



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

MIMIKA LIDSKÉ TVÁŘE

HUMAN FACIAL GESTURES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VEDOUCÍ PRÁCE
SUPERVISOR

ALEXANDER MELICHER

Ing. RADOVAN JOŠTH

BRNO 2009

Abstrakt

Bakalářská práce se zaměřuje na simulaci mimiky lidské tváře, která se v současnosti využívá ve velkém rozsahu při vývoji počítačových her a animovaných filmů. Úkolem práce bylo vytvořit 3D model v grafickém open-source programu Blender. Následně na to vyvinout aplikaci, která načte daný objekt a pomocí renderovacího algoritmu ho vykreslí na obrazovku. Aplikace umožní uživateli jednoduchým ovládním pomocí klávesnice vytvářet mimické výrazy. Na to byla aplikovaná technika posouvání vertexů v určitých vymezených oblastech tváře. Renderovací algoritmus byl napsán v jazyce C++ s využitím multiplatformové knihovny OpenSceneGraph, která představuje určitou nadstavbu OpenGL.

Abstract

Bachelor's thesis is concerned with the demonstration of human facial gestures, which has been frequently used in development of computer games and animated movies. The main brief of work was to create 3D model in open-source graphic software Blender. Subsequently, an application for loading of the created model and displaying the model on the screen was developed using a rendering algorithm. The application allow user to create common facial gestures by simple keyboard control. A technigue for moving the vertices was applied to some delimited areas on the face. Rendering algorithm was written in C++ language using a multiplatform library OpenSceneGraph, the extension of OpenGL.

Klíčová slova

mimika tváře, 3D grafika, Blender, C++, OpenSceneGraph.

Keywords

facial gestures, 3D graphics, Blender, C++, OpenSceneGraph.

Citace

Alexander Melicher: Mimika lidské tváře, bakalářská práce, Brno, FIT VUT v Brně, 2009

Mimika lidské tváře

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radovana Joštha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal

.....
Alexander Melicher
19. května 2009

Poděkování

Tímto bych chtěl poděkovat Ing. Radovanovi Jošthovi za poskytnutou pomoc a konzultace při tvorbě této práce.

© Alexander Melicher, 2009.

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

1	Teória	4
1.1	Anatómia tváre, hlavy a krku	4
1.1.1	Lebka	4
1.1.2	Svalstvo	4
1.1.3	Koža	6
1.1.4	Oči	6
1.2	Základné techniky pri rozpoznávaní mimiky	7
1.2.1	Kategorizácia mimických výrazov	8
1.2.2	Mimický jazyk	9
1.2.3	Facial Action Coding System	10
1.3	Tvorba mimiky v počítačovej grafike	11
1.3.1	Key-Frame Interpolácia	11
1.3.2	Performance-Based animácia	13
1.3.3	Priame parametrizované modely	13
1.3.4	Animácia založená na pseudosvaloch	14
1.3.5	Netypizované deformácie(Freeform Deformations)	14
1.3.6	Reálna animácia tváre	16
1.4	Zvolené metódy pre tento projekt	16
1.4.1	Lokalizácia vertexov v deformačných zónach	16
1.4.2	Mimika tvára založená na svalových vektoroch	17
1.5	Vertex Shader	20
2	Návrh riešenia	21
2.1	Modelovacia časť	21
2.1.1	Blender	21
2.1.2	Deformačné oblasti	21
2.1.3	Formát .obj	21
2.2	Programovacia časť	22
2.2.1	OpenSceneGraph	22
2.2.2	Tvorba mimických výrazov	22
3	Implementácia	23
3.1	Modelovanie	23
3.1.1	Sculpt mode	23
3.1.2	Retopológia	23
3.1.3	Normal mapping	23
3.1.4	Vytvorenie deformačných oblastí	24
3.2	Tvorba algoritmu	25

3.2.1	Main.cpp	25
3.2.2	Functions.cpp	26
4	Možný budúci vývoj	28
4.1	Pokračovanie vo vývoji	28
4.2	Vývoj v oblasti výukových aplikácií	29
4.2.1	Príprava na pracovný pohovor	29
4.2.2	Kriminalistika	29
4.3	Virtuálna moderátorka	29
5	Záver	30
A	Obsah CD	32
B	Manuál	33
B.1	Spustenie programu	33
B.2	Ovládanie programu	33
C	Plagát	35

Úvod

V posledných rokoch sme si mohli všimnúť veľký rozmach animovaných filmov, v ktorých sa hlavný dôraz kladie okrem iného aj na realistické stvárnenie charakterov postáv. Taktiež v hernom priemysle badať čoraz väčšiu snahu vývojárov o vytvorenie čo naj dôveryhodnejšie vyzerajúcej postavy hrdinu alebo vojaka. S tým súvisí okrem dôkladného modelu osoby aj reálne znázornenie mimických výrazov a gest. Vzhľadom na to sa algoritmy riešiacie tento problém neustále zdokonaľujú alebo sa vymýšľajú nové a efektívnejšie alternatívy. Keďže náročnosť výpočtov daných algoritmov neustále rastie, zároveň sa musia vyvíjať aj grafické karty schopné vykonať dané výpočty v rozumnom čase. Keďže kapacita grafických kariet je obmedzená, existuje snaha o maximálne využitie dostupných zdrojov na grafickej karte a zároveň dosiahnutie čo najlepšieho výsledku. To je dosiahnuteľné jedine implementovaním čo najefektívnejších algoritmov.

Úlohou práce je vytvoriť 3D model tváre a aplikáciu, ktorá daný model načíta a vykreslí na obrazovku. Daná aplikácia umožní jednoduchým ovládaním pohybovať určitými oblasťami modelu a tým vytvárať mimické výrazy.

V prvej, teoretickej kapitole, sa rozoberie ľudská hlava z anatomického hľadiska. Budú popísané jednotlivé časti od orgánov až po pokožku, ich vzájomné prepojenie a hlavne príčina „deformácie tváre“ v podobe mimiky. Následný opis tváre bude z pohľadu počítačovej grafiky, t.j. ako bola prezentovaná v minulosti, aké algoritmy sa využívali pri znázorňovaní mimických výrazov, čo sa zmenilo a aké techniky sa využívajú v dnešnej dobe. Nakoniec bude detailne opísaná vybraná metóda použitá v projekte, jej výhody a dôvod výberu.

V ďalšej kapitole sa predstaví návrh riešenia. Bude obsahovať zoznam aplikácií, ich stručný popis, možné alternatívy a dôvod, prečo sa použili práve spomenuté aplikácie a postupy. Načrtne sa tu spôsob vládnania aplikácie, hlavne čo sa týka tvorby mimických výrazov.

Tretia kapitola bude obsahovať celkový postup práce od modelovania tváre cez programovanie renderovacieho algoritmu až po dokončenie aplikácie. Budú v nej opísané a vysvetlené použité doporučené postupy pri modelovaní ako aj implementovaný renderovací algoritmus.

V predposlednej kapitole budú spomenuté možnosti budúceho vývoja a možné alternatívy k danému projektu.

V záverečnej kapitole budú zhodnotené výsledky práce. Ďalej budú objasnené dôvody, prečo sa to implementovalo takým spôsobom a v takom rozsahu. Na koniec bude umiestnený komentár riešiteľa o osobnom prínose pri vytváraní danej práce.

Kapitola 1

Teória

1.1 Anatómia tváre, hlavy a krku

Jedným z cieľov pri vytváraní 3D počítačovo-generovaného modelu tváre je vytvoriť modely, ktoré vyzerajú realisticky nielen v statickom zobrazení, ale taktiež sa dokážu pohybovať v animovaných sekvenciách. Kvôli tomuto je veľmi dôležité pochopiť stavbu a funkciu hlavy z anatomického hľadiska. Najviac používaná lekárska príručka je 40. vydanie knihy britského anatóma a chirurga Henryho Graya *Grayova anatómia*, ktorá poskytuje precízny a detailný opis anatómie tela. Ďalšou významnou príručkou je graficky ilustrovaná kniha *Atlas Anatómie*. Pre danú prácu je ale najpodstatnejšia kniha *Cunninghamova príručka praktickej anatómie, 3. vydanie: Hlava, krk a mozog*.

1.1.1 Lebka

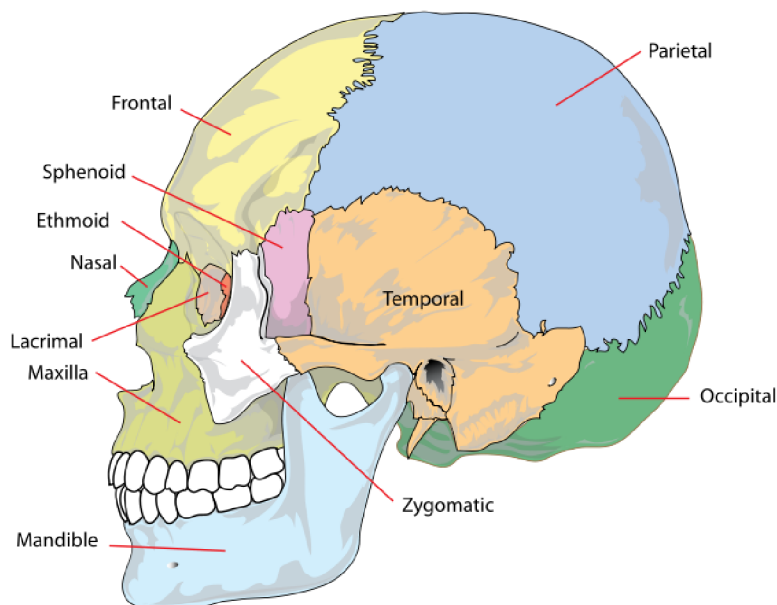
Lebka je vo svojej podstate ochranným púzdom pre mozog a poskytuje základ obrysu tváre. Lebečné kosti môžu byť rozdelené do dvoch hlavných častí, a to na:

- lebečnú časť, ktorá uschováva a ochraňuje mozog
- kostru tváre, kde je sánka jediná pohyblivá spojená kostná štruktúra

Lebečnú časť tvorí čelná kosť(frontal), záhlavová kosť(occipital) a kosť lebečná(parietal), ďalej spánková(temporal) a klinová kosť(sphenoid). Kostru tváre tvoria klinová kosť(sphenoid), kosť čuchová, podnebná kosť(palatine), horná čelusť(maxilla), dolná nosná mušľa (inferior nasal concha), líčne kosti(zygomatic), nosné kosti(nasal), slzná kosť(lacrimal), sánka, jazyčka(hyoid) a čerieslová kosť(vomer). Hlavné kosti a ich umiestnenie znázorňuje obrázok [1.1](#).

1.1.2 Svalstvo

Všeobecne sú svaly orgánmi pohybu. Ich kontrakcie spôsobujú pohyb rôznych častí tela. Energia ich kontraktíí je vytvorená efektívnym mechanizmom šliach, aponeuróz(šľachové blany pripájajúce svaly) a povázok(blany obopínajúce svaly), ktoré upevňujú konce svalov a kontrolujú riadenie ich ťahania. Svaly sú väčšinou umiestnené medzi dve pohybujúce sa časti, ako napr. dve kosti, kosť a kožu, dve odlišné kožné oblasti alebo dva orgány. Svaly sa môžu taktiež nachádzať medzi dvoma časťami, kde jedna z nich je upevnená a druhá pohyblivá. V takomto prípade sa uchytenie svaly na pevnú časť nazýva *pôsobisko* a uchytenie na pohyblivú časť *úpon*.



Obrázek 1.1: Bočný pohľad znázorňujúci jednotlivé lebečné kosti

Aktivita svalov spočíva v ich sťahovaní. Uvoľnenie svalstva je pasívne a dochádza k nemu pri zníženej alebo nulovej stimulácii. Sval je zásobovaný jedným alebo viacerými nervami, ktoré so sebou nesú stimulujúci impulz a ten vyvolá svalovú kontrakciu. Svaly môžu byť taktiež stimulované priamo signálmi z mozgu alebo aj elektrickým stimulom vychádzajúcim z nejakého zdroja. Týmto spôsobom francúzsky neurológ Duchenne de Boulogne v roku 1862 dokumentoval svoj výskum s fotografiami mimických výrazov.

Mimické svalstvo

Mimické svalstvo je bežný názov pre svaly nachádzajúce sa v oblasti tváre. Niektoré mimické svaly však vykonávajú ďalšie dôležité funkcie, ako napr. pohyb líc a pier počas žuvania a reči, alebo zatváranie a otváranie očných viečok. Mimické svaly sú povrchové a všetky sa pripájajú na vrstvu podkožného tuku a kože v ich úpone. Keď sú svaly uvoľnené, tukové tkanivo vyplňuje medzery a vyhladzuje hranaté prechody, aby bolo možné vidieť pôvodný tvar lebky. Mimické svaly pracujú synergisticky a nie nezávislo. Tvoria určité svalové skupiny, kde každý sval má špecifické funkcie, z ktorých jedna je primárna. Je zložitý odlišiť hranice medzi rôznymi svalmi, pretože ich konce sú medzi sebou poprepletané.

Svaly sú zásadnými stimulátormi mimických výrazov, kde počas svalovej kontrakcie sa konce svalov snažia pritiahnúť k sebe. Tento pohyb obyčajne zahŕňa aj pritiahnutie kože smerom k spojivovým bodom medzi kožou a svalmi. Svalové vlákna sa vyskytujú v rôznych dĺžkach. Niekedy obtiahnu celú dĺžku svalu a majú priemer $1 - 10\mu m$. Svalové vlákna sa skladajú z ešte menších častíc nazývaných myofibrily, ktoré obklopujú celú dĺžku vlákna. Každý myofibril je zhruba $0,1 - 0,2\mu m$ hrubý. Výsledkom je, že každé jednotlivé svalové tkanivo obsahuje stovky až tisícky myofibril.

Naprieč pozdĺžnej osi každej myofibrily je opakujúca sa vzorka vlákna nazývaná sar-

koméra. Jedná sa o štrukturálnu jednotku svalových vlákien a taktiež o funkčnú jednotku svalovej kontrakcie. Sarkoméry sú dlhé iba $0,1 - 0,2 \mu\text{m}$ a naťahujú sa pozdĺž svojej osi. Je to práve sériové skracovanie týchto sarkomér, ktoré spôsobí celkové skracovanie jednotlivých svalových vlákien. Následne na to celkové skrátenie svalu je spôsobené paralelnou kontrakciou niekoľkých vlákien.

Najzákladnejšou vrstvou sú dva sťahujúce proteíny *aktin* a *myozín*, ktoré formujú vlákna vo vnútri sarkoméry. Pričným rezom je možné pozorovať združenie v tvare šesťuholníka so šiestimi tenkými myozínovými vláknami, ktoré obklopujú každé hrubé aktinové vlákno. Svaly sa správajú ako perfektne zladené pružiny, ktorých činnosť určuje dĺžku svalov a úroveň nervovej aktivizácie.

Všeobecne môžu byť mimické svaly zoskupené podľa orientácie svalových vlákien a tým pádom rozdelené na dolnú a hornú časť tváre. V počítačovej grafike sú svaly na tvári rozdelené typicky do troch skupín z hľadiska ich spôsobu kontrakcie:

- lineárne/paralelné svaly, ktoré ťahajú kožu v smere pod určitým uhlom
- elipsovité/kruhovité zvieracie svaly, ktoré uzatvárajú očné a ústnu štrbinu
- „sheet muscles“ - správajú sa ako skupina lineárnych svalov

Ďalší spôsob delenia svalov je na funkčné celky. Pod týmto pojmom sa rozumejú skupiny svalov, ktoré sú pripojené na jeden určitý orgán a ich kontrakciami sa mení tvar daného orgánu. Tieto skupiny sú znázornené na obr. 1.2 a patria sem:

- svaly okolo ústnej štrbiny
- svaly okolo očnej štrbiny
- svaly nosa
- svaly kľemby lebečnej
- samostatné svaly

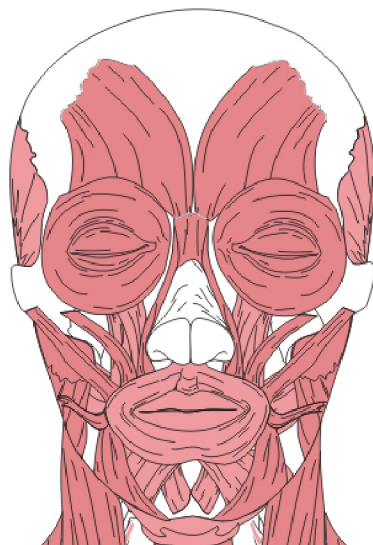
1.1.3 Koža

Koža pokrýva celý povrch ľudského tela a jej hlavnou činnosťou je funkcia stykovej plochy medzi telom a jeho okolím. Má zložitú mikroštruktúru, ktorej základ tvorí päť prepletených sietí kolagénu, nervových vlákien, malých krvných ciev a miazgy pokrytých vrstvou epitelu. Dané usporiadanie je v pravidelných odstupoch prerušované vlasmi a kanálikmi potných žliaz, viď obr. 1.3.

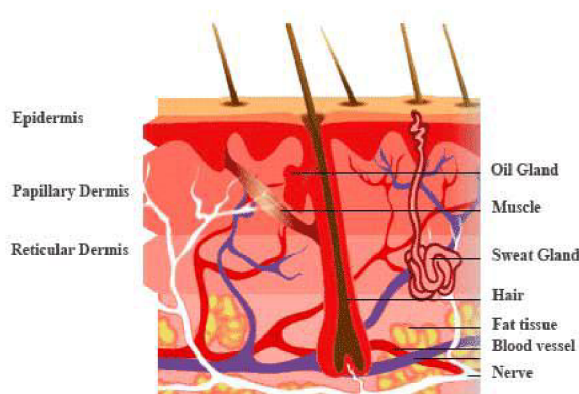
Ľudská koža má vrstvovú štruktúru pozostávajúcu z pokožky, povrchnú vrstvu mŕtvych buniek a tukového tkaniva. Pokožka a tukové tkanivo majú vlastné medzivrstvy.

1.1.4 Oči

Oči sú v skutočnosti oddelený mozgový lalok, ktorý vyrastá zo vzdialeného okraja optického nervu. Pred vonkajším poškodením sú chránené kosťami a viečkom. Oči sú konečným orgánom zrakového zmyslu. Uloženie očí v danom priestore zabezpečuje dôkladnú podporu a miesto pre uchytenie o svaly. Očné svaly umožňujú presné umiestnenie zrakovej osi pomocou nervovosvalovej kontroly a tým pomáhajú k zosúladieniu dvoch priestorových obrazcov pri binokulárnom videní.



Obrázek 1.2: Tzv. funkčné svalové celky



Obrázek 1.3: Anatomický rozbor jednotlivých vrstiev kože

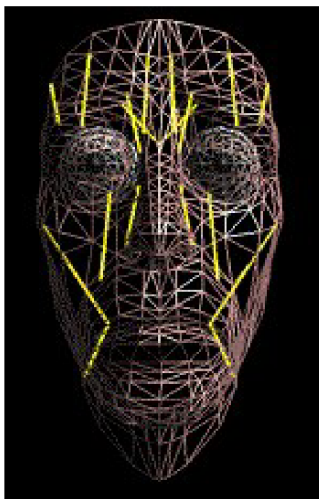
Počas vytvárania počítačom generovaných obrázkov tváre by mala byť na oči sústredená veľká pozornosť. Oči by mali byť nasmerované na určitý objekt a nie hľadieť do prázdna, zrenica by mala byť schopná rozširovať sa a odraz z osvetlenia mimo povrchu dúhovky a rohovky by mal byť starostlivo vymodelovaný. S týmito základnými charakteristikami môže počítačový model zväčšiť realistickosť tváre.

1.2 Základné techniky pri rozpoznávaní mimiky

Predtým, ako budú popísané jednotlivé techniky, ktoré používajú grafici pri vytváraní reálnych mimických výrazov, bude spomenutá tvorbu mimiky z hľadiska anatómie. Ďalej budú popísané základné kategórie mimických výrazov a nakoniec najpoužívanejšie techniky v počítačovej grafike.

1.2.1 Kategorizácia mimických výrazov

Ako bolo spomenuté v predchádzajúcich sekciách, mimické výrazy spôsobujú kontrakcie svalov alebo svalových skupín. Obrázok 1.4 znázorňuje sieťový 3D model ľudskej tváre a výrazne čiary vyznačujú smer a línie, po ktorých sa pohybujú svaly počas kontrakcie resp. počas uvoľnenia.



Obrázek 1.4: Žlté čiary znázorňujú smer „deformácie“ svalov počas kontrakcie

Výskum v oblasti ľudskej mimiky dospel k záveru, že existuje šesť *univerzálnych* kategórií mimických výrazov, ktoré sa dajú rozoznať v rôznych kultúrach[3]. Týmito kategóriami sú hnev, odpor, strach, radosť, smútok a prekvapenie, ako ukazuje obr. 1.5. Každá kategória môže mať širokú škálu intenzity výrazu a rôzne variácie v ich detailoch.

Americký umelec Gary Faigin vo svojej knihe prezentuje znamenitú rozpravu o mimických výrazoch a ich variáciách z umeleckého pohľadu[4]. Opisuje tu každú mimickú kategóriu a jej variácie z hľadiska dopadu na tri hlavné oblasti tváre a ich asociované zvrásnenie pokožky. Dané mimické oblasti tvorí obočie, oči a ústa.

Hnev

Počas hnevu sa vnútorné kútiky obočia stiahnu spolu nadol. Dolný okraj obočia je na rovnakej úrovni ako horné očné viečko. Oko je široko otvorené, ale tlak zníženého viečka zabráni očnému bielku ukázať sa nad úrovňou dúhovky. Ústa sú zatvorené s jemne stlačenou hornou perou. Tento mimický výraz spôsobí horizontálne vrásky nad horným viečkom a vertikálne medzi obočím.

Možné variácie hnevu zahŕňajú krik, zúrivosť a ostrú reč. Tieto varianty môžu zahrňovať jemne stlačené pery s vydutou bradou alebo otvorené ústa so zakrivenou hornou perou a rovnou dolnou perou.

Odpor

Odpor sa stupňuje od pohrdania až po znechutenie. Pri tomto výraze je obočie v klude. Svaly okolo očí sú taktiež uvoľnené alebo oči sú jemne privreté. Horná pera je nesymetricky zdvihnutá do strany. Spodná je v pokoji. Pozdĺž nosa je najhlbšie vrásnenie kože.

Pri opovrhovaní môžu byť očné viečka čiastočne zatvorené s očami hľadiacimi nadol. Pri znechutení (napr. z jedla) je obočie klesnuté, hlavne u vnútorných kútikov. Oči sú väčšinou zatvorené so škúliacim pohľadom. Horná pera môže byť zdvihnutá do roviny, v ktorej sa môžu ukázať zuby. Spodná pera je taktiež jemne vytlačená nahor. Medzi obočím sú vertikálne vrásky a horizontálne sa nachádzajú medzi vnútornými kútikmi obočia a nosom.

Strach

Výraz strachu má rozpätie od obavy až po vydesenie. Obočie je ťahané nahor do maxima. Vnútorné kútiky sú vytočené dohora. Oči sú v strehu. Ústa sú jemne otvorené a napäté po stranách.

Vrásky sa tvoria horizontálne na čele a vertikálne medzi obočím. Pri výraze obavy sú pery jemne stlačené do seba. Spodná pera je jemne vydutá oproti brade. Pri vydesení sú oči a ústa široko otvorené. Horná pera je v klude, zatiaľ čo spodná je široko natiahnutá a sánka je klesnutá na svoje maximum.

Radosť

Pri výraze radosti je obočie nečinné. Horné viečko je jemne posunuté nadol a spodné viečko ťahá tlak lícnych svalov nahor. Ústa sú široko otvorené s kútikmi smerujúcimi k ušiam. Ak sú ústa zatvorené, pery sú tenké a tlačené proti nosnej kosti. Pri otvorených ústach je horná pera rovná s odkrytými vrchnými zubmi. Spodná pera je pri strede rovná a zatočená v oblasti kútikov.

Radosť má veľmi veľa možných prejavov, ako napr. hurónsky smiech, úsmev s otvorenými alebo zatvorenými ústami, potláčaný, nedečkavý, melancholický, falošný, šibalský, vtieravý alebo neviazaný smiech. Falošný úsmev a smiech môžu prezradiť zmenšené vrásky pri rohoch očí alebo jemnou či žiadnou vráskou pod spodnými viečkami.

Smútok

Pri smutnom výraze tváre sú obyčajne vnútorné kútiky obočia vytočené nahor. Koža a jemné tkanivo pod obočím je nahromadené nad horným viečkom. Oči sú jemne zavreté, pretože viečka tlačí nadol tlak tkaniva nad viečkom a pohyb nahor spodného viečka. Počas jemného smútku sú ústa zatvorené.

Smútok má širokú škálu intenzity a variant, zahrňujúce plač s otvorenými alebo zatvorenými ústami, potlačený smútok, takmer plač a depresiu. Tieto možnosti sa môžu taktiež prejaviť úplným poklesom obočia, jemným privretím očí a otvorenými ústami do tvaru obdĺžnika.

Prekvapenie

Pri prekvapivom výraze je obočie vytiahnuté priamo hore do najväčšej možnej miery. Horné viečka sú otvorené čo najširšie, ale dolné viečka sú v klude. Ústa sú voľne otvorené bez svalového napätia a formujú oválny tvar.

1.2.2 Mimický jazyk

Mimický jazyk vyvinutý Carl-Hermanom Hjortsjöm[5] bol jedným z prvých pokusov, ako vyriešiť a systematizovať svalovú aktivitu, ktorá vytvára rôzne mimické výrazy. Hlavnou motiváciou bolo vyvinúť jazyk na opis mimických výrazov. Podľa Hjortsa je mimický výraz



Obrázek 1.5: Šesť základných mimických výrazov

priamy výsledok kombinácie statického aspektu štruktúry tváre a dynamického aspektu. V návaznosti na to, dynamické aspekty sú určené duševnou kondíciou a emočným stavom jednotlivca. Statické stránky sa predpokladajú byť ovplyvnené duševnou kondíciou jednotlivca pracujúcou s endokrinným systémom (týkajúci sa vnútornej sekrécie). Výhradne statická stránka mimiky tváre je určená kosťami tváre a formou jemného tvárového tkaniva. Dynamické resp. živé mimické výrazy sú vyvolané hrou črt, resp. zmenou tvaru a vzhľadu častí tváre, spôsobené činnosťou mimických svalov.

1.2.3 Facial Action Coding System

Na základe vyššie spomenutej knihy vznikol tzv. *Kódovací systém výrazov tváre* (FACS). Na základe poznatkov z anatómie opisuje pohyby mimických svalov, sánky a jazyka [2]. Systém obsahuje 44 základných pohybových jednotiek, tzv. *action units* (AUs). Kombináciami týchto nezávislých pohybových jednotiek sa generujú mimické výrazy. Ak napr. skombinujeme dvíhač vnútorného obočia (AU1), dvíhač líca (AU6), ťahač kútikov pier (AU12) a zvyrazňovač lícných jamiek (AU14), dostaneme výraz šťastia. Nasledujúce tabuľky prezentujú základné pohybové jednotky (tab. 1.1) a základné mimické výrazy vytvorené ich kombináciou (tab. 1.2). Kvôli ich relatívnej jednoduchosti je daný kódovací systém využívaný pri simulovacom mimiky založenej na pohybe svalov alebo pseudosvalov. Pri metódach využívajúcich svalové modely sa prekonáva obmedzenie *interpolácie* (bude vysvetlené neskôr) a tým pádom rozširuje škála výrazov tváre. Modelovanie založené na svaloch matematicky opisuje vlastnosti a správanie jednotlivých vrstiev hlavy. Naproti tomu, modelovanie na základe pseudosvalov iba imituje dynamiku ľudských tkanív s heuristickými geometrickými deformáciami. Napriek jeho popularite obsahuje niekoľko nedostatkov:

- AUs sú rýdzo orientované na lokálne oblasti tváre, ale pohyb tváre je len málokedy

AU	FACS názov	AU	FACS názov	AU	FACS názov
1	Vnútrotný dvíhač obočia	12	Ťahač kútikov pier	2	Vonkajší dvíhač obočia
14	Zvýrazňovač lícných jamiek	4	Spúšťač obočia	15	Stlačovač kútika pery
5	Dvíhač hornej pery	16	Stlačovač spodnej pery	6	Dvíhač líca
17	Dvíhač brady	9	Zvrásňovač nosa	20	Ťahač pier
23	Napínač pery	10	Dvíhač hornej pery	26	Pokles sánky

Tabulka 1.1: Ukážka základných pohybových jednotiek

Základný výraz	Zahrnuté pohybové jednotky
Prekvapenie	AU1, 2, 5, 15, 16, 20, 26
Strach	AU1, 2, 4, 5, 15, 20, 26
Hnev	AU2, 4, 7, 9, 10, 20, 26
Radosť	AU1, 6, 12, 14
Smútok	AU1, 4, 15, 23

Tabulka 1.2: Základné mimické výrazy a k nim prislúchajúce pohybové jednotky

obmedzený na jedno miesto.

- FACS ponúka síce pohyb v priestore, ale nie vznik dočasných komponent; v dočasnom poli pôsobnosti sa efekt presunu z jedného výrazu na druhý stráca

1.3 Tvorba mimiky v počítačovej grafike

Existuje najmenej päť základných postupov pri tvorbe animácie tváre. Medzi tieto postupy patrí *interpolácia*, *priama parametrizácia*, *performance-driven*, *pseudomuskulárna* a *muskulárna* animácia. Cieľom týchto techník je manipulovať povrchom tváre v priebehu času tak, aby tvár nadobúdala v každom časovom okamihu počas animácie požadovaný výraz. Tento proces zahŕňa priamu alebo nepriamu manipuláciu vrcholov polygónov alebo plochou tzv. „kontrolných vrcholov“ v priebehu času. Najviac využívaná technika je *interpolácia*.

1.3.1 Key-Frame Interpolácia

3D hry často používajú sekvenciu *klúčových snímok* na zobrazenie ľudí alebo iných kreatúr v rôznych postojoch[14]. Napr. postava môže mať definované animované sekvencie pre státie, beh, kľáčanie, útok a umieranie. Grafici nazývajú každú jednotlivú pózu, ktorú vytvorili pre 3D model, kľúčovou snímkou, tzv. *key-frame*.

Termín *klúčová snímka* pochádza z kreslenej rozprávky. Pri vytváraní animácie grafik najprv načrtne hrubú sekvenciu snímok na animáciu postavy. Namiesto kreslenia každej vyžadovanej snímky pre konečnú animáciu sa nakreslia iba dôležité, resp. kľúčové snímky. Neskôr výtvarník dokreslí zostávajúce snímky. Dokresľovanie týchto medzismímok je jednoduchšie, pretože predošlá a nasledujúca snímka slúžia ako referencie.

Počítačoví grafici používajú podobnú techniku. 3D grafik spraví kľúčovú snímku pre každý postoj kreslenej postavy. Dokonca aj státie si vyžaduje určitý počet snímok, ktoré

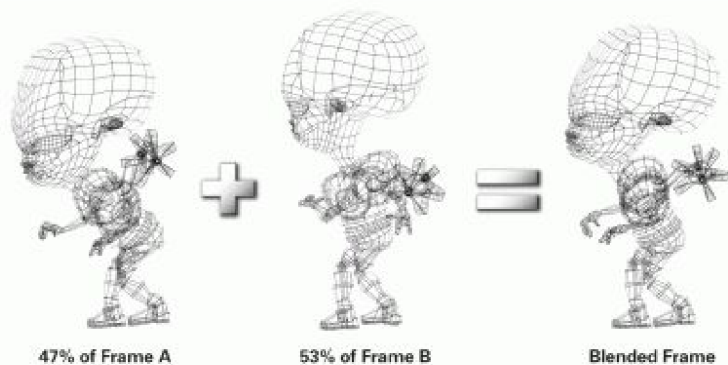
ukazujú presun váhy z jednej nohy na druhú. Každá kľúčová snímka pre model musí obsahovať presne rovnaký počet vertexov a každá snímka musí zdieľať rovnakú prepojitelnosť medzi vertexami. Vertex použitý v kľúčovej snímke určitého modelu musí zodpovedať tomu istému vertexu modelu v každej ďalšej kľúčovej snímke. Celková animovaná sekvencia udržiava tento súlad. Ale keďže sa jedná o animáciu, pozícia jednotlivých vertexov sa môže meniť pri každej snímke.

Pokiaľ vezmeme do úvahy *key-frame* model, hra animuje model vybratím dvoch kľúčových snímok a zmiešaním pozícií zhodných párov vertexov dohromady. Spojenie je váženým priemerom, kde sa celkový súčet váh rovná 100 percent. Aplikácia môže použiť vertex program na spojenie identických vertexov z dvoch kľúčových snímok. Toto zmiešavanie môže zahŕňať ďalšie procesy, ktoré objasňujú spájaný výraz a vytvárajú lepšiu predstavu reality. Obyčajne aplikácia špecifikuje jednu pozíciu pre každý vertex, ale pri *key-frame* spájaní má každý vertex dve pozície, ktoré sú kombinované s jednotným váhovým faktorom.

Interpolácia pomocou kľúčových snímok predpokladá, že počet a poradie vertexov je rovnaké vo všetkých kľúčových snímkach pre daný model. Tento predpoklad zaručuje, že vertex program vždy spája správne páry vertexov. Následujúci zlomok kódu vytvára medzipozíciu vertexu z dvoch kľúčových snímok:

$$blendedPosition = (1 - weight) * keyFrameA + weight * keyFrameB \quad (1.1)$$

Premenné *keyFrameA* a *keyFrameB* obsahujú XYZ súradnice spracovávaného vertexu na oboch kľúčových snímkach. Veľkosť premennej *weight* môže byť v intervale $\langle 0, 1 \rangle$. Ak sa veľkosť rovná napr. 0.33, program pridá 67 percent pozície vertexu kľúčovej snímky A k 33 percentám pozície vertexu v kľúčovej snímke B. Obrázok 1.6 ukazuje príklad daného typu animácie.



Obrázok 1.6: Príklad spájania vertexov pomocou metódy *key-frame*

Existuje niekoľko typov interpolácií. Medzi dve najbežnejšie formy pre *key-frame* interpoláciu patrí lineárna a kvadratická interpolácia. Pri lineárnej interpolácii sa prechod medzi pozíciami vertexu odohráva v konštantnom tempe:

$$blendedPosition = positionA(1 - f) + positionBf \quad (1.2)$$

Ako sa hodnota f mení od 0 po 1, mení sa prechodná pozícia medzi *positionA* a *positionB*. Ak sa f rovná 0, prechodná pozícia je v skutočnosti začínajúca *positionA*. Ak sa f rovná 1, prechodná pozícia je konečná pozícia, čiže *positionB*.

Lineárna interpolácia je často využívaná, ale niekedy je potrebné zmeniť rýchlosť presunu v priebehu času. V niektorých prípadoch je potrebné, aby presun z pozície *A* do *B* začal pomaly a postupne sa časom zrýchľoval. Pre tento prípad je možné použiť kvadratickú interpoláciu do tvaru:

$$intermediatePosition = position1 * (1 - f^2) + position2 * f^2 \quad (1.3)$$

1.3.2 Performance-Based animácia

Animácia zahŕňa používanie informácií získaných z merania skutočných ľudských výrazov na vytvorenie umelej povahy[1]. Daný postup bol prvý krát použitý v roku 1985 pre film Tony de Peltrie. Prvým krokom v rámci vývoja bolo digitalizovanie 20 odlišných výrazov skutočnej osoby. Bolo vytvorené prepojenie medzi tzv. neutrálnym výrazom (E_0) skutočnej tváre a neutrálnym výrazom animovanej postavy. Vzhľadom na to, že počet vertexov definujúcich animovanú tvár bol väčší než počet bodov definujúcich skutočnú tvár, bolo použité prepojenie $1 : N$. To znamená, že pre každý bod na skutočnej tvári bola definovaná skupina bodov tváre kreslenej postavy. Avšak každému bodu tváre postavy zodpovedal práve jeden bod v skutočnej tvári.

Po nadefinovaní prepojení jednotlivých bodov medzi oboma tvármi boli vypočítané mimické výrazy pomocou funkcie, ktorá mapovala reálne výrazy na model tváre. Funkcia bola založená na odlišnosti medzi skutočnými mimickými výrazmi. Rozdiel medzi neutrálnym a cieľným výrazom skutočnej tváre bol pridaný do neutrálného výrazu animovanej tváre. Rozdiel pre každý bod v skutočnej tvári bol pridaný do jeho zodpovedajúcej skupiny bodov modelu. Týmto spôsobom sa však na model budú mapovať z ľudskej tváre prehnane výrazy. Aby bolo možné predísť danému stavu, bol použitý určitý zosilňujúci faktor na zvýšenie alebo zníženie rozsahu pridaných zmien výrazov.

Bola vyvinutá knižnica kľúčových výrazov aplikovaním mapovaním výrazov nezávisle na päť oblastí tváre: ľavé a pravé obočie, ľavé a pravé viečka a zvyšok tvára zahrňujúci ústa. Taktiež bol vyvinutý trojrozmerný algoritmus na interpoláciu pomocou krivky, ktorý generoval prechodné výrazy medzi kľúčovými mimickými výrazmi.

1.3.3 Priame parametrizované modely

Ideou daného postupu bolo vytvoriť určitý ucelený všeobecný model, ktorý by generoval široké spektrum modelov tváre a mimických výrazov založenom na veľmi malej množine vstupných kontrolných parametrov. Cieľom bolo poskytnúť kontrolu nad mimikou a štruktúrou tváre pomocou nastavených hodnôt parametrov.

V ideálnom prípade by existoval model, ktorý by umožnil všetky možné tváre s akýmkoľvek mimickými výrazmi, ktoré by boli špecifikované vybratím vhodného zoznamu hodnôt parametrov. Modely vytvorené týmto spôsobom boli pomerne vzdialené od reality, ale umožňovali širokú škálu mimických výrazov pre celkom široký výber štruktúr tváre.

Na rozdiel od techník využívajúcich interpoláciu, parametrizovanie umožňuje priamu kontrolu špecifických konfigurácií tváre. Kombináciami parametrov sa dosiahne široká škála mimických výrazov s relatívne nenáročným výpočtovým výkonom.

Problémom je určiť efektívny zoznam kontrolných parametrov a implementovať model, ktorý používa tieto parametre na generovanie požadovanej škály tvári a mimických výrazov. Doteraz vyvinuté parametrické modely boli veľmi primitívne a nenárodné. Prístup k implementácii bol aplikovaním operácií ako rotácia, približovanie, vyváženie pozície a interpolácia v kombinácii s lokálnymi oblasťami tváre.

Tieto modely boli vyvinuté s malým teoretickým základom a bez starostlivého zamerania na anatómiu tváre. Boli experimentálne odvodené, aby prezentovali viditeľné povrchové črty tváre založené na pozorovaní a všeobecných údajoch fundamentálnych štruktúr tváre. Poskytnuté kontrolné parametre obsahovali:

- *Výraz tváre* - otvor viečka, oblúk obočia, oddelenie obočí, krúženie čeluste, šírka úst, výraz úst, pozícia hornej pery, pozícia kútika úst, pohľad oka
- *Štruktúru* - šírka čeluste, tvar čela, dĺžka a šírka nosa, tvar líca, brady a krku, veľkosť oka a medziodnej medzery, proporcie oblastí tváre a celkové proporcie tváre

Okolo desať mimických parametrov umožňuje animátorovi stanoviť a kontrolovať širokú škálu mimických výrazov. A okolo dvadsať parametrov je použitých na kontrolu limitovanej škály štruktúr tváre.

1.3.4 Animácia založená na pseudosvaloch

Komplexné vzájomné pôsobenie medzi tvárovým tkanivom, svalmi a kosťami a medzi svalmi navzájom majú za následok činnosť bežne nazývanú *mimika tváre*. Je očividné, že tieto vzájomné pôsobenia produkujú obrovský počet pohybových kombinácií. Myšlienkou pseudosvalového prístupu nie je presne simulovať detailnú anatómiu tváre, ale vytvoriť model s malým množstvom kontrolných parametrov, ktoré budú napodobňovať základné mimické pohyby svalov.

Na tento spôsob bolo vyvinutých niekoľko modelov. Jedným z nich bol model vyvinutý profesorom Magnenat-Thalmannom[8], v ktorom parametre kontrolovali tzv. *abstract muscle action*(AMA) procedúry. Tento prístup je založený na empirických modeloch a nie na fyzickej simulácii. AMA procedúry sú podobné ale nie rovnaké ako FACS činné jednotky. U FACS jednotkách sú svalové procedúry komplexnejšie než pri priamom parametrizovanom prístupe spomenutom vyššie. AMA procedúry pracujú na špecifických oblastiach tváre, kde každá procedúra imituje prácu svalov alebo skupiny blízko spojených svalov. Napr. vertikálny pohyb sánky je zodpovedný za otvorenie úst. Skladá sa z niekoľkých pohybov: zníženie kútikov úst, zníženie spodnej pery a časti hornej pery a celkové zaokrúhlenie pier. Bolo vyvinutých 30 AMA procedúr, ako napr. zavretie hornej pery, vertikálny pohyb ľavého oka atď. Tieto procedúry nie sú nezávislé, takže je veľmi dôležité správne poradie jednotlivých akcií.

1.3.5 Netypizované deformácie(Freeform Deformations)

Netypizovaná deformácia(ďalej ND) je technika na deformovanie pevných geometrických modelov. Môže byť použitá na kontrolu zmeny tvaru povrchových primitív každého typu alebo stupňa ako napr. roviny, kvadratické plochy, parametrické povrchové polia alebo implicitne definované povrchy. Daný spôsob je založený na deformovaní imaginárnych 3D plôch obklopujúcich objekt alebo časť objektu. Prenesene sa dá povedať, že sa jedná o deformáciu telesa z umelej hmoty, v ktorom sa taktiež deformujú vložené objekty. Pri ND sa deformované objekty prenášajú z jedného 3D priestoru do druhého využívajúc Bernsteinov polynóm. Deformácie sú špecifické presunom mriežky kontrolných bodov z ich počiatočných pozícií.

Lokálna deformácia

Lokálna deformácia je jedným z typov netypizovanej deformácie a teda ďalším spôsobom, ako získať z existujúceho modelu nový tvar. Lokálnou deformáciou sa rozumie zmena tvaru aplikovaná len na určitý diel objektu a nie na celkový objekt. Tento spôsob si vyžaduje vyznačenie oblastí objektu pre transformáciu a špecifikáciu transformácie ako takej.

Magnenat-Thalman[7] opísal päť spôsobov na označenie deformačnej oblasti polygónálneho modelu a štyri metódy transformácie. Navrhované schémy na vyznačenie sú nasledujúce:

- vybratie špecifického počtu vertexov
- vybratie vertexov objektu ohraničených ďalším objektom
- vybratie vertexov vnútri valcovitého rezu
- vybratie vertexov so špecifickou hodnotou farby
- vybratie vertexov, ktoré uspokojujú množinu operácií medzi regiónmi špecifikovanými jedným alebo viacerými zo štyroch predošlých metód

Navrhované metódy transformácie sú:

- Percento k vertexu - každý vybraný vertex je posunutý smerom k referenčnému bodu percentuálne vyjadrenou vzdialenosťou medzi vertexom a referenčným bodom
- Vodiaci posun - vektor posunutia je vypočítaný medzi bodom A a bodom B a určité percento tohoto vektoru je aplikované na všetky vybrané vektory
- Škálovanie roviny - škálovanie je aplikované na každý vybraný vertex; stupeň škálovania je priamo úmerný vzdialenosti medzi každým vertexom a stanovenej rovine
- Premennivý presun - faktor premenlivosti je použitý na kontrolu transformácie vnútri vyznačenej oblasti. Ak je premenlivý faktor rovný nule, potom sa všetky vyznačené vertexy budú posúvať rovnakým spôsobom. Ak je veľkosť faktora 1, vertex nachádzajúci sa v strede oblasti bude maximálne ovplyvnený posunom, avšak vertexy na hranách oblasti nebudú ovplyvnené vôbec. Vertexy sú modifikované podľa všeobecnej *rozkladacej funkcie* založenej na relatívnej pozícii k stredu oblasti.

Racionálna netypizovaná deformácia

Určítym rozšírením *free-form* deformácie je tzv. *racionálna deformácia*. Tá používa základné logické funkcie pri zostavovaní deformácie. Tieto funkcie obsahujú *váhový faktor* pre každý kontrolný bod v kontrolnej mriežke v tvare rovnobežnostena, väčšinou pravouhlého hranola.

Daný faktor poskytuje dodatočnú mieru slobody počas manipulácie s deformáciou oblasti tváre. Ak je hodnota váhového faktoru pre každý kontrolný bod jednotná, jedná sa o obyčajnú netypizovanú deformáciu.

Každý jednotlivý pohyb svalu je kontrolovaný výmenou kontrolných bodov na mriežke definovaných v deformačnej oblasti tváre. Pri simulovaní pohybov svalov na tvári sa definujú povrchové oblasti, ktoré zodpovedajú anatomickým oblastiam požadovaných svalových kontrakcií. Regulačná miera hranola je definovaná okolo každej svalovej oblasti. Svalové deformácie zodpovedajúce naťahovaniu, pomliaždeniu, zväčšovaniu a stláčaniu vytýčenej

deformačnej oblasti sú simulované vzájomne pôsobiacim premiestňovaním kontrolných bodov na mriežke a zmenou hodnoty váhového faktora u každého kontrolného bodu. Taktiež je využívaná interpolácia pri zisťovaní deformácií bodov ležiacich na hraniciach príslušných deformačných oblastí. Premiestňovanie kontrolných bodov je určitá forma napodobňovania činnosti svalov. Výsledná deformácia sa môže celkom dôverne zhodovať skutočnej práci svalov. Avšak špecifikácia premiestňovania kontrolných bodov je jednoduchšia než samotná simulácia práce svalu.

1.3.6 Reálna animácia tváre

Anatómiu hlavy a tváre tvorí zložitá pospájanie kostí, chrupaviek, svalov, nervov, ciev, žliaz, tukového tkaniva, spojovového tkaniva, kože a vlasov. Na vytvorenie takéhoto modelu je potrebné logicky poprepájať všetky spomenuté časti ľudskej hlavy. Jedná sa o najnáročnejší spôsob animácie ľudských výrazov. Dodnes nebol takýto komplexný a detailný model tváre vytvorený. Avšak pre zjednodušenie je možné zredukovať túto zložitú anatómiu na vytvorenie modelu, ktorého mimika bude založená na fyzických vlastnostiach svalov a kože.

V poslednom čase sa simulácia hybnosti kožného tkaniva dostáva do pozornosti hlavne pri plánovaní chirurgických zákrokov, kedy je vhodné nasimulovať si dôsledky operácie pred jej uskutočnením. Niektoré techniky boli aplikované na animáciu tváre pre zlepšenie dynamických vlastností modelov, ako napr. pri zvráštení čela. Bližšie bude predstavená technika využívajúca mechanizmus svalov, ktorá bude sčasti použitá aj pri vytváraní modelu v tejto práci.

1.4 Zvolené metódy pre tento projekt

Pre daný projekt boli použité techniky deformačných zón a tvorba mimických výrazov bude založená na posúvaní vertexov pomocou svalových vektorov. V princípe sa vytvorí model tváre, do ktorého budú vložené ďalšie objekty, ktorých úlohou bude vymedzovať množinu vektorov, na ktoré sa prevedie vertexový presun. Pre zjednodušenie hľadania vertexov nebudú objekty trojrozmerné, ale jednoduché 2D plochy, t.j. trojuholníky a obdĺžniky. Aby bolo možné vyselektovať hľadané vertexy v priestore, dané deformačné plochy budú naklonené pod určitým uhlom, resp. budú s osou z zvierajú určitý uhol.

1.4.1 Lokalizácia vertexov v deformačných zónach

Hľadanie vertexov v deformačných oblastiach bude vykonané v dvoch fázach. V prvej fáze budú hľadať vertexy pomocou osí X a Y . V prípade trojuholníka bude pre vyhľadávanie použitá všeobecná rovnica priamky a rovnica vzájomnej polohy priamky a bodu v rovine:

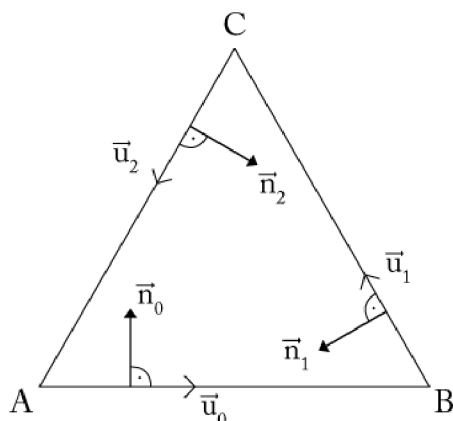
$$ax + by + c = 0 \tag{1.4}$$

Uvažujeme určitý trojuholník ABC , ktorý tvoria priamky \overleftrightarrow{AB} , \overleftrightarrow{BC} a \overleftrightarrow{CA} . Pre správne výsledky algoritmu je nutné správne nastaviť smerové vektory \vec{u} , resp. vektory \overrightarrow{AB} , \overrightarrow{BC} a \overrightarrow{CA} (obr. 1.7).

Smerový vektor \overrightarrow{AB} vyjadríme nasledujúcim spôsobom:

$$\vec{u} = B - A \Rightarrow \vec{u} = (B[x] - A[x], B[y] - A[y]) = (p, q) \tag{1.5}$$

Pre výpočet bude ale dôležitý normálový vektor. Keďže normálový vektor \vec{n} je kolmý na smerový vektor \vec{u} , jeho súradnice budú dané vzťahom:



Obrázek 1.7: Trojuholník ABC s naznačenými smerovými a normálovými vektormi

$$\vec{n}(a, b) = \vec{u}(-b, a) \quad (1.6)$$

Pre zistenie polohy bodu voči rovine bude ďalej potrebné vypočítať hodnotu konštanty c . Tá sa vypočíta dosadením súradníc pôsobiska vektoru do všeobecnej rovnice priamky za premenné x a y , z čoho výjde vzťah:

$$c = -ax - by \quad (1.7)$$

Keď sú známe všetky potrebné premenné, resp. a , b a c , za premenné x a y sa dosadia súradnice spracovávaného vertexu a z výsledku je najdôležitejšia hodnota znamienka. Tento cyklus sa opakuje aj pre smerové vektory \vec{BC} a \vec{CA} . Ak po dosadení hodnôt do všetkých troch rovníc priamok tvoriacich trojuholník bude vo výsledku rovnaké znamienko, znamená to, že daný bod sa nachádza v ploche trojuholníka[9].

V druhom kroku sa zistí, či daný vertex patrí do deformačnej plochy aj v rámci priestoru. V prvej fáze sa mohol vybrať aj vertex, ktorý síce spĺňa podmienku výskytu v deformačnej ploche, ale už nie výskytu v určenom priestore. Konkrétne sa jedná o vertexy na zadnej strane hlavy. Na to nám posluži už predošlá zmienka o naklonení deformačných plôch. V tomto prípade sa ako priestorové hraničné hodnoty vyberú minimálna a maximálna hodnota z -ovej osi z vertexov tvoriacich deformačnú plochu. Čiže ak je hodnota z -ovej súradnice spracovávaného vertexu v rozsahu $\langle \min Z, \max Z \rangle$, daný vertex spĺňa aj druhú podmienku. Takto vybraná množina vertexov sa bude presúvať pomocou algoritmu opísaného v nasledujúcej časti.

1.4.2 Mimika tvára založená na svalových vektorech

Ako názov napovedá, svalové vektory nesledujú hlavný smer a upnutie svalových vlákien. Zatiaľ čo skutočné svaly pozostávajú z niekoľkých samostatných vlákien, počítačový model predpokladá spoločný smer a úpon. S touto zjednodušenou podmienkou môže byť každý sval opísaný smerom a veľkosťou v dvoch aj troch rozmeroch. Pohyb svalu je orientovaný smerom k bodu úponu svalu na kosť a veľkosť stlačenia svalu závisí na pružnej konštante svalu a vyvinutom tlaku pri svalovej kontrakcii. Pri lineárnych alebo paralelných svaloch je obklopujúca koža sťahovaná smerom k statickému uzlu úponu svalu na kosť, až kým sa pri

určitej vzdialenosti sila nezmenší na 0. Pri zvieracích (kruhových) svaloch je kožné tkanivo stláčané smerom k imaginárnemu strediu, ako napr. pri nťahovaní sieťovej tašky. Skupina priamych svalov (napr. čelový sval) je široká rovná plocha radov svalového vlákna, ktorá nepremiestňuje svoje body úponu. Následkom toho je kontrakcia týchto svalov sústredovaná do určitého vymedzeného uzla než aby každý sval smeroval do samostatného bodu. V skutočnosti je takýto sval vlastne skupinou takmer súbežných svalov rozprestrených na určitej ploche.

Správanie lineárnych, zvieracích a skupinových svalov vychádza z kontraktí svalového tkaniva. Z tohoto dôvodu je nutné pri svalovom modeli vypočítať nové umiestnenie pre uzly deformácie, čím sa zabezpečí skutočná zmena tvaru kože.

Pri použití svalových vektorov sa kontrakcie svalov znázorňujú pomocou deformácie kože. Každý sval má určitú zónu vplyvu, pri ktorej určitou veľkosťou ovplyvňuje a tým pádom aj premiestňuje vertexy tvoriace kožu. Každý sval obsahuje pole smerových vektorov, začiatkový bod a bod, do ktorého dané vektory vstupujú. Medzi hlavné výhody patrí napr. nezávislosť na sieťovom modeli tváre, kde môžu byť použité rôzne modely. Ďalej je to možnosť rýchleho a uceleného zobrazenia a pridané parametrické vyjadrenie výrazov môže kontrolovať niekoľko svalových skupín. Medzi nevýhody patrí napr. zisťovanie pozície svalov, ktoré väčšinou býva časovo náročné. Taktiež do animácie nezahrňuje zakrivenie svalu a zložitost sa zväčšuje, keď je napr. jeden vertex pod vplyvom niekoľkých pohybov svalov. V kombinácii s určením deformačných oblastí je možné získať celkom dôveryhodnú simuláciu svalových kontraktí.

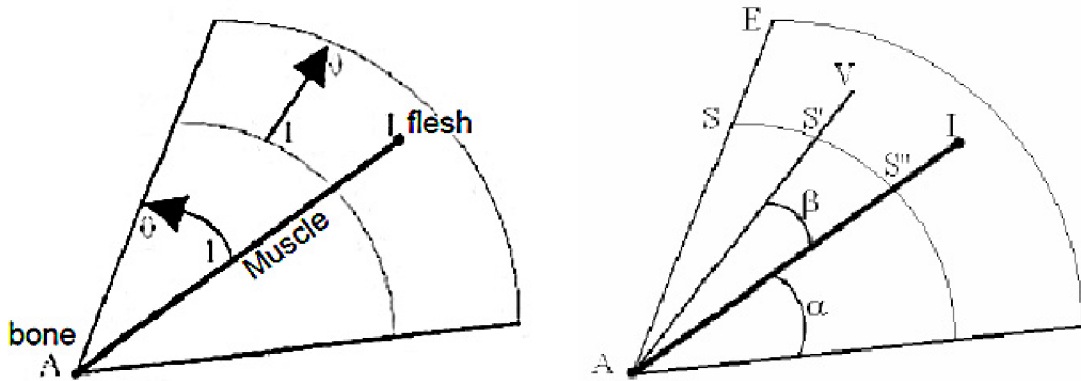
Parametre svalového vektora

Pri definovaní svalového vektora vychádzame z predpokladu, že sval je pripevnený na dvoch bodoch:

- bod A - pripevnenie svalu na kosť
- bod I - úpon svalu, pohyblivá plocha

V tomto prípade možno sval považovať za vektor \overrightarrow{AI} (obr. 1.8). Ak nenastane žiadna kontrakcia, body A a I sa nepresúvajú, čiže vektor si zachováva svoju dĺžku. Dané správanie sa podobá na magnet, ktorý priťahuje všetky vertexy v zóne vplyvu. Bod V predstavuje vrchol modelu, uhol α počiatkový uhol medzi bodom V a vektorom \overrightarrow{AI} . Uhol β zobrazuje maximálnu veľkosť uhla, ktorý môže zvierať určitý vertex s vektorom, aby na neho pôsobila deformácia. Svalová kontrakcia zoslabuje, čím sa veľkosť uhla α blíži veľkosti uhlu β . Oblúk S predstavuje počiatkovú hranicu, od ktorej začína kontrakcia smerom k oblúku E slabnúť. Na vertexy nachádzajúce sa v oblúku E pôsobí kontrakcia nulovou veľkosťou.

Nasledujúce rovnice slúžia na výpočet veľkostí rôznych koeficientov, pomocou ktorých sa v konečnom výsledku zistí veľkosť posunutia každého vertexu. Najdôležitejší spomedzi faktorov je tzv. faktor kontrakcie Cf . Môže nadobúdať hodnoty $< 0, 1 >$ a jeho veľkosť nám určí intenzitu posunu vertexov. δ_A je tzv. koeficient slabnutia súvisiaci s uhlovou vzdialenosťou medzi AV a AI . δ_R je koeficient slabnutia, ktorého veľkosť závisí od vzdialenosti vertexu V od oblúku S , ak sa V nachádza v ploche ohraničenej oblúkmi S a E . Taktiež je demonštrovaný príklad posunutia kontrolnej mriežky vertexov, ak sa veľkosť faktoru kontrakcie rovná 0,6.



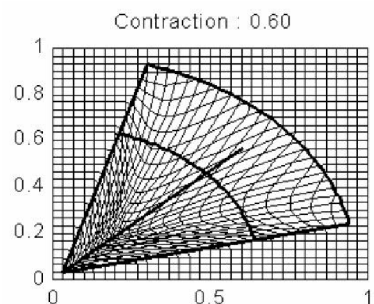
Obrázek 1.8: Vektor simulujúci prácu svalu počas kontrakcie

$$\delta \vec{V} = -Cf \cdot \delta \cdot \vec{AV} \quad (1.8)$$

$$\delta_A = \frac{\cos \beta - \cos \alpha}{1 - \cos \alpha} \quad (1.9)$$

$$\delta_R = \cos \left(\frac{\| \vec{S'V} \| \pi}{\| \vec{S'E} \| 2} \right) \quad (1.10)$$

$$\delta = \begin{cases} \delta_A \cdot \delta_R & \text{ak } \| AV \| > \| AS \| \\ \delta_A & \text{inak.} \end{cases} \quad (1.11)$$

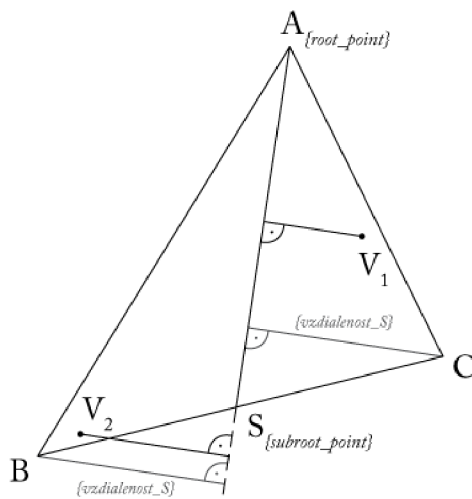


Z daných vzťahov vyplýva, že veľkosť posunu jednotlivých vertexov je tým menšia, čím daný vertex V zvierá s vektorom \vec{AV} väčší uhol a čím je daný vertex vzdialenejší od oblúka S [6].

Avšak v tomto projekte sa použije zjednodušená varianta, ktorá nebude využívať uhlovú vzdialenosť vertexu od pevného bodu uchytenia svalu na kosť. Využije sa kombinácia absolútnych vzdialeností vertexu od bodu úchyty. Za imaginárny bod úchyty sa vyberie vertex z množiny vertexov nachádzajúcich sa v deformačnej oblasti, tzv. *root point*. Všetky ostatné body vytvoria s daným bodom imaginárne spojenie, resp. trajektóriu pohybu. Keďže deformačné oblasti sú v tvare trojuholníka, v strede strany ležiacej oproti *root pointu* sa vytvorí imaginárny bod S , ktorý bude slúžiť ako určitá zložka na výpočet novej polohy vertexov. Uvažujeme trojuholník ABC , ktorého *root point* tvorí bod A . Hodnota faktoru kontrakcie Cf bude plne ovplyvňovať len vertexy nachádzajúce sa na imaginárnej spojnici bodov A a S (obr. 1.9). Ak sa vertex bude nachádzať v nejakej vzdialenosti od danej spojnice, pre tento vrchol V sa hodnota faktoru kontrakcie ešte vynásobí podielom rozdielu dĺžky úsečky CS a vzdialenosti vertexu od spojnice AS a dĺžky úsečky CS :

$$Cf_1 = Cf \cdot \left(\frac{|CS| - |V\vec{AS}|}{|CS|} \right) \quad (1.12)$$

Konečnou hodnotou faktoru sa vynásobia všetky 3 vzdialenosti posúvaného vertexu od *root pointu*, ktoré sa pridajú k pôvodným súradniciam. Tým pádom vzniknú nové súradnice vertexu. Daná operácia sa vykoná pre každý vertex. Bude potrebné pamätať si pôvodné



Obrázek 1.9: Deformačná oblasť znázorňujúca vzdialenosť vertexov V od vektoru AS

umiestnenie vertexov, keďže užívateľ bude môcť ovládať len zmenu veľkosti faktora Cf . Následkom toho sa pri každej zmene veľkosti Cf vytvoria nové súradnice vertexu z pôvodných a nie aktuálnych súradníc.

Pri deformačných plochách v tvare obdĺžnika bude postup podobný, len s tým rozdielom, že daný faktor Cf sa použije v plnej miere na všetky vertexy, ktoré sa nachádzajú v danej ploche pôsobenia. Budú vybraté 2 smerodajné body a to najvyšší vertex A a najnižší vertex C obdĺžnika, ktoré by spojením vytvorili stranu stranu obdĺžnika podobnú veľkosťou a umiestnením. Vektor \overline{AC} vzniknutý ich spojením bude slúžiť ako smerový vektor pre posúvanie jednotlivých vertexov. Vznikne nový parameter Cf_2 , ktorý bude obsahovať pomer z -ovej a y -ovej vzdialenosti bodov AC . Keďže v tomto prípade sa budú vertexy posúvať len zvislo a do hĺbky, x -ovú súradnicu neuvažujeme. Pre každý vertex bude vypočítaná pôvodná y -ová vzdialenosť od bodu C . Ak sa napr. nastaví faktor kontrakcie na hodnotu 0,5, danou hodnotou sa vynásobí y -ová vzdialenosť posúvaného vertexu od bodu C , ktorá sa pridá k pôvodnej súradnici. U z -ovej súradnice sa Cf vynásobí Cf_2 , až potom z -ovou vzdialenosťou od C a výsledok sa pridá k pôvodnej súradnici.

1.5 Vertex Shader

Shader je počítačový program určený pre spracovanie priamo na grafickej karte. Takýto program je najčastejšie napísaný v jazyku Cg (vyvinutý firmou nVidia), ktorý je univerzálny ako pre DirectX tak aj pre OpenGL. Ďalej to môže byť jazyk HLSL vyvinutý firmou Microsoft pre DirectX alebo GLSL pre OpenGL. Kód je prekladačom preložený do assembleru priamo pre danú grafickú kartu. Všetky tri jazyky sa syntakticky podobajú na jazyk C. V súčasnej dobe existujú tri typy shaderov: vertex, pixel a geometry shader[13].

Vertex shader je program, ktorý sa vykoná na každom vertexe vstupného geometrického modelu. Slúži k transformácii 3D scény do 2D plochy obrazovky. Prvý krát bol Vertex Shader podporovaný u DirectX od verzie 8.0 a vyššie. Čo sa týka hardwareovej podpory, prvý krát mal Vertex Shader podporu až v roku 2001 u grafickej karty GeForce 3 vyvinutej spoločnosťou nVidia.

Kapitola 2

Návrh riešenia

Cieľom práce bolo vytvoriť 3D model tváre a program, ktorý by ho načítal a pomocou určitých algoritmov presúval niektoré vertexy tak, aby z toho vznikol mimický výraz. Z tohoto dôvodu bude projekt rozdelený na 2 hlavné časti: modelovacia a programovacia časť

2.1 Modelovacia časť

2.1.1 Blender

Pre namodelovanie 3D objektu tváre bude použitá aplikácia **Blender**[12]. Blender je open source software šírený pod licenciou **GNU GPL** na modelovanie a vykresľovanie 3D počítačovej grafiky, animácií a filmov s využitím rôznych techník, napr.: raytracing, radiosity, scanline rendering, GI. Vlastné rozhranie je vykresľované pomocou knižnice **OpenGL**. OpenGL umožňuje nielen hardvérovú akceleráciu vykresľovania 2D a 3D objektov, ale predovšetkým ľahkú prenositeľnosť na všetky podporované platformy.

V Blenderi je možné okrem statických modelov a jednoduchých animácií vytvárať rôzne interaktívne 3D aplikácie ako napr. počítačové hry, priemyselné, vedecké a architektonické vizualizácie. Umožňuje to vstavaný modul *Game Engine*, ktorý môžu plne využívať aj užívatelia bez programátorských znalostí. Pre zložitejšie aplikácie je užitočné využiť možnosti skriptovacieho jazyka **Python**. Game Engine má v sebe od verzie 2.40 integrovaný fyzikálny engine *Bullet*, ktorý umožňuje verne simulovať dynamiku pevných telies[13].

2.1.2 Deformačné oblasti

Po dokončení 3D modelu tváre bude výsledný model doplnený o ďalšie objekty, ktoré budú mať funkciu tzv. deformačných zón. Tieto objekty budú mať tvar telies s jednoduchou geometriou, napr. trojuholník alebo obdĺžnik. Tieto objekty budú pridané do modelu tak, aby ich priestorom prechádzali určité vertexy modelu tváre. Dané objekty sa nebudú vykresľovať, ale budú slúžiť na ohraničenie vertexov, ktoré budú deformované. Výsledný model sa vyexportuje do formátu **.obj**, kde sa o ostatné veci postará programová časť.

2.1.3 Formát .obj

Prezentuje ASCII formát objektového súboru[11]. Je to otvorený formát a bol prijatý mnohými firmami, ktoré vyvíjajú 3D grafický software. Vytvorené obj súbory využívajú mnohé grafické programy ako napr. Blender, 3D Studio Max, Maya atď. Podporuje polygonálne objekty, rovnako aj voľné objekty ako napr. krivky. Polygonálna geometria používa body,

čiaru a steny telies. Tento jednoduchý dátový formát reprezentuje 3D geometriu zvlášť - menovite pozíciu každého vertexu, súradnice textúr, ktoré súvisia s vertexami, normálu na každom vertexe a plochy, ktoré vytvárajú polygóny.

2.2 Programovacia časť

Po vytvorení modelu tváre spolu s deformačnými oblasťami a vyexportovaní do OBJ súboru bude potrebné vytvoriť aplikáciu, ktorá:

- načíta obj súbor, ktorý obsahuje: vertexy, hrany, plochy, normály, uv súradnice, pozície jednotlivých objektov a textúru
- načíta jednotlivé deformačné oblasti - dané oblasti ale nebude vykresľovať na obrazovku
- bude schopná pomocou kláves meniť polohu vertexov tváre nachádzajúcich sa v jednotlivých deformačných plochách a v priebehu času daný posun vykresľovať na obrazovke
- uloží novovytvorený výraz ako obrázok vo formáte JPG

Dané algoritmy sa budú vyvíjať v jazyku C++. Na vykresľovanie modelu sa využije knižnica **OpenSceneGraph**.

2.2.1 OpenSceneGraph

OpenSceneGraph(OSG) je open source vysokovýkonný 3D grafický nástroj, ktorý sa používa vývojármi počítačových aplikácií v oblasti počítačových hier, virtuálnej reality, modelovania alebo vizuálnej simulácii. Bol napísaný v C++ s využitím knižnice OpenGL. Taktiež je to multiplatformový nástroj, ktorý funguje rovnako ako pod systémom Windows, tak aj Linux, FreeBSD, Mac OS X a Solaris.

Prezentuje objektovo orientovanú nadstavbu nad OpenGL, čím oslobodzuje programátora od implementácie nízkoúrovňových funkcií a poskytuje množstvo užitočných pomôcok pre vývoj grafických aplikácií. Pre urýchlenie vývoja aplikácie OSG obsahuje veľké množstvo prídavných knižníc, poskytuje rôzne optimalizácie vykresľovania a najnovšie funkcie OpenGL. Nástroj podporuje načítavanie a ukladanie širokého spektra formátov rastrových obrázkov a 3D modelov, ako napr. OBJ, 3DS, JPEG alebo PNG[13].

2.2.2 Tvorba mimických výrazov

Pomocou definovaných kláves bude možné ovládať intenzitu premiestňovania určitej skupiny vertexov, ktoré sa nachádzajú v určitej deformačnej oblasti. Pre každú deformačnú oblasť bude vyčlenená dvojica susediacich kláves, ktorá bude zvyšovať/znižovať silu kontrakcie zapríčínujúcu premiestňovanie vertexov tváre.

Kapitola 3

Implementácia

Ako bolo spomenuté v predošlej kapitole, práca na projekte bola rozdelená na 2 hlavné časti: modelovacia a programovacia. Pre naprogramovanie aplikácie bolo dôležité použiť realistický model tváre. Na modelovanie tváre bol použitý open-source software Blender. Renderovací algoritmus bol riešený v jazyku C++ s využitím knižnice `OpenSceneGraph`.

3.1 Modelovanie

3.1.1 Sculpt mode

Základ práce tvoril 3D kváder, z ktorého sa postupnou úpravou vytvoril model tváre. Prvým krokom úpravy bol v Blenderi `sculpt mode`. Jedná sa tzv. sochársky spôsob, kde sa postupným deformovaním pôvodného objektu vytvára výsledný tvar. Po vložení objektu bolo najprv nutné **zahustiť** daný objekt pomocou *subdivision surface* [13]. Jedná sa o systematické rovnomerné pridávanie vertexov a hrán. V prvom kroku bol formovaný základný tvar hlavy, ďalej boli naznačené obrysy uší, obočia, nosa, brady a krku. Po ďalšom *zahustení* boli vytvarované ústa a oči. Ďalej bolo možné vytvoriť nosné dierky a naznačiť svaly na tvári, líce kosti a obočie. Pri opätovnom *zahustení* bola práca sústredená na detaily ako sú ostré hrany očných viečok, lemovanie nosných dierok a naznačenie vrások.

3.1.2 Retopológia

Po vymodelovaní hlavy predošlým spôsobom prišla na radu *retopológia*. Jedná sa o funkciu, ktorá slúži k prichytávaniu vertexu na povrch iného objektu. To znamená, že na vytvorený objekt bola „nanesená“ ďalšia vrstva vertexov, hrán a trojuholníkov. Podľa odporúčaného postupu boli najprv nanesené vertexy po obvode oka, potom smerom dole na nos. Následne bola trojuholníkmi vyplnená oblasť pod nosom až po krk a následným kopírovaním vertexov až po oblasť šíje sa vytvorila kompletná sieťovina vymodelovanej tváre. Následne bol vypnutý mód retopológie a skrytý predlohový objekt. Na novovytvorenom objekte pomocou retopológie bol znova aktivovaný *sculpt* mód, pomocou ktorého boli dorobené jemné detaily, ako vrásky, svaly, hrany kostí atď.

3.1.3 Normal mapping

Týmto spôsobom nám síce vznikol veľmi kvalitný a realistický model, ale bohužiaľ zložený z veľkého množstva polygónov, čiže nepoužiteľný k real-time renderingu. Bolo nutné vytvoriť model porovnateľnej kvality, ale s oveľa menším množstvom polygónov. Tento cieľ bol

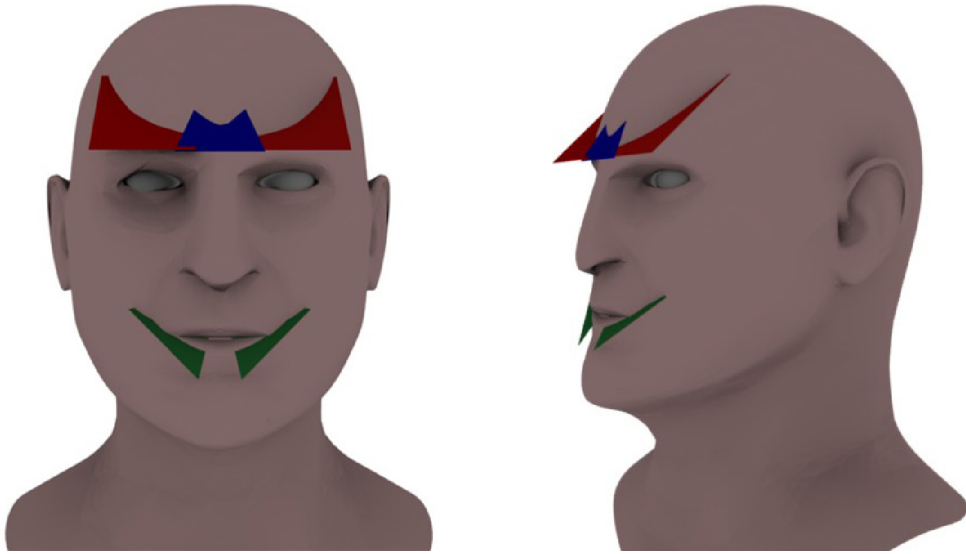
dosiahnutý použitím *normal mappingu*. Jedná sa o techniku textúrovania, ktorá vytvára ilúziu nerovnosti povrchu bez zmeny jej geometrie. Ilúzia nerovnosti sa dosahuje úpravou normály v každom pixeli plochy. Modifikovaná normála následne ovplyvní výpočet osvetlenia plochy[13]. Pridaním textúry a využitím normal mappingu vznikol nasledujúci model, do ktorého bolo potrebné už len pridať deformačné plochy (obr. 3.1).



Obrázek 3.1: Výsledný 3D model tváre

3.1.4 Vytvorenie deformačných oblastí

Deformačné plochy boli vytvorené tak, aby vyhľadávací algoritmus bol čo najefektívnejší z hľadiska doby výpočtu ako aj alokovanej pamäte. Pôvodne boli navrhnuté deformačné objekty v tvare 3D primitív ako kváder alebo ihlan. Od týchto návrhov bolo upustené po zistení, že návrh vyhľadávacieho algoritmu v priestore by bol veľmi náročný ako na vývoj, tak aj na výpočet. Tým boli vytvorené 2D plochy a umiestnené tak, aby mali pri pohľade spredu tvar trojuholníka alebo obdĺžnika a pri bočnom pohľade tvar úsečky. Tento tvar mal zaručiť zjednodušenie algoritmu, ako aj odstránenie komplikácií, ktoré by mohli nastať pri narušení tvaru úsečky plochy v bočnom pohľade. Bolo vytvorených päť základných deformačných oblastí: ľavé a pravé viečko, oblasť vnútorných okrajov viečok, ľavý a pravý kútik pier smerujúci do úsmevu(obr. 3.2).



Obrázek 3.2: Vytvorené deformačné oblasti vsunuté do modelu

3.2 Tvorba algoritmu

Program sa skladá z dvoch zdrojových súborov a jedného hlavičkového: `functions.h`, `functions.cpp` a `main.cpp`

3.2.1 Main.cpp

`Main.cpp` tvorí hlavný kód programu. Je v ňom uvedené načítanie modelu vo formáte OBJ, nadefinované príkazy, ktoré sa spustia stlačením určitej klávesy. Ďalej sa v ňom program odkazuje na funkcie, ktoré boli potrebné pri lokalizácii vertexov tváre v deformačných oblastiach (ďalej len DO), deformovaní týchto vertexov ako aj ukladaniu vytvorených výrazov do obrázkov. Ďalej sú tu nadefinované deformačné oblasti, ktorých súradnice vertexov boli nadefinované ako magické konštanty pre prípadné ľahšie doladovanie vytvorených výrazov. Po spustení aplikácie na najprv načíta model spolu s textúrou do štruktúry `model_file`. Keďže daná štruktúra obsahuje všetky objekty súboru, čiže okrem tváre a očí aj deformačné oblasti, vytvorí sa nová štruktúra `CelyModel`, ktorej každý element bude ukazovať na jeden objekt v načítanom modeli. Počet elementov v danej štruktúre bol nastavený na 8, keďže model tvorí 8 objektov: hlava, 2 oči a 5 deformačných oblastí. Keďže model načítaný toolkitom OGS má stromovú štruktúru, jednotlivé objekty sú medzi sebou oddelené a rozoznateľné podľa indexu. To bolo využité na vybratie objektov, ktoré sa mali načítať, ale nie vykresliť, čiže deformačných zón.

Po úspešnom načítaní bolo potrebné vytvoriť zoznam vertexov pre každú deformačnú oblasť. Na to boli vytvorené funkcie, ktoré riešili daný problém ako pre obdĺžnikové, tak aj pre trojuholníkové deformačné oblasti.

3.2.2 Functions.cpp

Selektovanie vertexov v deformačných oblastiach

Všetky dôležité funkcie vytvorené pre tento projekt sa nachádzajú v súbore `functions.cpp`. Po načítaní modelu sa pre každý vertex na tvári zisťuje, či a ktorej deformačnej oblasti patrí. Ako bolo spomenuté skôr, v dôsledku dvoch typov (tvarov) deformačných oblastí boli vytvorené 2 funkcie na zistenie lokalizácie vertexu v rámci `DO:is_in_deform_area_rectangle` a `is_in_deform_area_triangle`. Prvá funkcia je veľmi jednoduchá, keďže súradnice vertexu sa porovnávajú s krajnými hodnotami vertexov tvoriacich DO. Ak spracovávaný vertex spĺňa podmienku výskytu v rovine, kontroluje sa, či spĺňa podmienku výskytu aj v rámci priestoru. Vtedy sa taktiež porovná s krajnými hodnotami tvoriacimi priestor prednej časti tváre. Touto podmienkou sa vylúčia vertexy, ktoré spĺňajú prvú podmienku, ale nachádzajú sa buď príliš ďaleko alebo blízko v rámci vymedzeného priestoru. Typickým príkladom sú vertexy tvoriace zadnú časť hlavy a nachádzajúce sa v rovine úst alebo čela. Princíp lokalizácie vertexov v deformačnej oblasti tvaru trojuholníka bol implementovaný spôsobom popísaným na konci prvej kapitoly.

Princíp deformácie tváre

Po úspešnom vykreslení modelu okresaného o deformačné oblasti sa aktivuje OSG funkcia `TrackballManipulator`, ktorá umožní ovládať model v rámci celku. Čiže posúvať ho, otáčať a zoomovať. Ďalej je aktivovaná ďalšia OSG funkcia `KeyboardEventHandler`, ktorá umožní kontrolu nad aplikáciou pomocou kláves definovaním činností jednotlivým klávesám. Bol nadefinovaný zoznam kláves, z ktorého každá klávesa vykonáva jednu zo zoznamu funkcií:

- posúvať vertexy v deformačnej oblasti smerom nahor/nadol
- spraviť screenshot výrazu a uložiť ho ako JPG
- regenerovať model do pôvodného stavu (defaultný výraz)
- zvoliť iný mód zobrazovania
- zobrazíť/skryť určité objekty

Funkcie deformácií vertexov sú rozdelené rovnako ako zisťovanie vertexov v oblastiach na 2 typy deformácií: `vertex_transform_rectangle` a `vertex_transform_triangle2`.

Pri prvom type sa presúvajú vertexy v oblasti tvaru obdĺžnika. Je to jednoduchšia varianta, keďže pri tomto presune sa vertexy posúvajú len smerom nahor a do hĺbky, čiže nemenia svoju x -ovú súradnicu. Na výpočet hodnoty súradníc sa použijú 3 údaje. Súradnice tzv. `root_pointu`, ďalej **pôvodné súradnice** spracovávaného vertexu a nakoniec hodnota koeficientu posunutia Cf . Logika výpočtu spočíva v tom, že k pôvodnej z -ovej (výškovej) súradnici sa pripočíta rozdiel medzi z -ovou súradnicou root vertexu a spracovávaného vertexu, ktorý sa ešte vynásobí koeficientom Cf . V rámci hodnoty y -ovej (hĺbkovej) súradnice bol pokusmi zistený a stanovený pomer Y/Z na hodnotu $-0,1$. Touto konštantou sa vynásobí predošlá hodnota a pričíta sa k y -ovej súradnici vertexu. Znamienko mínus figuruje v konštante preto, lebo pri posúvaní vertexov nahor, resp. zvyšovaní hodnoty z -ovej súradnici sa vertex posúva do hĺbky, čiže znižuje hodnotu y -ovej súradnice. Pri deformovaní vertexov v DO v tvare trojuholníka je implementovaná technika popísaná na konci prvej kapitoly.

Vertexy tváre nachádzajúce sa vo viacerých DO

Kvôli týmto špecifickým vertexov museli byť modifikované funkcie na výpočet nových súradníc. Tie boli založené na princípe, že pri deformácii vertexov určitej deformačnej oblasti sa pre každý vertex zisťovalo, či sa náhodou nenachádza v ďalšej deformačnej oblasti. Ak sa potvrdil ďalší výskyt, vypočítali sa pre vertex nové súradnice v oboch oblastiach. Výsledná hodnota súradníc vertexu je vypočítaná ako pomer hodnôt koeficientov posunu,

$$vysledny_vertex = vertex_pom_part1 * C f_1 / total_koeff + vertex_pom_part2 * C f_2 / total_koeff \quad (3.1)$$

kde `vertex_pom_part1` je vertex vypočítaný pre DO_1 s hodnotou koeficientu $C f_1$ a druhý `vertex_pom_part2` je vypočítaný pre DO_2 s hodnotou koeficientu $C f_2$. Premenná `total_koeff` je súčet hodnôt týchto dvoch koeficientov.

Ukladanie obrázkov a zvyšné funkcie

Ak chce užívateľ uložiť vytvorenú „grimasu“, poslúži mu na to stlačenie klávesy P. To aktivuje OSG funkciu `Image`, ktorá uloží daný výraz ako obrázok vo formáte JPG. Aby nedošlo k prepisovaniu už uložených obrázkov, názov súboru obsahuje na konci okrem prefixu `screenshot` aj číselný údaj o počte sekúnd od roku 1970 po súčasnosť. Na to bola použitá funkcia `gettimeofday`, ktorá je windowsovou alternatívou obdobnej funkcie fungujúcej pod systémom Linux[10].

Ďalšou funkciou je resetovanie modelu, čiže nastavenie defaultného vertexu. Keďže sa deformovali len vertexy nachádzajúce sa v niektorej DO, aplikácia postupne prejde všetky DO a v každej nastaví pôvodnú hodnotu súradníc pre všetky vertexy v rámci DO. Model je možné zobrazovať v dvoch módoch: texture a wireframe mód. Taktiež aplikácia umožňuje zobrazovať a skryť deformačné oblasti.

Kapitola 4

Možný budúci vývoj

Úlohou práce bolo vytvoriť aplikáciu schopnú editovať mimické výrazy človeka. Táto práca má široké spektrum ďalšieho možného vývoja, preto v nasledujúcich riadkoch opíšem aspoň niektoré z nich.

4.1 Pokračovanie vo vývoji

Vzhľadom k náročnosti práce bol implementovaný len zjednodušený algoritmus vektorového svalu. Najväčší dôraz sa kládol na vymodelovanie čo naj dôveryhodnejšieho modelu tváre, ktorý by mohol byť využiteľný pri ďalšom vývoji danej aplikácie. Nakoľko sa nejedná o jednoduchý model, v ktorom by sa výsledná nereálnosť deformácie vytratila, pri implementovaní jednoduchých algoritmov na zložité modely môžu vzniknúť určité nereálne výrazy. Tieto nedostatky by som rád vyriešil v ďalšom vývoji aplikácie. Pohyby tváre ako napr. otváranie a zatváranie úst a očí si vyžadujú pri využití techniky deformačných oblastí náročnejšie algoritmy, ktoré zahrňujú posun len niektorých vertexov nachádzajúcich sa v deformačnej oblasti. Ďalšou alternatívou je namodelovanie sofistikovanejších oblastí, ktoré by presne vymedzovali okruh vertexov, s ktorým by bolo možné manipulovať. Týkalo by sa to hlavne oblastí úst, očí a nosa. To by si následne vyžadovalo implementáciu náročnejšieho algoritmu, ktorý by zisťoval množinu vertexov patriacu do danej deformačnej zóny ako aj prepracovnejší algoritmus určujúci premiestňovanie vybraných vertexov. Samozrejmosťou by bolo aj kvantitatívne zastúpenie deformačných oblastí. Keďže v tomto projekte bol definovaný len ich zlomok, boli by doplnené oblasti zodpovedné za pohyb sánky, očí, vrásnenie lícných svalov atď. Avšak deformačné oblasti by mohli byť doplnené pomocou špeciálnych čiernobielych textúr, ktoré by podľa farby rozoznali statické a pohyblivé vertexy. Ďalším náročnejším, ale za to oveľa kvalitnejším riešením, by bola implementácia svalových vektorov, kde by sa jednotlivé vertexy presúvali v závislosti od uhlovej vzdialenosti k daného vektoru. V grafickom programe by sa domodelovalo pár bodov reprezentujúcich imaginárne svaly, ktoré by aplikácia rozoznala, usporiadala a podľa určitých sofistikovaných algoritmov posúvala vertexy po určitých krivkách.

4.2 Vývoj v oblasti výukových aplikácií

4.2.1 Príprava na pracovný pohovor

Odvolaávajúc sa na pracovné agentúry, takmer každá vám potvrdí fakt, že pri prijímaní uchádzača o zamestnanie je rozhodujúcim merítkom priamy pohovor s uchádzačom a hlavne to, ako sa daný uchádzač o zamestnanie na pohovor pripraví a ako ho zvládne. Existuje tzv. business etika, t.j. spôsob, akým by mal uchádzač o zamestnanie pristupovať k pohovoru u potencionálneho zamestnávateľa. Rozhodujúcim faktom je okrem hlasu a tónu konverzácie aj správny spôsob sedenia, držania tela, správna gestikulácia a nepísané pravidlá zdravého sebedovedomého vystupovania pri odpovediach na otázky pracovníčok personálneho oddelenia zamestnávateľa.

V tomto prípade by ďalší vývoj napredoval k vývoju cvičebného programu pre pracovné agentúry, ktorý by mal slúžiť ako výukový program sústredený hlavne na absolventov vysokých škôl, ktorí ešte nepodstúpili priamu konfrontáciu pracovného pohovoru. Vývoj by zahrnul detailnejšie spracovanie ľudskej tváre, domodelovanie celého tela, rozšírený zoznam mimických výrazov ako aj pridanie gest. Na vytvorenie danej aplikácie by bol využitý implementovaný **GameEngine** v Bledneri spolupracujúci so skriptovacím jazykom Python.

4.2.2 Kriminalistika

V dnešnej dobe sa veľký význam vyšetrovania trestných činov sústreďuje na vypočúvanie svedkov a sledovanie ich správania, keďže počas vypočúvania sa môžu dostať pod ťarchou správne vedených otázok do stresu. Tým môže potencionálny páchatel spraviť nechcený mimický pohyb alebo gesto, ktorým pomôže vyšetrovateľom overiť pravdivosť jeho tvrdení. Ďalším možným vývojom by teda bolo vytvorenie výukovej aplikácie na psychológiu v kriminalistike. Jej vývoj by bol takmer totožný s predošlým. Jediný rozdiel by bol v spôsobe modelovania mimiky a gest. Aplikácia by mala dôverne simulovať možné spôsoby správania vypočúvaného ako napr. smer pohľadu očí dotyčného pri odpovedí, množstve a spôsobe vyklepania cigarety a mnoho ďalších dôležitých detailov.

4.3 Virtuálna moderátorka

Vývoj by sa však mohol uberať úplne odlišným smerom. Ďalšou alternatívou by mohol byť vývoj tzv. virtuálnej moderátorky. Jednalo by sa o osobu, ktorá by „čítala“ každodenné aktuálne správy. Jej mimický výraz by sa formoval na základe charakteru každej správy. Jednalo by sa o aplikáciu, ktorej vstupné údaje by boli textové súbory obsahujúce text správy a určitý príznak charakteru správy, či sa napr. jedná o správu veselú, tragickú alebo emočne neutrálnu. Pri vývoji danej aplikácie by sa práca sústredila hlavne okolo modelovania pohybu pier, jazyka a zubov počas reči, na čo by sa využil implementovaný *GameEngine* v Blenderi.

Kapitola 5

Záver

Úlohou práce bolo vytvoriť aplikáciu schopnú editovať mimické výrazy človeka. Daná aplikácia načíta 3D model vo formáte OBJ, ktorý bol vytvorený pomocou open-source modelovacieho softwaru Blender. Okrem modelu tváre obsahuje súbor objekty zastávajúce funkciu deformačných oblastí. Renderovací algoritmus s využitím knižnice OpenSceneGraph vykreslí daný model na obrazovku a prednastavený zoznam kláves umožní deformáciu určitých častí tváre. Daná deformácia je spôsobená premiestňovaním určitej skupiny vertexov nachádzajúcej sa v jednotlivých deformačných zónach. Tieto zóny boli kvôli zjednodušeniu namodelované ako jednoduché 2D geometrické objekty (trojuholník, obdĺžnik). Ich tvar a veľkosť boli zvolené tak, aby sa vytváraný mimický výraz podobal skutočnému mimickému výrazu. Vzhľadom k náročnosti projektu sa mi podarilo implementovať len minimum mimických výrazov, resp. deformačných oblastí. Neboli implementované náročnejšie algoritmy zodpovedné za otváranie/zatváranie úst alebo očných viečok, ani vytvorené sofistikovanejšie deformačné oblasti pomocou zložitých geometrických objektov. Riešenie týchto nedostatkov, resp. ďalší možný vývoj je spomenutý v predošlej kapitole.

Celková práca spojená s vytváraním danej aplikácie bola pre mňa osobne veľkým prínosom. Po naštudovaní množstva dokumentácií som sa naučil užívateľsky používať modelovací software pre prácu s 3D grafikou Blender. Ďalej som získal viac skúseností v programovaní v C++ a hlavne OpenSceneGraph. Osvojil som si odporúčané základné techniky pri tvorbe 3D modelu a programovania, ktoré by som rád v ďalšom štúdiu vylepšoval.

Literatura

- [1] Bergeron, P.; Lachapelle, P.: Controlling facial expressions and body movements. In *Advanced Computer Animation, SIGGRAPH '85 Tutorials, Volume 2*, ACM New York, 1985, s. 61–79.
- [2] Deng, Z.; Neumann, U.: *Data-Driven 3D Facial Animation*. Springer, 2007, 296 s., iSBN 1-846-28906-8.
- [3] Ekkman, P.: *The argument and evidence about universals in facial expressions of emotion*. Chichester: Wiley, 1989, in H. Wagner & A. Manstead (Eds.), *Handbook of social psychophysiology* (pp. 143-164).
- [4] Faigin, G.: *The Artist's Complete Guide to Facial Expression*. Watson-Guptill, 1990, iSBN 0-823-01628-5.
- [5] Hjortsjö, C.-H.: *Man's Face and Mimic Language*. Luden, 1970, 156 s.
- [6] Jerald, J.: Muscle based facial animation.
<http://www.cs.unc.edu/~lin/COMP259-S04/LEC/23a.ppt>, 2004-04-07.
- [7] Magnenat-Thalmann, N.; Minh, H.; deAngelis, M.; aj.: *Design, transformation and animation of human faces*. 1989, 32-39 s.
- [8] Magnenat-Thalmann, N.; Primeau, N. E.; Thalmann, D.: *Abstract muscle actions procedures for human face animation*. 1988, 290-297 s.
- [9] WWW stránky: Analytická geometrie - poloha bodu voči prímcu.
<http://maths.cz/clanky/analyticka-geometrie-poloha-bodu-vuci-primce.html>.
- [10] WWW stránky: Gettimeofday function for Windows.
<http://www.openasthra.com/c-tidbits/gettimeofday-function-for-windows/>.
- [11] WWW stránky: Object Files (.obj).
<http://local.wasp.uwa.edu.au/~pbourke/dataformats/obj/>.
- [12] WWW stránky: Oficiálna stránka projektu Blender. <http://www.blender.org>.
- [13] WWW stránky: Webová encyklopédia Wikipédia. <http://wikipedia.org>.
- [14] WWW stránky: Animation With Cg [online].
http://www.gamasutra.com/features/20030325/fernando_04.shtml, 2003-03-25.

Dodatek A

Obsah CD

- aplikácia + manuál
- zdrojové súbory aplikácie
- zdrojové súbory modelu a použité textúry
- výstupná správa + zdrojové súbory
- plagát

Dodatek B

Manuál

B.1 Spustenie programu

Aplikácia sa spúšťa súborom `projekt.exe`. Ostatné súbory, ako model, textúry a potrebné DLL sa musia nachádzať v rovnakom adresári.

B.2 Ovládanie programu

Po spustení programu sa zobrazí okno, v ktorom sa vykreslí načítaný model. Defaultné zobrazenie modelu pozostáva z priameho pohľadu na otextúrovanú tvár bez zobrazených deformačných oblastí (viď. obr. B.1). Veľkosť okna sa nastavuje kliknutím myši na pravý dolný okraj okna.

Nasledujúca tabuľka B.1 obsahuje zoznam kláves a ich jednotlivých funkcií. Obrázky z animácie sa ukladajú do formátu JPG a aby nedošlo k ich prípadnému prepísaniu, názov súboru obsahuje postfix v tvare číselnej hodnoty sekúnd od roku 1970 po súčasnosť.

Klávesa	Funkcia
Q/A	dvíha/znižuje oblasť pravého obočia
W/S	dvíha/znižuje oblasť stredu obočí
E/D	dvíha/znižuje oblasť ľavého obočia
R/F	posúva smerom od/do úst oblasť pravého kútika pery
T/G	posúva smerom od/do úst oblasť ľavého kútika pery
P	spraví screenshot
X	regeneruje model do pôvodného stavu
C	wireframe mode
V	textured mode
N	zobrazí deformačné oblasti
M	skryje deformačné oblasti
LM+Up/Down/Left/Right	rotuje objektom
RM+Up/Down	zoomuje out/in objekt
MM(holding)	presúva objekt
ESC	exit

Tabuľka B.1: Tabuľka prezentuje zoznam kláves a k nim prislúchajúce funkcie



Obrázek B.1: Úvodná obrazovka aplikácie

Dodatek C

Plagát