$\kappa_{1,2} = H \pm \sqrt{H^2 - K}\ . \tag{2-22}$$

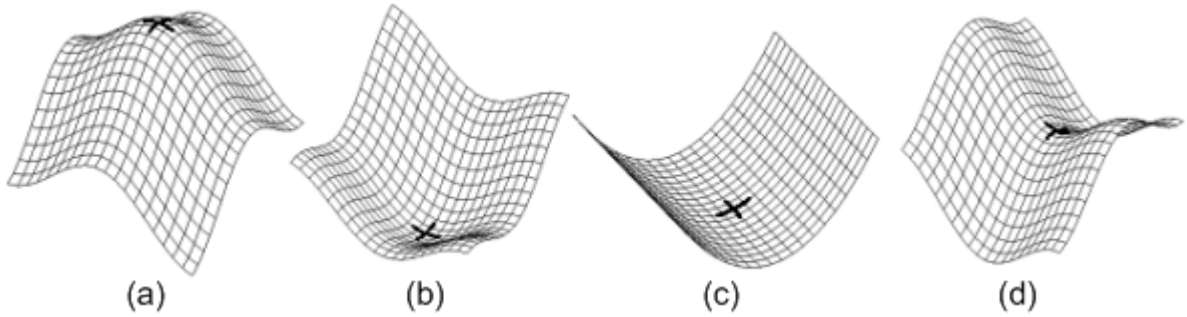
Vzorec 2-22 je použitelný ve většině případů. U polygonálních sítí však není zaručeno, že je H^2 vždy větší než K, proto v některých případech vychází odmocnina ze záporného čísla. Pro spojitou 3D plochu je argument odmocniny vždy nezáporný, přičemž nulové hodnoty nabývá, pouze pokud se hlavní křivosti rovnají. Z toho lze usuzovat, že v případě polygonální sítě nabývá argument odmocniny záporných hodnot v důsledku chyby aproximace tehdy, kdy jsou hodnoty hlavních křivostí podobné. Pokud zanedbáme rozdíl hodnot těchto křivostí, můžeme je nahradit hodnotou střední křivosti.

2.4.3 Určení kostry

Jak bylo uvedeno v kapitole 2.2.8, kostra se skládá z řetězů hran, které mají koncové vrcholy zvané šipky nebo se sbíhají ve vrcholech zvaných rohy. Otázkou je, které vrcholy a hrany sítě vybrat jako součást kostry.

Podle hodnot a především podle znaménka křivostí popisovaných v kapitole 2.4.1 lze zjistit určité charakteristiky plochy v bodě. Podle toho, zda Gaussova křivost nabývá kladných, nulových nebo záporných hodnot lze určit, že je plocha lokálně eliptická (vypouklá nebo dutá), rovinná či parabolická, nebo hyperbolická. Vypouklost nebo dutost plochy rozhodneme podle znaménka

hlavních křivostí resp. střední křivosti. Rovinnost od parabolicity odlišíme podle toho, zda se nule blíží obě nebo jen jedna z hlavních křivostí resp. podle toho, zda se střední křivost blíží nule.



Obrázek 2-16: *Příklady lokálního zakřivení ploch: (a) vypouklé, (b) vyduté, (c) parabolické, (d) hyperbolické.*

Hodnoty jednotlivých křivostí tedy mohou být užitečné při rozlišování vnějších a vnitřních hran, ploch, sedlových bodů, výčnělků a výdutí. Samy o sobě ale nejsou rozhodující, protože pouze poukazují na lokální charakter v jednom bodě, zatímco pro posouzení tvaru plochy je nutné brát v potaz větší okolí. Na základě znalosti tohoto tvaru lze potom odvodit umístění vrcholů a hran kostry. Řetězy by měly sledovat hřebeny parabolických útvarů. Šipky by měly ležet na konci řetězů v místech, kde parabolické útvary přecházejí do ploch s malou křivostí. Rohy se mohou vyskytovat na špičkách výčnělků, na spodku výdutí nebo v centru hyperbolických útvarů, samozřejmě pokud v okolí leží nějaké řetězy.

2.5 3D spline plocha

Informace o splinech jsem čerpal především z [Kršek08] a [Weisstein09b]. Pod pojmem spline (*česky „pružné pravítko“*) chápeme křivku proloženou množinou bodů, obvykle definovanou jako po částech polynomiální. Spline může procházet přímo jednotlivými body nebo mezi nimi – podle toho rozlišujeme interpolační a aproximační spliny. Protože změnou polohy jednotlivých bodů lze upravovat tvar splinu, nazýváme tyto body řídící.

Souřadnice jednotlivých bodů křivky je dána součtem členů, které mají určitý vztah k řídícím bodům. Pozici vzhledem ke křivce udává parametr t, jehož hodnota leží v intervalu $\langle 0;1 \rangle$, kde 0 odpovídá počátečnímu bodu, 1 koncovému. Počet řídících bodů může být pevný (např. Beziérovy kvadriky – 3, kubiky – 4) nebo libovolný shora neomezený (např. B-spline – 2 a více). To, že je spline polynomiální po částech, znamená, že lze dílčí křivky skládat do jednoho většího celku (více o navazování segmentů viz kapitola 2.5.1 – Spojitost).

Tak jako u 2D křivky body $Q(t)$ závisí na vektoru řídících bodů, u 3D spline plochy závisí body $Q(u,v)$ na řídících bodech uspořádaných obvykle v obdélníkové mřížce. Obdélníková mřížka není jediný způsob definice 3D spline plochy, jedná se však o nejpoužívanější variantu, kterou se také zabývám v této práci. Pro nás je velice důležitá korespondence takové obdélníkové mřížky s quadrilaterální sítí (viz kapitola 2.2).

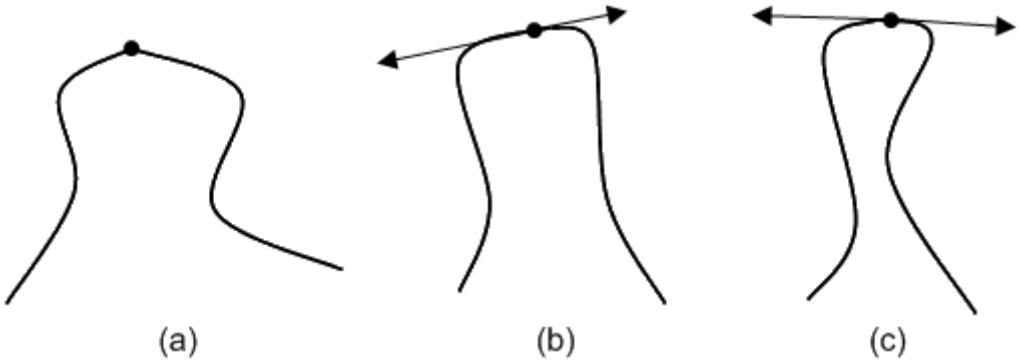
Obrázek 2-17: *(a) vektor řídících bodů 2D spline křivky, (b) obdélníková mřížka řídících bodů 3D spline plochy.*

2.5.1 Vlastnosti splinu

Na tvar splinu můžeme mít určité požadavky, jako např. minimalizace křivosti, lokalita změn, stupeň spojitosti při navazování segmentů nebo invariantnost vůči transformacím. Podle splnění různých kritérií se každý druh splinu hodí k různým účelům. V současnosti se jako standard ve 3D grafice používají NURBS plochy, které jsou postupně nahrazovány modernějšími T-spline.

Spojitost

Často je spline potřeba poskládat z dílčích segmentů, a proto nás také zajímá, jakým způsobem na sebe jednotlivé části navazují. Pro některé účely postačuje, pokud mezi segmenty nevzniká žádná mezera a přechod je na pohled hladký. Jiné aplikace naopak mohou vyžadovat spojitost určitého stupně. Pokud má spline spojitost stupně *n*, značí se C^n a musí platit, že hodnoty všech derivací v rozsahu *0-n* musí být v koncových bodech shodné. Z toho plyne, že spojitost spline křivky C^0 je totožnost koncových bodů, C^1 navíc totožnost tečných vektorů (vyjma orientace), C^2 navíc totožnost vektorů druhé derivace. Kromě toho lze ještě uplatnit oslabenou podmínku spojitosti, značenou jako G^n, která nevyžaduje totožnost tečných vektorů do n-té derivace, ale pouze jejich lineární závislost.



Obrázek 2-18: *Spojitost křivek stupně: (a) C^0 – koncové body jsou totožné, (b) G^1 – tečné vektory jsou lineární kombinací, (c) C^1 – tečné vektory jsou totožné.*

Váhové koeficienty

Spliny lze obecně rozdělit na racionální a neracionální. V prvním případě lze jednotlivým řídícím bodům přiřadit váhu, čímž se ovlivní velikost jejich vlivu na tvar splinu. U neracionálních splinů mají všechny řídící body stejnou váhu, jedná se tedy o speciální případ racionálních splinů.

Invariantnost vůči transformacím

Při práci se spliny se často aplikují různé transformace. Ve 3D grafice se používají lineární transformace jako např. posunutí, rotace, perspektivní projekce apod. Z tohoto důvodu je vhodné, aby byl použitý druh splinu invariantní vůči těmto transformacím. V opačném případě by měla stejná křivka či plocha odlišný tvar v závislosti na použitých transformacích, což by samozřejmě znemožňovalo efektivní práci.

Obecnost

Důležitou otázkou je, zda lze určitý spline použít pro popis libovolného tvaru. Řada splinů si neporadí s kuželosečkami, přitom taková kružnice nebo elipsa bude zcela určitě hodně používaný tvar. Tento problém jistě přispěl k evoluci splinů, která směřuje k co nejobecnějšímu popisu. V rámci obecnosti umožňují moderní spliny popsat celou křivku či plochu jako jeden celek a není tedy potřeba řešit skládání dílčích segmentů.

2.5.2 NURBS

NURBS (*non uniform rational B-spline*) v současnosti představuje jednu z nejobecnějších definicí splinu. 2D křivka stupně k skládající se z n úseků je popsána $n+1$ řídícími body P_i, kterým je možné nastavit váhu ω_i, a neuniformním uzlovým vektorem U_t délky $n+k+1$, který se skládá z neklesající posloupnosti hodnot v intervalu $\langle a;b\rangle$, přičemž prvních resp. posledních $k+1$ hodnot musí být a resp. b. Jednotlivé úseky splinu není nutné navazovat, protože společně tvoří jeden celek. Závislost na příslušných řídících bodech určuje rekurentně definovaná B-spline bázová funkce:

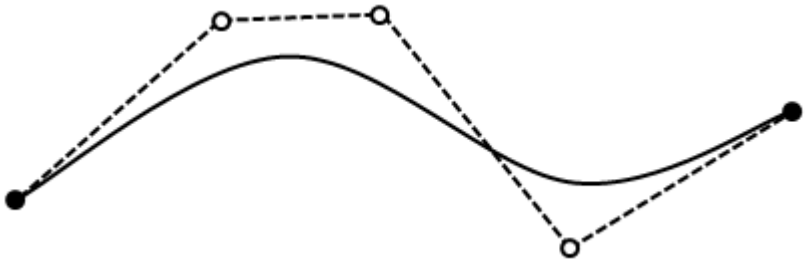
$$N_i^0(t) = \begin{cases} 1 & t \in \langle t_i; t_{i+1}) \\ 0 & t \notin \langle t_i; t_{i+1}) \end{cases}$$
$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} \cdot N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} \cdot N_{i+1}^{k-1}(t) \quad . \tag{2-23}$$

Konkrétní souřadnice bodů spline křivky lze pro $t \in \langle 0;1\rangle$ určit podle následujícího vztahu (v souladu s tím, co uvádí [Weisstein09c]):

$$Q(t) = \frac{\sum_{i=0}^{n} \omega_i \cdot N_i^k(t) \cdot P_i}{\sum_{i=0}^{n} \omega_i \cdot N_i^k(t)} \quad . \tag{2-24}$$

Protože je počet řídících bodů libovolný, může být křivka popsána jako jeden celek a není tedy potřeba řešit navazování dílčích segmentů. Díky zavedení neuniformního uzlového vektoru lze vkládat řídící body při zachování tvaru a je možné modelovat základní kuželosečky. K důležitým vlastnostem také patří invariantnost vůči afinním[1] a perspektivní transformaci.

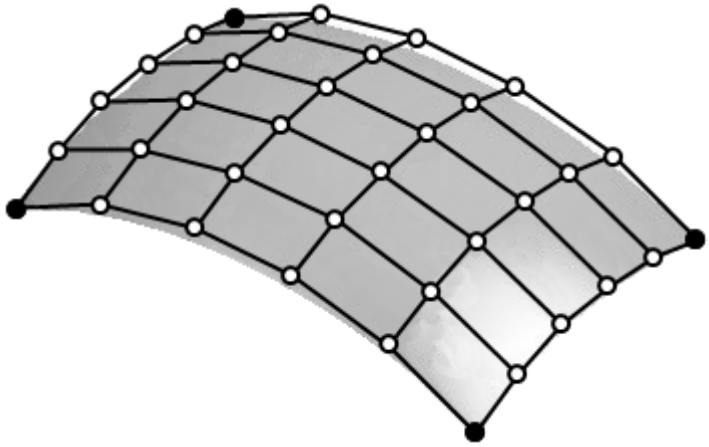
[1] Při afinních transformacích se zachovává rovnoběžnost přímek.



Obrázek 2-19: *NURBS křivka a její řídící body. Křivka prochází pouze okrajovými body, ostatní aproximuje.*

Analogicky k 2D křivce je 3D spline plocha stupně (p, q) skládající se z $n \times m$ úseků popsána sítí $(n+1)\times(m+1)$ řídících bodů $P_{i,j}$ tvořících obdélníkovou mřížku, kterým lze nastavit váhu $\omega_{i,j}$, a neuniformní uzlovou maticí $M_{u,v}$ rozměru $(p+1)\times(q+1)$, jejíž řádky i sloupce odpovídají neuniformním uzlovým vektorům. Souřadnice bodů 3D spline plochy lze pro $u,v \in \langle 0;1 \rangle$ určit podle následujícího vztahu (v souladu s [Weisstein09c]):

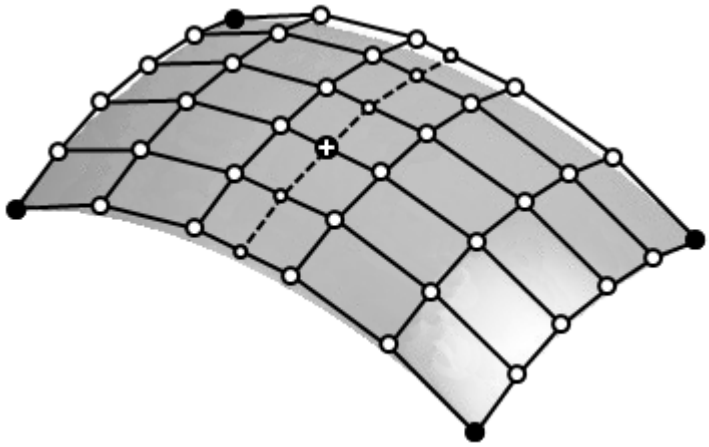
$$Q(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} \omega_{i,j} \cdot N_i^p(u) \cdot N_j^q(v) \cdot P_{i,j}}{\sum_{i=0}^{m}\sum_{j=0}^{n} \omega_{i,j} \cdot N_i^p(u) \cdot N_j^q(v)}. \tag{2-25}$$



Obrázek 2-20: *Plát NURBS plochy a odpovídající síť řídících bodů. Plocha prochází pouze rohovými řídícími body, ostatní aproximuje.*

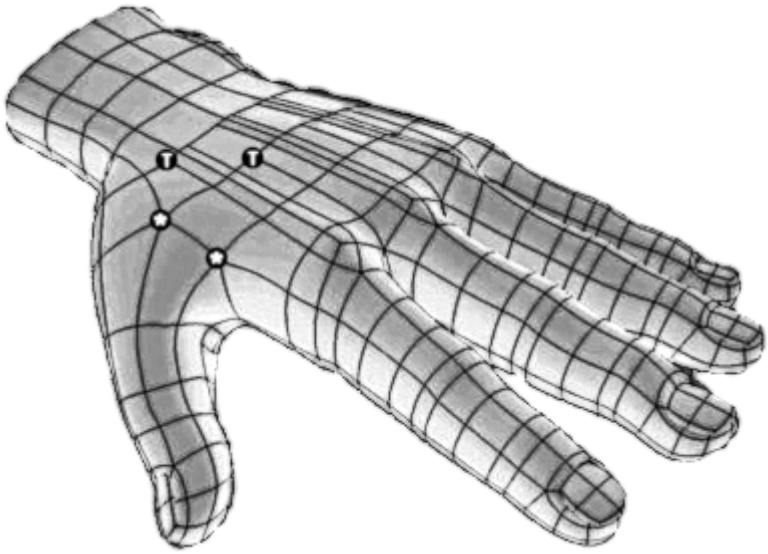
2.5.3 T-spline

Díky obecnosti definice jsou NURBS v současnosti asi nejpoužívanějším druhem spline křivek i ploch. V případě NURBS ploch ale narážíme na jistá omezení. Přestože definice umožňuje popsat plochu jako jeden celek, v reálných aplikacích se model musí poskládat z dílčích plátů. V důsledku přidávání řídících bodů je nutné kaskádovitě doplňovat řadu dalších, tak aby síť řídících bodů stále tvořila obdélníkovou mřížku. Při modelování komplexních modelů by ale taková síť byla velice složitá a znemožňovala by efektivní práci. Proto se složitější modely skládají z dílčích plátů, které jsou jednodušší, zato se musí řešit jejich spojitost.



Obrázek 2-21: *Vložení řídícího bodu a kaskádovité doplnění dalších, aby byla zachována obdélníková mřížka.*

Tento nedostatek řeší T-spline plochy, jejichž síť řídících bodů může obsahovat T-zakončení a tzv. hvězdicové body, viz obrázek 2-22. Tak jak uvádí [T-Splines09], díky existenci T-zakončení je možné provádět změny lokálního charakteru bez růstu složitosti řídící sítě. Pomocí hvězdicových bodů lze potom jednoduše modelovat póly, které by se jinak vytvářely značně obtížně. T-spline plochu lze navíc převést na NURBS, takže je možné zajistit zpětnou kompatibilitu se staršími aplikacemi, které s T-spline pracovat neumí.



Obrázek 2-22: *Ilustrace T-zakončení a hvězdicových bodů na modelu ruky.*

3 Návrh a implementace

3.1 Prostředky

Pro implementaci jsem si vybral programovací jazyk C++, který je stále hojně využívaný pro tvorbu grafických aplikací. Jako vývojové prostředí jsem z počátku používal Microsoft Visual Studio 2003, později jsem přešel na verzi 2005. V implementaci používám knihovny OSG (ke stažení na www.openscenegraph.org) a MDSTk (ke stažení na www.fit.vutbr.cz/~spanel/mdstk). Knihovna OSG slouží jako objektově orientovaná nástavba OpenGL. Knihovna MDSTk je vyvíjena v rámci školy (FIT VUT v Brně) a zahrnuje řadu modulů využitelných pro 2D i 3D grafické aplikace. V mé aplikaci využívám konkrétně součásti LAPACK a VectorEntity. LAPACK je vysoce optimalizovaná knihovna pro řešení soustav rovnic, VectorEntity slouží pro strukturované ukládání 3D entit (vrcholy, hrany, trojúhelníky) polygonálních trojúhelníkových 3D sítí a jejich manipulaci.

3.2 Struktura aplikace

Aplikace se skládá ze dvou celků, hlavního programu a knihovny funkcí, pracovně nazvaných MeshConverter a Toolkit359.

3.2.1 Knihovna Toolkit359

Tuto knihovnu jsem vytvořil pro oddělení znovupoužitelného kódu od samotné aplikace. Obsažené funkce jsou následující:

1. Šablonové adaptéry kolekcí z knihovny STL, které zjednodušují použití a přidávají další funkčnost.
2. Zjednodušení práce s poli.
3. Sjednocení různých matematických funkcí knihovny math.h, LAPAC a dalších.
4. Rutiny a usnadnění práce s řetězci a základními datovými typy.
5. Hierarchie výjimek inspirovaná jazykem Java.
6. Načítání parametrů z příkazové řádky.
7. Třídy objektů pro počítání referencí a automatické uvolňování paměti.
8. Další užitečné funkce.

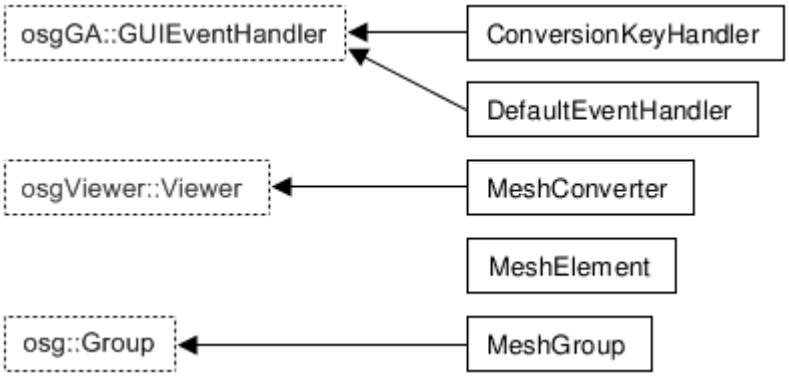
Třídy jsou rozděleny do několika složek, z nichž v následujícím textu jsou uvedeny složky *collections* a *types*. Bližší popis této knihovny je k nalezení v programové dokumentaci v příloze C.

3.2.2 Hlavní program

Hlavní program se skládá z několika logických celků, které jsou rozděleny do samostatných složek. Toto rozdělení se snaží oddělovat programový kód s rozdílným zaměřením funkčnosti. V následujících odstavcích stručně popíšu hierarchii vytvořených tříd, podrobnější popis je opět k nalezení v programové dokumentaci v příloze C.

meshConverter

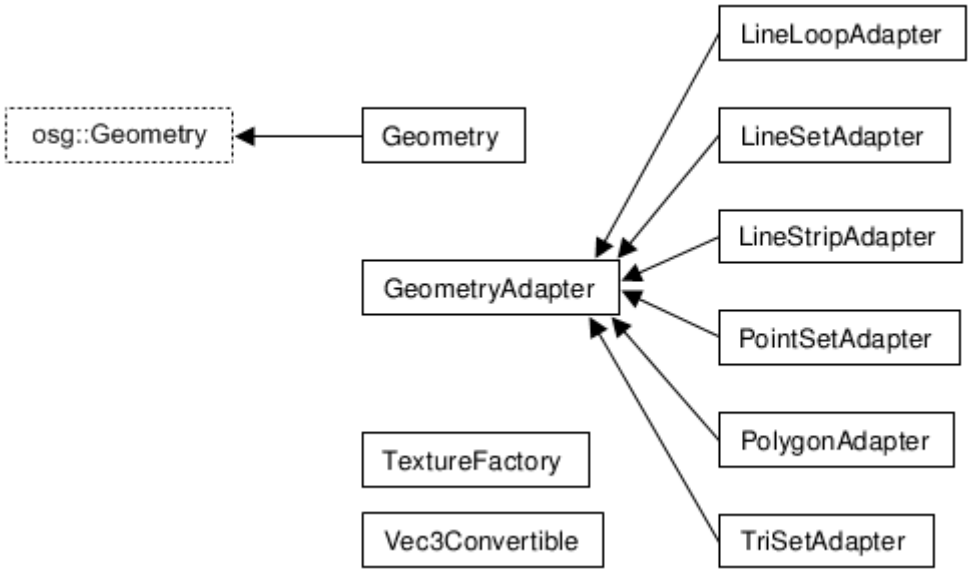
Složka *meshConverter* představuje vrchol aplikační struktury. Obsahuje funkci *main*, která zajistí načtení uživatelských parametrů a vytvoření instance třídy *MeshConverter*. Tento objekt spolu se třemi ovladači událostí (*angl. event handlers*) *ConversionKeyHandler*, *DefaultEventHandler* a *StatusListener*, zajišťuje veškerou další režii v podobě řízení převodu, komunikace s uživatelem a zobrazení výstupů. Další důležitou třídou je *MeshGroup*, která slouží pro vytvoření grafické reprezentace zpracovávané sítě a dalších entit pro znázornění převodu.



Obrázek 3-1: *Hierarchie tříd složky meshConverter (čárkované třídy jsou z jiné složky či externí knihovny).*

osgUtil

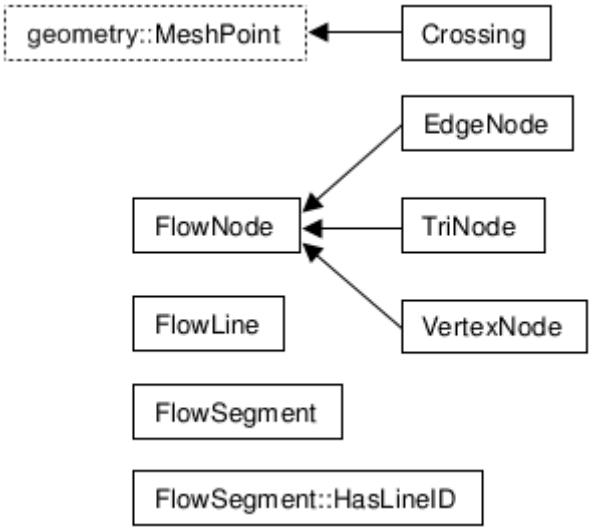
Ve složce *osgUtil* najdeme adaptéry pro vytváření OSG geometrií a funkce pro generování textur.



Obrázek 3-2: *Hierarchie tříd složky osgUtil (čárkované třídy jsou z jiné složky či externí knihovny).*

flowElements

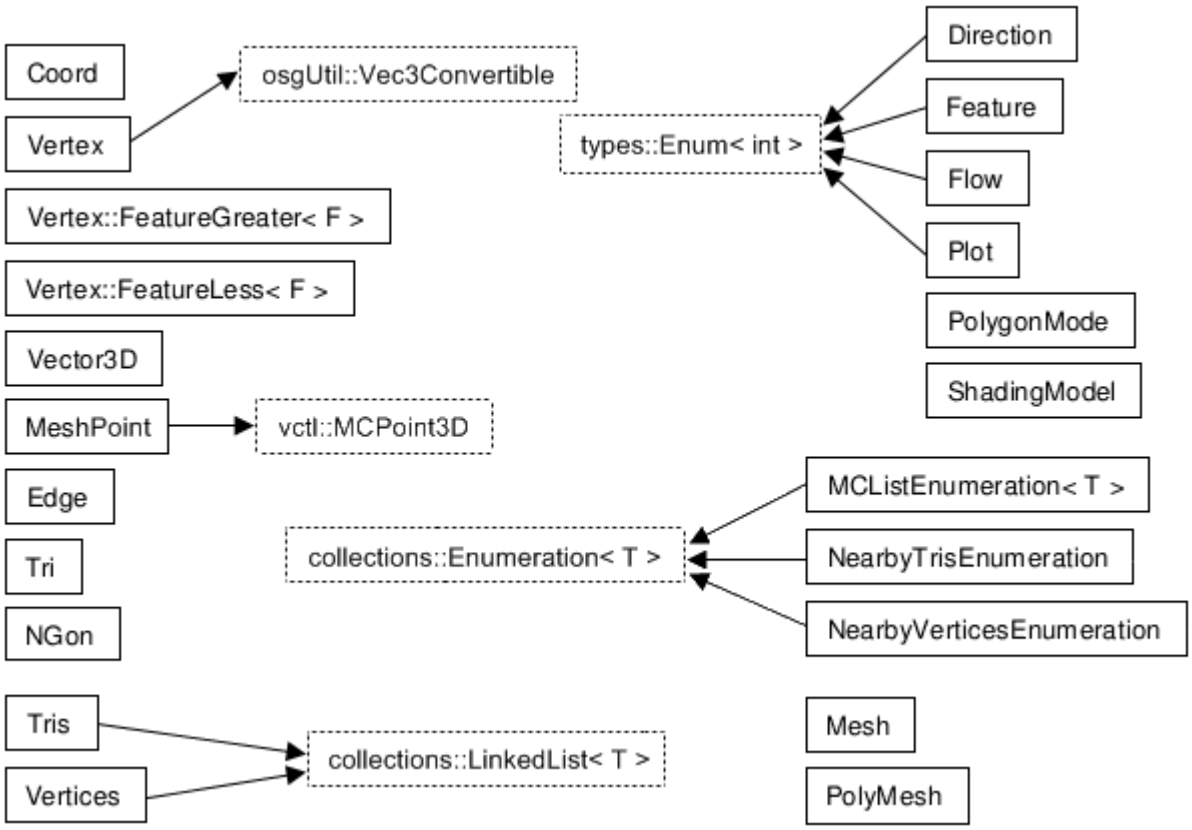
Složka *flowElements* obsahuje reprezentace entit používaných při vytváření sítě linek. Na nejvyšší úrovni to je objekt typu *FlowLine* (plovoucí linka), který obsahuje objekty typu *FlowSegment* (segment plovoucí linky) a *FlowNode* (uzel plovoucí linky), jehož implementace může být realizována třídami *EdgeNode*, *TriNode* a *VertexNode*. Při hledání průsečíků linek jsou vytvářeny objekty typu *Crossing*, které se ukazateli propojují do odpovídající sítě.



Obrázek 3-3: *Hierarchie tříd složky flowElements (čárkované třídy jsou z jiné složky či externí knihovny).*

geometry

Složka *geometry* obsahuje klíčovou funkčnost pro přesíťování trojúhelníkové sítě na quadrilaterální. Celý proces je realizován prostřednictvím objektu typu *Mesh*, jehož metodami jsou počítány jednotlivé kroky převodu.



Obrázek 3-4: *Hierarchie tříd složky geometry (čárkované třídy jsou z jiné složky či externí knihovny).*

3.3 Převod na quadrilaterální síť

Jak bylo vysvětleno v kapitole 2, převod nestrukturované trojúhelníkové 3D sítě na 3D spline plochu je realizován ve dvou etapách. Nejdříve je nutné vytvořit quadrilaterální 3D síť a tu následně převést na výslednou 3D spline plochu. Převod vstupní sítě na quadrilaterální sestává z několika dílčích kroků. Každý krok tohoto převodu má konkrétní vstup a výstup, takže lze v podstatě říct, že celý proces odpovídá jisté výpočetní lince.

3.3.1 Inicializace

Na počátku je třeba načíst vstupní model do vhodné datové struktury. Protože pracujeme pouze s geometrií jednolitého modelu (žádné textury, materiály nebo dílčí objekty), plně nám postačuje formát STL, viz kapitola 3.3.1 – Formát STL. Pro uchování trojúhelníkové sítě může být použita např. knihovna VectorEntity (viz kapitola 3.1), která také poskytuje funkci pro načtení modelu z STL souboru. Před započetím převodu musí být jednotlivým vrcholům sítě přiřazeny unikátní indexy, nejlépe tvořící posloupnost celých čísel v intervalu $\langle 0; n-1 \rangle$, kde n je počet vrcholů.

Formát STL

Popis formátu STL jsem čerpal z [Rypl05]. Jedná se o jednoduchý datový formát pro popis geometrie modelu. Existuje textová i binární varianta. Model je tvořen sítí trojúhelníků nazývaných jako *facet*, přičemž popis každého trojúhelníku sestává ze souřadnic normály a jeho tří vrcholů. Podle původní specifikace musí být splněny následující podmínky:

1. Každé dva sousední trojúhelníky sdílí právě dva vrcholy.
2. Normála trojúhelníka má jednotkovou délku a směřuje ven z modelu.
3. Vrcholy trojúhelníka jsou uvedeny v pořadí podle směru hodinových ručiček při pohledu z venku.
4. Model leží v celo-kladném oktantu, souřadnice všech vrcholů jsou tedy kladné.

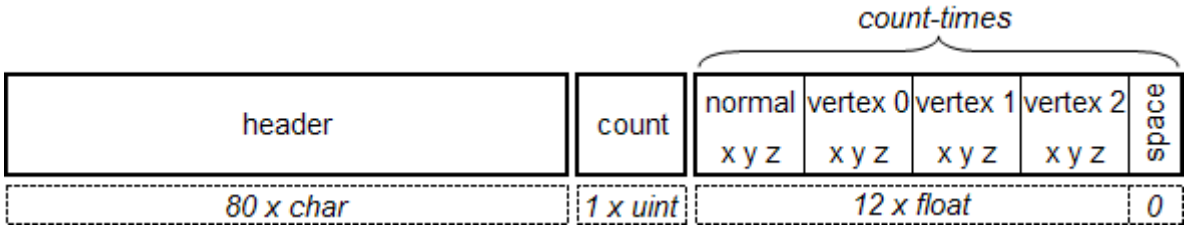
Řada aplikací však tyto podmínky nedodržuje. Podmínka 1 se porušuje při popisu non-manifold modelů. Normála představuje nadbytečnou informaci, protože lze odvodit ze souřadnic vrcholů, a proto se někdy neuvádí a je nahrazena nulami. Podmínka 4 má význam ve stereolitografii, ve 3D grafice však nemá opodstatnění.

Textová varianta má následující strukturu:

```
solid [...]
  facet normal <x> <y> <z>          +
    outer loop
      vertex <x> <y> <z>
      vertex <x> <y> <z>
      vertex <x> <y> <z>
    endloop
  endfacet
endsolid
```

Klíčové slovo *solid* většinou slouží pro identifikaci textové varianty a bývá následováno doplňujícími informacemi jako název modelu, autor, datum apod. Slovo *facet* uvozuje definici trojúhelníku, *normal* definici souřadnic normály a *outer loop* blok souřadnic vrcholů, které jsou uvozeny slovem *vertex*. Ke slovům *solid*, *facet* a *outer loop* existují jim odpovídající slova *endsolid*, *endfacet* a *endloop* ukončující daný blok.

Binární varianta má úspornější strukturu. 80bajtová hlavička obsahující doplňující informace je následována 4bajtovým celým číslem bez znaménka udávajícím počet trojúhelníků, jejichž definice následuje v podobě 50bajtových sekvencí. Každá z těchto sekvencí obsahuje souřadnice normály a vrcholů v podobě trojic 4bajtových reálných čísel a je zakončena dvěma prázdnými bajty jako zarovnání.



Obrázek 3-5: *Struktura binárního STL.*

3.3.2 Vytvoření kostry

Výběr hran náležících kostře polygonální sítě může být analogický vyhledání hran ve 2D rastrovém obraze. Přestože polygonální síť představuje zakřivenou plochu oproti 2D rastru, který je reprezentován pravoúhlou rovinnou mřížkou, můžeme nalézt jednu významnou společnou vlastnost. Tak jako má každý obrazový bod své sousední body, každý vrchol sítě má sousední vrcholy. Díky tomu lze pro vytvoření kostry použít některé metody používané v oblasti segmentace obrazu, v případě, že počet sousedních elementů a jejich poloha není rozhodující.

Výpočet křivostí

Před konstrukcí kostry je třeba spočítat křivosti ve vrcholech. Jak bylo vysvětleno v kapitole 2.4.2, přesný výpočet v rámci polygonální 3D sítě není možný, můžeme ale použít určité aproximace. Jednou alternativou mohou být vzorce 2-14, 2-15 a 2-16, kterými získáme požadované hodnoty křivostí ve vrcholech. Celý výpočet tedy lze provést pro každý vrchol zvlášť v rámci jednoho průchodu přes kolekci vrcholů. Jediné nebezpečí představuje odmocnina ve vzorci 2-16, jejíž argument nemusí být vždy nezáporný. V těchto případech pouze použijeme hodnotu střední křivosti jako hodnotu hlavních křivostí.

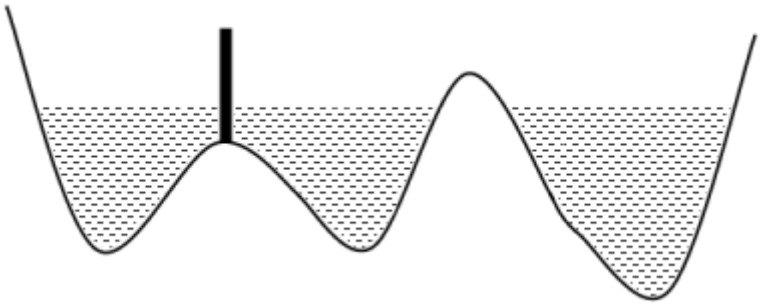
Metoda rozvodí (watersheds)

Tento algoritmus se nazývá metoda rozvodí (*angl. watersheds*), protože je analogií zaplavování terénu a oddělování jednotlivých povodí hrázemi, aby se nespojily. Stejně jako je terén zaplavován od nejnižších oblastí po nejvyšší, u 2D obrazu procházíme jednotlivé body seřazené vzestupně podle gradientu, který obvykle reprezentuje velikost změny jasové složky. Pro každý bod mohou nastat tyto tři situace:

1. V okolí se nenachází žádné povodí – je vytvořeno nové.
2. Bod leží na okraji jednoho povodí – je pohlcen.

3. Bod leží na hranici více povodí – je vytvořeno rozvodí.

Body, které byly označeny jako rozvodí, leží v místech lokálního maxima gradientu. Výhodou tohoto algoritmu je spolehlivé vyhledání uzavřených segmentů s jednoznačnými hranicemi. Největší nevýhodou je potom velká náchylnost na šum v datech, což vede k nadsegmentaci (vytvoření příliš velkého množství segmentů). Tento jev lze řešit různými způsoby – např. prahování, hierarchické slučování nebo značení počátečních segmentů.

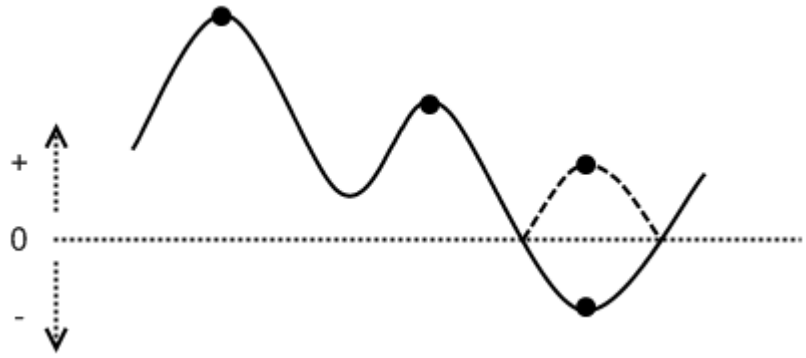


Obrázek 3-6: *Ukázka zaplavování terénu a vznik rozvodí (silná linka).*

Metoda rozvodí pro trojúhelníkovou síť

Pro použití metody rozvodí ve spojení s trojúhelníkovou sítí, která představuje po částech lineární aproximaci 2D plochy zakřivené v prostoru, předpokládáme křivost spočítanou ve všech vrcholech (viz kapitola 2.4.2), která dosahuje extrémních hodnot v těch vrcholech, které budou tvořit kostru. V případě spojité plochy by bylo asi nejvhodnější zohlednit maximální křivost, protože řetězy hran kostry by měly sledovat hřebeny parabolických zakřivení. Protože ale pracujeme s trojúhelníkovou sítí, výpočet křivostí je pouze aproximací, a proto lze také uvažovat o použití např. střední křivosti.

Narozdíl od původního algoritmu, který zohledňuje pouze kladné extrémy, my potřebujeme nalézt i záporné. Kladné extrémy totiž představují vnější hrany modelu, zatímco záporné představují vnitřní hrany (hrany směřující dovnitř modelu). Tento problém lze jednoduše vyřešit použitím absolutní hodnoty křivosti.



Obrázek 3-7: *Hledání kladných i záporných extrémů.*

Konstrukce kostry

V souladu s metodou rozvodí je třeba seřadit vrcholy vzestupně podle spočtené křivosti. K tomuto účelu jsem použil datovou strukturu *std::multiset* z knihovny STL, která uchovává vložená data uspořádaná podle dané relace uspořádání, přičemž umožňuje i rovnost (některé vrcholy mohou mít shodnou křivost). Samotná segmentace potom probíhá následovně:

1. Postupně pro všechny vrcholy sítě ve vzestupném pořadí podle absolutní hodnoty zohledňované křivosti zjisti, do jakého segmentu patří okolní vrcholy.
   a/ Pokud žádný z okolních vrcholů nepatří do žádného segmentu, vytvoř nový segment.
   b/ Pokud se v okolí nachází vrcholy, které patří právě do jednoho segmentu, přidej vrchol do tohoto segmentu.
   c/ Pokud se v okolí nachází vrcholy z více segmentů, nastav vrchol jako hraniční.
   Σ ab/ V návaznosti na možnosti *a* a *b* přidej do daného segmentu všechny okolní vrcholy se stejnou křivostí a rekurzivně pokračuj na okolní vrcholy těchto vrcholů. Takto se zamezí vzniku více segmentů v případných oblastech s konstantní křivostí, u kterých není zaručeno správné pořadí výběru vrcholů.
2. Pro všechny vrcholy sítě v libovolném pořadí zjisti, jestli jsou hraniční. Pokud jsou, vytvoř hranu kostry ke všem okolním hraničním vrcholům, které mají vyšší index.

Popsaným způsobem se vytvoří maximum možných segmentů, které však bude nejspíše vyšší, než potřebujeme. Nejjednodušším, přesto docela účinným řešením je prahování hodnot blízkých nule. Všechny křivosti nižší než zadaný práh se považují za nulové. V oblastech s malou křivostí potom nevzniká množství zbytečných segmentů.

Použití kostry

Konstrukce kostry má dva hlavní důvody. Prvně rohové vrcholy (viz kapitola 2.2.8) by měly být použity jako počáteční semínka při vytváření plovoucích linek, podrobněji v kapitole 3.3.3 – Síť plovoucích linek. Za druhé hrany kostry by měly být použity při konstrukci nové sítě (viz kapitola 3.3.4). Toho se dosáhne tak, že při vytváření sítě plovoucích linek jsou vytvořeny dodatečné vrcholy na hranách kostry, které jsou během vytváření nové sítě spojeny se sousedními vrcholy kostry. Kostra je tak začleněna do výsledné sítě.

3.3.3 Vytvoření sítě plovoucích linek

První částí konstrukce quadrilaterální sítě je vytvoření sítě plovoucích linek, jejíž linky sledují buďto gradientní nebo izoparametrické vektorové pole definované podél povrchu vstupního modelu reprezentovaného trojúhelníkovou sítí. Obě vektorová pole se spočítají z hodnot skalárního pole, které konverguje k předem definovaným extrémům ležícím v tzv. omezených vrcholech.

Omezené vrcholy

Výběr omezených vrcholů může být ponechán na uživateli a lze realizovat nejlépe pomocí myši. Protože omezené vrcholy odpovídají lokálním extrémům skalárního pole, které mohou nabývat libovolné nenulové hodnoty, lze u každého z těchto vrcholů nastavit jakousi váhu. Čím vyšší bude tato hodnota (v absolutní hodnotě), tím větší vliv bude mít omezený vrchol na průběh skalárního pole. Pro jednoduchost však postačí, když budou mít všechny kladné extrémy váhu 1 a záporné extrémy váhu -1. Hodnoty přiřazené daným omezeným vrcholům se potom uloží na příslušné indexy vektoru odpovídajícího vektoru *b* rovnice 2-6.

Skalární pole

Skalární pole je řešením lineární soustavy 2-6. Protože velikost této soustavy odpovídá počtu vrcholů v síti, kterých může být řádově 1000–100 000, je zcela nezbytné použít nějaký optimalizovaný způsob řešení. Při sestavování matice L lze využít toho, že je ještě před nastavením extrémů symetrická a tedy stačí spočítat pouze jednu polovinu hodnot a druhou doplnit podle první poloviny. Pro samotný výpočet je potom důležité, že matice pro síť o n vrcholech, tedy velikosti $n \times n$, bude obsahovat maximálně $3n$ nenulových hodnot, což je nesrovnatelně méně, než počet všech hodnot n^2. Pro řešení řídkých matic poskytuje knihovna MDSTk funkci

```
solve(CSparseMatrix<double> &, CVector<double> &, CVector<double> &)
```

ve jmenném prostoru *mds::math*, která zvládne výpočet podstatně rychleji, než klasická Gaussova eliminační metoda.

Vektorová pole

Na základě skalárního pole se počítá gradientní a izoparametrické vektorové pole. Protože skalární pole představuje po částech lineární funkci, vektory těchto vektorových polí jsou neměnné v rámci každého trojúhelníka. Z toho důvodu stačí pro každý trojúhelník sítě spočítat a uchovat po jednom vektoru z obou vektorových polí.

Síť plovoucích linek

Počátek každé plovoucí linky je určen semínkem. Linka následně prochází trojúhelníky vstupní sítě ve směru vektorů jednoho z vektorových polí. Každá linka se skládá ze segmentů, které jsou spojeny uzly. Jednotlivé uzly leží na hranách, případně ve vrcholech či uvnitř trojúhelníka (pouze pokud se jedná o semínko, jehož poloha na povrchu tělesa může být zcela libovolná). Je nutné ošetřit všechny tři případy, protože v každém se postup linky řeší jinak:

1. Pokud leží uzel ve vrcholu, je nutné prohledat všechny okolní trojúhelníky a pro postup vybrat ten, ve kterém linka postoupí nejdále.
2. Nejčastěji leží uzel na hraně. Pokud se nejedná o semínko, linka pokračuje přes sousední trojúhelník, než kterým přichází k dané hraně. V případě semínka je nutné otestovat přilehlé dva trojúhelníky.
3. Pouze pokud je uzel semínko, může ležet uvnitř trojúhelníku. V tom případě se hledá cesta pouze uvnitř daného trojúhelníku.

Během hledání cesty plovoucí linky je nutné testovat vzdálenost od ostatních linek. K tomu je možné použít dočasné plovoucí linky ortogonálního toku, jejichž délka bude rovna polovině hodnoty udané distanční funkcí. Pokud se taková linka neprotne s žádnou linkou stejného toku, jako je vytvářená linka, může linka pokračovat dál. V opačném případě se jedná o přiblížení k jiné lince a tedy ukončení. Při vytváření izoparametrických linek se nesmí zapomenout na to, že linky mohou vytvořit smyčku a tedy konec musí být navázán na počátek, který leží obecně uvnitř trojúhelníku, ne na hraně jako konec linky.

Během hledání cesty každé plovoucí linky musejí být po obou stranách vytvářena semínka dalších linek. Vzdálenost udává distanční funkce, konkrétní umístění může být určeno opět pomocí dočasné plovoucí linky ortogonálního toku. Nová semínka se ukládají do fronty s prioritou podle vzdálenosti od linky, vedle které byla vytvořena (větší vzdálenost znamená větší prioritu). Počáteční

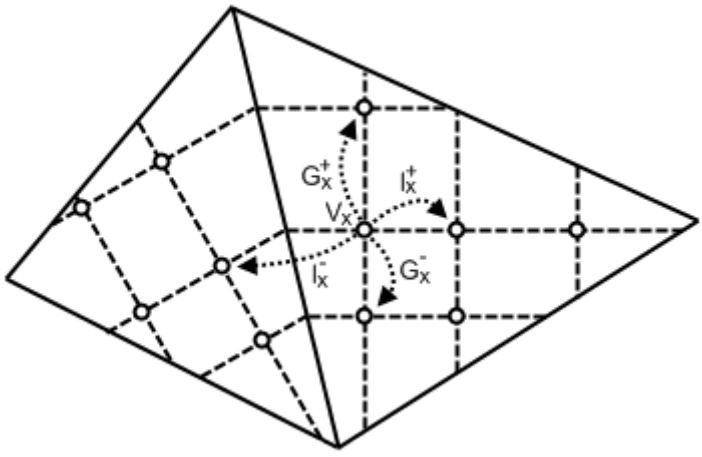
semínka se umísťují do omezených vrcholů a rohových vrcholů kostry a mají nejvyšší možnou prioritu. V případě izotropní sítě se používá fronta bez priority, což odpovídá nastavení stejné priority všem semínkům. Vytváření linek daného toku končí v okamžiku vyčerpání všech semínek.

3.3.4 Vytvoření nové sítě

Vrcholy a hrany nové sítě

Průsečíky plovoucích linek tvoří vrcholy nové sítě. K tomu, aby bylo možné vytvořit hrany mezi těmito vrcholy, je třeba během výpočtu uchovávat informaci o sousednosti vzniklých vrcholů. Toto lze provést pomocí ukazatelů. Jejich správné nastavení lze zajistit následovně:

1. Ke každému trojúhelníku asociovat pro gradientní a izoparametrický tok po jednom seznamu ukazatelů na segmenty plovoucích linek, které přes něj procházejí. Během vytváření plovoucích linek se uloží ukazatel na každý nově vytvořený segment do příslušného seznamu trojúhelníka odpovídajícího danému segmentu. Po vytvoření sítě linek tak bude známo, které segmenty linek leží uvnitř jakého trojúhelníka.
2. Pro každý trojúhelník zvlášť spočítat průsečíky plovoucích linek tak, že se vyřeší průsečíky všech dvojic segmentů gradientních a izoparametrických linek uložených ve dvojici seznamů asociovaných s daným trojúhelníkem.
3. Nalezené průsečíky musejí být správně seřazeny na příslušných plovoucích linkách podle místa výskytu. Toto lze zajistit seřazením průsečíků uvnitř každého segmentu zvlášť. Pořadí všech průsečíků linky potom podléhá pořadí segmentů linky.
4. Odpovídající nastavení ukazatelů mezi průsečíky lze realizovat iterací přes průsečíky každé linky zvlášť s tím, že se pro každý průsečík nastaví ukazatel na jeho předchůdce a následníka vzhledem k toku dané linky. Každý průsečík může mít svého předchůdce a následníka podél gradientního a izoparametrického toku, dohromady tedy čtyři sousední průsečíky.



Obrázek 3-8: *Znázornění gradietních a izoparametrických plovoucích linek, jejichž průsečíky se počítají zvlášť uvnitř každého trojúhelníka. Ve výsledku má libovolný vrchol V_x nastaveny ukazatele na okolní vrcholy ve směru a proti směru gradientního i izoparametrického toku ($G_x^{\pm}$, $I_x^{\pm}$).*

Polygony nové sítě

Polygony nové sítě jsou vymezeny smyčkami hran, tak jak ukazuje obrázek 2-9. Pro vytvoření všech polygonů je třeba nalézt všechny takové smyčky. Jestliže vytváříme levotočivé polygony, bude hrana spojující vrcholy V_j^i a V_{j+1}^i polygonu P_i vždy tou nejlevější možnou hranou. To znamená, že se bude při hledání vrcholů každého polygonu testovat, zda je nastaven ukazatel na sousední vrchol nejdříve ve směru doleva, potom dopředu a nakonec doprava oproti aktuálnímu směru. Pro vytvoření všech levotočivých polygonů je třeba vyhledat všechny levotočivé smyčky s počátkem v každém vrcholu sítě a počátečním směrem prohledávání v opačném směru oproti izoparametrickému (viz I_x^- na obrázku 3-8). V případě vytváření pravotočivých polygonů se postupuje analogicky, pouze počáteční směr bude v izoparametrickém směru (viz I_x^+ na obrázku 3-8) a bude se vybírat vždy ta nejpravější hrana. Protože každou hranu lze v daném směru použít pouze jednou, je vhodné každý použitý ukazatel vynulovat a tím zamezit opětovnému použití hrany.

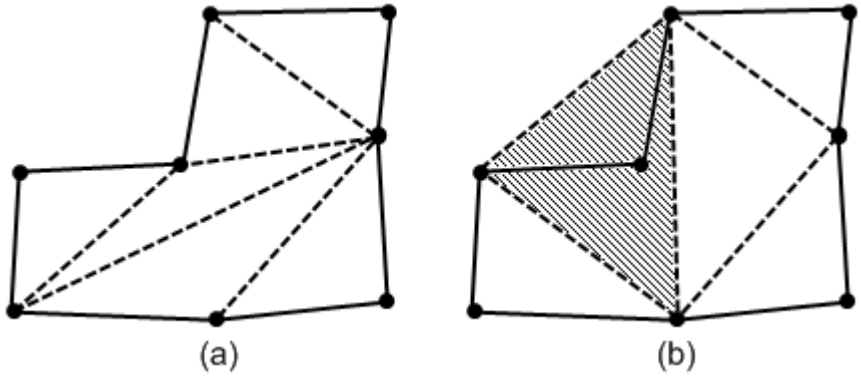
3.3.5 Teselace polygonů

Aby bylo možné výslednou quadrilaterální síť zobrazit nebo uložit do souboru ve formátu STL, je třeba provést teselaci polygonů na trojúhelníky. Pro tento účel jsem navrhnul a implementoval jiný algoritmus než ten, který je popsán v kapitole 2.3.1.

Moje řešení se skládá ze dvou částí. Nejprve je polygon o n vrcholech rozdělen na $n-2$ libovolných trojúhelníků a následně je toto rozdělení přeskupeno tak, aby měly jednotlivé trojúhelníky co největší kvalitu. Tento způsob sice není přímočarý, ale umožňuje lepší dekompozici problému. Totiž samotná teselace polygonu, jehož vrcholy neleží v rovině nýbrž na povrchu libovolně deformované plochy, není triviální záležitostí. Nalezení libovolného platného rozdělení je mnohem jednodušší. Takové rozdělení však nemusí být optimální, proto se provede přeskupení.

Rozdělení polygonu na trojúhelníky

Na počátečním rozdělení příliš nezáleží, musí být ale zaručeno, aby měly všechny trojúhelníky správnou orientaci. Při nahodilém rozdělení by mohly vzniknout překryvy, které by obsahovaly otočené trojúhelníky, jejichž normála by směřovala dovnitř tělesa. V důsledku toho by některé trojúhelníky nebyly vidět nebo by naopak přečnívaly přes hranice polygonu. Nejenom že toto způsobuje nehezké efekty při zobrazení, především to porušuje pravidla, která musejí být dodržena při práci s polygonálními sítěmi.



Obrázek 3-9: *Příklad (a) vhodného, (b) špatného rozdělení s překryvem a přesahem trojúhelníků.*

Konkrétní rozdělení probíhá tak, že jsou postupně odřezávány okrajové trojúhelníky. Algoritmus lze popsat následovně:

1. Nastav první vrchol polygonu jako aktuální.
2. Otestuj počet vrcholů polygonu.
   a/ Pokud se skládá z více než tří vrcholů, pokračuj krokem 3.
   b/ Pokud se skládá právě ze tří vrcholů, vytvoř z nich trojúhelník a ukonči dělení.
3. Otestuj, zda lze odříznout trojúhelník *ABC* (viz kapitola 3.3.5 – Odříznutí trojúhelníka), kde *A* je aktuální vrchol, *B* následuje za *A* a *C* předchází *A*.
   a/ Pokud lze, odřízni trojúhelník *ABC*, nastav vrchol následující za *B* jako aktuální a pokračuj krokem 2.
   b/ Pokud nelze, nastav vrchol *B* jako aktuální a pokračuj krokem 2.

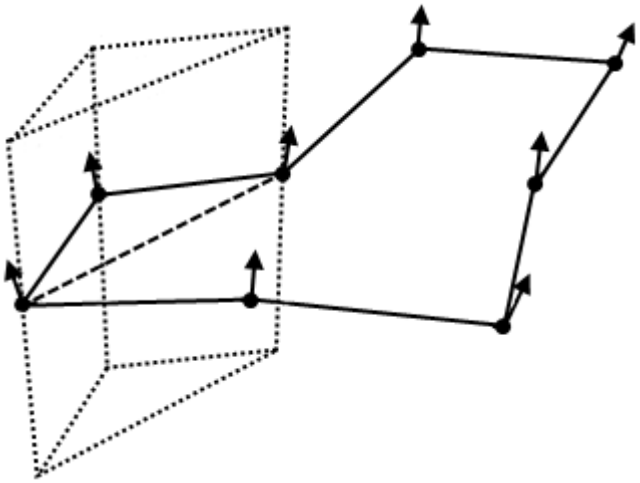
Odříznutí trojúhelníka

Aby mohl být trojúhelník odříznut od polygonu, musí být splněny tyto dvě podmínky:

1. Normála musí mít správnou orientaci.
2. Trojúhelník nesmí zakrývat žádný vrchol.

Pokud máme referenční vektor, který určuje orientaci povrchu v dané oblasti, lze orientaci normály zjistit pomocí skalárního součinu s tímto vektorem (za předpokladu, že jsou oba vektory normalizované). Kladná hodnota znamená, že normála směřuje do stejného poloprostoru jako referenční vektor. Protože každý z vrcholů trojúhelníka leží uvnitř nějakého trojúhelníka původní sítě, lze referenční vektor spočítat jako normalizovaný součet normálových vektorů trojúhelníků odpovídajících jednotlivým vrcholům.

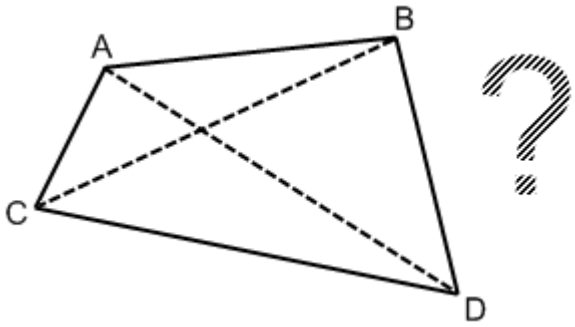
Druhou podmínku lze vcelku jednoduše otestovat v rovině tak, že se pro každý vrchol polygonu zjistí, zda leží uvnitř nebo vně trojúhelníka. V prostoru toto samozřejmě nelze provést, protože všechny vrcholy polygonu neleží ve stejné rovině jako vrcholy testovaného trojúhelníka. Z toho důvodu musíme namísto s trojúhelníkem pracovat s trojbokým hranolem. Stěny takového hranolu mohou být vymezeny rovinami, které procházejí vrcholy trojúhelníka a jejich normálový vektor je určen jako vektorový součin normálových vektorů ve vrcholech.



Obrázek 3-10: *Znázornění trojúhelníka, který má být odříznut od polygonu. Normálové vektory ve vrcholech polygonu jsou získány z trojúhelníků původní sítě a slouží k výpočtu normál rovin vymezujících trojboký hranol, uvnitř kterého nesmí ležet žádné další vrcholy polygonu, aby mohl být daný trojúhelník odříznut.*

Přerozdělení získaných trojúhelníků

Přerozdělení probíhá iterativně, dokud lze vylepšit kvalitu některého trojúhelníka. Pro každý trojúhelník *ABC* a jeho sousední trojúhelník *BDC*, kde *BC* je nejdelší hrana trojúhelníka *ABC*, se testuje, zda nahrazením těchto trojúhelníků trojúhelníky *ABD* a *ADC* nevznikne vhodnější rozdělení. Vhodnost rozdělení se posuzuje podle kvality trojúhelníků. Pokud je kvalita obou nových trojúhelníků lepší než kvalita hůře ohodnoceného trojúhelníka z původní dvojice, proběhne jejich nahrazení. V důsledku toho může být ohodnocení obou nových trojúhelníků horší než původního lépe ohodnoceného trojúhelníka, nebude ale horší než kvalita původního hůře ohodnoceného trojúhelníka. Tímto způsobem se vlastně nahradí nejhůře ohodnocené trojúhelníky lépe ohodnocenými na úkor náhrady těch nejlépe ohodnocených hůře ohodnocenými, čímž se zmenší rozptyl kvalit jednotlivých trojúhelníků.



Obrázek 3-11: *Výběr vhodnějšího rozdělení: ABC-BDC nebo ABD-ADC.*

3.4 Vytvoření 3D spline plochy

Kapitola 3.3 popisuje implementaci aplikace, která umožňuje vytvoření quadrilaterální sítě a její následnou teselaci na trojúhelníky. Teselace je pro některé účely nezbytná, pro převod na 3D spline plochu však nevhodná. Jak je uvedeno v rámci teoretického rozboru v kapitole 2.5, 3D spline plochy obvykle používají síť řídících bodů uspořádaných v obdélníkové mřížce. Této struktuře právě odpovídá quadrilaterální síť s tím rozdílem, že obsahuje T-zakončení. Tento problém by bylo možné určitým způsobem řešit (např. tak, jak je popsáno v kapitole 2.2.11), ve skutečnosti to ale není nutné. Pro další postup jsem se totiž rozhodnul pro převod na T-spline plochu, která přítomnost T-zakončení dovoluje.

Pro práci s T-spline plochami existují nástroje od firmy T-Splines, Inc. (viz [T-splines09]). Jednou z možností je modul pro modelovací program Rhinoceros. Z toho důvodu je třeba přenést quadrilaterální síť do tohoto prostředí, ve kterém lze T-spline plochu vytvořit. Jako nejvhodnější způsob se jeví export dat ve formátu OBJ.

3.4.1 Export ve formátu OBJ

Informace o formátu OBJ jsem čerpal z [Riggs05]. Jedná se o jednoduchý textový formát pro přenos 3D dat. Počáteční znak každého řádku udává typ obsažené informace, následují hodnoty oddělené mezerami. Pro naše potřeby vystačíme s konstrukcemi uvedenými v tabulce 3-1.

Tabulka 3-1: *Nejdůležitější konstrukce formátu OBJ.*

Konstrukce	***Použití***
`# comment`	Komentář
`o object`	Název objektu
`v x y z`	Souřadnice vrcholu
`vt u v [w]`	Texturovací souřadnice
`vn x y z`	Normálový vektor
`f v1[/vt1][/vn1] v2[/vt2][/vn2] v3[/vt3][/vn3] ...`	Popis polygonu – indexy souřadnice vrcholů, texturovacích souřadnic a normálových vektorů

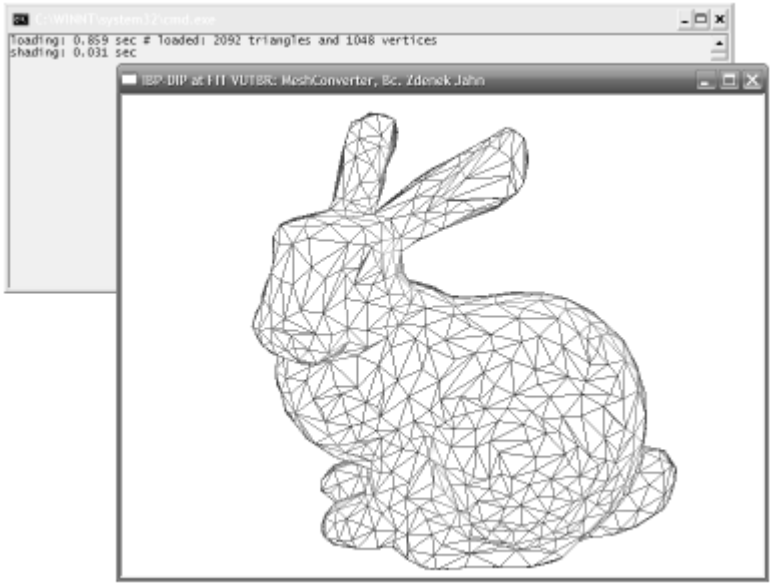
Na začátku souboru bývá obvykle v rámci komentáře uveden název zdroje dat. Pro definování geometrie tělesa je třeba uvést bloky se souřadnicemi vrcholů, volitelně textur a normál, následované blokem polygonů, který se prostřednictvím indexů (počítáno od 1) odkazuje na uvedené souřadnice. Z hlediska rozeznání jednotlivých objektů uložených v souboru je třeba před definicí každého objektu uvést jeho název. Experimentální cestou jsem také došel k tomu, že je vhodné nastavit pevný počet desetinných míst pro ukládání reálných čísel (např. 6), protože některé programy mají jinak problém s načtením.

4 Výsledky

Protože tato práce navazuje na mou předchozí bakalářskou práci, již na počátku jsem měl funkční aplikaci, která umožňovala převod nestrukturované trojúhelníkové sítě na síť strukturovanou quadrilaterální. Tento převod měl však jisté nedostatky, které by bylo vhodné odstranit. Z počátku jsem se tedy věnoval vylepšení stávající funkcionality a následně jsem přistoupil k řešení problému vytvoření 3D spline plochy.

4.1 Aplikace

Aplikace se skládá z jednoho konzolového a jednoho grafického okna. V rámci prvního okna jsou zobrazovány informativní výpisy, především co se týče délky výpočtu jednotlivých kroků převodu. Druhé okno slouží pro zobrazení a manipulaci s modelem. Na počátku je možné interaktivně určit omezené vrcholy, během převodu potom lze zobrazit jednotlivé mezikroky v odpovídající formě.



Obrázek 4-1: *Výsledná aplikace.*

Oproti původní aplikaci bylo poupraveno ovládání. Většina možností, které se nastavovaly při spuštění prostřednictvím parametrů, byla nahrazena přepínáním pomocí kláves za běhu aplikace. Bližší popis ovládání je k nalezení v příloze A-4.

4.2 Konstrukce quadrilaterální sítě

Konstrukce quadrilaterální sítě sestává z několika kroků. Původní proces zůstal zachován, některé jeho části však byly optimalizovány nebo poupraveny. Důležitými rozšířeními, na kterých jsem pracoval, byla teselace polygonů a vytváření kostry, jejíž účel je zvýšení kvality výsledné quadrilaterální sítě.

4.2.1 Omezené vrcholy

Výběr omezených vrcholů, které slouží pro přizpůsobení toku skalárního pole potažmo tvaru výsledné sítě, je plně v uživatelské režii. Prostřednictvím myši lze vybrat konkrétní vrcholy a určit, zda budou představovat maximum nebo minimum skalárního pole.

4.2.2 Skalární pole

Skalární pole je řešením lineární soustavy, jejíž velikost odpovídá počtu vrcholů sítě. Klasickou Gaussovou eliminační metodou trvá výpočet takové soustavy značně dlouho již pro malé polygonální sítě. Pro větší není možný jak z časového, tak prostorového hlediska. Díky tomu, že je řešená matice řídká, bylo možné použít optimalizovanou metodu, která výpočet zvládá dostatečně rychle s přiměřenými paměťovými nároky.

4.2.3 Vektorová pole

Výpočet gradientního a izoparametrického vektorového pole je triviální záležitostí, která spočívá v řešení jednoduchých lineárních soustav pro každý trojúhelník sítě zvlášť. Výpočet je tedy značně rychlý i pro větší sítě.

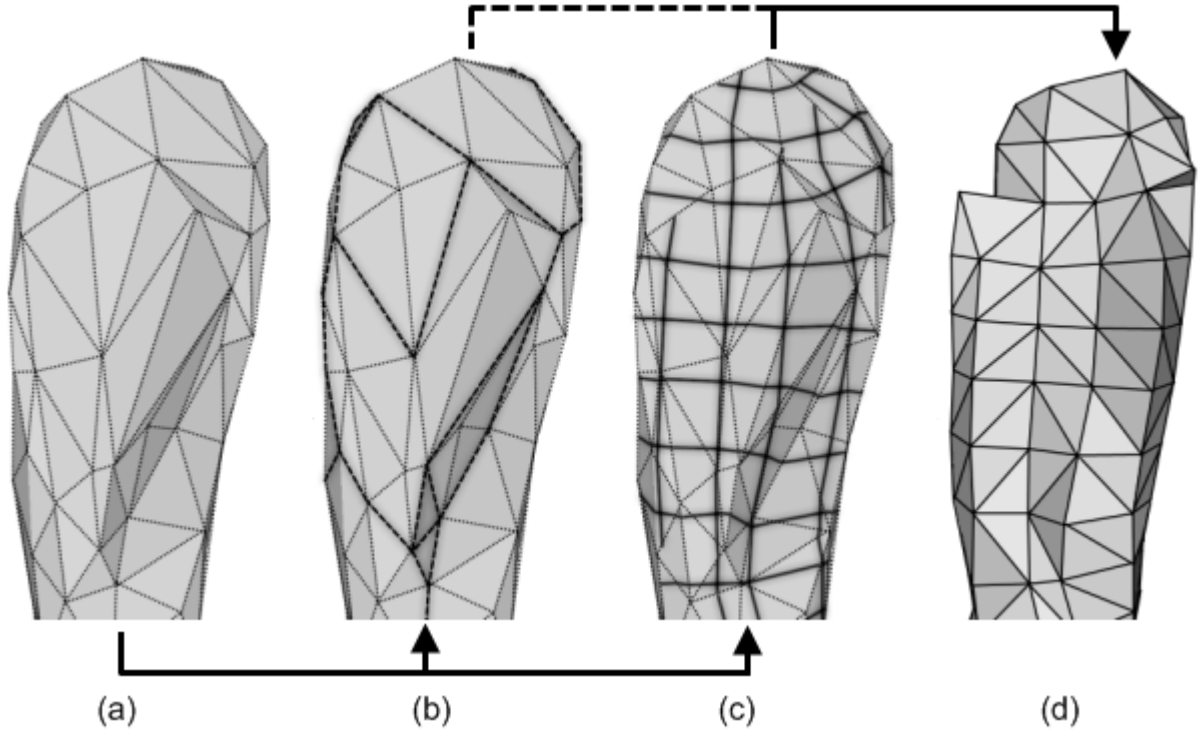
4.2.4 Síť plovoucích linek

Vytvoření plovoucích linek je asi nejnáročnější krok převodu jak implementačně tak výpočetně. Je třeba zajistit kontrolu řady stavů, tak aby se program nezaseknul v nekonečné smyčce a aby výsledek odpovídal požadovaným parametrům.

1. Je třeba nalézt cestu plovoucí linky podél jednoho z vektorových polí. K tomu je třeba rozlišit situace popisované v kapitole 2.2.6.
2. Je třeba zajistit správné ukončení linky v případě, že:
   a/ Není kam dál pokračovat.
   b/ Linka se přiblíží k jiné lince na poloviční vzdálenost udávanou distanční funkcí (vzdálenost se určuje podél sítě ve směru ortogonálního vektorového pole).
   c/ Izoparametrická linka vytvoří smyčku.
3. Vygenerovat semínka dalších linek ve správné vzdálenosti od linky.

4.2.5 Kostra

Účelem kostry je nalezení význačných linií modelu, které je nutné přenést do výsledné sítě, jinak může dojít k degradaci tvaru. Pro výběr hran tvořících kostru jsem použil upravenou metodu rozvodí používanou především v oblasti segmentace obrazu. Jak je vidět na obrázku 4-2b, prostřednictvím tohoto algoritmu je možné nalézt řadu hran vhodných pro uchování, získaná kostra však obsahuje také další hrany, které nejsou relevantní. Z důvodů, které popisuji v kapitole 5.2.1 - Kostra, v současné implementaci nejsou ve výsledné síti zahrnuty hrany kostry a proto v případě některých modelů stále k degradaci tvaru dochází, což je vidět na obrázku 4-2d.



Obrázek 4-2: *Model ucha zajíce. Na základě (a) vstupní sítě se vytvoří (b) kostra a (c) síť plovoucích linek. Protože (d) výsledná síť neobsahuje hrany kostry, dochází k degradaci tvaru..*

4.2.6 Vytvoření quadrilaterální sítě

Výpočet průsečíků plovoucích linek, které představují vrcholy quadrilaterální sítě, probíhá v rámci každého trojúhelníka vstupní sítě zvlášť. Díky tomu se značně snižuje kombinatorická složitost a výpočet je tedy velice efektivní. Stejného principu využívá nalezení sousedících vrcholů, jejichž pořadí je odvozeno z pozice v rámci segmentů plovoucích linek a odpovídající zřetězení ukazateli lze vyhodnotit opět velice rychle, narozdíl od předchozí implementace, která používala řazení v rámci linky jako jednoho celku. Vytvoření výsledných polygonů na základě daných vrcholů propojených ukazateli podle sousednosti potom představuje relativně jednoduchou operaci. Pouze okolí pólů představují komplikovaná místa, ale díky správnému průchodu grafem vrcholů jsou pokryta i tato.

4.2.7 Teselace polygonů quadrilaterální sítě

Původní implementace na úrovni bakalářské práce používala pro teselaci polygonů šablony, v důsledku čehož vznikaly ve výsledné síti díry. Každá šablona odpovídala určité konfiguraci polygonu, tímto způsobem ale nešlo pokrýt všechny případy. Aktuální implementace používá odlišný algoritmus, který by měl vytvořit vhodnou teselaci pro libovolný polygon, jaký se může v síti vyskytovat. Přestože se jedná o algoritmus, který iterativně vylepšuje původní teselaci, dokud není dosažena taková, kterou již nelze vylepšit, výpočet probíhá docela rychle.

4.3 Konstrukce 3D spline plochy

Přímo v aplikaci, kterou jsem implementoval, nedochází k vytváření žádné 3D spline plochy. Z hlediska využití by to možná ani nemělo význam. Protože je potřeba s modelem ve formě 3D spline plochy dále pracovat, aplikace umí exportovat quadrilaterální síť do souboru ve formátu OBJ. Tento soubor je potom možné otevřít např. v modelovacím programu Rhinoceros, ve kterém lze prostřednictvím speciálního modulu vytvořit T-spline plochu a dále s ní manipulovat. Tento postup funguje pouze s jednou menší komplikací. Výstupní síť obsahuje jisté množství polygonů, které se skládají z více než čtyř vrcholů. Tyto polygony potom v aplikaci Rhinoceros způsobují problém při vytvoření T-spline plochy. Proto je nutné tyto polygony teselovat, což jsem zatím vyřešil načtením a opětovným uložením aplikací Blender.

Obrázek 4-3: *Lišta modulu pro práci s T-spline plochami v aplikaci Rhinoceros. Tlačítko pro (a) načtení souboru ve formátu OBJ, (b) přepnutí mezi zobrazením sítě řídících bodů a 3D spline plochy.*

4.4 Příklady

4.4.1 Ukázkové modely

Pro předvedení převodu trojúhelníkové sítě na quadrilaterální a následně na T-spline plochu jsem vybral 5 ukázkových modelů. Průběh převodu jednotlivých modelů je vyobrazen v příloze B. Parametry jednotlivých modelů jsou uvedeny v tabulce 4-1.

Tabulka 4-1: *Ukázkové modely.*

	Bunny	Femur	Lopatka	Koleno	Býček
Trojúhelníků	2 092	4 998	9 998	10 908	29 998
Vrcholů	1 048	2 501	4 999	5 456	15 001

Bunny

Model zajíce. Jako omezené vrcholy je vhodné vybrat špičky uší a čenich, které budou představovat maxima skalárního pole, a střed ocásku, který bude minimem. Skalární pole tak dobře pokryje celý model včetně uší, viz obrázky B-1-2 a B-1-3. Pouze místo mezi ušima a na čele představuje problém, protože zde dochází ke zlomu toku skalárního pole a síť plovoucích linek tudy neprochází plynule, viz obrázek B-1-7. Model obsahuje určité linie, které by bylo vhodné zachovat. Jak je vidět na obrázku B-1-6, některé z těchto linií jsou součástí kostry, některé však chybí a kostra naopak obsahuje určité hrany zbytečně. Celý převod je vyobrazen v příloze B-1.

Femur

Model části stehenní kosti s hlavicí kyčelního kloubu. Z ukázkových modelů je asi tím nejjednodušším pro převod, neobsahuje totiž žádná komplikovaná místa či hrany, které by měly být

zachovány. Nejvhodnější výběr omezených vrcholů je pravděpodobně maximum umístěné uprostřed hlavice femuru a minimum na vrcholu přilehlého výčnělku, viz obrázky B-2-2 a B-2-3. Ukázky z převodu zachycuje příloha B-2.

Ilium

Model lopatky kyčelního kloubu. Z hlediska výběru omezených vrcholů se jedná o komplikovanější model. Jako nejoptimálnější variantu jsem shledal umístit maximum na horní hraně lopatky a minimum uvnitř jamky, do které zapadá hlavice stehenního kloubu. Skalární pole tak vychází z jamky, na cestě k horní hraně lopatky dobře pokrývá všechny horní plochy a také respektuje tvar kosti v okolí otvoru v dolní části. Toto je vidět na obrázku B-3-3 a především B-3-7 na průběhu plovoucích linek (vrch kosti je na obrázku umístěn vlevo, spodek vpravo). Kostra modelu vcelku dobře pokrývá většinu význačných linií kosti, jmenovitě horní a dolní hranu a obvod jamky a otvoru, což je vidět na obrázku B-3-5. Ukázky z převodu zachycuje příloha B-3.

Koleno

Model části stehenní kosti, jež tvoří horní část kolenního kloubu. U tohoto modelu lze nalézt více relevantních možností pro výběr omezených vrcholů. Na obrázcích B-4-2 a B-4-3 je vidět umístění dvojice maxim na zadních výstupcích dolní části kosti a jednoho minima úplně nahoře. Takto je zajištěno dobré pokrytí modelu sítí plovoucích linek, viz obrázek B-4-7, způsobuje však vznik kritického místa ve vnitřní části kloubu, což je také vidět na toku skalárního pole. Kostra dobře zachycuje horní hranu kosti a několik menších útvarů, které mají ostré okraje, viz obrázek B-4-6. Výsledná T-spline plocha, která je vidět na obrázcích B-4-8 a B-4-9, byla vytvořena na základě sítě, u které byly vybrány pouze dva omezené vrcholy – maximum uvnitř kloubní jamky, minimum opět úplně nahoře.

Býček

Model hliněného býčka. Jedná se o nejsložitější ukázkový model z hlediska počtu vrcholů a trojúhelníků. Na tomto modelu lze předvést výběr více omezených vrcholů s cílem dobrého pokrytí řady výstupků. Jak ukazují obrázky B-5-2 a B-5-3, maxima jsou umístěna na rozích a uších, minima potom ze spodu kopyt. V důsledku výběru více omezených vrcholů však vzniká řada kritických bodů, které můžeme vidět např. na temeni a mezi uchem a rohem. Na obrázku B-5-5 je vidět kostra modelu, která dobře vystihuje význačné linie kolem očí, tlamy a v oblasti kopyt, poněkud hůře potom na rozích a uších. Obrázky B-5-7 a B-5-8 ukazují, jak lze díky různé hustotě plovoucích linek vytvářet síť s různou úrovní detailu. Výslednou T-spline plochu se bohužel nepodařilo vytvořit.

4.4.2 Rychlost převodu

Během převodu se do konzolového okna vypisují délky trvání jednotlivých kroků. Změřené časy jsou uvedeny v tabulce 4-2. Na základě znalosti implementace lze v souladu se zjištěnými časy určit, že výpočet křivostí, výpočet skalárního pole a vytvoření kostry má lineární časovou složitost vzhledem k počtu vrcholů, výpočet vektorových polí má lineární časovou složitost vzhledem k počtu trojúhelníků a načtení vstupu z STL souboru lineární časovou složitost vzhledem k počtu vrcholů

a trojúhelníků dohromady. Ostatní kroky jsou ovlivněny řadou dalších faktorů a nelze tedy jednoduše určit kategorii časové složitosti závisející na vstupních datech.

Tabulka 4-2: *Tabulka časů strávených během jednotlivých fází převodu trojúhelníkové 3D sítě na quadrilaterální (hodnoty jsou uvedeny v sekundách).*

		Bunny	Femur	Ilium	Koleno	Býček
1.	Načtení vstupu z STL	0,641	0,781	1,594	1,813	6,109
2.	Výpočet křivostí	0,297	0,687	1,406	1,532	4,234
3.	Výpočet skalárního pole	2,047	4,531	9,094	9,781	32,204
4.	Výpočet vektorových polí	0,078	0,109	0,188	0,204	0,609
5.	Vytvoření kostry	0,313	0,750	1,422	1,719	4,500
6.	Vytvoření sítě plovoucích linek	11,844	15,797	37,781	61,250	100,547
7.	Výpočet průsečíků linek	1,203	1,250	3,500	5,875	5,657
8.	Vytvoření polygonů	0,610	0,281	0,922	2,422	1,344
9.	Teselace na trojúhelníky	0,609	0,266	0,890	2,328	1,234
10.	Uložení výstupu do OBJ	0,250	0,125	0,421	1,125	0,594
	Celkem	17,892	24,577	57,218	88,049	157,032

Dále jsem také přibližně změřil, jak dlouho trvá vytvoření sítě řídících bodů a následný výpočet T-spline plochy v aplikaci Rhinoceros. Zjištěné časy jsou uvedeny v tabulce 4-3.

Tabulka 4-3: *Tabulka časů strávených během vytváření T-spline plochy v aplikaci Rhinoceros (hodnoty jsou uvedeny ve formátu minuty:sekundy).*

		Bunny	Femur	Ilium	Koleno	Býček
1.	Vytvoření sítě řídících bodů	0:06	0:03	0:11	0:06	-
2.	Výpočet T-spline plochy	1:36	5:45	2:45	1:23	-

5 Závěr

Téma této práce je značně široké a lze tedy zpracovávat delší dobu. Původně jsem na tomto tématu začal pracovat v rámci bakalářské práce a po úspěšném obhájení jsem navázal touto diplomovou prací. Za celou dobu jsem toho zvládnul vcelku dost nastudovat a implementovat. Přesto je stále řada věcí, které by stálo za to dodělat či vylepšit. Současná aplikace zvládá převod nestrukturované trojúhelníkové sítě na strukturovanou quadrilaterální, umožňuje nastavit různé parametry a vizualizovat jednotlivé kroky převodu. Výstupem tohoto převodu je soubor ve formátu OBJ, který lze následně použít pro vytvoření T-Spline plochy.

5.1 Osobní přínos

Co se týče mého přínosu je to implementace nové a dle mého názoru poměrně složité metody pro transformaci polygonálních sítí, která by mohla mít i reálné využití např. v medicínských aplikacích. Kýženým cílem je mít k dispozici sadu (polo)automatických nástrojů, jejichž prostřednictvím by bylo možné vytvořit z nasnímaných modelů kostí či jiných tělesných struktur reprezentaci, která by byla vhodná pro zpracování v některém z běžně dostupných modelovacích programů. Jak již bylo zmíněno v úvodu, převodem dekompozičního modelu, který je vhodný pro uložení nasnímaných 3D dat, získáme nestrukturovanou trojúhelníkovou 3D síť. Prostřednictvím aplikace, kterou jsem implementoval, je možné tuto síť převést do formy, kterou lze již existujícími prostředky převést na mnohem lépe použitelnou T-spline plochu.

5.2 Další vývoj

Myslím, že současné řešení slouží jako dobrý základ pro další vývoj reálně použitelného nástroje, zbývá však ještě dost práce. Některé aspekty převodu musely ustoupit v zájmu dosažení hlavního cíle, vytváření 3D spline plochy, některé se nepodařilo implementovat zcela uspokojivě. S odstupem mě také napadají různá vylepšení nebo nové přístupy řešení již vyřešených problémů.

5.2.1 Nedostatky

Kostra

Současná implementace používá pro aproximaci křivostí metriku, která bere v potaz pouze bezprostřední okolí vrcholu. Jak můžeme vidět na obrázcích B-1-5, B-2-5, B-3-4, B-4-5 a B-5-6, křivosti se mění nahodile vrchol od vrcholu. V důsledku toho vyhledání hran kostry není vždy optimální. Řada hran zachycuje nezřetelné výčnělky, které však mají lokálně velkou křivost. Naopak některé linie nejsou v kostře zahrnuty, přestože je křivost v dílčích vrcholech velká. Rozptyl hodnot v sousedících vrcholech totiž zamezuje vzniku jednoznačné linie. Pro řešení tohoto problému je tedy při výpočtu křivostí nutné zohlednit větší oblast.

Co se týče metody rozvodí použité pro detekci hran, nejsem si zcela jistý vhodností této volby. S tímto algoritmem totiž souvisí problém v podobě nadsegmentace, který se mi podařilo vyřešit jen

částečně. Pomocí prahování lze odstranit vznik množství segmentů v oblastech s malou křivostí. Určení vhodného prahu však není zcela jednoduché, navíc často nemusí být univerzální pro celý model. Vhodnější by mohlo být hierarchické slučování segmentů, které by lépe postihlo stěžejní segmenty a hranice mezi nimi. Další záležitost, která stojí za zvážení, představuje struktura vytvořené kostry. Prostřednictvím metody rozvodí jsou vytvořeny pouze řetězy hran, které začínají i končí v rozích, zatímco řetězy končící šipkami nevzniknou. Tento jev ale nemusí být vždy žádoucí. Kostra by měla být spíše minimalistická a z toho důvodu by měla obsahovat pouze hrany, které korespondují s význačnými liniemi. Toto by se případně dalo vyřešit dodatečným výběrem relevantních hran např. na základě zvoleného prahu maximální či střední křivosti.

Z důvodu popisovaných problémů nemá aktuálně vytvářená kostra požadované vlastnosti, proto ji také nelze použít pro začlenění do výsledné quadrilaterální sítě. Vyřešení daných problémů a použití kostry by značně vylepšilo kvalitu výsledných modelů. Toto lze nejlépe ukázat na modelu býčka v příloze B-5. Na obrázku B-5-1 jsou vidět zřetelné oči, které ve výsledné síti zanikají, jak můžeme vidět na obrázku B-5-8. Stejně tak tvar uší a kopyt neodpovídá vzorovému.

Plovoucí linky

Implementace vytváření sítě linek je poměrně robustní a představuje asi nejsložitější část programu. Přesto se dají nalézt případy, ve kterých program selže. Největší problém představuje zacyklení, kdy linka „zabloudí“ např. vinou nepředvídaného stavu výpočtu a podmínky jejího ukončení potom nejsou nikdy dosaženy.

Výsledná síť

Výsledkem převodu je quadrilaterální síť, kterou je možné uložit do souboru ve formátu OBJ nebo teselovat na trojúhelníky a zobrazit na grafickém výstupu. Quadrilaterální síť však obsahuje určité odchylky od quadrilaterální struktury. V okolí T-zakončení a omezených vrcholů vznikají polygony s více než čtyřmi vrcholy, které by měly být vhodně rozděleny na trojúhelníky a quadrilaterály. To, že se výsledná síť neskládá ryze z quadrilaterálů případně trojúhelníků způsobuje zmiňovaný problém při načítání generovaného OBJ souboru aplikací Rhinoceros.

5.2.2 Rozšíření

Výběr omezených vrcholů

Omezené vrcholy se dají vybrat ručně, což na jedné straně umožňuje uživateli značně ovlivnit výsledek, na druhé straně může být zdrojem mnoha chyb. Pro správné určení omezených vrcholů a hodnot v nich ležících extrémů lze navrhnout několik různých postupů, které mohou být polo nebo zcela automatické. Lze vycházet např. z toho, že se jako omezené vrcholy obvykle vybírají špičky výčnělků, ve kterých budou křivosti nabývat lokálně extrémních hodnot.

Anizotropní síť

Obrázky B-1-7 a B-1-8 dobře vystihují slabinu izotropních sítí. Vzdálenost linek je zdola omezena nastavenou konstantou, v důsledku čehož vzniká ve velkých plochách zbytečně velké množství quadrilaterálů, naopak v tvarově komplikovaných místech jich může být nedostatek. Danou konstantu můžeme nastavit tak, že se odstraní jeden z těchto problémů, ne oba zároveň. Čím bude vzdálenost

linek menší, tím lépe vystihneme detaily, získáme tak ale složitější model, a naopak. Vytvoření anizotropní sítě by bylo velkým přínosem pro dobrou strukturovanost výsledné sítě potažmo sítě řídících bodů následně vytvořené T-spline plochy, čímž by se mnohem lépe využil její potenciál.

Skupiny objektů

Některé modely se mohou skládat z několika dílčích částí (např. klouby). Z hlediska daného účelu aplikace je počítáno pouze s jedním celkem, případné další části se ignorují. Reálně použitelná aplikace by ale mohla tyto části umět rozpoznat a umožnit jejich souběžné zpracování.

5.3 Shrnutí

Řešení daných problémů někdy nebylo zcela jednoduché. Popisovaná metoda pro převod trojúhelníkové sítě na quadrilaterální má zcela určitě funkční implementaci, kterou jsem však neměl k dispozici. Teoretické podklady byly vesměs srozumitelné a dobře vysvětlené, ale při implementaci se potom objevil určitý problém, jehož řešení nebylo až tak zřejmé. Občas existovalo několik možných přístupů, z nichž jsem si mohl vybrat, jindy jsem měl pocit, že řešení neexistuje. Za zmínku stojí, že se některé programové konstrukce velice špatně ladily, protože jedinou možností posouzení problému bylo implementovat další kód pro jeho vizualizaci, což samozřejmě vedlo k dalšímu ladění. Navzdory tomu mě tato práce bavila. Naučil jsem se hodně co se týče programování v jazyce C++, použití cizích programových knihoven a programování aplikace pro práci s polygonálními sítěmi. Přestože byl často problém najít dostatek času pro řešení, věřím, že se mi podařil udělat velký kus práce, a také doufám, že výsledek této práce nalezne své uplatnění.

Literatura

[Academia93] Kolektiv autorů: *Pravidla českého pravopisu*. Academia, Praha, 1993, ISBN 80-200-0475-0.

[Alliez03] Alliez, P., Cohen-Steiner D., Devillers O., Levy B., Desbrun M.: *Anisotropic Polygonal Remeshing.* ACM Transactions on Graphics, 2003, str. 485–493. Dostupné na URL: ftp://ftp-sop.inria.fr/prisme/alliez/anisotropic.pdf (duben 2007).

[Garland04] Garland, M., Ni, X., Hart, J. C.: *Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh.* ACM Transactions on Graphics, 2004, 23(3), str. 613–622. Dostupné na URL: http://graphics.cs.uiuc.edu/~jch/papers/morsecut.pdf (duben 2007).

[Garland05] Garland, M., Dong, S., Kircher, S.: *Harmonic Functions for Quadrilateral Remeshing of Arbitrary Manifolds.* Computer Aided Geometry Design, Special Issue on Geometry Processing, 2005, 22(5), str. 392–423, 2005. Dostupné na URL: http://mgarland.org/files/papers/harmonic-preprint.pdf (duben 2007).

[Hulík08] Hulík, R.: *Detekce zubů na 3D počítačovém polygonálním modelu čelisti*. Bakalářská práce, Brno, FIT VUT v Brně, 2008, str. 8, 9, 13, 14. Dostupné na URL: http://www.fit.vutbr.cz/study/DP/rpfile.php?id=5311 (květen 2009).

[Jahn07] Jahn, Z.: *Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy*. Bakalářská práce, Brno, FIT VUT v Brně, 2007. Dostupné na URL: http://www.fit.vutbr.cz/study/DP/rpfile.php?id=4022 (květen 2009).

[Jahn09] Jahn, Z.: *Převod trojúhelníkových polygonálních 3D sítí na 3D spline plochy*. Semestrální projekt, Brno, FIT VUT v Brně, 2009.

[Kim02] Kim, S. J., Kim, Ch. H., Levin, D.: *Surface simplification using a discrete curvature norm*. Computer and Graphics 26, 2002, str. 658-659. Dostupné na URL: http://kucg.korea.ac.kr/~sjkim/paper/cag2002.pdf (duben 2008).

[Kršek08] Kršek, P., Španěl, M.: *Křivky v počítačové grafice*. Přednášky IZG, Brno, FIT VUT v Brně, 2008. Dostupné na URL: https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IZG-IT/lectures/izg_slide_krivky_print.pdf (květen 2009).

[Owen99] Owen, G. S.: *Marching Cubes*. ACM SIGGRAPH – archival website, 16. dubna 1999. Dostupné na URL: http://old.siggraph.org/education/materials/HyperVis/vistech/volume/surface4.htm (květen 2009).

[Rábová08] Rábová, Z., Hanáček, P., Peringer, P., Přikryl, P., Křena, B.: *Užitečné rady pro psaní odborného textu.* Brno, 1993-2008. Dostupné na URL: http://www.fit.vutbr.cz/info/statnice/psani_textu.html (květen 2009).

[Rypl05] Rypl, D., Bittnar, Z.: *Triangulation of 3D Surfaces Described by Stereolithography Files*. ČVUT v Praze, 5. listopadu 2005. Dostupné na URL: http://mech.fsv.cvut.cz/~dr/papers/Lisbon04/lisbon04.html (květen 2009).

[Riggs05] Riggs, R.: *OBJ File Format.* Dostupné na URL: http://www.royriggs.com/obj.html (květen 2009).

[Sochor96] Sochor, J., Žára, J., Beneš, B.: *Algoritmy počítačové grafiky*. Skripta ČVUT, 1996, ISBN 80-01-01406-1.

[T-Splines09] *T-Splines technology*. T-Splines inc., 2009. Dostupné na URL: http://www.tsplines.com/about/technology.html (květen 2009).

[Weisstein07] Weisstein, E. W.: *Kronecker Delta*. MathWorld - A Wolfram Web Resource. Dostupné na URL: http://mathworld.wolfram.com/KroneckerDelta.html (květen 2007).

[Weisstein09a] Weisstein, E. W.: *Curvature.* MathWorld – A Wolfram Web Resource. Dostupné na URL: http://mathworld.wolfram.com/Curvature.html (květen 2009).

[Weisstein09b] Weisstein, E. W.: *Spline*. MathWorld – A Wolfram Web Resource. Dostupné na URL: http://mathworld.wolfram.com/Spline.html (květen 2009).

[Weisstein09c] Weisstein, E. W.: *NURBS Curve*. MathWorld – A Wolfram Web Resource. Dostupné na URL: http://mathworld.wolfram.com/NURBSCurve.html (květen 2009).

[Weisstein09d] Weisstein, E. W.: *NURBS Surface*. MathWorld – A Wolfram Web Resource. Dostupné na URL: http://mathworld.wolfram.com/NURBSSurface.html (květen 2009).

Seznam příloh

Příloha A – manuál aplikace
Příloha B – grafické výstupy
Příloha C – datový nosič CD

1. Zdrojové kódy programu.
2. Dokumentace zdrojových kódů.
3. Testovací modely.
4. Technická zpráva ve formátu DOC a PDF.

Příloha A Manuál aplikace

A-1 Kompilace

Program je implementován v jazyce C++ podle normy ISO C++ 98 v programovacím prostředí Microsoft Visual Studio 2005. Testován byl pod operačním systémem Windows XP ve spojení s nativním kompilátorem použitého IDE. Nevyužívá však žádné platformě závislé komponenty, tudíž by měl být plně přenositelný (to však nebylo testováno). Pro kompilaci je možné využít libovolný překladač, který podporuje zmiňovanou normu.

Kompilace vyžaduje knihovny MDSTk a OSG. Obě knihovny jsou obsaženy na přiloženém CD v adresáři *libraries* nebo dostupné na adresách:

- http://www.fit.vutbr.cz/~spanel/mdstk/
- http://www.openscenegraph.com

OSG je Open Source, MDSTk obdobně, je však nutné dodržet přiloženou licenci:

- *libraries/MDSTk-v0.5.3beta.zip/MDSTk/LICENSE.TXT*

Instrukce pro kompilaci knihoven jsou k nalezení v souborech:

- *libraries/OpenSceneGraph-2.6.1/OpenSceneGraph-2.6.1/README.txt*
- *libraries/MDSTk-v0.8.0beta.zip/MDSTk/README.txt*

Ke kompilaci knihovny OSG jsou potřeba další knihovny, které jsou obsaženy v souboru *OpenSceneGraph-dependencies.zip*. Při kompilaci v prostředí Visual Studio dochází k jistým chybovým výpisům ve spojení s OSG. Ty lze vyřešit zadáním parametru kompilátoru /GR. Složka se zdrojovými kódy obsahuje také projektové soubory pro Visual Studio, je třeba ale upravit cesty ke knihovnám.

A-2 Prerekvizity

Samotný program se neinstaluje, je ale nutné, aby byly v systému přítomny dynamické knihovny OSG. Ne nezbytným, každopádně vhodným požadavkem je výkonný počítač se solidní grafickou kartou.

A-3 Spuštění

Program se spouští z příkazové řádky s jedním povinným parametrem, který specifikuje zdrojový soubor ve formátu STL. Příklad spuštění se zdrojovým souborem *model.stl*:

```
> meshConverter.exe input:model.stl output:model.obj line-span:2
```

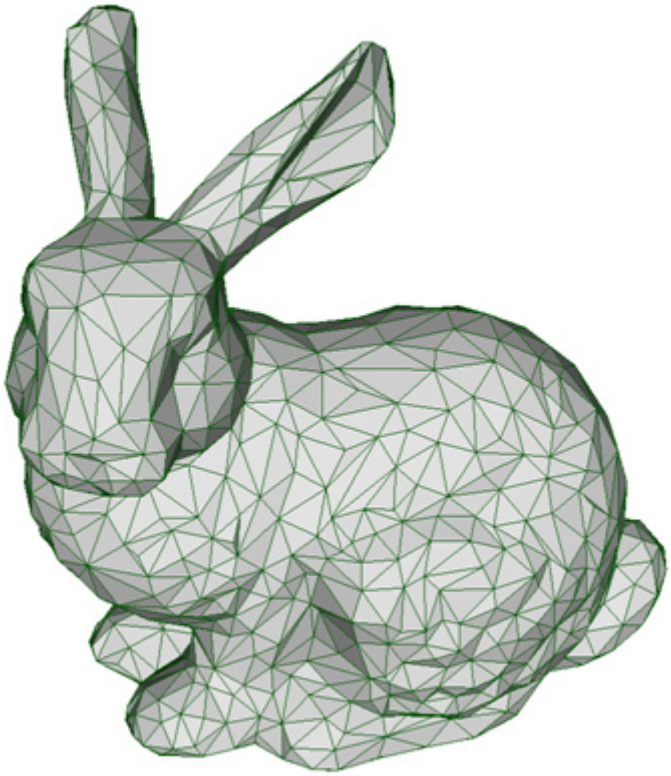
A-4 Ovládání

Pomocí myši lze s modelem otáčet (při držení levého tlačítka), měnit přiblížení (při držení pravého tlačítka) a posouvat pohled (při držení obou tlačítek). Další funkce ilustruje následující tabulka:

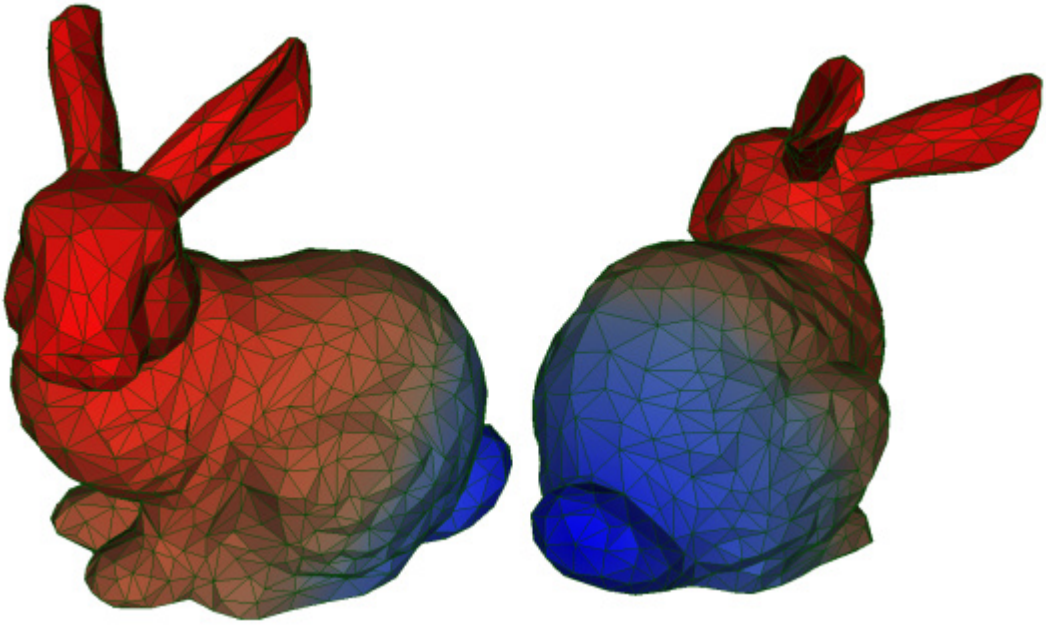
Funkce	***Způsob nastavení***
Název vstupního STL souboru	Povinný parametr `input:<path>`
Název výstupního OBJ souboru	Parametr `output:<path>`
Práh minimální křivosti při vytváření kostry	Parametr `curvature-treshold:<path>`
Počáteční vzdálenost linek	Parametr `line-span:<value>`
Výběr omezujících vrcholů	Maximum: stisk levého tlačítka myši Minimum: stisk pravého tlačítka myši Zrušení: stisk obou tlačítek zároveň
Výpočet křivostí	Klávesa 1
Vytvoření kostry	Klávesa 2
Výpočet skalárního pole	Klávesa 3
Výpočet vektorového pole	Klávesa 4
Vytvoření sítě plovoucích linek	Klávesa 5
Výpočet křížení linek	Klávesa 6
Vytvoření polygonů nové sítě	Klávesa 7
Teselace polygonů nové sítě	Klávesa 8
Uložení výstupu do souboru ve formátu OBJ	Klávesa 9
Přepnutí módu zobrazení polygonů	Klávesa W
Přepnutí stínovacího modelu	Klávesa M
Zobrazení vrcholů	Klávesa V
Zobrazení hran	Klávesa E
Zobrazení kostry	Klávesa S
Výběr zobrazení skalárního pole / křivostí	Klávesa F
Způsob zobrazení skalárního pole / křivostí	Klávesa P
Zobrazení gradientních vektorů	Klávesa G
Zobrazení izoparametrických vektorů	Klávesa I
Zobrazení normálových vektorů	Klávesa N
Zobrazení vektorů	Klávesa R
Zobrazení plovoucích linek	Klávesa L
Zobrazení křížení plovoucích linek	Klávesa C
Zobrazení původního modelu	Klávesa [
Zobrazení výsledného modelu	Klávesa]

Příloha B Ukázkové modely

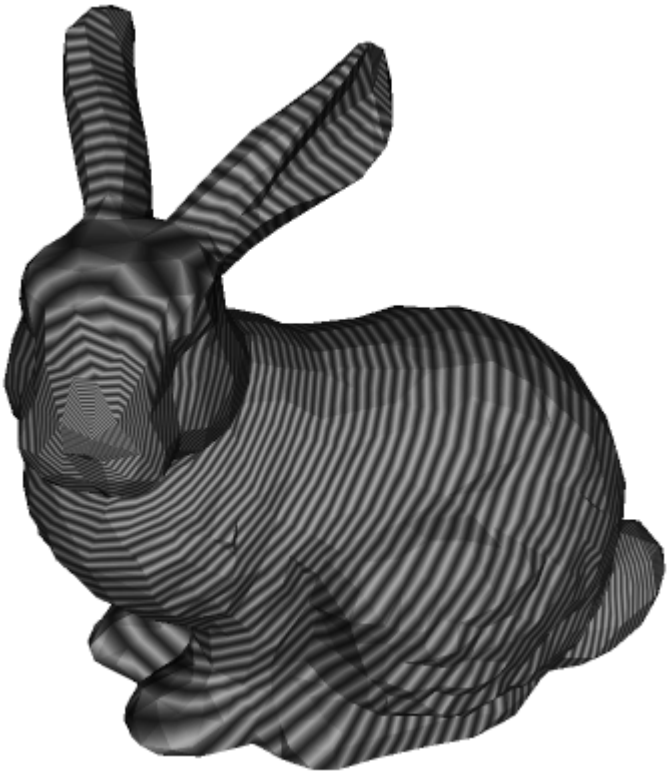
B-1 Bunny



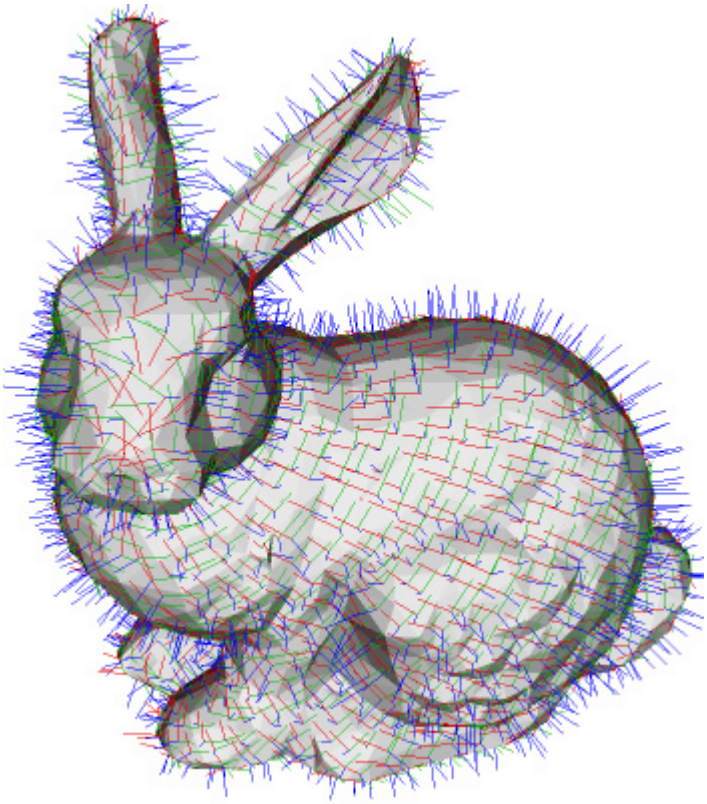
Obrázek B-1-1: *Vstupní trojúhelníková síť reprezentující model zajíce.*



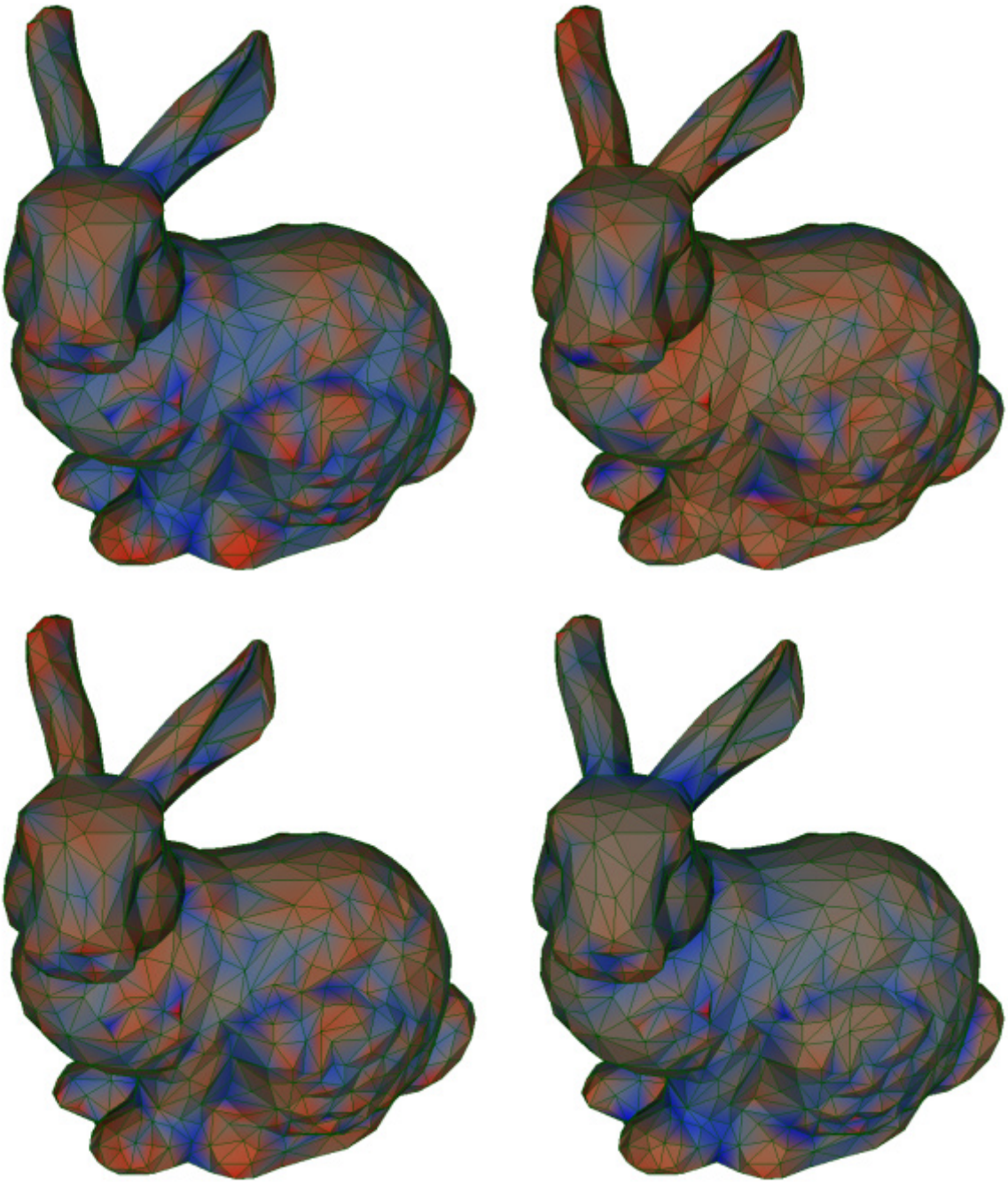
Obrázek B-1-2: *Skalární pole (modrá reprezentuje záporné hodnoty, červená kladné).*



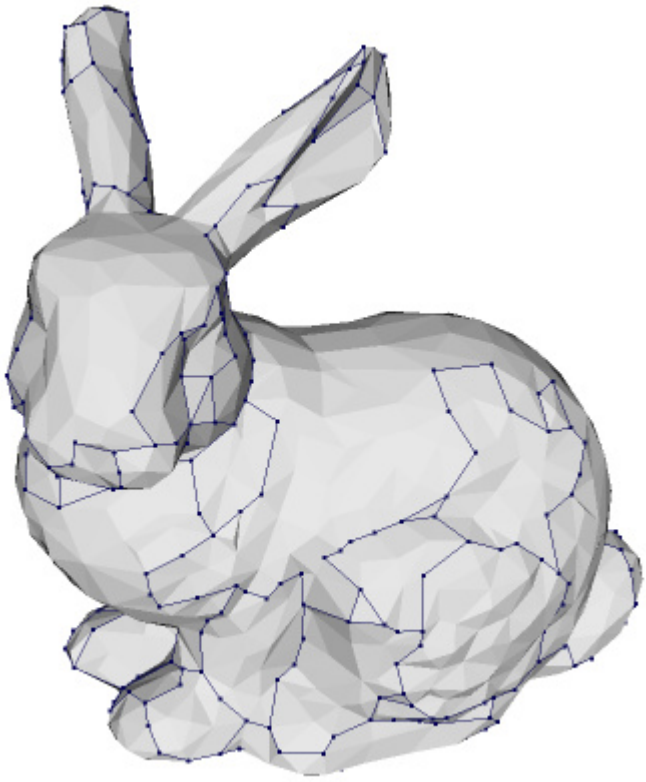
Obrázek B-1-3: *Izolinie skalárního pole.*



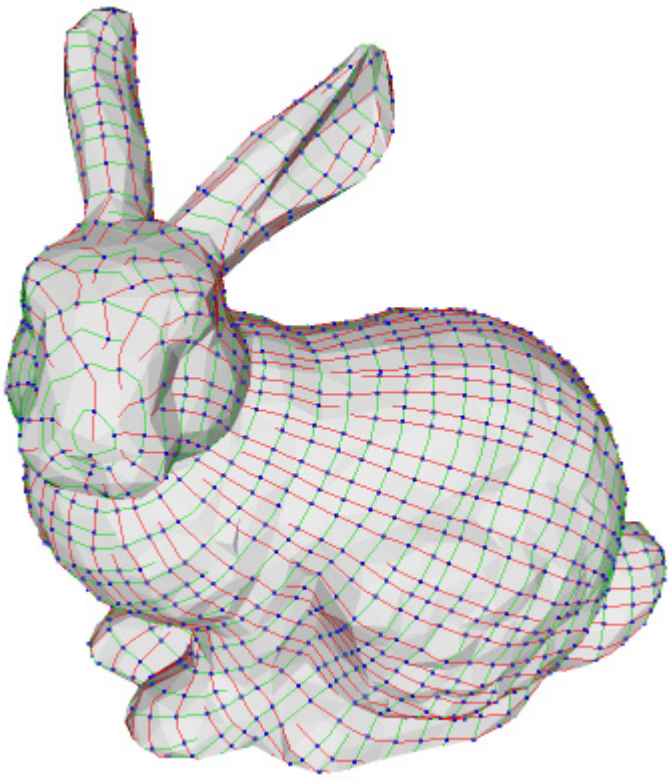
Obrázek B-1-4: *Vektorová pole (gradientní vektory jsou červené, izoparametrické zelené, normálové modré).*



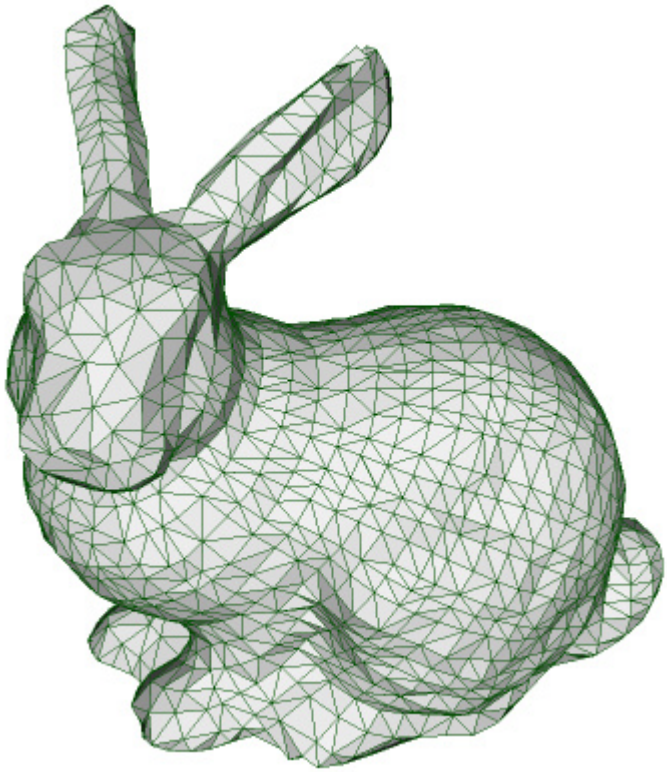
Obrázek B-1-5: *Průběh křivostí: minimální, maximální, střední a Gaussova (modrá reprezentuje záporné hodnoty, červená kladné).*



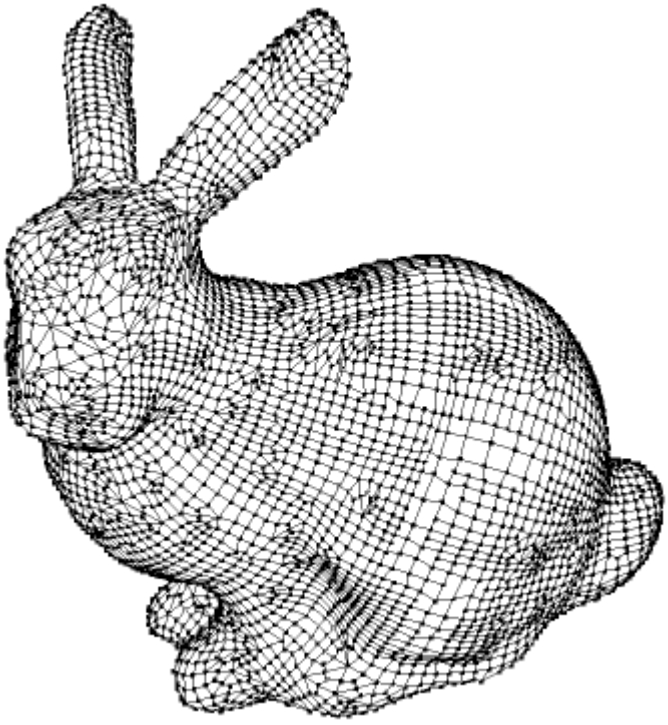
Obrázek B-1-6: *Hrany a vrcholy kostry.*



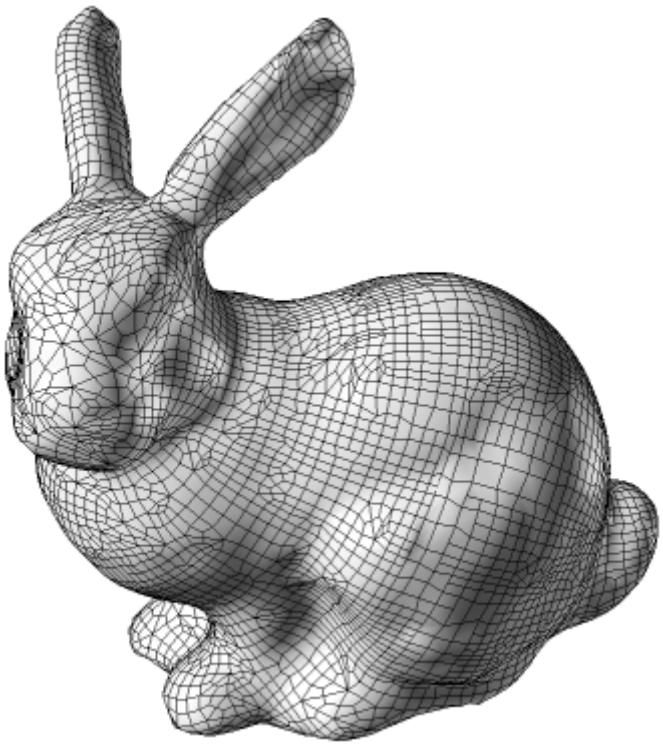
Obrázek B-1-7: *Síť plovoucích linek a jejich průsečíky*
(gradientní linky jsou červené, izoparametrické zelené, průsečíky modré).



Obrázek B-1-8: *Výsledná quadrilaterální síť.*

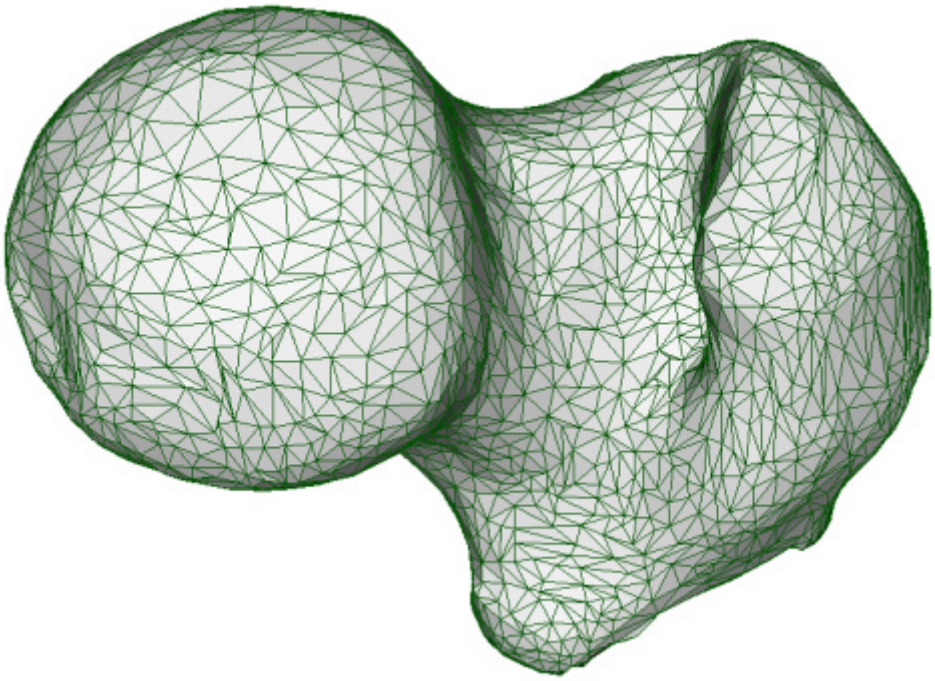


Obrázek B-1-9: *Síť řídících bodů T-Spline plochy.*

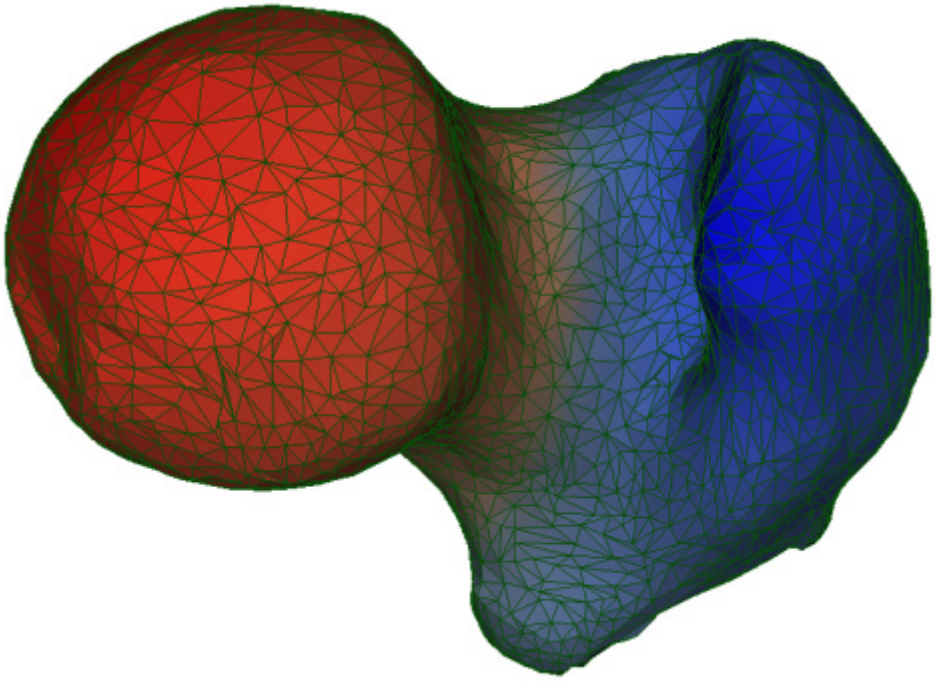


Obrázek B-1-10: *T-Spline plocha generovaná v aplikaci Rhinoceros.*

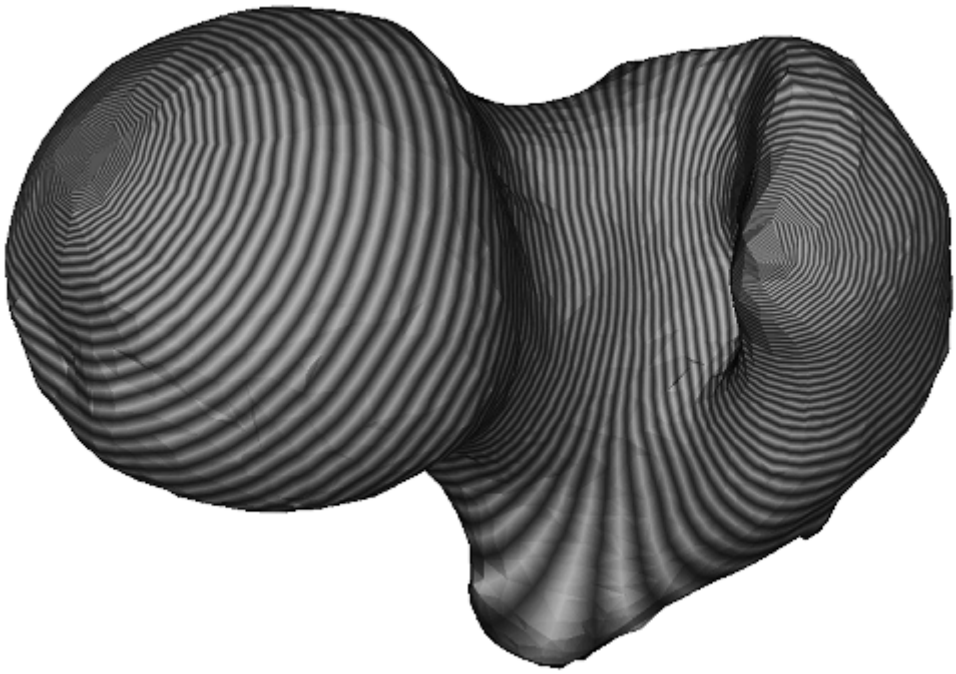
B-2 Femur



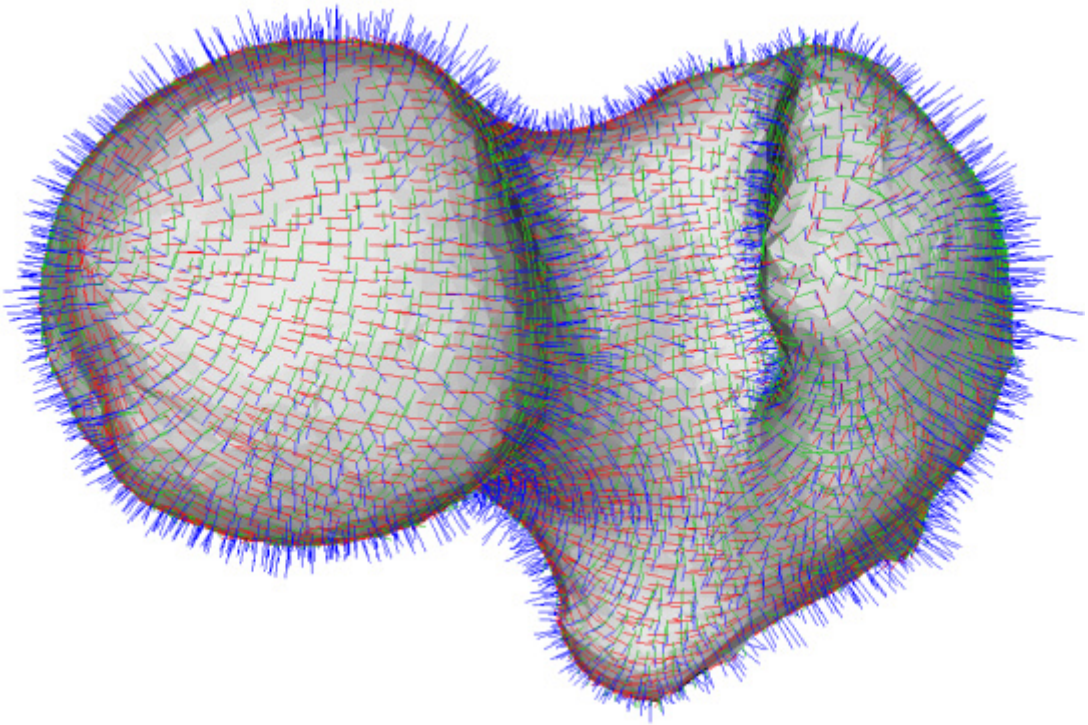
Obrázek B-2-1: *Vstupní trojúhelníková síť reprezentující model kyčelního kloubu.*



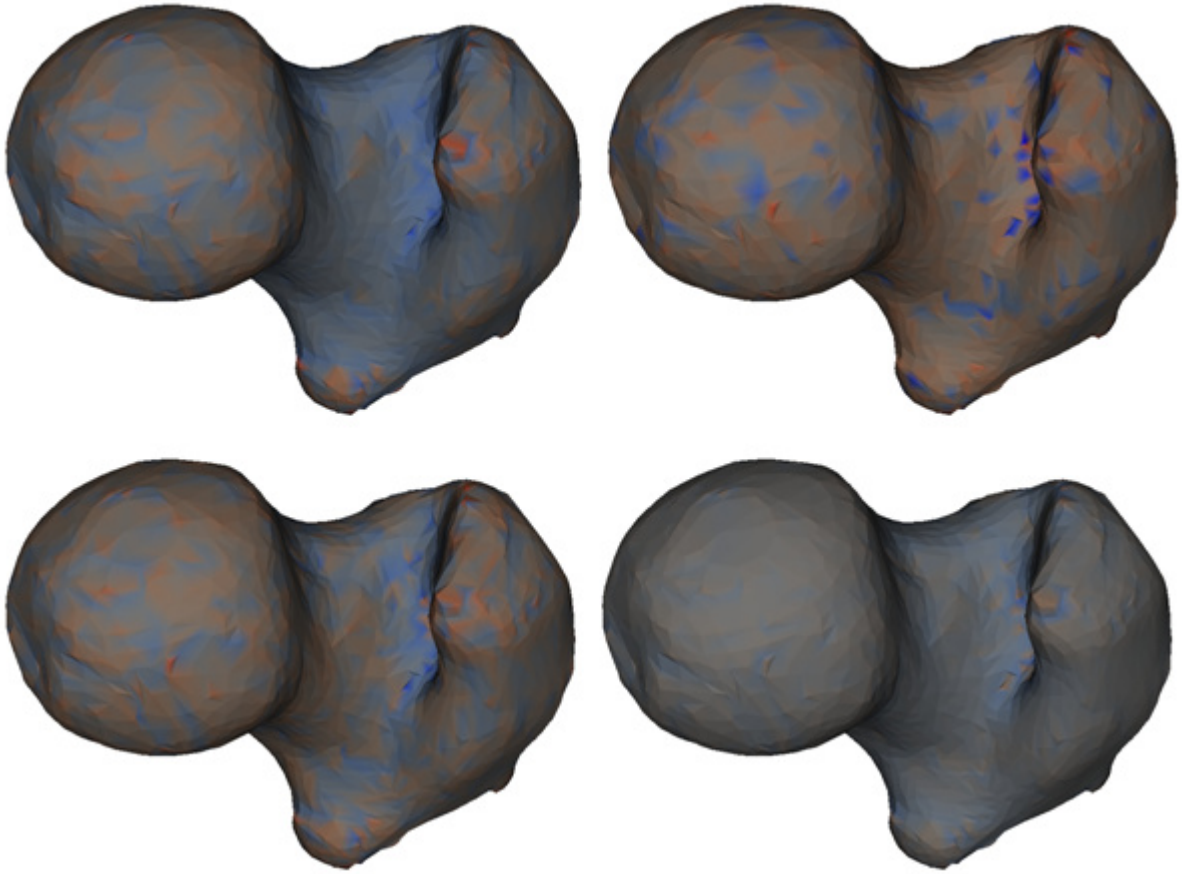
Obrázek B-2-2: *Skalární pole (modrá reprezentuje záporné hodnoty, červená kladné).*



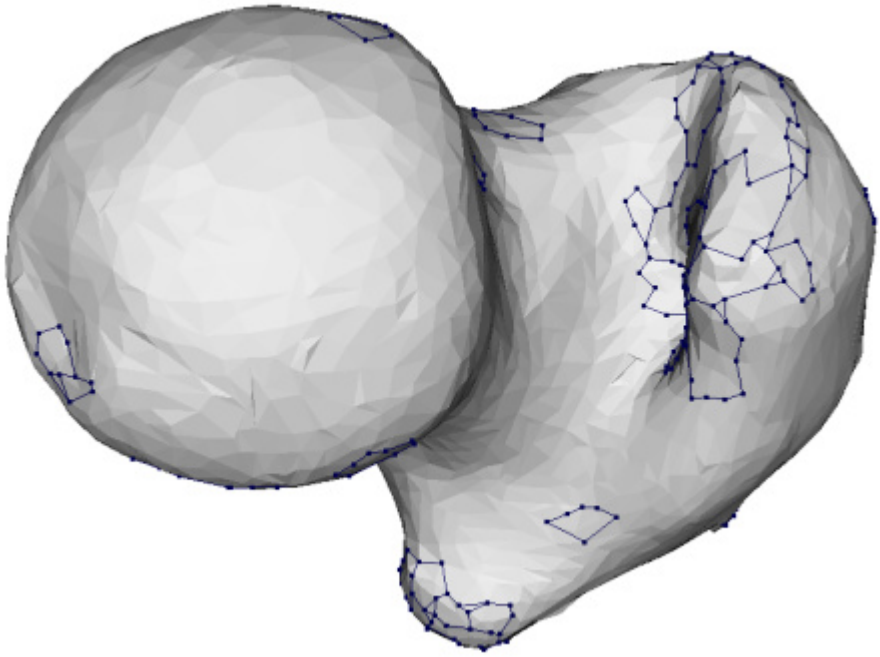
Obrázek B-2-3: *Izolinie skalárního pole.*



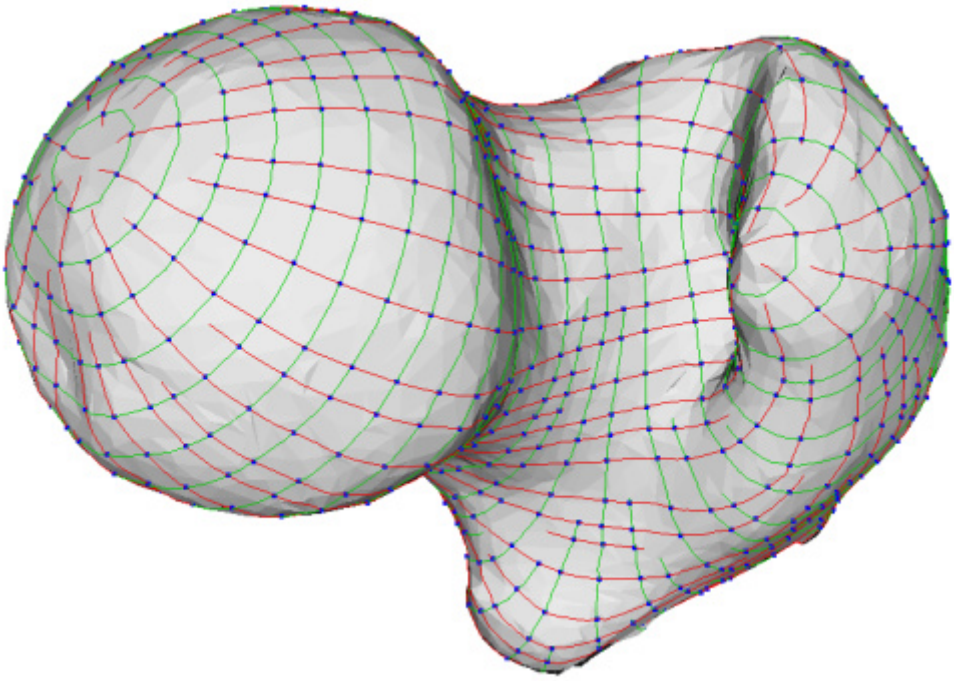
Obrázek B-2-4: *Vektorová pole (gradientní vektory jsou červené, izoparametrické zelené, normálové modré).*



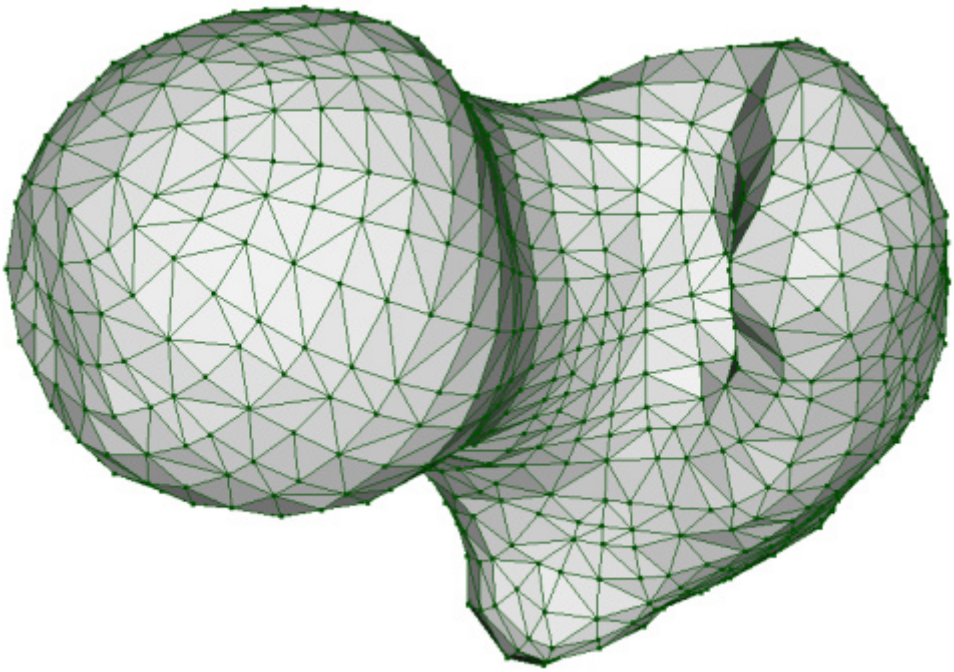
Obrázek B-2-5: *Průběh křivostí: minimální, maximální, střední a Gaussova (modrá reprezentuje záporné hodnoty, červená kladné).*



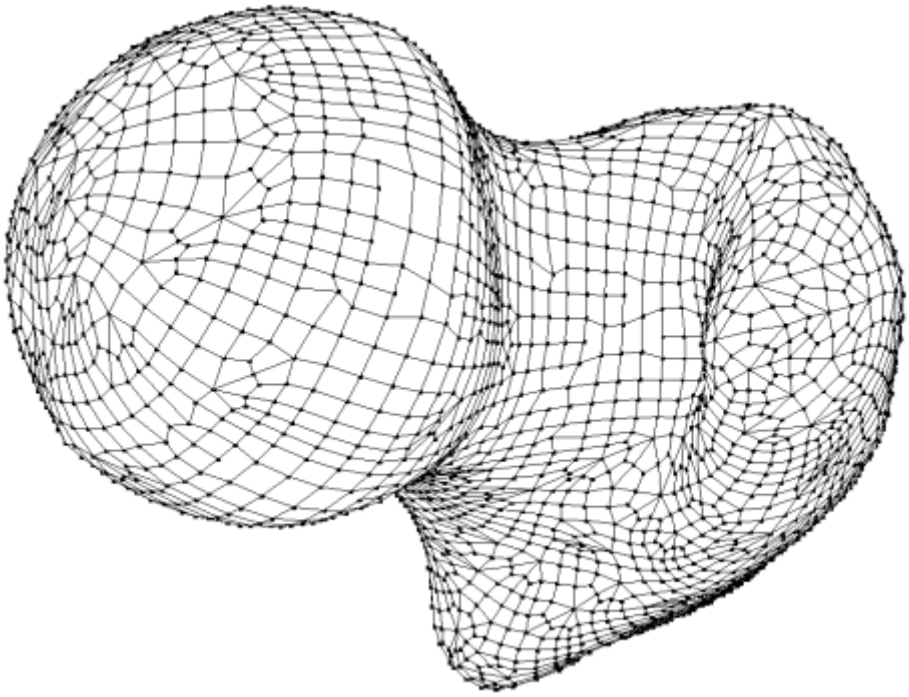
Obrázek B-2-6: *Hrany a vrcholy kostry.*



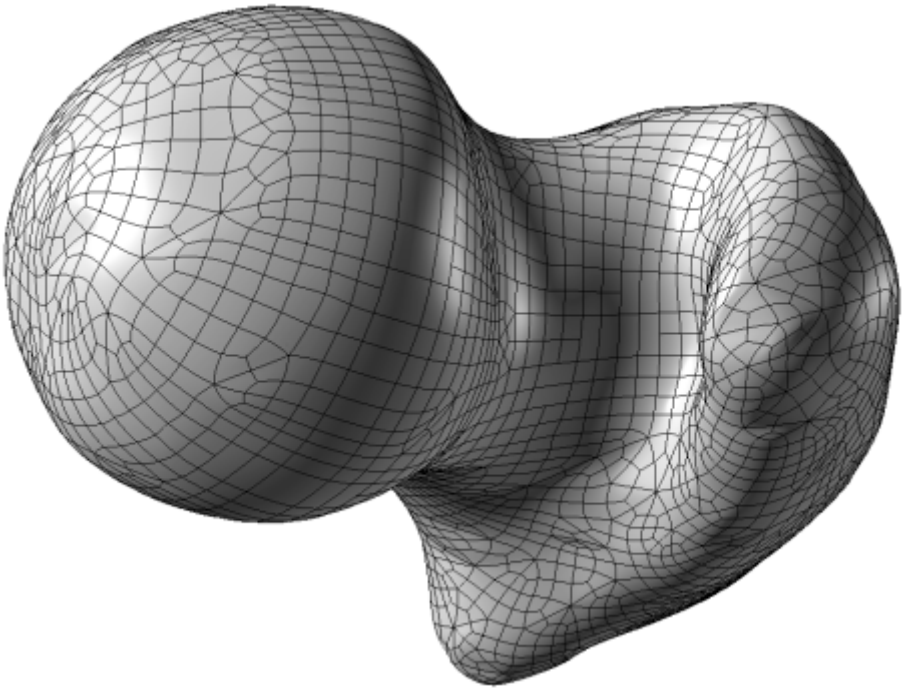
Obrázek B-2-7: *Síť plovoucích linek a jejich průsečíky (gradientní linky jsou červené, izoparametrické zelené).*



Obrázek B-2-8: *Výsledná quadrilaterální síť.*

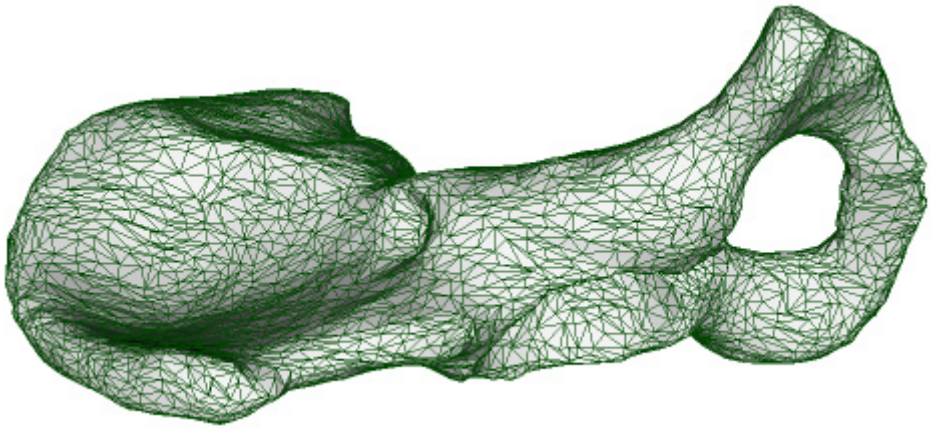


Obrázek B-2-9: *Síť řídících bodů T-Spline plochy.*

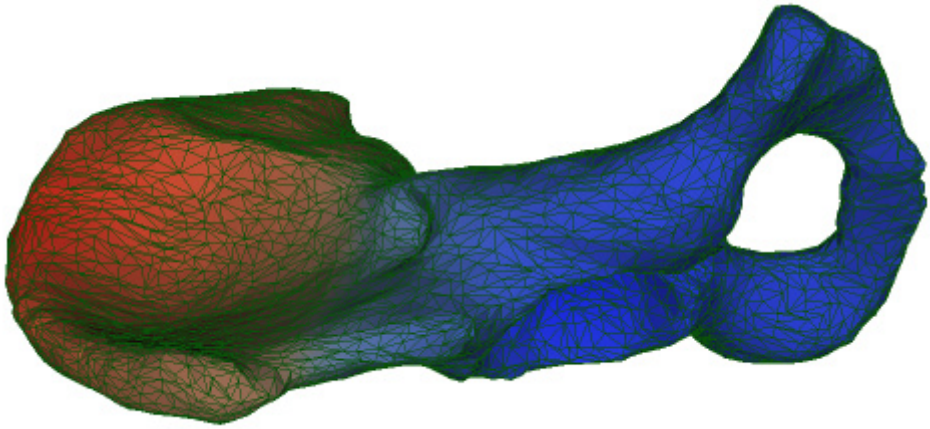


Obrázek B-2-10: *T-Spline plocha generovaná v aplikaci Rhinoceros.*

B-3 Ilium

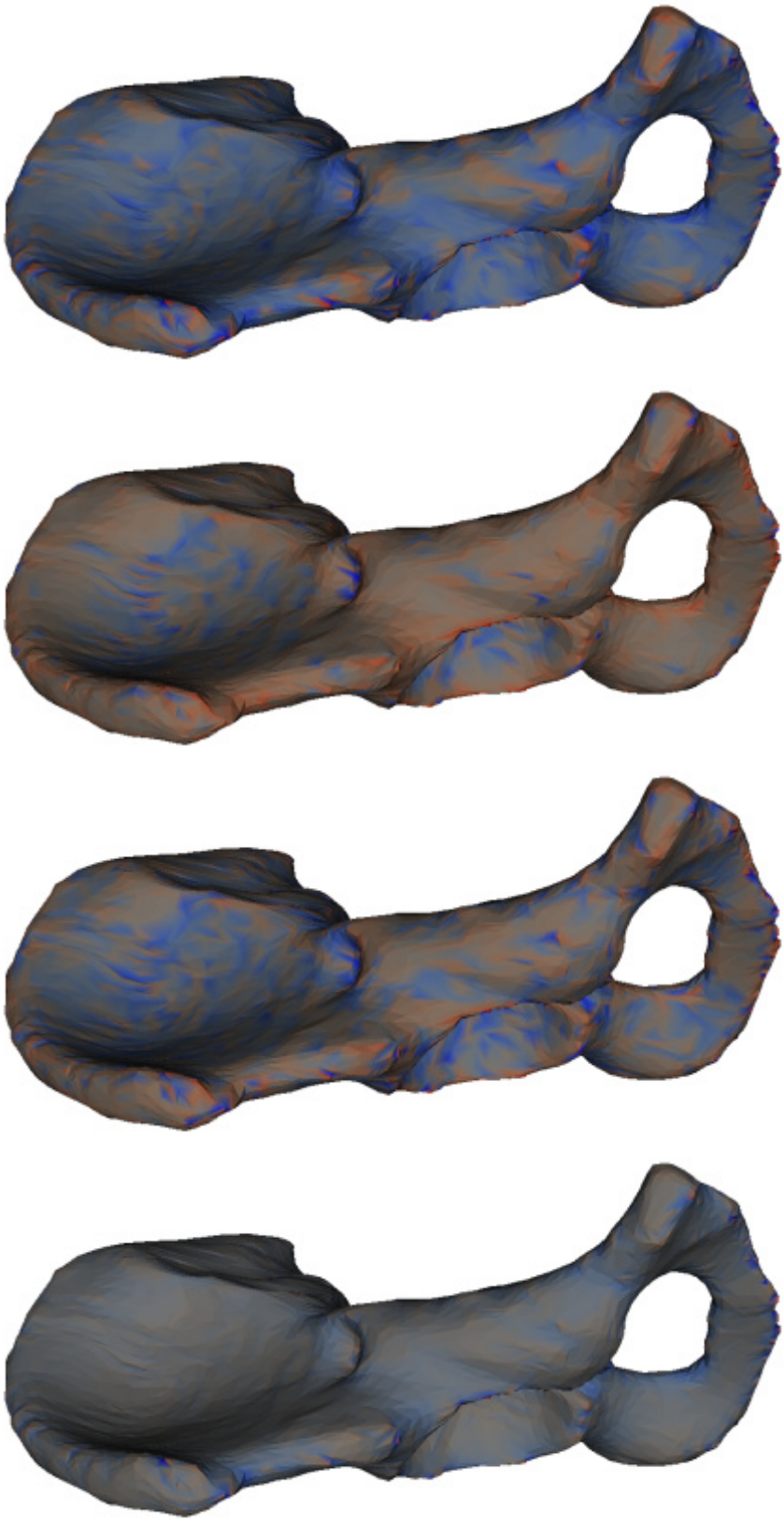


Obrázek B-3-1: *Vstupní trojúhelníková síť reprezentující model lopatky kyčelního kloubu.*

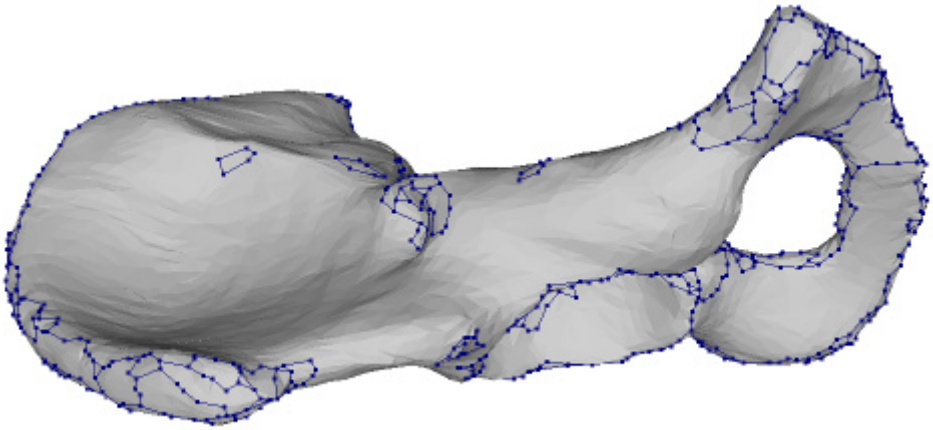


Obrázek B-3-2: *Skalární pole (modrá reprezentuje záporné hodnoty, červená kladné).*

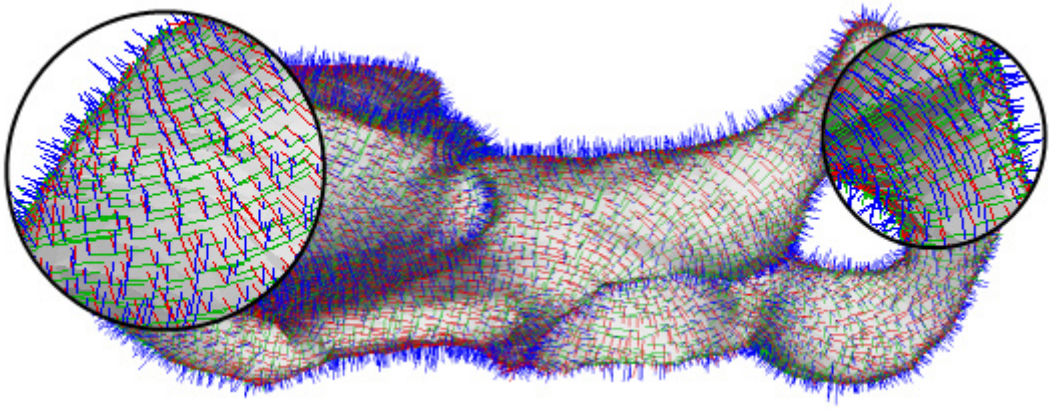
Obrázek B-3-3: *Izolinie skalárního pole.*



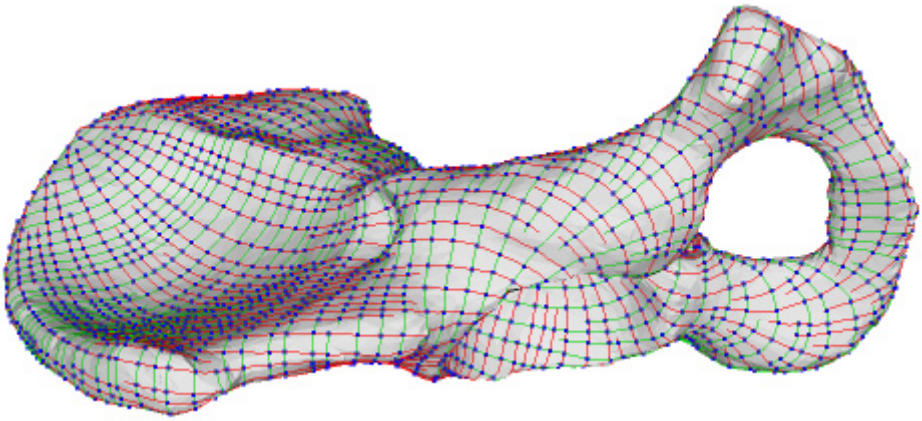
Obrázek B-3-4: *Průběh křivostí: minimální, maximální, střední a Gaussova (modrá reprezentuje záporné hodnoty, červená kladné).*



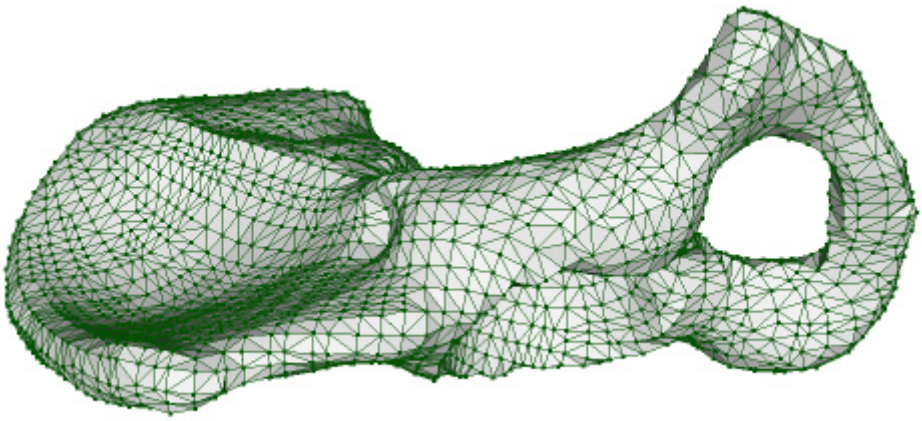
Obrázek B-3-5: *Hrany a vrcholy kostry.*



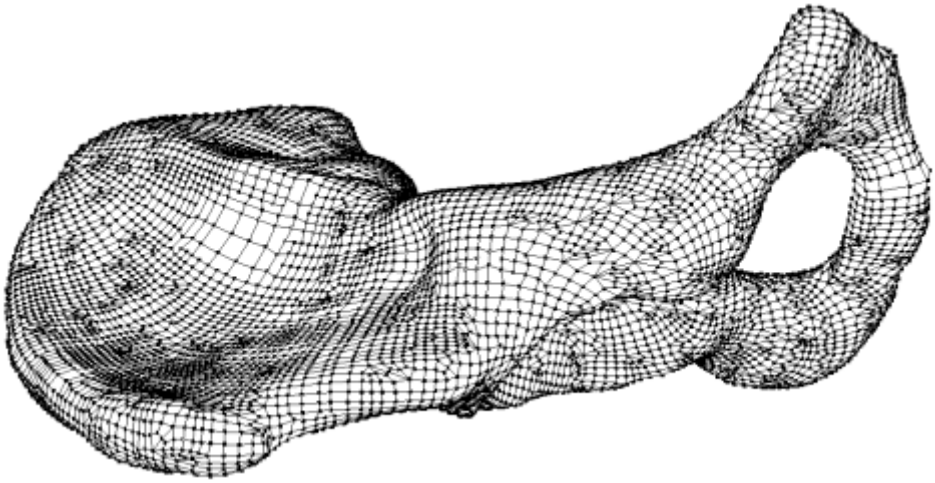
Obrázek B-3-6: *Vektorová pole (gradientní vektory jsou červené, izoparametrické zelené, normálové modré).*



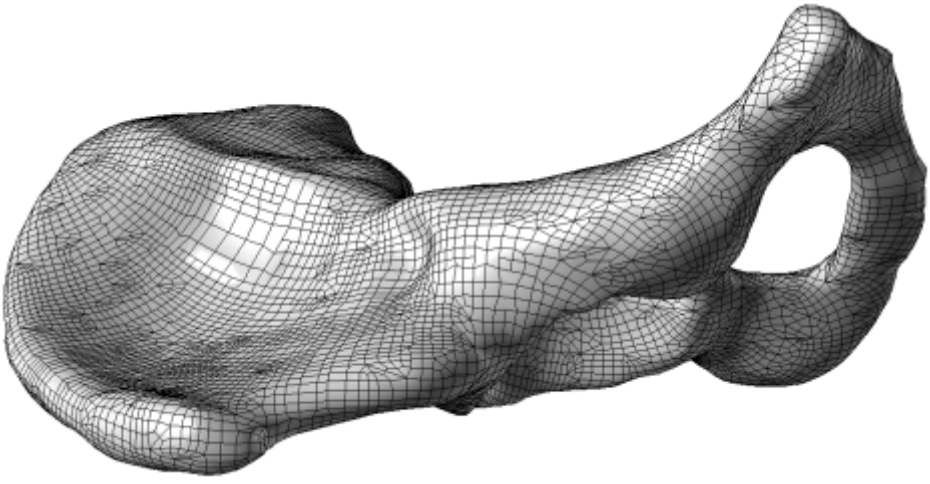
Obrázek B-3-7: *Síť plovoucích linek a jejich průsečíky*
(gradientní linky jsou červené, izoparametrické zelené, průsečíky modré).



Obrázek B-3-8: *Výsledná quadrilaterální síť.*



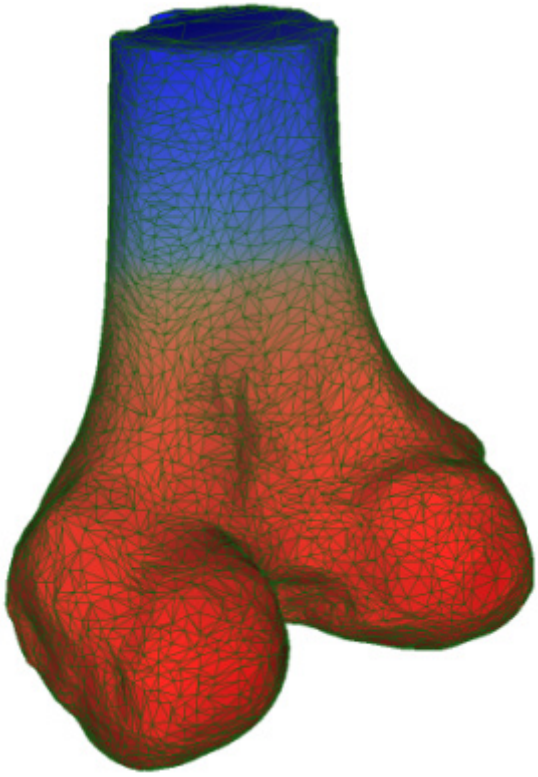
Obrázek B-3-9: *Síť řídících bodů T-Spline plochy.*



Obrázek B-3-10: *T-Spline plocha generovaná v aplikaci Rhinoceros.*

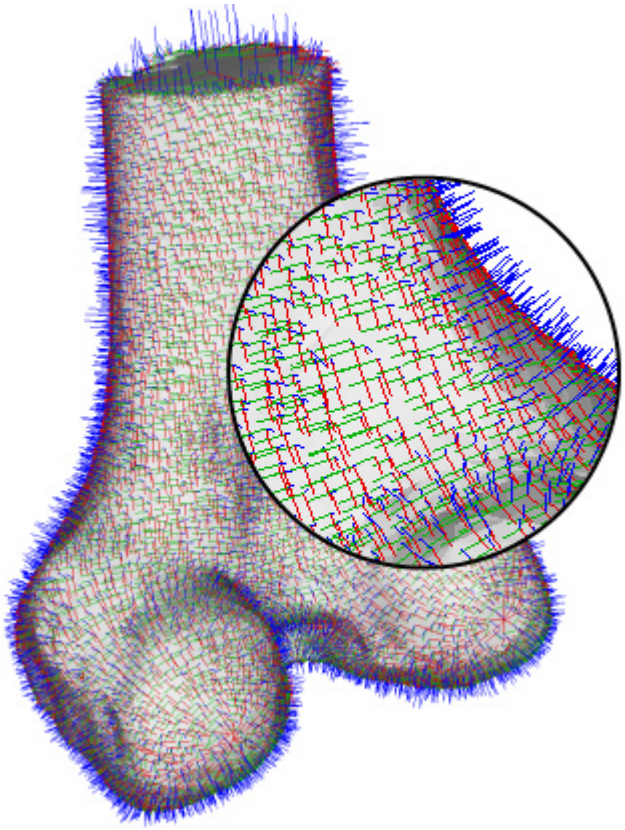
B-4 Koleno

Obrázek B-4-1: *Vstupní trojúhelníková síť reprezentující model části kolenního kloubu.*

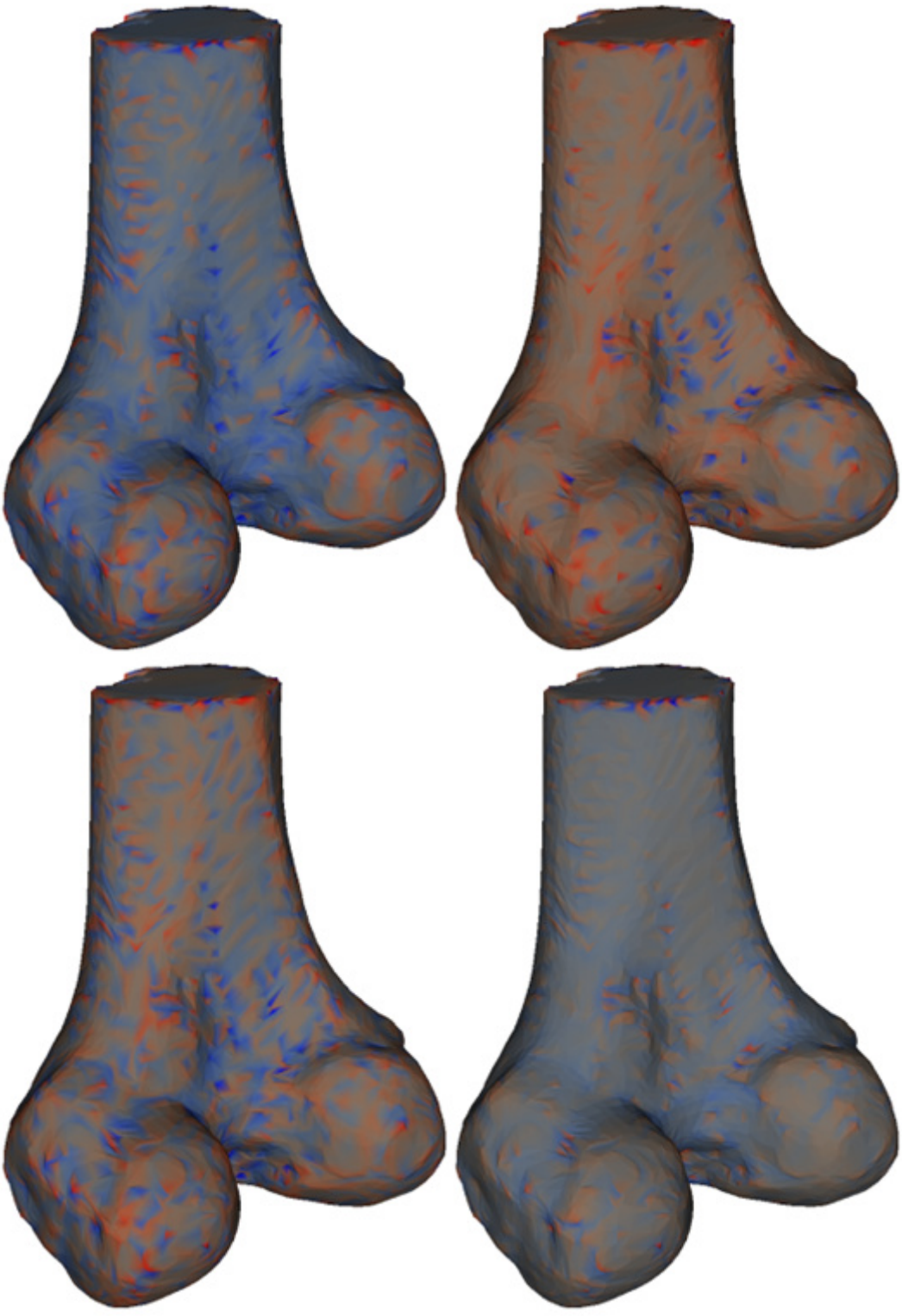


Obrázek B-4-2: *Skalární pole (modrá reprezentuje záporné hodnoty, červená kladné).*

Obrázek B-4-3: *Izolinie skalárního pole.*

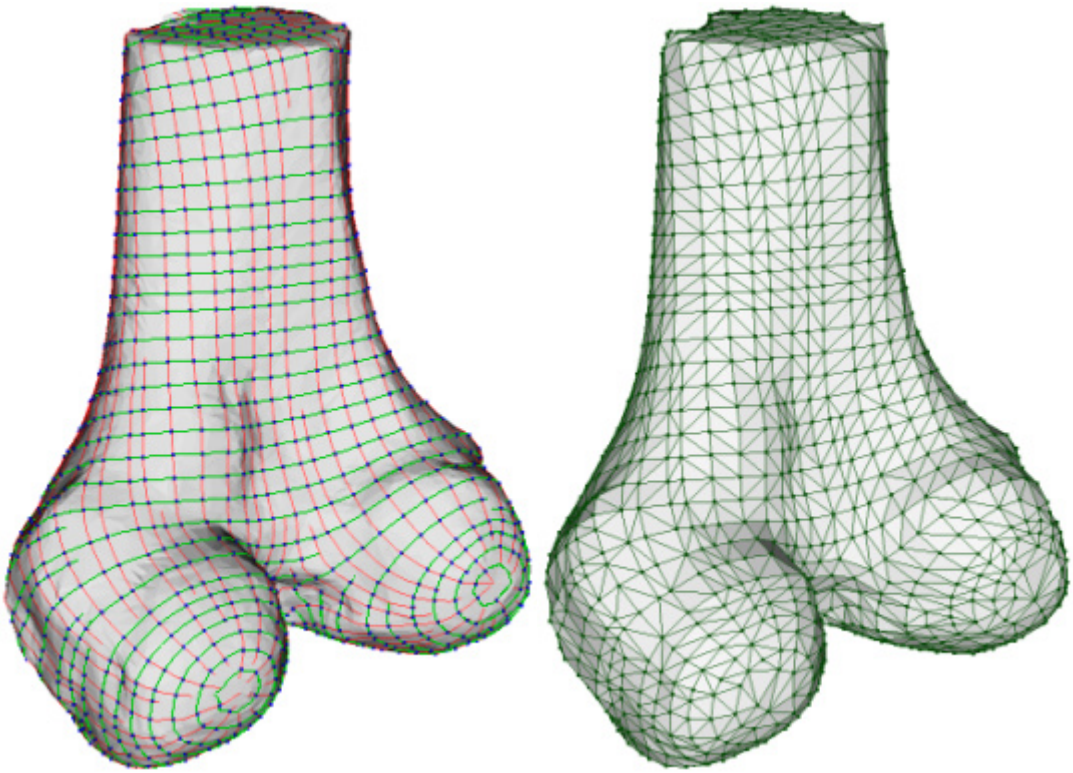


Obrázek B-4-4: *Vektorová pole (gradientní vektory jsou červené, izoparametrické zelené, normálové modré).*

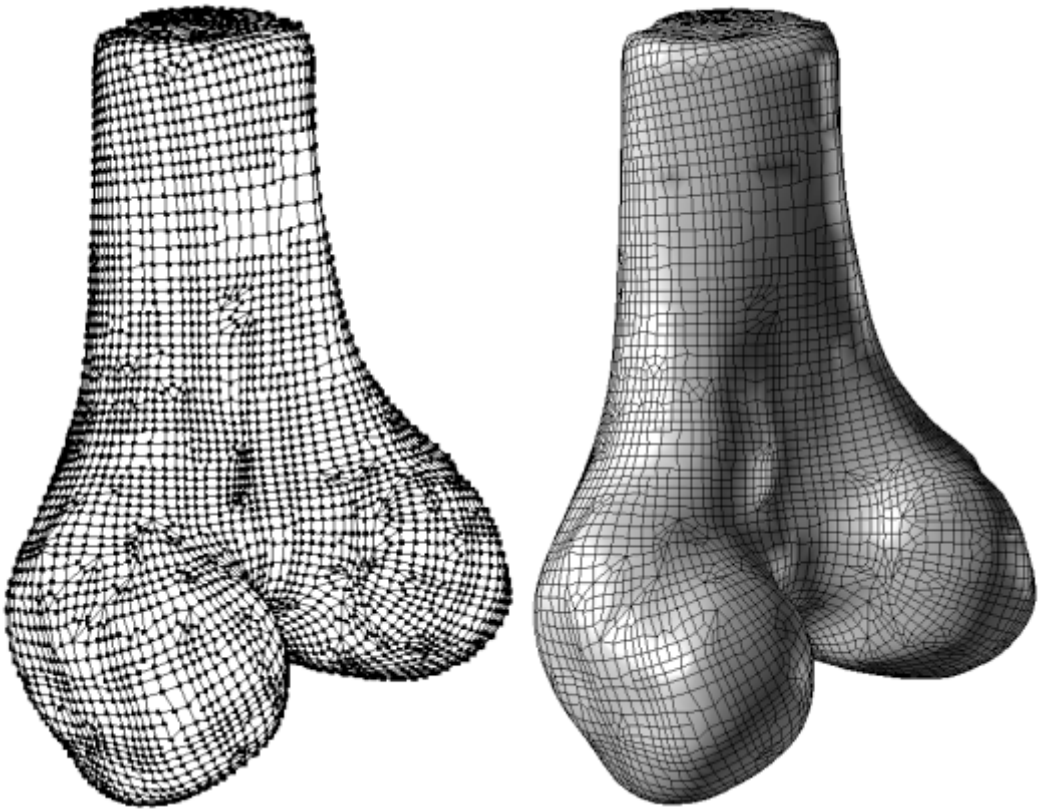


Obrázek B-4-5: *Průběh křivostí: minimální, maximální, střední a Gaussova (modrá reprezentuje záporné hodnoty, červená kladné).*

Obrázek B-4-6: *Vrcholy a hrany kostry.*

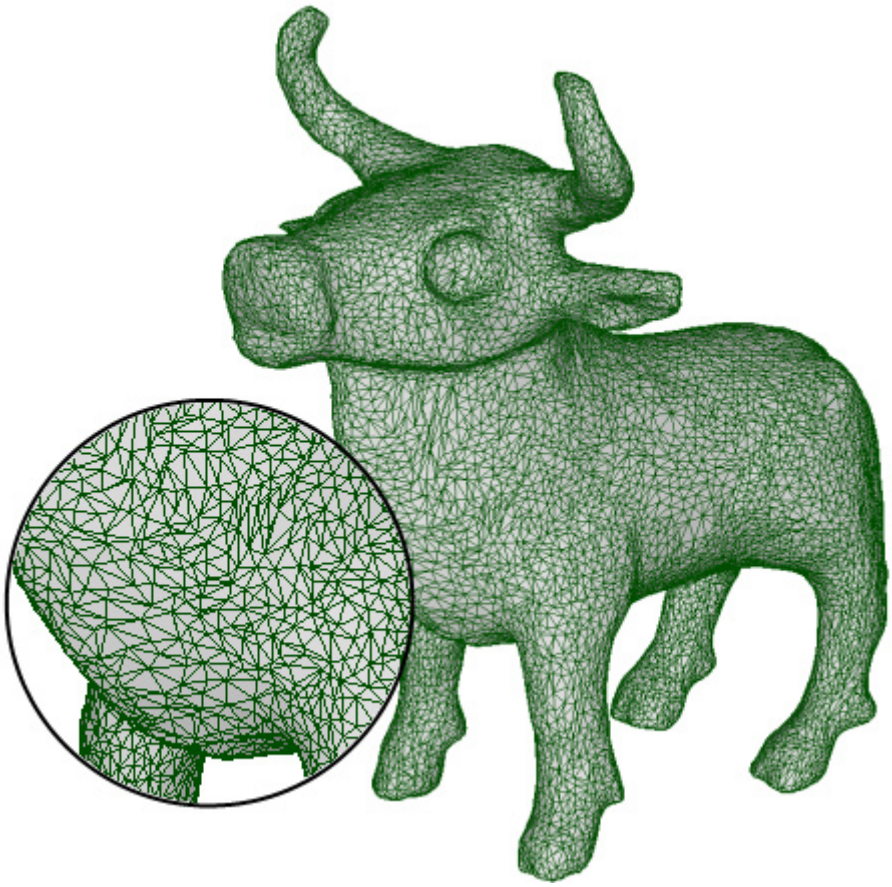


Obrázek B-4-7 / B-4-8: *Vlevo síť plovoucích linek a jejich průsečíky (gradientní linky jsou červené, izoparametrické zelené, průsečíky modré), vpravo výsledná quadrilaterální síť.*

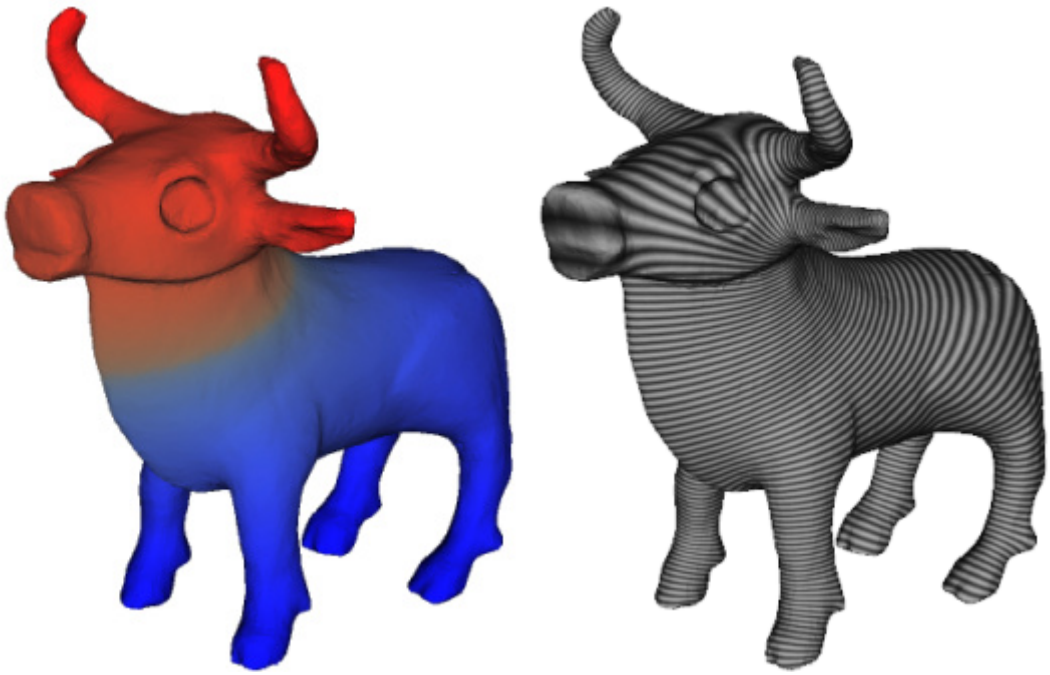


Obrázek B-4-9 / B-4-10: *Vlevo síť řídících bodů, vpravo T-Spline plocha generovaná v aplikaci Rhinoceros.*

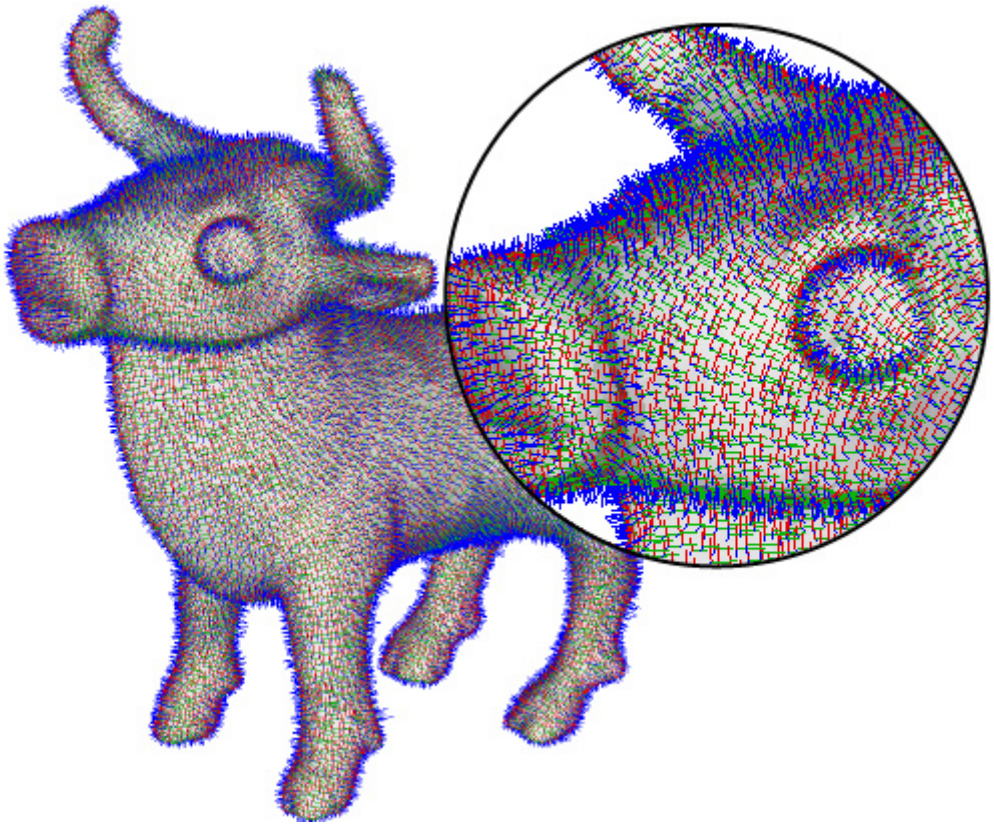
B-5 Býček



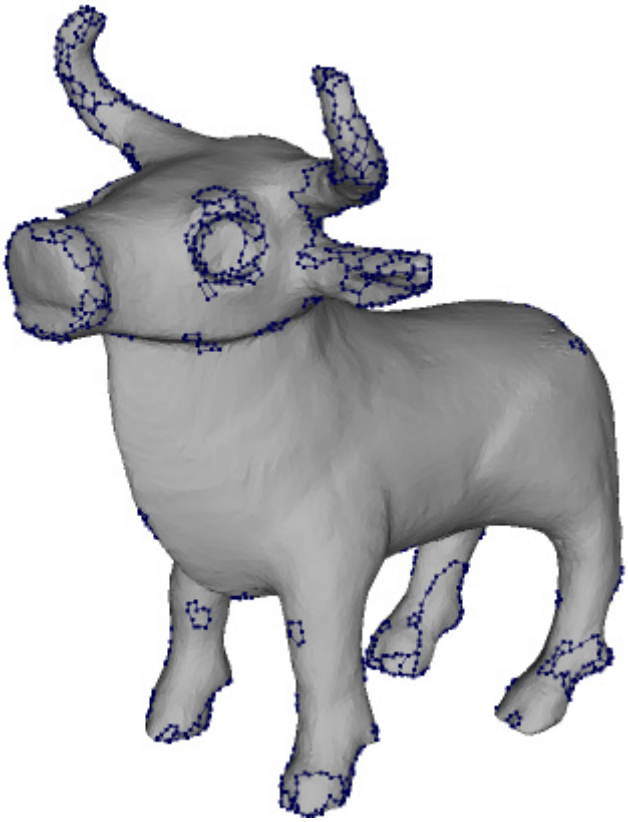
Obrázek B-5-1: *Vstupní trojúhelníková síť reprezentující model býka.*



Obrázek B-5-2 / B-5-3: *Skalární pole (modrá reprezentuje záporné hodnoty, červená kladné) a jeho izolinie.*

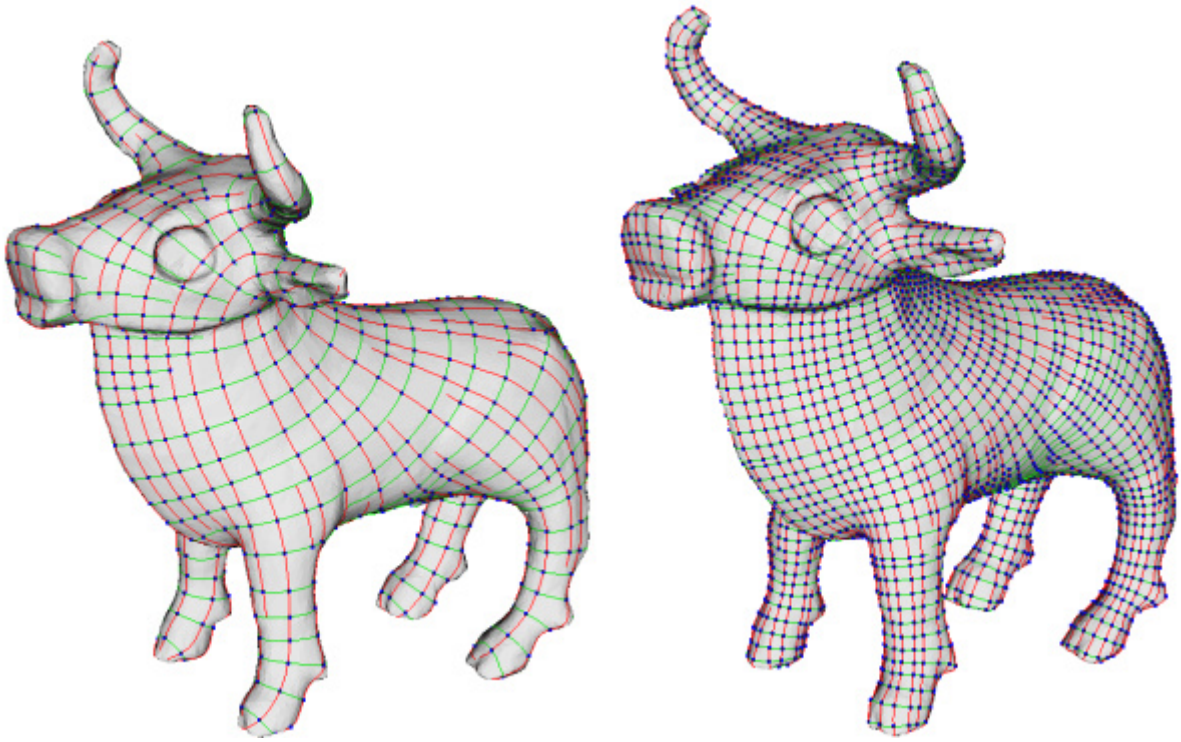


Obrázek B-5-4: *Vektorová pole (gradientní vektory jsou červené, izoparametrické zelené, normálové modré).*

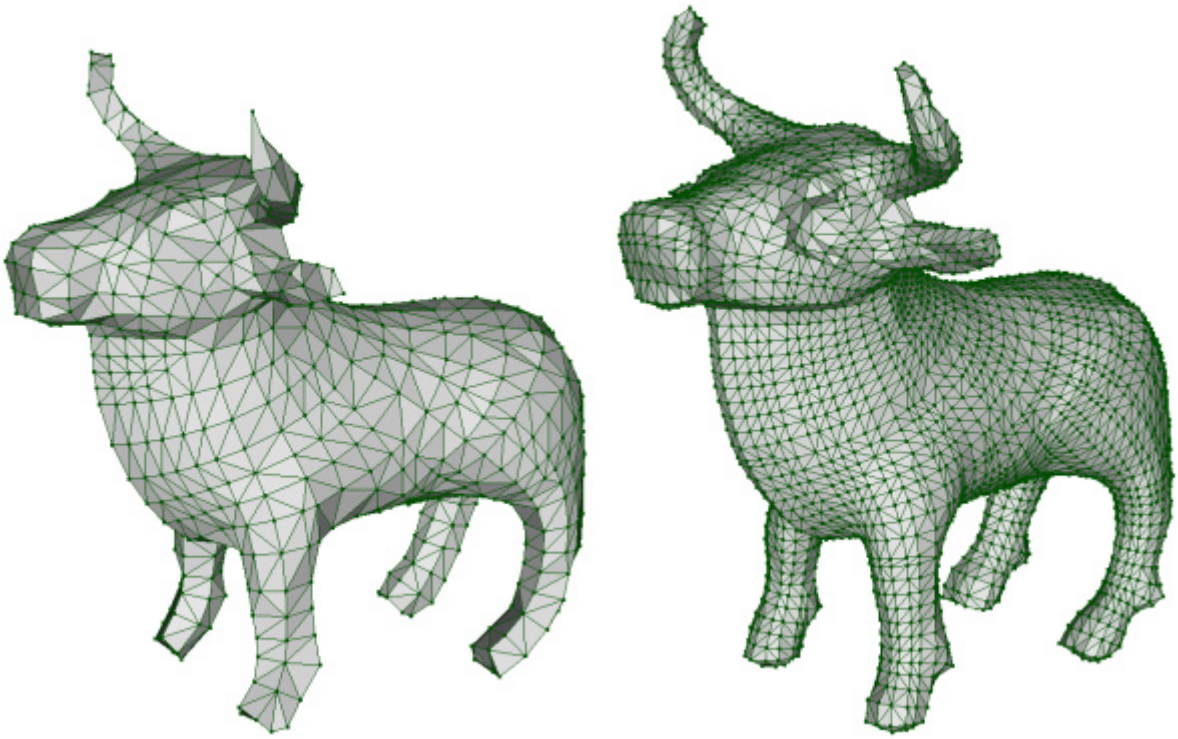


Obrázek B-5-5: *Hrany a vrcholy kostry.*

Obrázek B-5-6: *Průběh křivostí: minimální, maximální, střední a Gaussova (modrá reprezentuje záporné hodnoty, červená kladné).*



Obrázek B-5-7: *Síť plovoucích linek, vlevo menší hustota, vpravo větší (gradientní vektory a linky jsou červené, izoparametrické zelené, normálové linky a průsečíky modré).*



Obrázek B-5-8: *Výsledná quadrilaterální síť, vlevo menší detail, vpravo větší.*