

Titulní strana

Rád bych poděkoval svému vedoucímu práce Mgr. Václavu Krajíčkovi, který mi umožnil pracovat na tomto tématu a který mi byl ochoten kdykoliv poradit jak při vývoji programu, tak při psaní tohoto textu.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne:

Jan Bím

Obsah

1. Úvod	6
2. Problematika objemových dat a haptické odezvy	8
2.1 Objemová data.....	8
2.1.1 Úvod do objemových dat a jejich využití.....	8
2.1.2 Zobrazování objemových dat	8
2.1.3 Objemová data získaná z vyšetření CT.....	10
2.2 Haptické zařízení.....	11
2.2.1 Úvod do haptických zařízení a jejich použití.....	11
2.2.2 Stručný úvod do generování haptické odezvy.....	11
2.3 Existující hardware a software	12
2.3.1 Hardware.....	12
2.3.2 Knihovny.....	12
2.4 Metody generování haptické odezvy.....	13
2.4.1 Generování haptické odezvy v geometrických datech.....	13
2.4.2 Generování haptické odezvy v objemových datech.....	14
2.4.3 Generování haptické odezvy v CT datech.....	15
3. Implementace	16
3.1 Použité knihovny	16
3.1.1 MedV4D framework.....	16
3.1.2 VTK.....	16
3.1.3 Chai3D	17
3.1.4 Qt.....	17
3.1.5 Boost	17
3.2 Struktura aplikace.....	17
3.2.1 Zobrazení objemových dat	18
3.2.2 Haptický kurzor.....	18
3.2.3 Rozhraní pro nastavení aplikace	19
3.3 Předzpracování dat	19
3.4 Implementační detaily.....	20
3.4.1 Data.....	20

3.4.2	Zobrazení.....	21
3.4.3	Přechodová funkce.....	24
3.4.4	Práce s haptickým zařízením.....	24
3.4.5	Rozhraní pro nastavení aplikace	29
3.5	Překlad.....	30
4.	Výsledky.....	32
4.1	Příprava dat.....	32
4.2	Funkce aplikace.....	33
4.3	Srovnání polohovacích zařízení.....	35
4.4	Konfigurace přechodové funkce a její vliv na haptickou odezvu .	38
4.5	Sledování hrany v objemových datech.....	40
5.	Závěr	41
	Použitá literatura	43
	Dodatek A – obsah příloženého média.....	44
	Dodatek B – uživatelská dokumentace	45
	Úvod.....	45
	Spuštění.....	45
	Předzpracování dat.....	46
	Zobrazení objemových dat	46
	Okno s ovládacími prvky aplikace.....	47

Název práce: Uživatelské rozhraní pro haptickou interakci s objemovými daty

Autor: Jan Bím

Katedra (ústav): Kabinet software a výuky informatiky

Vedoucí bakalářské práce: Mgr. Krajíček Václav

e-mail vedoucího: Vaclav.Krajicek@mff.cuni.cz

Abstrakt: Hlavním cílem této práce je vytvořit program, který by umožňoval průchod a interakci s objemovými daty za pomoci haptického zařízení. Práce obsahuje popis vlastností objemových dat spolu s možnostmi využití. Rovněž jsou zde uvedeny některé metody, které slouží k jejich vizualizaci. Hlavním těžištěm práce je vysvětlení různých způsobů generování haptické odezvy. Na tuto část navazuje popis implementace zvoleného algoritmu, který se jeví jako nejvhodnější pro práci s objemovými daty. Dále se zde zabýváme testováním vytvořeného software na reálných i syntetických datech, částečně formou uživatelské studie. Tyto experimenty jsou navrženy za účelem ověření, jakou měrou je haptická odezva při interakci s objemovými daty přínosem pro rychlost či přesnost vykonání daných úloh. Výsledky těchto testů jsou v závěru podrobeny diskusi.

Klíčová slova: objemová data, zobrazení objemových dat, haptické zařízení

Title: User interface for haptic interaction with volumetric data

Author: Jan Bím

Department: Department of Software and Computer Science Education

Supervisor: Mgr. Krajíček Václav

Supervisor's e-mail address: Vaclav.Krajicek@mff.cuni.cz

Abstract: The primary objective of this work is to develop the application which allows throughput of volumetric data and interaction with them by a haptic device. The work contains examples of use of volumetric data and description of their properties. Also there are some methods for volume rendering. The explanation of different algorithms for the haptic feedback generation is the key point of this work. The thesis continues by the description of the implementation of the algorithm which seems the most suitable for the main goal of this project. Next there are noted some tests of the application which should prove whether it is an improvement of speed or accuracy to use the haptic device with volumetric data or not. The results of these tests are discussed in the end of the thesis.

Keywords: volumetric data, visualization of volumetric data, haptic device

1. Úvod

Neustále stoupající rychlost počítačů nyní již umožňuje realizovat real-time zobrazení objemových dat. A jedním z nejznámějších zástupců těchto dat jsou data lékařská, která byla vyprodukována pomocí CT (computed tomography) nebo MRI (magnetic resonance imaging). Lékaři a radiologičtí pracovníci v dnešní době již nemusí pracovat s velmi velkými plochami pokrytými 2D obrázky, ale mohou pracovat relativně kvalitními plnohodnotnými 3D modely naskenovaných objektů.

Ruku v ruce s vývojem počítačů kráčí mílovými kroky též vývoj jejich periférií. Jedním druhem periférií, které jsou dnes na vynikající úrovni, jsou haptická zařízení. Právě přítomnost haptické odezvy, která by doplnila 3D zobrazení lékařských dat, by poskytla pocit přítomnosti a cítění dat a umožnila, aby lékařské analýzy probíhaly rychleji a s mnohem větší přesností, protože by lékaři neměli jen obecné povědomí o datech, které jim poskytuje zobrazení, ale též by byli schopni mnohem přesněji vnímat lokální vlastnosti malých oblastí. Takto by bylo virtuální vyšetření opět o krok blíže vyšetření reálnému.

Tato práce by měla přinášet právě toto spojení. Cílem práce je poskytovat haptickou odezvu nad lékařskými daty, která byla získána pomocí CT. Odezva by měla vylepšovat vnímání lékařských dat a umožňovat snazší a přesnější orientaci v datech. Zároveň by tato práce měla vyústit nejen v samostatnou aplikaci, ale měla by umožnit používání haptické odezvy nad objemovými daty jakékoliv aplikace vyvíjené pod frameworkem MedV4D [1], skrz integraci do tohoto projektu.

Výsledkem této práce je nejen samotná implementace aplikace, která poskytuje jak zobrazení objemových dat, tak zprostředkování haptické odezvy, ale též provedení testů, zda je skutečně haptická odezva v tomto případě přínosem, kde nabízí uživateli něco navíc a kde je naopak pouze práce s haptickým zařízením zbytečná.

V kapitole 2 je rozvedena problematika objemových dat a haptických zařízení, jsou zde vysvětleny základy zobrazovacích technik pro objemová data a též základy generování haptické odezvy. Ke konci kapitoly jsou zmíněny některé vlastnosti dat, která byla pořízena pomocí CT. V kapitole 3 je popis vlastní implementace, který se věnuje rozdělení aplikace, implementaci důležitých částí a podrobně hlavně generování haptické odezvy. V kapitole 4 jsou prezentovány výsledky práce. Především jsou to ukázky vlastností aplikace a demonstrace výsledků provedených testů. V kapitole 5 je diskutováno provedení práce a směr, kterým by se tento projekt měl ubírat do budoucnosti. Tyto kapitoly jsou následovány přehledem použité literatury. Za tímto přehledem je umístěn dodatek A, ve kterém lze nalézt obsah příloženého média. V dodatku B je uživatelská dokumentace, která obsahuje návod jak spustit program, zadat mu vstupní

data a jak tato data vytvořit. Dále je zde popis grafického rozhraní aplikace, vysvětleny všechny její funkce a návod pro používání aplikace.

2. Problematika objemových dat a haptické odezvy

Tato kapitola vysvětluje pojem objemová data a uvádí do problematiky haptických zařízení a jejich odezvy. Kapitola představuje některé ze základních metod pro zobrazování a základní metody pro generování haptické odezvy.

2.1 Objemová data

2.1.1 Úvod do objemových dat a jejich využití

Pro reprezentaci objektů v třídimenzionálním prostředí existují dva hlavní způsoby. Jeden z nich se nazývá vektorová, nebo též geometrická data a podstatou tohoto druhu dat je reprezentovat 3D scénu pomocí různých geometrických primitiv, jako jsou body, čáry, či nějaké plochy. Z těchto primitiv se poté skládají složitější třídimenzionální modely. Druhý způsob reprezentace 3D scény se nazývá objemová data a je předmětem následujícího textu. Tato práce využívá právě tohoto druhu dat a data geometrická jsou zmiňována pouze pro případná srovnání.

Objemová data reprezentují hodnoty nějaké vlastnosti v bodech v prostoru. Dobrý příklad takové vlastnosti je hustota prostředí daném bodě. Nejmenší částí objemových dat je voxel a jeho nejčastější podoba je podoba kvádrů. Voxel je analogický pixelu, který je znám z 2D grafiky. Většinou bývají voxely uspořádané v pravidelné třídimenzionální mřížce. Zde nabývají objemová data nepřesnosti, protože hodnota voxelu je integrována (průměrována) přes celý jeho objem, tudíž není známa hodnota v jednotlivých dílčích bodech voxelu.

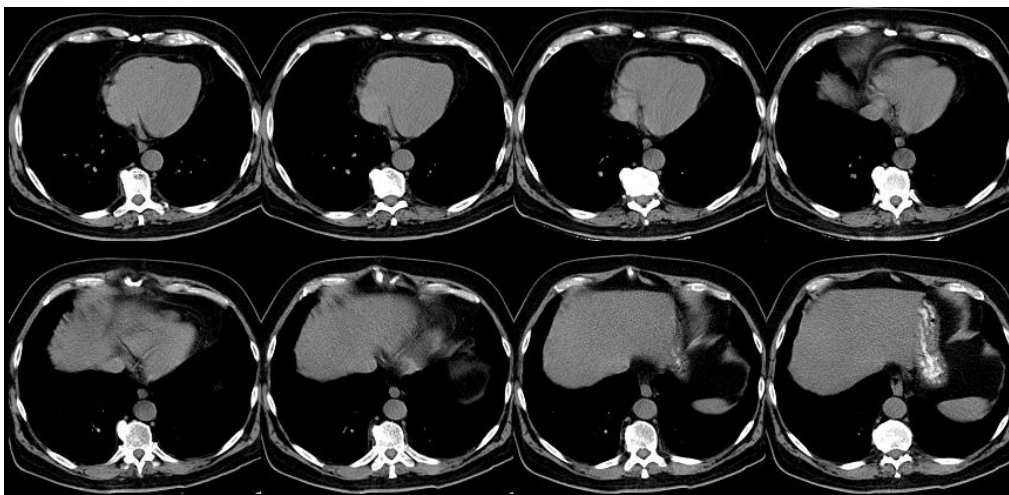
Objemová data mají mnoho uplatnění, v dnešní době se nejvíce používají v počítačových hrách a medicíně. Uplatnění v počítačových hrách nacházejí převážně pro modelování a zobrazování objektů, u nichž se předpokládá změna tvaru v průběhu hry. V medicíně se jich používá hlavně k uchovávání a zpracovávání naměřených dat z CT, MRI a speciálních ultrazvukových vyšetření, která jsou prováděna na 3D ultrazvukových přístrojích. Další z možných vyšetření jsou PET (Positron emission tomography) nebo SPECT (Single photon emission computed tomography).

2.1.2 Zobrazování objemových dat

Pro zobrazení objemových dat se používá řady metod. Pokud má být výstupem dvojdimenzionální zobrazení, pak se nejčastěji vykresluje jednotlivé řezy jako samostatné obrázky. Použití tohoto algoritmu je závislé na technologii, kterou byla pořízena objemová data. Tyto obrázky vzniknou

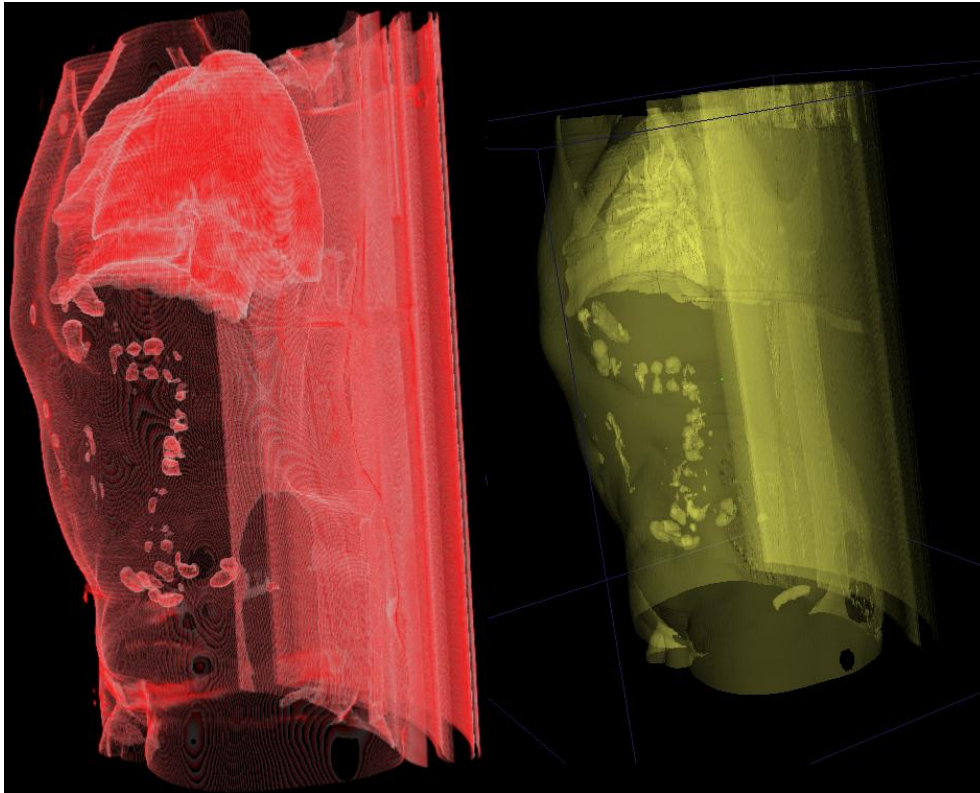
následujícím způsobem. Je zvolena jedna z os objemových dat jako ta, podle které se budou vykreslovat jednotlivé řezy. Pro každou souřadnici na této ose se vytvoří rovina kolmá na tuto osu a pro každou takovouto rovinu vznikne jeden obrázek, ve kterém každý pixel odpovídá jednomu voxelu, který tato rovina protнула. Nejčastěji se toto zobrazení používá právě v medicíně a to v černobílé škále viz obrázek 2-1.

Pro třídimenzionální zobrazení je více často používaných metod. Tyto metody se rozdělují do dvou kategorií. První z nich je nepřímé vykreslování objemových dat, které před samotným vykreslením převede data do geometrického formátu. Příklad takového algoritmu je algoritmus zvaný Marching Cubes [2]. Tento algoritmus vyhledává v objemových datech isoplochy, což je oblast v objemových datech, kde se nacházejí voxely se stejnou hodnotou. Jeho výstupem je polygonový model těchto isoploch, který je již snadný na vykreslení.



Obrázek 2-1: Zobrazení řezů objemových dat

Druhá kategorie se jmenuje přímé vykreslování objemových dat (DVR – direct volume rendering), a jak název napovídá, jedná se o metody, které pracují přímo s objemovými daty. Nejznámější touto metodou je Volume Ray Casting. Tato metoda poskytuje vysoce kvalitní výstup, ale je též velmi náročná na výpočet. Příklad zobrazení objemových dat touto metodou je na obrázku 2-2.



Obrázek 2-2: Zobrazení objemových dat pomocí metody Volume Ray Casting na levé straně a pomocí algoritmu marching cubes na pravé

2.1.3 Objemová data získaná z vyšetření CT

Tato práce byla zpracovávána pro používání s lékařskými objemovými daty a to konkrétně s daty naměřenými při vyšetření CT. V těchto datech odpovídají hodnoty počtu Haunsfieldových jednotek, které udávají propustnost materiálu pro rentgenové záření.

Tato data jsou měřena CT skenery po jednotlivých řezech. Nejčastěji má jeden řez rozlišení 512x512 voxelů. Tím pádem nejběžnější tvar voxelu, který je naměřen takovým přístrojem odpovídá tvaru kvádru. Rozměry tohoto kvádru jsou v poměru 1:1:X. Toto X bývá zpravidla podstatně větší než jedna. Tato délka odpovídá délce posunu zařízení po jednotlivých řezech. Tento jev znesnadňuje následnou práci s daty, protože jednotlivé voxely mají tvar kvádru, kde se jeden jeho rozměr může lišit často až desetkrát oproti ostatním dvěma jeho rozměrům. Tento jev má za následek zvýšení náročnosti jakéhokoliv použití těchto pro prostorové výpočty a zobrazení.

2.2 Haptické zařízení

2.2.1 Úvod do haptických zařízení a jejich použití

Tato zařízení přináší možnost interakce s uživatelem pomocí jiného smyslu než je zrak a sluch, které jsou v dnešní době nejrozšířenější, a tímto smyslem je hmat. Tato zařízení dokážou působit silou na uživatele v různých osách. Haptická zařízení se rozdělují podle stupňů volnosti, které vyjadřují v kolika osách je zařízení schopno snímat pohyb a v kolika z nich je schopno vyvinout nějakou sílu a tím poskytovat haptickou odezvu. První tři stupně volnosti zpravidla určují v kolika dimenzích prostoru je možno s haptickým zařízením pohybovat a následující tři stupně volnosti zpravidla určují v kolika osách je možno otáčet nástroj na konci ramene haptického zařízení. Další stupně volnosti nejčastěji reprezentují nějaký možný pohyb specifický pro daný nástroj. Jako příklad lze uvést nůžky, které mají možnost různého rozevření. Zde však třeba rozlišovat anglická slova haptic a tactile, protože obě se dají do českého jazyka přeložit jako hmatový. Odezva haptických zařízení dávají odezvu ve formě úhrnné síly, kdežto taktilní zařízení dávají odezvu, která simuluje vnímání podobné tomu na bráškách prstů. Například přímou texturu povrchu materiálu.

Haptická zařízení mají mnoho použití v různých oborech. V letectví se například používají k poskytování odezvy ovládacích prvků letadla, která se vytratila se začátkem používání hydraulického ovládání. Též se používá těchto zařízení pro výcvik pilotů na simulátorech. Dále se používá haptických zařízení při dálkovém ovládní různých strojů. I v herním průmyslu se začala používat haptická zařízení, i když zde je haptická odezva teprve v počátcích. Jedno z nejdůležitějších použití nalézají haptická zařízení v medicíně. Používají se pro nácvik chirurgických zákroků a při vykonávání operací pomocí vzdáleně řízených robotů.

2.2.2 Stručný úvod do generování haptické odezvy

Vytváření haptické odezvy pomocí počítače, neboli haptic rendering, odpovídá procesu, v němž jsou požadované smyslové podněty přenášeny na uživatele, za účelem zprostředkování informací o virtuálním haptickém objektu, neboli objektu který má definovány některé z haptických vlastností jako jsou tvar, tvrdost, přilnavost, struktura povrchu a další.

Haptická odezva se nejčastěji používá ve spojení s geometrickými daty. Generování haptické odezvy je založeno na detekci kolize s daným objektem. Základní haptické algoritmy reagují pouze na geometrický tvar, ale existuje též mnoho algoritmů generujících haptickou odezvu i podle tření daného objektu a jeho struktury. Většina těchto algoritmů je však kompromisem mezi přesností a rychlostí, protože jsou používány v real-time aplikacích.

2.3 Existující hardware a software

2.3.1 Hardware

Na dnešním trhu (rok 2010) s haptickými zařízeními jich dnes existuje celá řada, jež se liší svou přesností a počtem stupňů volnosti. Jako příklad lze uvést zařízení omega.6 vyráběné firmou Force Dimension [3]. Jedná se o zařízení s šesti stupni volnosti, z nichž tři jsou aktivní a tři pasivní a zařízení též disponuje systémem pro kompenzaci gravitační síly. Jedná se o zástupce vyšší třídy zařízení.

Zařízení, pro které a se kterým byla vyvíjena tato práce se jmenuje Novint Falcon a vyrábí jej firma Novint Technologies, Inc. [4]. Toto zařízení bylo vyvinuto jako haptické zařízení zejména pro herní průmysl. Je to jediné haptické zařízení v dnešní době, které je směřované právě na herní trh. Díky tomuto zaměření je zařízení velmi snadno dostupné, ale zároveň nedosahuje takové přesnosti jako zařízení konkurenční. Zařízení disponuje třemi aktivními stupni volnosti a zobrazuje jej obrázek 2-3.

Pro snazší vytváření aplikací, které jsou určeny pro práci s haptickým zařízením vzniklo též virtuální haptické zařízení. Exemplář tohoto lze nalézt například v knihovně CHAI 3D. Toto zařízení má tři stupně volnosti a jeho pozici lze zadat pomocí souřadnic, nebo pohybem myši. Konkrétně toto zařízení vykazuje někdy chybné chování, protože nevrací správný rozsah svého pohybu. Též jeho obnovovací frekvence se pohybuje zhruba na 60-80 Hz, což pro některé aplikace nemusí být dostačující.

2.3.2 Knihovny

Pro práci s haptickými zařízeními existuje několik knihoven. Knihovny, které lze používat se zařízením Novint Falcon jsou Novint Falcon SDK [4], CHAI 3D [5] a H3DAPI [6]. Novint Falcon SDK je knihovna, která podporuje práci pouze se zařízením Novint Falcon. Její přístup je nízkoúrovňový. Knihovny CHAI 3D a H3DAPI jsou obě velmi podobné v několika rysech. Obě dvě obsahují komplexní framework nejen pro práci s haptickým zařízením, ale též pro vytváření simulací a zobrazování 3D grafiky. Obě tyto knihovny poskytují podporu několika haptických zařízení.

H3DAPI je velmi obsáhlý projekt, který v dnešní době obsahuje spoustu podprojektů. Jedním z těchto podprojektů je Volume Haptic Toolkit, který umožňuje použití haptických zařízení nad objemovými daty. H3DAPI obsahuje též část zvanou MedX3D, která umožňuje práci s lékařskými daty a hlavně jejich zobrazování.

Pro tuto práci byla vybrána knihovna CHAI 3D, které poskytuje velmi kvalitní vysokoúrovňový přístup ke komunikaci s haptickým

zařizním. Obsahuje dokonce nástroje pro modelování těles s haptickými vlastnostmi a vytváření třídimenzionálních scén z takovýchto těles.

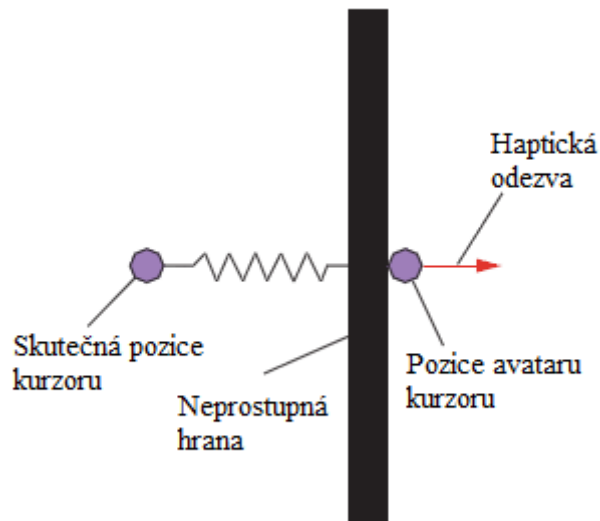


Obrázek 2-3: Zařízení Novint Falcon

2.4 Metody generování haptické odezvy

2.4.1 Generování haptické odezvy v geometrických datech

Nejčastěji používaným algoritmem pro generování haptické odezvy je algoritmus založený na proxy objektu [7]. Proxy objekt reprezentuje bod, který reaguje na haptické vlastnosti prostředí. Mezi skutečným nástrojem a proxyobjektem je natažena pružina o nějaké síle. Při pohybu ve volném prostoru je vždy proxy objekt ve shodném místě jako skutečné zařízení. V momentě kolize však nástroj pokračuje dále, ale proxy objekt zůstává na hranici tělesa, do kterého narazil, nebo je nějakým způsobem upravena jeho rychlost, či směr pohybu. V tomto momentě se začíná natahovat pružina mezi proxyobjektem a skutečným nástrojem a haptická odezva generovaná tímto modelem odpovídá směru a velikosti síly, kterou působí pružina na skutečné zařízení.



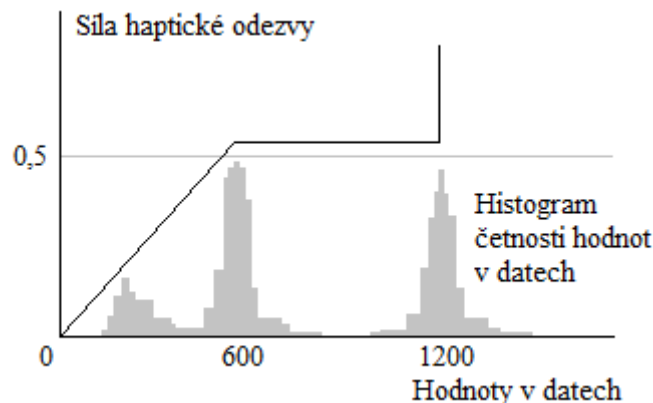
Obrázek 2-4: Grafické znázornění funkce algoritmu založeném na použití proxy objektu při naražení do nepropustné hrany

2.4.2 Generování haptické odezvy v objemových datech

Pro výpočet haptické odezvy v objemových datech je možno využít stejného principu práce s proxyobjektem [8] [9]. Rozdíl spočívá v tom, že v objemových datech nejsou přesně dány hranice jednotlivých objektů. Omezení pohybu proxyobjektu se v případě objemových dat realizuje přechodovou funkcí, která určuje zpravidla maximální rychlost pohybu proxyobjektu na dané pozici v datech.

Jiná metoda spočívá v generování haptické odezvy pouze z aktuální pozice zařízení a gradientu hodnoty objemových dat v daném místě. Obvykle se jako směr i velikost haptické odezvy dosadí právě tento gradient.

Ať už je velikost haptické odezvy počítána přímo z hodnot v datech, nebo je tohoto dosaženo za pomoci gradientu, tak je většinou využita přechodová funkce, která přiřazuje jednotlivým hodnotám v datech různou sílu haptické odezvy. Příklad přechodové funkce je zobrazen na obrázku 2-5.



Obrázek 2-5: Znázornění příkladu přechodové funkce, která od hodnoty v datech 0 až do hodnoty 600 přiřazuje lineárně rostoucí sílu haptické odezvy. Pro hodnoty 600 až 1200 je haptická odezva stále stejně silná a pro hodnoty větší než 1200 skokově roste. Funkce se mění při hodnotách, které se v datech často vyskytují.

2.4.3 Generování haptické odezvy v CT datech

Generování haptické odezvy v těchto datech je stejné jako generování odezvy v jakýchkoliv jiných objemových datech s tím rozdílem, že musí řešit dva problémové body, které vznikají právě kvůli použití dat z vyšetření CT.

Prvním z nich je již zmiňovaný tvar voxelu. Voxely mají tvar kvádrů a tím je způsobeno, že jsou mezi nimi ostré přechody. Algoritmus pro generování haptické odezvy je tudíž třeba upravit tak, aby i při dostatečném přiblížení zůstala změna síly plynulá, přestože změna hodnot v prostředí, které bude procházeno, bude skoková. Pokud se algoritmus neupraví, bude docházet k nežádoucím projevům haptického zařízení, jako jsou vibrace a podobné. V případě velké skokové změny síly, které zařízení vyvíjí na člověka, dochází k prudkému pohybu připomínajícímu kopnutí a člověk není schopen na tuto změnu dostatečně rychle reagovat a zařízení se může dostat daleko od své původní pozice, čímž je znehodnocena kvalita haptické odezvy a nabeurána funkce polohovacího zařízení.

Druhým problémem je výskyt šumu v datech. Zvláště snímky z CT skenerů často trpí velkým výskytem šumu. Tyto artefakty by mohly způsobovat zdánlivý výskyt tvrdých částic uprostřed měkkých tkání a podobně. Je třeba vhodně upravit algoritmus pro generování haptické odezvy nebo provést nějaké filtrování snímků před tím, než se z nich bude odezva generovat.

3. Implementace

3.1 Použité knihovny

Knihovny, které byly použity v tomto projektu, jsou popsány v následující kapitole. Kromě verze a stručného popisu funkcí knihovny je zde stručně uvedeno využití dané knihovny v aplikaci HapticInterface.

3.1.1 MedV4D framework

Tento framework poskytuje nástroje pro práci s lékařskými daty. Jedná se hlavně o sadu různých algoritmů pro zpracování lékařských dat a následné numerické a statistické analýzy, zobrazovací metody jak pro 2D tak 3D zobrazení lékařských dat implementované pro výpočet na GPU, interaktivní a přívětivé uživatelské rozhraní a nástroje pro práci se standardy, které jsou běžné v medicínské praxi jako je například DICOM nebo PACS.

Celá aplikace byla navržena a optimalizována právě pro budoucí integraci do projektu MedV4D framework. Jedna z hlavních tříd tohoto frameworku se stala kostrou aplikace a byla pouze dále upravována, přičemž zachovala rozhraní které jí MedV4D framework poskytuje. Pomocí tohoto rozhraní se například načítají vstupní data pro aplikaci a je do budoucna zajištěna komunikace s uživatelským rozhraním, které framework nabízí.

3.1.2 VTK

Verze: 5.0.3

VTK neboli „Visualisation Toolkit“ je knihovna určená pro práci s 3D počítačovou grafikou, zpracování obrázků a zobrazování. Knihovna VTK nabízí širokou škálu algoritmů pro libovolnou práci s daty, jako jsou metody pro skalární či vektorová data, textury nebo objemová data. VTK též nabízí mnoho modelovacích technik, jako jsou například implicitní modelování, ořezávání, vyhlazování, nebo Delaunayova triangulace.

VTK je v aplikaci použito pro zobrazení dat a práci s nimi. Jak 3D zobrazení, tak zobrazení jednotlivých řezů dat je implementováno přímo nástroji knihovny VTK. V aplikaci jsou hojně využity metody pro modelování objektů, díky kterým je reprezentován avatar kurzoru a hranice oblasti, ve které se kurzor může pohybovat. Dále jsou zde použity filtry pro předzpracování dat a algoritmus Marching cubes, který v objemových datech nalezne isoplochy vhodné pro třídimenzionální zobrazení. A v posledním případě je VTK použito na zobrazení řezu přímo z objemových dat.

3.1.3 Chai3D

Verze: 2.0.0

Nejdůležitějším, co zprostředkovává tato knihovna, je práce s haptickými zařízeními. Je zde implementována podpora pro nejčastěji se objevující zařízení s třemi, šesti a sedmi stupni volnosti. Dále jsou v této knihovně obsaženy nástroje pro zobrazování a interaktivní simulaci.

Tato knihovna je použita pouze pro komunikaci s haptickým zařízením. Využity jsou metody pro identifikaci haptického zařízení, získání různých informací o daném zařízení, získání pozice, na které se právě zařízení nachází a pro nastavení haptické odezvy.

3.1.4 Qt

Verze: 4.3.3

Knihovna Qt je univerzální sadou nástrojů pro tvorbu libovolných aplikací a uživatelského rozhraní nezávisle na cílové platformě. Obsahuje velmi velké množství různých metod pro tvorbu uživatelského rozhraní, práci s 3D i 2D grafikou, práci s vlákny, práci s multimédií, metody pro využívání XML a různých databází a též metody pro práci se sítí.

Tato knihovna je využita pro tvorbu uživatelského rozhraní, které je realizováno pomocí formuláře a oblastí pro vykreslování přechodové funkce. Do této oblasti je kresleno právě pomocí metod knihovny Qt na práci s 2D grafikou a též je zde použito několik metod pro interakci s uživatelem pomocí myši. Dále jsou využity základní metody knihovny Qt pro spuštění aplikace, zobrazení okna a komunikace mezi jednotlivými částmi aplikace. Tato komunikace je realizována pomocí systému signálů a slotů, který tato knihovna obsahuje.

3.1.5 Boost

Verze: 1.38

Knihovna Boost je rozšířením knihovny STL (Standard Template Library) pro programovací jazyk C++. Obsahuje velké množství funkcí pro práci se systémovými nástroji. Například to jsou nástroje pro práci se souborovým systémem, časem, vlákny, sítí a správou paměti.

Knihovna Boost je využita pro tvorbu vlákn, ve kterém probíhá komunikace s haptickým zařízením. Dále jsou použity nástroje pro řízení více vláknové aplikace.

3.2 Struktura aplikace

Aplikace se skládá z těchto logických celků: Zobrazení objemových dat, Haptický kurzor a Rozhraní pro nastavení aplikace. Tyto logické celky jsou

sestavěny do jednoho spustitelného souboru. Pro implementaci byl použit programovací jazyk C++. Všechny uváděné soubory zdrojového kódu jsou uvedeny bez přípon a všechny jsou složeny ze dvou souborů, jeden z příponou .cpp a druhý hlavičkový s příponou .h.

3.2.1 Zobrazení objemových dat

Třídy, ve kterých je tato část implementována:

- aggregationFilterForVTK
- m4dGUIOGLHapticViewerWidget
- ViewerWindow

Tento modul je hlavní kostrou celé aplikace. Je vlastníkem ostatních částí a zprostředkovává jejich komunikaci. Její samotná úloha spočívá v třech základních úkolech. Tyto úkoly jsou: načítání a předzpracování dat, zobrazení 3D modelu objemových dat a zobrazení objemových dat ve formě řezů. Využívá funkcí z knihovny Qt [10], MedV4D framework a knihovny VTK [11].

3.2.2 Haptický kurzor

Třídy, ve kterých je tato část implementována:

- cursorInterface
- hapticCursor
- transitionFunction

Jak napovídá název této části aplikace, tak jejím hlavním úkolem je uchovávat informace o kurzoru a zprostředkovávat haptickou odezvu uživateli. Implementuje funkce pro zjištění aktuální pozice kurzoru a zjištění pozice a rozměrů oblasti, ve které se může kurzor pohybovat. Implementace je provedena pro různá haptická zařízení, se kterými umí komunikovat knihovna CHAI 3D. Testována je však jen pro zařízení Novint Falcon, protože jiné nebylo v době vývoje k dispozici. Pro některé části správy paměti a pro práci s vlákny je použita knihovna Boost [12].

Tato část aplikace by měla být použitelná i v jiných projektech než je tento. Třída hapticCursor vyžaduje pro své vytvoření pouze ukazatel na data, která jsou reprezentována třídou vtkImageData, ukazatel na vtkRenderWindow a ukazatel na přechodovou funkci, kterou zastupuje třída transitionFunction.

3.2.3 Rozhraní pro nastavení aplikace

Třídy, ve kterých je tato část implementována:

- settingsBoxWidget
- transitionFunctionRenderArea

Poslední částí aplikace je grafické rozhraní pro nastavování všech nastavitelných parametrů aplikace. Majoritní částí tohoto rozhraní je editor přechodové funkce. Dalšími prvky lze nastavovat zvětšení a zmenšení oblasti, ve které se pohybuje kurzor, navrácení kurzoru do původní pozice a zapnutí či vypnutí záznamu pozic kurzoru. Celé rozhraní je implementováno pomocí knihovny Qt.

3.3 Předzpracování dat

Z důvodu výskytu šumu v datech, pro která je aplikace implementována, se předpokládá předzpracování dat. Data by měla být přefiltrována pomocí mediánového filtru. Předzpracování dat není realizováno aplikací HapticInterface. Tato aplikace předpokládá, že toto předzpracování proběhne, a na vstupu pro tuto aplikaci již budou předzpracovaná data.

Mediánový filtr je velmi častá technika používaná při digitálním zpracování obrazu. Tento filtr je používán převážně k redukci šumu, neboť právě redukce šumu je častým případem předzpracování obrazu pro následné použití jiných technik, jako například pro hledání hran. Právě pro tento účel se mediánový filtr hodí nejvíce, protože zachovává hrany. Principem mediánového filtru je projít všechna data a do každého pole dat dosadit medián spočítaný v daném poli a jeho okolí. Velikost tohoto okolí je předem stanovena a závisí na ní výsledek tohoto filtru. Jeho funkci lze předvést na jednoduchém příkladě. Necht pole $x = [2, 80, 4, 3, 60, 1]$. Aplikací mediánového filtru (velikost okolí necht je rovna 3) na toto získáme pole y , které má obsahovat hodnoty po aplikaci filtru, následovně:

```
y[1] = medián(2, 2, 80) = 2
y[2] = medián(2, 80, 4) = medián(2, 4, 80) = 4
y[3] = medián(80, 4, 3) = medián(3, 4, 80) = 4
y[4] = medián(4, 3, 60) = medián(3, 4, 60) = 4
y[5] = medián(3, 60, 1) = medián(1, 3, 60) = 3
y[6] = medián(60, 1, 1) = medián(1, 1, 60) = 1
y = [ 2, 4, 4, 4, 3, 1 ]
```

Při použití tohoto filtru ve vícerozměrných polích se používá okolí tvaru kruhu, čtverce, nebo například kříže. Provedení různých testů

ukázalo, že pro nejlepší výsledek při generování haptické odezvy je vhodné zvolit poloměr tohoto filtru na hodnotu 3. Je třeba dbát na vhodné zvolení poloměru filtru pro úkol, který je s daty vykonáván, protože mediánový filtr může odstranit i struktury, které byly žádoucí, například tenké oblasti dat, které se nacházejí mezi dvěma oblastmi se stejnou hodnotou.

Data je též vhodné ořezat, aby zde zbyla pouze ta část, která má být předmětem zkoumání. Toto je hlavně z důvodu rychlosti a též kvůli odstranění nesmyslných dat. Například na snímcích CT z oblasti abdomenu se velmi často vyskytuje lehátko, na kterém pacient ležel v průběhu snímání. Velikost dat ovlivňuje rychlost startu aplikace, protože se zde data zpracovávají několika průchody přes celá data. Pro příklad, necht' jsou dána data o rozměrech 512x512x100 voxelů. Po ořezání částí, ve kterých jsou data, která nenesou žádnou hodnotu o zkoumaném objektu, se zmenší rozměr dat na 380x300x100 voxelů. Velikost tohoto ořezání je běžná i v praktickém použití a úbytek datových polí je větší než polovina, protože původní data obsahovala zhruba 26 miliónů voxelů, ale ořezaná data obsahují jen zhruba 11 miliónů voxelů. Rychlost startu aplikace může v takovémto případě být až dvakrát větší. Nástroje pro provedení ořezání a aplikování mediánového filtru obsahuje MedV4D framework v projektu „tools“. Popis použití těchto nástrojů je uveden v dodatku B.

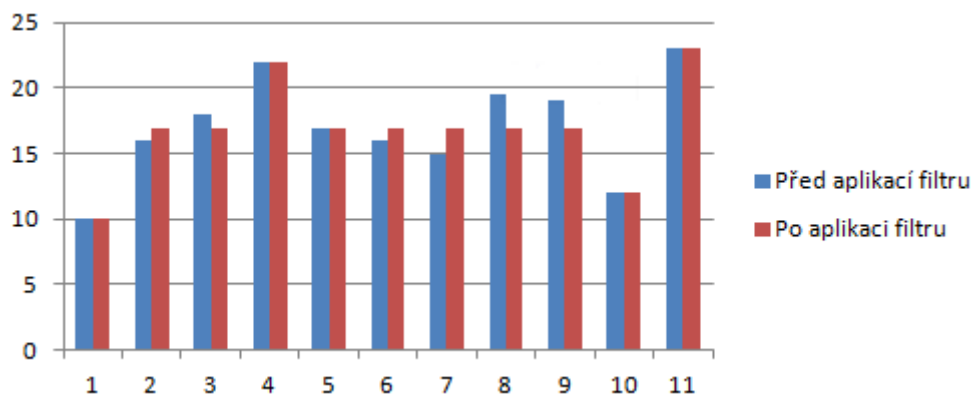
3.4 Implementační detaily

3.4.1 Data

Načítání dat je realizováno z již předem předpřipraveného souboru, který může obsahovat jakákoliv objemová data a nutné informace o nich. Do tohoto souboru jsou data převedena aplikací DICOM2ImageDump, které je součástí frameworku MedV4D.

Načítaný soubor se načítá do datových struktur tohoto frameworku a poté se přidělí aplikaci. Při spouštění aplikace v hlavní metodě main je pomocí třídy ImageFactory načten do třídy AImage::Ptr a z této je pak vytvořena třída ConnectionTyped, která se předává hlavní třídě aplikace m4dGUIOGLHapticViewerWidget v jejím konstruktoru. Po načtení do paměti v podobě datových struktur MedV4D frameworku jsou tyto pomocí nástrojů, které framework obsahuje převedeny do datových struktur, se kterými pracuje knihovna VTK. Jedná se o nástroje části frameworku MedV4D, která se nazývá vtkIntegration. Pro tento účel je vytvořena třída m4dImageDataSource, která již umí vracet vtkAlgorithmOutput a to je třída, se kterou umí pracovat všechny třídy, které se používají v konstrukci VTK pipeline. Vizualizace této pipeline je v následující kapitole na obrázku 3-2.

V jedné větvi jsou ještě data předzpracována pomocí agregačního filtru. Agregační filtr má přednastavený interval a všem hodnotám z tohoto intervalu přiřazuje jednu předem danou hodnotu. Příklad použití agregačního filtru je na obrázku 3-1.



Obrázek 3-1: Příklad použití agregačního filtru. Hodnoty z intervalu $\langle 15, 20 \rangle$ byly všechny převedeny na hodnotu 17.

Filtrování je z důvodu následného použití algoritmu „Marching cubes“. Tento algoritmus se používá k hledání isoploch v objemových datech a pro dostatečně kvalitní zobrazení 3D modelu je třeba zmenšit přesnost dat. Tímto se odstraní výkyvu hustot v jednotlivých tkáních a je zaručeno, že následný 3D model, který se z těchto dat tvoří, je celistvý.

3.4.2 Zobrazení

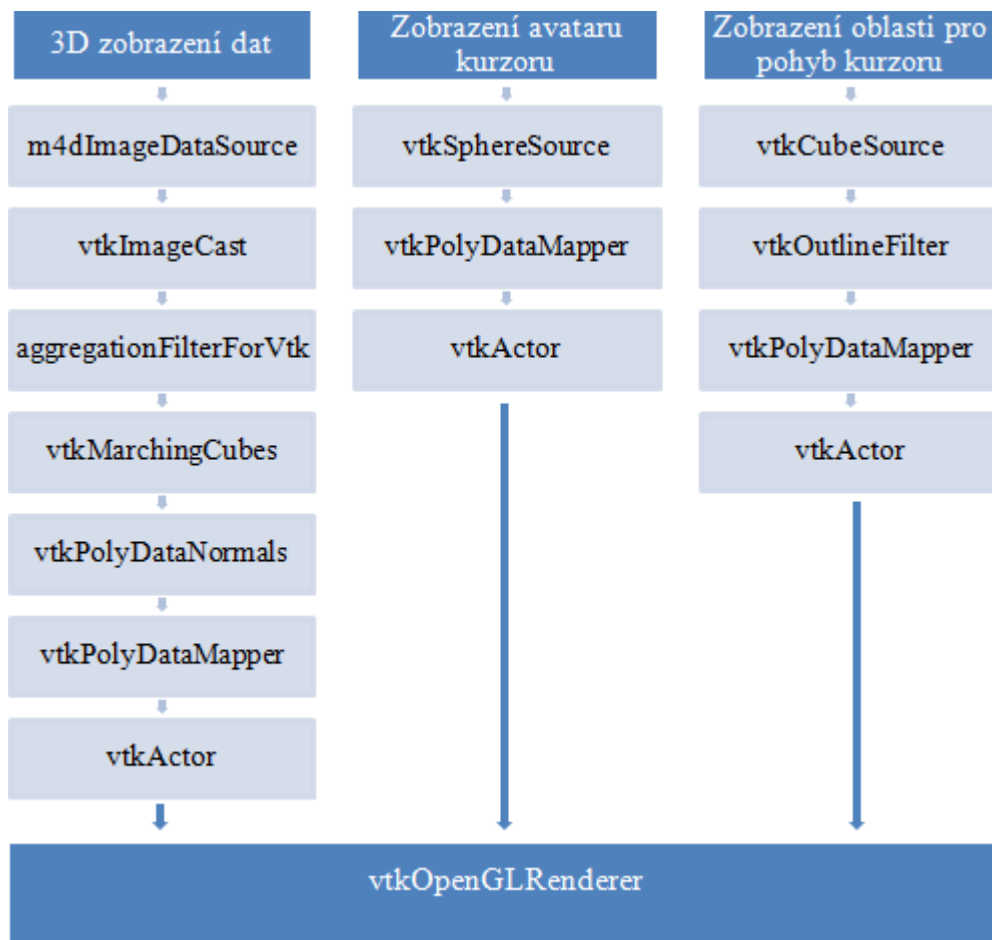
Vykreslování třídimenzionálního modelu dat je zprostředkováváno knihovnou VTK. Vykreslování je standardně nastaveno tak, aby bylo prováděno na grafické kartě. Nástroje VTK k tomu používají knihovnu OpenGL. V případě, že v počítači nebude přítomna grafická karta podporující správnou verzi knihovny OpenGL, bude použit k vykreslení procesor.

Vykreslují se předem vybrané isoplochy. Právě kvůli isoplochám, vykreslování 3D modelu používá filtrovaná data. V aplikaci je předem nastaveno a lze snadno změnit, kolik isoploch a pro jaké hodnoty hustoty se má v datech hledat. Nastavení je možné směnit pouze úpravou přímo v kódu. Ve stávající implementaci jsou hledány isoplochy pro hodnotu 600, která je však předtím pomocí agregačního filtru dosazena na místo všech hodnot v intervalu $\langle 550, 700 \rangle$. Toto nastavení je třeba volit velmi uvážlivě, protože průběh algoritmu „Marching cubes“ je velmi náročný a jeho zpracování například pro datové pole o rozměru $512 \times 512 \times 300$ voxelů trvá v řádech desítek sekund pro jednu hodnotu isoploch.

Jednotlivé isoplochy jsou vykresleny každá jinou barvou a jinou průhledností. Tyto vlastnosti lze v kódu snadno ovlivnit. Při ladění aplikace se ukázalo jako nejlepší řešení to, co je aktuálně v aplikaci implementováno, což znamená vykreslování právě pouze jedné isoplochy. Při použití příliš mnoha isoploch se ztrácí přehlednost 3D zobrazení. Též přílišné používání průhlednosti činí zobrazení velmi nepřehledným. Průhlednost je však třeba používat z důvodu dobré viditelnosti avataru kurzoru.

Vykreslení je prováděno pomocí VTK pipeline. Tato pipeline začíná třídou `vtkImageCast`, které předala data výše zmiňovaná třída `m4dImageDataSource`. Další částí pipeline je agregační filtr zastoupen třídou `aggregationFilterForVtk`. Tato pak předává data třídě `vtkMarchingCubes`, která zprostředkovává průběh stejnojmenného algoritmu. Následujícími třídami jsou `vtkPolyDataNormals`, která přidá k získaným polygonům normály a třída `vtkPolyDataMapper`, která mapuje získaná data na grafická primitiva a nakonec je zde zařazena třída `vtkActor`, která zastupuje danou entitu ve scéně, která se poté vykresluje. Samotné vykreslení zprostředkovává třída `vtkOpenGLRenderer`. Vizualizace této pipeline je na obrázku 3-2. Při použití s více isoplochami se tato pipeline za agregačním filtrem větví a je třeba vytvořit novou třídu `vtkMarchingCubes` a všechny třídy, které v pipeline následují znovu.

Krom isoploch jsou v třídimenzionálním zobrazení ještě následující objekty: avatar kurzoru a oblast, v níž se může kurzor pohybovat. Avatar kurzoru je zobrazen jako zelená neprůhledná koule. Tyto parametry spolu s velikostí avataru jsou výsledkem ladění a testování. Tuto kouli reprezentuje třída `vtkSphereSource`, která nese údaje o pozici a velikosti. Oblast, v níž se kurzor smí pohybovat, je znázorněna drátovým modelem modré krychle. Model krychle je vytvořen pomocí třídy `vtkCubeSource`, která opět uchovává informace o pozici a své velikosti, ze které je vytvořen drátový model třídou `vtkOutlineFilter`. Barvy jsou upraveny až ve třídě `vtkActor`.



Obrázek 3-2: Vizualizace VTK pipeline pro vykreslování třídimenzionálního modelu dat, kurzoru a oblasti pro pohyb kurzoru

Implementace zobrazovací oblasti 3D modelu obsahuje též metody pro interakci s uživatelem. Toto je realizováno pomocí třídy `vtkRenderWindowInteractor`, která zpracovává události vytvořené pohyby myši, mačkáním tlačítek myši a používáním kolečka na myši. Reakce jsou přibližování a oddalování modelu z isoploch, jeho otáčení a posun.

Vykreslení objemových dat v podobě řezu, ve kterém se nachází kurzor, je též implementováno pomocí knihovny VTK a to konkrétně pomocí třídy `vtkImageMapper`, které se pak tvoří `vtkActor2D`. Kvůli větší přesnosti toto zobrazení používá nefiltrovaná data. Též toto zobrazení neobsahuje žádnou jinou interakci s uživatelem, než zobrazení aktuálního řezu podle pozice kurzoru, případně řezu žádného, pokud kurzor není v oblasti s daty.

Zobrazení řezu je realizováno jako plocha, jejíž každý bod se obarví patřičnou barvou podle hodnoty bodu v datech. Hodnoty dat se

automaticky přiřadí na černobílou stupnici, černé body mají nejnižší hodnotu, kterou data obsahují a bílé naopak nejvyšší.

V zobrazení jednoho řezu se též vykresluje avatar kurzoru a oblast ve které se daný kurzor smí pohybovat. Avatar je zastoupen zeleným křížem a oblast pro pohyb je vyznačena modrým čtvercem. Tyto dvě věci jsou obě vykresleny pomocí čar. Čáry jsou zařazeny do pipeline tak, jak popisuje následující text. Zdrojem dat je zde `vtkLineSource`, který uchovává pozici a délku čáry, a ten je poté mapován pomocí `vtkPolyDataMapper2D` a zastupován na scéně pomocí `vtkActor2D`. Všechny třídy `vtkActor2D` jsou poté předány třídě `vtkRenderer`, která má na starosti samotné vykreslování.

3.4.3 Přejížděcí funkce

K výpočtu haptické odezvy je využívána přejížděcí funkce, která určuje hodnoty odporu prostředí pro jednotlivé hodnoty v objemových datech. Všechny dále popisované vlastnosti implementuje třída `transitionFunction`.

Funkce je uložena pomocí bodů, kterými prochází. Hodnota funkce v daném bodě se získává dvěma způsoby. Pokud daný bod je jeden z těch, které reprezentují funkci, tak se vrací přímo jeho hodnota, pokud je to bod, který ve funkci není, poté se jeho hodnota určí jako hodnota v daném bodě funkce přímky, spojující dva nejbližší body, pomocí kterých je přejížděcí funkce definována. Přejížděcí funkce je vždy určena alespoň dvěma body a to jsou okrajové hodnoty intervalu všech hodnot, které se nacházejí v objemových datech.

Další implementovanou funkcí přejížděcí funkce je uchovávání dvou hranic. Tyto jsou reprezentovány nějakými hodnotami z objemových dat. Hranice určují, do které a od které hodnoty se má prostředí s danou hodnotou hustoty chovat jako neprostupné. Tyto nemusí být vždy nastaveny.

V Přejížděcí funkci jsou též implementovány metody pro uložení, nebo načtení ze souboru. Soubor, do kterého se funkce ukládá, je textový a obsahuje data oddělená koncem řádky. První, co se v něm vyskytuje, je počet bodů funkce, poté následují jednotlivé body, vždy obě souřadnice jednoho bodu na jedné řádce oddělené mezerou a na konci jsou x-ové souřadnice hranic pro neprostupné prostředí.

3.4.4 Práce s haptickým zařízením

Implementaci, o které hovoří tato kapitola, zapouzdřuje třída `hapticCursor`. Pro umožnění práce s haptickým zařízením je nejprve provedena jeho detekce. Aplikace se při svém spuštění pokouší nalézt libovolné haptické zařízení, se kterým umí spolupracovat knihovna `Chai3D`. Toto se děje prostřednictvím třídy `cHapticDeviceHandler`. Pokud takové zařízení neexistuje, potom se kurzor nastaví do středu dat podle všech souřadných

os a již s ním není dále možno hýbat. Pokud takové zařízení existuje, načtou se do datových struktur, které uchovává třída `cGenericHapticDevice`, všechny informace o něm a provede se kalibrace zařízení, je-li to třeba. Kalibraci též zprostředkovává tato třída. Získané informace, které tato třída uchovává, obsahují například meze pro pohyb haptického zařízení, jméno a výrobce zařízení, maximální sílu haptické odezvy, kterou je zařízení schopno působit a podobně.

Kvůli citlivosti haptických zařízení je třeba dosáhnout dostatečně velké obnovovací frekvence směru a síly haptické odezvy, aby zařízení nevykonávalo nějaké neplynulé reakce. Obnovovací frekvence pozice zařízení je 1kHz. Pro dostatečně kvalitní haptickou odezvu je třeba, aby aplikace dokázala komunikovat se zařízením a obnovovat hodnoty jeho síly alespoň s frekvencí 500Hz. Z tohoto důvodu se veškerá komunikace s haptickým zařízením odehrává v separátním vlákně. Toto vlákno provádí pouze několik matematických výpočtů, které nezaberou téměř žádný procesorový čas, proto je vlákno v každém cyklu uspáno na dobu 1ms. Při této implementaci dosahuje průměrná frekvence komunikace se zařízením hodnoty 700Hz. Tato hodnota je pro kvalitní odezvu dostatečná.

Komunikace s haptickým zařízením je prováděna v cyklu, který vykonává následující operace: zjištění aktuální pozice zařízení, přepočítání souřadného systému haptického zařízení do souřadného systému objemových dat, výpočet haptické odezvy a nastavení směru a síly, kterou má zařízení působit. Haptická odezva se zařízením předává jakožto třísloužkový vektor.

Přepočítání souřadného systému haptického zařízení na systém, který se nachází v datech, je takový, že krajní pozice haptického zařízení odpovídají krajním pozicím oblasti pro pohyb kurzorem. Při zmenšení oblasti se tím dosahuje mnohem větší přesnosti, protože je fyzický pohyb zařízení mapován na menší pohyb virtuální. Přepočtu odpovídá následující pseudokód:

```
X.x = SouřadnicePoziceHaptickéhoZařízení.y  
X.y = SouřadnicePoziceHaptickéhoZařízení.z  
X.z = SouřadnicePoziceHaptickéhoZařízení.x  
s = DelkaHranyKrychleOblastiProPohyb / 2,0 /  
MaximálníVýchylkaHaptZařízení  
SouřadniceKurzoruVDatech = StředOblastiProPohyb + X / s
```

Do proměnné `X` se uloží vektor aktuálních souřadnic haptického zařízení, ale s osami přemapovanými na osy v datech. Poté se do proměnné `s` uloží měřítko. Toto zaručuje, že i při změně velikosti oblasti pro pohyb kurzoru je namapován pohyb kurzoru zařízení na pohyb v celé této oblasti. Souřadnice v datech se pak již jednoduše určí přičtením upravených souřadnic haptického kurzoru pomocí měřítka k středu oblasti pro pohyb zařízení.

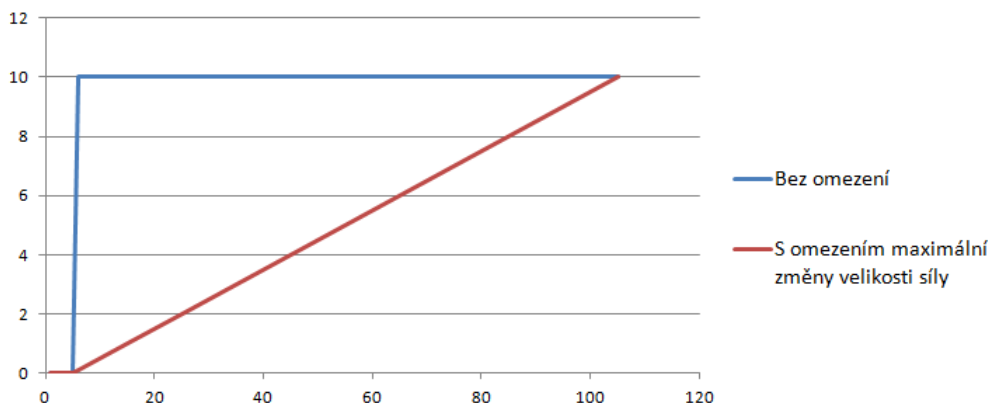
Samotný výpočet haptické odezvy je rozdělen na dva stavy. Stav kdy je kurzor v běžném prostředí a stav kdy se nachází kurzor v prostředí definovaném přechodovou funkcí jako neprostupné. Při spuštění aplikace je vždy stav zařízení nastaven na běžný režim. Jakmile se kurzor dostane do neprostupné oblasti, jsou uloženy patřičné informace a stav je přepnut pro výpočet v neprostupných oblastech. Pokud zařízení splní podmínky pro vystoupení z této oblasti, a nachází se skutečně v datech tam, kde je běžná hodnota přechodové funkce, vrací se stav do normálního režimu.

Výpočet haptické odezvy v běžném režimu má simulovat tuhost prostředí neboli velikost odporu prostředí vůči procházení nějakým tělesem. Nejdříve se určí směr síly, který odpovídá směru opačného vektoru k vektoru pohybu kurzoru bezprostředně před výpočtem haptické odezvy. Toto simuluje, že síla jde vždy přesně proti směru pohybu zařízení. Poté se směrový vektor normalizuje. Pomocí přechodové funkce se určí velikost síly, jakou má haptická odezva působit, a směrový vektor se jí vynásobí. Tento výsledný vektor se pošle zařízení jako nastavení haptické odezvy.

Z důvodů přílišné přesnosti zařízení, které způsobuje, že kromě krajních pozic zařízení nikdy nevrátí svou pozici shodnou s předchozí pozicí, je potřeba vektor pohybu zařízení před výpočtem též vypočítat, z pohybu zařízení na větší vzdálenosti, než je rozdíl posledních dvou pozic. Pokud by se tak nedělo, tak i v případě, že se uživatel snaží udržet zařízení v nějakém bodě v klidové poloze, se toto nemůže podařit a při příliš pomalém pohybu by docházelo k odchylkám směrového vektoru vůči správnému směru. V implementaci aplikace se průměruje posledních 30 směrových vektorů, které se získají jako rozdíl posledních dvou pozic v každém průběhu cyklu pro komunikaci s haptickým zařízením. Toto číslo je nastaveno jako konstanta přímo ve zdrojovém kódu třídy `hapticCursor` s názvem `NUMBER_OF_VECTORS`. Velikost tohoto čísla byla testována a hodnota 30 vykazovala nejlepší vlastnosti. Průměr vektorů je tedy při průměrné obnovovací frekvenci 700Hz počítán z pohybu, který proběhl v posledních 0,043 sekundy. Tento časový interval je dostatečně krátký aby zohledňoval pouze pohyb bezprostředně před výpočtem a zároveň je dostatečně dlouhý pro zjištění směru tohoto pohybu. Díky tomuto zprůměrování, dává haptické zařízení odezvu skutečně do protisměru pohybu tohoto zařízení.

Implementace obsahuje ještě jeden prvek pro korekci výpočtu haptické odezvy. Pokud se totiž velikost síly odezvy změní skokově, zařízení provede příliš nárazovou reakci a vychýlí uživatele z požadované pozice. Na takto rychlé „kopnutí“ není schopen lidský organismus včas reagovat, proto je zde implementována kontrola velikosti změny síly, kterou má haptická odezva působit. Pokud velikost změny překročí danou konstantu, je tato síla změněna právě jen o zmiňovanou konstantu. Tímto způsobem nedochází k náhlým „kopnutím“, ale k plynulé změně síly. Znázornění rozdílu nárůstu síly při použití tohoto algoritmu a bez něj znázorňuje

obrázek 3-3. Konstantu pro omezení této změny je třeba volit velmi obezřetně. V případě příliš velké této konstanty bude i nadále docházet ke „kopání“ a v případě příliš malé konstanty bude mít reakce na prostředí příliš velké zpoždění, nebo se dokonce znemožní dosažení maximální síly vůbec. Ve stávající implementaci programu je tato konstanta nastavena na 0,1, což je číslo vyjadřující změnu síly v Newtonech. Lze ji změnit ve zdrojovém kódu třídy `hapticCursor` a její název je `EPSILON`.



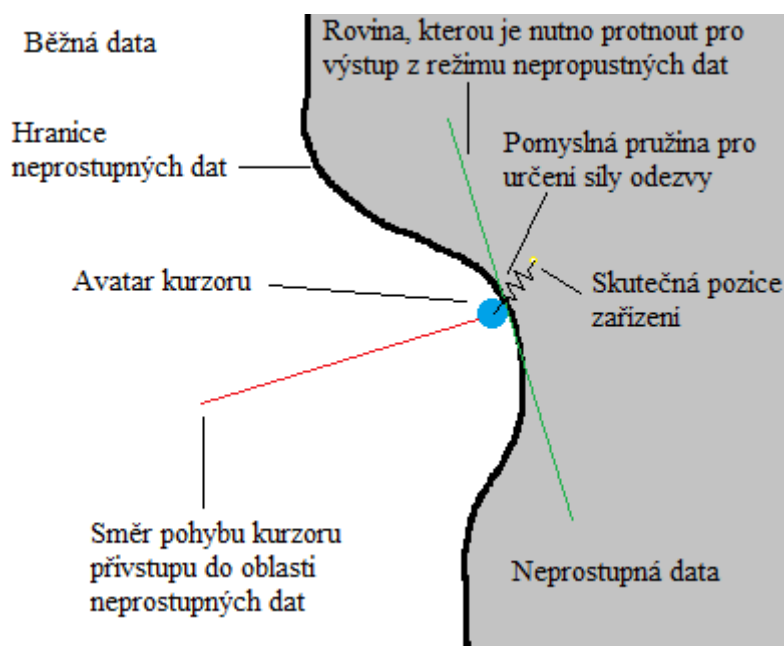
Obrázek 3-3: Znázornění rozdílu v nárůstu síly haptické odezvy při omezení maximálního nárůstu této síly a bez něj. Na ose x je uveden čas v počtu průběhů cyklu pro komunikaci s haptickým zařízením a na ose y je velikost síly v Newtonech.

Pokud je výpočet haptické odezvy nastaven na stav pro neprostupné oblasti, vypočítává se haptická odezva algoritmem, který simuluje natažení virtuální pružiny mezi kurzorem fyzického zařízení a bodem, kde došlo k průniku do oblasti neproniknutelných dat. V tomto bodě též zůstává avatar kurzoru, aby navodil iluzi, že kurzor skutečně nemůže pokračovat skrz tuto oblast. Algoritmu s natažením pružiny se využívá hlavně kvůli plynulému nárůstu síly, aby ani zde nedocházelo ke „kopání“ ze strany haptického zařízení. Tento algoritmus se běžně používá pro výpočet haptické odezvy nad geometrickými daty. Zařízení, pro které byla tato aplikace implementována má maximální sílu haptické odezvy 10N. Tato velikost síly průnik běžnému člověku neznemožní. Proto síla musí narůstat dostatečně rychle, aby tato změna budila dojem neprostupnosti. Situaci při používání této metody ilustruje obrázek 3-4.

Algoritmus se spustí v okamžiku, kdy se kurzor dostane do místa, které přechodová funkce označí jako neprostupné. Kurzor přestane vykazovat změny pozice a tím pádem dojde k zastavení pohybu avataru kurzoru na obrazovce. Z místa, kde zůstal avatar se k fyzickému zařízení natahuje pomyslná pružina o předem dané tuhosti. Tuhost pružiny musí být volena se stejným zřetelem jako omezení pro maximální změnu síly

v předchozí metodě, jinak bude haptická odezva vykazovat totožné chyby. Haptická odezva má pak směr, ve kterém je fyzický kurzor přitahován touto pružinou k místu, kde zůstal avatar a velikost určenou podle velikosti síly, kterou je fyzický kurzor přitahován k avatru.

Pro změnu metody výpočtu na původní je potřeba, aby se fyzické zařízení vrátilo do oblasti dat, kde je běžný odpor prostředí a aby při tomto návratu protnulo tzv. vstupní rovinu kurzoru do neprostopné oblasti. Tato rovina má následující parametry: prochází bodem, kde zůstal avatar kurzoru a její normálový vektor je roven vektoru pohybu kurzoru bezprostředně předtím, než se dostal do oblasti neprostopného prostředí. Opět je pro určení normálového vektoru roviny použito vektoru, který vznikl průměrem několika posledních směrových vektorů. Průchod rovinou je nutný pro správné detekování opuštění neprostopné oblasti, protože zařízení se svou odezvou sice snaží vracet kurzor přesně do místa, kudy se do neprostopné oblasti dostal, ale při pohybu zpět nemusí kurzor protnout přesně stejné souřadnice.



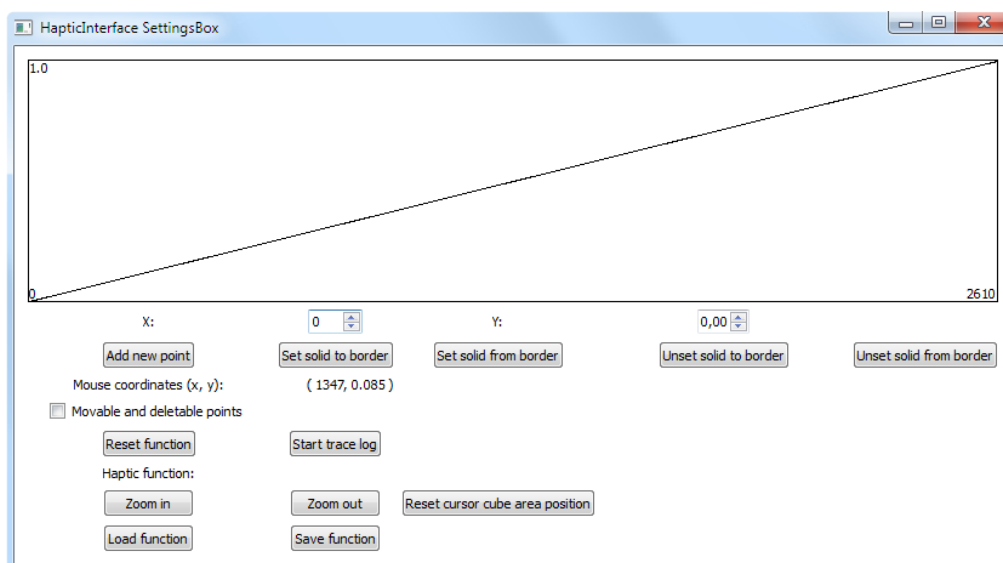
Obrázek 3-4: Znázornění stavu, kdy je výpočet haptické odezvy přepnut do režimu neprostopných dat, ve chvíli, kdy kurzor již vstoupil do oblasti s neprostopnými daty.

Zařízení rovněž implementuje funkce, pomocí kterých je možno spustit, či ukončit ukládání souřadnic haptického kurzoru. Souřadnice jsou ukládány do textového souboru oddělené koncem řádky. Ukládání není závislé na čase, nebo rychlosti pohybu kurzoru, ale uloží se vždy ty souřadnice, které jsou rozdílné od souřadnic, na nichž se kurzor nacházel

v minulém cyklu komunikace s haptickým zařízením. Jedná se o souřadnice v souřadnicovém systému objemových dat, tudíž zde není problém s přílišnou přesností haptického zařízení.

3.4.5 Rozhraní pro nastavení aplikace

Rozhraní pro nastavení aplikace je složeno z dvou hlavních částí. Část, kde se vykresluje přechodová funkce a je ji možno editovat za pomoci myši a přilehlých nástrojů a část, která obsahuje právě zmíněné nástroje a další ovládací prvky, kterými je možno ovlivňovat chování aplikace. Rozhraní je zobrazeno na obrázku 3-5.



Obrázek 3-5: Okno s rozhraním pro nastavení aplikace

Část pro vykreslení má implementovány dva stavy. V prvním, implicitním stavu reaguje na kliknutí do oblasti pro kreslení pouze přidáním bodu do přechodové funkce. Při přepnutí do druhého stavu se kolem každého bodu funkce vykreslí zelený kruh, který symbolizuje oblast, ve které se označí daný bod a je možno s ním provádět úpravy.

V módu pro úpravy přechodové funkce se při každém kliknutí levého tlačítka myši spočítá, zda se kurzor v momentě kliknutí nacházel v blízkém okolí nějakého bodu. Pokud ano, tak se tento považuje za označený a při pohybu myši se tento bod přesunuje. Prostředí kontroluje, zda není přesunut mimo interval sousedních dvou bodů. V takovém případě je implementována grafická odezva, funkce v okolí tohoto bodu zčervená. Toto chování je identické pro pokus o jakoukoliv neplatnou operaci. Kontrolují se souřadnice kurzoru vůči této ploše, zda se neocitl mimo ni. Aplikace též implementuje reakci na událost stisknutí pravého tlačítka myši. Pokud bylo

stisknuto tlačítko v blízkosti některého z bodů, pokusí se jej aplikace odstranit.

Zobrazovací oblast má též implementováno vykreslení hranic pro neprostupné prostředí. Pokud přechodová funkce tyto hranice obsahuje, tak jsou znázorněny svislými barevnými čarami. Celá tato část je implementována pomocí nástrojů pro kreslení ve 2D prostředí knihovny Qt. Veškeré vykreslování se odehrává při vyvolání události žádosti o aktualizaci. Tato událost se vyvolává při libovolné změně přechodové funkce, nebo při jakémkoliv pohybu myši, když je označen nějaký bod, aby se zaručilo uživateli vždy aktuální zobrazení přechodové funkce.

Zbytek rozhraní pro nastavení aplikace je implementován pomocí běžných formulářových prvků. Jsou to hlavně tlačítka zprostředkávající zvětšení či zmenšení oblasti pro pohyb kurzoru, přidání hranic hodnot funkce pro neprostupné prostředí, vyvolání dialogů pro uložení a načtení funkce ze souboru a dalších funkcí. Všechny akce, které jsou vyvolány v tomto rozhraní, jsou předány pomocí mechanismu signálů a slotů knihovny Qt hlavní části aplikace.

3.5 Překlad

Aplikace funguje pouze s operačním systémem Microsoft Windows XP, Vista a Seven. Byla vyvíjena a překládána pomocí vývojového prostředí Microsoft Visual Studio 2008 a překlad pomocí jiných překladačů nemusí být funkční. Aplikace byla testována v revizi 1578 a pro jiné revize tento postup překladu nemusí fungovat. Pro překlad aplikace je nutné provést následující kroky:

1. Stáhnout aktuální verzi frameworku MedV4D z repozitáře, který se nachází na adrese

`svn://cgg.mff.cuni.cz/MedV4D/trunk.`

2. Nainstalovat program pro generování projektu Cmake. Tento program lze získat na adrese

`http://www.cmake.org.`

3. Stáhnout ze stránky

`http://cgg.mff.cuni.cz/trac/medv4d/wiki/PrecompiledLib
Precom`

ze sekce pro Microsoft Visual Studio 2008 všechny soubory. A ze stránky

<http://cgg.mff.cuni.cz/trac/medv4d/attachment/wiki/HapticInterface/>

stáhnout soubor chai3d-2.0.0-precompiled.zip.

4. Rozbalit a zkopírovat obsah souborů dcmtk_msvc2008.zip, vtk-with-qt_msvc2008.zip, qt-reduced_msvc2008.zip a OtherLibs.zip do adresáře trunk, který byl stažen z repozitáře. A soubory boost_root.zip a chai3d-2.0.0-precompiled.zip rozbalit každý do svého adresáře.
5. Nastavit proměnnou prostředí CHAI3D_ROOT na cestu k adresáři, do kterého byl vykopírován obsah souboru chai3d-2.0.0-precompiled.zip a proměnnou BOOST_ROOT na cestu k adresáři, do kterého byl vykopírován obsah souboru boost_root.zip.
6. Spustit program cmake-gui.exe a do pole, které se jmenuje „Where is the source code:“ vložit adresu

trunk/src/cmake_project/Applications/HapticInterface

kde trunk je adresář stažený z repozitáře.

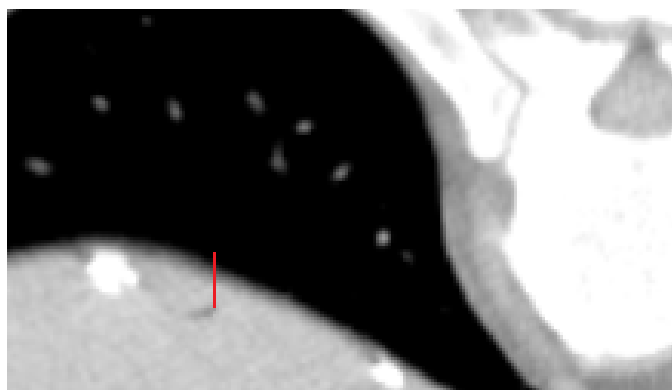
Do pole s názvem „Where to build the binaries:“ napsat adresu adresáře, do kterého chcete vygenerovat soubory projektu.

7. Stisknout tlačítko „Configure“. Při výběru překladače zvolte Microsoft Visual Studio 2008. Pokud po zpracování tohoto požadavku bude v proměnných, které se zobrazí vprostřed okna povolena knihovna Cg (CG_ENABLED), tak je třeba použití této knihovny zakázat. Znovu stisknout „Configure“ a poté stisknout tlačítko „Generate“.
8. V adresáři, který byl zadán pro vygenerování souborů projektu, se vytvoří soubor HapticInterface.sln, který je možno otevřít v Microsoft Visual Studiu 2008.

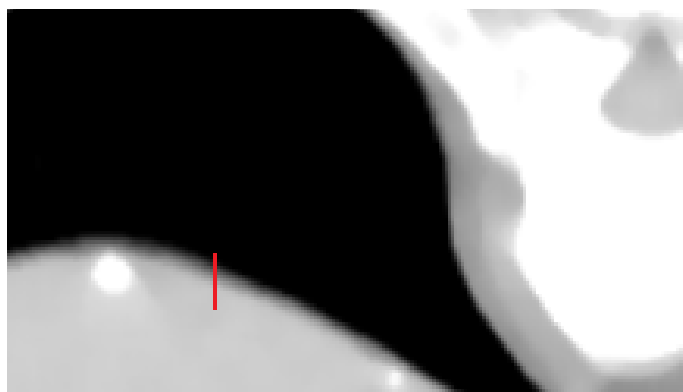
4. Výsledky

4.1 Příprava dat

Data, se kterými byly prováděny testy, byla naměřena při vyšetření CT, převážně z oblasti abdomenu. Data v některých oblastech obsahují nezanedbatelný šum, který zapříčiňuje špatnou funkci haptické odezvy a způsobuje artefakty v zobrazení objemových dat. Z tohoto důvodu byla vstupní data pro uskutečnění testů upravena mediánovým filtrem s poloměrem 3 voxely. Neupravená originální data jsou na obrázku 4-1 a pro srovnání data upravená výše zmiňovaným mediánovým filtrem jsou na obrázku 4-2.

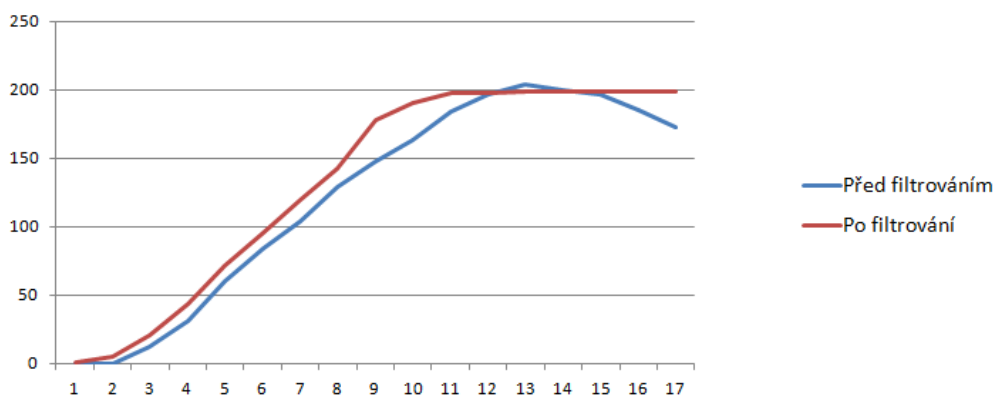


Obrázek 4-1: Zobrazení výřezu objemových dat v originální podobě před aplikací mediánového filtru, červená linka znázorňuje místo kde bylo provedeno měření hodnot pro graf na obrázku 4-3



Obrázek 4-2: Zobrazení výřezu dat, totožného jako na obrázku 4-1, po aplikaci mediánového filtru, červená linka znázorňuje místo kde bylo provedeno měření hodnot pro graf na obrázku 4-3

Jak znázorňují obrázky, mediánový filtr odstraní většinu artefaktů a přitom zachovává hrany. Obě tyto vlastnosti jsou pro generování haptické odezvy velmi důležité. Nedochází k nečekaným změnám síly uprostřed homogenních oblastí a přechody mezi jednotlivými oblastmi zůstávají jasně zřetelné. Porovnání hodnot na obrázcích v místě červené čáry je vyobrazeno na grafu v obrázku 4-3. Z grafu je snadno patrné, že přechod zůstal zachován, ale následné hodnoty ve světlé oblasti jsou již vyfiltrovány od šumu.



Obrázek 4-3: Obrázek znázorňuje průběh hodnot v datech v místě červené čáry na obrázcích 4-1 a 4-2.

4.2 Funkce aplikace

Pro testování a demonstraci funkcí aplikace HapticInterface byla zvolena data pořízena CT skenerem. Velikost dat je 380x300x30 voxelů a jedná se o výřez z dat pro vyšetření oblasti abdomenu. Tento výřez obsahuje spodní část plic a horní část břišní dutiny. Data byla ořezána oproti původnímu datasetu o část, kde se nacházelo lehátko, na kterém byl pacient položen a prázdné části dat.

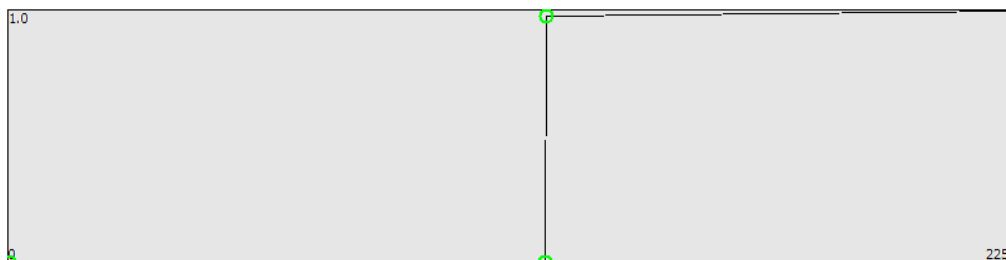
Spuštění aplikace s touto množinou dat trvá 50 sekund (Tohoto času bylo dosaženo na počítači s procesorem taktovaným na 3 GHz a disponujícím 6GB RAM o frekvenci 800 MHz). Největší část tohoto času jsou předpřipravována data pro průběh algoritmu „Marching Cubes“ a poté jeho samotný výpočet. Doba trvání je ovlivněna velikostí dat a počtem isoploch, které se v nich nacházejí.

Po spuštění je přechodová funkce nastavena na lineární průběh mezi dvěma body a tyto body přiřazují nulovou odezvu pro nejnižší hodnotu v datech a maximální odezvu pro hodnotu nejvyšší, kterou obsahují zpracovávaná data. Při použití této funkce je možno při procházení haptickým zařízením cítit odpor v pohybu odpovídající hustotě daného prostředí, ale nejsou zde cítit jemné přechody, kde se hodnota dat nezmění

dostatečně mnoho. Hranice jednotlivých částí dat jsou dobře identifikovatelné pomocí haptického zařízení pouze tam, kde se nachází velmi ostrý přechod. Stále jsou však tyto přechody velmi nepřesné, protože je prostor pro pohyb kurzoru příliš velký. Při změně velikosti a pozice oblasti pro pohyb kurzoru tak, aby obsahovala například jen obratel, se docílí mnohem přesnější identifikace hrany pomocí haptického zařízení.

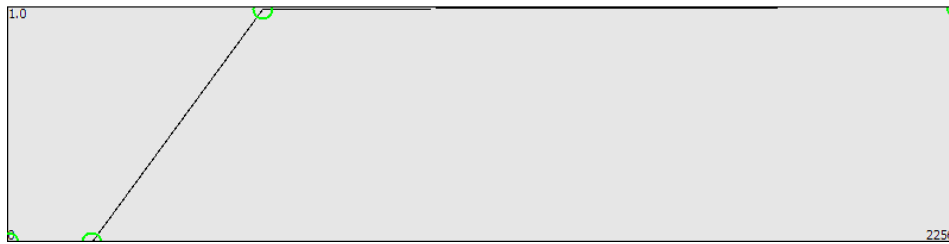
Pro zlepšení haptické odezvy je třeba upravit přechodovou funkci. Nejlepší vlastnosti haptická odezva vykazuje v případě hledání hrany mezi dvěma oblastmi dat. Toto použití bude demonstrováno na dvou příkladech.

Jako první příklad je nastavení přechodové funkce pro pozorování kosti. Toto nastavení je realizováno omezením haptické odezvy do hodnoty 1200 v datech a prudké navýšení odezvy od této hodnoty. Hodnota 1200 odpovídá na snímcích z CT skeneru oblasti s měkčími kostmi. Tuto hodnotu by neměla dosahovat žádná tkáň. Jedinou výjimkou může být výskyt kontrastní látky pro specializovaná vyšetření. Nastavení přechodové funkce, jak bylo popsáno výše, zobrazuje obrázek 4-4. Toto nastavení se opravdu projevuje nulovou odezvou na jakýkoliv pohyb haptického zařízení, dokud se kurzor nedostane do prostředí kosti. Změna v odporu zařízení je velmi výrazná a tudíž je velice snadné identifikovat jakoukoliv kost, která se nachází v datech.



Obrázek 4-4: Nastavení přechodové funkce pro detekci kostí v objemových datech, která byla vytvořena při vyšetření CT

Druhý příklad je používán pro procházení plic. Přechodová funkce je do hodnoty dat 200 nastavena tak, aby haptické zařízení nedávalo žádnou odezvu a v pro hodnoty 600 a vyšší dávalo maximální možnou odezvu. Mezi těmito body je v přechodové funkci přímka, tak jak je znázorněno na obrázku 4-5.



Obrázek 4-5: Nastavení přechodové funkce pro zkoumání plic v objemových datech, která byla vytvořena při vyšetření CT

Tato funkce umožňuje průchod plícemi s velmi přirozenou detekcí různých míst s větší hodnotou, které se v této oblasti nacházejí. Tato místa by měla odpovídat malým cévám a trubicím. Jejich přítomnost je v datech reprezentována právě hodnotami 200 – 600 a díky namapování celého rozsahu síly haptického zařízení na malý rozsah hodnot dat je možno i průchodem zařízení určit mohutnost té které oblasti. Okolí plíce se chová jako naprosto tvrdé, protože plíce obklopuje ze všech stran tkáň, která vykazuje hodnoty 900 a vyšší.

4.3 Srovnání polohovacích zařízení

Pro možnost určení, zda má haptická odezva při zkoumání objemových dat nějaký pozitivní efekt na přesnost či rychlost byl proveden experiment, který byl zaměřen na porovnání zkoumání dat při využití haptické odezvy a bez ní. Pokus spočíval v měření obvodu plic, což by v reálném případě mohlo napomocť k změření jejich objemu. Tento test byl zvolen hlavně proto, že plíce jsou obklopeny tkání s velkým rozdílem hodnot oproti hodnotám v plicích, je zde velmi zřetelné a ostré rozhraní a toto by měl podle dosavadních pozorování být přesně ten druh experimentu, kde haptická odezva vykáže nějaký přínos pro měření.

Pokus probíhal na předpřipravené datové množině a předem určené rovině, ve které se má změřit obvod dané plíce. Pro změření obvodu bylo použito možnosti uložit záznam pohybu haptického zařízení a poté se ze záznamu rekonstruovala naměřená délka. Testování probíhalo za tří různých podmínek. První měření bylo prováděno za pomoci haptického zařízení a nastavené přechodové funkce tak, že vnitřní část plic nevykazovala žádnou odezvu a vše ostatní odezvu maximální. Druhé měření bylo provedeno za pomoci haptického zařízení, ale s potlačenou jakoukoliv haptickou odezvou. Poslední měření bylo prováděno za pomoci myši. Uživatelé, kteří tento pokus prováděli, neměli žádnou speciální kvalifikaci pro manipulaci s haptickým zařízením, či práci s lékařskými daty. Většina z těchto uživatelů manipulovala s haptickým zařízením poprvé. Před prováděním pokusu však všichni měli nějaký čas pro seznámení se

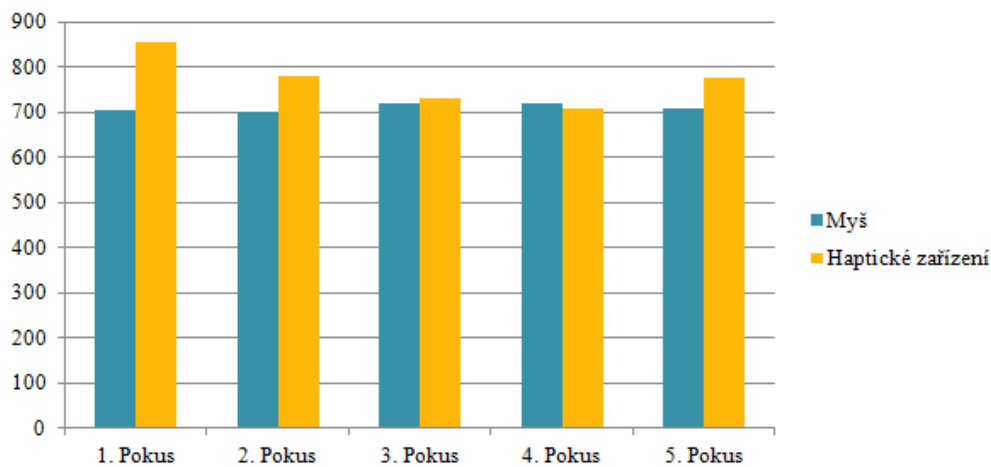
se zařízením. Při průběhu pokusů byla nastavena přechodová funkce tak, jak ji zobrazuje obrázek 4-6. Do hodnoty 700 byla síla haptické odezvy nulová a od této hodnoty maximální.



Obrázek 4-6: Nastavení přechodové funkce při měření obvodu plic

Výsledky pokusů, prováděných pomocí haptického zařízení s potlačenou haptickou odezvou, se ukázaly být velmi podobné výsledkům, které byly naměřeny pomocí myši. Jediným rozdílem se stala většinou nepatrně větší hodnota měřené délky obvodu plíce a to hlavně z toho důvodu, že uživatelé nebyli schopni udržet haptický kurzor v dané rovině a tudíž se kurzor místy dostával do jiných řezů, než ve kterém měl být pokus prováděn. Rozdíl vzdálenosti byl však nepatrný. Srovnání pokusů provedených pomocí myši a pomocí haptického zařízení s haptickou odezvou znázorňuje graf na obrázku 4-7.

Z grafu je patrné, že výsledky, naměřené pomocí myši, mají až na jeden případ nižší hodnotu a mnohem menší rozptyl. Měření pomocí haptického zařízení mají rozdíl mezi nejnižší a nejvyšší hodnotou 149 jednotek, což odpovídá asi 19,3% z průměrné hodnoty, které byla naměřena pomocí haptického zařízení. U pokusů provedených myši činí rozdíl mezi nejvyšší a nejnižší naměřenou hodnotou 19 jednotek a toto číslo odpovídá 2,6% z průměrné hodnoty obvodů, které byly naměřeny pomocí myši.



Obrázek 4-7: Graf zobrazující naměřené vzdálenosti obvodu plic při jednotlivých pokusech

Při pozorování průběhu jednotlivých pokusů bylo zřejmé, že udržení kurzoru myši přesně na barevném rozhraní nečinilo uživatelům žádný problém. Uživatelé pracující s myší byli schopni provést měření v kratším čase, než při použití haptického zařízení. Při používání haptického zařízení docházelo k zastavení kurzoru uživatelem při náhlém nárůstu haptické odezvy a následné snaze o korekci pozice. Korekci pozice uživatelé nejčastěji prováděli pohybem zpět do vnitřní části plíce, ale většinou se jim nedařilo zastavit na jejím okraji, nýbrž při náhlém poklesu haptické odezvy se kurzor dostal dál od hrany plíce.

Hodnocení samotných uživatelů se shodovalo v názoru, že pohodlnější a jednodušší je provádět daný úkol s myší. Při použití haptického zařízení si stěžovali na časté „zachycení“ zařízení v okolní tkáni.

Pro další srovnání polohovacích zařízení byl proveden pokus spočívající v identifikaci hrany kosti při pohybu po předem dané přímce. Pokus byl proveden za pomoci myši a za pomoci haptického zařízení, které mělo nastaveno přechodovou funkci tak, že haptická odezva pro hodnoty v datech menší než 1200 byla nulová a od hodnoty 1200 byla data nastavena jako nepropustná.

Při použití myši byli uživatelé schopni zastavit kurzor přesně na hledaném rozhraní, ale aby se jim to podařilo, tak museli snížit rychlost pohybu myši před rozhraním. Při použití haptického zařízení dokázali uživatelé identifikovat hledanou hranu se stejnou přesností jako při použití myši, ale nedocházelo zde ke zpomalování pohybu.

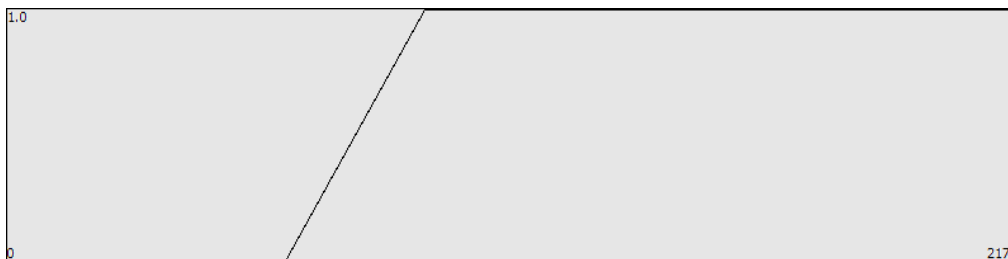
Podle vyjádření uživatelů bylo k této práci pohodlnější a jednodušší použít haptické zařízení. Uživatelé se shodovali na tom, že při použití haptického zařízení mají větší jistotu při nacházení požadované hrany.

4.4 Konfigurace přechodové funkce a její vliv na haptickou odezvu

Vliv konfigurace přechodové funkce na haptickou odezvu byl testován na uměle vytvořených datech. Data obsahovala pouze krychli o straně 30 voxelů, jejíž hodnoty byly pro v každém řezu jednotné a ve směru osy Z rostoucí. Hodnota voxelů v první rovině byla 900 jednotek. Rozdíl hodnoty v první a poslední rovině činil 300 jednotek. Zbylé hodnoty v datech, které tuto krychli obklopovaly, měly hodnotu 0.

Implicitní nastavení přechodové funkce pro tato data byla lineární funkce přidávající hodnotě 0 žádnou haptickou odezvu a hodnotě 900 odezvu maximální. Maximální síla haptické odezvy, kterou je schopno vykonat zařízení Novint Falcon je 10N. Tudíž při tomto nastavení odezva v krychli na straně s menší hustotou působí silou o velikosti 6,6N a v posledním řezu působí silou 10N. Navýšení síly mezi jednotlivými řezy bude zhruba 0,1N. V takovémto případě má průchod krychlí ve směru osy Z následující vlastnosti. Při proniknutí kurzoru do krychle je velmi dobře patrná její hrana a při rychlém pohybu kurzoru je též dobře znatelný nárůst síly. Při pohybu pomalém nebylo cítit nárůst odezvy mezi jednotlivými řezy. Z odezvy bylo znát její postupný nárůst, ale hranice jednotlivých voxelů byla neidentifikovatelná.

Pro druhý test byla přechodová funkce upravena tak, že do ní byl přidán bod přiřazující hodnotě 600 haptickou odezvu s nulovou silou. Díky této úpravě se vytvořila lineární funkce mezi body (600, 0) a (900, 1.0), kde druhá souřadnice reprezentuje poměr síly vůči maximální síle. Tato funkce je zobrazena na obrázku 4-8. Toto nastavení dává opět v posledním řezu odezvu 10N a při přechodu mezi jednotlivými řezy je rozdíl v síle odezvy 0,33N. Při této konfiguraci je velmi zřetelný vzrůst síly při pohybu uvnitř krychle, ale není identifikovatelná její hrana. Opět též přechod mezi jednotlivými řezy je nemožno zařízením identifikovat.



Obrázek 4-8: Nastavení přechodové funkce při druhém testu, lineární přechod mezi bodem 600 a 900

Další pokus byl proveden při zachování původního nastavení a umístěním hranice, od které má být prostředí neprostupné na hodnotu 700.

Tudíž při vstupu do desátého řezu krychle dojde k okamžitému navýšení haptické odezvy na 10N. Tato funkce je ilustrována obrázkem 4-9. Vstup do krychle zůstal opět neznatelný. Ovšem hranice desátého řezu je velmi dobře identifikovatelná, čemuž napomáhá i fakt, že se avatar kurzoru zastaví na této hranici. Zastavení kurzoru v zobrazení dat a prudké zvýšení síly haptické odezvy by mělo simulovat opravdu neprostupnou hranici, díky tomu, že lidský mozek spojí oba tyto vjemy dohromady. Tohoto faktu se využívá při použití haptických zařízení nad geometrickými daty. Pro téměř dokonalou iluzi neprostupné stěny zde stačí zobrazit stěnu z běžně známého neprostupného materiálu a prudce navýšit sílu odezvy. Při provádění tohoto pokusu není efekt natolik znatelný, protože sice dochází k zastavení kurzoru, ale zbytek obrazového vjemu nezpůsobuje dojem neprostupné hrany.



Obrázek 4-9: Nastavení přechodové funkce při třetím testu. Funkce obsahuje lineární přechod mezi hodnotami 600 a 900. V hodnotě 700 je umístěna hranice neprostupnosti

Tento pokus byl opakován s podobným nastavením, ale byla odebrána hranice neprostupnosti, místo které přibyly do funkce dva body a to (699, 0.33) a (700, 1.0). Tímto přechodová funkce simuluje předchozí pokus. Do hodnoty 699 se opět jedná o plynulý přechod a při hodnotě 700 dochází ke skokovému zvýšení síly haptické odezvy na 10N. Hranice mezi devátým a desátým řezem, jež je předmětem tohoto pozorování byla opět velmi dobře identifikovatelná, ale podle subjektivních pocitů ne tak dobře jako při předchozím pokusu. Též síla, kterou zařízení působilo, se subjektivně zdála menší. Toto je způsobeno tím, že se při průchodu hranicí nezastaví avatar kurzoru zařízení.

Další pokus byl prováděn pro zjištění, jakou hodnotu má mít rozdíl sil na rozhraní hodnot, aby bylo toto identifikovatelné pomocí haptického zařízení. Pokus měl dvě části, které se zabývaly přechodem mezi dvěma druhy rozhraní. První z těchto rozhraní bylo mezi prostředím s haptickou odezvou a prostředím bez ní, druhé rozhraní bylo mezi prostředími s různou hodnotou haptické odezvy. Pro správnou identifikaci hrany mezi prostředím bez odezvy a s haptickou odezvou bylo zapotřebí, aby odezva dosahovala alespoň 0.8N. Pro identifikaci rozhraní mezi prostředími s různou odezvou

byl pokus prováděn pro prostředí se silou odezvy 5N a větší silou odezvy druhého prostředí. V tomto případě byl přechod identifikovatelný až při změně o 1,6N. Tudíž velikost síly odezvy druhého prostředí byla rovna 6,6N. Výsledky tohoto pokusu jsou subjektivní a mohou se u různých uživatelů lišit.

4.5 Sledování hrany v objemových datech

Poslední pokusy byly provedeny pro zjištění, zda lze jen za pomoci haptického zařízení sledovat nějakou hranu mezi dvěma oblastmi s různými hodnotami v datech. Tyto pokusy byly provedeny na několika testovacích datech. Všechna tato data byla pořízena pomocí CT skenerů a obsahovala výsledky vyšetření oblasti abdomenu. Pokusy byly prováděny na různých hranách. Tyto hrany byly různých tvarů, byly zde zastoupeny rovné hrany a hrany do oblouků, které byly sledovány jak z vnitřní, tak z vnější strany. Pokusy byly prováděny s nastavením přechodové funkce tak, aby jedno z prostředí nevykazovalo žádnou haptickou odezvu a druhé, aby vykazovalo odezvu s maximální možnou silou. Hrana byla vždy sledována z prostředí, které mělo haptickou odezvu eliminovanou.

Pokusy byly prováděny za dvou druhů nastavení. V jednom byla maximální odezva realizována nastavením bodu přechodové funkce a v druhém byla zvolena hranice pro označení dat jako neprostupných. V obou těchto případech byly vyzorovány stejné výsledky a to na jakémkoliv tvaru hrany. Bylo zjištěno, že není možné zařízením identifikovat hranu. Při všech pokusech o pohyb po sledované hraně docházelo k tomu, že kurzor haptického zařízení se dostal do oblasti s maximální odezvou a uživatel jej vždy přesunul zpět do oblasti bez haptické odezvy. Toto bylo způsobeno tím, že zařízení vykazovalo odezvu, až ve chvíli kdy byl kurzor v oblasti s haptickou odezvou a v tomto momentě působí zařízení silou proti směru pohybu i když se zařízení pohybuje rovnoběžně s hranou. Dokud se však kurzor zařízení nedostal dovnitř této oblasti, tak zařízení nevykazovalo žádnou odezvu. Bez pomoci vizuálního kontaktu s prostředím bylo možno sledovat pouze hrany rovné a ty, co byly na vnitřní straně oblouku. Sledování však nebylo realizováno plynulým pohybem podél hrany, ale pohybem opisujícím křivku, která s velkou frekvencí přetínala sledovanou hranu.

5. Závěr

Výsledná aplikace umožňuje zobrazení objemových dat a jejich procházení pomocí haptického zařízení, které poskytuje haptickou odezvu. Aplikace umožňuje nastavení přechodové funkce pro definování haptické odezvy pro různé hodnoty vstupních dat. Bohužel se nepodařilo aplikaci plně integrovat do frameworku MedV4D z důvodu probíhajících změn ve struktuře tohoto frameworku.

Dle názoru jednotlivých uživatelů soudím, že ačkoliv aplikace disponuje možností zobrazení třídimenzionálního modelu, tak uživatelé pro orientaci v objemových datech používali zobrazení jednotlivých řezů. Přesto zobrazení 3D modelu má svůj smysl v nápomoci při orientaci v datech jako celku.

Z výsledků testů vyplývá, že používání haptických zařízení pro zkoumání objemových dat určitě má smysl. Největším přínosem je možnost pohybu jedním zařízením ve všech směrech, tak jak je to běžnému člověku přirozené. Přítomnost haptické odezvy u těchto zařízení je též přínosem. Ve stávající implementaci umožňuje snadné nacházení přechodů mezi jednotlivými oblastmi hodnot v datech a díky možnosti nastavení přechodové funkce i takových hran, které nejsou z vizuální interpretace dat patrné. Výsledky testů též ukázali, že haptická odezva tak, jak je realizována v této implementaci není vhodná pro sledování hran a rozhraní mezi oblastmi dat. Zde je naopak haptická odezva prvkem, který znesnadňuje požadovaný úkol.

Kvůli naměřeným výsledkům a provedeným pozorováním by v budoucích implementacích mělo dojít ke změnám vykreslovacího algoritmu pro třídimenzionální zobrazení. Stávající algoritmus by bylo vhodné nahradit některým z již existujících algoritmů pracujících metodou real-time raycastingu. Takovýto algoritmus by přinesl přehlednější a kvalitnější zobrazení, které by mohlo nahradit stávající dvojdimenzionální zobrazení při orientaci v datech. Další možnost, jak zkvalitnit zobrazení dat v budoucích implementacích, je přidání možnosti nastavovat jas a kontrast u dvojdimenzionálního zobrazení jednotlivých řezů.

Pro větší přesnost a kvalitu haptické odezvy bude v následujících implementacích zakomponován plnohodnotný algoritmus pro výpočet haptické odezvy na bázi proxy objektu. Tento algoritmus bude vyžadovat náročnější předzpracování dat a výpočet gradientu hodnot.

Aplikace bude integrována do frameworku MedV4D, který by jí měl poskytnout přívětivější uživatelské rozhraní s možností výměny dat za běhu aplikace a měl by též přinést některé nástroje pro provádění různých měření v datech. Z aplikace by měla být oddělena část, která zprostředkovává

komunikaci s haptickým zařízením a konfiguraci přechodové funkce a tato část by se měla stát samostatnou částí frameworku MedV4D, aby bylo možné používat haptické zařízení i v jiných aplikacích, které jsou vyvíjeny za pomoci tohoto frameworku.

Doufám, že se v budoucnosti dostane aplikace do fáze vývoje, kdy bude nasazena na skutečných lékařských pracovištích a umožní rychlejší a přesnější zkoumání objemových dat.

Použitá literatura

- [1] *MedV4D project*. <http://cgg.mff.cuni.cz/trac/medv4d>.
- [2] Cline, William E. Lorensen and Harvey E.: *Marching cubes: A high resolution 3D surface construction algorithm*. 1987. Proceedings of the 14th annual conference on Computer graphics and interactive techniques.
- [3] *Force dimension*. <http://www.forcedimension.com/>.
- [4] *Novint Technologies, Inc*. <http://home.novint.com/>.
- [5] *CHAI3D: The Open Source Haptics Project*. <http://www.chai3d.org>.
- [6] *H3DAPI - Open source haptics*. <http://www.h3dapi.org/>.
- [7] K. Salisbury, F. Barbagli and F. Conti.: *Haptic Rendering: Introductory Concepts*. 2004. Special Issue of IEEE Computer Graphics and Applications.
- [8] Karljohan Lundin, Anders Ynnerman and Björn Gudmundsson.: *General Proxy-based Haptics for Volume Visualization*. Pisa, Italy : s.n., 2005. Proceedings of the World Haptics Conference.
- [9] Karljohan Lundin, Anders Ynnerman and Björn Gudmundsson.: *Proxy-based Haptic Feedback from Volumetric Density Data*. University of Edinburgh, United Kingdom : s.n., 2002. Proceedings of the Eurohaptics Conference.
- [10] *Qt - Cross-platform application and UI framework*. <http://qt.nokia.com/>.
- [11] *VTK - The Visualization Toolkit*. <http://www.vtk.org/>.
- [12] *boost C++ libraries*. <http://www.boost.org/>.

Dodatek A – obsah přiloženého média

Na přiloženém médiu se nachází:

- /Aplikace – spustitelná verze aplikace, odpovídající zdrojovému kódu v revizi 1578
- /Data – vstupní data pro testovací účely
- /MedV4D
 - /boost_root – adresář s knihovnou Boost
 - /chai3d – adresář s knihovnou CHAI 3D
 - /trunk – adresář se zdrojovými kódy frameworku MedV4D a aplikace HapticInterface v revizi 1578. Zdrojové kódy aplikace jsou umístěny v /src/Applications/HapticInterface
- /Obrazky – ilustrace použité v tomto textu
- /Text – binární soubory tohoto dokumentu

Dodatek B – uživatelská dokumentace

Úvod

Aplikace HapticInterface je určena pro práci s objemovými daty pořízenými pomocí přístrojů CT, poskytuje možnost jejich zobrazení a procházení těchto dat pomocí haptického zařízení. Aplikace nabízí zobrazení objemových dat jak v podobě 3D modelu, tak v podobě zobrazení jednotlivých řezů.

Aplikace je připravena pro použití se zařízením Novint Falcon vyráběným firmou Novint Technologies, Inc. Krom tohoto zařízení lze též použít virtuální zařízení, které je součástí frameworku CHAI 3D. S tímto virtuálním zařízením však není přítomna všechna funkčnost aplikace.

Spuštění

Vlastní aplikace je pouze jeden soubor, který se jmenuje „HapticInterface.exe“. Pro běh aplikace je však třeba mít v adresáři kde se nachází spustitelný soubor aplikace též soubor „hdFalcon.dll“, který je součástí CHAI 3D frameworku, nebo knihovny Novint Falcon SDK. Pro spuštění aplikace je třeba použít příkazovou řádku a jako parametr udat cestu k souboru s daty. Tento vstupní soubor je typu ImageDump, který je součástí frameworku MedV4D. Příklad příkazu pro spuštění aplikace:

HapticInterface c:\data\data.dump.

Pokud je aplikace spuštěna a je k ní připojeno zařízení Falcon, tak pečlivě sledujte signalizační světlo na těle zařízení. Pokud toto zčervená, je třeba provést kalibraci zařízení, která se provede protažením všech ramen do obou krajních poloh. Signalizační světlo zčervenat nemusí, protože zařízení Falcon je třeba kalibrovat pouze jednou po připojení k počítači. Pokud je aplikace spouštěna bez jakéhokoliv haptického zařízení, nebo pouze se zařízením virtuálním, není třeba žádné kalibrace.

Spouštění aplikace může trvat delší dobu, která je závislá na velikosti vstupních dat. Tato doba může dosáhnout až několika minut. Po spuštění aplikace se zobrazí tři okna tak, jak to znázorňuje obrázek 1. Jedno obsahuje konzoli, druhé ovládací prvky a zobrazení přechodové funkce a třetí obsahuje zobrazení dat. Okno konzole obsahuje pouze výpisy činnosti aplikace. Při zavření tohoto okna se ukončí celá aplikace.

Předzpracování dat

Pro možnost zadání vstupních dat aplikaci je třeba tyto převést do formátu ImageDump, jak bylo zmíněno výše v textu. Převod adresáře s daty ve formátu DICOM lze převést pomocí nástroje DICOM2ImageDump, který obsahuje framework MedV4D v části tools. Tento nástroj se spouští z příkazové řádky a jako první parametr je adresář s daty a druhý parametr určuje adresář pro umístění výstupní souboru. Výstupní soubor se vždy jmenuje DICOMseries.dump. Příklad příkazu pro použití tohoto nástroje:

```
DICOM2ImageDump c:\data\slozkaDICOM\ c:\data\
```

Dalším užitečným nástrojem, který je v sekci tools, je aplikace pro aplikaci mediánového filtru. Této aplikaci se jako parametry zadávají název vstupního a výstupního souboru a parametr `-r`, který určuje poloměr tohoto filtru. Vstupní soubor musí být ve formátu ImageDump. Příklad příkazu pro použití tohoto filtru:

```
MedianFilter -r 3 c:\data\DICOMseries.dump  
c:\data\filtrovany.dump
```

Následující nástroj se používá pro ořezávání obrázků a jmenuje se ImageCrop. Jeho vstupní soubor opět musí být ve formátu ImageDump a jako parametry se musí zadat seznam souřadnic pro dva body. Tyto body jsou okrajovými body tělesové úhlopříčky kvádrů, jehož vnitřní část bude výstupem tohoto nástroje. Dále je třeba uvést vstupní a výstupní soubor. Příklad použití tohoto nástroje:

```
ImageCrop -r 0 -r 0 -r 0 -l 10 -l 10 -l 10  
c:\data\velky.dump c:\data\orezany.dump
```

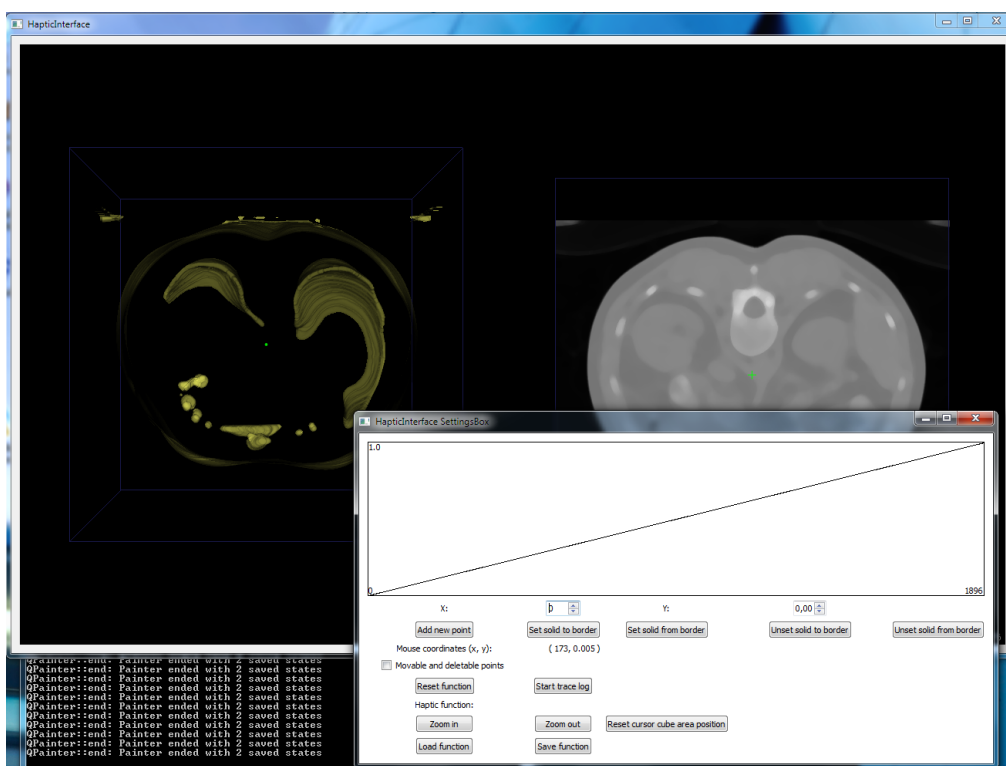
Zobrazení objemových dat

Okno se zobrazením objemových dat obsahuje tři zobrazovací oblasti. V levé půlce se nachází zobrazení 3D modelu. Toto zobrazení je interaktivní. Při stisknutí levého tlačítka myši a následném pohybu myši lze rotovat obrázkem, při stisku pravého tlačítka a pohybu myši se obrázek posouvá. Pro oddálení, nebo přiblížení obrázku se používá kolečko na myši. 3D model zobrazuje objemová data, avatar kurzoru haptického zařízení a oblast v níž je možno se pohybovat, jež je reprezentována tmavě modrou krychlí.

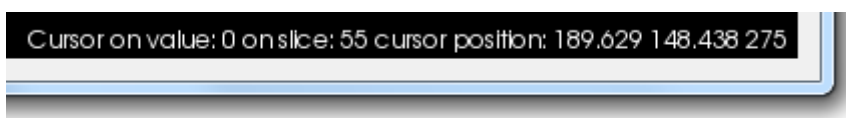
V pravé části okna se nachází zobrazení jednotlivých řezů. Zobrazen je vždy ten řez, ve kterém se nachází kurzor haptického zařízení. Pokud je kurzor na pozici, kde žádný řez není, je zobrazena pouze černá plocha. Toto

zobrazení též zobrazuje avatar kurzoru, jež je vykreslen pomocí zeleného kříže a také oblast pro pohyb kurzoru, kterou reprezentuje tmavě modrý čtverec.

Třetí zobrazovací část se nachází v pravém dolním rohu tohoto okna. V této oblasti je výpis, který informuje uživatele o hodnotě dat na aktuální pozici kurzoru, souřadnici řezu, ve kterém se kurzor nachází a souřadnice kurzoru v tomto řezu. Tuto část ukazuje obrázek 2.



Obrázek 1: Aplikace ihned po spuštění



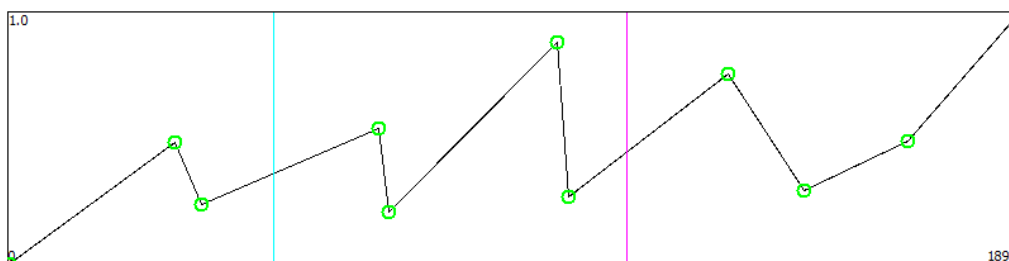
Obrázek 2: Detail třetí zobrazovací oblasti v okně s vizualizací objemových dat

Okno s ovládacími prvky aplikace

Toto okno je rozděleno do dvou hlavních oblastí. První z těchto oblastí je obdélník v horní části okna, kde se vykresluje přechodová funkce. Spodní okraj obdélníku slouží jako x-ová osa, jejíž maximální hodnota je vypsána

v pravé dolní části obdélníku. Levý okraj tohoto obdélníku slouží jako y-ová osa a v levém horním rohu se nachází maximální hodnota této osy. Hodnoty x-ové osy odpovídají hodnotám v objemových datech a hodnoty na y-ové ose určují poměr síly haptické odezvy vůči maximální odezvě (0 je pro žádnou odezvu a 1,0 pro odezvu maximální). V tomto obdélníku je též vykreslena čára znázorňující průběh přechodové funkce. Tato oblast též může vykreslovat až dvě barevné svislé čáry. Azurově modrá čára znázorňuje hodnotu, do které jsou data považována za neprostupná, a fialová čára označuje hodnotu, od které jsou data označována jako neprostupná.

Zobrazení v této oblasti má dva režimy. V režimu, v jakém se aplikace spustí je do funkce možno přidávat body kliknutím levého tlačítka myši do obdélníku, ve kterém je funkce vykreslena. Zobrazovací oblast v tomto režimu znázorňuje obrázek 3.

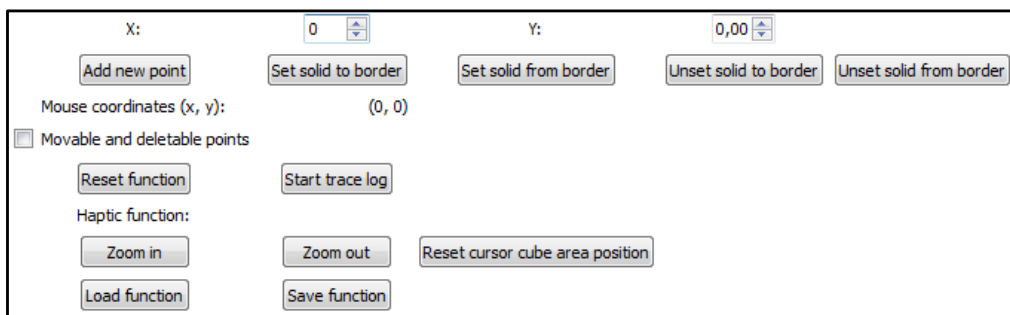


Obrázek 3: Oblast pro zobrazení přechodové funkce, která je přepnuta do režimu možnosti editace přechodové funkce pomocí myši. V oblasti jsou též zobrazeny hranice pro nastavení neproniknutelných oblastí v datech

Pokud se zobrazení přepne do druhého režimu, což je realizováno zaškrtnutím možnosti „Moveable and deleteable points“ ve spodní části tohoto okna, tak se kolem každého bodu přechodové funkce navíc vykreslí zelené kolečko. V tuto chvíli je možno upravovat funkci pomocí myši ještě několika dalšími způsoby. Po kliknutí do blízkosti nějakého bodu pravým tlačítkem myši, bude tento odstraněn z přechodové funkce. Po kliknutí levým tlačítkem myši do dostatečné vzdálenosti od ostatních bodů bude přidán nový bod do přechodové funkce na souřadnice, na něž bylo kliknuto. Při stisknutí levého tlačítka myši do blízkosti nějakého bodu je možno upravit pozici tohoto bodu v přechodové funkci. V momentě stisknutí tlačítka se čáry spojující tento bod s body okolními změní na modré. Poté je možno pomocí pohybu myši za stálého držení levého tlačítka bod přesunout. Bod je možno umístit pouze dovnitř zobrazovací oblasti a pouze na x-ové souřadnice, které leží mezi jeho sousedními body. Potvrzení pozice se provede uvolněním tlačítka myši. V případě přetažení bodu na neplatnou pozici čáry, které jej spojují s okolními body, zčervenají. Pokud dojde

k uvolnění tlačítka myši ve chvíli, kdy se nachází bod v neplatné pozici, bude tato změna ignorována.

Ve spodní polovině tohoto okna se nacházejí standardní formulářové ovládací prvky. Tuto oblast znázorňuje obrázek 4. Do oblasti pro zadávání čísel je možno zadat souřadnice bodu. Po stisknutí tlačítka „Add new point“ je do funkce přidán bod právě s těmito souřadnicemi.



Obrázek 4: Spodní část okna, které slouží pro ovládání aplikace

Po stisku tlačítka „Set solid to border“ se do funkce přidá hranice na předem zadanou x-ovou souřadnici, do jejíž hodnoty budou objemová data označena jako neprostupná. Tlačítko „Set solid from border“ přidává hranici stejným způsobem, ale data budou považována za neprostupná od této hranice. Je možné, aby funkce obsahovala obě tyto hranice. Stisk tlačítek, jejichž název začíná „Unset“ vyvolá odstranění příslušné hranice z přechodové funkce.

V poli označeném „Mouse coordinates“ je zobrazena pozice kurzoru myši v oblasti pro vykreslení přechodové funkce.

Zaškrtnutím pole „Moveable and deleteable points“ se přepíná režim vykreslování přechodové funkce do již zmíněného druhého stavu. Stiskem tlačítka „Reset function“ se provede kompletní vymazání bodů přechodové funkce a případných hranic neprostupnosti. Ve funkci zůstanou přítomny pouze krajní body a to v pozicích, v jakých se nacházely při spuštění aplikace. Tlačítka „Save function“ a „Load function“ umožňují uložení přechodové funkce do souboru a její opětovné načtení.

Tlačítka „Zoom in“ a „Zoom out“ zprostředkovávají zvětšení či zmenšení oblasti pro pohyb kurzoru. Tato oblast má vždy tvar krychle a její střed se umístí na aktuální pozici kurzoru haptického zařízení. Rozsah pohybu haptického zařízení je vždy namapován tak, aby pokrýval celý prostor pro pohyb. Tímto se při zmenšování tohoto prostoru dosahuje větší přesnosti procházení a haptické odezvy. Tlačítkem „Reset cursor cube area position“ se vrací pozice a velikost oblasti pro pohyb kurzoru do stejné podoby jako měla při spuštění aplikace.

Tlačítko „Start trace log“ zapíná ukládání aktuální pozice kurzoru do souboru. Pozice je uložena při jakékoliv změně. Vypnutí této funkce se

docílí stisknutím téhož tlačítka, které bude mít změněný název na „Stop trace log“.

Pro vypnutí aplikace je třeba zavřít okno konzole.