VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYUŽITÍ 3D POČÍTAČOVÉ GRAFIKY PRO APLIKACE V MEDICINĚ

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE | Bc. JAN JAVŮREK
AUTHOR

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYUŽITÍ 3D POČÍTAČOVÉ GRAFIKY PRO APLIKACE V MEDICINĚ

USING 3D COMPUTER GRAPHICS FOR MEDICAL APPLICATIONS

DIPLOMOVÁ PRÁCE
MASTER‘S THESIS

AUTOR PRÁCE — Bc. JAN JAVŮREK
AUTHOR

VEDOUCÍ PRÁCE — Ing. PŘEMYSL KRŠEK, Ph.D.
SUPERVISOR

BRNO 2007

Zadání diplomové práce/4962/2006/xjavur01

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Javůrek Jan, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Využití 3D počítačové grafiky pro aplikace v medicíně**

Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s problematikou 3D počítačového geometrického modelování v medicíně.
2. Analyzujte základní možnosti tvorby a funkce počítačových grafických systémů pro 3D modelování v medicíně.
3. Navrhněte počítačový 3D grafický systém pro konkrétní aplikaci 3D modelování v medicíně.
4. Implementujte vybranou část navrženého systému ve vybraném jazyce (C/C++, Java, Python, C#).
5. Zhodnoťte dosažené výsledky a stanovte další vývoj projektu

Literatura:

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

Při obhajobě semestrální části diplomového projektu je požadováno:

- Splnění prvních 3 bodů zadání.

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

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

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

Vedoucí: **Kršek Přemysl, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 28. února 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Bc. Jan Javůrek**
Id studenta: 49393
Bytem: Česká Rybná 86, 561 85 Česká Rybná
Narozen: 21. 02. 1983, Ústí nad Orlicí
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

....

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   diplomová práce

Název VŠKP: Využití 3D počítačové grafiky pro aplikace v medicíně
Vedoucí/školitel VŠKP: Kršek Přemysl, Ing., Ph.D.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP: ....

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☒ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: ....

....	....
Nabyvatel	Autor

Abstrakt

Tato diplomová práce se zbývá problematikou 3D počítačového geometrického modelování a návrhem grafického systému pro využití v medicíně. Navržený grafický systém byl úspěšně implementován pro konkrétní aplikaci ve stomatologii.

Klíčová slova

počítačové modelování, geometrie, scéna, graf scény, grafický objekt, transformace, matice, implantát, model, arcball

Abstract

This thesis describes problems 3D computer graphic modeling and design graphic system for use in medicine. Design graphic system was successfully implement for concrete application in stomatology.

Keywords

computer modeling, geometry, scene, graph scene, graphic object, transformation, matrix, implant, model, arcball

Citace

Jan Javůrek: Využití 3D počítačové grafiky pro aplikace v medicíně, Brno, FIT VUT v Brně, 2007

Využití 3D počítačové grafiky pro aplikace v medicíně

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Přemysla Krška, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……………………
Jan Javůrek
22.5.2007

Poděkování

Poděkování patří Ing. Přemyslu Krškovi, Ph.D. za konzultace a pomoc při tvorbě této diplomové práce.

© Jan Javůrek, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah 1
1 Úvod 3
2 3D počítačové modelování 4
2.1 Reprezentace 3D počítačových modelů 4
2.1.1 Hraniční reprezentace (B-rep) 4
2.1.2 Konstruktivní geometrie (CSG) 11
2.1.3 Šablonování 11
2.1.4 Implicitní plochy 12
2.1.5 Dekompoziční modely 13
2.2 Medicínská obrazová data 14
2.2.1 Vektorizace prostorových řezů 15
2.3 Transformace 16
2.3.1 Homogenní souřadnice 17
2.3.2 Posunutí ve 3D 17
2.3.3 Rotace kolem osy ve 3D 18
2.3.4 Rotace kolem obecné osy ve 3D 18
2.3.5 Změna měřítka ve 3D 19
2.3.6 Skládání transformací 20
2.4 Reprezentace 3D počítačové scény 20
2.4.1 Graf scény 21
3 Návrh 3D grafického systému 23
3.1 Virtuální scéna 24
3.1.1 Návrh základního grafického objektu 24
3.1.2 Návrh scény 28
3.1.3 Manipulace s objekty a scénou 30
3.1.4 Návrh měřících nástrojů 32
3.1.5 Návrh reprezentace zubního implantátu 32
3.2 Okno grafu scény 33
3.3 Okno menu s ovládáním 34
4 Implementace 36
4.1 Knihovna wxWidgets 36
4.2 Knihovna OpenGL 37
4.3 Knihovna VectorEntity 38
5 Výsledky 39

5.1 Ovládací lišta ... 39
5.2 Menu ... 40
5.3 Graf scény ... 40
5.4 Okno scény ... 42
5.5 Měření ... 43
6 Závěr ... 45
Literatura ... 46
Seznam příloh ... 47
Seznam obrázků ... 47

1 Úvod

S prudkým rozvojem informačních technologií koncem minulého století se i v moderní medicíně začalo stále více používat nejrůznějších technických zařízení a přístrojů, které umožňují zlepšovat úroveň diagnostiky a ošetření pacientů. Jedním ze současných světových trendů je individuální přístup k ošetření pacienta, protože nikdo nejsme stejní. V lékařských oborech jako je stomatologie, chirurgie, ortopedie a plastická chirurgie jsou právě tyto odlišnosti v rozměrech, proporcích a drobných detailech každého jedince zásadní, proto je jedním z klíčů k naplnění těchto specifických požadavků využití počítačového modelování.

Tato diplomové práce se zabývá analýzou možností využití 3D počítačového modelování v medicíně a implementací systému využitelného pro aplikaci ve stomatologii.

Teoretický úvod práce popisuje 3D počítačové modelování, možnosti reprezentací počítačových modelů a způsob práce s těmito modely v trojrozměrném prostoru. Zároveň je popsán způsob jakým se v dnešní době získávají medicínské modely částí lidského těla.

Další kapitola se věnuje návrhu vytvářeného 3D grafického systému. V této části práce jsou uvedeny návrhy grafického uživatelského rozhraní, virtuální scény, grafických objektů, měřících nástrojů, manipulačních nástrojů atd.

Kapitola Implementace popisuje jakým způsobem byl realizován 3D grafický systém a jaké nástroje k tomu byly využity. Po popisu implementace následuje část věnující se výsledkům a ukázkám vytvořené aplikace.

Poslední část, závěr, zhodnocuje vytvořený grafický systém a možnosti přínosu informačních technologií pro medicínu.

2 3D počítačové modelování

Pojmem modelování v 3D počítačové grafice rozumíme znázornění třírozměrných modelů(objektů) na obrazovce počítače. Pokud chceme modely zobrazovat musíme nejprve stanovit jejich reprezentaci v trojrozměrném prostoru.

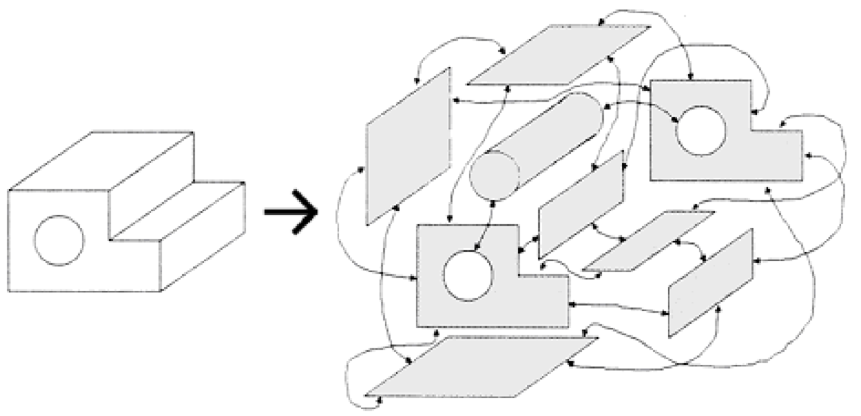
2.1 Reprezentace 3D počítačových modelů

Mnoho objektů zobrazovaných v trojrozměrném prostoru si obvykle lze představit jako tělesa, která jsou obdobou skutečných hmotných předmětů, které zaujímají určitý objem. Na těleso proto můžeme nahlížet jako na množinu bodů v trojrozměrném prostoru, splňující určitá kritéria. Definujeme-li relaci sousednosti bodů, pak můžeme nahlížet na těleso jako na sjednocení dvou disjunktních množin, množin vnitřních bodů a množinu hraničních bodů.[Žára] Každý vnitřní bod sousedí pouze s vnitřními nebo hraničními body. Hraniční bod sousedí alespoň s jedním hraničním bodem, vnitřním bodem a bodem vnějším, nepatřícím do žádné z uvedených dvou množin. Těleso je přitom chápáno jako spojitý útvar, tvořený jedním celkem. Tato definice vylučuje ze skupiny těles objekty jako: úsečky či křivky v prostoru, části rovin a obecně plochy. Tyto objekty totiž nemají žádné vnitřní body ve smyslu tělesa, ale přesto jsou cenné, neboť slouží k popisu množiny hraničních bodů.

V této práci jsou popsány typické způsoby reprezentace těles. Nejvíce prostoru je věnováno hraniční reprezentaci, která je nejběžnějším způsobem popisu trojrozměrných objektů v počítačích. Její zobrazení se snadno provádí na dnešních grafických procesorech. Přesto se také práce věnuje uvedení způsoby popisu těles, používaných zejména v CAD systémech a technikám pro modelování pomocí šablonování, dekompozice prostoru a implicitních ploch.

2.1.1 Hraniční reprezentace (B-rep)

Hraniční reprezentace je jedním z nejběžnějších způsobů reprezentace těles spočívá v popisu hranice (boundary representation, B-rep), tedy v popisu množiny hraničních bodů jak ukazuje obrázek 1. Informace o vnitřních bodech tělesa se buď neuchovávají, nebo je lze odvodit z popisu hranice. Hranice tělesa je jeho přirozenou reprezentací, vždyť většina lidí kreslí tělesa právě pomocí jejich obrysu, který je podmnožinou hranice.



Obr. 1 Popis tělesa je v hraniční reprezentaci převeden na popis pláště

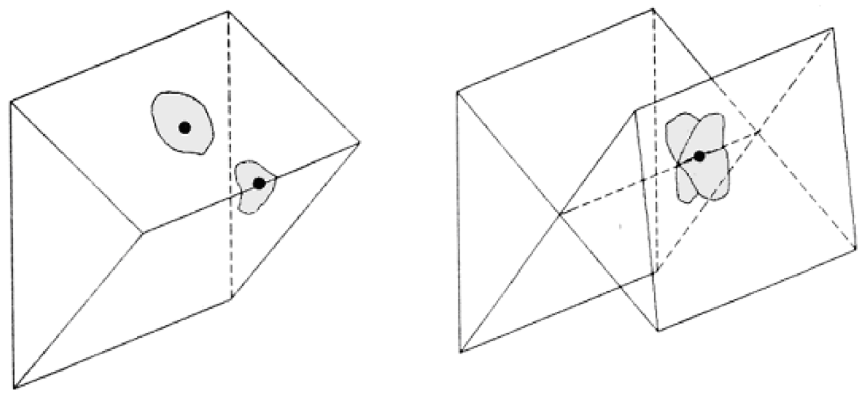
2.1.1.1 Manifold objekty

Před dalším popisem hraniční reprezentace je nutné si definovat základní pojmy, protože hraniční reprezentace je z praktického hlediska příliš široká. Dovoluje totiž popsat i objekty, které nelze ve skutečnosti vyrobit. Proto se zavádí pojem *manifold* pro modely těles, které lze skutečně vyrobit. Někdy se tyto objekty také nazývají *2-manifoldy*.

Důvodem proč může hraniční popis určovat i nevyrobitelné těleso, vychází z použití matematické a geometrické abstrakce. V matematice a geometrii si snadno dokážeme představit přímku, která je nekonečně tenká nebo že se dva objekty dotýkají právě v jednom bodě. Ve skutečném světě však není možné vyrobit nekonečně tenké lanko, ani nedokážeme spojit dvě tělesa ideálně bodovým svárem. Obrázek 2 vpravo ukazuje příklad nevyrobitelného tělesa nazývaného *nonmanifold*.

Manifold objekt je vyrobitelný a je tvořeny vzájemně propojenými hraničními stěnami. Každý bod stěny má v množině sjednocených stěn otevřené okolí (topologie kruhu) a pro každou hranu platí, že sdílí pouze dvě stěny (obr. 2 vlevo).

Nonmanifold objekt je nevyrobitelný, hrana resp. vrchol spojuje dvě části tělesa. Bod stěny nemá v množině sjednocených stěn otevřené okolí.



Obr. 2 2-manifold a nonmanifold objekt

2.1.1.2 Eulerovy rovnice

Kontrolovat topologii mnohostěnů umožňují Eulerovy rovnice. Mnohostěn je těleso, které je ohraničeno množinou mnohoúhelníkových stěn, každou sdílí vždy sudý počet stěn (v případě manifoldu sdílí každou hranu dvě stěny) a splňuje další podmínky. Jednoduchý mnohostěn je těleso, které lze volnou plastickou deformací převést na kouli, je to těleso bez děr.

Hraniční reprezentace jednoduchého mnohostěnu splňuje Eulerovu rovnost, která udává vztah mezi počtem vrcholů (V, vertex), hran (E, edge) a stěn (F, face):

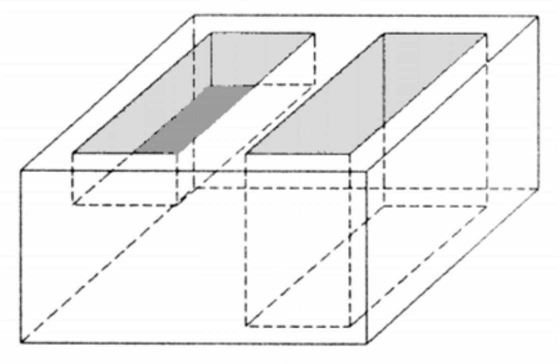
$$F + V = E + 2 .$$ Eulerova rovnost, pro jednoduché objekty bez otvorů. (2.1.1.2.a)

Rovnice platí pro manifold tělesa, kde pro všechny hrany, vrcholy a stěny platí:

- hrana spojuje dva vrcholy
- ve vrcholu jsou minimálně tři hrany
- stěny se neprotínají

Pro manifoldy, které mají otvory, platí zobecněná Eulerova rovnost (2.1.1.2.b). Mezi otvory se nepočítají slepé prohlubně. V obecné Eulerově rovnosti jsou přidány další členy, a to počet vnitřních smyček hran (R, ring), počet oblastí neboli samostatných komponent tělesa (C, component) a počet otvorů procházejících tělesem (H, hole). Vnitřní smyčky vymezují otvory ve stěnách, nikoli otvory v tělese.

$$F + V = E + 2 \cdot (C - H) + R .$$ Obecná Eulerova rovnost. (2.1.1.2.b)



$$F + V = E + 2 \cdot (C - H) + R$$
$$15 + 24 = 36 + 2 \cdot (1 - 1) + 3$$
$$39 = 39$$

Obr. 3 Ukázka výpočtu Eulerovi rovnosti na modelu s děrami

2.1.1.3 Trojúhelníky a jejich sítě

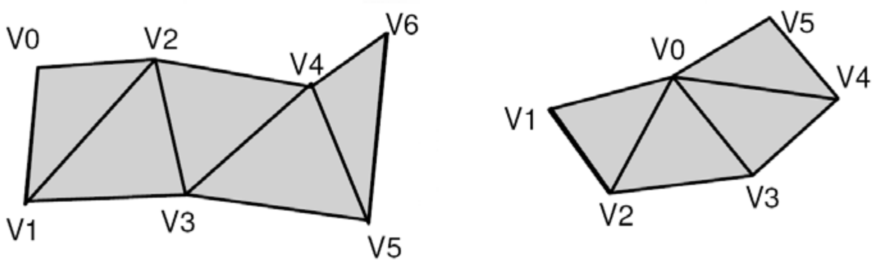
Díky své jednoduchosti a mnoha dobrým vlastnostem je trojúhelník oblíbeným stavebním kamenem většiny reprezentací. Na rozdíl od obecných mnohoúhelníků je vždy konvexní a všechny jeho vrcholy leží v rovině. Pro jeho vyplňování jsou vymyšleny velmi rychlé algoritmy jako např.: Pinedův algoritmus, inverzní řádkové vyplňování a další. Zobrazení je podporováno grafickým procesorem. Mnoho geometrických výpočtů nad trojúhelníkem lze optimalizovat, například výpočty průsečíku

paprsku s trojúhelníkem, a proto se někdy místo výpočtu průsečíku se složitým objektem tento objekt převádí na ploškovou reprezentaci a výpočty průsečíků se provádějí s ní.

Sítí trojúhelníků je nazývána množina trojúhelníků, které sdílejí své hrany. Datová struktura popisující síť bývá rozdělena do dvou logických částí geometrické a topologické. V geometrické části jsou zaznamenány souřadnice vrcholů trojúhelníků, topologická část udržuje údaje o tom, které vrcholy tvoří trojúhelník, případně o tom, které trojúhelníky spolu sousedí. Toto rozdělení je praktické pro některé operace se sítí, např. při geometrických transformacích se vypočítávají nové souřadnice vrcholů. Ty poskytuje právě geometrická část sítě nezávisle na části topologické.[Žára]

Samotná trojúhelníková síť není výhodná pro modelování tvaru těles, protože při reprezentaci klademe důraz na přesné a úsporné vyjádření tvaru, který síť vyjadřuje. A proto modelovací programy obyčejně pracují s NURBS křivkami, Bézierovými křivkami atd. Při převodu do sítě trojúhelníků je třeba síť optimalizovat tak, aby při co nejmenším počtu trojúhelníků vyjadřovala co nejpřesněji vymodelovaný tvar.

Oddělení geometrických a topologických údajů do dvou datových struktur je pro některé úlohy nevyhovující. Například při zpracování dat v grafickém procesoru je většinou nutno popsat síť pouze pomocí jedné lineární struktury (pole), přitom však tak, aby byl minimalizován počet operací prováděných s jednotlivými vrcholy sítě. Toho dosáhneme výběrem trojúhelníků do posloupnosti, která odpovídá pruhu trojúhelníků (triangle strip obr. 4 vlevo). Toto uspořádání zajišťuje, že každý vrchol v pruhu bude zpracován právě jednou. Datová struktura je v tomto případě posloupností souřadnic vrcholů. Každý vrchol tvoří spolu s předchozími dvěma vrcholy jeden trojúhelník. Obdobně lze trojúhelníky uspořádat do vějíře trojúhelníků (trianlge fan obr. 4 vpravo). Vějíř vznikne například při triangulaci konvexního polygonu.



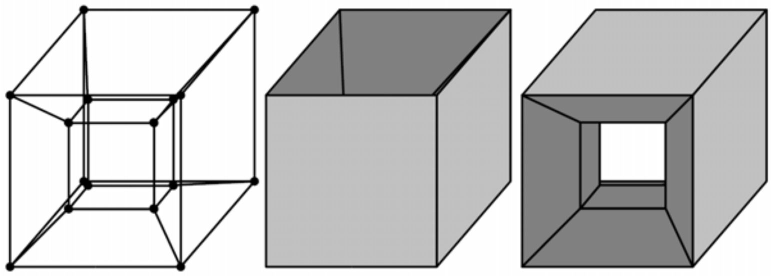
Obr. 4 Triangle strip a Triangle fan

Nalezené optimální množiny pruhů či vějířů trojúhelníků pro libovolnou síť trojúhelníků je náročný výpočetní problémem, který souvisí nejen s charakterem zpracovávané sítě na vstupu, ale především s různými požadavky kladenými na výstup (počet pruhů, pročet trojúhelníků v pruzích apod.). Mezi další praktické požadavky při zpracování trojúhelníků patří algoritmy zjednodušující trojúhelníkovou síť (tzv. decimace).

Trojúhelníky a jejích sítě mají také své nevýhody. Mezi ně patří zejména nesnadné mapování textur, u něhož je třeba explicitně určit vztah mezi obrázkem textury a každým vrcholem trojúhelníka. Další nevýhodou je tzv. geometrický alias. Tento nežádoucí jev se projevuje při změně měřítka. Těleso které bylo původně pokryto několika samostatnými sítěmi dotýkajícími se na hranách, se při změně měřítka může opticky roztrhnout, neboť dojde k numerické chybě vlivem konečné reprezentace čísel v paměti počítače.

2.1.1.4 Drátový model (wire-frame model)

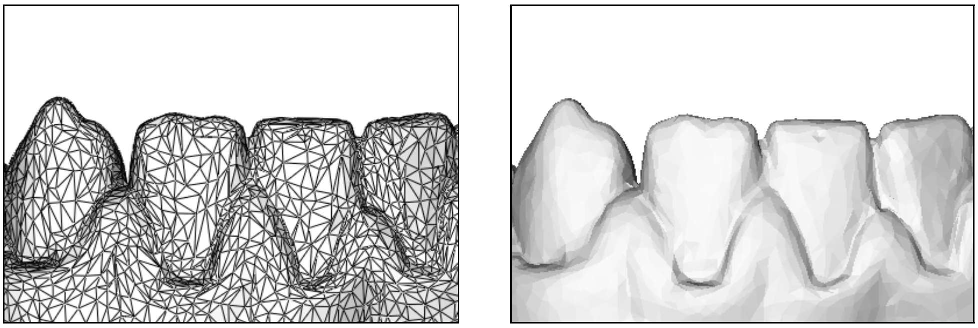
Drátový model je nejstarší a nejjednodušší metodou pro popis hranice tělesa. Reprezentace spočívá pouze v zápisu vrcholů (body) a hran (úsečky, křivky). Tato informačně chudá hranová reprezentace připomíná prostorové drátové modely těles pro školní účely. Drátová struktura obsahuje minimum topologických informací, a proto nelze tento model jednoznačně interpretovat. Prosté vykreslení všech hran modelu (obr. 5 vlevo), neposkytuje dostatečnou informaci o tvaru tělesa ani o jeho pozici vzhledem k pozorovateli. Jeden model může být interpretován jako několik různých těles (obr. 5 uprostřed a vpravo). Samotné vykreslení hran je však praktické a rychlé pro vytvoření náhledu, s jehož pomocí lze prozkoumat vnitřek tělesa, který by jinak byl zakryt hraničními plochami.



Obr. 5 Nejednoznačnost drátového modelu

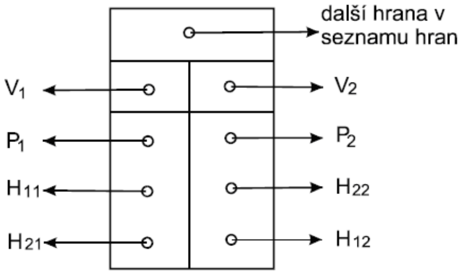
2.1.1.5 Polygonální modely

Polygonální model uvažuje hranici tvořenou plochami. Reprezentace je přirozeným rozšířením předchozího modelu složeného pouze z vrcholů a hran o další vrstvu ploch. Plochy mohou být uspořádány do sítě trojúhelníků nebo obecných polygonů. Obvykle pod pojmem polygonální model je myšleno model tvořený sítí trojúhelníků, protože jak již bylo vysvětleno mají trojúhelníky a sítě trojúhelníků mnoho dobrých vlastností pro počítačovou grafiku.



Obr. 6 Ukázka polygonálního modelu

Asi nejznámější datová struktura pro uložení polygonálních modelů navrhl Bruce G. Baumgart. Je známá pod názvem *okřídlená hrana* (winged-edge), neboť grafické znázornění jedné hrany společně s prvky, které s ní sousedí, připomíná křidélka. Datový záznam hrany obsahuje ukazatele na sousední geometrické elementy (plochy, hrany, vrcholy). Všechny okřídlené hrany jsou zřetězeny. Na nejnižší úrovni je seznam vrcholů. Na střední úrovni je seznam okřídlených hran a na nejvyšší úrovni seznam ploch daného tělesa.[Baum]



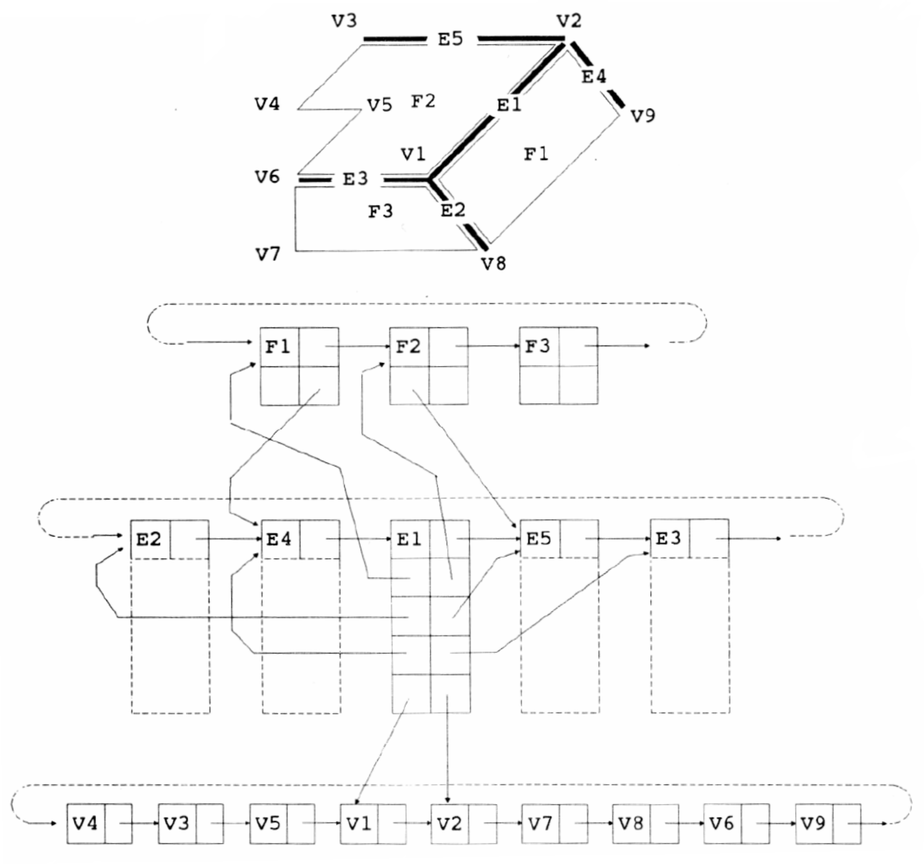
Obr. 7 Datová struktura hrany

Na obrázku 7 je schéma záznamu pro okřídlenou hranu. Kromě odkazu na oba koncové vrcholy (V1, V2) v něm nalezneme ukazatele na sousední plochy (P1, P2). Navíc obsahuje informace o dalších čtyřech hranách. V levé části schématu jsou hrany sousedící s levou plochou (H11, H12), v pravé části hrany sousedící s pravou plochou (H22, H12), a to s ohledem na orientaci těchto ploch. Horní ukazatel pak reprezentuje spojnici na libovolnou následující hranu v zřetězeném seznamu okřídlených hran.

Seznamy vrcholů a ploch jsou mnohem jednodušší. Záznam každé plochy obsahuje ukazatel na libovolnou z jejích hran, případně na hranu patřící k vnitřní hranici, volitelně pak normálový vektor, barvu či jinou materiálovou charakteristiku. Ze záznamu okřídlené hrany je možno odvodit mnoho topologických údajů. Lze například snadno nalézt:

- plochy sousedící s danou plochou
- plochy incidující s danou hranou
- plochy stýkající se v daném vrcholu
- vrcholy a hrany dané stěny

Okřídlená hrana reprezentuje typické vlastnosti datových struktur používaných v trojrozměrné grafice. Samotné geometrické údaje, tj. souřadnice vrcholů umístěné v seznamu vrcholů, zabírají přibližně jen 25% paměti. Zbývající tři čtvrtiny objemu struktur popisujících těleso jsou použity k popisu topologie. Datová struktura okřídlená hrana je určena pouze pro manifoldy, neboť každá (okřídlená) hrana inciduje právě se dvěma stěnami. Pro reprezentaci nonmanifoldů se používá rozšíření datové struktury.

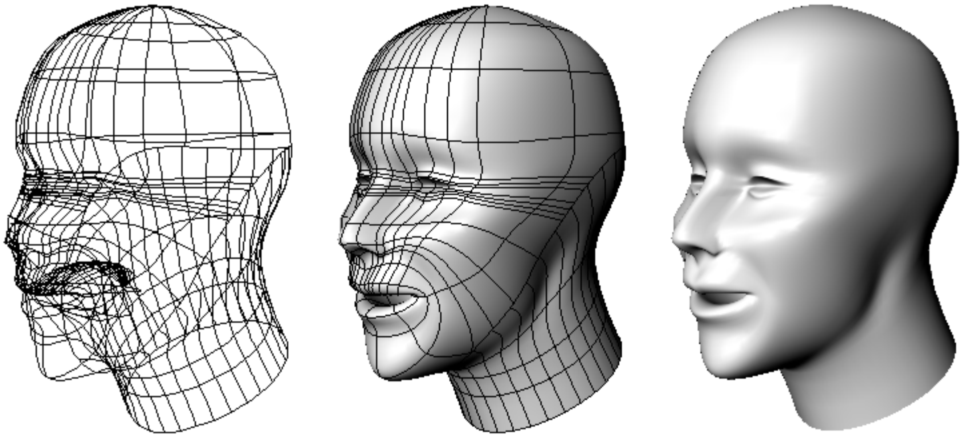


Obr. 8 Příklad okřídlené hrany

2.1.1.6 Plošné spline modely

Motivy, které vedly k rozvoji spline modelů, jsou charakterizovány především snahou odstínit uživatele od matematiky a zaručit maximální jednoduchost tvorby generovaných ploch. Uživatel tak obyčejně zadává řídící body, váhy, či tečné vektory a jejich pomocí modeluje tvar výsledné plochy. Takto popsaný model nese úplnou informaci pro popis objektu, jeho přesnost je dána aproximací spline ploch (NURBS, B-Spline, Bézier …).

Nejjednodušší metodou jak zobrazit spline plochy je jejich polygonizace postupným dosazování do vzorců pro výpočet ploch. Tímto způsobem lze získat síť trojúhelníků, kterou můžeme zobrazovat běžnými metodami pro polygonální modely.



Obr. 9 Spline model lidského obličeje

2.1.2 Konstruktivní geometrie (CSG)

V oblasti CAD se často tělesa popisují způsobem, který odráží postupy používané konstruktérem při navrhování tvaru těles. Metoda nazývaná konstruktivní geometrie těles (CSG, Constructive Solid Geometry), je založena na reprezentaci těles stromovou strukturou (CSG stromem), uchovávající historii dílčích konstrukčních kroků. Z jednoduchých geometrických objektů, tzv. CSG primitiv, je pomocí množinových operací a prostorových transformací vytvořen výsledný objekt. Jako primitiva slouží jednoduchá tělesa (kvádr, koule, válec, kužel, jehlan či toroid), lze však použít i abstraktní entity. [Kršek06]

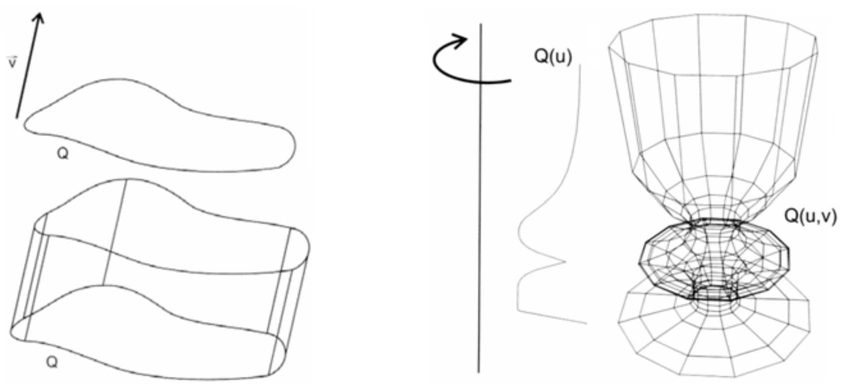
Množinové operace odpovídají konstrukčním postupům a mohou být prováděny jak s CSG primitivy, tak s celými CSG stromy. Zjednodušeně lze říci, že sjednocení představuje spojování součástí (svaření), rozdíl je obdobou vrtání a průnik odpovídá odříznutí či zbroušení. Nevýhodou CSG modelu je poměrně složité nalezení hraniční reprezentace.

2.1.3 Šablonování

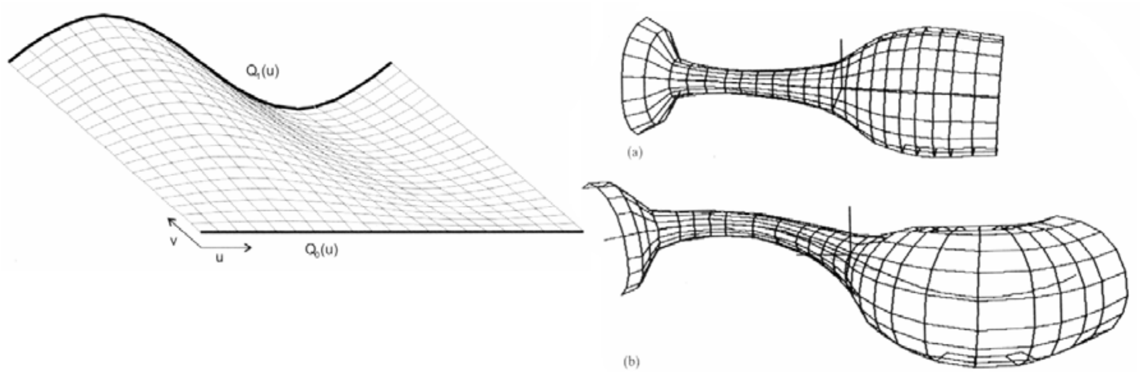
Šablonování (sweeping) je poměrně silnou moderní modelovací technikou, při které získáváme plochu posunem dvourozměrného obrysu (tzv. profilu) po trojrozměrné křivce (tzv. páteři). Někdy se šablonování bere jako součást CSG operací, primitiva jsou tvořena šablonováním. Pro generování šablonové plochy lze s výhodou použít invariance spline křivek a ploch vůči lineárním transformacím. Šablonování rozdělujeme do tři kategorii.

2.1.3.1 Translační

Při translačním šablonování je obrys tažen po úsečce. Tímto způsobem tedy vzniká přímková plocha, která se v jednom směru skládá z úseček. Příkladem takovéto plochy je např. válcová plocha hranolová plocha, nebo plocha uvedena na obrázku 10 vlevo.



Obr. 10 Translační a rotační šablonování



Obr. 11 Šablonování potahováním

2.1.3.2 Rotační

Rotační plochy získáme z rovinné profilové křivky (obr. 10 vpravo) rotací okolo osy. Řezem rotační plochy libovolnou rovinou kolmou na osu rotace je tedy kružnice. Pro reprezentaci rotační plochy (kružnice) se nejčastěji využívají NURBS křivky. S technologií NURBS je k dispozici široký repertoár základních tvarů, od úseček a lomených čar, přes Hermitovské a Bézierovy křivky až po neuniformní neracionální B-spline křivky.

2.1.3.3 Potahování

Šablonování potahováním není interpolací izolovaných bodů v prostoru nýbrž interpolací křivek jak ukazuje obrázek 11.

2.1.4 Implicitní plochy

Přestože geometrické modely objektů používaných v počítačové grafice jsou založeny především na parametrickém vyjádření, praktické aplikace používají i modelování, založené na implicitním vyjádření objektu.

Implicitní plocha je množina bodů P ve třírozměrném prostoru, pro které platí:

$$F(P) = 0$$

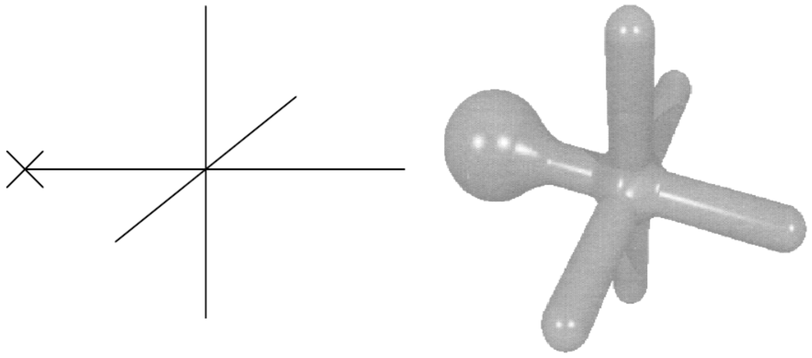
$$F(P) = \sum_{i=1}^{n} c_i F_i(r_i) \cdot$$

n - počet prvků kostry

c_i - velikost potenciálu prvku i

r_i - vzdálenost bodu P od prvku i

F_i - směšovací funkce



Obr. 12 Implicitní plocha vzniklá ze tří úseček a jednoho bodu

Potenciální funkce závisí pouze na vzdálenosti v prostoru. Každý prvek kostry je obklopen vlastním polem, jehož intenzita je maximální na povrchu a klesá se vzdáleností od prvku.

Modelování pomocí kostry má mnoho výhod. Kostra je poměrně intuitivní aproximací mnoha objektů reálného světa (např. lidské tělo má kostru) a je snadno zobrazitelná. Modelování je tedy názorné, relativně snadné na rozdíl od klasického parametrického modelování. Kostra obsahuje zhuštěnou informaci. Nejedná se o plnou reprezentaci, ale na základě kostry implicitního tělesa, je možné získat poměrně slušnou představu o modelovaném objektu. Podobně jako u modelů CSG vzniká složité těleso kombinací těles jednoduchých.

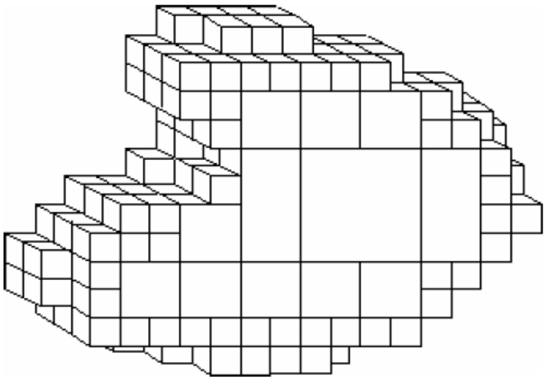
Zobrazení implicitních ploch probíhá převážně pomocí převodu na hraniční reprezentaci. Převody jsou založeny na hledání bodů na hranicích. Další možností zobrazení je použití metod sledování paprsku.

2.1.5 Dekompoziční modely

V některých případech není k dispozici geometrický popis těles, ale pouze sada vzorků v určitém místě povrchu či objemu. Tato sada může obsahovat rozptýlená data, kde každý vzorek z množiny dat nese kromě údajů o jasu, hustotě či jiné fyzikální veličině v daném bode také své souřadnice. V jiných případech může být strukturovaná do podoby pravidelných nebo nepravidelných mřížek. Rozptýlení vzorků do pravidelné pravoúhlé mřížky je typické například pro data získaná pomocí počítačových tomografů, nepravidelný tvar mají data získaná při simulaci proudění kapalin a rozptýlená data jsou výsledkem meteorologických měření teploty a tlaku vzduchu.

U dat uspořádaných do mřížky je důležitý tvar mřížky (domény). Dělení geometrických tvarů mřížky navrhli Spray a Kennon, kteří rozdělili obvyklé tvary do sedmi tříd. Obvykle se využívá kartézská mřížka (prostor je dělený rovnoměrně na krychličky). Nejmenší element ve trojrozměrném

diskrétním prostoru se nazývá voxel a je jako analogie dvourozměrného pixelu. Voxel je vyplněný kvádr, který má v celém svém objemu konstantní hodnotu. Jako hodnota v poloze mezi středy voxelů se obvykle bere hodnota voxelu, který je nejblíže.



Obr. 13 Příklad dekompozičního modelu

2.2 Medicínská obrazová data

Dnes mohou lékaři vyšetřit pacienta, bez jediného řezu skalpelem pomocí medicínských zobrazovacích systémů, např. počítačové tomografie (angl. *Computer Tomography - CT*) nebo magnetické rezonance (*Magnetic Resonance - MR*).

Moderní CT/MR zařízení umožňují pořizovat nejen standardní rovinné (2D) snímky (řezy), ale celé série prostorově (3D) navazujících rovinných snímků. Získávají se tak 3D informace o situaci, struktuře a vlastnostech lidských tkání a kostí v nasnímaném úseku těla pacienta. Tato data jsou dnes nejčastěji ukládána ve standardním komunikačním a datovém formátu DICOM 3.0, který zachovává 100% kvalitu dat. Tyto nasnímané (3D) navazující rovinné snímky se musí upravit do podoby 3D počítačového modelu, aby bylo možné s nimi snadno pracovat v medicínských aplikacích. Tvorba 3D modelů lidských tkání a kosti je založena na segmentových datech z CT/MR. Postup převodu probíhá ve čtyřech krocích:

1. Vektorizace – proces převodu diskrétních dat na trojúhelníkové povrchové modely. Dochází ke změně typu reprezentace dat, diskrétní na vektorový. Nejčastěji je používána metoda Marching Cubes. Tato metoda je schopna tvořit regulérní geometrické modely zcela automaticky a nezávisle na složitosti geometrie objektů viz. kapitola 2.2.1.
2. Vyhlazení – z důvodu vektorizace jsou geometrické modely vrstevnaté (hranaté). Pro odstranění tohoto artefaktu je nutné geometrické modely vyhladit.
3. Decimace – Modely vytvořené vektorizací se vyznačují velkým počtem malých trojúhelníků. Proto se provádí odstranění přebytečných trojúhelníků při maximálním zachování geometrie. Snižuje se tak datová náročnost a tím zvyšuje použitelnost modelů v dalších aplikacích
4. Uložení – Výsledná trojúhelníková síť se obvykle ukládá do souboru formátu STL, který je jedním ze základních formátů pro uložení trojrozměrných modelů tvořených trojúhelníky.

Detailní informace o segmentačních technikách obrazu a převodu CT/MR dat lze získat ve [Španěl] a [KršekPDF].

Obr. 14 Geometrický model lidské lebky

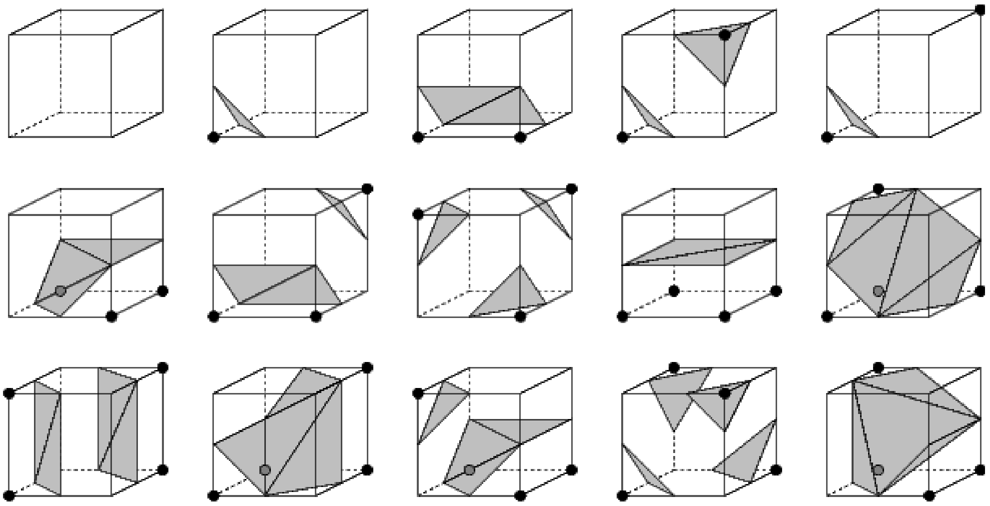
(Zleva: výsledek vektorizace metodou Marching Cubes, uprostřed vyhlazený model a vpravo model po redukci počtu polygonů. Obrázek použit z práce [Španěl] dodány P. Krškem, který se intenzivně věnuje problematice tvorby 3D modelů tkání a jejich klinickým aplikacím na Ústavu počítačové grafiky a multimédií FIT VUT v Brně.)

2.2.1 Vektorizace prostorových řezů

Velmi častým postupem vektorizace prostorových řezů je pomocí aproximace povrchu sítí trojúhelníků. Povrchy se obvykle předpokládají v místech s konstantní hodnotou vzorku, proto se takto vzniklé sítě nazývají izoplochy. Nejznámější a nepopulárnějším algoritmem na vytvoření trojúhelníkové sítě z prostorové mřížky je metoda pochodující kostky (Marching Cubes).

Algoritmus Marching Cubes byl publikovaný v roce 1987. [Baum] Vstupem do algoritmu je prostorová mřížka neboli soustava řezů zkoumaného objektu. Algoritmus nepracuje s celým objemem dat najednou, ale pro jednu vrstvu krychlí je nutné algoritmu načíst řezy po dvojicích. Krychli tvoří vždy čtyři sousední voxely z obou pater. Po sestavení je pole krychlí procházeno a zpracováváno. Hodnoty intenzit vrcholů jednotlivých krychliček (tj. intenzit 8 voxelů z nichž je krychle složena) je porovnávána s prahovou hodnotou (konstantní či zadanou) a ohodnocena jako vnitřní nebo vnější. Jestliže je všech osm vrcholů ohodnoceno stejně (vně či uvnitř), pak se dalšího zpracování neúčastní.

Vzniklých osm hodnot je reprezentováno jako číslo a je použito k nalezení odpovídající posloupnosti vrcholů úhelníků výsledné plochy. Základních případů konfigurací krychle je patnáct (obr. 15), po otočení kolem os a invertování vznikne 256 možných případů. Vypočítané číslo je použito jako index v uspořádané 256-prvkové tabulce obsahující v každém řádku správné posloupnosti vrcholů. Vrcholy musí být předem správně uspořádány, aby nedocházelo ke světelným kolizím u zobrazeného objektu. Je nutné správně vypočítat normály na plochách v jednotlivých konfiguracích a podle nich seřadit vrcholy úhelníků. Na základě zjištěných posloupností vrcholů úhelníků se vytváří síť trojúhelníků.



Obr. 15 Základní případy konfigurace vrcholů krychle u algoritmu Marching cubes

Existuje mnoho dalších metod triangulace povrchu. Za zmínku stojí další jedna z nejjednodušších metod a to je algoritmus Marching Tetrahedas (v překladu "Pochodující čtyřstěny") je to varianta, která se snaží zamezení nejednoznačnosti při triangulaci povrchu. Místo umísťování vrcholů trojúhelníků na hranách krychličky se krychle rozdělí na pět čtyřstěnů a trojúhelníky se umísťují do nich. Tento algoritmus řeší problém děr, které vznikají algoritmem Marching Cubes, ale za cenu dalšího neúměrného nárůstu počtu trojúhelníků [Kršek06].

2.3 Transformace

Pokud již umíme popsat model je třeba s tímto modelem umět efektivně manipulovat. A právě k manipulaci s grafickými objekty slouží geometrické transformace, které jsou jednou z nejpoužívanějších operací v počítačové grafice. Transformace lze rozdělit do dvou kategorií lineární a nelineární. V této práci se budeme zabývat pouze lineárními transformacemi mezi které patří posunutí, otáčení, změna měřítka, zkosení a operace vzniklé jejich skládáním. S nelineárními transformacemi se v počítačové grafice setkáváme při složitějších změnách tvaru grafických objektů, např. u deformací prostorových modelů nebo u warpingu obrazu. S transformacemi se v počítačové grafice nejčastěji setkáváme při převodu trojrozměrné scény do roviny obrazu nebo při transformaci objektů ve scéně.

Aby bylo možné provádět transformace s objekty, je nutné, aby objekty byly popsány svými souřadnicemi, které jsou vztaženy ke zvolenému souřadnému systému. Geometrické transformace mohou být aplikovány na jednotlivé souřadnice objektu, které tak mění svou polohu.

Transformací bodu P, který ve třech rozměrech má kartézské souřadnice [X, Y, Z] získáme bod P' o souřadnicích [X', Y', Z']. Transformací objektu rozumíme aplikaci transformace na všechny body, ze kterých se objekt skládá.

2.3.1 Homogenní souřadnice

Před dalším popisem transformací souřadnic bodů v prostoru je nutné se seznámit se způsobem zadávání souřadnic. V počítačové grafice se používají homogenní souřadnice pro výpočty transformací a to z několika důvodů. Homogenní souřadnice umožňují vyjádření nejčastěji používaných lineárních transformací pomocí jediné matice. Další často používanou transformací vyjádřitelnou pomocí matice v homogenních souřadnicích je perspektivní promítání. Skládání transformací se realizuje jako násobení matic, inverzní transformace je reprezentována inverzní maticí, atd.

Uspořádaná čtveřice čísel [x, y, z, w] představuje homogenní souřadnice bodu P v trojrozměrném prostoru s kartézskými souřadnicemi [X, Y, Z], pro které platí:

$$X = \frac{x}{w},\ Y = \frac{y}{w},\ Z = \frac{z}{w},\ w \neq 0.$$

Bod P je svými homogenními souřadnicemi určen jednoznačně. Souřadnici w též nazýváme váhou bodu. Homogenní souřadnice transformovaného bodu P' s kartézskými souřadnicemi [X', Y', Z'] označujeme [x', y', z', w']. Často se volí w = 1, potom jsou homogenní souřadnice bodu [X, Y, Z, 1]. Toho se dobře využije např. při rozdílu dvou bodů A a B dostaneme vektor $\vec{p} = (a_0 - b_0, a_1 - b_1, a_0 - b_2, 0)$, sečtením bodu a vektoru dostaneme bod.

Obecnou matici 4 x 4 reprezentující lineární transformaci bodu P = [x, y, z, w] na bod P' = [x', y', z', w'] označujeme obvykle písmenem **A**, její speciální případy pak podle druhu transformace, např. M(move), R(rotation), S(scale), Sh(shear). Lineární transformace jsou vyjádřeny vztahem P' = P · A, kde P je bodem, který transformujeme maticí A.

$P = [x, y, z, w]$ - vrchol

$P' = [x', y', z', w']$ - transformovaný vrchol

$P' = P \cdot A$ - vyjádření transformace

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}$$ - transformační matice 4 x 4

2.3.2 Posunutí ve 3D

Transformace posunutí nebo také translace (translation, move) bodu P = [X, Y, Z] je určena vektorem posunutí $\vec{V} = (V_X,\ V_Y,\ V_Z)$. Matice transformace posunutí A_M a k ní inverzní matice A_M^{-1} mají tvar:

$$A_M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ V_X & V_Y & V_Z & 1 \end{bmatrix} \quad A_M^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -V_X & -V_Y & -V_Z & 1 \end{bmatrix}$$

2.3.3 Rotace kolem osy ve 3D

Otáčením (rotation) bodu P okolo jedné z os soustavy souřadnic o úhel α získáme bod P'.

Rotační matice A_R kolem osy X má tvar:

$$A_{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{Rx}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotační matice A_R kolem osy Y má tvar:

$$A_{Ry} = \begin{bmatrix} 1 & \cos\alpha & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{Ry}^{-1} = \begin{bmatrix} 1 & \cos\alpha & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotační matice A_R kolem osy Z má tvar:

$$A_{Rz} = \begin{bmatrix} 1 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_{Rz}^{-1} = \begin{bmatrix} 1 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.4 Rotace kolem obecné osy ve 3D

Pro rotaci kolem obecné osy lze použít složení několika dílčích transformací kolem os x, y, z, nalezení příslušných transformací tedy úhlů, natočení však není jednoduché. Lze však s výhodou využít reprezentaci rotací pomocí kvaternionu, který lze snadno převést na rotační matici.

2.3.4.1 Kvateriony

Kvaterniony byly navrženy irským matematikem W.R.Hamiltonem v 19.století jako analogie komplexních čísel v prostoru. Pro usnadnění si můžeme kvaternion představit jako dvojici složenou z vektorové ($\vec{v}$) a skalární části (s).

$q = (\vec{v}, s) = (x, y, z, w)$ - notace kavaternionu

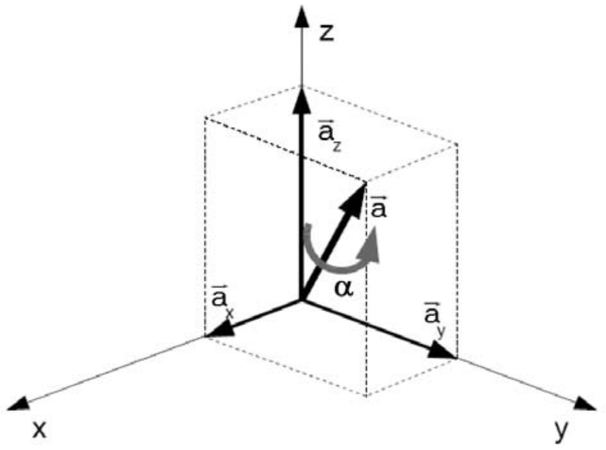
Velikost kvaternionu q je $|q| = \sqrt{x^2 + y^2 + z^2 + w^2}$. Kvaternion, jehož velikost je jedna se nazývá jednotkový kvaternion.

Rotace znázorněná na obrázku 16 je popsána jednotkovým 3D vektorem $\vec{a} = (a_x, a_y, a_z)$, který reprezentuje osu otáčení a úhlem α. Takováto rotace odpovídá jednotkovému kvaternionu definovánu následovně:

$$q = (a_x \sin(\alpha/2),\ a_y \sin(\alpha/2),\ a_z \sin(\alpha/2),\ \cos(\alpha/2))$$

Obvykle se tento zápis zkracuje na:

$$q = a\sin(\alpha/2) + \cos(\alpha/2)$$



Obr. 16 Rotace kolem vektoru $\vec{v}$ o úhel α

Snadno lze tedy převést rotaci kolem obecné osy na kvaternion. Pokud nyní chceme získat rotační matici z vytvořeného kvaternionu stačí použít převod jednotkového kvaternionu $q = (x, y, z, w)$ na rotační matici R:

$$R(q) = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2zw & 2xz+2yw & 0 \\ 2xy+2zw & 1-2x^2-2z^2 & 2yz-2xw & 0 \\ 2xz-2yw & 2yz+2xw & 1-2x^2-2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.5 Změna měřítka ve 3D

Změna měřítka (scale) ovlivňuje současně polohu i velikost transformovaného objektu ve směru souřadnicových os. Pokud je absolutní hodnota koeficientu změny měřítka S v intervalu (0, 1), dochází ke zmenšení a přiblížení transformovaného objektu k počátku souřadnic. Je-li absolutní hodnota koeficientu S větší než jedna, dojde k prodloužení, je-li znaménko koeficientu záporné, dochází k prodloužení či zmenšení v opačném směru.

$$A_S = \begin{bmatrix} S_X & 0 & 0 & 0 \\ 0 & S_Y & 0 & 0 \\ 0 & 0 & S_Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_S^{-1} = \begin{bmatrix} 1/S_X & 0 & 0 & 0 \\ 0 & 1/S_Y & 0 & 0 \\ 0 & 0 & 1/S_Z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.6 Skládání transformací

Pokud chceme objekt posunout a zároveň např. otočit musíme aplikovat více transformací na tento objekt. Při tomto postupném aplikování transformací na body objektu záleží na pořadí, v jakém se transformace provádějí. Je rozdíl, jestliže bod posuneme a poté otočíme okolo počátku souřadnicového systému, nebo zda bod nejprve otočíme a poté provedeme transformaci posunutí. Transformaci vzniklou složením z více transformací lze vyjádřit jedinou maticí, jíž získáme postupným násobením matic, reprezentující dílčí transformace. Protože záleží na pořadí transformací, záleží také na pořadí násobení matic. Jelikož používáme zápis P' = A * P, musíme matice reprezentující pozdější transformace přidávat do složené transformace zleva. Pokud jsou tedy aplikovány transformace v pořadí A_1, A_2, je bod P transformován vztahem P' = A_2 * A_1 * P tyto matice se násobí zprava.

2.4 Reprezentace 3D počítačové scény

Scénou je množina prostorových objektů doplněných dalšími informacemi potřebnými pro jejich zobrazení. Přestože se zdá, že vytvoření scény je poměrně jednoduchým závěrečným krokem po předchozím vymodelování individuálních prostorových objektů, je tvorbu prostorové scény nutné chápat jako samostatnou úlohu. Zatímco systémy pro modelování těles zpracovávají geometrická data definující tvar jednotlivých objektů, systémy pro tvorbu scény předávají k objektům transformace do jejich cílové polohy, určují informace potřebné pro zobrazení a také umožňují do scény opakovaně vkládat stejně nebo podobně vypadající objekty.

Scéna obvykle obsahuje:

- nezobrazované objekty – kamery, osvětlení scény
- zobrazované objekty – jejich geometrie, barevné vlastnosti, textury
- prvky definující logickou strukturu scény – definice skupin a jejich instancí
- transformace – definované hierarchicky kvůli snadnější manipulaci s objekty

Kamerou se nazývá množina informací týkajících se jedné pohledové transformace. Kamera není reprezentována žádným tělesem, neboť představuje virtuálního pozorovatele, který si scénu prohlíží. Z charakteristiky kamery zobrazovací systém získá údaje o umístění pozorovatele, směru jeho pohledu a způsobu promítání. Ve scéně může být umístěno více kamer nastavených tak, aby poskytly pohledy na různá, významná místa ve scéně. Osvětlení scény zahrnuje informace o světelných zdrojích.

Tělesa a další zobrazované objekty jsou vhodně uspořádány do struktury nazývané graf scény. Tato struktura umožňuje logické seskupování těles, efektivně je transformovat a jejich instance vkládat úsporným způsobem do prostoru scény.

2.4.1 Graf scény

Graf scény (scene graph) je n-ární strom, tj. takový graf, v němž lze pro každý uzel nalézt právě jednoho předchůdce. Výjimku tvoří kořen stromu, který stojí na nejvyšší úrovni. Graf scény může obsahovat i několik stromů, tzv. les. Graf scény není vždy definován stejným způsobem, odlišnosti jsou v typech uzlů, v pravidlech pro stavbu stromu i pro interpretaci dat ve stromu uložených. [Žára]

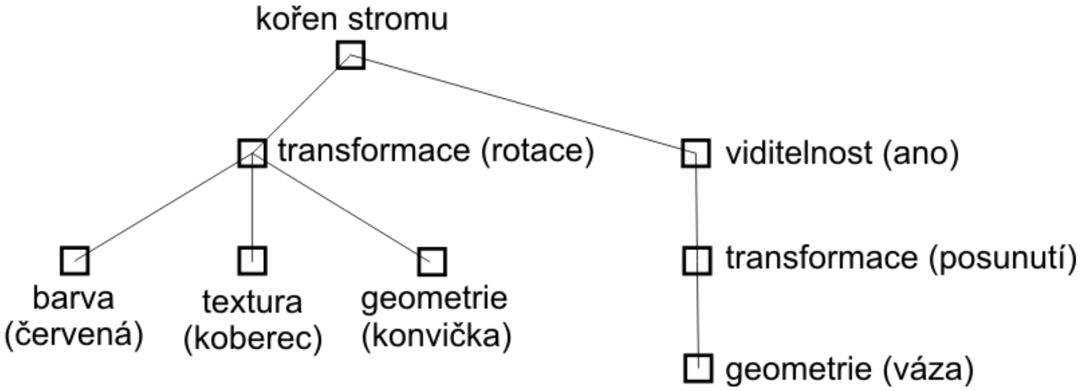
Důležitou vlastností stromu je schopnost vyjádřit vztahy mezi uzly. Jedním z těchto vztahů je dědičnost. Umožňuje v jednom místě stromu definovat vlastnost, která bude platná pro řadu dalších uzlů. Rozsah platnosti je dán vzájemnou polohou uzlů v rámci stromu. Lze například stanovit, že vlastnost definovaná v uzlu je platná pro všechny následníky tohoto uzlu. Některé systémy využívají graf scény pro zadání pevného pořadí následníků uzlu a předpokládávají jejich systematické zpracování, například zleva doprava. To umožňuje rozšířit dědění, resp. sdílení určité vlastnosti i na uzly na stejné úrovni, umístěné v grafu scény vpravo od daného uzlu. Pravidla pro dědění mohou být definována různými způsoby, příklad je uvedený na obrázku 17.

Je zřejmé, že každé těleso může být pevně umístěno do své cílové pozice, tj. veškeré souřadnice tělesa mohou být předem transformovány pomocí posunutí, natočení, případně změny velikosti. Pro manipulaci se scénou je však mnohem vhodnější ponechat tělesa v jejich základních polohách (lokálním souřadnicovém systému) a potřebné transformace zapsat do grafu scény, například v podobě transformační matice. Hierarchické uspořádání scény umožní transformace skládat a změnou jedné transformace ovlivnit celý podstrom.

Skládání transformací se projeví při interpretaci grafu scény. Před vykreslením scény se systematicky prochází celý strom shora dolů a zleva doprava. Při sestupu do nižší úrovně se nalezené transformace ukládají do zásobníku. Naopak při přesunu do vyšší úrovně se transformace odebírají. Jakmile se dojde k uzlu, který uchovává geometrické údaje, aplikuje se na něj transformace složená ze všech transformací mezi ním a kořenem stromu. Při systematickém procházení grafu scény můžeme obdobně zpracovávat i další údaje potřebné k zobrazení objektů, například textury a materiálové vlastnosti. Tyto prvky se však na rozdíl od transformací neskládají, vždy je aplikována naposledy nalezená hodnota.

Graf scény může ovlivnit rozsah působnosti světelných zdrojů a napomoci k rychlejšímu vyhodnocení osvětlovacího modelu. Pokud tvůrce scény ví, že nějaký zdroj světla neovlivní vzhled určité části scény, umístí zdroj světla do podstromu. Díky stromové struktuře je působnost světla logicky omezena jen na část podstromu a v ostatních podstromech není nutné světelný zdroj vyhodnocovat.

Graf scény nemusí vždy odpovídat stromu z obrázku 17. Scénu je možné rozdělit na několik logických sekcí s informacemi stejného typu, např. na skupiny definic textur, skupiny koeficientů odrazivosti, skupinu souřadnic normál apod. Objekty potom obsahují kromě své geometrické reprezentace i sadu ukazatelů do výše uvedených skupin. Z hlediska konkrétního uložení scény je praktické používat více souborů, ať již pro textury či pro definice vzorů. Výhodou je nejen použití různých kompresních metod příslušných určitému typu dat, ale též možnost snadné změny vzhledu scény záměnou obsahu datových souborů. Některé formáty pro uložení scény např. VRML dovolují do scény vkládat i odkazy na vzdálené soubory (světy) umístěné kdekoli na Internetu. Rozdělení grafu scény do více souborů ovšem zvyšuje nebezpečí neúplnosti dat.



Obr. 17 Ukázka grafu scény

3 Návrh 3D grafického systému

Návrh systému vycházel z předpokladu, že se pro implementaci využije objektově orientovaný programovací jazyk C++, tool kit wxWidgets, knihovna pro práci s 3D grafikou OpenGL a knihovna VectorEntity pro práci s trojúhelníkovými sítěmi. Detailní popis knihoven je uveden v kapitole implementace.

Při objektově orientovaném přístupu se program skládá z množství individuálních jednotek nazývaných objekty. Každý objekt je vždy instancí třídy objektu, která definuje vlastnosti a schopnosti objektu (možnosti zpracování dat, posílání a přijímání zpráv od dalších objektů atd.). Tato modularita zvyšuje flexibilitu, zjednodušuje návrh a pozdější provádění změn. Pokud jsou jednotlivé části navržené dobře korespondují s reálnými aspekty problémů.

V této diplomové práci je popsán návrh 3D grafického počítačového systém zaměřeného na modelování ve stomatologii. Navržený systém je realizován jako grafické uživatelské rozhraní. Grafické uživatelské rozhraní (anglicky Graphical User Interface, známe pod zkratkou GUI) je druh komunikace s počítačem mající podobu interaktivních grafických prvků. Tyto prvky lze ovládat klávesnicí nebo jednodušším způsobem myší. V dnešní době se nejčastěji skládá z oken zabírajících větší či menší část počítačového monitoru.

Obr. 18 Základní rozložení prvků v okně 3D systému

Grafické uživatelské rozhraní navrženého grafického systému musí navenek působit přehledně a dát se snadno ovládat, proto jsem rozdělil hlavní okno aplikace do čtyř částí(frames) jak ukazuje obr. 18. První oblast „Menu s výběrem“ slouží pro ovládání a výběr nástrojů. Levá část hlavního okna umožňuje zobrazovat a manipulovat s přehlednou textovou podobu grafu scény. Největší část okna virtuální scéna slouží k zobrazení pohledu na trojrozměrnou virtuální scénu a také její ovládání. Poslední část informační lišta neboli StatusBar ve spodní části okna je nedílnou součástí, každé dnešní aplikace a je využita obvykle pro zobrazení rychlé nápovědy nebo detailnějších informací při

ovládání aplikace. Jednotlivá okna jsou navzájem propojena a mohou mezi sebou komunikovat, a proto změnou některého parametru v jednom okně může dojít ke změně v ostatních oknech.

3.1 Virtuální scéna

Okno virtuální scéna jak již bylo řečeno slouží k ovládání a vykreslování virtuální scény, ve které lékař bude pracovat. Toto okno je navrženo jako nová vizuální komponenta grafické plátno GLCanvas, která je odvozena ze třídy vizuální komponenty wxGLCanvas. Třída wxGLCanvas z tool kitu wxWidgets umožňuje zobrazování OpenGL grafiky.

Nově vytvořená třída grafického plátna zajišťuje ovládání a vykreslování trojrozměrné scény pomocí instance třídy scény GOScene. Ovládání spočívá v interpretaci získaných událostí od uživatele (klávesnice, myš) a následné úpravě parametrů grafu scény dle zvoleného nástroje. Jedna z vlastností třídy je druh projekce (perspektivní, ortogonální) podle kterého se nastaví projekční matice. Dále třída má ukazatel na instanci okna, které zobrazuje textovou podobu grafu scény, aby mohla toto okno ovládat.

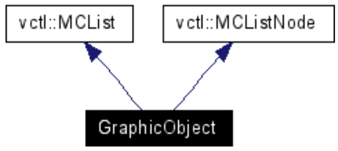
Před dalším popisem navrženého grafického plátna je nutné popsat návrh abstrakce grafického objektu a grafu scény. OpenGL tyto abstrakce neobsahuje, umí (pouze) zobrazovat základní trojrozměrná grafická primitiva a promítat je do dvourozměrného okna.

3.1.1 Návrh základního grafického objektu

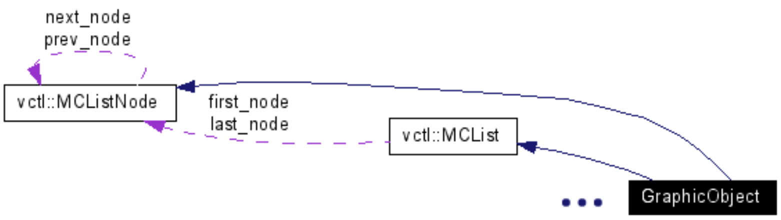
Návrh grafického objektu taky někdy označovaného zkratkou GO vycházel z předpokladu, že vše co se bude chtít zobrazit ve scéně je nutné odvodit z této základní třídy. Proto byla nová třída GraphicObject navrhována s co největším důrazem na její snadné rozšíření.

3.1.1.1 Struktura

Základní předpokladem je, že grafický objekt je tvořen sám sebou a dalšími částmi, které jsou také grafickými objekty. Tento požadavek lze snadno splnit jak ukazuje dědičnost na obr. 19. Grafický objekt je potomkem dvou tříd z knihovny VectorEntity a to MCList a MCListNode. Je tedy využito vícenásobného dědění. Třída MCList zapouzdřuje práci s obousměrně vázaným seznamem, kde položka seznamu je třídy MCListNode. Děděním z třídy MCList umožňuje grafickému objektu práci s obousměrně vázaným seznamem. A protože grafický objekt dědí zároveň třídu MCListNode může být i sám prvkem seznamu. Touto deklarací je dosáhnuto, možnosti přidat do grafického objektu podřízené grafické objekty jako prvky jeho obousměrně vázaného seznamu (obr. 20).



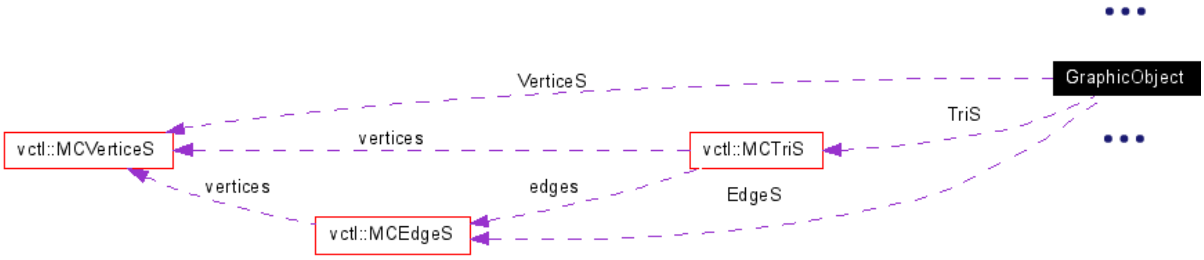
Obr. 19 Diagram dědičnosti grafického objektu



Obr. 20 Část diagramu tříd grafického objektu s obousměrně vázaným seznamem

3.1.1.2 Reprezentace v prostoru

Pro reprezentace grafického objektu v trojrozměrném prostoru je zvolena hraniční reprezentace pomocí sítě trojúhelníků z důvodu její podpory ze strany dnešních grafických akcelerátorů. K uložení trojúhelníkových modelu do grafického objektu, je využito tříd MCVerticeS, MCEdgeS a MCTriS opět z knihovny VectorEntity. MCVerticeS zaštituje uložení a práci se seznam vrcholu, MCEdgeS práci se seznamem hran a MCTriS se seznamem trojúhelníků. Tyto tři třídy jsou navzájem svázány podobným způsobem jako okřídlená hrana popsaná v kapitole 2.1.1.5, takže lze snadno získat např. seznam okolních hran a trojúhelníků k vrcholům. Obrázek 21 ukazuje, že každý grafický objekt má ukazatele na tyto instance tříd, do kterých si ukládá vlastní hraniční reprezentaci.



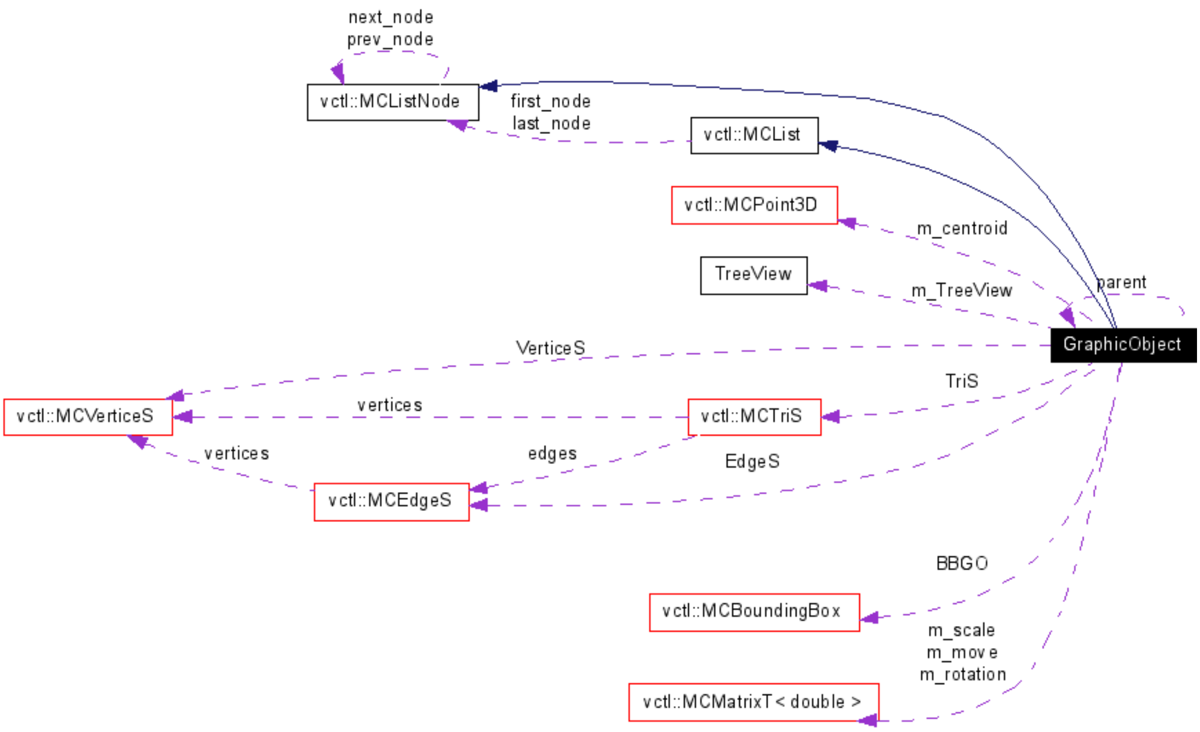
Obr. 21 Datové struktury pro uložení trojúhelníkového modelu v grafickém objektu

Grafický objekt má svou hraniční reprezentaci, ale objektem před jeho zobrazením je třeba umět pohybovat, měnit jeho natočení a velikost. K tomu jsem využil transformace, které jsou reprezentovány maticemi.

3.1.1.3 Transformace

Pro práci s homogenními transformacemi pomocí matic a vektorů jsou navrženy dvě nové třídy MCMatrix a MCVector. MCMatrix zapouzdřuje práci s maticemi (např. výpočet inverzní matice,

výpočet rotační matice z kvaternionu atd.). MCVector zapouzdřuje práci s trojrozměrnou reprezentací bodu a vektoru (skalární součit, vektorový součin vektorů, násobení maticí atd.) Grafický objekt pro každý druh transformace využívá jednu instanci třídy MCMatrix. Pro transformaci posunutí matici m_move, otočení m_rotation a zvětšení matici m_scale (obr. 22 dole).

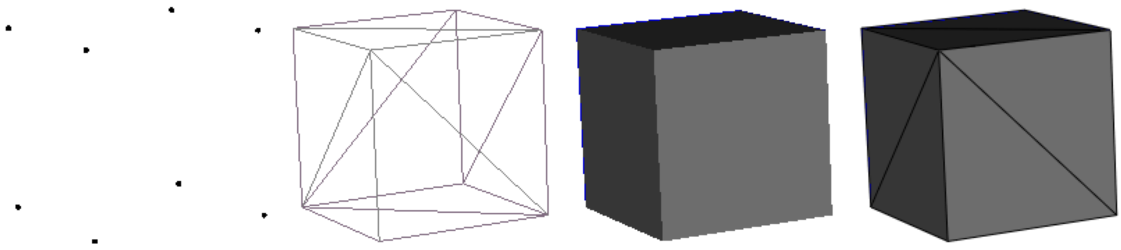


Obr. 22 Kompletní diagram tříd grafického objektu

3.1.1.4 Zobrazení

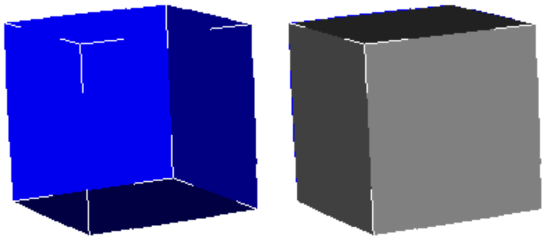
Grafický objekt umožňuje zobrazení trojúhelníkové sítě uložené ve strukturách MCTriS, MCEdgeS a MCVerticeS pomocí příkazů OpenGL. Pro zobrazení sítě jsou navrženy čtyři možné způsoby:

1. reVertices - vykreslení jen bodů
2. reWireframe - vykreslení jako drátový model
3. reSolid - vykreslení z vyplněných trojúhelníků
4. reSolid_Wireframe - vykreslení z vyplněných trojúhelníků se zvýrazněnými hranami



Obr. 23 Navržené způsoby vykreslení trojúhelníkové sítě

Způsob zobrazení je určen hodnotou vlastnosti m_render, která je výčtového typu TRender. Při vykreslování OpenGL dále umožňuje skrývání předních nebo zadních stěn. Skrývání stěn může být v některých situacích výhodné např. pohled skrz přední stěny nebo urychlení vykreslení skrytím zadních stěn, proto i tento parametr je v grafickém objektu zahrnut jako vlastnost m_hidefaces výčtového typu THideFaces (obr. 24).



Obr. 24 Skrytí předních nebo zadních stěn

Aby bylo možné grafický objekt v budoucí scéně jednoznačně identifikovat je jedna z jeho vlastností jméno celočíselného typu. Jména objektů si bude automaticky spravovat scéna. Další navrženou součástí je ohraničující kvádr (anglicky nazývaný bounding box) realizovaný třídou MCBoundingBox. Tento kvádr je zarovnaný s osami souřadného systému scény a objemem pokrývá celý prostor zobrazeného trojúhelníkového modelu. Pro jeho výpočet postačuje seznam vrcholů uložený v seznamu MCVerticeS a je nutné uvažovat transformace objektu, které můžou objekt zvětšit a natočit. Tato struktura se využívá, pro další z vlastností m_select a to je označenost objektu. Pokud má objekt nastavenu vlastnost označen, tak navíc při zobrazení pozmění své barvy tak, aby byl světlejší a nakonec vykreslí pomocí čar svůj obalující kvádr jak ukazuje obr 24.

3.1.1.5 Komunikace s oknem grafu scény

Grafický objekt není pouze zobrazován v trojrozměrné scéně, ale také ve stromu v okně graf scény. Proto má grafický objekt ukazatel na instanci tohoto okna a také identifikátory položek ve stromu, ke kterým umožňuje zobrazení kontextového menu. Popis položek v okně grafu scény je popsán v kapitole okno graf scény.

3.1.1.6 Import trojrozměrného modelu ze souboru

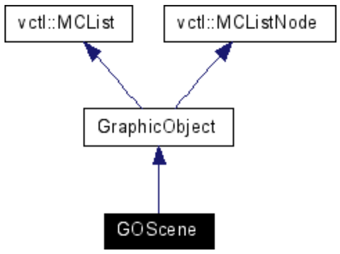
Aby bylo možné snadno do grafického objektu nahrát medicínský model uložený obvykle v podobě souborů v paměti počítače, bylo navrženo čtení z binárního souboru typu STL. Typ STL je jedním z nejjednodušších formátů je podporován většinou 3D grafických aplikací, protože se snadno implementuje jeho čtení a ukládání. Soubory typu STL obsahuje pouze souřadnice vrcholů trojúhelníků a jejich normály. Načtená data ze souboru se uloží do struktur (MCTriS, MCEdgeS a MCVerticeS) v grafickém objektu.

3.1.1.7 Uložení grafického objektu

Všechny grafické objekty, které se budou nacházet ve scéně je nutné umět uložit do souboru, proto každý grafický objekt umožňuje uložit svůj obsah do instance třídy wxString (dynamický textový řetězec). Do textového řetězce se ukládá typ třídy a všechny její vlastnosti. Pokud je do grafického objektu naimportován trojrozměrný model ze souboru, do výsledné textové podoby objektu se ukládá jméno souboru a ne trojrozměrná reprezentace. Pro další rozšířené grafické objekty např. měřící nástroje, reprezentace implantátu atd. je nutné aby tyto nové třídy ukládali do textového řetězce svá rozšíření a svůj typ třídy, aby bylo možné z těchto údajů zpětně vytvořit příslušnou instanci třídy.

3.1.2 Návrh scény

Návrh scény vychází z návrhu grafického objektu. Samotný grafický objekt vytváří strom vzájemných vztahů, protože může mít n-podřízených grafických objektů a tyto mohou mít další atd. Tím způsobem samotný GO vytváří n-ární strom. Přidáním definice ukazatele na rodičovský grafický objekt vzniká kompletně informovaný strom tzn. každý grafický objekt zná svého rodiče a má seznam jemu podřízených objektů. Jediný uzel ve stromu, který nemá svého rodiče je kořen stromu. Nová třída GOScene je odvozena z grafického objektu a představuje kořen a zároveň reprezentuje celou scénu. Třída scény sama neobsahuje žádný polygonální model, slouží jen jako kořen scény a její ovládání.



Obr. 25 Diagram dědičnosti pro objekt scény

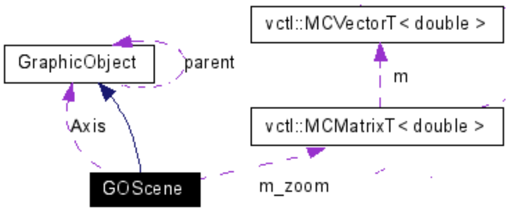
3.1.2.1 Zobrazení scény

Zobrazování scény je proces postupného zobrazování uložených grafických objektů ve stromu. Při zobrazování grafických objektů je nutné, aby se projevily jednotlivé transformace nadřízených grafických objektů i na jejich podřízené objekty, proto se při zobrazování grafických objektů provádí skládání transformací. Při sestupu stromem do nižších úrovní se transformace slučují s předchozími, které se ukládají do zásobníku a aplikují před vykreslením. Při přesunu zpět do vyšší úrovně se transformace odebírají ze zásobníku. Tento návrh lze realizovat využitím knihovny OpenGL, která umožňuje pracovat se zásobníkem transformací příkazy glPushMatrix() a glPopMatrix(). Princip tohoto skládání transformací si lze snadno představit např. na lidské paží. Pokud pohybujeme prsty

měníme pouze jejich pozicí, ale pokud pohneme paži v rameni mění se poloha celé paže s prsty, dlaní atd.

Při návrhu grafického objektu jsme stanovili potřebu tří transformačních matic. Při zavolání zobrazování grafického objektu si sám grafický objekt aplikuje své transformace v pořadí rotace, zvětšení a změna polohy. Toto pořadí není náhodně stanoveno a je klíčové při interpretaci scény. Intuitivně chápeme otočením grafického objektu v trojrozměrném prostoru jako otočením kolem jeho těžiště nebo středu obalujícího kvádru (bounding boxu). Proto jsou souřadnice modelu při jeho vzniku vhodně přepočítány, tak aby střed objektu byl i počátkem souřadného systému [0, 0, 0]. Po této úpravě vrcholů polygonálního modelu se rotační transformace bude chovat intuitivně (otáčení kolem středu modelu). Pokud bychom nejprve aplikovali posunutí a následně rotaci všechna práce se změnou polohy vrcholů modelu by byla ztracena a nedocházelo by k otáčení kolem středu modelu.

Protože objekt scény je odvozen z grafického objektu, dědí i pořadí aplikování transformací před zobrazením. Ve třídě scény(kořene) je aplikování transformací změněno. Nejprve se provádí změna polohy, poté otáčení a následně přiblížení nebo oddálení. Pro přibližování a oddalování je možné využít stávající matici pro zvětšení m_scale, ale z důvodu přehlednosti jsem do návrhu třídy objektu scény přidal novou transformační matice m_zoom. Transformační matice pro zvětšení je v objektu scény nevyužita. Tento návrh umožňuje pohyb ve scéně aniž by bylo nutné měnit polohu kamery.



Obr. 26 Diagram rozšíření tříd scény

3.1.2.2 Osy souřadného systému

Nedílnou součástí každé virtuální scény je zobrazování os (anglicky axis) souřadného systému. Za tímto účelem je navržena třída GOAxis, která je odvozena ze základního grafického objektu. Aby se tato instance třídy nepletla, např. při textovém výpisu scény mezi ostatní objekty. Je GOAxis oddělena od scény, tak že není umístěna v seznamu podobjektů kořene scény, ale scéna má přímo ukazatel na její instanci jak ukazuje obrázek 26. Tímto způsobem se do budoucna ulehčilo mnoho situací, kdy by bylo potřeba rozlišit instanci GOAxis od standardních objektů ve scéně.

3.1.2.3 Uložení scény

Protože každý grafický objekt umí uložit svou textovou reprezentaci jsou tyto reprezentace získány a uloženy podle vztahů určených grafem scény.

3.1.3 Manipulace s objekty a scénou

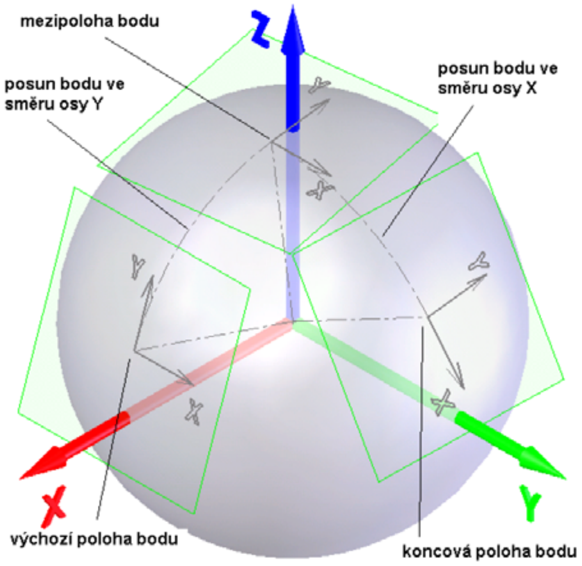
Manipulací s objekty ve scéně se rozumí změna jejich polohy, natočení nebo velikosti. Manipulací scénou chápeme jako změnu její polohy nebo natočení vůči pozorovateli. Tyto manipulace se provádějí změnou transformačních matic uložených v objektech grafu scény.

Manipulace jsou zadávány pomocí změny polohy myši, takzvaným tažením v okně. Tažením myší je myšleno stisknutí tlačítka a následná změna polohy kurzoru. Problém je v tom, že myš se pohybuje po dvourozměrném okně a je nutné stanovit jakým způsobem se projeví tyto dvourozměrné změny souřadnice kurzoru na objekt uložený v grafu scény.

Změnu polohy grafického objektu nebo scény je jednoduché si představit, jako změnu transformační matice udávající posun objektu. Tento posun je upravován tak, aby docházelo k pohybu objektu po rovině v trojrozměrném prostoru rovnoběžné s rovinou pohledu. Posunutí dopředu a dozadu je zajištěno při držení klávesy CTRL. Při těchto změnách je také nutné uvažovat natočení získané z nadřízených objektů(scéna je také nadřízený objekt), protože tyto nadřízené objekty svými rotacemi, také ovlivňují výslednou polohu a natočení svých pod objektů. Vypočítaná změna polohy se ještě upravuje podle tohoto natočení(vynásobením inverzní maticí natočení).

Změna velikosti grafického objektu lze realizovat jednodušeji, protože stačí stanovit koeficient změny velikost a tímto koeficientem upravit matici zvětšení grafického objektu.

Pro rotaci grafickým objektem nebo scénou kolem tří os je navržen matematický aparát zvaný ArcBall. Základní myšlenka je jak dvou rozměrné souřadnice definované pomocí pozice myši převést na souřadnice třírozměrné. Tuto transformaci lze provést tak, že si dvourozměrný pohyb myši představíme jako virtuální pohyb dvou na sebe kolmých vektorů, tvořící tečnou rovinu k povrchu koule. Na tečné rovině je realizován pohyb kurzoru myši a na kouli je pak tento pohyb promítnut. Pro výpočet rotace se využívá vektorový počet. Známe-li tedy polohu středu koule a dva body na kouli můžeme jednoduše stanovit kvaternion, který představuje trojrozměrnou rotaci. Tuto rotaci musíme před jejím přičtením k rotaci objektu upravit podobným způsobem jako při posunu, protože natočení objektu ovlivňují i jeho nadřazené objekty. Poloha koule ArcBallu je volena do středu grafického objektu.

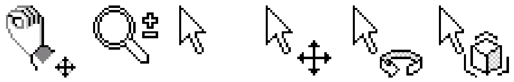


Obr. 27 Pohyb po povrchu koule

Aktuální druh manipulace je určený vlastností m_mode třídy grafického plátna. Pro manipulaci bylo navrženo šest režimů :

1. moShift - posun celé scény
2. moZoom - přibližování a oddalovaní ve scéně
3. moSelect - rotace scénou pomocí ArcBallu
4. moMove - změna polohy vybraného grafického objektu
5. moRotate - rotace vybraného grafického objektu pomocí ArcBallu
6. moScale - změna měřítka vybraného grafického objektu

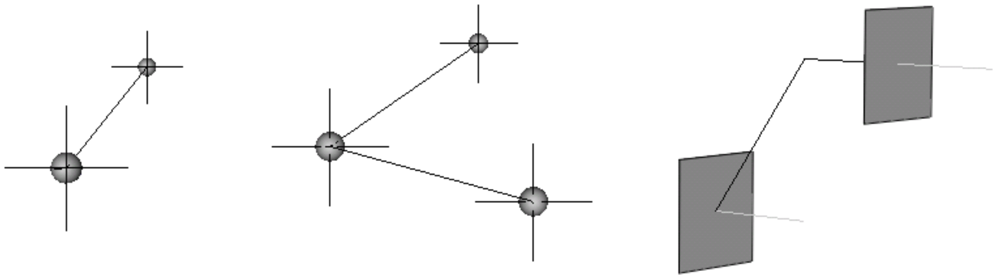
První dva režimy slouží pouze pro ovládání polohy scény. Zbylé čtyři režimy umožňují navíc při stisku levého tlačítka myši změnit aktuálně vybraný grafický objekt nad kterým se provádí zvolená manipulace. Zjištění přítomnosti zobrazeného trojrozměrného grafického objektu na konkrétních souřadnicích plátna je zajištěno pomocí stencil bufferu z OpenGL. Pokud pod kurzorem myši není nalezen žádný grafický objekt, zůstává vybraný původní. Grafické znázornění označení je již navrženo v základní třídě grafického objektu a stačí proto vybranému objektu nastavit vlastnost označení (m_select). Pro rychlejší práci s režimy je na pravé tlačítko myši automaticky volen režim moZoom a na prostřední tlačítko moShift. Aby bylo možné rozpoznat v jakém režimu se plátno nachází, byly jednotlivé režimy odlišeny různými ukazateli kurzoru myši (obr. 28).



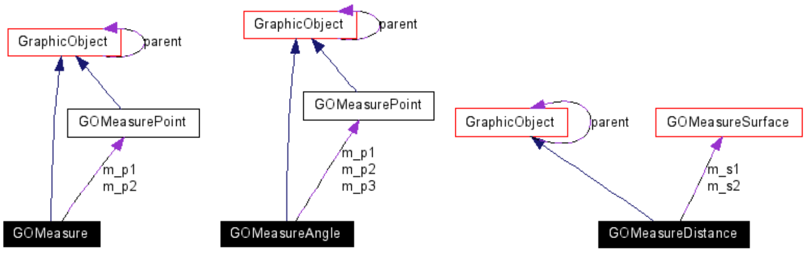
Obr. 28 Rozlišení režimů manipulace kurzorem

3.1.4 Návrh měřících nástrojů

Měřící nástroje jsou objekty, které se zobrazují tak jako ostatní objekty a navíc vypočítávají potřebné veličiny. Podle druhu měřícího nástroje je tento nástroj tvořen několika samostatnými částmi (podobjekty obr. 29). S těmito podobjekty uživatel může standardně manipulovat a nadřízený měřící nástroj při této změně vypočítá měřenou veličinu. V navrženém systému jsou k dispozici nástroje pro měření vzdáleností a úhlů. Měření vzdálenosti mezi dvěma body umožňuje třída GOMeasure, měření vzdálenosti mezi dvěmi rovnoběžnými rovinami třída GOMeasureDistance a měření úhlů třída GOMeasureAngle. Body, které tvoří měřící nástroje jsou třídy GOMeasurePoint a roviny (čtverce) jsou třídy GOMeasureSurface viz obr. 30.



Obr. 29 Měřící nástroje



Obr. 30 Diagramy tříd měřících nástrojů

Podřízené objekty měřících nástrojů jsou odvozeny ze základního grafického objektu. Jsou rozšířeny o předdefinovaný vzhled a zasílání události do nadřízeného objektu při změně jejich vlastností. Měřící nástroj při příchodu události vypočítá měřenou veličinu a má možnost ovlivnit své podobjekty, na které má ukazatele (obr. 30).

3.1.5 Návrh reprezentace zubního implantátu

Návrh třídy zubního implantátu vychází ze vzhledu a funkce dnes používaných implantátu. Zubní implantát(obr. 31) je umělý zubní kořen, který se obvykle zavádí do čelistní kosti. Tyto implantáty se také nazývají nitrokostní implantáty. Slouží k připevnění korunky, můstku nebo snímatelné zubní protézy. Zubní implantát se dá trochu přirovnat ke hmoždince, ale ne tak úplně. Implantát je nejčastěji malý šroubek, který je do kosti zaveden podobně jako vrut do dřeva. Proto se mu říká šroubový implantát. Jeho průměr bývá okolo čtyř milimetrů a délka jeden až jeden a půl centimetru. Je celý

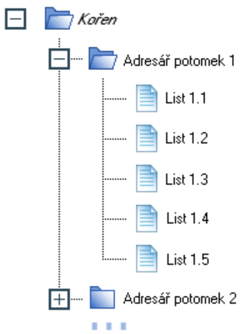
skryt v kosti. Implantát má uvnitř dutinu se závitem, do kterého se po vhojení implantátu zašroubuje pilíř. Ten již vyčnívá do úst. Na něj se potom nacementuje nebo přišroubuje vlastní protetická náhrada, kterou je umělá korunka či můstek, nebo zvláštní zařízení, sloužící k upevnění snímatelné zubní protézy. Implantát sám o sobě tedy žádnou korunku nemá.

Obr. 31 Zubní implantát(jednofázový, dvoufázový) a jeho zjednodušená reprezentace

Z hlediska navrženého systému je zubní implantát model, který se umísťuje do modelů představující čelist nebo její část a je jeho podřízeným grafickým objektem. Na obrázku 31 je vidět tvar implantátu a charakter závitu, které zajišťují stabilitu. Pokud má lékař k dispozici trojrozměrný model implantátu od výrobce může ho načíst do základního grafického objektu a pracovat s tímto objektem. Obvykle má lékař k dispozici pouze parametry implantátů bez jejich trojrozměrných počítačových modelů. Pro vyřešení této situace byla navržena nová třída grafického objektu GOImplant. V navržené třídě je tvar implantátu zjednodušen na podobuválečku bez závitů (obr. 31 vpravo). Tudíž rozšířenými vlastnostmi třídy implantátu jsou délka a poloměr válce. Při změně těchto vlastností dochází ke změně trojúhelníkové sítě tak, aby odpovídala zadaným parametrům.

3.2 Okno grafu scény

Pro zobrazování grafu scény v podobě stromu je navržena nová vizuální komponenta TreeView (obr. 32). Tato třída je odvozena z třídy vizuální komponenty wxTreeCtrl. Komponenta umožňuje práci se stromem kde každá položka je jednoznačně identifikována a může mít zvolenu ikonu a popisek.

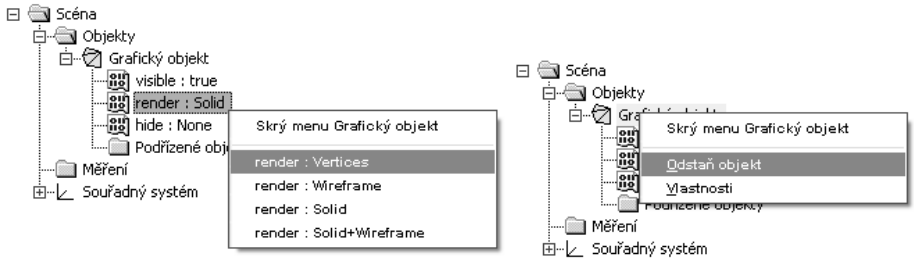


Obr. 32 Zobrazení stromu

Okno virtuální scény(GLCanvas) a také scéna(GOScene) mají jako jednu ze svých vlastností ukazatel na tuto komponentu. Při přidávání nového grafického objektu do scény je také tomuto objektu nastaven ukazatel na komponentu grafu scény. Potom každý grafický objekt scény může manipulovat ze stromem (přidávat, odebírat a měnit položky ve stromu).

Pro reprezentaci scény a jejích grafických objektů je navržen jednotný způsob práce ze stromem. Samotná scéna reprezentována instancí třídy GOScene vytváří kořen stromu a má tři potomky. První dva jsou adresáře s názvy Objekty a Měření. Do adresáře Objekty se přidávají grafické objekty scény mimo objektů, které slouží k měření, ty se umísťují do adresáře Měření. Posledním potomkem je adresář představující grafický objekt souřadného systému třídy GOAxis. Každý grafický objekt se do stromu přidává jako nový adresář. Obsah tohoto adresáře je závislý na druhu objektu. Každá třída odvozená ze základního grafického objektu může do adresáře vkládat různé položky. Základní grafický objekt(GraphicObject) do svého adresáře vkládá listy reprezentující hodnoty jeho základních vlastností(viditelnost, způsob zobrazení, způsob skrytí stěn) a jeden adresář, do kterého se umísťují jemu podřízené grafické objekty. Grafický objekt má identifikátory na položky ve svém adresáři, a proto při změně jeho vlastnosti může také změnit text příslušného listu.

Vizuální komponenta TreeView kromě zobrazení stromu také umožňuje odebírání a změnu jednotlivých vlastností objektů pomocí tzv. kontextového menu. Požadavek na jeho vykreslení se zajišťuje stisknutím pravého tlačítka myši nad některou z položek stromu. Obsah kontextové menu a také, jestli se vůbec má zobrazit nad konkrétní položkou je přenecháno scéně a grafickým objektům v ní umístěných. Pokud vznikne požadavek na kontextové menu, dohledá se příslušný objekt, kterému patří konkrétní položka stromu. Grafický objekt určí obsah kontextového menu a také událost, která se má provést po výběru některé z možností. Tímto způsobem je zajištěna snadná změna ovládání pouhým rozšířením třídy grafického objektu. Základní grafický objekt umí pomocí kontextového menu měnit své vlastnosti, odebrat se ze scény a také otevřít okno detailních vlastností.



Obr. 33 Navržená komponenta TreeView s kontextovým menu

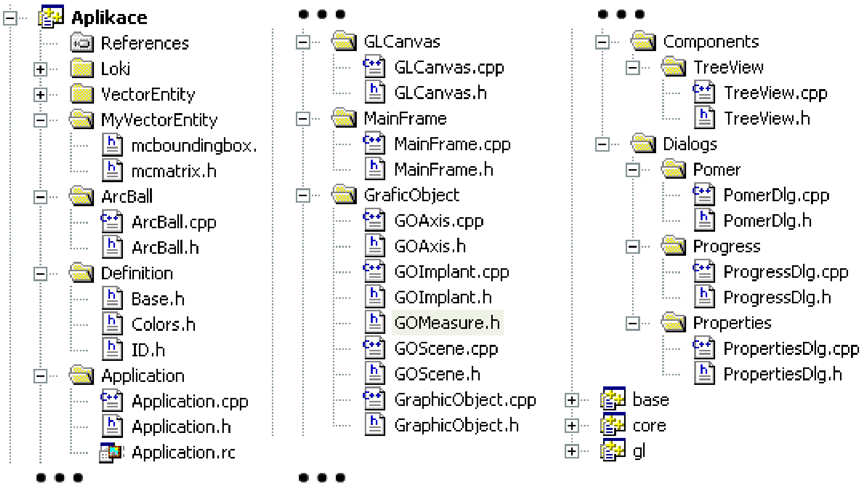
3.3 Okno menu s ovládáním

Okno je tvořeno dvěmi vizuálními komponentami z knihovny wxWidgets. Horní část je menu třídy wxMenu tato komponenta vytváří standardní systémové menu v horní části okna aplikace. Pod menu je umístěna komponenta wxToolBar, která umožňuje vytvářet panel ve, kterém jsou malé ovládací

tlačítka. Menu slouží k ovládání aplikace, správě projektu, vyberu režimu plátna, přidávání a odebírání objektů scény, a také k přidávání měřících nástrojů. Ovládací panel je navržený tak aby umožnil výběr možností, takzvané přepínání. Mezi tyto možnosti patří zvolení druhu projekce využité v okně virtuální scény, změna režimu manipulace, a také možnost uložení obrázku grafického plátna.

4 Implementace

Aplikace 3D grafického systému, je nedílnou součástí této diplomové práce a byla vytvořena na základě návrhu popsaného v předchozí kapitole. Aplikace je napsána v programovacím jazyce C++ a využívá knihoven wxWidgets, OpenGL a VectorEntity. Pro vývoj projektu jsem zvolil vývojové prostředí *Microsoft Visual Studio 7.1(2003)*. Toto vývojové prostředí usnadňuje práci programátora, protože obsahuje editor zdrojového kódu, kompilátor a také debugger. Snadná správa celého projektu je v tomto vývojovém prostředí zajištěna tzv. stromem řešení. Obrázek 34 znázorňuje strom řešení, kde jsou vidět všechny součásti implementovaného grafického systému.



Obr. 34 Strom řešení

Detailní informace o implementaci programu lze získat z programové dokumentace, která je uložena ve formě html stránek na přiloženém CD (adresář /doc/index.html). Tato dokumentace, byla psána během vývoje projektu přímo do zdrojového kódu a to jako poznámky se speciálními značkami systému pro dokumentaci Doxygen. Doxygen je systém pro dokumentaci zdrojových kódů v C++, C, Javě, PHP atd podrobnější informace lze získat na domovském www: http://www.doxygen.org [1. 5. 2007].

4.1 Knihovna wxWidgets

WxWidgets je multiplatformní knihovna napsaná v jazyce C++. Umožňuje psaní aplikací pro platformy Linux, Windows (16bit, 32bit), Mac OS, embed zařízení,... Zahrnuje nástroje pro psaní nejen GUI aplikací, ale také obsahuje třídy pro síťovou komunikaci, thready, streamy a mnoho dalších.

Existuje několik důvodů proč je výhodné tuto knihovnu zvolit pro vývoj aplikací. Vývojáři se snažili o co nejširší a spolehlivou portovatelnost. Pod daným operačním systémem se knihovna snaží využívat co nejvíce nativních prostředků dané platformy. To činí knihovnu přijatelně rychlou a spolehlivou. Další výhodou wxWidgets je velmi podrobná programátorská dokumentace. Pro řešení problému je na webu přístupný i MAIL-LIST, jak pro vývojáře knihovny, tak pro její uživatele. Další nespornou výhodou je, že knihovna byla uvolněna pod licencí GPL, proto mají programátoři přístup i ke zdrojovým kódům knihovny. Otevřenost knihovny umožňuje také její rychlý vývoj za přispění mnoha dobrovolníků na celém světě. Detailnější informace o knihovně jsou dostupné na domovském webu wxWidgets http://www.wxwidgets.org/ [1. 5. 2007].

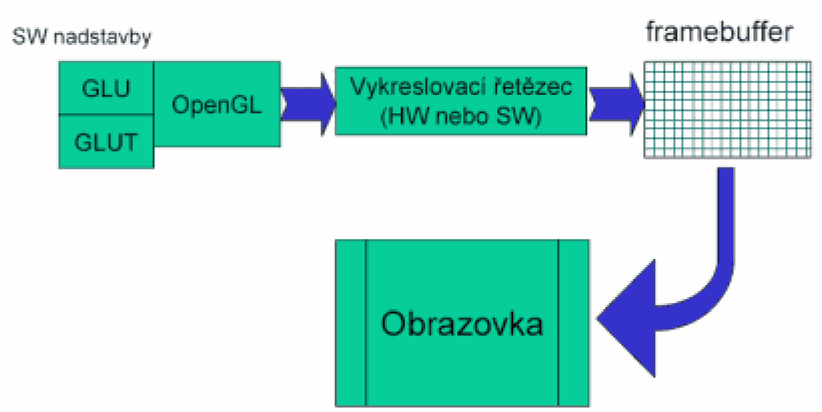
4.2 Knihovna OpenGL

Knihovna OpenGL™ (Open Graphics Library) byla navržena firmou SGI (Silicon Graphics Inc.) jako aplikační programové rozhraní (Application Programming Interface - API) k akcelerovaným grafickým kartám resp. celým grafickým subsystémům. Předchůdcem této knihovny byla programová knihovna IRIS GL (Silicon Graphics IRIS Graphics Library). OpenGL byla navržena s důrazem na to, aby byla použitelná na různých typech grafických akcelerátorů a aby ji bylo možno použít i v případě, že na určité platformě žádný grafický akcelerátor není nainstalován - v tom případě se použije softwarová simulace. V současné době lze knihovnu OpenGL použít na různých verzích unixových systémů (včetně Linuxu a samozřejmě IRIXu), OS/2 a na platformách Microsoft Windows.

Logo OpenGL a název OpenGL™ je registrovaná známka firmy Silicon Graphics Inc. Programátorské rozhraní knihovny OpenGL je vytvořeno tak, aby knihovna byla použitelná v téměř libovolném programovacím jazyce. Primárně je k dispozici hlavičkový soubor pro jazyky C a C++. V tomto souboru jsou deklarovány nové datové typy používané knihovnou, některé symbolické konstanty (např. GL_POINTS) a sada cca 120 funkcí tvořících vlastní rozhraní.

Z programátorského hlediska se OpenGL chová jako stavový automat. To znamená, že během zadávání příkazů pro vykreslování lze průběžně měnit vlastnosti vykreslovaných primitiv (barva, průhlednost) nebo celé scény (volba způsobu vykreslování, transformace) a toto nastavení zůstane zachováno do té doby, než ho explicitně změníme. Výhoda tohoto přístupu spočívá především v tom, že funkce pro vykreslování mají menší počet parametrů a že jedním příkazem lze globálně změnit způsob vykreslení celé scény. Vykreslování scény se provádí procedurálně - voláním funkcí OpenGL se vykreslí výsledný rastrový obrázek.

Výsledkem volání těchto funkcí je rastrový obrázek uložený v tzv. framebufferu, kde je každému pixelu přiřazena barva, hloubka, alfa složka popř. i další atributy. Z framebufferu lze získat pouze barevnou informaci a tu je možné následně zobrazit na obrazovce viz obr. 35.



Obr. 35 Vykreslování scény v OpenGL

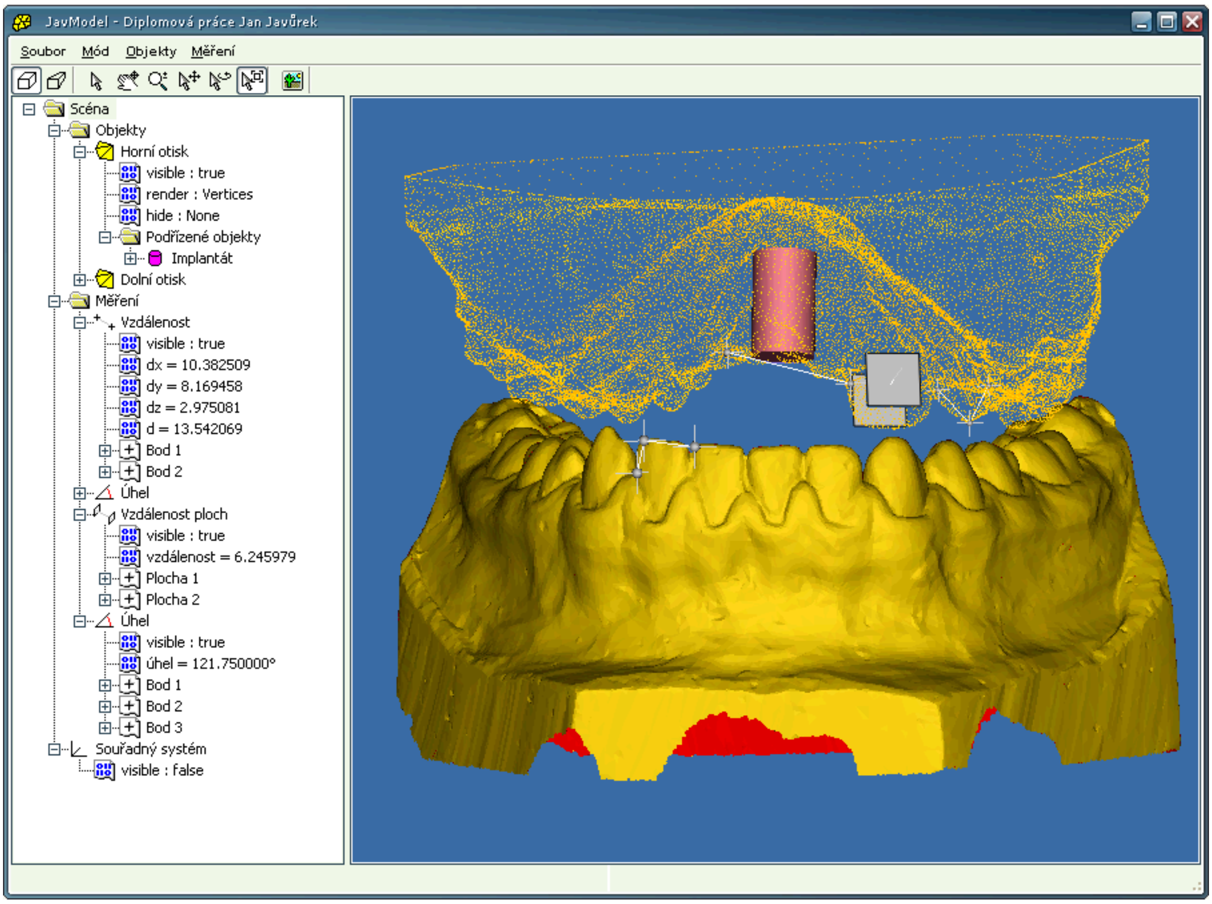
Pomocí funkcí poskytovaných knihovnou OpenGL lze vykreslovat obrazce a tělesa složená ze základních geometrických prvků, které nazýváme grafická primitiva. Mezi tato primitiva patří bod, úsečka, trojúhelník, čtyřúhelník, plošný konvexní polygon, bitmapa (jednobarevný rastrový obraz) a pixmapa (barevný rastrový obraz). Existují i funkce, které podporují proudové vykreslování některých primitiv. Na vrcholy tvořící jednotlivá grafická primitiva lze aplikovat různé transformace (otočení, změna měřítka, posun, perspektivní projekce), pomocí kterých lze poměrně jednoduše vytvořit animace. Vykreslovaná primitiva mohou být osvětlena nebo pokryta texturou.

4.3 Knihovna VectorEntity

Knihovna Vector entity library je vyvíjena Ing. Přemyslem Krškem, Ph.D. na Vysokém učení technickém v Brně, fakultě informačních technologií, ústavu počítačové grafiky a multimédií. Knihovna je napsána v C++ a umožňuje efektivní práci s trojúhelníkovými a tetrahedrálními 3D modely.

5 Výsledky

Výsledkem je vytvořená aplikace grafického systému znázorněná na obrázku 36, na kterém je vidět horní menu s ovládací lištou, graf scény a okno scény.



Obr. 36 Vytvořený 3D grafický systém

5.1 Ovládací lišta

Ovládací lišta urychluje práci s aplikací díky přehlednému výběru nejpoužívanějších vlastností.

- Zvolení druhu projekce (obr. 38)

  Ortogonální (paralelní projkekce) Perspektivní (středová projekce)

- Výběr manipulačního nástroje, který se používá ve scéně

  Výběr objektů a rotace scénou Posouvání scénou

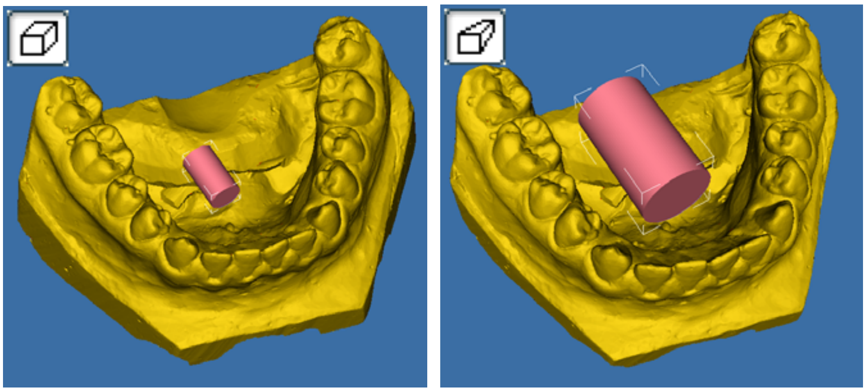
  Přiblížení a oddálení scény Výběr a změna polohy objektu

  výběr a změna natočení objektu výběr a změna velikosti objektu

- Uložení aktuálního obsahu okna scény do souboru typu BMP

Při výběru lze vybrat vždy jen jednu z možností z dané kategorie, pro přehlednost jsou jednotlivé kategorie odděleny separátorem (čárou) jak ukazuje obr. 37.

Obr. 37 Ovládací lišta



Obr. 38 Ukázka změny projekce (ortogonální a perspektivní)

5.2 Menu

Menu je tvořeno čtyři záložkami. Záložka Soubor je pro práci s projekty a aplikací. Do projektového souboru se ukládá nejen obsah scény, ale také vzhled(rozbalenost adresářů) okna grafu. V záložce označené mód lze nastavit režim scény podobně jako v ovládací liště. Pro práci s objekty je určena záložka Objekty a pro přidávání měřících nástrojů je vytvořena záložka Měření.

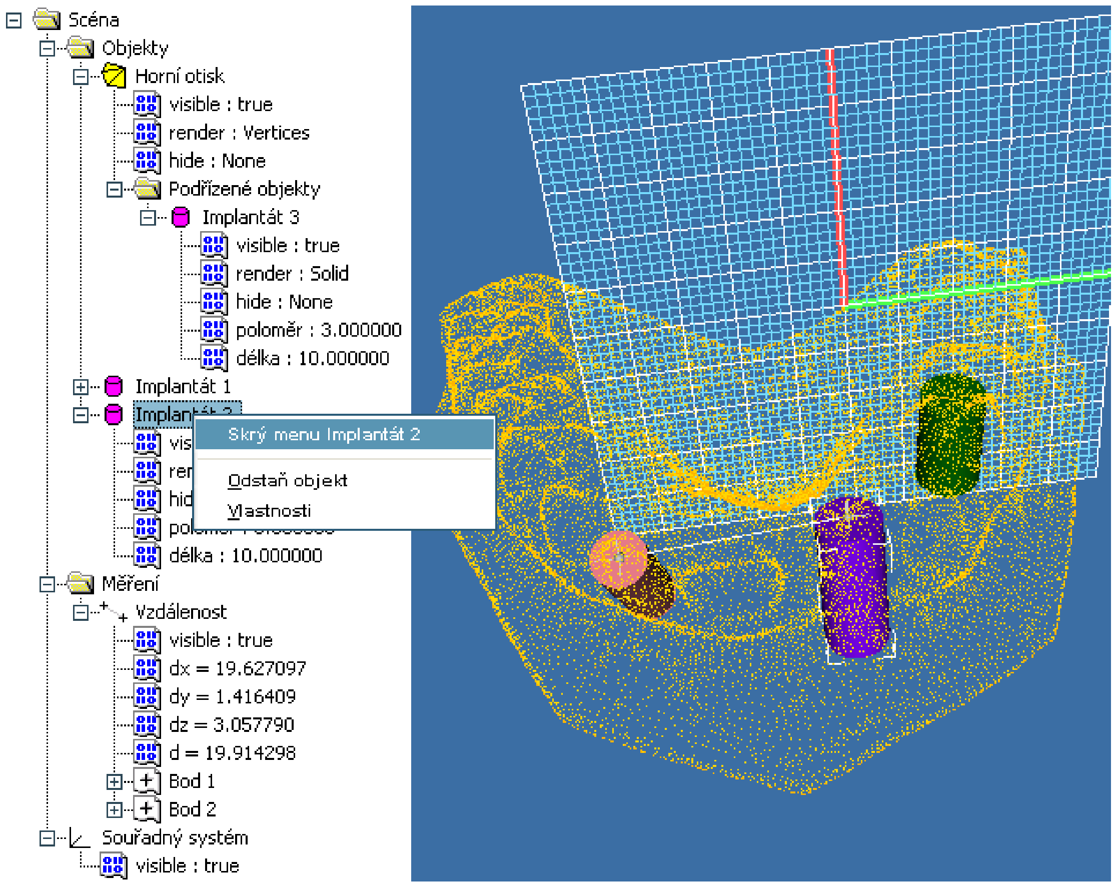
5.3 Graf scény

V okně grafu scény(obr. 39 vlevo) je možné odstraňovat objekty nebo měnit jejich vlastnosti pomocí kontextového menu. Např. pro změnu parametru typu ano/ne se zobrazí kontextové menu s opačnou hodnotou než je aktuálně nastavena(obr. 39 vlevo nahoře), pro více stavové typy se zobrazí všechny varianty. Ke změně hodnoty dojde po výběru jedné z možností.

Obr. 39 Ukázka kontextových menu

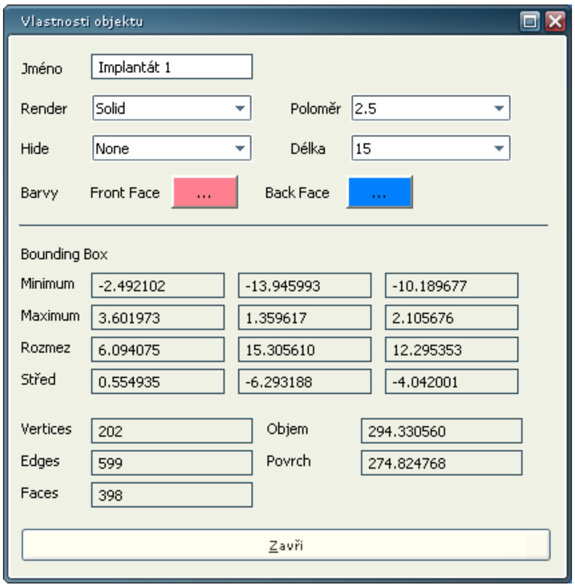
Na obrázku 40 je uveden příklad stromu, který obsahuje jeden medicínský model (Horní otisk), tři implantáty a jeden měřící nástroj. Implantát 3 je podobjektem horní čelisti a chová se jako jeho součást(společně se pohybují, otáčejí atd.).

Strom grafu scény dále umožňuje změnu úrovně objektu ve stromu pomocí přetažení. Pokud bychom chtěli Implantát 3 také zařadit mezi podřízené objekty horního otisku, uchopíme implantát a přetáhneme ho do adresáře podřízených objektů horního otisku. Při této změně se automaticky mění transformační matice přesouvaného objektu, tak aby se přesun neprojevil na jeho pozici a natočení v trojrozměrném prostoru.



Obr. 40 Ukázka stromu scény

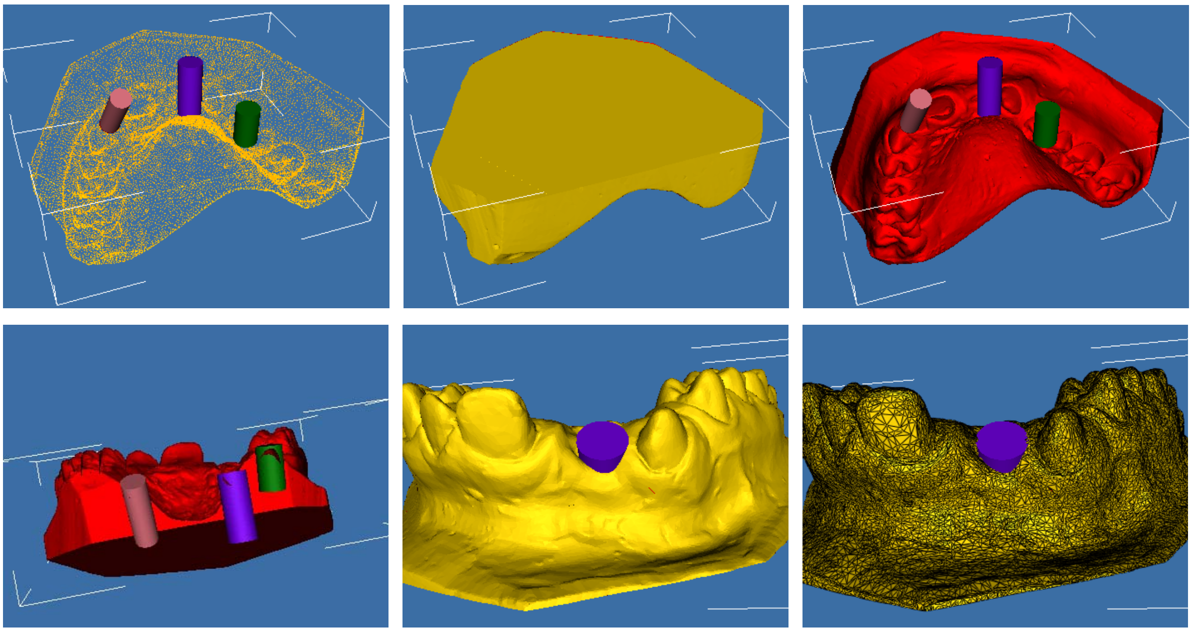
U některých objektů ve stromu nejsou zobrazovány veškeré jejich vlastnosti. K detailním vlastnostem objektu slouží dialogové okno, které se vyvolá pomocí řádku kontextového menu Vlastnosti. Např. u vlastností implantátu lze měnit jeho poloměr, délku, barvu atd. Okno vlastností(obr. 41) navíc zobrazuje informace vypočítané z trojúhelníkové sítě (objem, povrch, bounding box a počty elementů).



Obr. 41 Okno změny vlastností grafického objektu

5.4 Okno scény

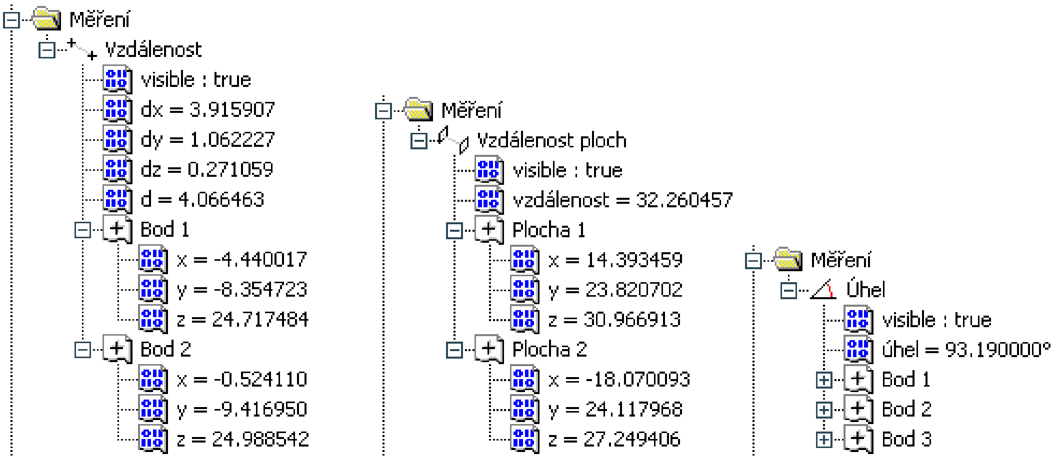
Okno scény je hlavní částí aplikace, která zobrazuje a umožňuje manipulovat s objekty grafu scény. Na obrázku 42 jsou uvedeny příklady různého zobrazení např. skrytí předních stěn čelisti umožňuje získat snadný přehled umístění implantátů.



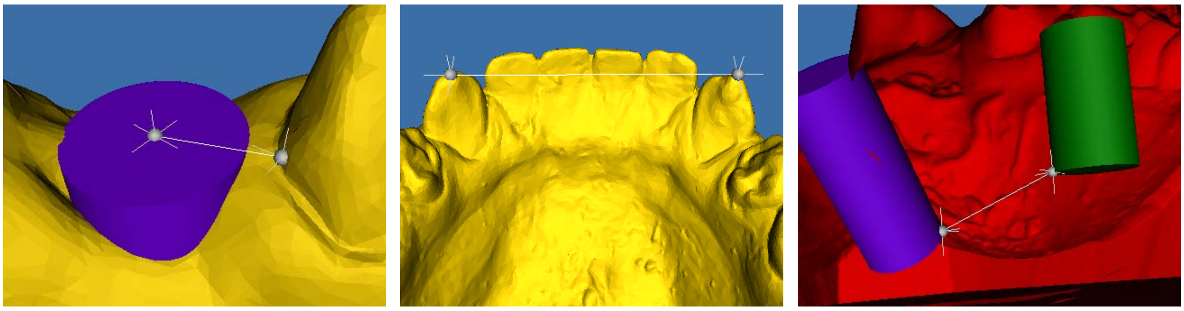
Obr. 42 Ukázka zobrazení modelů z různých pohledů a typů vykreslení

5.5 Měření

Měření se provádí manipulací s vloženými měřícími nástroji ve scéně. Změřená veličina se zobrazuje v okně grafu scény (obr. 43). Vzdálenostní jednota(mm, dm, m atd.) naměřených veličin je závislá na jednotce importovaného modelu.

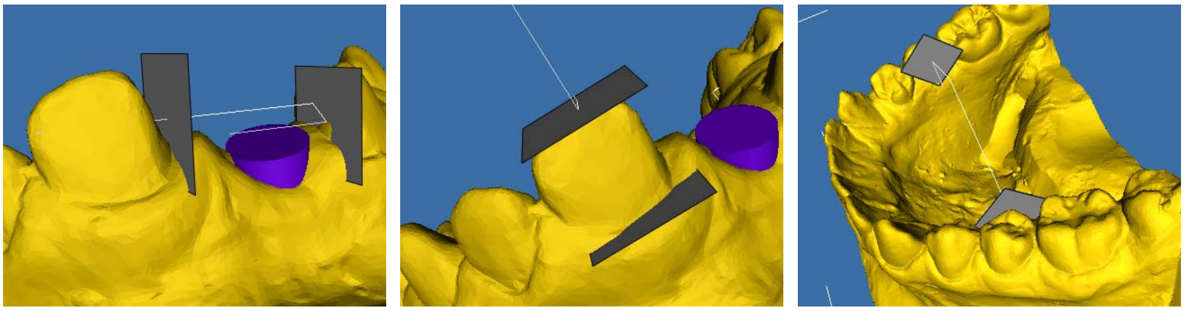


Obr. 43 Zobrazení měřících nástrojů s výsledky



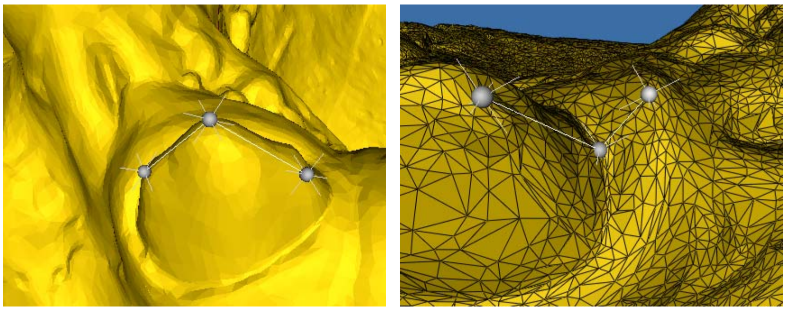
Obr. 44 Ukázky měření vzdálenosti pomocí dvou bodů

(vlevo: změření vzdálenosti středu implantátu od vedlejšího zubu, uprostřed: změření vzdálenosti vrcholků zubů, vpravo: změření vzdálenosti mezi dvěma implantáty)



Obr. 45 Ukázky měření vzdálenosti mezi dvěmi rovnoběžnými plochami

(vlevo: změření vzdálenosti zubů, uprostřed: změření výsky zubu, vpravo: změření šířky dutiny ústní)



Obr. 46 Ukázka měření úhlů

(vlevo: měření skosení zubu, vpravo: měření úhlu mezi zubem a dásní)

6 Závěr

V diplomové práci byly analyzovány možnosti 3D počítačového modelování a jeho využití pro modelování v medicíně. Na základě analýzy reprezentací 3D počítačových modelů byl navržen a implementován 3D grafický systém zaměřený na konkrétní využití ve stomatologii.

Vytvořený systém poskytuje zubnímu lékaři neocenitelný nástroj, který mu pomůže před vlastním lékařským zákrokem naplánovat nejvhodnější operační postup a popřípadě vybrat nejvhodnější typ a velikost použitého implantátu. Toho je docíleno snadným pohybem a práci v trojrozměrné scéně. Trojrozměrná scéna zobrazuje modely jako např. naimportované modely částí lidského těla (modely čelistí), implantáty a měřící nástroje. Trojrozměrné modely pacientovy čelisti nebo jiných částí těla lékaři dnes získávají z dat počítačové tomografie nebo magnetické rezonance. Systém je umožňuje neimportovat z binárního souborů typu STL. Lékaři je v systému umožněno měnit parametry implantátu a tím vybrat ten nejvhodnější pro konkrétního pacienta, pokud by nebylo možné použít standardní velikost implantátu, může lékař provádět trojrozměrná měření a na jejich základě objednat specielní např. zkosený implantát. Aktuální stav práce v systému se všemi nastaveními a objekty ve scéně si lékař může snadno uložit a v budoucnu znovu otevřít.

Tato diplomová práce může sloužit jako jednoduchý odrazový můstek v použití 3D počítačové grafiky pro modelování v medicíně, protože popisuje základní reprezentace trojrozměrných modelů, možnosti jejich transformací a také systém, který dokáže pracovat s trojrozměrnou počítačovou scénou. Pro další rozvoj implementovaného grafického systému je nutné:

- rozšířit možnost pohledu na trojrozměrnou scénu více okny (simulace lidského vidění, pohledy z různých stran)
- vylepšení měření o možnost propojení měřícího nástroje s konkrétním grafickým objektem
- vytvořit nové třídy grafických objektů, které by reprezentovaly můstky, zubní korunky, protézy a zařízení pro kotvení protéz
- rozšířit import modelů z dalších standardních souborových formátů (např. WaveFront OBJ, LightWave LWO)

Dle mého názoru má 3D počítačové modelování velký potenciál pro využití v medicíně. Modelování lze použít nejen při návrhu tvaru a umístění implantátu, ale také může velkou měrou přispět ke zvýšení úspěšnosti lékařských zákroků.

Literatura

[Baum] BAUMGART B.G., *Winged-Edge Polyhedron Representation for Computer Vision*, Článek z mezinárodní počítačové konference 1975, Dostupné z www: http://www.baumgart.org/winged-edge/winged-edge.html [1.5.2007]

[Clunie] CLUNIE D. A., *DICOM Structured Reporting*, PixelMed Publishing, 2001. 394s. ISBN 0-9701369-0-0

[Heesch] HEESCH. *Dokumentace k dokumentačnímu systému Doxygen* Dostupné z www: http://www.stack.nl/~dimitri/doxygen/manual.html [1. 5. 2007]

[Herout] HEROUT, *Učebnice jazyka C*, 2. vyd. České Budějovice: nakladatelství KOPP, 1993, 269s, ISBN 80-85828-02-2

[Kořínek] KOŘÍNEK, *Implantologie*, Článek o implanetologii, Dostupní z www: http://www.mudrkorinek.cz/cz/implantologie.html [1.5.2007]

[Kršek05] KRŠEK, *Problematika 3D modelování tkání z medicínských obrazových dat*, Odborný článek, Dostupné z www: http://www.fit.vutbr.cz/~krsek/publications/neurologi_pro_praxi_0305_krsek_draft.pdf [1.5.2007]

[Kršek06] KRŠEK, *Reprezentace 3D objektů část 1 a 2*. Podklady k přednášce z předmětu IZG na FIT VUT v Brně, Dostupné z privátní www: https://www.fit.vutbr.cz/study/courses/IZG/private/lecture03/izg_3d_objekty_1.pdf https://www.fit.vutbr.cz/study/courses/IZG/private/lecture03/izg_3d_objekty_2.pdf [1. 5. 2007]

[Mart] MART, ROEBLING, ZEITLIN, a kol. *Programátorská dokumentace ke knihovně wxWidgets 2.4.2* Dostupné z www: http://www.wxwidgets.org/manuals/2.4.2/wx.htm [1.5.2007]

[Schildt] SCHILDT, *Nauč se sám C++*. 1. vyd. Praha: SoftPress s.r.o. 2001. 619s. ISBN 80-86497-13-5

[Stephen] STEPHEN *Mistrovství v C++*, 2. vyd. Brno: Computer Press, 2004. 1006s. ISBN 80-251-0098-7

[Španěl] ŠPAŇEL, BERAN, *Obrazové segmentační techniky, Přehled existujících metod* Dostupné z www: http://www.fit.vutbr.cz/~spanel/segmentace/ [1.5.2007]

[Žára] ŽÁRA, BENEŠ, FELKL, *Moderní počítačová grafika*, 1 vyd. Praha: Computer Press, 1998. 448s. ISBN 80-7226-049-9

Seznam příloh

Příloha 1. Přiložené CD

Obsah média:

- Spustitelná aplikace 3D grafický systém (v adresáři app)
- Zdrojové kódy 3D grafického systému (v adresáři scr)
- Programová dokumentace (v adresáři doc)
- Medicínské modely s testovacími projekty (v adresáři models)
- Využité nástroje (v adresáři tools)
- Využité knihovny (v adresáři scr\librarys)

Seznam obrázků

Obr. 1 Popis tělesa je v hraniční reprezentaci převeden na popis pláště 5
Obr. 2 2-manifold a nonmanifold objekt 5
Obr. 3 Ukázka výpočtu Eulerovi rovnosti na modelu s děrami 6
Obr. 4 Triangle strip a Triangle fan 7
Obr. 5 Nejednoznačnost drátového modelu 8
Obr. 6 Ukázka polygonálního modelu 9
Obr. 7 Datová struktura hrany 9
Obr. 8 Příklad okřídlené hrany 10
Obr. 9 Spline model lidského obličeje 11
Obr. 10 Translační a rotační šablonování 12
Obr. 11 Šablonování potahováním 12
Obr. 12 Implicitní plocha vzniklá ze tří úseček a jednoho bodu 13
Obr. 13 Příklad dekompozičního modelu 14
Obr. 14 Geometrický model lidské lebky 15
Obr. 15 Základní případy konfigurace vrcholů krychle u algoritmu Marching cubes 16
Obr. 16 Rotace kolem vektoru $\vec{v}$ o úhel α 19
Obr. 17 Ukázka grafu scény 22
Obr. 18 Základní rozložení prvků v okně 3D systému 23
Obr. 19 Diagram dědičnosti grafického objektu 25
Obr. 20 Část diagramu tříd grafického objektu s obousměrně vázaným seznamem 25
Obr. 21 Datové struktury pro uložení trojúhelníkového modelu v grafickém objektu 25
Obr. 22 Kompletní diagram tříd grafického objektu 26

Obr. 23 Navržené způsoby vykreslení trojúhelníkové sítě ... 26
Obr. 24 Skrytí předních nebo zadních stěn ... 27
Obr. 25 Diagram dědičnosti pro objekt scény ... 28
Obr. 26 Diagram rozšíření tříd scény ... 29
Obr. 27 Pohyb po povrchu koule ... 31
Obr. 28 Rozlišení režimů manipulace kurzorem ... 31
Obr. 29 Měřící nástroje ... 32
Obr. 30 Diagramy tříd měřících nástrojů ... 32
Obr. 31 Zubní implantát(jednofázový, dvoufázový) a jeho zjednodušená reprezentace ... 33
Obr. 32 Zobrazení stromu ... 33
Obr. 33 Navržená komponenta TreeView s kontextovým menu ... 34
Obr. 34 Strom řešení ... 36
Obr. 35 Vykreslování scény v OpenGL ... 38
Obr. 36 Vytvořený 3D grafický systém ... 39
Obr. 37 Ovládací lišta ... 40
Obr. 38 Ukázka změny projekce (ortogonální a perspektivní) ... 40
Obr. 39 Ukázka kontextových menu ... 40
Obr. 40 Ukázka stromu scény ... 41
Obr. 41 Okno změny vlastností grafického objektu ... 42
Obr. 42 Ukázka zobrazení modelů z různých pohledů a typů vykreslení ... 42
Obr. 43 Zobrazení měřících nástrojů s výsledky ... 43
Obr. 44 Ukázky měření vzdálenosti pomocí dvou bodů ... 43
Obr. 45 Ukázky měření vzdálenosti mezi dvěmi rovnoběžnými plochami ... 43
Obr. 46 Ukázka měření úhlů ... 44