

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

Diplomová práce

Dekompozice modelu svalu na svalová vlákna

Plzeň, 2013

David Cholt

Prohlášení

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

V Plzni dne 12. května 2013

.....
David Cholt

Poděkování

Tímto bych chtěl poděkovat panu Ing. Josefu Kohoutovi, Ph.D. za podklady, rady a usměrnění mých myšlenek při vzniku této práce po celý akademický rok 2012/2013. Dále bych chtěl poděkovat rodině a přátelům za jejich morální podporu v průběhu celého mého studia, bez níž bych neměl možnost se k této práci dostat.

Práce byla částečně podpořena z projektu VPHOP (FP7 ICT-223865) programem Information Society Technologies Programme of the European Commission.

Abstract

Decomposition of Muscle Model into Muscle Fibres

Decomposition of a virtual muscle model into virtual muscle fibers is a process of filling a surface muscle model with a set of polylines that resemble individual muscle fibers that would exist in a real muscle. The purpose of this thesis is to design, implement and test a new method of muscle decomposition that uses additional data to guide the decomposition process and leads to a more realistic fibre appearance, especially when used to decompose some of the more complex multiheaded muscles. The theoretical part of this thesis describes the method itself, addresses the complex muscle decomposition and deals with problems that arose during the design process. The implementation part of this thesis describes interpolation methods and libraries used, lists various implementation details and concludes in presenting the capabilities of the implemented method and visual results of the decomposition. Time consumption dependency of the method on various input parameters is also evaluated.

Obsah

1	Úvod	1
2	Z anatomie svalů	2
2.1	Dělení dle průběhu vláken	2
2.1.1	Svaly s rovnoběžnými vlákny	2
2.1.2	Pennate svaly	3
2.1.3	Kruhové svaly	4
2.2	Dělení dle počtu oblastí úponu	4
3	Současné metody a projekt VPHOP	5
3.1	Reprezentace povrchu svalu a povrchových vláken	5
3.2	Reprezentace vnitřní struktury svalů	5
3.3	VPHOP a metody dekompozice povrchových modelů svalů	6
3.3.1	Advanced Slicing Method (ASM)	6
3.3.2	Kukačková metoda	7
3.3.3	Morphed Mass Spring System	7
3.3.4	Omezení metod	7
4	Přehled navrhované metody	8
5	Podrobný popis navrhované metody	11
5.1	Vstupní data	11
5.1.1	Požadavky na vstupní data	11
5.1.2	Dostupná data	12
5.1.3	Získávání dat	13
5.2	Předpříprava dat	14
5.3	Konstrukce vláken na povrchu svalu	16
5.4	Výpočet normál	17
5.5	Vzorkování objemu svalu	20
5.5.1	Štěpení svalu	20
5.5.2	Test korektnosti směru normál	22
5.6	Dekompozice objemu svalu	23
5.6.1	Generování vrstev v objemu svalu	23
5.6.2	Postranní vlákna	25
5.7	Upnutí vláken k oblastem úponu	25
5.8	Vyhlazení vláken a dokončení	28

6 Implementace	29
6.1 Interpolační metody	29
6.1.1 Kubická spline interpolace	29
6.1.2 Omezený kubický spline	30
6.1.3 Catmull-Rom spline	32
6.2 Použité knihovny a frameworky	34
6.2.1 VTK a jeho pipeline	34
6.2.2 MAF2 a MAF3	35
6.2.3 libhungarian	37
6.3 Implementace dekompoziční metody	37
6.3.1 vtkMuscleDecomposer	37
6.3.2 AddFiberData()	38
6.3.3 AddAttachmentData()	39
6.3.4 Execute()	39
6.4 Demonstrační aplikace	41
6.5 Získání umělých dat	41
7 Výsledky	43
7.1 Dekomponovatelné typy svalů	43
7.2 Ukázky výsledků dekompozice	44
7.3 Rychlost dekompozice	48
7.3.1 Závislost na vstupních datech	48
7.3.2 Závislost na vstupních parametrech	51
8 Závěr	53
Přílohy	57

Seznam obrázků

2.1	Typy svalů dle průběhu vláken	3
2.2	Lidské pennate svaly	4
5.1	Ukázka dat vstupních řídicích vláken	13
5.2	Ilustrace indexace vláken	16
5.3	Příklad konstrukce vlákna na povrchu svalu	17
5.4	Výpočet normály na povrchu svalu	18
5.5	Ilustrace indexace vláken	19
5.6	Křížení vláken při přímém použití normál	19
5.7	Ukázka normál a vzorkování objemu svalu	19
5.8	Vzorkování objemu svalu	20
5.9	Ukázka štěpení svalu	21
5.10	Metoda štěpení svalu	22
5.11	Ukázka efektu "hrábě"	23
5.12	Generování postranních vrstev vláken	25
5.13	Ilustrace metriky	26
5.14	Problém pennate svalů	27
5.15	Ukázka vláken po jejich upnutí	28
6.1	Protínání vláken způsobené oscilací kubické spline křivky	30
6.2	Interpolace pomocí omezeného splinu	32
6.3	Příklad rekursivního pyramidového výpočtu Catmull-Rom splinu	33
6.4	Příklad Catmull-Rom interpolace	34
6.5	Logo VTK	34
6.6	Pipeline systému VTK	35
6.7	Nástroj přichytnutí k povrchu	42
7.1	Výsledky: Modelový bipennate sval	43
7.2	Výsledky: Modelový čtyřhlavý sval	44
7.3	Výsledky: Dekompozice poloviny kruhového svalu	45
7.4	Dekompozice svalu <i>Iliacus</i>	45
7.5	Dekompozice svalu <i>Adductor Magnus</i>	46
7.6	Dekompozice svalu <i>Rectus Femoris</i>	46
7.7	Dekompozice svalu <i>Biceps Femoris</i>	47
7.8	Dekompozice svalu <i>Gluteus Maximus</i>	48
7.9	Data pro testování rychlosti dekompozice	48
7.10	Závislost doby dekompozice na velikosti dat	49
7.11	Závislost doby dekompozice na počtu řídicích vláken	50
7.12	Závislost doby dekompozice na počtu segmentů řídicích vláken	50
7.13	Závislost doby dekompozice na parametru <code>Interp.Subdivision</code>	51
7.14	Závislost doby dekompozice na parametru <code>SurfaceSubdivision</code>	52
7.15	Závislost doby dekompozice na parametru <code>VolumeSubdivision</code>	52
A.1	Běh programu	58
B.1	Dekompozice svalu <i>Iliacus</i>	59
B.2	Dekompozice svalu <i>Adductor Magnus</i>	59
B.3	Dekompozice svalu <i>Rectus Femoris</i>	60
B.4	Dekompozice svalu <i>Biceps Femoris</i>	60
B.5	Dekompozice svalu <i>Gluteus Maximus</i>	61
B.6	Nastavení exportu OBJ souboru v programu Blender	61

Upřesnění pojmů a zkratk

Pro porozumění následujícímu textu je zapotřebí definovat několik pojmů:

PCS	Physiological Cross-Section, neboli řez kolmý na průběh všech vláken. Umožňuje vyjádření počtu vláken ve svalu a tedy určuje maximální sílu svalu.
Pennate úhel	Úhel mezi vláknem svalu a osou stahu svalu.
Svalové vlákno	V kontextu dekompozice svalu jde o lomenou křivku v prostoru, která svým tvarem a průběhem připomíná vlákno skutečného svalu; Z pohledu anatomie jde o základní stavební jednotku příčně pruhovaných svalů.
Řídicí vlákno	Navzorkovaná data vybraného vlákna na svrchním povrchu svalu, použítá pro řízení dekompozice celého svalu
Vláknový řez	Jde o lomenou křivku danou množinou bodů získanou vybráním vrcholů se stejným indexem v lomených křivkách vláken v dané množině vláken, neboli tzv. iso-indexový řez.
Oblast úponu	Oblast na povrchu šlachy či kosti, ke které se sval upíná svým počátkem či koncem. Tato oblast je ohraničena hranicí oblasti úponu.
Bod úponu	Jeden bod v prostoru ležící na povrchu oblasti úponu. Ke každému bodu úponu je upnuto právě jedno vlákno.
Vícehlavý sval	Sval, který je rozštěpen na více částí či hlav. Tyto svaly mají na počátku jednu oblast úponu, ale na konci oblastí více. Mohou se také upínat k více kostem.
VPHOP	Osteoporotic Virtual Physiological Human, projekt číslo FP7 ICT-223865, zabývající se výzkumem zlomenin kostí u pacientů s osteoporózou.
VTK	Visualisation Toolkit, knihovna pro vizualizaci dat.
MAF	Multimod Application Framework, framework pro rapidní komponentový vývoj aplikací s medicínským zaměřením.

1 Úvod

Dekompozice virtuálního svalu na svalová vlákna je proces "naplnění" povrchového modelu svalu, v našem případě triangulované sítě, množinou lomených křivek, které představují a co nejlépe odpovídají svalovým vláknům reálného svalu. Takto dekomponovaný sval tak nabízí nové možnosti studia lidského pohybového ústrojí, například pro analýzu silového působení svalů, prevenci onemocnění pohybové soustavy či interaktivní trojrozměrné studium anatomie svalů při přípravě na lékařské povolání.

V oblasti počítačové grafiky zabývající se tímto problémem je již známo několik dekompozičních metod, tyto metody však mají některá omezení, která jsou dána dostupností vstupních dat a způsobem jejich využití. Cílem této práce je navrhnout novou dekompoziční metodu, která bude díky využití dodatečných řídicích dat schopna provádět dekompozice topologicky složitějších tříd svalů, zejména svalů upínajících se na počátku či konci k více oblastem úponů či dokonce více kostem, tedy svalů rozštěpených do více částí či hlav (tzv. vícehlavé svaly). Tato data je možné získat patologicky, nové metody v oblasti difuzní magnetické rezonance však v poslední době nabízejí slibné výsledky při získávání dat povrchových vláken svalu neinvazivně a bylo by tak v budoucnu možné tyto metody pro získání řídicích dat použít.

Navrhovaná metoda také využívá odlišný přístup k vytváření vláken, který díky vnitřní organizaci vláken do abstraktní trojrozměrné mřížky nabízí lepší kontrolu nad vlákny a možnost individuálních úprav jednotlivých vlákních, umožňujících dosažení kvalitních výsledků i pro složitější svaly.

Pro dekompozici využijeme navzorkovaných anatomicky přesných povrchových vláken jako dat pro řízení dekompozice. Tím je možné dosáhnout lepší aproximace tvaru vláken v celém objemu svalu, lepšího respektování průběhu vláken a způsobu jejich úponu, než je tomu u současných metod. Jako u ostatních metod se také budeme zabývat potlačením viditelných problémů, jakými jsou například nežádoucí ostré ohýbání vláken nebo křížení a kroucení vláken v objemu svalu.

V sekci 2 této práce se okrajově podíváme na anatomii lidských svalů, jejich klasifikaci a geometrické vlastnosti svalů. V sekci 3 rozebereme nám současně známé metody dekompozice a jejich omezení a příbuzné metody z této oblasti počítačové grafiky. V sekcích 4 a 5 provedeme popis a detailní teoretický rozbor navrhované metody, analýzu a řešení úskalí metody a adresaci požadavků na schopnosti metody vzhledem ke svalovým typům. V implementační části v sekci 6 provedeme analýzu, implementaci a výběr vhodné interpolační metody, na které je dekompoziční metoda silně závislá, probereme použité knihovny a kompatibilitu s frameworky a popíšeme některé detaily implementace. V sekci 7 provedeme vizuální analýzu výsledků metody a měření závislosti rychlosti dekompozice na vstupních datech a parametrech. Na závěr, v sekci 8, shrneme přednosti a nedostatky navržené dekompoziční metody.

2 Z anatomie svalů

Jelikož se tato práce zabývá dekompozicí svalů na svalová vlákna, podíváme se nejprve okrajově na lidské tělo z pohledu anatomie. Vzhledem k nám dostupným datům a účelu práce se budeme zabývat pouze příčně pruhovanými svaly, běžně označovanými také jako kosterními svaly. V literatuře jsou označovány také jako vůlí ovladatelné svaly (*Voluntary muscles*).

V Greyově Anatomii Lidského Těla [GL18] lze v kapitole IV nalézt popis těchto svalů. Tvoří je stahovatelné myofibrily (ne delší než přibližně 4 centimetry), uspořádané do delších svalových vláken, která jsou poté uspořádána do svalových snopců. Tyto pak probíhají délkou svalu, a na obou koncích se upínají ke šlachám a ty poté ke kostem. Pro naše účely je velmi důležité hledat a pozorovat geometrické a topologické vlastnosti vláken pro různé typy svalů.

2.1 Dělení dle průběhu vláken

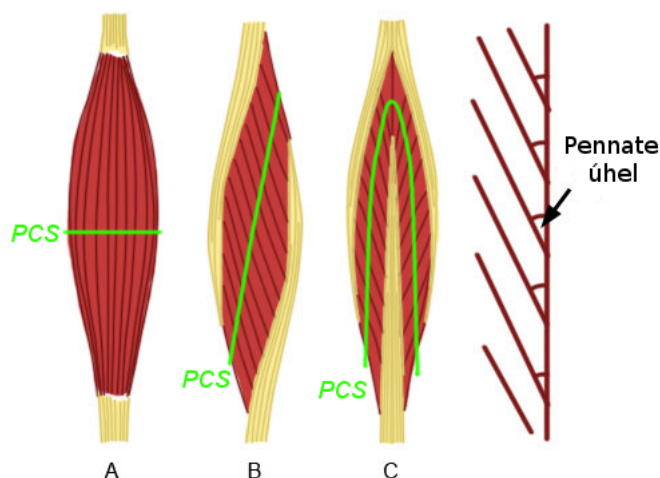
Podle průběhu vláken délkou svalu a dle způsobu jejich upnutí k šlachám lze svaly rozdělit do několika skupin. Na obrázku 2.1 lze vidět náčrtek těchto typů svalů. Lze také vidět tvar fyziologického průřezu svalu (*Physiological Cross-Section, PCS*), který představuje řez "přes vlákna", tedy řez kolmý na průběh vláken. Na rozdíl od anatomického řezu, který je kolmý na osu stahu svalu, má řez PCS pro různé typy svalů pozorovatelné geometrické vlastnosti a bude užitečný pro korektní získávání dat a některé myšlenky dekompozičního algoritmu.

Obecná dekompoziční metoda by samozřejmě měla průběh vláken a způsob či úhel jejich úponu co nejvěrněji respektovat. Současné metody však průběh vláken odhadují z tvaru povrchu svalu a orientace úponových oblastí, zpravidla tedy například pennate úhel zmíněný v sekci 2.1.2 příliš nerespektují.

2.1.1 Svaly s rovnoběžnými vlákny

Vlákna těchto svalů jsou z geometrického pohledu paralelní k ose stahu svalu a oblast úponu je téměř kolmá k vláknům svalu. Tyto svaly lze pak dále dělit na pásové svaly, které jsou velmi dlouhé a tenké (například sval Krejčovský - *Sartorius*), fusiform svaly které se na koncích zužují a na uprostřed se rozšiřují (příkladem jsou například hlavy většiny bicepsů) a konvergentní či trojúhelníkové svaly, které jsou na počátku vějířovité a na konci zúžené (například sval Kyčelní - *Iliacus* nebo prsní svaly).

Směr PCS těchto svalů je přibližně rovnoběžný s oblastí úponů vláken ke kosti. Tyto svaly se vyznačují velkým rozsahem stahu.



Obrázek 2.1: Typy svalů dle průběhu vláken (převzato z [GL18]). A - Fusiform sval, B - Unipennate sval, C - Bipennate sval. PCS označuje fyziologický průřez svalu. Převzato a upraveno z [http://biomechanics.byu.edu/exsc362\(hunter\)/chapter11.html](http://biomechanics.byu.edu/exsc362(hunter)/chapter11.html)

2.1.2 Pennate svaly

Vlákná s osou stahu svalu svírají určitý úhel (*pennate angle*). Vlákna jsou tedy ke šlaše upnutá šikmo, což svalu umožňuje vyvinout větší sílu, než v případě svalů s rovnoběžnými vlákny, ale za cenu menšího rozsahu stahu. Tyto svaly lze dále dělit do podskupin:

- Unipennate - tyto svaly mají jednu skupinu vláken upnutou pod daným úhlem. PCS těchto svalů již není nutně rovnoběžný s oblastmi úponu vláken (či se šlachou), díky pennate úhlu. Příkladem jsou natahovače prstů - *Extensor digitorum longus* v dolních končetinách a *Extensor digitorum communis* v předloktí.
- Bipennate - tyto svaly průběhem vláken připomínají ptačí pero. Mají dvě skupiny vláken pod dvěma pennate úhly. Řez PCS je ohnutý či zlomený, to je způsobeno právě zmíněným rozdělením vláken do dvou částí. Příkladem je přední hlava čtyřhlavého svalu stehenního (hlava *Rectus femoris* svalu *Quadriceps femoris*).
- Multipennate - topologicky jde o nejsložitější svaly. Vlákna jsou ke šlaše upnuta pod mnoha úhly vzhledem k ose stahu svalu. Příkladem je sval *Deltoid*.

Na obrázku 2.2 jsou ilustrace lidských pennate svalů s vyznačením průběhu šlachy a vláken ve svalu. Zelená křivka vyznačuje šlachy (přesněji jde o aponeurózu), modrá křivka značí průběh vláken. Je patrný úhel, který tyto dvě křivky svírají, který je velmi ostrý.



Obrázek 2.2: Lidské pennate svaly. (a) Natahovač prstů (unipennate) (b) Čtyřhlavý sval stehenní (bipennate). Převzato a upraveno z [ČGMH01]

2.1.3 Kruhové svaly

Jde o svaly, které mají kruhový průběh vláken. Jde o svaly očí, úst a o svěrače. Pro tyto typy svalů však nemáme data a proto se jimi nebudeme podrobněji zabývat. V závěru práce (v sekci 7.1) však ukážeme, že navrhovaná dekompoziční metoda pro tyto svaly s menší úpravou teoreticky také vyhovuje.

2.2 Dělení dle počtu oblastí úponu

Dalším faktorem, dle kterého lze svaly rozdělit, je počet oblastí či šlach, ke kterým jsou vlákna svalu na počátku či konci svalu upnuta. Například každá hlava vícehlavých svalů (jako bicepsy, tricepsy atd) je zpravidla upnuta k odlišné oblasti na odlišné kosti. Dalším příkladem může být velký přitahovač (*Adductor Magnus*), upínající se k různým oblastem na stehenní kosti (*Femur*).

Obecná dekompoziční metoda by tuto vlastnost svalů měla také respektovat, a vlákna v objemu svalu věrohodně rozdělit tak, aby sval pokryl všechny své oblasti úponu. Současné metody jsou však zpravidla schopny pracovat pouze se dvěma oblastmi, tedy pouze s oblastí počáteční a koncovou.

3 Současné metody a projekt VPHOP

3.1 Reprezentace povrchu svalu a povrchových vláken

V mnohých real-time aplikacích jsou svaly reprezentovány pomocí parametrického povrchu, například pomocí válce ([WG97], [TSC96]) či pomocí kvadratické Beziérovovy plochy ([LA07]). Svaly reprezentované pomocí těchto přístupů příliš neodpovídají realitě, nicméně jejich deformace je relativně nenáročná. Pro anatomicky přesnější přístup jsou zpravidla využívány trojúhelníkové povrchové modely (například pro deformace svalů [KKM12], nebo při rekonstrukci povrchu svalů [LRJ⁺12]).

Průběh vláken na povrchových modelech svalů lze získat pomocí klinické metody *Diffusion Tensor Imaging (DTI)* nebo také *Diffusion Magnetic Resonance Imaging (dMRI)*, což je neinvazivní způsob zkoumání procesu rozptylování vodních molekul v tkáních s různými vlastnostmi a detekce jejich chování při kontaktu s překážkami, vlákny, membránami a podobně. Autoři článku [LDW⁺07] ukázali, že lze tuto metodu aplikovat mimo jiné také pro získání geometrie jednotlivých svalových vláken na povrchu svalu. Autoři článku [LGMP11] však poukázali na nízký poměr signálu k šumu metody DTI a navrhli velmi komplexní metodu pro zlepšení jejich výsledků. Na poli zkoumání vlastností svalů však tato metoda může teoreticky být využívána pro získání dat o struktuře a průběhu vláken svalů pacienta, bezpečně a neinvazivně.

3.2 Reprezentace vnitřní struktury svalů

V oblasti biomedicínské počítačové grafiky je však důležitá i reprezentace vnitřní svalové struktury a jednotlivých svalových vláken. Ng Thow Hing [Ng 01] reprezentuje svaly pomocí B-spline tělesa, jehož tvar je přizpůsoben tvaru povrchových dat svalu a dat patologicky získaných vzorků vláken. V této reprezentaci pak lze nalézt jednotlivá vlákna jako iso-křivky probíhající od počátku svalu k jeho konci. Metoda však nabízí malou kontrolu nad tvarem svalu, autoři článku [LZRM11] však tento problém řeší použitím snadněji deformovatelných NURBS těles. V obou případech je ale přizpůsobování tvaru tělesa tvaru svalu složité a nepraktické.

Populární reprezentací modelů svalu je použití FEM sítí (*Finite Element Mesh*). Sval je v tomto případě reprezentován trojrozměrnou sítí složenou z hexahedronů či tetrahedronů. Každá buňka této sítě obsahuje informace o směru průběhu vláken a je poté využita jejich rekonstrukci. Změnou tvaru této sítě se pak automaticky mění tvar i vnitřní vlákna svalu. Příkladem použití FEM sítí je článek [LZRM11], ve kterém autoři využili NURBS reprezentace povrchu svalu pro vygenerování směrů vláken v buňkách odpovídající FEM sítě, kterou pak již lze použít pro deformaci svalu.

Autoři článku [BD05] také využívají FEM sítě pro reprezentaci svalů, jejich řešení však průběh vláken aproximuje umělými, ale anatomicky korektními šablonami vláken. Tyto šablony jsou pro každý typ svalu dány množinou o libovolném počtu B-spline křivek uvnitř jednotkové krychle, které reprezentují jednotlivá vlákna. Geometrie těchto křivek je poté namapována na jednotkovou krychli reprezentovanou pomocí ručně vytvořené FEM sítě. Tato síť je poté pomocí komerčního softwaru TrueGRID (XYZ Scientific Application) mapována pomocí projekce na povrchový model cílového svalu. Výsledná FEM síť pak svým tvarem odpovídá tvaru povrchového modelu svalu, ale díky mapování jsou deformovány i B-spline křivky reprezentující vlákna.

3.3 VPHOP a metody dekompozice povrchových modelů svalů

Dekompozice povrchového modelu svalu na svalová vlákna je proces "naplnění" povrchového modelu svalu množinou lomených křivek, které představují a co nejlépe odpovídají svalovým vláknům reálného svalu. Jednou z aplikací, ve které se dekomponované svaly využívají, je projekt VPHOP [VPH10], který se zabývá predikcí zlomenin kostí u pacientů s osteoporózou, vytvořením virtuálního hypermodelu pacientova těla, simulací pohybu a měřením stresových sil působících na kosti a chrupavky. Pro pohled "dovnitř" svalů je tedy vhodné znát vnitřní strukturu svalů a tedy i tvar a průběh svalových vláken. Jelikož metody implementované pro projekt VPHOP souvisejí s navrhovanou metodou nejvíce, probereme si je v následujících odstavcích podrobněji.

3.3.1 Advanced Slicing Method (ASM)

První dekompoziční metodou je Advanced Slicing Method (ASM) [KCMV12a, KCMV12b] využívající povrchového modelu svalu pro odhad průběhu vláken svalem. Tato metoda využívá šablon vláken převzatých z metody [BD05], které reprezentují průběh svalových vláken B-spline křivkami. Tato krychle je nejprve transformována tak, aby odpovídala osově zarovnanému ohraničujícímu kvádru (AABB) pomocí afinní transformace. Nejprve je tedy určena hlavní osa svalu jako úsečka mezi centroidem počáteční a centroidem koncové oblasti úponu, tato je pak ztotožněna se svislou osou krychle křivek, a poté je krychle otáčena kolem této osy, dokud není nalezeno nejvhodnější zarovnání oblastí úponů s odpovídajícími stranami krychle pro její transformaci v osově zarovnaný ohraničující kvádr.

Poté je provedena sekvence řezů kvádru a povrchového modelu svalu, v obou případech kolmých na hlavní osu. Na každém řezu je pak obdélník daný řezem kvádru rozčleněn tak, aby počet vrcholů nového obdélníkového polygonu odpovídal počtu vrcholů polygonu daného řezem povrchového modelu svalu. Tyto dva polygony jsou poté ztotožněny morphingem, a pomocí obecných barycentrických souřadnic jsou také morphovány body vnitřních vláken na daném řezu. Postupným posouváním řezu je tak morphován celý sval.

3.3.2 Kukačková metoda

Tato metoda [KCea12] se od předchozí metody liší ve způsobu řezání povrchového modelu svalu. Na povrchu svalu je nejprve sestaveno harmonické skalární pole [DKM05] které se šíří a rovnoměrně interpoluje skalární hodnoty ve vrcholech povrchového modelu od počátku svalu k jeho konci. V tomto poli jsou poté vyhledány isohodnoty, kterými jsou proloženy řezy. Tento přístup tak provádí řezy povrchovým modelem s ohledem na jeho tvar¹, což vede na přesnější dekompozici. Skalární pole však může díky tvaru svalu vytvářet nechtěné rotace rezných rovin, které vede na zkroucení vláken uvnitř svalu.

3.3.3 Morphed Mass Spring System

Tato metoda zvaná Morphed Mass Spring System(MMSS) [KCea12] využívá částicového systému, ve kterém jsou částice v prostoru drženy pružinami, a pospojovány do vláken dodatečnými pružinami. Podobně jako v metodě ASM, vnější (povrchové) částice jsou řez po řezu upevněny na povrch svalu. Poté je pružinový systém relaxován a jednotlivé částice se díky pružinám rovnoměrně rozmístí po objemu svalu.

3.3.4 Omezení metod

Omezením výše uvedených metod je, že jsou schopny korektně dekomponovat pouze svaly, které se na obou koncích upínají pouze k jedné oblasti úponu, tedy jedné šlaše či kosti. V případě svalů s více oblastmi je tedy nutné se buď omezit pouze na zpravidla plošně největší oblast úponu pro oba konce svalu, a ostatní oblasti úponu zanedbat, což vede na nekorektní dekompozici rozštěpených svalů (muže tak být oříznuta celá část svalu), nebo oblasti spojit a zcela tak ignorovat rozštěpení svalu.

¹Metoda ASM tvar svalu aproximuje pouze hlavní osou svalu

4 Přehled navrhované metody

V kontrastu s metodami popsány v sekci 3, popisovaná metoda využívá do-datečných dat pro dekompozici povrchového modelu svalu na jednotlivá vlákna. Tato data, nazvaná řídicí vlákna, představují soubor lomených křivek reprezentujících svalová vlákna na zpravidla svrchním² povrchu svalu. Z těchto dat lze poté odvodit průběh vláken v celém objemu svalu, s respektováním typu svalu a jeho vlastností. Popis metody v této sekci je pouze stručným souhrnem průběhu dekompozice, popisem a ilustracemi jednotlivých fází dekompozice a vysvětlením použitých myšlenek se budeme do detailu zabývat v následující sekci 5.

Vstupem metody je povrchový model svalu, řídicí vlákna, hranice oblastí úponů vláken ke kostem a povrchový model kostí, ke kterým se sval upíná. Tato data byla vytvořena v rámci projektu LHDL financovaného Evropskou komisí a jsou volně dostupná z <https://www.physiomespace.com/>. Vstupními parametry metody jsou počet nově generovaných vláken mezi vlákny na povrchu a v objemu svalu, rozlišení vláken po jejich délce pro jejich vyhlazení, výběr metody interpolace vláken a další parametry, které zavedeme v průběhu sekce 5. Výstupem metody je soubor lomených křivek reprezentujících jednotlivá vlákna v objemu svalu, tedy výsledek dekompozice.

Prvním krokem metody je předzpracování dat oblastí úponů. Každé vlákno výsledné dekompozice je nutné upnout k vhodnému bodu v těchto oblastech. Pro cílový počet vláken tedy vytvoříme počet bodů v těchto oblastech. Předpokládáme, že hranice oblastí jsou definovány skupinou bodů ležící na povrchových modelech kostí. Z těchto modelů tedy vyřízneme tu část, kterou hranice ohraničuje, tedy extrahujeme trojúhelníky pro tuto oblast vnitřní. Na tomto výřezu poté vygenerujeme, pokud možno rovnoměrně, nové body úponu vláken.

Následuje rekonstrukce povrchových vláken svalu. Mezi řídicí vlákna vložíme ("rozmnožíme") pomocí příčné interpolace nová vlákna (dle parametru rozlišení povrchu) jejichž tvar je dán právě řídicími vlákny. V těchto vrcholech povrchových vláken nalezneme povrchové normály jako průměr normál trojúhelníků definovaných okolními vrcholy daného vrcholu.

Ve směru těchto normál pokračujeme vzorkováním objemu svalu. Vrháme paprsky v opačném směru normály z vrcholů povrchových vláken směrem do objemu povrchového modelu svalu. Hloubka vzorku je pak odečtena jako vzdálenost mezi vstupním a výstupním průsečíkem paprsku s povrchem. V případě nekonvexnosti či otvoru v modelu svalu vzdálenosti nalezených úseků sčítáme, a tento součet označíme jako hloubku svalu.

²Patologický způsob získání trojrozměrných dat z mrtvého těla výrazně komplikuje či dokonce nedovoluje extrakci dat z hlubších vrstev svalstva resp. hlouběji uložených svalů. Je nutný postupný přístup od povrchu směrem ke kosti

V případě, že je vzorek hloubky menší, než prahová hodnota, detekujeme situaci, kdy je povrchové vlákno mimo povrchový model svalu. Tato situace značí zpravidla dva případy:

- Chyba ve vstupních datech způsobila, že některé řídicí vlákno je mimo vzorkovatelný povrch svalu. O této situaci je informován uživatel a dekompozice pokračuje, produkuje však zpravidla nekorektní výsledky. Ty je možné vizualizovat a chybu v řídicích vláknech či modelu svalu identifikovat
- Sval má více oblastí úponů a tedy detekujeme situaci, kdy vygenerované povrchové vlákno leží nad oblastí rozštěpu svalu. V tomto případě nalezneme řídicí vlákna na obou stranách rozštěpu a povrchová vlákna mezi nimi rozdělíme a upravíme tak, aby byl i ve vláknech rozštěp zaveden

Poté generujeme vlákna v objemu svalu, generováním nových vrcholů pomocí lineární interpolace ve směru normál do vzdálenosti navzorkované hloubky svalu. Pospojování vrcholů do lomených křivek odpovídá vláknům na povrchu svalu, čímž nedochází k problémům, jako například kroucení vláken. Vlákna v objemu svalu generujeme po vrstvách, můžeme tak s nimi i nadále pracovat a zlepšovat kvalitu dekompozice. Díky případnému rozštěpení povrchových vláken se také analogicky štěpí i objemová vlákna. Pro zaoblení svalu také vygenerujeme vlákna po stranách svalu.

Dalším krokem je upnutí vláken k vygenerovaným bodům v oblastech úponu. Každé vlákno lze upnout pouze na jeden takový bod. S výběrem vhodné metriky kvality úponu nalezneme nejvhodnější přiřazení vláken k bodům úponu. Výslednou celkovou kvalitu upnutí vláken zkontrolujeme měřením úhlu zlomení vláken při upnutí. Příliš zlomená vlákna nejsou žádána, tento problém však může nastat u vnitřních vláken svalů s ostrým pennate úhlem, jelikož metoda o průběhu vnitřních vláken nemá dostupná data. Postupným zkracováním vnitřních vláken po vrstvách však lze tento problém vyřešit, jelikož se vlákna .

Posledním bodem algoritmu je vyhlazení jednotlivých vláken, tedy jejich podélná interpolace, pomocí vhodné interpolační metody. Vygenerovaná vlákna poté přeneseme na výstup.

Postup metody lze shrnout v pseudokódu 1.

```
// Příprava bodu na oblastech uponu
PrepareAttachments(VolumeSubdivision * SurfaceSubdivision
                  * InputFiberCount);
// Rekonstrukce povrchových vláken
InterpolateSurface(SurfaceSubdivision, InterpolationMethod);
// Vypocet normal ve vrcholech povrchových vláken
ComputeNormals();
// Vzorkovani objemu svalu
ComputeThickness();

// Vytvoreni vláken po vrstvach uvnitr objemu
float step = 1 / VolumeSubdivision;
for (float percent = 0; percent <= 1; percent += step) {
    // Vytvoreni vrstvy vláken
    BuildLayer(percent);
    // Zpracovani vrstvy vláken (poruseni pravidelnosti)
    OffsetLayer(percent);
}
// Vytvoreni postrannich vláken
BuildSideLayers();

// Pripojeni vláken k bodum uponu
ConnectAttachments();

// Kontrola uhlu zlomeni vláken
while (!CheckConnection())
{
    // Zkraceni vláken
    ShortenFibersByLayers();
    // Nove pripojeni vláken k bodum uponu
    ConnectAttachments();
}

// Finalni vyhlazeni
InterpolateFibers(InterpolationSubdivision, InterpolationMethod,
                 OUT OutputFibers);
return OutputFibers;
```

Zdrojový kód 1: Pseudokód navrhované metody

5 Podrobný popis navrhované metody

5.1 Vstupní data

V této kapitole se budeme zabývat požadavky na vstupní data, analýzou dat, která jsou nám k dispozici a způsobu získávání umělých i reálných dat.

5.1.1 Požadavky na vstupní data

Pro korektní běh navrhované metody je nutné dodržet následující požadavky na vstupní data:

- Povrchový model dekomponovaného svalu a modely kostí pro extrakci povrchu oblasti úponů musí být manifoldní uzavřené trojúhelníkové síť.
- Oblasti úponů musí být definovány hranicí oblasti, uspořádanou množinou bodů
- Řídící vlákna by měla ležet na svrchním povrchu svalu, nesmí se křížit či kroutit, nesmí sdílet vrcholy, měla by pokrýt co největší část svrchního povrchu (jelikož rekonstrukce povrchu probíhá pouze mezi těmito vlákny), koncové body vláken nesmí přesahovat za oblasti úponů a v případě rozštěpení svalu je nutná přítomnost alespoň jednoho řídicího vlákna na povrchu každé hlavy svalu. Vlákna také musí být prostorově seřazena tak, aby příčný přechod přes vlákna odpovídal postupnému posunu po povrchu svalu³. Řídících vláken musí být nejméně dvě, ideální je však dekompozice pomocí čtyřech a více řídicích vláken.

Řídící vlákna musí mít stejný počet vrcholů, nejlépe alespoň pět, a jednotlivé vláknové řezy by měly v ideálním případě odpovídat (v případě rozštěpení svalu i nespojitým) řezům PCS svrchní vrstvy vláken skutečného svalu (z důvodu popsaného v sekci 5.5.2). Řídící vlákna mohou protínat povrchový model svalu. Lomené křivky vláknových řezů řídicích vláken mohou částečně ležet mimo sval pouze v oblastech štěpení svalu.

Průběh vláken by měl odpovídat anatomickému průběhu skutečných svalových vláken. Počátek a konec vláken, a tedy směr vlákna či směr růstu indexů lomené křivky vlákna, musí odpovídat označení počátečních a koncových oblastí úponu (vlákna musí probíhat od počátečních oblastí ke koncovým).

Demonstrační aplikace, která je součástí této práce podporuje pouze data v souborech formátu VTK [Kol10].

³Jinými slovy, vlákna nesmí být "na přeskáčku"

5.1.2 Dostupná data

Součástí datového balíku projektu VPHOP, který byl pro tuto práci použit, jsou následující vstupní data:

- Povrchové modely kostí a svalů - v aktuální verzi datového balíku jsou tato data již velmi dobrá, modely jsou manifoldní, neobsahují díry, ostré hrany ani jiné artefakty. K dispozici jsou modely kostí a svalů spodní části těla. Modely jsou velmi jemné (řádově tisíce až desetitisíce trojúhelníků), k dispozici jsou však také decimované verze některých modelů⁴. Tyto modely byly získány rekonstrukcí povrchových dat z MRI obrazů a ručním odstraněním artefaktů.

Menší nepříjemností dostupných povrchových dat vícehlavých svalů je fakt, že některé vícehlavé svaly jsou rozděleny do více povrchových modelů, pro každou hlavu oddělený model (například *Quadriceps femoris* je rozdělen do čtyř povrchových modelů), ale některé mají hlavy v jednom povrchovém modelu (například *Biceps femoris*). Tím se jejich dekompozice pro obecné dekompoziční metody komplikuje, jelikož je v této situaci nutné vlákna uvnitř svalu rozštěpit bez informace o tom, která část povrchového modelu patří které hlavě svalu.

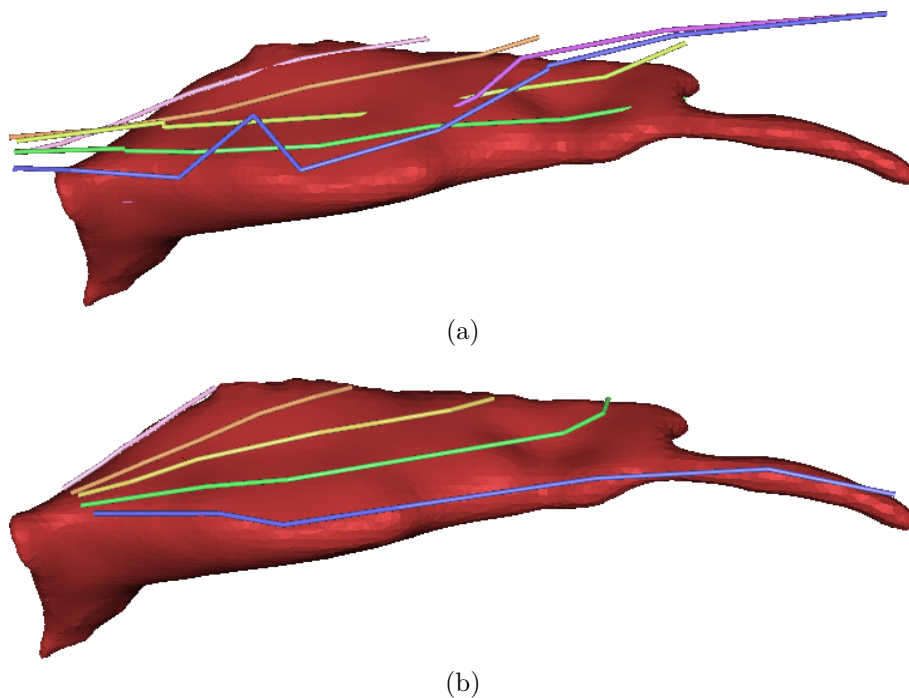
- Oblasti úponů - tyto oblasti jsou reprezentovány množinou bodů na povrchu modelů kostí, které označují hranici oblasti, ke které je sval k dané kosti upnut. Není k dispozici konzistentní a neměnná indikace o příslušnosti oblasti ke svalu či kosti (pouze měnitelný slovní popis), ani zda jde o oblast začátku svalu či koncového úponu svalu (opět pouze měnitelný slovní popis).
- Řídící vlákna - původním záměrem při návrhu metody bylo využít i dostupných řídicích dat. Tato data byla změřena na povrchu svalů, ručním sledováním vybraného vlákna a postupným zaznamenáváním několika vzorků polohy vlákna po jeho délce. Každé vlákno je pak, dle očekávání, reprezentováno lomenou čarou.

Analýzou dostupných dat, která byla provedena v rámci řešení této diplomové práce, bylo zjištěno, že data řídicích vláken obsahují mnoho chyb, jakými jsou křížení vláken, nespojitosti způsobené nekorektním pospojováním bodů, v některých případech dochází i k chybné indexaci a spojení dvou vláken v jednom vrcholu. Řídící vlákna také často zcela zjevně nevystihují vlastnosti svalu, například úhel pennace nebo rozštěpení svalu. Vlákna také neodpovídají svým tvarem povrchovým modelům a ve většině případů řídicí vlákna leží zcela mimo povrchový model svalu, což je zřejmě způsobeno

⁴Vytvořené decimacním filtrem v Blenderu [Ble13]

odlišnou polohou subjektu při získávání povrchových modelů svalů a dat řídicích vláken. Data také nejsou dostupná pro více než polovinu svalů. Z tohoto tedy vyplývá, že data nejsou pro navrhovanou dekompoziční metodu použitelná, rozhodl jsem se tedy řídicí data pro vybrané svaly ručně vymodelovat.

Ukázka dostupných dat je na obrázku 5.1a. Patrné jsou také nedostatky, jako křížení a nepovrchovost vláken. Rozštěpení svalu v povrchovém modelu také neodpovídá rozštěpení v řídicích vláknech, data mají kromě odlišné pozice i jiný tvar. Na obrázku 5.1b jsou ručně vytvořená řídicí data bez těchto problémů.



Obrázek 5.1: Ukázka dat vstupních řídicích vláken. (a) Data dostupná v balíku VPHOP. Jsou patrné nedostatky řídicích dat a jejich nízká srovnatelnost s povrchovým modelem. (b) Uměle vytvořená data

5.1.3 Získávání dat

Dostupná řídicí data, která byla získána pitevně z lidského kadaveru, bohužel nejsou dostatečně kvalitní pro použití v navrhované dekompoziční metodě. Bylo by sice možné měření provést znovu, přesněji, zejména ve stejné pozici kadaveru při měření polohy vláken a získávání povrchových modelů těchto vláken. Tato metoda je však velmi invazivní, na živém pacientovi samozřejmě neproveditelná a původní kadaver byl samozřejmě pohřben. Získávání vstupních dat

tímto způsobem je také mimo naši specializaci a vzhledem k časové náročnosti a omezenému počtu expertů v tomto oboru tento způsob získávání dat není finančně únosný ani dlouhodobě praktický. Obtížné je také získat dobrovolný souhlas člověka, který by po jeho smrti umožnil toto použití jeho těla pro vědecké účely.

Nabízí se tedy možnost použití DTI pro získání dat povrchových vláken metodami popsanými v článkách [LDW⁺07] a [LGMP11]. Tyto metody vykazují pro umělá data velmi slibné výsledky a bylo by tedy teoreticky možné je v budoucnu použít pro získání skutečných řídicích vláken na povrchu svalu. Je však pravděpodobné, že by byl nutný zásah experta tak, aby takto získaná vlákna odpovídala požadavkům ze sekce 5.1.1. Šlo by však o bezpečnou, neinvazivní metodu získávání těchto dat.

Pro účely této práce tedy použijeme uměle vytvořená data (příkladem je obrázek 5.1b). Tato data jsem ručně vytvořil v aplikaci Blender [Ble13] dle mého nejlepšího umu; nejsem však expertem v oboru anatomie a takto vytvořená vlákna jsou tedy pravděpodobně použitelná pouze pro testovací účely. Postup vytváření vláken lze nalézt v sekci 6.5.

5.2 Předpříprava dat

V předpřípravné části metody jsou zpracována data oblastí úponu svalů ke kostem. Na těchto oblastech je nutné vygenerovat dostatek bodů úponu, ke kterým budou v závěru metody upínána dekomponovaná vlákna (vizte sekci 5.7). Dostupná vstupní data se však sestávají pouze z bodů hranice oblastí a povrchových modelů kostí. Potřebujeme tedy nejprve získat povrch, na kterém bude možné body úponu generovat.

Z pohledu anatomie se svalové vlákna upínají ke šlachám, nikoliv ke kostem. Data tvaru šlach však zpravidla nejsou v datech, která nám jsou k dispozici, dostupná⁵, což v případě velmi komplexních svalů může představovat problém. Pokud by však tato data byla pro všechny svaly dostupná (tzn. ve formě povrchového modelu šlach), pro korektní dekompozici by bylo možné provést sjednocení těchto dat s povrchovými modely kostí a povrch oblasti úponu pak hledat na povrchu tohoto sjednocení. Bylo by možné tato data také ručně vymodelovat, to však vyžaduje hlubší znalost anatomie pohybové soustavy. Tvar vláken na povrchu svalu je dobře ilustrován v anatomických atlasech a vytváření těchto dat je tedy relativně jednoduché; modelování trojrozměrných povrchových modelů šlach, kdy v některých svalech tyto šlachy procházejí částí objemu svalu, je záležitost podstatně složitější).

Nejpřesnější způsob, jak povrch oblastí úponu generovat, je z povrchových modelů kostí vyříznout trojúhelníky, které oblasti úponu svalu náleží. Pro tento účel

⁵Je dostupný pouze povrchový model kolenní šlachy čtyřhlavého stehenního svalu

je již v projektu VPHOP implementován filtr `vtkMAFPolyDataCutOutFilterEx`, který body hranice vloží do povrchového modelu kosti jako nový vrchol, tedy vyhledá nejbližší trojúhelník k bodu hranice a tento trojúhelník rozdělí na tři trojúhelníky definované vrcholy tohoto trojúhelníka a vloženým vrcholem. Filtr pak vychází z prvního bodu hranice, a pomocí Dijkstra algoritmu hledá nejkratší cestu po hranách sítě k dalšímu bodu hranice. Proces se pak v dalším bodu hranice opakuje, dokud není známa uzavřená cesta po hranách sítě obsahující všechny body hranice.

Poté je vybrán trojúhelník který sousedí s hranicí oblasti zprava a jsou prohledány jeho sousedé, za předpokladu, že při přechodu na souseda algoritmus nepřejde přes hranu náležící hranici oblasti. Tito sousedé jsou také označeni a vloženi do fronty. Algoritmus trojúhelníky z fronty vybírá a prohledává neoznačené sousedy, opět se zakázáním přechodu hranice oblasti. Výstupem filtru jsou pak dvě části, síť označených trojúhelníků a síť neoznačených trojúhelníků. Jelikož je načítání dat kostí pro vyřezávání pro jemné sítě časově náročné a jelikož se ale oblasti úponu ani kosti nemění, lze podstatně menší výřez sítě uložit do souboru a později znovu použít.

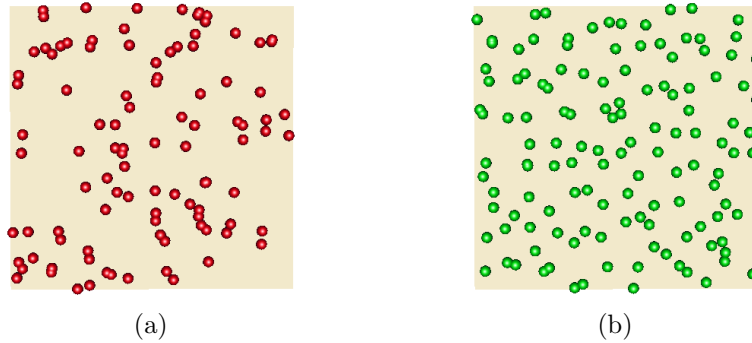
Filtr `vtkMAFPolyDataCutOutFilterEx` však na vstupní data pokládá dodatečné požadavky. Body hranice oblasti musí být seřazeny po obvodu hranice, hranice musí být spojitá a uzavřená, body hranice musí ležet na povrchu kosti a povrch kosti uvnitř oblasti nesmí obsahovat díry. Pro data nám dostupná jsou však tyto požadavky zpravidla splněny. Narazil jsem však několik případů, kdy data požadavky nesplnila, jmenovitě oblast úponu velkého hýžd'ového svalu (*Gluteus Maximus*) na křížové kosti, jejíž model obsahuje tvarem kosti způsobené díry či oblast úponu svalu *Vastus Intermedius* na stehenní kosti, která není uzavřená.

My předpokládáme, že síť s menší plochou je námi hledaný výřez oblasti úponu svalu, jelikož se svaly zpravidla neupínají na nadpoloviční část plochy kosti. Na tomto výřezu nyní můžeme generovat body úponu.

Prosté náhodné vybírání trojúhelníků pro vygenerování bodů úponů však není příliš vhodné, jelikož může produkovat shluky bodů a prázdná místa, což je způsobeno nedokonalým rozložením generovaných náhodných čísel. Ukázka tohoto nevhodného přístupu je na obrázku 5.2a, kde jsme vygenerovali 100 bodů na čtverci o 128 trojúhelnících. Je patrné, že distribuce bodů není vhodná.

Zvolili jsme tedy přístup, který bere v potaz plochu jednotlivých trojúhelníků. Zkonstruujeme mapu, ve které počet polí přiřazených trojúhelníku odpovídá jeho relativní ploše vzhledem k nejmenšímu trojúhelníku v síti. Mapu pak rozdělíme na počet úseků odpovídajících požadovanému počtu generovaných bodů úponu, a v každé oblasti vybereme trojúhelník (v úseku první), na kterém bod vygenerujeme. Plocha trojúhelníka tedy koresponduje s pravděpodobností, že bude trojúhelník vybrán pro generování bodu úponu.

Pokud jsou trojúhelníky sítě dostatečně malé, je generování rovnoměrnější, než v případě prostého náhodného výběru trojúhelníka. V případě velkých trojúhelníků



Obrázek 5.2: Ilustrace indexace vláken (a) Příímý způsob indexace (b) Obrácený způsob indexace. Červeně je načrtnut sval z pohledu na jeho svrchní vrstvu. Vlákna leží na svrchním povrchu svalu.

se však výhoda mapy bohužel ztrácí - vzhledem k používaným datům, která mají dostatečné rozlišení, toto však není výrazným problémem.

Pro vybrané trojúhelníky je pak vygenerována pseudonáhodná sada barycentrických souřadnic, s jejichž pomocí je vygenerován na povrchu trojúhelníka nový bod úponu. Ukázka distribuce bodů této metody je na obrázku 5.2b. Distribuce je mnohem pravidelnější, pro nás zcela dostačující.

Na každé oblasti úponu vygenerujeme body pro upnutí všech vláken svalu, čímž zajistíme, že jich bude dostatek a že v případě štěpení svalu nebudou vlákna nucena upínat se k nevhodné oblasti. Spoléháme tak na dobře navržená či získaná řídicí vlákna a na metodu, která umožní vhodné upnutí vláken pouze dle jejich tvaru a směru. Toto však navrhovaná dekompoziční metoda dovoluje díky respektování průběhu řídicích vláken.

5.3 Konstrukce vláken na povrchu svalu

Prvním krokem samotné dekompozice svalu na vlákna metody je konstrukce vláken svrchního povrchu svalu. Pro každý vláknový řez⁶ řídicích vláken provedeme interpolaci pomocí parametrické spline interpolační metody a na každém segmentu získané příčné interpolační křivky vygenerujeme s lineárním krokem parametru nové vrcholy pro nová vlákna. Vlastnosti různých vyzkoušených metod interpolace jsou popsány v implementační části v sekci 6.1. Jelikož má parametrická spline interpolace implicitně definováno seřazení podle parametru, můžeme podle tohoto seřazení konstruovat vlákna.

Předpokládejme n řídicích vláken o m úsecích lomené křivky, a parametr rozdělení povrchu $R_p \in N$, tedy počet požadovaných vláken mezi původními

⁶Vizte prosím definici vláknového řezu v sekci *Upřesnění pojmů a zkratk*

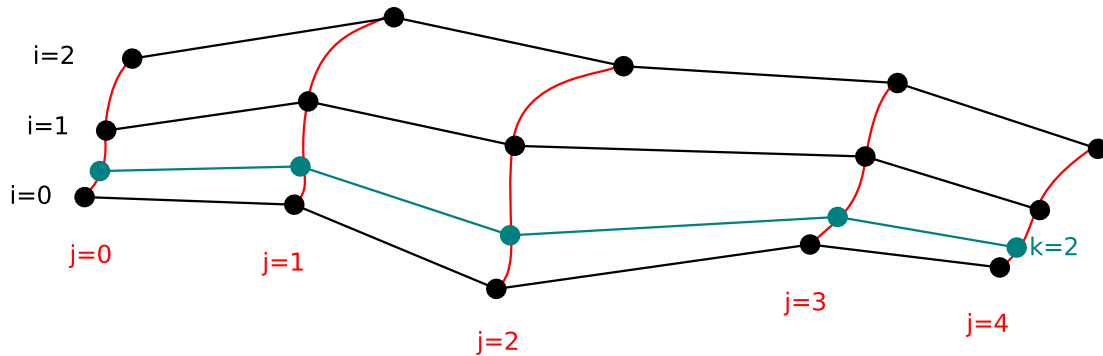
řídícími vlákny. Pro každý vláknový řez $0 < j < m$ provedeme spline interpolaci nad všemi n vrcholy řezu. Tím získáme m příčných spline interpolační křivek, každá o $n - 1$ segmentech. Jednotlivé segmenty tedy můžeme označit S_{ij} , tedy i -tý segment j -té příčné spline interpolační křivky, $0 \leq i < n$.

Pro každý segment $S(t)_{ij}$ známe počáteční hodnotu parametru t_{ij0} a koncovou hodnotu parametru t_{ij1} . Na segmentu tedy vygenerujeme body s parametrem t_{ijk} , kde $t_{ij0} < t_{ijk} < t_{ij1}$ a

$$t_{ijk} = t_{ij0} + k * \left(\frac{t_{ij1} - t_{ij0}}{R_p} \right),$$

kde $k \in N, 0 < k < R_p$. Jde tedy o lineární interpolaci parametru t_{ijk} v hranicích segmentu, s krokem daným rozdílem parametrů v koncových bodech segmentu a počtem kroků R_p .

Pro každé i a k tedy získáme množinu j bodů $S(t_{ijk})_{ijk}$, která odpovídá novému vláknu, které dle pořadí indexu j sestavíme. Příklad konstrukce vlákna $i = 0, k = 2$ je na obrázku 5.3.

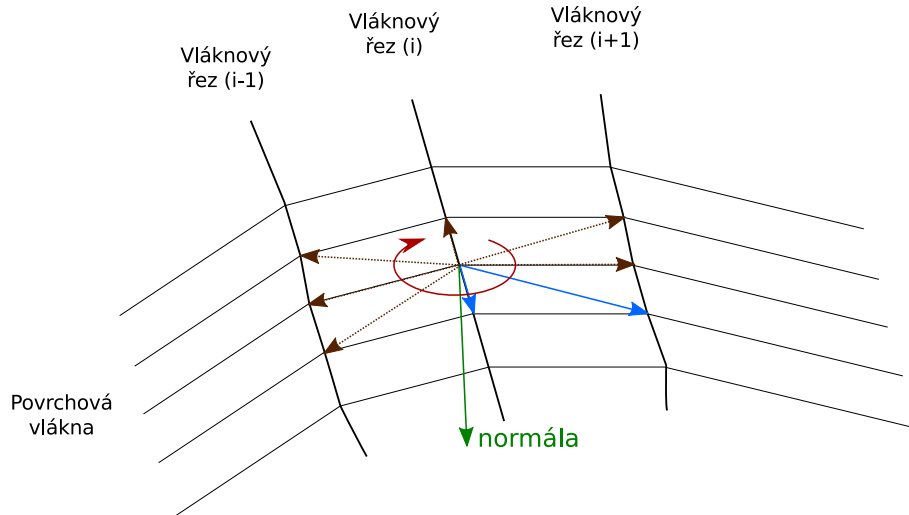


Obrázek 5.3: Příklad konstrukce vlákna $i = 0, k = 2$ na povrchu svalu. Červeně jsou označeny vláknové řezy j_0 až j_4 a myšlené příčné spline interpolační křivky, modře nově vytvořené vlákno pro $k = 2$. Na tomto náčrtu je R_p přibližně rovno pěti

5.4 Výpočet normál

Ve vrcholech vláken na svrchním povrchu svalu nyní provedeme vyhledání povrchových normál. Jelikož je v povrchových vláknech a jejich vláknových řezech implicitně zavedeno řazení dle indexů, lze definovat mřížku, na které je "horizontální" rozměr dán indexy vláknových řezů a "svislý" rozměr dán pořadím vláken. Na této mřížce je pak možné pro každý (středový) vrchol jednoduše nalézt nejvýše osm bezprostředně sousedních okolních vrcholů. Normála každého středového vrcholu je pak nalezena jako součet vektorů, které jsou dány jako vektorový součin dvou vektorů vedoucích od středového vrcholu k sousedům. Tyto dva vektory jsou sousední ve smyslu rotace okolo středového vrcholu po směru

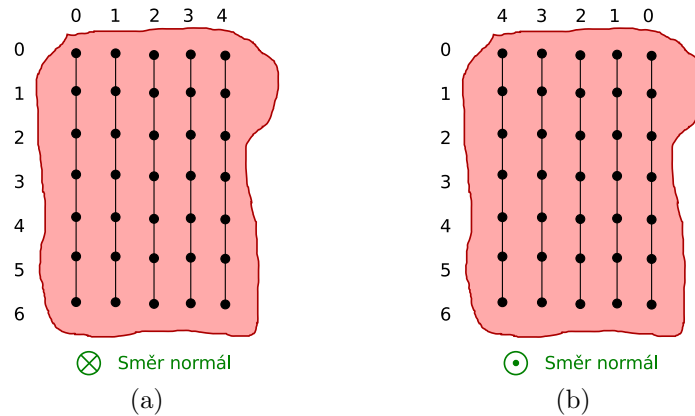
hodinových ručiček (náčrtek je na obrázku 5.4). Při přímé indexaci vláken toto zaručuje, že normála bude mířit dovnitř svalu. Normála je po nalezení normalizována (její složky jsou vyděleny délkou této normály), jelikož nám jde pouze o její směr.



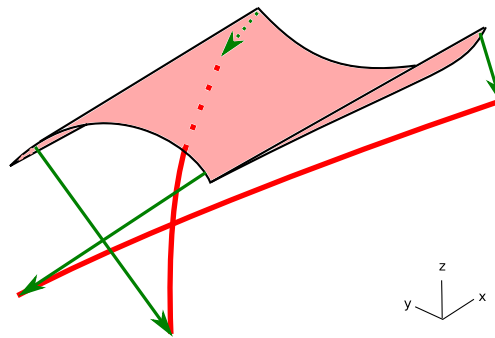
Obrázek 5.4: Výpočet normály na povrchu svalu. Šipky označují vektory v okolí středového bodu použité pro výpočet, modré šipky značí příklad výběru dvou vektorů pro vektorový součin, po směru červené čipky

Přímá indexace vláken je načrtnuta na obrázku 5.5a. Náčrtek je z pohledu na svrchní vrstvu svalu a na ní ležící povrchová vlákna. Můžeme si povšimnout, že při použití popsaného výpočtu normál budou normály mířit do objemu svalu. Může však nastat i situace, kdy bude pořadí vláken obráceno (obrázek 5.5b). Tato situace může nastat omylem při získávání dat, nebo záměrně, kdy řídicí vlákna nereprezentují svrchní, ale spodní povrch svalu. V tomto případě je normála (a tedy směr dekompozice svalu) také obrácena. Tuto situaci však je možné detekovat a adresovat, vizte prosím sekci 5.5.2.

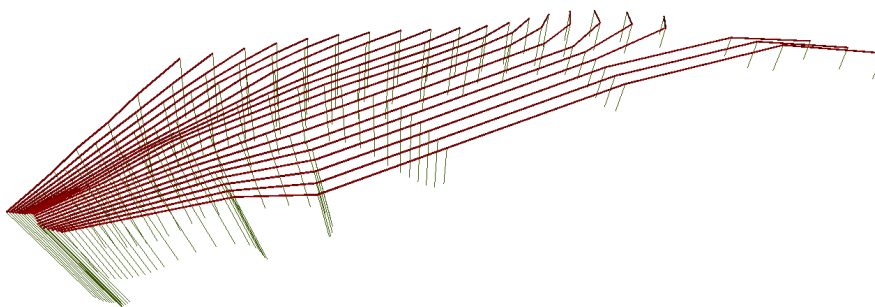
Pokud bychom však takto získané normály použili pro dekompozici objemu svalu, došlo by ke křížení vláken (obrázek 5.6), jelikož povrch svalu zpravidla obsahuje konvexní i konkávní části. Pro každý vláknový řez tedy vypočtené normály sečteme a výsledný vektor normalizujeme a dosadíme do všech vrcholů jako novou normálu (normály tedy zprůměrujeme). Tím docílíme paralelních směrů vzorkování a směrů dekompozice objemu svalu, při kterém ke křížení vláken nedochází. Reálná ukázka těchto normál (s vizualizací délky vzorku) je na obrázku 5.7.



Obrázek 5.5: Ilustrace indexace vláken (a) Přímý způsob indexace (b) Obrácený způsob indexace. Červeně je načrtnut sval z pohledu na jeho svrchní vrstvu. Vlákna leží na svrchním povrchu svalu.



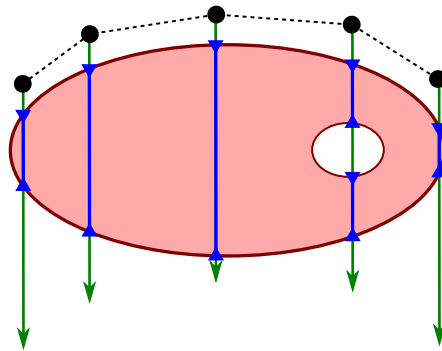
Obrázek 5.6: Křížení vláken při přímém použití normál. Zeleně jsou vyznačeny normály povrchu, červeně vlákna. Díky konvexnímu i konkávnímu tvaru povrchu svalu se vlákna kříží.



Obrázek 5.7: Ukázka normál a vzorkování objemu svalu. Zelené úsečky reprezentují normály v každém vrcholu povrchových vláken, jejich délka odpovídá délce vzorku objemu svalu.

5.5 Vzorkování objemu svalu

Objem svalu je vzorkován jako délka úsečky daná průsečíky paprsku s povrchovým modelem svalu. Paprsky jsou konstruovány s počátkem v daném vrcholu povrchového vlákna a se směrem normály v tomto vrcholu. Hloubka vzorku pak odpovídá vzdálenosti mezi nalezenými průsečíky. V případě nekonvexnosti či otvoru v modelu svalu může být nalezeno více průsečíků a tedy úseků paprsku, které jsou uvnitř svalu. Délky těchto úseků sčítáme, a tento součet označíme jako hloubku svalu. Ilustrace vzorkování na jednom vláknovém řezu je na obrázku 5.8, vizualizace těchto vzorků je také na obrázku 5.7.



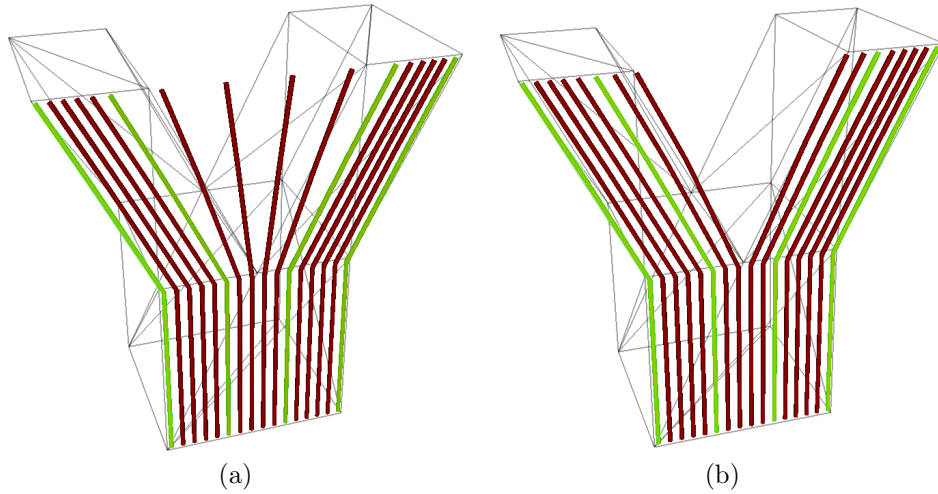
Obrázek 5.8: Vzorkování objemu svalu. Modré úseky zelených normál značí délku vzorku.

5.5.1 Štěpení svalu

Může však nastat situace, kdy je sval rozštěpen na dvě části. Na modelové ukázce svalu ve tvaru písmene Y na obrázku 5.9a můžeme pozorovat, že postup vytváření nových vláken na povrchu svalu popsany v sekci 5.3 vytváří zde čtyři červená vlákna mezi řídicími vlákny, která jsou zvýrazněna zeleně, v chybném pravidelném rozmístění v prostoru, kde je sval rozštěpen.

Je však možné detekovat, které vrcholy vláken jsou nad oblastí rozštěpu pouhým prahováním hodnoty vzorku. Je-li vzorek hloubky v nějakém vrcholu kratší než nastavená minimální hloubka (dána konstantou, experimentálně vhodnou hodnotou je 1.0), předpokládáme, že je vrchol nad oblastí, kde je sval rozštěpen. Tento vrchol pro další potřeby označíme jako vrchol V . Nejprve tedy nalezneme na vláknovém řezu, jemuž vrchol V náleží, oba dva vrcholy náležící řídicím vláknům, ve kterých je hloubka svalu dostatečná⁷, tedy řídicí vrchol V_p předcházející (na řezu) vrcholu V a řídicí vrchol V_n následující (na řezu) vrcholu V (náčrtek je na

⁷Pokud zde není hloubka korektní, dochází k chybě. Pokud takové vlákno neexistuje, také dochází k chybě. Uživatel je v obou případech informován, že je detekována chyba v řídicích datech - řídicí data neleží na povrchu svalu.



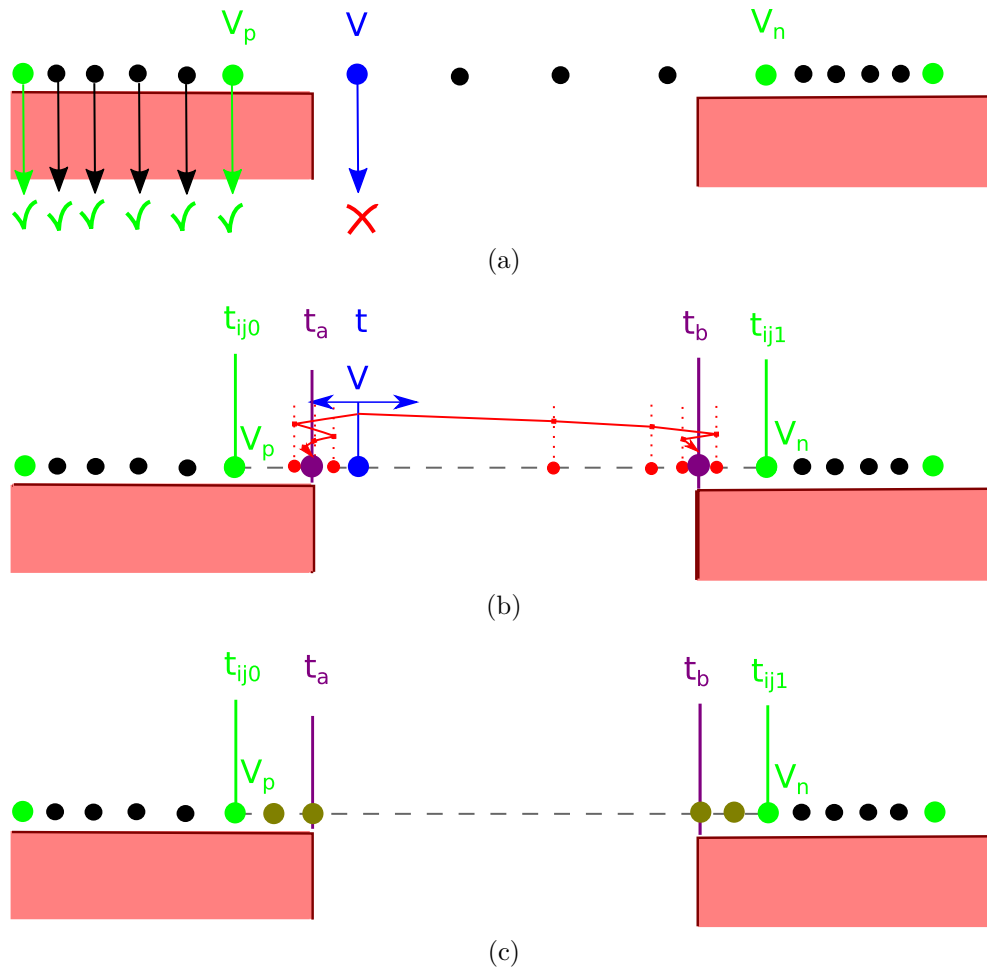
Obrázek 5.9: Štěpení modelového svalu. (a) Pouze metoda popsaná v sekci 5.3 (b) Metoda s detekcí a opravou vláken při štěpení. Zeleně zvýrazněná jsou řídicí vlákna.

obrázku 5.10a). Máme tedy nalezen segment řezu, který je nad oblastí rozštěpu svalu.

Nyní je tedy potřeba vlákna rozdělit do dvou skupin, a přeorganizovat je tak, aby kopírovala tvar rozštěpu. Z vrcholu V tedy provedeme dvojí binární hledání postupně směrem k vrcholům V_p a V_n (obr. 5.10b). Postupujeme po křivce (resp. právě po nalezeném segmentu) tohoto vláknového řezu půlením intervalů $\langle t_{ij0}, t \rangle$ a $\langle t; t_{ij1} \rangle$ a hledáme takový vrchol, ve kterém hloubka svalu dosahuje přijatelné hodnoty (požadujeme alespoň 80% hloubky ve vrcholu V_p resp. V_n , experimentálně tato hodnota vykazuje dobré výsledky). Tímto nalezneme hodnoty parametru pro vrcholy na okrajích rozštěpu svalu, označíme je jako t_a (na straně vrcholu V_p) a t_b (na straně vrcholu V_n). Počet iterací algoritmu půlení intervalu je omezen, 20 iterací se experimentálně ukázalo jako vhodný počet, který zajišťuje přesnost na přibližně 10^{-6} délky zpracovávaného segmentu.

Vlákna v segmentu rozštěpu rozdělíme na dvě poloviny. Bylo by vhodné samozřejmě detekovat, zda některé vlákna segmentu již neleží nad povrchem svalu, tím však ale může nastat situace, kdy je prostřední vlákno na některých řezech přiřazeno ke straně vrcholu V_p a na jiných řezech ke straně vrcholu V_n (jelikož je zpracování rozštěpu lokální vzhledem k řezu). Rozdělením vláken do dvou polovin, vzhledem ke konstantnímu počtu vláken v každém segmentu, tento problém nenastává.

Vrcholy vláken v každé z těchto polovin nyní převzorkujeme podobně jako v sekci 5.3, s tím rozdílem, že použijeme jiné hodnoty parametrů t_{ij0} a t_{ij1} , tedy jiné konce segmentu. Na straně vrcholu V_p počáteční hodnotu t_{ij0} ponecháme, a jako koncovou hodnotu t_{ij1} použijeme hodnotu t_a . Na straně vrcholu V_n bude počáteční t_{ij0} rovno t_b , a hodnotu t_{ij1} ponecháme. Poté na obou stranách vygenerujeme nové pozice vrcholů, které v pořadí opět daném indexací na řezu a seřazením dle



Obrázek 5.10: Postup metody štěpení svalu, pohled na řez svalu. (a) Vzorkování objemu selhává na prvním vláknu segmentu. (b) Binární hledání vrcholů na okrajích rozštěpu. (c) Generování nových vrcholů a rozštěpení vláken.

parametru t přiřadíme vrcholům mezilehlých vláken. Tím tato vlákna efektivně rozštěpíme (obr. 5.10c, výsledek pro modelový sval je na obr. 5.9b).

Z předchozího textu je patrné, že je vhodné, aby vláknové řezy odpovídaly i z anatomicky nekorektně nespojitým PCS řežům svalu, tedy tak, aby při dekompozici byl pro rozštěp dostatek místa pro ohnutí vláken. Distribuce vláken na povrchu a v objemu svalu je pak rovnoměrná a nedochází k zhuštění vláken v oblasti řezu.

5.5.2 Test korektnosti směru normál

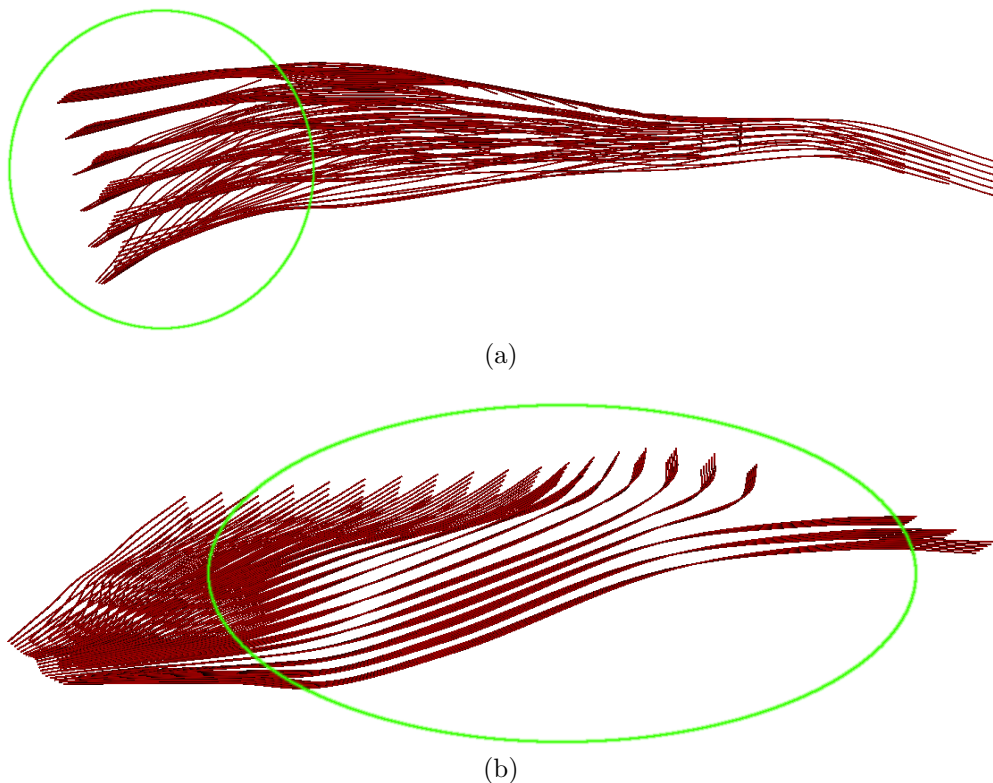
Po provedení předchozí sekce by již měla všechna povrchová vlákna skutečně ležet na povrchu svalu, za předpokladu, že není v řídicích datech chyba. Můžeme tedy otestovat, zda normály vypočtené v sekci 5.4 mají korektní směr do objemu

svalu. Vybereme tedy v povrchových vláknech náhodně několik vrcholů, a po směru normály ve vzdálenosti poloviny vzorku vytvoříme testovací bod, pro který určíme, zda je skutečně uvnitř povrchového modelu svalu. Pokud tento test selže pro většinu testů, předpokládáme, že jsou normály opačně, a všechny normály vynásobíme konstantou -1 . Toto chování však lze zakázat vstupním parametrem (implementováno jako `bool DoFlipNormals`).

5.6 Dekompozice objemu svalu

5.6.1 Generování vrstev v objemu svalu

Konečně máme připravena všechna data pro vytvoření vláken v objemu svalu. Postupujeme ve směru normál a generujeme vlákna po vrstvách, jejichž počet odpovídá specifikovanému vstupnímu parametru rozlišení vláken v objemu svalu. Vytváříme tak abstraktní trojrozměrnou mřížku, kde první směr je dán křivkami vláknových řezů, druhý směr vlákny samotnými, a třetí směr je dán směrem normál. Takovýto způsob generování však, díky paralelnosti normál a pravidelnosti umístění vrstev v objemu svalu, vytváří vizuálně pozorovatelný efekt který zveme "hrábě". Tento efekt lze pozorovat na obr. 5.11



Obrázek 5.11: Ukázka efektu "hrábě" způsobeného (a) pravidelností vrstev vláken (b) paralelností normál

Jelikož tento efekt není vizuálně příliš příjemný, nabízí se možnost vlákna pozměnit tak, aby byl efekt potlačen.

Jednotlivé vrstvy vláken jsou generovány vlákno po vláknu. Označíme-li si vlákno na svrchním povrchu svalu jako F_i , pak pozici jeho j -tého vrcholu označíme jako $F_i[j]$. Generujeme-li vrcholy vlákna G_i na vrstvě v , potom pozice vrcholu

$$krok = \frac{1}{R_v}$$

$$G_i[j] = F_i[j] + n_{ij} * v * krok * d_{ij} \quad \forall i, j$$

kde n_{ij} je normála ve vrcholu $F_i[j]$, R_v značí parametr rozlišení vláken objemu svalu a d_{ij} je vzorek hloubky svalu ve vrcholu $F_i[j]$. Pospojování vlákna G_i do lomené křivky odpovídá pospojování vlákna F_i .

Vertikální efekt "hrábě" můžeme odstranit tak, že proměnné $krok$ definujeme rozptyl, tedy generování upravíme jako

$$krok = \frac{1}{R_v}$$

$$posun = -krok/2 + rand(krok)$$

$$G_i[j] = F_i[j] + n_{ij} * v * krok * d_{ij} + posun \quad \forall i, j$$

Funkce $rand(a)$ generuje pseudonáhodné číslo v intervalu $\langle 0, a \rangle$. Hodnotu proměnné $posun$ vygenerujeme vždy pro každé vlákno, čímž celé vlákno vertikálně posuneme o náhodnou hodnotu z intervalu $\langle -\frac{krok}{2}, \frac{krok}{2} \rangle$, čímž zajistíme, že se vlákna vrstev neprotnou, a že do vlákna nezavedeme nechtěné vlnění.

Horizontální efekt "hrábě" odstraníme jednodušším přístupem. Pro všechny vrcholy $G_i[j]$ získáme vektory $\vec{V}_j = G_{i+1}[j] - G_i[j]$, samozřejmě za podmínky, že $i < (n - 1)$, a jejich pozici upravíme jako

$$posun = rand(1)$$

$$G_i[j] = G_i[j] + V_j * posun$$

Opět hodnotu proměnné $posun$ vygenerujeme nově pro každé vlákno vrstvy.

Tento přístup má však zjevnou nevýhodu, že shrnuje vlákna na stranu svalu. To však není díky náhodnosti a postranním vrstvám vláken (vizte prosím níže) příliš pozorovatelné, a výpočetní a implementační složitost je mnohem nižší, jelikož na vrstvách již nemáme dostupná data křivek vláknových řezů, které bychom ideálně potřebovali pro posun vláken. Toto posunování vláken také aplikujeme pouze na vnitřní vrstvy (tedy ne na svrchní a spodní vrstvu) abychom neporušili vzhled povrchových vláken.

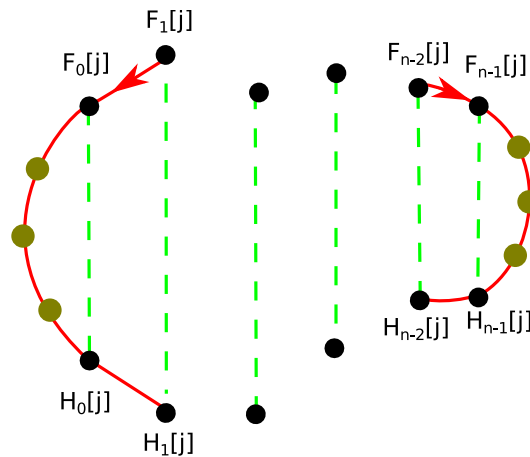
5.6.2 Postranní vlákna

Svislá rekonstrukce vláken může způsobit viditelnou plochost ve vláknech na stranách svalu. Sval tedy zaoblíme tak, že vytvoříme dodatečné svislé postranní vrstvy vláken. Tato vlákna generujeme po křivkách daných krajními vlákny na svrchní a spodní vrstvě svalu. Pokud tedy máme n vláken, $F_i[j]$ stále značí vrcholy svrchních vláken a $H_i[j]$ označíme vrcholy spodních vláken, křivky pro postranní vlákna definujeme po řezech čtyřmi vrcholy, tedy v pořadí

$F_1[j], F_0[j], H_0[j], H_1[j]$ pro křivky na levé straně svalu

$F_{n-2}[j], F_{n-1}[j], H_{n-1}[j], H_{n-2}[j]$ pro křivky na pravé straně svalu

Nad těmito vrcholy provedeme na každém řezu j spline interpolaci, v prostředním segmentu křivky vygenerujeme již známým způsobem nová postranní vlákna, jejichž počet odpovídá parametru objemového rozlišení. Náčrtek postupu generování postranních vláken je na obrázku 5.12



Obrázek 5.12: Generování postranních vrstev vláken. Červeně jsou načrtnuty křivky stran, žlutě jsou označeny vrcholy postranních vláken na tomto řezu

5.7 Upnutí vláken k oblastem úponu

Při vytváření každého vlákna je nutné koncové vrcholy vláken zařadit do seznamů otevřených konců, jeden pro počátek svalu a druhý pro konec svalu. Po dokončení dekompozice je pak nutné všechna vlákna přiřadit ke vhodným bodům v oblastech úponu svalu, vygenerovaných postupem v sekci 5.2. Postup pro obě strany je stejný, je však nutné strany oddělit, aby nedocházelo k upnutí vláken k nesprávné (opačné) straně svalu.

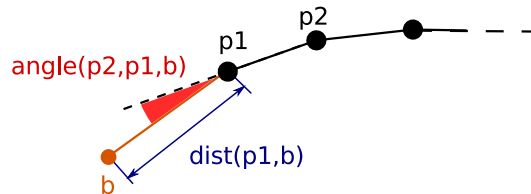
Vstupem tohoto problému jsou tedy dva seznamy, seznam bodů úponu a seznam otevřených vrcholů vláken. Hledáme vhodné párování v těchto seznamech

tak, aby byl výsledek optimální vzhledem k penalizaci za upnutí vlákna k nevhodnému úponu. Tato optimalizační úloha se nazývá "Problém přiřazení", a existuje na ní již známý Maďarský algoritmus [Kuh05].

Jako vhodná metrika se ukázala metrika

$$d = \text{dist}(p1, b) * \text{angle}(p2, p1, b)$$

kde $\text{dist}(p1, b)$ značí vzdálenost mezi koncovým bodem vlákna $p1$ a bodem úponu b a $\text{angle}(p2, p1, b)$ značí úhel mezi vektory $p2 \rightarrow p1$ a $p1 \rightarrow b$, kde $p2$ je vnitřní vrchol vlákna po směru vlákna bezprostředně předcházející koncovému vrcholu $p1$ a b je bod úponu. Tato metrika tak bere v potaz vzdálenost, na jakou se musí vlákno upnout (tedy minimalizuje se křížení způsobené upínáním k vzdáleným bodům úponu) a ohnutí vlákna po úponu (které způsobuje nepřirozené hrany či smyčky na koncích vláken). Metrika je ilustrována na obrázku 5.13.



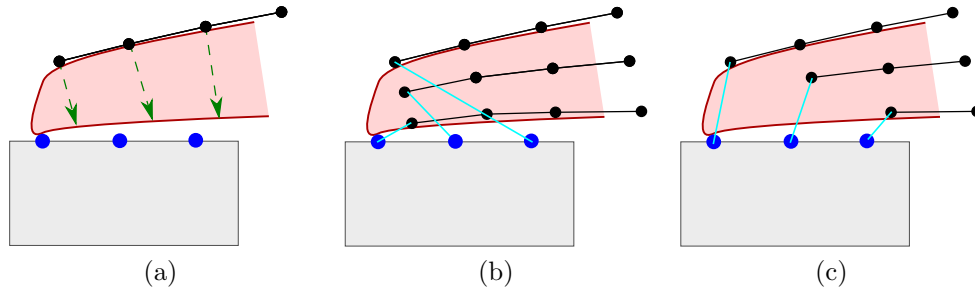
Obrázek 5.13: Ilustrace metriky. Modře je označena vzdálenost $\text{dist}(p1, b)$, červeně je označen úhel $\text{angle}(p2, p1, b)$

I přes tuto metriku však může pro svaly s velkým pennate úhlem nastat situace, že není možné najít vhodné přiřazení, a tedy kvalita dekompozice strmě klesá. U těchto svalů jsou totiž vlákna na svrchním povrchu svalu typicky delší, než na spodním povrchu. Data vláken spodního povrchu však nemáme k dispozici, a tak nastává situace ilustrovaná na obr. 5.14b. Vzhledem ke směru normál jsou vlákna dekomponována tak, že jejich upnutí možné bez viditelných zlomení vláken. My bychom však uvítali výsledek na obrázku 5.14c, tedy zkrácení vláken na spodním povrchu svalů a adekvátní úprava vláken v objemu svalů, aby připojení bylo korektní v celém objemu svalů.

Pro řešení tohoto problému znovu využijeme vrstvy vláken vytvářené při dekompozici objemu svalů. Po prvním upnutí vláken "na zkoušku" zkontrolujeme, zda úhel $\text{angle}(p2, p1, b)$ z předešlé metriky pro žádné z vláken nepřesáhl uživatelem povolenou mez. Pokud tato situace nastane, iterativně a podle hloubky vrstvy vláken vlákna zkracujeme a upínání opakujeme, dokud test ohybu vláken není úspěšně splněn.

Zkracování ale neprovedeme odstraňováním vrcholů řídicích vláken, jelikož je to nepraktické a řídicí vlákna příliš velký počet vrcholů k odříznutí nemají, použijeme přesnější zkracovací postup pro oříznutí konců vláken⁸. Průchodem

⁸Postup pro počátek vláken je ekvivalentní, pouze se změnil směr měření a indexace



Obrázek 5.14: Ilustrace problému pennate svalů (v řezu) (a) dostupná data (b) dekompozice a upnutí bez detekce lomu vláken (c) požadovaný výsledek. Modře jsou označeny body úponu na šedé kosti.

přes každé vlákno od počátku do konce změříme jeho délku d_v jako součet délek všech jejích segmentů. Mezivýsledky v každém vrcholu ukládáme jako hodnoty d_i . V závislosti na hloubce vlákna ve svalu, která odpovídá vrstvě vláken při dekompozici, vypočteme koeficient zkrácení vlákna. Hloubka vrstvy v objemu svalu je při dekompozici dána číslem l z intervalu $(0, 1)$, kde hodnota $l = 0$ udává svrchní vrstvu vláken a hodnota $l = 1$ udává spodní vrstvu vláken. Koeficient zkrácení tak vypočteme jako

$$1 - \text{max_zkraceni} * l$$

Novou délku vlákna d_n tedy vypočteme jako

$$d_n = d_v * (1 - \text{max_zkraceni} * l)$$

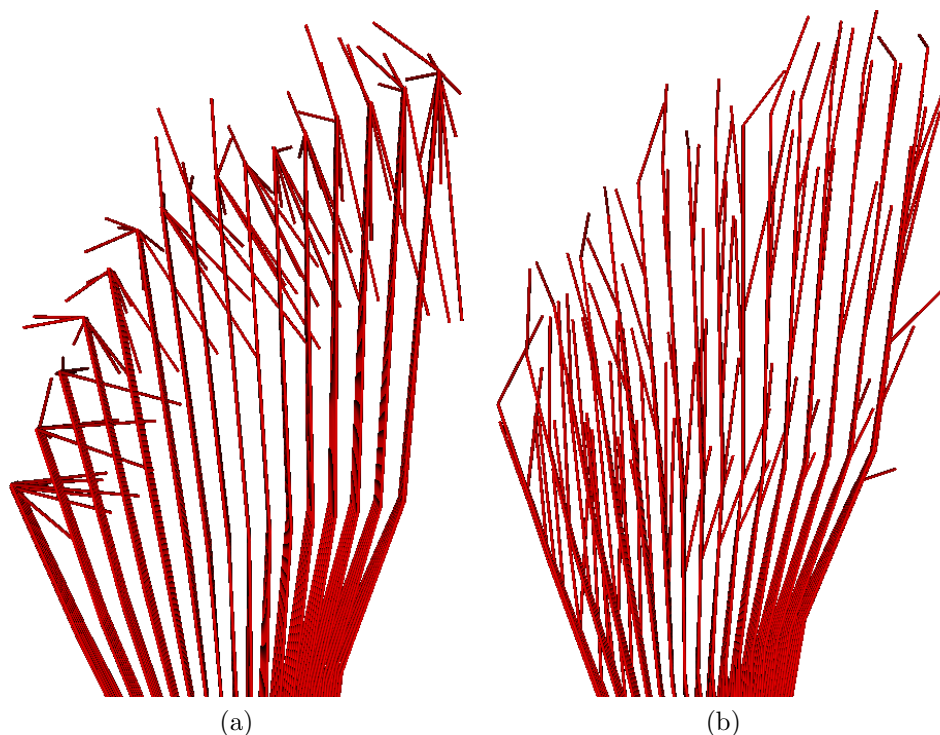
Znovu postupujeme po vlákně, a odřezáváme od konce segmenty, pro něž je hodnota d_i v obou krajních vrcholech větší, než hodnota d_n . Tím způsobíme, že poslední segment takto oříznutého vlákna, tedy segment mezi vrcholy P_{n-1} a P_n , je ten, na kterém lze hledat nový koncový vrchol vlákna. Novou pozici koncového vrcholu P_n nyní již snadno vypočteme jako

$$k = \frac{d_n - d_i(P_{n-1})}{d_i(P_n) - d_i(P_{n-1})}$$

$$P_n = P_{n-1} + k * (P_n - P_{n-1})$$

pro všechny tři osy prostoru. Tímto je vlákno oříznuto v závislosti na jeho hloubce v objemu svalu. Po zkrácení všech vláken opakujeme upínání, znovu testujeme a případně postup opakujeme.

Může však nastat i situace, že i tento postup selže, zejména při velmi restriktivním nastavením mezního úhlu zlomení vlákna. V tomto případě iterování po určitém počtu iterací zastavíme a informujeme uživatele. Koeficient *max_zkraceni* je vhodné nastavit na 0.1, tedy při jedné iteraci bude vlákno zkráceno maximálně o 10%. Ukázka aplikace zkracování vláken je na obrázku 5.15 (bez vyhlazení a posunu vláken, pro zvýraznění problému ohýbání vláken).



Obrázek 5.15: Ukázka upnutí vláken svalu *Iliacus*. (a) původní metoda, vlákna mají nežádoucí zpětný ohyb (b) po detekci a opravě ohybu vláken (viditelné na stromovitém tvaru vláken), sval se také upíná k větší ploše.

5.8 Vyhlazení vláken a dokončení

Po upnutí vláken k oblastem úponu je posledním krokem dekompozice podélné vyhlazení vláken. To probíhá opět pomocí spline interpolace, nyní ovšem po směru vlákna. Parametr R_f , tedy rozlišení vláken, určuje počet nových vrcholů, které jsou v segmentech interpolačních křivek vláken vygenerovány pro jejich vyhlazení.

Data vláken jsou poté předána na výstup jako množina lomených křivek, kdy každá křivka reprezentuje jedno dekomponované vlákno. Počet vláken na výstupu je $R_p * R_v + 2(R_v)$ (člen $2(R_v)$ započítává postranní vrstvy vláken).

6 Implementace

6.1 Interpolační metody

Chování v předchozích sekcích popsané dekompoziční metody a kvalita jejích výsledků přímo závisí na výběru metody spline interpolace. Proto jsme implementovali několik metod spline interpolací, a porovnali jejich chování a použitelnost pro dekompozici. Pozorovali jsme úroveň oscilace interpolační křivky, která je v napnutých vláknech nerealistická, její vizuální hladkost a výpočetní náročnost, která vzhledem k počtu potřebných interpolací není zcela zanedbatelná.

6.1.1 Kubická spline interpolace

První implementovanou metodou je klasický parametrický kubický spline s využitím chordálové parametrizace (tedy parametrizace pomocí délky řídicího segmentu) [Jež10]. Tato metoda je dobře známá, ale rozebereme si ji, jelikož některé části výpočtů použijeme v popisu dalších metod. Parametrické rovnice interpolační křivky segmentu křivky jsou popsány jako:

$$s_i(t) = a_i \left(\frac{t - t_i}{t_{i+1} - t_i} \right)^3 + b_i \left(\frac{t - t_i}{t_{i+1} - t_i} \right)^2 + c_i \left(\frac{t - t_i}{t_{i+1} - t_i} \right) + d_i \quad (1)$$

kde $0 \leq i < n$ a n je celkový počet segmentů (= počtu vrcholů lomené křivky mínus jedna). Pro všechny segmenty tedy získáme $4n$ neznámých proměnných. Pro nalezení jejich hodnot tedy potřebujeme $4n$ podmínek. Podmíníme tedy

$$\begin{aligned} s_i(t_i) &= x_i, & 0 \leq i < n \\ s_{n-1}(t_n) &= x_n \end{aligned} \quad (2)$$

kde $x(i)$ je jednorozměrná pozice vrcholu i (ve třech dimenzích provádíme výpočet třikrát, separovaně, pro stejnou parametrizaci) a $t(i)$ je hodnota parametru ve vrcholu i . Podmíníme také

$$s_{i-1}(t_i) = s_i(t_i) \quad (\text{spojitost křivky}) \quad (3)$$

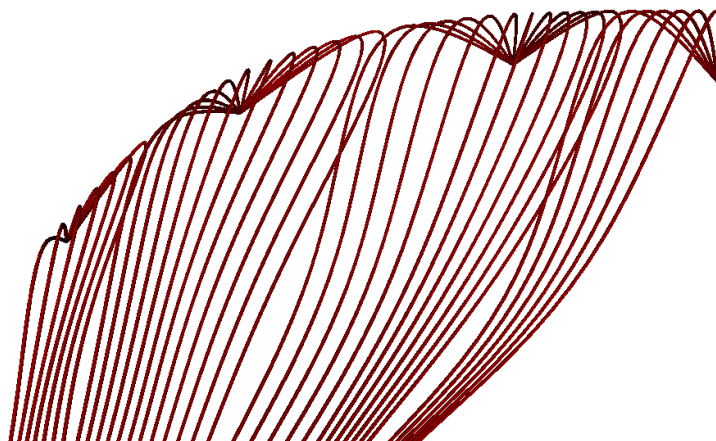
$$s'_{i-1}(t_i) = s'_i(t_i) \quad (\text{spojitost první derivace}) \quad (4)$$

$$s''_{i-1}(t_i) = s''_i(t_i) \quad (\text{spojitost druhé derivace}) \quad (5)$$

kde $1 \leq i < n$, neboť spojitost je definovatelná pouze na vnitřních uzlech křivky. Rovnice (2), (3), (4) and (5) generují $(n + 1) + 3(n - 1) = 4n - 2$ podmínek. Doplníme tedy dvě další okrajové podmínky. Implementujeme tři typy okrajových podmínek (v prvním a posledním vrcholu lomené křivky):

- Přírozený spline - druhé derivace v koncových vrcholech jsou rovny $s_0''(t_0) = 0$ a $s_{n-1}''(t_n) = 0$. Povšimněte si prosím, že máme $n+1$ vrcholů indexovaných $0..n$.
- Spline s parabolickým doběhem - druhé derivace v koncových bodech jsou nastaveny na $s_0''(t_0) = s_1''(t_1)$ a $s_{n-1}''(t_n) = s_{n-2}''(t_{n-1})$
- Spline s kubickým doběhem - druhé derivace v koncových bodech jsou nastaveny na $s_0''(t_0) = 2s_1''(t_1) - s_2''(t_2)$ a $s_{n-1}''(t_n) = 2s_{n-2}''(t_{n-1}) - s_{n-3}''(t_{n-2})$

S těmito dodatečnými podmínkami již můžeme sestavit a s jednoznačným řešením vyřešit soustavu rovnic pro proměnné a_i, b_i, c_i, d_i a vrcholy lomené křivky můžeme interpolovat pomocí předpisu (1). Metoda je známá pro velmi hladký průběh interpolace jelikož metoda zaručuje spojitost až do druhé derivace interpolační křivky. Můžeme ale pozorovat, zejména na koncích některých vláken, že výsledná křivka této metody může díky vysoké hladkosti velmi oscilovat, což může vést až k nežádoucímu protínání vláken (příklad je na obrázku 6.1). Pro spliny s parabolickým a kubickým doběhem dochází na konci splinu k z definice vyšší oscilaci (způsobené větším obloukem při koncích těchto křivek).



Obrázek 6.1: Protínání vláken způsobené oscilací kubické přírozené spline křivky způsobené její vysokou hladkostí.

Berme také v potaz, že by interpolační křivky měly průběhem odpovídat průběhu svalového vlákna v lidském těle, které jsou zpravidla napnuté. Křivky kubického splinu jsou naopak relaxované. Pro další experimenty tedy použijeme metody, jejichž křivky obětují hladkost za menší oscilaci a více "napnutý" tvar.

6.1.2 Omezený kubický spline

Omezený kubický spline (*Constrained cubic spline*) [Kru01] je prvním adeptem na vhodnou interpolační metodu pro dekompozici, která nezachovává spojitost druhé

derivace. Formálně jde o "spline s konečnou diferencí", tedy spline, pro který je první derivace ve vrcholech lomené křivky aproximována pomocí tři vrcholů okolo každého vrcholu (včetně tohoto vrcholu).

Omezený kubický spline vychází z rovnice (1) a využívá podmínek (2), (3) and (4) ze sekce 6.1.1. První derivace (tangenty) jsou ale aproximovány jako

$$\begin{aligned} s'_i(t_i) &= \frac{2}{\frac{x_{i+1} - x_i}{t_{i+1} - t_i} + \frac{x_i - x_{i-1}}{t_i - t_{i-1}}} \\ &= 0 \text{ if } \text{sign} \left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i} \right) \neq \text{sign} \left(\frac{x_i - x_{i-1}}{t_i - t_{i-1}} \right) \end{aligned} \quad (6)$$

pro $0 < i < n$. Tyto rovnice jsou aplikovatelné pouze na vnitřní body křivky, pro koncové body tedy doplníme

$$s'_0(t_0) = \frac{3(x_1 - x_0)}{2(t_1 - t_0)} - \frac{s'_1(t_1)}{2} \quad (7)$$

$$s'_{n-1}(t_n) = \frac{3(x_n - x_{n-1})}{2(t_n - t_{n-1})} - \frac{s'_{n-2}(t_{n-1})}{2} \quad (8)$$

Jelikož aproximujeme první derivace, druhé derivace a tedy i parametry a_i, b_i, c_i a d_i mohou být přímo získány bez nutnosti řešit soustavu rovnic z následujících vztahů:

$$s''_i(t_i) = -\frac{2[s'_i(t_{i+1}) + 2s'_i(t_i)]}{(t_{i+1} - t_i)} + \frac{6(x_{i+1} - x_i)}{(t_{i+1} - t_i)^2} \quad (9)$$

$$s''_i(t_{i+1}) = \frac{2[2s'_i(t_{i+1}) + s'_i(t_i)]}{(t_{i+1} - t_i)} - \frac{6(x_{i+1} - x_i)}{(t_{i+1} - t_i)^2} \quad (10)$$

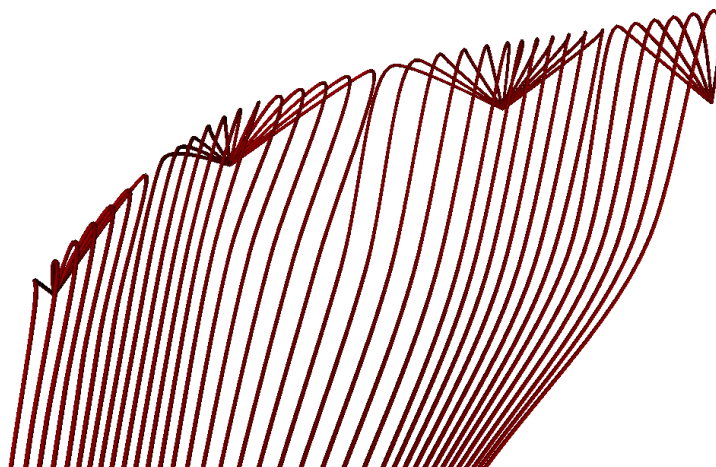
$$a_i = \frac{f''_i(t_{i+1}) - f''_i(t_i)}{6(t_{i+1} - t_i)} \quad (11)$$

$$b_i = \frac{t_{i+1}f''_i(t_{i+1}) - t_i f''_i(t_i)}{2(t_{i+1} - t_i)} \quad (12)$$

$$c_i = \frac{(x_{i+1} - x_i) - c_i(t_{i+1}^2 - t_i^2) - d_i(t_{i+1}^3 - t_i^3)}{(t_{i+1} - t_i)} \quad (13)$$

$$d_i = y_i - c_i x_i - b_i x_i^2 - a_i x_i^3 \quad (14)$$

Výsledný spline je pouze C^1 spojitý, nicméně jeho křivka osciluje méně, možnost vláken k protnutí je omezena a průběh vláken je více paralelní a napnutý. Nicméně, pokud řídicí lomená křivka obsahuje ostrou změnu směru, může v interpolační křivce dojít k velmi ostrému ohybu. Příklad výsledků této metody pro dekompozici je na obrázku 6.2.



Obrázek 6.2: Příklad interpolace pomocí omezeného splinu. Křížení vláken je v porovnání s klasickým kubickým splinem obtížnější, ohyb vláken je však příliš prudký

6.1.3 Catmull-Rom spline

Catmull-Rom spline [CR74] je také ve třídě C^1 spojitých splinů, které si zachovávají výhody omezeného kubického splinu popsaného v sekci 6.1.2, ale nevytvářejí ostré ohyby a svým tvarem se tak podobají tvaru C^2 spojitých křivek, do určité míry. Metoda Catmull-Rom je velmi podobná metodě omezeného splinu, s rozdílem, že výpočet tangenty z rovnice (6) přepíšeme pro uniformně rozprostřená data jako

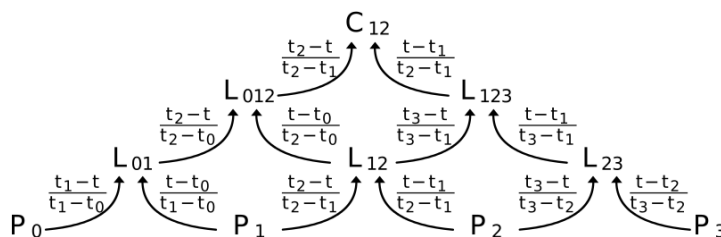
$$s_i(t_i) = \frac{x(t_{i+1}) - x(t_{i-1}))}{2} \quad (15)$$

neboli tangenta odpovídá spojnici předchozího a následujícího vrcholu lomené křivky (vzhledem k zpracovávanému vrcholu). Koncové vrcholy jsou zpracovány pomocí stejné rovnice, pro tyto výpočty však dočasně řídící lomenou křivku doplníme o dodatečné vrcholy, zrcadlové kopie druhého resp. předposledního vrcholu okolo prvního resp. posledního vrcholu (podobně jako pro přirozený spline).

Autoři článku [BG88] navrhli algoritmus pro konstrukci neuniformního Catmull-Rom splinu libovolného stupně, který je, pokud je nám známo, korektní. Tento algoritmus využívá rekursivního pyramidového schématu pro výpočet křivky. Příklad je na obr. 6.3 pro segment C_{12} mezi vrcholy P_1 a P_2 .

Aby toto schéma bylo použitelné pro naše účely (pro interpolační rutinu pracující s rovnicí (1)), provedli jsme statickou derekurzi pyramidy pro kubický Catmull-Rom spline a označili výsledné koeficienty proměnných t^3, t^2, t^1 resp. t^0 v této derekurzi jako parametry a_i, b_i, c_i resp. d_i . Tím opět není nutné řešit soustavu rovnic pro přímé získání interpolační Catmull-Rom křivky.

Autoři článku [YSK09] prozkoumali vliv parametrizace použité v tomto interpolačním schématu a ukázali, že centripetální parametrizace (parametrizace



Obrázek 6.3: Příklad rekursivního pyramidového výpočtu neuniformního Catmull-Rom splinu pro segment C_{12} (mezi vrcholy P_1 a P_2). Převzato z [YSK09].

odmocninou vzdálenosti vrcholů řídicí lomené křivky) vykazuje očekávané a korektní výsledky pro tuto třídu křivek, my tedy také implementujeme tedy tento typ parametrizace.

Existuje také velmi populární technika pro konstrukci neuniformního parametrického Catmull-Rom splinu jednodušeji za pomoci následujících tangent:

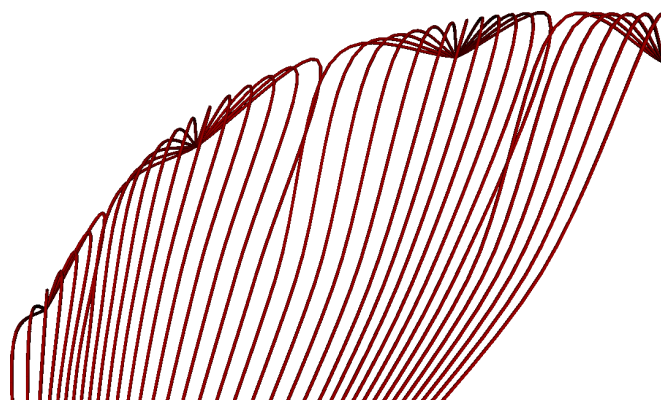
$$s_i(t_i) = \frac{x(t_{i+1}) - x(t_{i-1})}{t_{i+1} - t_{i-1}}, \quad (16)$$

nám ale není známo ověření, zda je toto zjednodušení korektní, a využíváme tedy derekurzi pyramidového vyhodnocování.

Na obrázku 6.4 je možné pozorovat výsledek Catmull-rom interpolace. K proutnutí vláken na těchto datech dochází, ale do menší míry než je tomu na obr. 6.1, ale vlákna jsou uspořádanější, přímější a vzhledově připomínají vlákna svalů nejvíce. Došli jsme tedy k závěru že i pro popsanou dekompoziční metodu bude třída Catmull-Rom neuniformních parametrických interpolačních spline křivek nejvhodnější⁹.

Na závěr je nutné poznamenat, že data pro ukázky interpolací v této sekci jsou velmi extrémní. V reálném svalu nedochází k upínání více vláken k jednomu bodu úponu, jako je tomu v těchto testovacích datech a náročnost na (ne)oscilaci interpolační metody je tedy pro reálnou dekompozici podstatně menší. Považovali jsme však za vhodné schopnosti metod ověřit i v krajním případě.

⁹Catmull-Rom křivky jsou také používány i pro jiné dekompoziční metody, jako např. ASM [KCMV12a, KCMV12b]

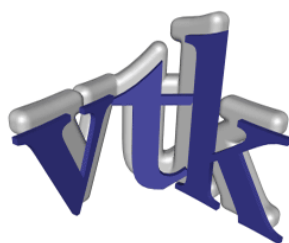


Obrázek 6.4: Příklad Catmull-Rom interpolace. Stále dochází k protínání vláken, ale do menší míry než je tomu u klasické spline metody v sekci Fig.6.1

6.2 Použité knihovny a frameworky

6.2.1 VTK a jeho pipeline

Balíkem knihoven *Visualisation Toolkit (VTK)* zprostředkovává mimo jiné knihovny pro reprezentaci, zpracování a vizualizaci dat prostřednictvím knihovny OpenGL. Systém je vyvinut v jazyce C++, je však přizpůsoben pro použití i v jiných jazycích (Java, Python, ...). Velmi dobrými zdroji pro pochopení práce s VTK jsou tutoriál [Bel04] a dokumentace [Kol03].



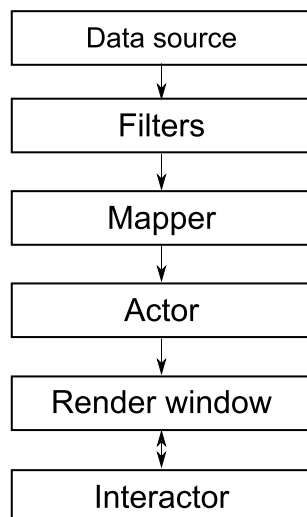
Obrázek 6.5: Logo VTK. Převzato z [Kol03]

VTK používá systém bloků, kaskádovitě seřazených do *potrubí - pipeline*. Tuto pipeline ilustruje obrázek 6.6.

Zdroj dat (*Data source*) poskytuje data ke zpracování. Nemusí však jít jen o data načtená ze souboru, VTK nabízí balík modulů pro získávání dat jako výsledků matematických funkcí, analyticky popsaných těles apod. Načtená či vygenerovaná data jsou poté mapována na primitiva v prostoru o zvolené dimenzi pomocí `vtkMapperu`. Namapované hodnoty jsou poté pomocí třídy `vtkActor` vizuálně reprezentovány (barvou, průhledností), třída také umožňuje specifikovat další parametry vizualizace, jako stínování, reprezentace pomocí drátového modelu apod. Posledním blokem je již `vtkRenderer`, který se stará o samotné

zobrazení v náhledovém okně. S daty je možno interagovat pomocí několika typů interactorů (myš, klávesnice, ...).

Pipeline funguje na principu požadavků, tedy bloky v kaskádě pozdější žádají o připravená data předchozí vrstvy, která opět žádá o data předchozí vrstvy, atd. Výpočty či načítání dat jsou tak na požádání a při případných změnách parametrů se provádějí jen nutné výpočty.



Obrázek 6.6: Pipeline systému VTK. Převzato a upraveno z [Bel04].

Tento balík knihoven nám nejen usnadní zobrazení výsledků naší práce, ale také definuje abstraktní třídu `vtkPolyDataSource`, kterou použijeme pro implementaci našeho kódu. Jakákoliv třída dědicí od této abstraktní třídy totiž může být zařazena na začátek *pipeline* a poskytnout nová data, tedy dekompozici svalu na vlákna. Přirozeně má i dekompoziční rutiny vstupy, ale sémantika dat se mění, a nejde tak o filtr `vtkPolyDataToPolyDataFilter`. Hlavní metodou této třídy je metoda `Execute()` která je spuštěna kdykoliv jsou potřebná data vyžadována následujícím blokem *pipeline*. Implementací této metody se budu věnovat v kapitole 6.3.4.

Knihovna je distribuována ve formě zdrojových kódů, je tedy ji nutné nejprve přeložit pomocí nástroje CMake. Datové soubory pro balík VTK jsou uloženy v interním souborovém formátu `.vtk`, popsaném v [Kol10]. Z důvodu kompatibility s frameworkem MAF2 používáme starší verzi VTK 4.0.2.

6.2.2 MAF2 a MAF3

Multimod Application Framework (MAF) [MAF13] verze 2 je framework pro rapidní komponentový vývoj aplikací s medicínským zaměřením. Framework je postaven na čtyřech základních komponentách, či rozhraních, které dovolují jednodušší vývoj aplikací.

Data uvnitř frameworku jsou uložena v kontejnerech *Virtual Medical Entity* (VME), které definují způsob uložení a reprezentaci až 4D dat. K dispozici jsou abstraktní datové typy i kontejnery pro uložení povrchových modelů dat, objemových dat či pouze bodových dat. Je tak možné definovat procedurální zdroje VME, například pro získávání řezů objemových dat (slicing). Data jsou uspořádána do stromové struktury, která umožňuje ukládat v čase proměnné reprezentace dat, matici závislosti na rodičovském prvku a textová metadata pro popis dat.

Prezenční vrstva *View* nabízí vytváření jednoduchých i kompozitních náhledových oken na data, je tedy možné specifikovat, která VME budou zobrazena v kterém náhledovém okně či jeho části a jak spolu náhledy souvisí. Vrstva také automaticky reaguje na změny ve VME datech, způsobem podobným pipeline VTK. Uživateli je také umožněn výběr dat (jednotlivě i skupinově). Při zobrazování a výběru dat je také respektována stromová struktura dat.

Nad vybranými daty lze ve vrstvě *Operations* spouštět operace, která tato data získají na svém vstupu, a je kontrolováno, zda operace daný typ dat podporuje či nikoliv. Podporovány jsou také operace pro import a export dat z jiných datových formátů. Operace jsou do systému začleněny pomocí Plug-in systému, kdy je operace zaregistrována a aplikaci jsou předány požadavky na vstupní a/nebo výstupní typy dat.

Pomocí vrstvy *Interactions* je pak možné se zobrazenými daty manipulovat pomocí mnoha typů vstupních zařízení. Je definována abstraktní vrstva pro až 6D manipulaci s trojrozměrnými daty, včetně haptických zařízení, mapování vstupních zařízení na akce uvnitř aplikace a implementace běžných zařízení (myš, klávesnice, trackball, pero, ...).

Framework nativně využívá knihoven VTK pro práci s daty, převod z dat ve formátu VME na formát používaný ve VTK je tedy velmi jednoduchý, stejně jako import dat ve formátu VTK. Jelikož je filtr pro dekompozici od počátku vyvíjen pod knihovnamí VTK, je také kompatibilní s frameworkem MAF2. Pod tímto frameworkem je také vyvinut software projektu VPHOP, lhpBuilder, který byl během vývoje velmi dobrým nástrojem pro zobrazení testovacích dat i výsledků dekompozice.

Původním cílem této práce však byla kompatibilita s prostředím MAF3, které je v současné době ve vývoji. MAF3 je však zpětně nekompatibilním pokračovatelem MAF2, se zcela novým jádrem a filosofií přístupu k vývoji aplikací a bohužel s nekonzistentní či dokonce neexistující dokumentací. Nevýhodou je také vyčlenění knihoven VTK pouze jako volitelného periferního rozšíření MAF3. Největším problémem však je, že framework MAF3 není zpětně kompatibilní a tedy nepodporuje formát dat použitý v prostředí MAF2, ve kterém bohužel máme všechna data uložena. Chybí také podpora grafického uživatelského rozhraní, která by byla pro dekompoziční metodu nutná pro výběr vstupních dat. Po konzultaci a výslovném svolení vedoucího práce jsem tedy bohužel tento cíl opustil. Implementace pod knihovnou VTK je ale nadále kompatibilní s prostředím MAF2 a potažmo softwarem projektu VPHOP.

6.2.3 libhungarian

Knihovna `libhungarian` [Sta04] je implementací Maďarského algoritmu pro řešení Problému přiřazení v jazyce C. Vstupem pro knihovnu je cenová matice, kde v našem případě řádky představují body úponu vláken a sloupce představují vlákna a hodnotami matice jsou výsledky výpočtů metriky ze sekce 5.7. Výstupem knihovny je přiřazovací matice, ve které číslo '1' označuje upnutí bodu úponu k vlákně.

V prvotní implementaci byla použita knihovna z webu <http://robotics.stanford.edu/~gerkey/tools/hungarian.html>, na které je současná verze knihovny založena, ta však byla velmi nestabilní a pravidelně u ní docházelo k zacyklení. Nepodporovala také matice typu $m * n$, což bylo pro vývojové účely a testování tohoto přístupu k upínání vláken nevhodné.

6.3 Implementace dekompoziční metody

Implementaci jsem provedl v jazyce C++ s využitím vývojového prostředí Microsoft Visual Studio 2010 a za použití výše uvedených knihoven. Samotná dekompoziční metoda je implementována jako třída `vtkMuscleDecomposer` která dědí od třídy `vtkPolyDataSource` balíku VTK. Pro testování a demonstrační účely jsem vytvořil testovací aplikaci `MuscleDecomposer.exe`.

6.3.1 vtkMuscleDecomposer

Třída `vtkPolyDataSource` vyžaduje na svém výstupu objekt typu `vtkPolydata` s výstupními `data`, a hlavní metodu `Execute()` která je kaskádou VTK spuštěna při požadavku na `data`. Vstupní `data` jsou třídě předána následujícími metodami:

- `AddFiberData(vtkPolyData* data)` - přidá na vstup třídy `data` jedné lomené křivky reprezentující řídící vlákno. Pořadí přidávání vláken odpovídá povrchovému seřazení vláken
- `AddAttachmentData(vtkPolyData* data, vtkPolyData* bone, int side)` - přidá na vstup třídy hranici úponové oblasti jako `data`, a povrchový model odpovídající kosti jako `bone`. Parametr `side` nabývá hodnot `ATTACHMENT_START = 0` pokud je oblast na počátku svalu, či `ATTACHMENT_END = 1` pokud je oblast na konci svalu
- `AddMeshData(vtkPolyData* data)` - přidá na vstup třídy `data` povrchového modelu svalu

Dále je možné specifikovat parametry dekompozice. Hodnoty uvedené v hranatých závorkách jsou výchozí hodnoty nastavené konstruktorem:

- `SetInterpolationSubdivision(int InterpolationSubdivision)` [10] - počet vytvořených segmentů v jednom segmentu vlákna při vyhlazování vlákna
- `SetSurfaceSubdivision(int SurfaceSubdivision)` [5] - počet vláken vytvořených mezi dvěma řídicími vlákny
- `SetVolumeSubdivision(int VolumeSubdivision)` [5] - počet vrstev vláken vytvořených v objemu svalu
- `SetInterpolationMethod(int InterpolationMethod)` [SPLINE_CATMULL_ROM] - nastaví interpolační metodu. Na výběr jsou následující interpolační metody:
 - `SPLINE_NATURAL` - Přirozený kubický spline
 - `SPLINE_PARABOLIC_RUNOUT` - Kubický spline s parabolickým doběhem
 - `SPLINE_CUBIC_RUNOUT` - Kubický spline s kubickým doběhem
 - `SPLINE_CONSTRAINED` - Omezený spline
 - `SPLINE_CATMULL_ROM` - Catmull-Rom spline
- `SetDoOffsetFibers(bool DoOffsetFibers)` [*true*] - Povolí či zakáže porušení pravidelnosti dekompozice posunováním vláken
- `SetDoFlipNormals(bool DoFlipNormals)` [*true*] - Povolí či zakáže automatickou detekci a korekci orientace normál
- `SetDoForceFlipNormals(bool DoForceFlipNormals)` [*false*] - Vynutí otočení orientace normál (pro modely svalů s opačnou orientací trojúhelníků)
- `SetPennationTreshold(double PennationTreshold)` [$\pi * 4/9$] - Nastaví maximální úhel zlomení vlákna při jeho upínání k oblasti úponu (v radiánech)

Filtr samotný může být manuálně spuštěn metodou `Update()`. Po provedení dekompozice je výsledek v objektu `vtkPolyData` dostupný metodou `GetOutput(0)`.

Následující text popisuje implementační detaily některých metod. Pro detailní popis implementace však prosím vizte zdrojový kód na přiloženém DVD.

6.3.2 AddFiberData()

Metoda do filtru přidá data nového vlákna. Ta jsou po spuštění dekompozice převedena metodou `InitFibers()` na interní reprezentaci. Všechny vrcholy jsou reprezentovány jako objekty třídy `Point` a jsou umístěny v jednom seznamu `PointCloud`. Základem každého objektu třídy `Point` jsou vektory pozice a normály, třída však obsahuje další pomocné proměnné pro interpolace, štěpení vláken,

data pro úpon vlákna apod. Každé vlákno je reprezentováno objektem třídy `Fiber`, která implementuje seznam `PointIDs` reprezentující pospojování vrcholů do vlákna pomocí indexů do seznamu `PointCloud`. Vrstva či množina vláken je reprezentována jako `vtkstd::vector<Fiber*>`.

6.3.3 AddAttachmentData()

Metoda provede zpracování oblasti úponu. Je-li specifikován parametr `bone`, metoda provede výřez úponové oblasti z povrchového modelu kosti. Tyto povrchové výřezy jsou poté uloženy do dvou seznamů¹⁰ a jsou dostupné pomocí metody `vtkstd::vector<vtkPolyData*> GetAttachmentsSurfaces()` pro možnost uložení do souboru a znovupoužití. V případě, že je parametr `bone` roven `NULL`, metoda očekává v objektu `data` již dříve vyříznutý povrch oblasti úponu.

6.3.4 Execute()

Hlavní metoda dekompozice. Metoda nejprve inicializuje výstupní datové objekty metodou `InitData()` a volá hlavní metodu `BuildMesh()`, která implementuje algoritmus 1 popsany v sekci 4 pro dekompozici svalu na vlákna. Data jsou pak předána na výstup metodou `DoneMesh()` a alokovaná paměť je uvolněna v metodě `ClearMesh()`.

Jelikož dekompozice velmi často využívá interpolaci vláken, implementovali jsme univerzální interpolační třídu `SplineInterpolator`. Pro interpolaci vlákna třídy `Fiber` pak dekompoziční třída implementuje metodu `InterpolateFiber()`.

`SplineInterpolator` implementuje všechny metody interpolace popsané v sekci 6.1 přímo nad vrcholy vlákna. Třída `Point` tedy pro interpolátor zavádí jakékoliv potřebné proměnné včetně parametru t a koeficientů a , b , c a d , není tak nutné převádět lomené křivky na jiné datové objekty. Po aplikaci metody interpolace metodou `SplineInterpolator::Interpolate()` jsou pak do předaného vlákna vloženy indexy nových vrcholů, jejichž souřadnice jsou přirozeně vloženy do pole `PointCloud`.

V případě interpolace povrchu v metodě `InterpolateSurface()`, kterou hlavní metoda `BuildMesh()` začíná, je sestaveno dočasné vlákno `CrossPoints`, které představuje jeden vláknový řez, nad kterým je volána metoda `InterpolateFiber()`, která dle zadaných parametrů vlákno interpoluje. Vlákna povrchu jsou tedy sestavována po řezech jejich postupnými interpolacemi. Podobný postup je použit i pro vytvoření postranních vrstev vláken.

Vzorkování objemu svalu je implementováno pomocí třídy `vtkOBBTree` [Kol]. Tato třída nad modelem svalu nejprve sestaví strom orientovaných hraničních

¹⁰Pro počátek a konec svalu

boxů (*Oriented Bounding Box*, OBB), tedy stromovou strukturu, ve které je povrchový model postupně dělen na menší části, pro které jsou nalezeny orientované boxy. Kořenem stromu je tedy box celého modelu. Pro získání potomků rodičovského uzlu je část modelu ohraničená boxem rodiče rozříznuta na dvě další části s přibližně stejným počtem primitiv, pro které jsou opět nalezeny menší orientované boxy. Pro výpočet průsečíků přímky a povrchového modelu pak `vtkOBBTree` nalezne průsečík s tímto stromem, získá omezenou množinu trojúhelníků původního modelu uvnitř hierarchicky nejnižšího boxu, a tím zkrátí i sekvenční hledání přesného průsečíku s modelem.

Vlákna v objemu svalu jsou vytvořena v metodě `BuildBottomFibers()` a po vrstvách zpracovávána metodou `OffsetFibers()` pro odstranění efektu "hrábě" popsaného v sekci 5.6.1. Jelikož je použita interpolace lineární po směru normál, `SplineInterpolator` zde není použit.

Přiřazení vláken k bodům úponu je během zpracovávání svalu dáno hodnotou proměnné `Point::AttachmentConnectionID`, body úponů tedy nejsou při výpočtu upnuté do vlákna fyzicky zařazeny jako nové vrcholy. Tím je zachována konzistence dat a urychleny¹¹ případné opravy po kontrole lámání vlákna. Mechanismus upínání vláken je také separován na část na počátku svalu a část na konci svalu, většina metod je však navržena tak, aby volbou parametru `int side` byla schopna zpracovávat obě části. Data jsou také pro obě části separována, pomocí dvojrozměrných polí indexovatelných konstantami `ATTACHMENT_START` a `ATTACHMENT_END`.

Vlákna jsou také v průběhu dekompozice sémanticky oddělena. Proměnná `Fibers` obsahuje vlákna na svrchním povrchu svalu, proměnná `BottomFibers` obsahuje vlákna v objemu svalu a proměnné `LeftFibers` a `RightFibers` obsahují postranní vrstvy vláken.

Po dokončení dekompozice jsou tedy jednotlivé množiny vláken kompletovány metodou `BuildAndOutputConnectedData()`, která fyzicky vlákna upne vložením bodů úponu na oba konce vláken a zruší sématické rozdělení vláken jejich zkopírováním do množiny vláken `OutputFibers`. Tato množina je poté podélně interpolována metodou `InterpolateFibers()`, kdy je každé vlákno interpolováno jednotlivě opět pomocí metodou `InterpolateFiber()`. Množina `OutputFibers` je pak připravena pro konverzi do objektu `vtkPolyData` metodou `DoneMesh()`.

Třída uvolňuje všechna vnitřní data, neuvolňuje však data vstupní, tato data je nutné uvolňovat externě. Třidu je možné pro stejná data využít opakovaně (tedy lze měnit vstupní parametry a volat metodu `Update()` opakovaně), pro nová data je však nutné vytvořit novou instanci třídy `vtkMuscleDecomposer`.

¹¹operace nad polem `vector` jsou lineární složitosti

6.4 Demonstrační aplikace

Demonstrační a testovací aplikace je v souboru `demoApp.cpp`. Tato aplikace požaduje jako vstupní parametr příkazové řádky pouze složku se vstupními daty svalu pro dekompozici. Data v této složce musí být ve formátu VTK [Kol10] a musí splňovat následující pojmenování:

- `m0.vtk` (muscle) - soubor s povrchovým modelem svalu
- `ab#.vtk` (side a, bone) - soubor s povrchovým modelem kosti na počátku svalu. # značí číslo oblasti (0..n)
- `ao#.vtk` (side a, outline) - soubor s body hranice oblasti úponu na počátku svalu. # značí číslo oblasti (0..n)
- `bb#.vtk` (side b, bone) - soubor s povrchovým modelem kosti na počátku svalu. # značí číslo oblasti (0..n)
- `bo#.vtk` (side b, outline) - soubor s body hranice oblasti úponu na počátku svalu. # značí číslo oblasti (0..n)
- `f#.vtk` (fiber) - soubor s řídicím vláknem

Číslování začíná pro obě strany svalu od nuly. Příkladem je tedy sval Iliacus, který má na obou stranách pouze jednu oblast úponu, a jeho data jsou sestavena ze souborů `m0.vtk`, `ab0.vtk`, `ao0.vtk`, `bb0.vtk` a `bo0.vtk`.

Po provedení dekompozice je výsledek uložen do této složky pod názvem `output.vtk`, a povrchové modely vyříznutých oblastí pod názvy ve formátu `aa#.vtk` (side a, area) a `ba#.vtk` (side b, area), opět s # značícím pořadové číslo oblasti. Demoaplikace tyto soubory také vyhledává, a dává jim přednost před novým výřezem, pro urychlení dekompozice.

Demonstrační aplikace také implementuje pipeline VTK pro zobrazení výsledku dekompozice v kontextu povrchových modelů svalů a kostí. Aplikace je umístěna na přiloženém DVD, které také obsahuje v kořenovém adresáři soubor `Readme.txt` s popisem struktury DVD. Uživatelskou příručku k aplikaci lze nalézt v příloze A.

6.5 Získání umělých dat

Umělá data pro testování dekompoziční metody jsem vytvářel pomocí nástrojů Blender [Ble13] a `lhpBuilder`, který je součástí softwarového vybavení projektu VPHOP [VPH10]. Všechna potřebná data, včetně potřebného povrchového modelu svalu, jsou uložena ve VME formátu frameworku MAF2, pomocí aplikace `lhpBuilder` je však možné je exportovat do formátu *STL* (*STereoLithography*), který je podporován aplikací Blender.

Pro vytvoření dat jednoho svalu jsem nejprve importoval STL soubor povrchového modelu svalu do aplikace Blender. Následně jsem vytvořil jedno modelové vlákno opakovaným použitím modifikátoru *Extrude* na předem vytvořený vrchol. Volil jsem vhodný počet segmentů takto vytvořeného vlákna (zpravidla do deseti) a vrcholy vlákna ve *Vertex Edit* módu upravil tak, aby vlákno leželo na svrchním povrchu svalu, tzn. na povrchu směrem od kosti, pokud možno ke kraji tohoto povrchu. Pro výrazné usnadnění práce lze použít nástroj přichytnutí k povrchu.



Obrázek 6.7: Nástroj přichytnutí k povrchu

Poté jsem vlákno několikrát duplikoval, a duplikáty upravil, aby pokrývaly celý povrch svalu, se zavedením seřazení po tomto povrchu (z pohledu na povrch tedy zleva doprava) a pojmenoval jsem je f_0 až f_n (zpravidla jsem vytvořil okolo pěti vláken). Vlákna jsem rozmístil po šířce povrchu rovnoměrně s ohledem na náčrty v anatomických atlasech, v případě vícehlavých svalů jsem při krajích povrchu v oblasti rozštěpu nechal dostatek prostoru pro rozštěpená vlákna.

Vytvořená vlákna jsem pak označil a exportoval do formátu *OBJ* (*Wavefront OBJ file*), jelikož Blender neexportuje přímo do formátu VTK. Nastavení exportu jsou na obrázku B.6 v příloze B. Pro převod do formátu VTK jsem vytvořil vlastní pomocnou aplikaci *objToVtkBatch* (také součástí přiloženého CD), která po spuštění s parametrem jména OBJ souboru provede jeho rozdělení do jednotlivých souborů řídicích vláken a konverzi na formát VTK.

Povrchové modely svalu, kostí a data oblastí úponu lze přímo exportovat do formátu VTK z aplikace *lhpBuilder*. Soubory jsem pak přejmenoval, aby odpovídaly požadavkům ze sekce 6.4.

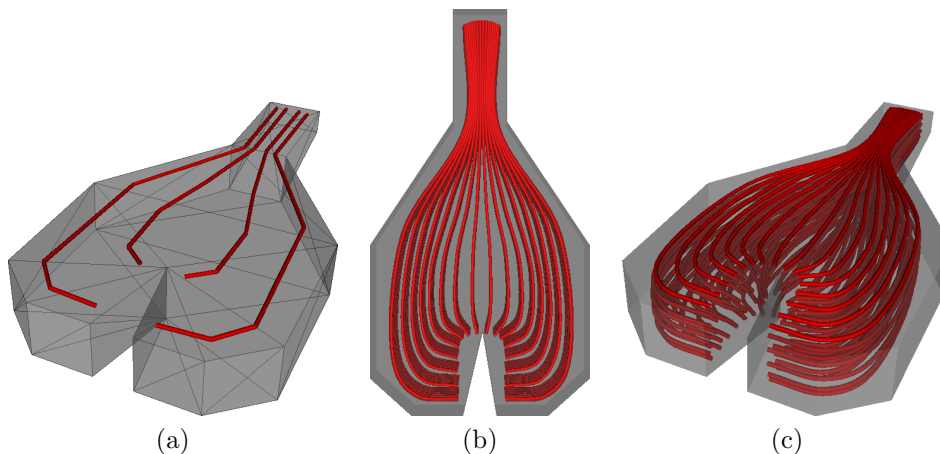
7 Výsledky

7.1 Dekomponovatelné typy svalů

Jak již bylo naznačeno, popsaná dekompoziční metoda je schopna dekomponovat i složitější typy svalů. Pro otestování schopnosti metody vytvořit očekávaný průběh vláken jsem tedy vytvořil umělá, ale topologicky odpovídající testovací data, navržená tak, aby byly výsledky dekompozice zřetelné. Pro přehlednost jsem také nezahrnul data pro úpony vláken k šlachám či kostem. Pro všechny dekompozice byly použity následující parametry:

```
InterpolationSubdivision = 10
SurfaceSubdivision = 5
VolumeSubdivision = 5
```

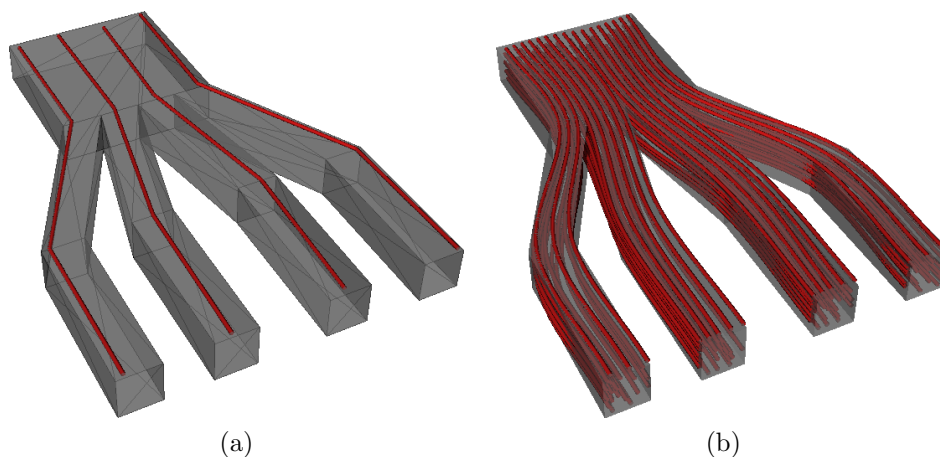
Prvním testem je schopnost metody dekomponovat pennate a bipennate svaly. Jelikož je pennate úhel dán průběhem řídicích vláken, která jsou použita pro dekompozici celého objemu svaly, pennate svaly jsou zpracovány zcela stejným způsobem, jako přímé nebo fusiform svaly. Speciálně ukázka umělého bipennate svaly je na obrázku 7.1.



Obrázek 7.1: Modelový bipennate sval (a) Vstupní data (b,c) Výsledek dekompozice.

V sekci 5.5.1 jsme si popsali schopnost metody dekomponovat vícehlavé svaly. V lidském těle se vyskytují nejvýše čtyřhlavé svaly, metoda je však tyto složitější svaly schopna také zpracovat. Ukázka dekompozice modelového čtyřhlavého svaly je na obrázku 7.2. Je možné pozorovat menší nedostatek metody způsobeným rovnoměrným dělením vláken na dvě skupiny o stejném počtu vláken v oblasti rozštěpu. Krajní hlavy svaly tak mají menší počet vláken, než hlavy mezi nimi. Toto chování však ale není příliš na závadu, jelikož při reálné dekompozici nejsou

svaly takto pravidelné a souměrné a pro zachování detailů povrchu je samozřejmě vhodné hlavy svalu popsat více než jedním řídicím vláknem. Metoda však samozřejmě není omezena na čtyřhlavé svaly, je možné dekomponovat svaly s libovolným počtem hlav (za předpokladu, že jsou k dispozici korektní řídicí data).



Obrázek 7.2: Modelový čtyřhlavý sval (a) Vstupní data (b) Výsledek dekompozice.

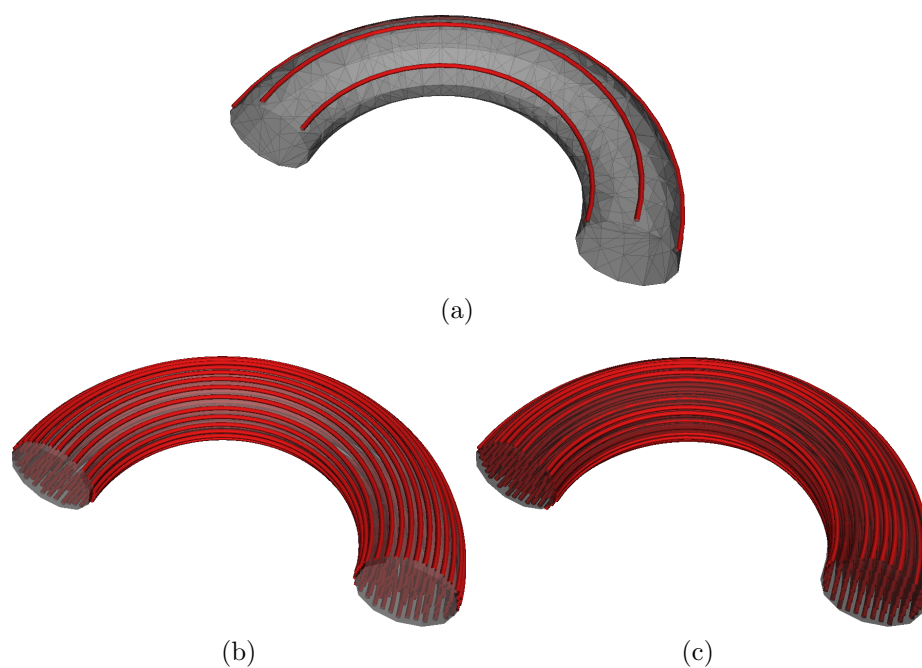
V sekci 2.1.3 jsme naznačili, že by bylo možné metodu použít i pro kruhové svaly. Tyto svaly by bylo možné dekomponovat rozdělením svalu na dvě "půlkruhové" části, podle šlachy, ke které se z obou stran vlákna upínají. Obě poloviny pak lze dekomponovat jako klasické paralelní svaly. Topologicky pak jde o dekompozici poloviny anuloidu, jak je tomu na obrázku 7.3. Poznamenejme, že pro tuto dekompozici bylo použita klasická kubická interpolace s parabolickým doběhem, která lépe kopíruje kruhový průřez anuloidu při dekompozici povrchu. Na obrázku 7.3 Je ukázka dekompozice s použitím výchozí Catmull-Rom interpolace. Jelikož je tato metoda použita i pro interpolaci vláknových řezů, a jelikož tato metoda nabízí napnutější vlákna, můžeme vidět, že některá vlákna jsou pod kruhovým průřezem anuloidu, tedy uvnitř povrchového modelu svalu.

7.2 Ukázky výsledků dekompozice

Pro otestování metody na skutečných datech jsem pro vybrané svaly vytvořil umělá řídicí data a provedl jejich dekompozici. Pro tuto sekci je použito následující nastavení dekompozice pro větší přehlednost obrázků v textu:

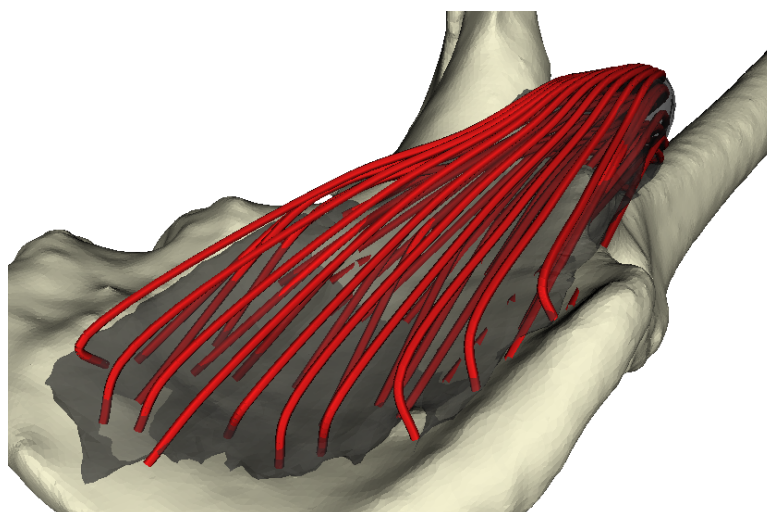
```
InterpolationSubdivision = 10
SurfaceSubdivision = 5
VolumeSubdivision = 5
```

Ukázky svalů jsou však také v přílohách B.2 - B.5 při nastavení ze sekce 7.1 v lepší kvalitě.



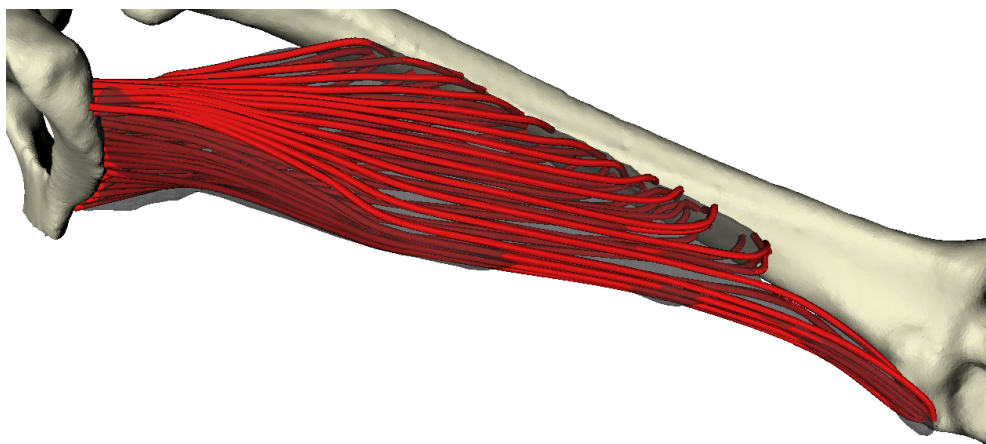
Obrázek 7.3: Dekompozice poloviny kruhového svalu (a) Vstupní data (b) Výsledek dekompozice s použitím kubické interpolace (c) Výsledek s použitím výchozí Catmull-Rom interpolace

První ukázkou je sval *Iliacus* na obrázku 7.4. Tento sval má v oblasti pánve velký pennate úhel, ukazuje se tedy funkčnost způsobu zkracování vláken ze sekce 5.7 pro potlačení zpětně ohnutých vláken.



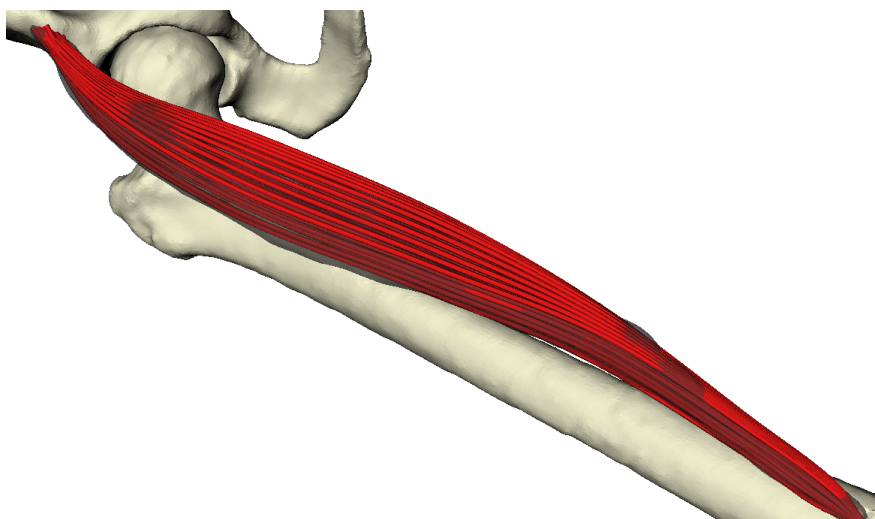
Obrázek 7.4: Dekompozice svalu *Iliacus*. Tento sval se vyznačuje ostrým pennate úhlem v oblasti pánve.

Příkladem rozštěpeného svalu je sval *Adductor Magnus* na obrázku 7.5. Sval se štěpí na hlavní část upnutou ke straně *Femuru*, a menší část upnutou ke kolenní části *Femuru*. Tento na svém počátku objemný sval se také na konci výrazně zužuje, je tedy vhodnou ukázkou distribuce vláken v objemu svalu.



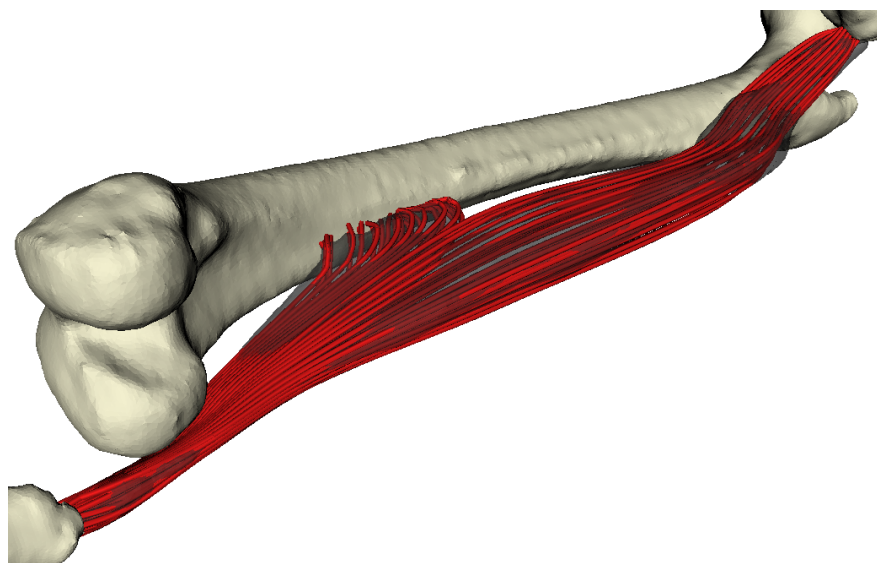
Obrázek 7.5: Dekompozice svalu *Adductor Magnus*. Tento sval je na konci rozštěpen

Na obrázku 7.6 je ukázka dekompozice svalu *Rectus Femoris*. Tento sval je ve skutečnosti typu bipennate, bohužel se mi však nepodařilo i při velkém uměleckém snažení vytvořit taková řídicí vlákna, která by tento fakt po dekompozici odrážela. Po nahlédnutí do anatomického atlasu (obrázek 2.2 v sekci 2) můžeme zjistit, že mi k tomu chyběla data počáteční šlachy (zvané proximální aponeuróza), která bipennate tvarování vláken tohoto svalu způsobuje. Sval je nicméně v této podobě dobrým příkladem fusiform svalu.



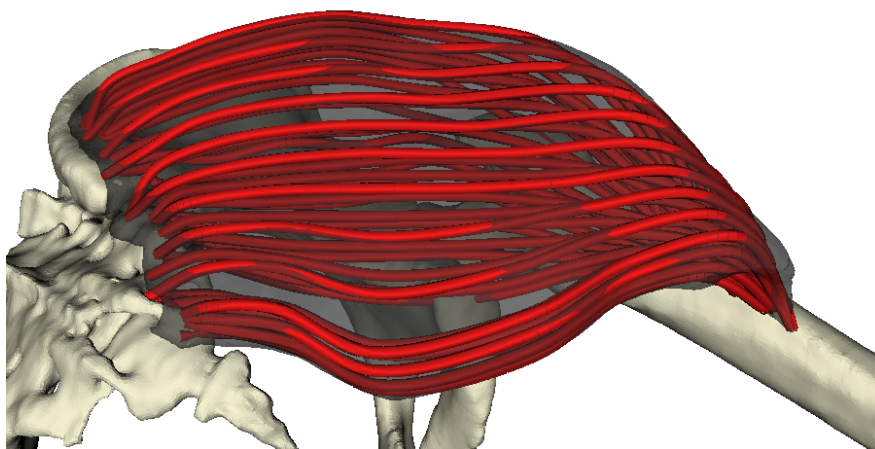
Obrázek 7.6: Dekompozice svalu *Rectus Femoris*. Ve skutečnosti je tento sval typu Bipennate, nepodařilo se mi však taková řídicí vlákna vytvořit

Nejsložitějším svaem, na který jsem narazil, je sval *Biceps Femoris* na obrázku 7.7. Ačkoliv je dekompozice tohoto svalu také možná, musel jsem řídicí data vytvořit na spodní, rovnější části svalu (blíže k *Femuru*). Složitost svalu spočívá v jeho tvaru - delší hlava svalu svojí střední částí obepíná sval *Adductor Magnus*, ale přesto je krátká hlava v pohledu shora v zákrytu touto střední částí dlouhé hlavy, což vede k obtížné dekompozici vláken krátké hlavy, jelikož metoda jen obtížně hledá ohraje oblasti rozštěpu. Pro tento tvar svalu by tedy bylo vhodnější rozštěp detekovat jinak, například porovnáním délky hlav nebo rozdělením povrchového modelu a řídicích dat na dvě hlavy.



Obrázek 7.7: Dekompozice svalu *Biceps Femoris*. Sval je dvojhlavý, upíná se ke třem kostem a ohýbá se okolo kosti *Femur*.

Poslední ukázkou dekompozice je obrázek 7.8, což je výsledek dekompozice svalu *Gluteus Maximus*. Tento sval není topologicky nejsložitější i přes jeho dvojhlavovost, problém, který u tohoto svalu nastává je, že povrchový model kosti Křížové (*Sacrum*) obsahuje otvory (dané tvarem kosti, nikoliv chybou extrakce dat). Metoda výřezu oblasti úponů však s takovými daty není schopna zcela korektně pracovat. Jako řešení jsem tedy byl nucen ručně triangulovat hranici oblasti, a povrch oblasti úponu takto aproximovat.

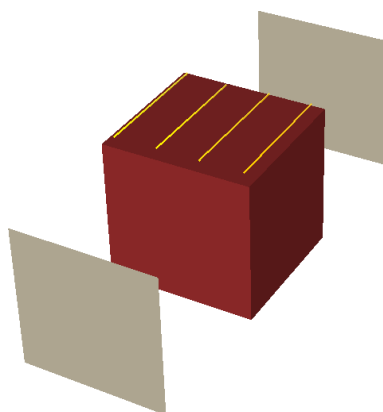


Obrázek 7.8: Dekompozice svalu *Gluteus Maximus*. Sval je dvojhlavý, upíná se ke třem kostem.

7.3 Rychlost dekompozice

Rychlost dekompozice byla měřena na laptopu Packard Bell Easynote TS s procesorem Intel Core i5-2410M při 2.3GHz, 6GB RAM při 666MHz a 640GB diskem o 5400 otáčkách za minutu s 8MB cache paměti, pod operačním systémem Windows 7 Home.

7.3.1 Závislost na vstupních datech



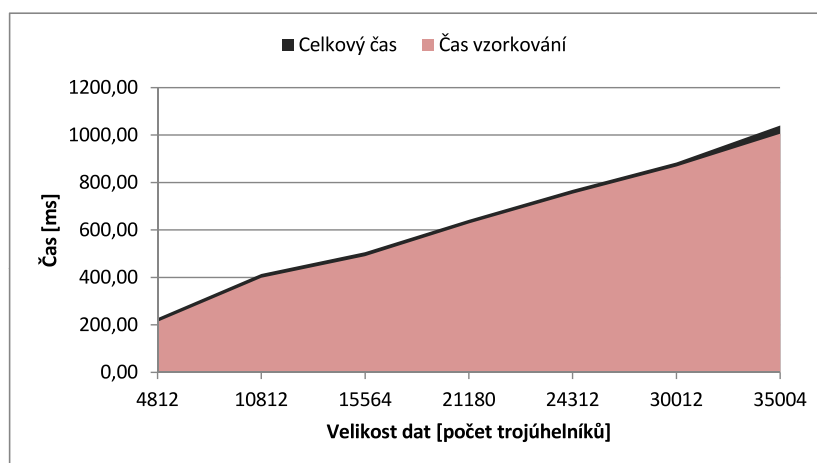
Obrázek 7.9: Data pro testování rychlosti dekompozice. Červeně je označen dekomponovaný sval, šedě oblasti úponů, žlutě řídicí vlákna

Pro otestování rychlosti dekompozice jsem vytvořil jednoduchá testovací data, jejichž ukázka je na obrázku 7.9. Tato data je možné snadno upravovat, tedy změnit počet řídicích vláken, počet segmentů řídicích vláken i počet trojúhelníků

dekomponovaného svalu. Pro následující testy byla použita nastavení dekompozice dle sekce 7.1.

Prvním testem je závislost na doby dekompozice na počtu trojúhelníků dekomponovaného svalu. Jako testovací povrchový model svalu byly použity přerozdělované krychle (vytvořené použitím operátoru `Subdivide` na trojúhelníky jednoduché krychle) s rostoucími počty trojúhelníků. Měřil jsem mezičasy po každé fázi algoritmu dekompozice (fáze odpovídají podsekcím sekce 5). Použil jsem 5 řídicích vláken o 10 segmentech.

Dle očekávání se projevila nejvíce časová náročnost fáze vzorkování objemu modelu svalu, jelikož ta jediná s povrchovým modelem svalu pracuje. Z grafu 7.10 je možné pozorovat lineární závislost - třída `vtkOBBTree` tedy sestavuje strom v lineárním čase. Číselné výsledky měření jsou v tabulce B.1.

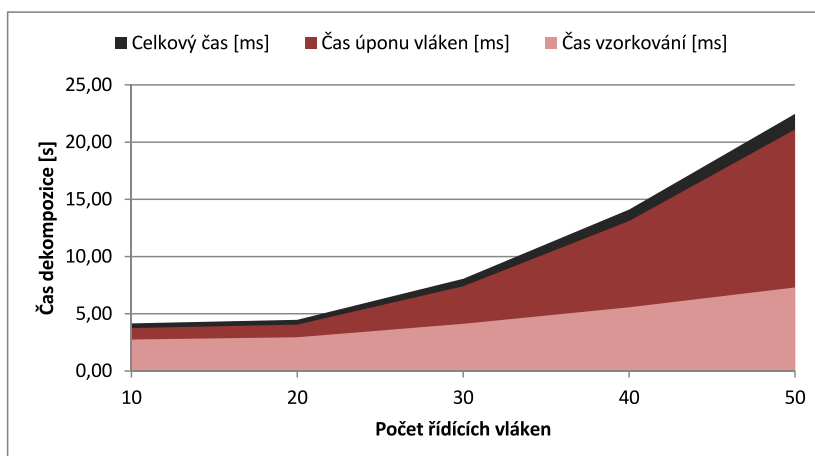


Obrázek 7.10: Závislost doby dekompozice na počtu trojúhelníků povrchového modelu svalu

Dalším testem je závislost na doby dekompozice na počtu řídicích vláken. Pro tento test jsem vytvořil sady dat o stoupajícím počtu řídicích vláken a opět měřil mezičasy v každé fázi dekompozice. Jako model svalu jsem použil krychli o přibližně 15ti tisících trojúhelníků, což je běžný počet v nám dostupných povrchových modelech svalů. Všechna vlákna měla 10 segmentů.

V grafu 7.11 se projevuje lineární závislost doby vzorkování, jelikož každé vlákno generuje nové vrcholy na povrchu svalu, ve kterých sval vzorkujeme. Projevuje se také kvadratická časová složitost času upínání vláken k bodům úponu, jelikož každé řídicí vlákno generuje nová neupnutá vlákna. Složitost Maďarského algoritmu je $O(n * m)$ kde n je počet koncových bodů vláken (dle parametrů dekompozice závislých na vstupních parametrech) a m je počet bodů úponů. Jelikož ale chceme upnout všechna vlákna, pak $m == n$ a tedy výsledná složitost upínání je $O(n^2)$. Číselné výsledky měření jsou v tabulce B.2.

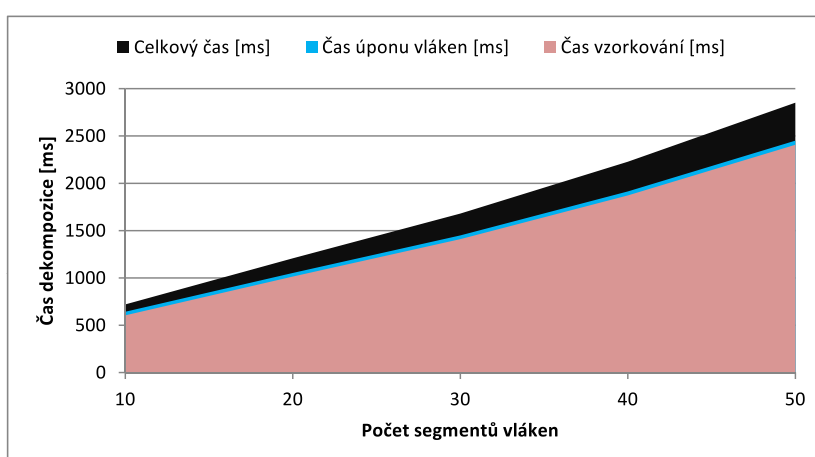
Posledním testem v této kategorii testů je závislost na doby dekompozice na počtu segmentů řídicích vláken. Pro tento test jsem vytvořil sady dat pěti



Obrázek 7.11: Závislost doby dekompozice na počtu řídicích vláken

řídicích vláken se stoupajícím počtem segmentů a opět měřil mezičasy v každé fázi dekompozice. Jako model svalů jsem opět použil krychli o přibližně 15ti tisících trojúhelníků.

V grafu 7.12 se opět projevuje lineární závislost doby vzorkování, jelikož každý segment představuje nový vrchol na povrchu svalů. Číselné výsledky měření jsou v tabulce B.3. V celkovém čase dekompozice se také více projevuje lineární růst času výpočtu v interpolačních fázích dekompozice - s rostoucím počtem segmentů roste počet příčných interpolací, které jsou díky dodatečným paměťovým manipulacím pomalejší, než interpolace podélné. V tomto testu se však počet vláken nemění a doba potřebná pro jejich upnutí je tak konstantní.



Obrázek 7.12: Závislost doby dekompozice na počtu segmentů řídicích vláken

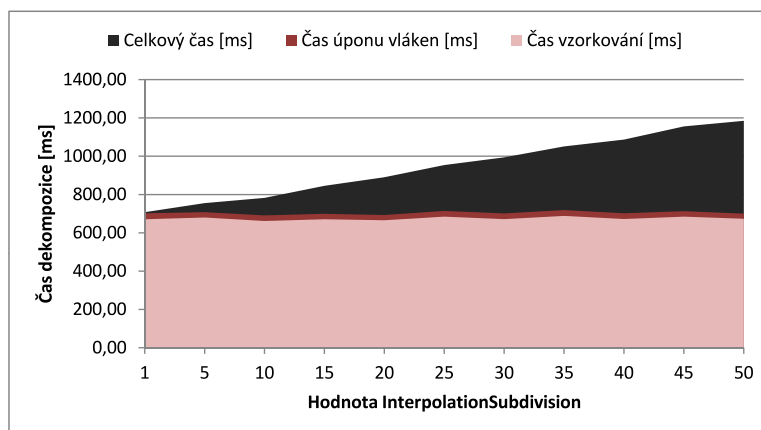
O složitosti metody v závislosti na vstupních datech se tedy dá říci, že je $O(n_f * n_s * n_t)$ kde n_f je počet vláken, n_s je počet segmentů vláken a n_t je počet

trojúhelníků povrchového modelu svalu (prohledávání OBB stromu je ekvivalentní binárnímu hledání).

7.3.2 Závislost na vstupních parametrech

V této sekci použijeme vždy pět řídicích vláken o deseti segmentech a model svalu s přibližně 15 tisíci trojúhelníky, měnit budeme vstupní parametry a budeme pozorovat rychlost dekompozice a poměr v mezíchasech výpočtu v jednotlivých fázích dekompozice. Není-li určeno jinak, parametry dekompozice opět odpovídají parametrům v sekci 7.1.

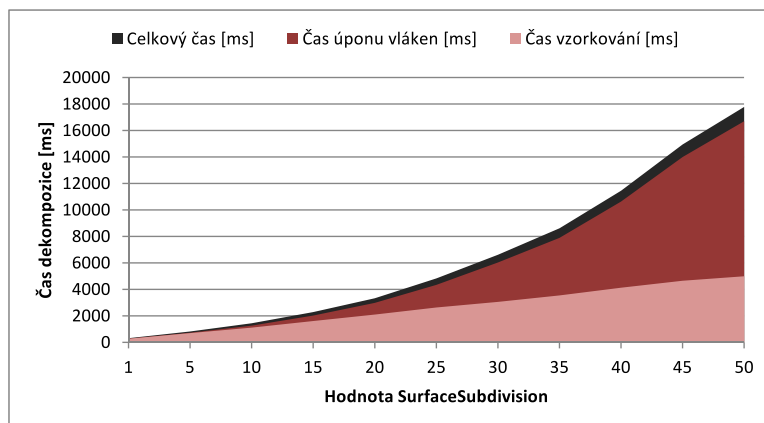
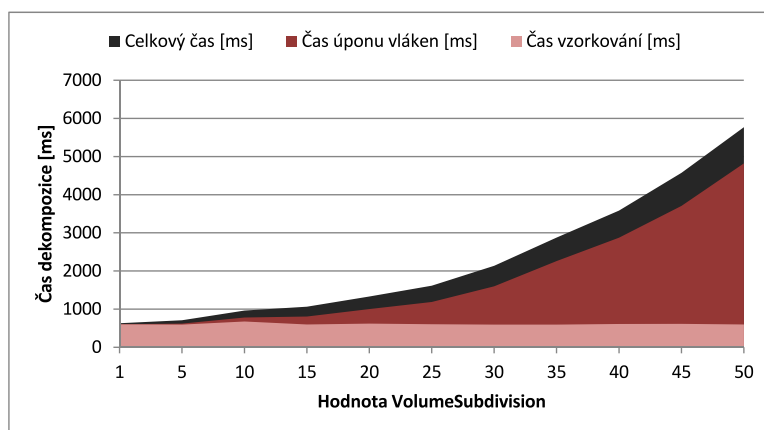
Nejprve si rozebereme závislost doby dekompozice na hodnotě parametru `InterpolationSubdivision`, tedy na počtu vytvořených segmentů mezi původními vrcholy vláken pro jejich vyhlazení. Můžete si povšimnout, že jako v případě grafu 7.10 je i v grafu 7.13 dominantní doba vzorkování objemu. Rozdílem je, že se počet trojúhelníků povrchového modelu svalu se nemění, stejně jako se nemění počet vláken. Doba vzorkování je tak konstantní, stejně jako doba úponu vláken. Jelikož je však poslední operací dekompozice interpolace všech vláken, můžeme pozorovat lineární závislost vzhledem k hodnotě `InterpolationSubdivision` (v grafu 7.13 černá část). Číselné výsledky tohoto měření jsou v tabulce B.4.



Obrázek 7.13: Závislost doby dekompozice na parametru `InterpolationSubdivision`

V grafu 7.14, stejně jako v grafu 7.11 vidíme lineární závislost doby vzorkování a kvadratickou závislost doby úponu vláken. Jde o závislost na parametru `SurfaceSubdivision`, který definuje rozmnožení vláken na povrchu svalu. Každé povrchové vlákno opět způsobuje další vzorkování objemu svalu. Tato vlákna jsou také předlohou pro vlákna v objemu svalu, a tedy roste počet vláken, které je nutné na koncích upnout. Číselné výsledky tohoto měření jsou v tabulce B.5.

Posledním testem zde je závislost doby dekompozice na hodnotě parametru `VolumeSubdivision`, zobrazená v grafu 7.15. Více vláken v objemu svalu však neovlivňuje počet potřebných vzorků, a doba vzorkování je tak konstantní. Doba upínání vláken je však stále s rostoucím počtem vláken kvadratická.

Obrázek 7.14: Závislost doby dekompozice na parametru `SurfaceSubdivision`Obrázek 7.15: Závislost doby dekompozice na parametru `VolumeSubdivision`

Celkově se tedy o časové složitosti metody dá říci, že interpolace pomocí Catmull-Rom metody jsou lineární složitosti, vzorkování je po sestavení stromu také lineární, ale upínání vláken k bodům úponů je kvadratické složitosti. Implementace dekompoziční metody je však na vstupních parametrech složitostí výpočtu závislá kubicky, jelikož počet řídicích vláken je násoben rozlišením povrchu, tento počet vláken na povrchu je násoben je poté násoben rozlišením objemu, a vlákna objemu jsou poté interpolována s podélným rozlišením, celkově tedy $O(\text{SurfaceSubdivision} * \text{VolumeSubdivision} * \text{InterpolationSubdivision})$, zatímco složitost upínání je pouze $O(\text{SurfaceSubdivision} * \text{VolumeSubdivision})$.

Pokud dáme dohromady výsledky sekcí 7.3.1 a 7.3.2, dojdeme k výslednému hornímu složitosti:

$$O(\text{SurfaceSubdivision} * n_f * n_t * \text{VolumeSubdivision} * \text{InterpolationSubdivision} * n_s)$$

8 Závěr

Metoda dekompozice svalů na svalová vlákna popsaná v této diplomové práci vyžaduje dodatečná, prozatím nesnadno získatelná data pro řízení dekompozice, je však schopna lépe dekomponovat i složitější typy svalů. Metoda respektuje průběh řídicích vláken při konstrukci všech vláken v objemu svalu a za pomoci kontroly ohybu při jejich úponu metoda umožňuje zachovat pennate úhel v celém objemu svalu. V případě vícehlavovosti svalu metoda oblast rozštěpu detekuje a přizpůsobuje vlákna tak, aby rozštěpení bylo zavedeno i do dekomponovaného svalu.

Implementace metody je provedena v jazyce C++, s využitím knihoven *VTK* pro kompatibilitu s frameworkem *MAF2*. Implementace pro prostředí *MAF3* nebyla po dohodě s vedoucím práce provedena z důvodu zásadních změn v jádře a nedostatečnou dokumentací frameworku.

Výsledky dekompozice ukazují, že je možné pomocí navržené a implementované metody dosáhnout realistické dekompozice svalů v relativně krátkém čase. Implementaci by ale nebylo možné použít pro aplikace reálného času, jelikož je metoda určena pro statickou dekompozici a uložení jejích výsledků pro znovupoužití.

Bylo by však pravděpodobně možné metodu výrazně urychlit. Nejvíce časově náročné jsou při dekompozici fáze vzorkování a fáze připojení vláken k oblastem úponu. Vzorkování je dobře separovatelný problém, jelikož jsou jednotlivé vzorky navzájem nezávislé. Paralelním výpočtem vzorkování například po jednotlivých vláknových řezech bychom tak mohli čas strávený v této fázi zkrátit. Připojování vláken k bodům úponu by mohlo být urychleno greedy heuristikou, která by vlákna připojila dle nejmenší metriky, bez ohledu na minimalizaci celkové ceny. Bylo by však nutné takovéto připojení poté zkontrolovat, zda nezpůsobuje nechtěné připojení vláken na dlouhou vzdálenost, křížem přes ostatní vlákna (což může při greedy připojování snadno nastat).

K menším nedostatkům metody patří její horší stabilita v oblastech rozštěpu u svalů s dlouhou a krátkou hlavou a omezení využívané třídy pro extrakci oblasti úponu způsobené jejími požadavky na spojitost a bezděrovost vstupního povrchu. V případě stability by řešením mohlo být dodatečné podpůrné hledání hlav svalu (například měřením odklonu vláken či vzdálenosti oblastí úponu hlav) či ruční specifikace rozštěpu (tedy specifikace řídicích vláken, mezi kterými dochází k rozštěpu) a adekvátní úprava metody pro využití této informace. V případě extrakce oblasti úponu by v případě neúspěchu mohla být hranice oblasti triangulována jako aproximace povrchu oblasti a vlákna by po připojení takto získané oblasti pak mohla být povrchovým modelem kosti oříznuta.

Sazba tohoto dokumentu byla vytvořena pomocí nástroje \LaTeX [Lat10].

Reference

- [BD05] Silvia S. Blemker and Scott L. Delp. Three-dimensional representation of complex muscle architectures and geometries. *Annals of Biomedical Engineering*, 33(5):661–673, 2005.
- [Bel04] John T. Bell. Visualization Toolkit (VTK) Tutorial, 2004. [Online] <http://www.cs.uic.edu/~jbell/CS526/Tutorial/Tutorial.html>, akt. duben 2013.
- [BG88] Phillip J. Barry and Ronald N. Goldman. A recursive evaluation algorithm for a class of catmull-rom splines. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '88, pages 199–204, New York, NY, USA, 1988. ACM.
- [Ble13] Blender, a suite of tools enabling the creation of and replay of linear and real-time, interactive 3D content, 2013. <http://www.blender.org/>.
- [ČGMH01] R. Čihák, M. Grim, M. Med, and I. Helekal. *Anatomie 1*. Grada, 2001.
- [CR74] E. Catmull and R. Rom. A class of local interpolating splines. *Computer aided geometric design*. Academic Press., pages 317–326, 1974.
- [DKM05] S. Dong, S. Kircher, and Garland M. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design*, 22(5):392–423, 2005.
- [GL18] H. Gray and W.H. Lewis. *Anatomy of the human body*. Lea & Febiger, 1918. [Online] <http://www.bartleby.com/107/>, akt. duben 2013.
- [Jež10] František Ježek. Geometrické a počítačové modelování, 2010. Pomocný učební text, verze 12, [Online] <http://geometrie.kma.zcu.cz/index.php/www/content/download/1122/3185/file/gpm-all-FJ.pdf>, akt. duben 2013.
- [KCea12] Josef Kohout, Gordon Clapworthy, and et al. Final morphable musculoskeletal model, work package wp10: Morphable musculoskeletal model. Technical report, VPHOP, 2012.
- [KCMV12a] J. Kohout, G. Clapworthy, S. Martelli, and M. Viceconti. Fast realistic modelling of muscle fibres. In *Communications in Computer and Information Science (CCIS)*, Springer-Verlag, 2012. Přijato k publikaci.

-
- [KCMV12b] J. Kohout, G. Clapworthy, S. Martelli, and M. Viceconti. Muscle fibres modelling. In *GRAPP/IVAPP*, pages 58–66, 2012.
- [KKM12] Josef Kohout, Petr Kellnhofer, and Saulo Martelli. Fast deformation for modelling of musculoskeletal system. In *Proceedings of the International Conference on Computer Graphics Theory and Applications: GRAPP 2012*, pages 16–25, Rome, February 2012.
- [Kol] Kolektiv autorů. vtkOBBTree class reference. [Online] <http://www.vtk.org/doc/release/4.0/html/classvtkOBBTree.html>, akt. duben 2013.
- [Kol03] Kolektiv autorů. VTK 4.2.1 documentation, 2003. revize 1.1495, [Online] <http://www.vtk.org/doc/release/4.2/html/>, akt. duben 2013.
- [Kol10] Kolektiv autorů. VTK file formats for VTK version 4.2, březn 2010. [Online] <http://www.vtk.org/VTK/img/file-formats.pdf>, akt. duben 2013.
- [Kru01] C J C Kruger. Constrained cubic spline interpolation for chemical engineering applications. 1, 2001. [Online] <http://www.korf.co.uk/spline.pdf>, akt. duben 2013.
- [Kuh05] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52(1):7–21, 2005.
- [LA07] Keng Siang Lee and Golam Ashraf. Simplified muscle dynamics for appealing real-time skin deformation. In *CGVR*, pages 160–160, 2007.
- [Lat10] LaTeX, a document preparation system, 2010. [Online] <http://www.latex-project.org/>.
- [LDW⁺07] Drew A Lansdown, Zhaohua Ding, Megan Wadington, Jennifer L Hornberger, and Bruce M Damon. Quantitative diffusion tensor mri-based fiber tracking of human skeletal muscle. *Journal of Applied Physiology*, 103(2):673–81, 2007.
- [LGMP11] David I.W. Levin, Benjamin Gilles, Burkhard Mädler, and Dinesh K. Pai. Extracting skeletal muscle fiber fields from noisy diffusion tensor data. *Medical Image Analysis*, 15(3):340 – 353, 2011.
- [LRJ⁺12] D. Lee, K. Ravichandiran, K. Jackson, E. Fiume, and A. Agur. Robust estimation of physiological cross-sectional area and geometric reconstruction for human skeletal muscle. *Journal of Biomechanics*, 45(8):1507–13, 2012.

-
- [LZRM11] Y. T. Lu, H. X. Zhu, S. Richmond, and J. Middleton. Modelling skeletal muscle fibre orientation arrangement. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(12):1079–1088, 2011.
- [MAF13] Multimod application framework, a software framework for the rapid development of computer aided medicine applications., 2013. [Online] https://www.biomedtown.org/biomed_town/MAF, akt. duben 2013.
- [Ng 01] V. Ng Thow Hing. *Anatomically-based Models for Physical and Geometric Reconstruction of Humans and Other Animals*. PhD thesis, University of Toronto, 2001.
- [Sta04] Cyrill Stachniss. C implementation of the hungarian method, 2004. [Online] <http://www.informatik.uni-freiburg.de/~stachnis/resources.html>, akt. duben 2013.
- [TSC96] D. Thalmann, Jianhua Shen, and E. Chauvineau. Fast realistic human body deformations for animation and VR applications. In *Computer Graphics International, 1996. Proceedings*, pages 166–174, 1996.
- [VPH10] VPHOP: The osteoporotic virtual physiological human., 2010. <http://www.vphop.eu/>.
- [WG97] Jane Wilhelms and Allen Van Gelder. Anatomically based modeling. In *SIGGRAPH*, pages 173–180, 1997.
- [YSK09] Cem Yuksel, Scott Schaefer, and John Keyser. On the parameterization of catmull-rom curves. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, SPM '09, pages 47–53, New York, NY, USA, 2009. ACM.

Přílohy

A Uživatelská příručka

Program je na přiloženém DVD¹² jak v přeložené podobě, tak v podobě zdrojových kódů. Pro překlad aplikace je nutné nejprve získat a přeložit knihovnu VTK (lze stáhnout z webu <http://www.vtk.org/>). Pro překlad této knihovny je nutný nástroj CMake (www.cmake.org). Aplikace je kompatibilní s VTK verze 4.2 a vyšší. Po přeložení knihovny pak bude pravděpodobně nutné upravit cesty ke knihovně VTK, tedy po otevření `MuscleDecomposition.sln` otevřít vlastnosti projektu (menu Project→Properties) a upravit cesty

- V sekci Configuration Properties→C/C++→General - položka Additional Include Directories: upravit cesty k hlavičkovým souborům (tedy části cest "C:\MAF\MAF2\Libraries\VTK" nahradit skutečnou cestou k souborům VTK)
- V sekci Configuration Properties→Linker→General - položka Additional Library Directories: upravit cestu k přeloženým VTK knihovnám (tedy upravit "C:\MAF\MAF2\bin\Debug" na skutečnou složku Debug s přeloženými .lib soubory knihovny VTK)

Poté by již měl překlad proběhnout bez problémů.

Přeložený program `MuscleDecomposition.exe` se spouští jako konzolová aplikace. Prvním, povinným, parametrem programu je adresář s daty. Tato data musí být kompletní a datové soubory musí být pojmenovány podle popisu v sekci 6.4. Další tři parametry jsou nepovinné, jde o celočíselné parametry v pořadí `InterpolationSubdivision(IS > 0)`, `SurfaceSubdivision(SS > 0)` a `VolumeSubdivision(VS > 0)`. Pokud jsou tyto parametry specifikovány, je možné specifikovat i desetinný parametr `PennationTreshold(PT ∈ ⟨π/2, π⟩)`. Formálně jsou tedy parametry následující

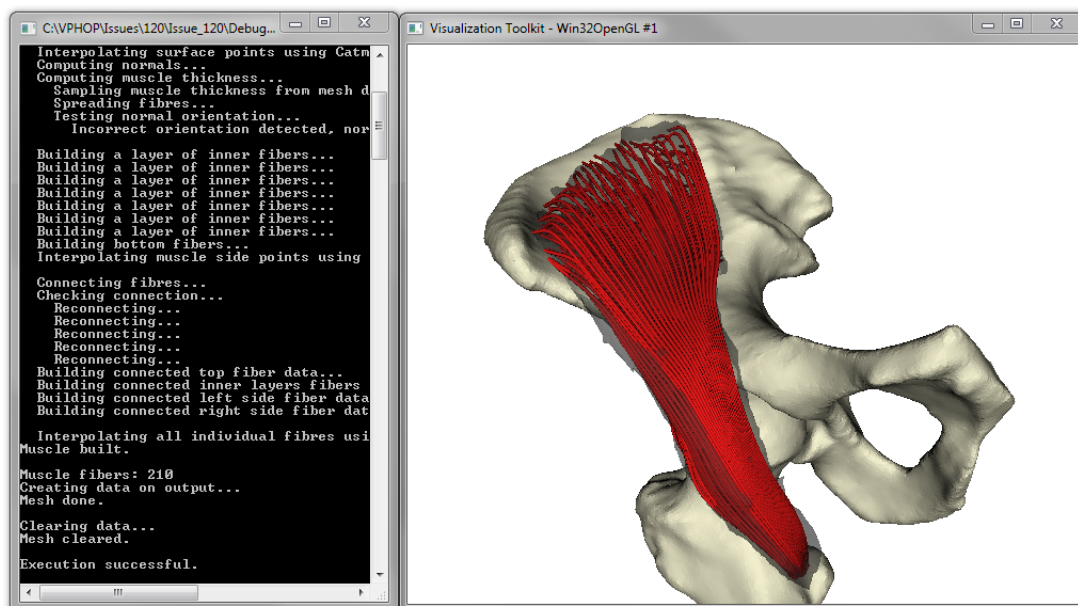
```
MuscleDecomposition.exe adresář_dat [IS SS VS [PT]]
```

Výsledek dekompozice je uložen do specifikovaného adresáře pod názvem `Output0.vtk`. Statistiky doby běhu jsou ukládány do souboru `Times.csv` jako mezičasy po dokončení fází algoritmu. Běh programu je zachycen na obrázku A.1.

¹²Obsah DVD je popsán v souboru `Readme.txt` v kořenovém adresáři disku

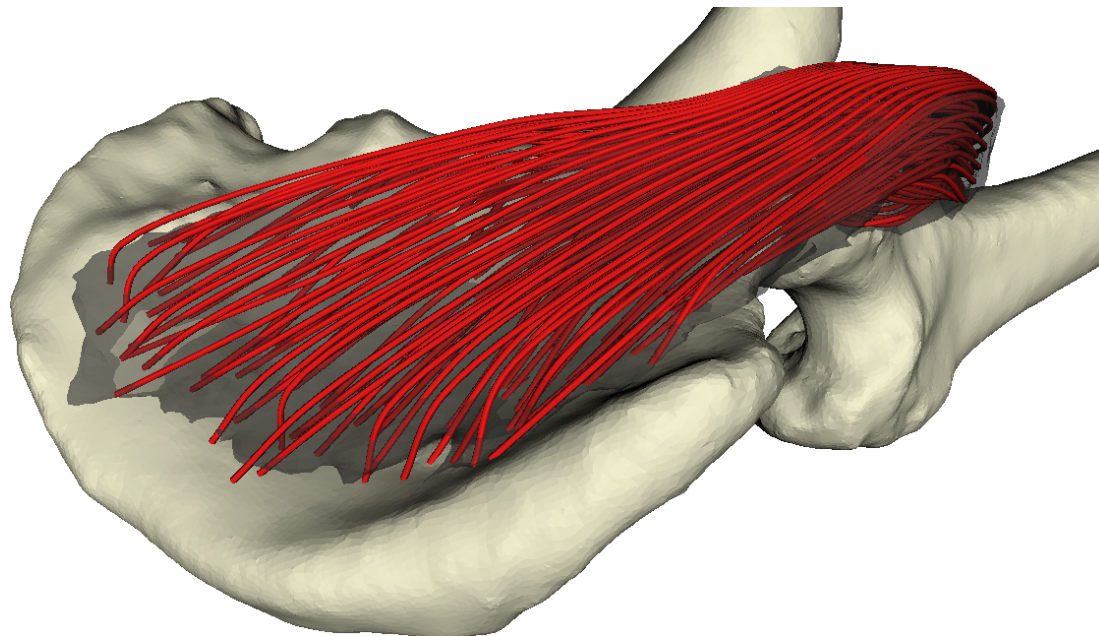
Po zobrazení okna s vizualizací dat je možné modelem otáčet, přibližovat jej a posouvat myší. Ovládání je závislé na použité verzi knihovny VTK, pro doporučenou verzi VTK 4.2 je však následující:

- Levé tlačítko myši + tažení myši - trackballové otáčení zobrazenými modely
- Kolečko myši nebo klávesa shift + tažení myši - posun zobrazených modelů
- Levé tlačítko + tažení myši - přiblížení a oddálení zobrazení

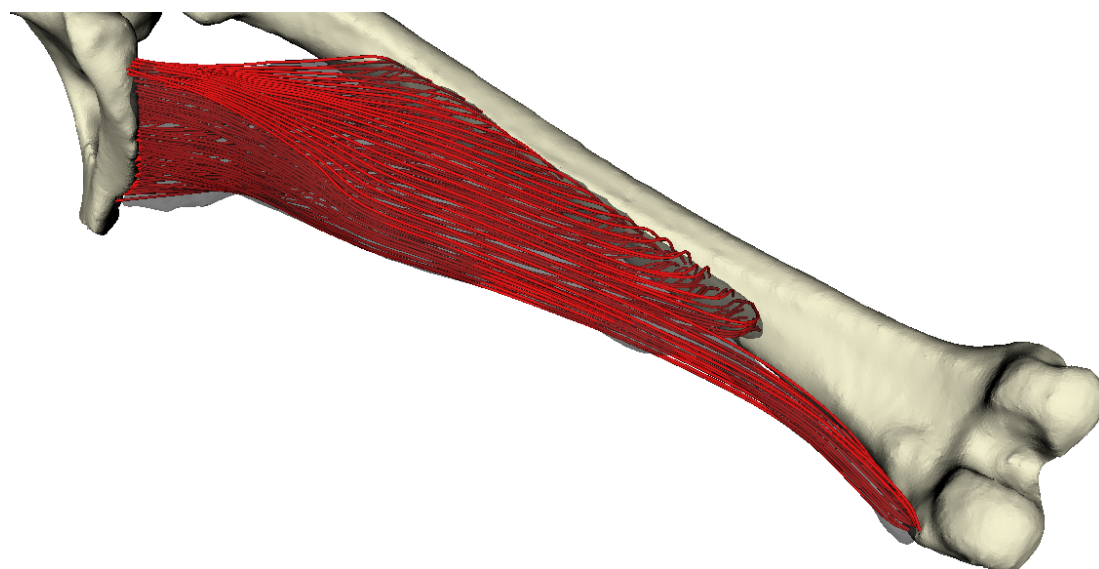


Obrázek A.1: Běh programu

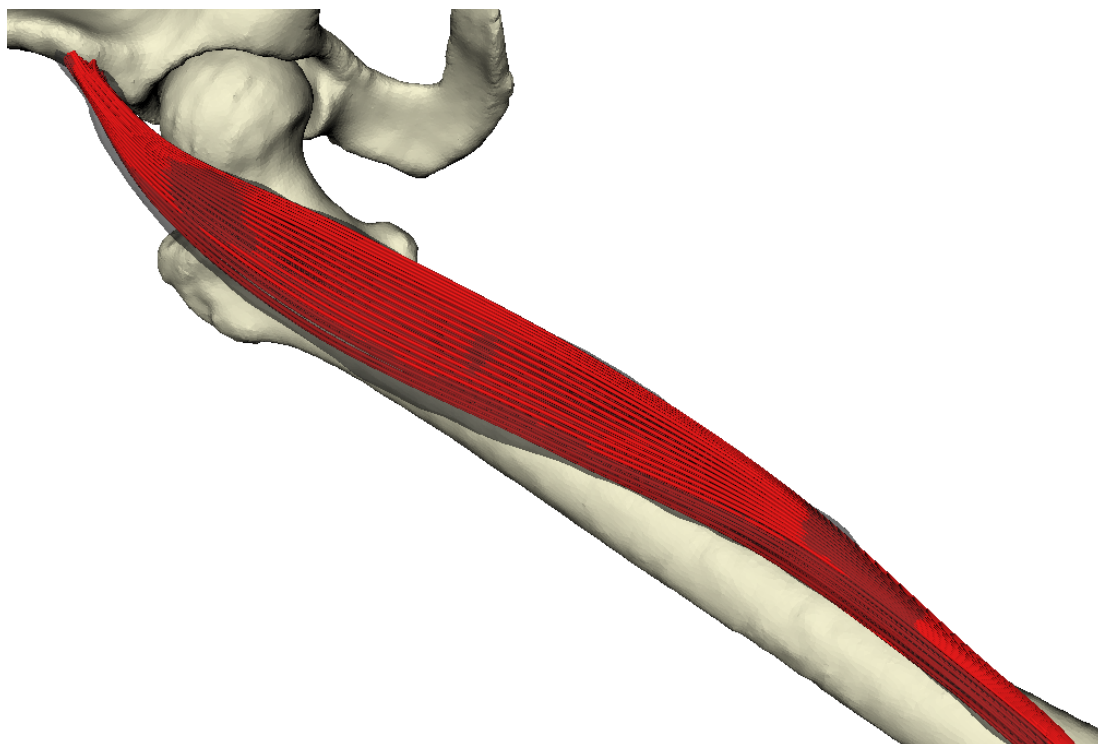
B Obrázky a tabulky



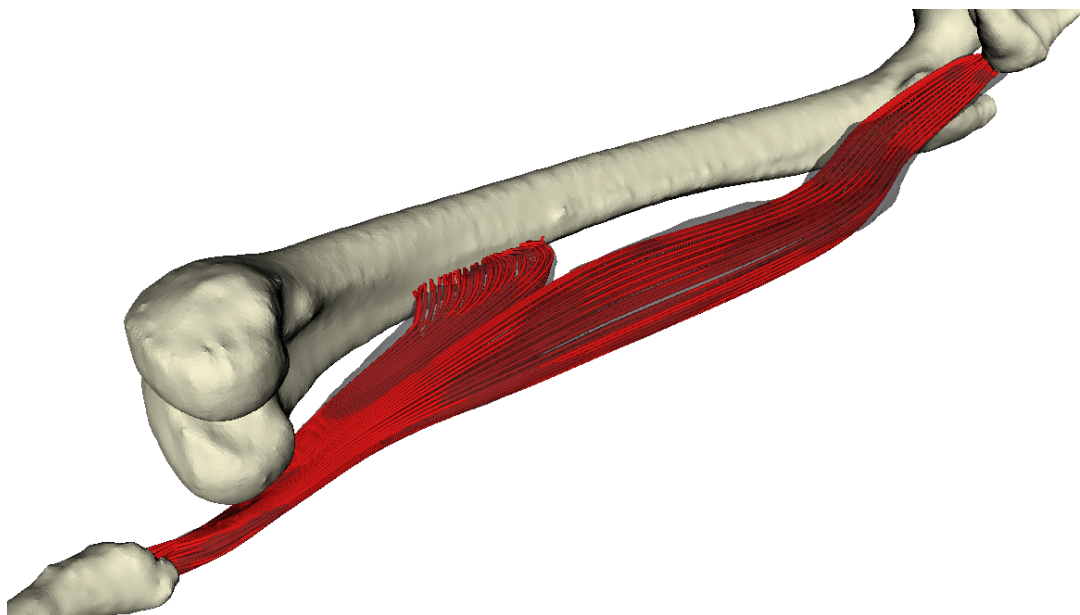
Obrázek B.1: Dekompozice svalu *Iliacus*. Tento sval se vyznačuje ostrým pennate úhlem v oblasti pánve.



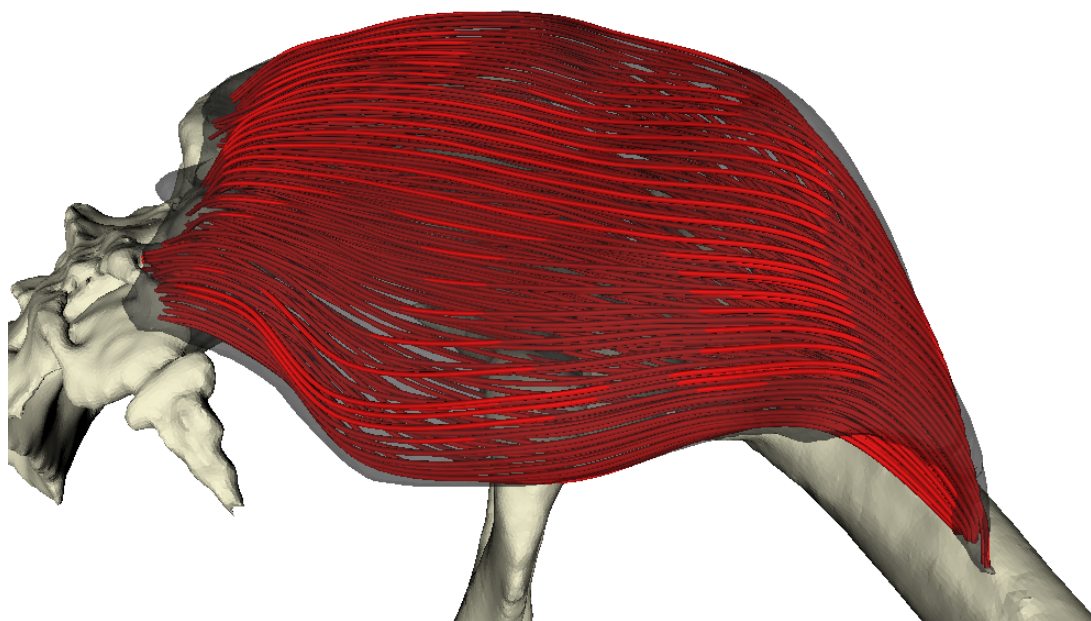
Obrázek B.2: Dekompozice svalu *Adductor Magnus*. Tento sval je na konci rozštěpen



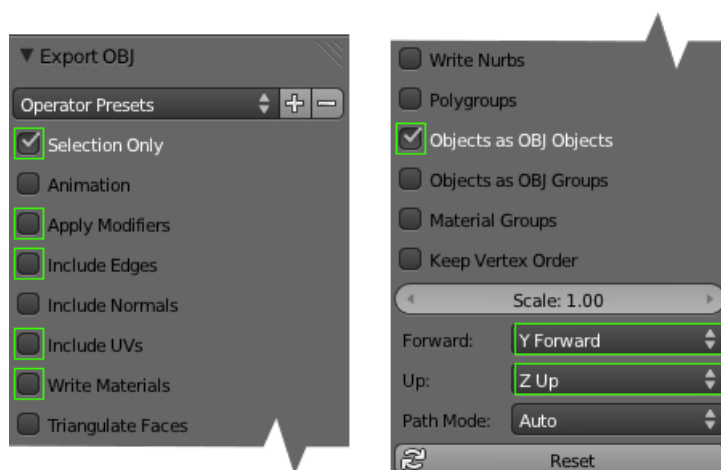
Obrázek B.3: Dekompozice svalu *Rectus Femoris*. Ve skutečnosti je tento sval typu Bipennate, nepodařilo se mi však taková řídicí vlákna vytvořit



Obrázek B.4: Dekompozice svalu *Biceps Femoris*. Sval je dvojhlavý, upíná se ke třem kostem a ohýbá se okolo kosti *Femur*.



Obrázek B.5: Dekompozice svalu *Gluteus Maximus*. Sval je dvojhlavý, upíná se ke třem kostem.



Obrázek B.6: Nastavení exportu OBJ souboru v programu Blender

Počet Δ modelu svalu	4812	10812	15564	21180	24312	30012	35004
Mezičas po přípravě oblastí [ms]	0,67	1,00	1,00	1,00	1,00	0,67	0,67
Mezičas po dekomp. povrchu [ms]	2,00	2,00	2,00	2,00	2,00	1,67	2,00
Mezičas po výp. normál [ms]	3,00	3,00	3,00	3,00	3,00	2,67	3,00
Mezičas po vzorkování [ms]	213,67	399,00	490,00	628,33	753,33	868,33	1006,33
Mezičas po dekomp. objemu [ms]	216,00	401,33	492,67	630,33	755,67	871,67	1010,00
Mezičas po úponu vláken [ms]	229,33	415,00	506,67	643,67	769,33	871,67	1023,00
Celkový čas [ms]	229,33	415,00	506,67	643,67	769,33	885,33	1040,67

Tabulka B.1: Závislost času dekompozice na počtu trojúhelníků dekomponovaného svalu

Počet vláken	10	20	30	40	50
Mezičas po přípravě oblastí [s]	0,00	0,00	0,00	0,00	0,00
Mezičas po dekomp. povrchu [s]	0,01	0,01	0,01	0,02	0,02
Mezičas po výp. normál [s]	0,01	0,01	0,02	0,03	0,04
Mezičas po vzorkování [s]	2,76	2,96	4,14	5,57	7,31
Mezičas po dekomp. objemu [s]	2,77	2,97	4,14	5,58	7,32
Mezičas po úponu vláken [s]	3,76	4,07	7,41	13,12	21,11
Celkový čas [s]	4,17	4,48	8,06	14,12	22,47

Tabulka B.2: Závislost času dekompozice na počtu řídicích vláken

Počet segmentů vláken	10	20	30	40	50
Mezičas po přípravě oblastí [ms]	1	1	0,67	1	0,67
Mezičas po dekomp. povrchu [ms]	2	3,33	5	5,67	6
Mezičas po výp. normál [ms]	3,67	5,67	8,67	10,33	12
Mezičas po vzorkování [ms]	606	1015,67	1411,33	1872,33	2407,67
Mezičas po dekomp. objemu [ms]	609	1020	1417	1880	2417,33
Mezičas po úponu vláken [ms]	632,67	1043,67	1440,33	1904,67	2440,67
Celkový čas [ms]	721	1208,33	1681	2229	2852,33

Tabulka B.3: Závislost času dekompozice na segmentů řídicích vláken

Hodnota InterpolationSubdivision	1	5	10	15	20	25	30	35	40	45	50
Mezičas po přípravě oblastí [ms]	1,00	1,00	1,00	1,00	1,00	1,00	1,33	1,33	1,00	1,00	1,00
Mezičas po decomp. povrchu [ms]	2,67	2,33	2,00	2,33	3,00	3,33	2,67	3,00	3,00	2,67	2,67
Mezičas po výp. normál [ms]	4,00	4,33	3,33	4,33	4,33	5,00	4,33	5,00	4,00	4,33	3,67
Mezičas po vzorkování [ms]	671,00	680,00	662,33	671,33	665,67	684,67	671,67	688,67	672,33	684,67	674,00
Mezičas po decomp. objemu [ms]	673,67	682,67	665,33	674,00	668,33	687,67	674,67	692,33	675,00	688,33	676,67
Mezičas po úponu vláken [ms]	702,33	708,33	691,33	700,00	693,33	714,33	701,67	719,00	702,33	713,67	700,67
Celkový čas [ms]	708,00	755,67	782,67	845,67	890,67	954,00	994,33	1051,67	1087,00	1156,00	1185,00

Tabulka B.4: Závislost času dekompozice na hodnotě parametru InterpolationSubdivision

Hodnota SurfaceSubdivision	1	5	10	15	20	25	30	35	40	45	50
Mezičas po přípravě oblastí [ms]	0	1	1	2	3	3	2	2	3	4	4
Mezičas po decomp. povrchu [ms]	1	3	4	7	8	9	12	12	15	18	18
Mezičas po výp. normál [ms]	1	5	6	10	13	17	20	20	24	28	29
Mezičas po vzorkování [ms]	286	711	1123	1623	2111	2645	3067	3548	4132	4656	4998
Mezičas po decomp. objemu [ms]	288	714	1128	1628	2116	2654	3076	3557	4141	4667	5009
Mezičas po úponu vláken [ms]	290	741	1268	2030	2985	4355	6044	7895	10644	14003	16699
Celkový čas [ms]	320	829	1448	2294	3333	4839	6618	8618	11460	14949	17783

Tabulka B.5: Závislost času dekompozice na hodnotě parametru SurfaceSubdivision

Hodnota VolumeSubdivision	1	5	10	15	20	25	30	35	40	45	50
Mezičas po přípravě oblastí [ms]	0	1	1	1	2	2	2	2	3	3	4
Mezičas po decomp. povrchu [ms]	2	2	2	3	3	3	4	4	4	4	6
Mezičas po výp. normál [ms]	3	3	4	4	4	5	5	5	5	6	7
Mezičas po vzorkování [ms]	602	595	679	600	623	606	597	597	613	614	600
Mezičas po decomp. objemu [ms]	603	598	684	609	633	618	611	613	632	635	623
Mezičas po úponu vláken [ms]	605	623	781	808	1004	1188	1600	2261	2875	3708	4823
Celkový čas [ms]	633	711	965	1064	1332	1617	2136	2878	3585	4574	5773

Tabulka B.6: Závislost času dekompozice na hodnotě parametru VolumeSubdivision