

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 703

ANIMACIJA POKRETANA PRAĆENJEM LICA

Toni Kork

Zagreb, lipanj 2014.

Zagreb, 9. ožujka 2014.

DIPLOMSKI ZADATAK br. 703

Pristupnik: **Toni Kork (0036405624)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Animacija pokretana praćenjem lica**

Opis zadatka:

Praćenje lica je postupak kojim se tehnikama računalnog vida slijedi ljudsko lice i njegove ključne točke u video slici. Ova tehnologija može se koristiti za pokretanje animacije lica virtualnog lika, što je poznato i pod engleskim nazivom performance animation. Potencijalne primjene su u području igara, zabave i komunikacija.

Na Zavodu za telekomunikacije dostupan je sustav za praćenje lica u stvarnom vremenu. Potrebno je odabrati odgovarajući grafički sustav (preporučeno je Torque 3D) te ga na odgovarajući način povezati sa sustavom praćenja lica.

Vaša je zadaća proučiti postojeće sustave, te predložiti i implementirati izvedbu sustava za animaciju pokretanu praćenjem lica.

Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 14. ožujka 2014.

Rok za predaju rada: 30. lipnja 2014.

Mentor:


Prof. dr. sc. Igor Sunday Pandžić

Djelovođa:


Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
diplomski rad profila:


Prof. dr. sc. Siniša Srbljić

Zahvaljujem nepoznatim pojedincima u Torque 3D zajednici koji su me uputili na potrebne funkcije u pokretačkom sustavu bez koji ne bi bilo moguće napraviti animaciju

Zahvaljujem Anthoniju Wardu na modelu Kila, koji se izvrsno uklopio u ovaj projekt

Zahvaljujem asistentima Miroslavu Frljaku i Petru Stojancu na ukazanoj pomoći

Sadržaj

Uvod	1
1. Visage SDK	2
2. Torque 3D.....	3
3. Integracija sustava	5
3.1. Ideja	5
3.2. Implementacija	5
3.2.1. Priprema za prevođenje	5
3.2.2. Povezivanje Torque 3D-a i Visage SDK-a.....	6
3.3. Problemi pri implementaciji	7
4. Animacija	9
4.1. Model.....	9
4.2. Mapiranje izraza lica	10
4.3. Implementacija animacijske funkcije	10
4.4. Problemi u mapiranju	11
Zaključak	13
Literatura	14
Sažetak.....	15
Summary.....	16
Skraćenice.....	17
Privitak	18

Uvod

Porastom računalne snage, digitalna obrada videa dosegla je mogućnost prepoznavanja ljudskih lica i izraza u realnom vremenu. Iz prijenosa kamerom uživo moguće je odrediti broj i položaj ljudi na slici, kao i njihove izraze lica. Iako je tehnologija još uvijek u razvoju, trenutni rezultati su vrlo dobri za svakodnevne potrebe. Za filmsku industriju kvaliteta prepoznavanja još nije dovoljno dobra.

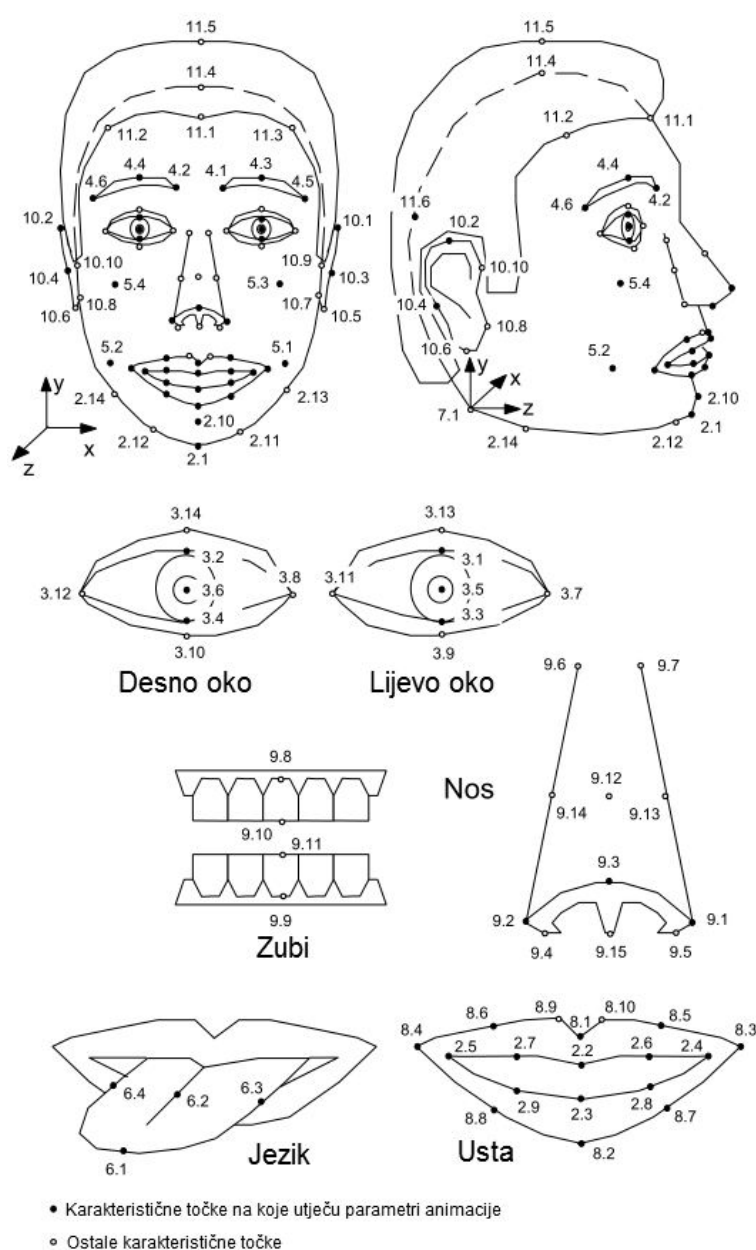
Jedna od mogućnosti praćenja lica, jest animiranje virtualnog lika tako da on prati izraz lica operatera pred kamerom. Na taj način moguće je brzo i jednostavno spremati animacijske sekvence za računalne igre, ili koristiti te podatke u video konferenciji. U video konferenciji, umjesto cijelog videa, mrežom se prenose samo animacijski parametri, koji ne trebaju mnogo mrežne propusnosti za potpuni prijenos. Na drugoj strani animira se model osobe sa kojom je uspostavljena veza. To je formalno definirano standardom MPEG4-FBA.

Ova demonstracija animacije lica u realnom vremenu koristi Visage SDK softverski paket za prepoznavanje i plaćanje lica i Torque 3D pokretački sustav (engl. *engine*) za izradu 3D igara iz prvog lica (engl. *First Person Shooter* - FPS). Web kamerom se prati lice igrača, a njegov izraz lica preslikava se na virtualni lik kojim igrač upravlja. U mrežnoj igri, uz pomoć aplikacije za prijenos glasa preko interneta, igrači na ovaj način ostvaruju video konferenciju. Budući da virtualni likovi ne moraju biti nalik na stvarne osobe koje ih upravljaju, osigurava se djelomična anonimnost (a ako je glasovni prijenos izobličen, tada je anonimnost potpuna).

1. Visage SDK

Visage SDK (*Software Development Kit*) je paket biblioteka koje omogućuju ugradnju funkcionalnosti prepoznavanja i praćenja lica iz slike, videa ili uživo sa kamere u samostalnoj aplikaciji. Visage SDK koristi tehnike računalnog vida kojima slijedi ljudsko lice i njegove karakteristične točke u video slici.

Praćene karakteristične točke lica određene su standardom MPEG4-FBA (Slika 1.1).



Slika 1.1 Raspored karakterističnih točaka na licu prema MPEG4-FBA standardu

2. Torque 3D

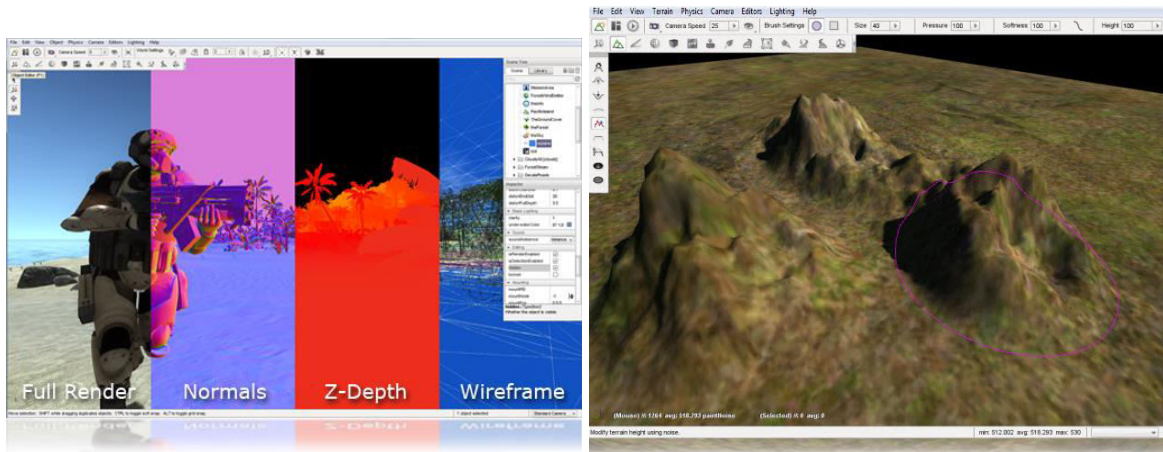
Pokretački sustav Torque 3D izvorno je razvila tvrtka Dynamix^[1], Inc. 2001. godine za igru Tribes 2. Članovi Dynamixa osnivaju tvrtku Garage Games, koju nakon niza poslovnih uspjeha i padova tvrtku otkupljuje InstantAction, koji u studenom 2010. godine najavljuju prestanak rada, te se Garage Games ponovo osamostaljuje u siječnju 2011. godine. Izvorni kod pokretačkog sustava Torque 3D objavljuju pod MIT licencom u listopadu 2012. U lipnju 2013. Garage Games službeno predaje Torque 3D u ruke konzorcija programera koji nadzire promjene nad repozitorijem koji se nalazi na GiT Hubu, te je od tada Torque 3D u potpunosti *open source* bez nadzora izvornih autora.



Slika 2.1 Logotip Torque 3D pokretačkog sustava

Mogućnosti pokretačkog sustava su mnogobrojne, a od najvažnijih se ističu^[2]: podrška za COLLADA modele as kosturnim sustavom i predefiniranom animacijom, napredni model osvjetljenja sa mapama normala i sjenama, podrška za PhysX i Bullet simulacijske sustave za fiziku objekata u igri, tereni oblikovani visinskim mapama, vozila, zgrade i drugi statički ili interaktivni objekti te mogućnost mrežne igre za do 64 igrača istovremeno.

Torque 3D sustav radi na principu klijent-poslužitelj, gdje klijenti šalju poslužitelju svoje radnje, a poslužitelj sinkronizira klijente. Radi smanjenja mrežnog prometa koristi se predviđanje i korekcija prilikom simulacije, princip poznat kao *dead reckoning*.



Slika 2.2 Faze iscertavanja i urednik visinskih mapa za teren



Slika 2.3 Podrška za Nvidia PhysX simulacijski sustav



Slika 2.4 Demonstracijska scena sa mnogo detalja koji se iscertavaju u stvarnom vremenu

3. Integracija sustava

Kako Torque 3D pokretački sustav nije unaprijed predviđen za rad sa Visage SDK sustavom, trebalo je proširiti izvorni kod Torque 3D-a dodacima koji integriraju funkcionalnost praćenja i animiranja lica.

3.1. Ideja

Visage SDK može vraćati parametre praćenja lica u obliku tzv. akcijskih jedinica ili u obliku pozicija karakterističnih točaka lica. Budući da Torque 3D ne podržava modele promjenjivih oblika (engl. *blendshapes*) koji koriste akcijske jedinice, nego modele sa kosturnom animacijom (engl. *bone rig*), korišteni su podaci o pozicijama karakterističnih točaka lica. Kako je model virtualnog lika napravljan tako da na licu ima mnogo kostiju koje otprilike odgovaraju karakterističnim točkama definiranim MPEG4-FBA standardom, treba samo napraviti mapiranje pozicija tih točaka na kosti u modelu.

3.2. Implementacija

Iako je pokretački sustav Torque 3D moguće prevesti i izvoditi na više platformi^[2], radi pojednostavljenja instalacijske procedure i lakše integracije sa Visage SDK sustavom, odabrana je Microsoft Windows platforma.

3.2.1. Priprema za prevođenje

Izvorni kod pokretačkog sustava Torque 3D unaprijed je pripremljen za prevođenje sa razvojnim sustavom Microsoft Visual Studio 2008/2010. Kako bi prije prvog prevođenja bili sigurni da su sve staze do potrebnih biblioteka i zaglavlja ispravne, treba pokrenuti generator rješenja i projekata. Generator rješenja i projekata je PHP skripta koja uključuje mnogobrojne module, koji se navode u konfiguracijskoj datoteci Torque 3D projekta. Npr. PhysX, LeapMotion, Razer Hydra, i dr. koji se po potrebi uključuju u projekt ovisno o platformi i ulaznim jedinicama.

Radi smanjenja duljine staza, pretpostaviti ćemo da je Torque 3D pokretački sustav instaliran u D:\Torque3D, iako ga je moguće instalirati bilo gdje na disku.

Generator rješenja i projekata prvo treba proširiti modulom za Visage SDK. U datoteku D:\Torque3D\Tools\projectGenerator\modules\visage.inc zapisuju se relativne staze do potrebnih datoteka, u odnosu na korijenski imenik Visage SDK paketa, po uzoru na zapis ostalih modula. U konfiguracijsku datoteku projekta treba uključiti novi modul visage, navesti stazu do korijenskog imenika Visage SDK paketa i pokrenuti generator.

3.2.2. Povezivanje Torque 3D-a i Visage SDK-a

Torque 3D sustav ima implementiran vlastiti skriptni jezik, Torque Script, za proširivanje funkcionalnosti sustava i izradu interaktivnih misija igre. Povezivanje Torquea i Visagea je hibrid proširenja izvornog koda Torque 3D sustava i skripti napisanih u Torque Script jeziku.

Svaka naredba, metoda ili elementarna funkcija Torque Script jezika definirana je u Torque 3D sustavu kao C++ funkcija ili metoda klase. Svaka od njih registrirana je u tablici funkcija. Kako bi registrirali vlastitu funkciju za korištenje u Torque Script jeziku, treba pozvati makro `DEFINE_ENGINE_FUNCTION` sa odgovarajućim parametrima: tip povratne vrijednosti, lista ulaznih parametara sa opcionalnim zadanim vrijednostima, kratki opis upotrebe funkcije i tijelo same funkcije.

U svrhu korištenja sustava za praćenje lica, definirane su nove Torque Script funkcije:

- `VisageInit()`
 - Inicijalizacija Visage sustava za praćenje lica
- `VisageUnload()`
 - Zatvaranje sustava za praćenje lica
- `VisageStartTracking()`
 - Pokretanje praćenja lica iz videa sa web kamere
- `VisageStopTracking()`
 - Zaustavljanje praćenja lica
- `VisageIsTrackerActive()`
 - Vraća trenutno stanje aktivnosti sustava za praćenje lica
- `VisageGetTrackingData(faceBones)`
 - Dohvaćanje podataka o praćenom licu – parametar `faceBones` je rječnik sa nazivima kostiju modela kao ključem, i grupom i indeksom u MPEG4-FBA modelu, kao vrijednošću

- `faceTrackedPlayerAnimate(playerID, faceBones, faceScales, faceOffsets, debug)`
 - postavlja model virtualnog lika na izraz lica dohvaćen sa sustava za praćenje – parametri su: `playerID`: identifikator objekta koji se animira, `faceBones`: rječnik za kostima modela (isto kao i za funkciju `VisageGetTrackingData`), `faceScales`: rječnik sa MPEG4-FBA grupom kao ključem i `x`, `y`, `z` vrijednostima faktora za skaliranje kao vrijednošću, `faceOffsets`: rječnik sa odmacima (po istom principu kao i `faceScales`), i logička vrijednost `debug` koja određuje da li će se, ili ne, ispisivati detaljni podatci o praćenju i animaciji u konzolu igre

Funkcije se nalaze u `D:\Torque3D\My Projects\FPS Tutorial\source\Visage\visage.cpp`. Za povezivanje likova iz demo igre sa Visage sustavom, napravljene su slijedeće funkcije ili proširenja postojećih funkcija u Torque Script jeziku:

- Nova skripta `tracker.cs` u kojoj se inicijalizira sustav praćenja lica i definiraju rječnici koji povezuju kosti lica modela sa MPEG4-FBA karakterističnim točkama, skalama i odmacima pojedinih grupa karakterističnih točaka, kako bi se podaci sustava za praćenje prilagodili modelu virtualnog lika
- Proširenje skripte `init.cs` pozivom za izvršavanje skripte `tracker.cs`
- Proširenje funkcije `GameConnection::initialControlSet(%this)` u skripti `serverConnection.cs` gdje se postavlja globalna varijabla `$PlayerID` na identifikator virtualnog lika kojim upravlja igrač
- Proširenje skripte `default.bind.cs` funkcijama:
 - `GrabFaceData()` koja se izvodi svakih 50ms ukoliko je sustav za praćenje lica aktivan i postavlja lice modela prema podacima primljenim iz sustava za praćenje pozivajući funkciju `faceTrackedPlayerAnimate` odgovarajućim parametrima.
 - `ToggleVisageTracker(%val)` koja uključuje i isključuje praćenje lica ovisno o trenutnoj aktivnosti sustava
 - Mapiranjem tipke F12 za poziv `ToggleVisageTracker` funkcije

3.3. Problemi pri implementaciji

Uz Torque 3D sustav distribuira se korisnička dokumentacija, koja je prvenstveno namijenjena krajnjim korisnicima, koji kane sustav koristiti u obliku kakvom se distribuira. Za sve one koji žele proširivati sustav vlastitim dodacima, dokumentacija je vrlo oskudna i uglavnom svedena na komentare u izvornom kodu Torque 3D sustava. Zbog toga je utrošeno mnogo vremena na metodu pokušaja i pogrešaka.

Zbog vremenskih ograničenja, implementirana je samo lokalna animacija, na jednom računalu. Daljim proučavanjem Torque 3D sustava, animacija se može proširiti na videokonferenciju među igračima u mreži.

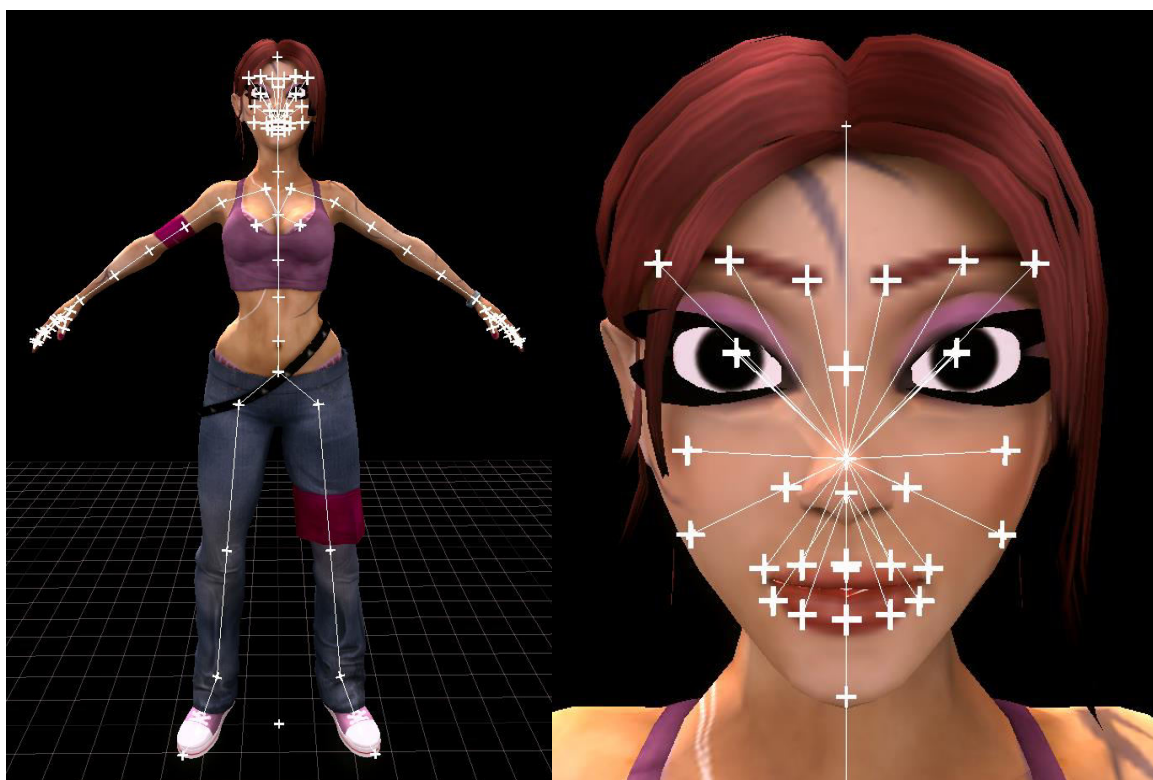
Za vrijeme razvoja postojao je problem rušenja aplikacije ukoliko bi se pokrenulo praćenje lica bez kamere. Upotreba novije inačice Visage SDK paketa i implementacija objekta za vraćanje poruke o grešci, riješili su taj problem.

4. Animacija

Torque 3D sustav nije predviđen za rad sa proceduralnim animacijama ili animacijama u stvarnom vremenu, već koristi predefinirane animacijske sekvence spremljene u datoteci modela. No da bi sustav mogao animirati modele po predefiniranim animacijama mora posjedovati funkcionalnost animiranja kao takvu. Korištenjem tih elementarnih funkcija moguće je ostvariti vlastiti oblik animacije.

4.1. Model

Upotrijebljeni model virtualnog lika_[4] napravljen je od poligonske mreže i poduprt sustavom kostiju. Za ovaj model nije napravljena nikakva predefinirana animacija.



Slika 4.1 Model u punom planu (lijevo) i detalj kostiju lica (desno)

Model je izvorno dostupan u internom formatu alata Autodesk Maya, te ga je trebalo konvertirati u COLLADA format prije uvoženja u Torque 3D.

4.2. Mapiranje izraza lica

Ako je sustav za praćenje lica uspješno inicijaliziran i aktiviran, animacija lica obavlja se nad onim kostima lica za koje je sustav za praćenje lica uspješno odredio poziciju karakterističnih točaka. Ukoliko je dio lica zaklonjen, ostaje u zadnje zatečenom stanju, isto vrijedi i za cijelo lice ako je zaklonjeno ili van slike.

Od Visage sustava podaci o praćenju dohvaćaju se u obliku relativnih pozicija karakterističnih točaka lica. Visage sustav koristi desni koordinatni sustav definiran OpenGL specifikacijama i metar kao osnovnu mjernu jedinicu. Vrijednosti su relativne u odnosu na ishodišne koordinate glave, koje su otprilike na središtu između očiju praćenog lica.

Torque 3D sustav koristi lijevi koordinatni sustav definiran DirectX specifikacijama i metar kao osnovnu mjernu jedinicu. Koordinate kostiju u modelu relativne su u odnosu na ishodište modela.

4.3. Implementacija animacijske funkcije

Postavljanje modela na položaje dohvaćene sustavom za praćenje lica obavlja funkcija `faceTrackedPlayerAnimate`. Funkcija iterira kroz rječnik kostiju lica i traži indekse tih kostiju u modelu zadanim identifikatorom igrača. Ukoliko se model ne može pronaći, ispisuje se poruka o grešci, a ako se kost u modelu ne može pronaći, preskače se.

Ako je kost pronađena, dohvaćaju se njezina MPEG4-FBA grupa i indeks, te se dohvaćaju podaci o praćenju te karakteristične točke. Ukoliko je sustav za praćenje lica ispravno prepoznao karakterističnu točku, obavlja se skaliranje vrijednosti i dodaje odmak, te se kost modela postavlja na novu poziciju (samo translacija). Ukoliko karakteristična točka nije prepoznata u videu, kost lica ostaje na zadnje postavljenom položaju.

Prilikom skaliranja vrijednosti dohvaćenih iz sustava za praćenje lica, treba zamijeniti vrijednosti y i z koordinate te negirati vrijednost x koordinate kako bi se obavila konverzija iz desnog u lijevi koordinatni sustav korišten u Torque 3D sustavu za iscrtavanje. Zamjena vrijednosti y i z koordinata implementirana je u samoj animacijskoj funkciji `faceTrackedPlayerAnimate`, dok je negacija vrijednosti x koordinate dio faktora za skaliranje po x osi.

4.4. Problemi u mapiranju

Iako je korišten model u demo igri u potpunosti prožet kostima, kosti lica nisu napravljene prema MPEG4-FBA standardu u potpunosti (Slika 4.1 desno). Gornja usnica i obrve imaju 1:1 podudaranje u broju kostiju i njihovim pozicijama, ali donja usnica, nos i obrazi, nisu potpuno usklađeni. Donja usnica ima višak kostiju, koje nisu animirane, no unatoč tome, to se ne primjećuje znatno. Donja usnica je u hijerarhijskoj strukturi modela podobjekt donje čeljusti. Odmaci su prilagođeni uspravnom liku, umjesto se referenciraju na ishodište glave, jer su koordinate kostiju relativne u odnosu na ishodište modela, pa je vrijednosti odmaka tako lakše ugoditi. To u ovoj izvedbi animacije ne pravi problem budući da lik nema nikakvih drugih animacija. Ali bi moglo praviti problem ako se doda animacija tijela (hodanje, trčanje, i dr.). K tome animacija je napravljena tako da koristi samo translaciju, što je dovoljno dobro za većinu lica, ali ne i za donju čeljust, koja uz translaciju, obavlja i rotaciju prilikom otvaranja usta (Slika 4.2).



Slika 4.2 Greška pri animaciji ako se koristi samo translacija (desno) za animiranje donje čeljusti

Sam sustav za praćenje lica neke elemente prati bolje od drugih. Tako su usnice i obrve vrlo dobro praćene, ali obrazi vrlo slabo. Vertikalni pomak čeljusti je dobro registriran, dok horizontalni nije.

Iako model podržava animiranje očiju, u ovoj implementaciji one nisu animirane. Razlog tome je što sustav za praćenje lica ima problema sa određivanjem otvorenosti očiju, a animiranje očiju bi trebalo koristiti samo rotaciju, dok se u animacijskoj funkciji, radi jednostavnosti implementacije, koriste samo translacije.

Model također podržava i animaciju jezika, koja nije implementirana zato što sustav za praćenje lica ne podržava praćenje jezika.

Teško je specificirati koje se kosti isključivo transliraju, koje rotiraju, a koje oboje prilikom animiranja. U tu svrhu razvijen je princip akcijskih jedinica koje opisuju što se događa na licu u kojoj mjeri, te je ukupni izraz lica linearna kombinacija akcijskih jedinica. Ovakav način je pogodan za modele se promjenjivim oblicima (engl. *blendshapes*), no takvi modeli nisu podržani u Torque 3D sustavu.

Zaključak

Ovim radom demonstrirano je integriranje sustava Visage SDK i Torque 3D u svrhu ostvarenja animacije virtualnih likova temeljenoj na praćenju lica. Ukratko su opisana najvažnija svojstva navedenih sustava i napravljena je aplikacija koja ih koristi.

Zbog oskudnosti dokumentacije, mnogo je vremena utrošeno na istraživanje sustava Torque 3D, te je zbog toga opseg rada smanjen, kako bi se mogao napraviti na vrijeme. Umjesto planiranog videokonferencijskog prijenosa animacije, napravljena je samo lokalna animacija jednog lika na jednom računalu.

Budući da je u implementaciji korištena kosturna animacija modela, primjetne su greške u animaciji lica. Greške bi bile manje da Torque 3D sustav podržava tehnike interpolacije (engl. *morph-target* ili *blendshapes*), ali se i ovom implementacijom postiže dovoljno dobra kvaliteta za računalne igre. Odabir i raspored kostiju lica igra ključnu ulogu u kvaliteti animacije, kao i preslikavanje podataka sa sustava za praćenje lica na model. Kvaliteta kamere i osvjetljenja dodatno utječu na kvalitetu prepoznavanja lica. Uz sve navedene probleme, kašnjenje animacije za pokretima operatera je primjetno, ali dovoljno malo da ne smeta u radu.

Kako je korišten sustav Torque 3D razvijen za izradu igara, ovo je jedan zanimljiv aspekt komunikacije među igračima koji je još uvijek komercijalno neiskorišten. No moguće je da ćemo u skorom vremenu naići na igre koje će primjenjivati metode opisane ovim radom u svrhu komunikacije među igračima.

Literatura

- [1] WIKIPEDIA, Torque Game Engine,
http://en.wikipedia.org/wiki/Torque_%28game_engine%29 (lipanj 2014)
- [2] GARAGE GAMES, Torque 3D product page,
<http://www.garagegames.com/products/torque-3d> (lipanj 2014.)
- [3] VISAGE TECHNOLOGIES AB, Visage SDK documentation, 2014.
- [4] ANTHONY WARD, Game Character Creation Series: Kila,
<http://cgi.tutsplus.com/series/game-character-creation-series-kila--cg-31010>
(30.5.2014)

Sažetak

Animacija pokretana praćenjem lica

Sažetak: U ovom radu ukratko se opisuju alati Visage SDK i Torque 3D i njihovom promjenom izgrađena je aplikacija koja demonstrira preslikavanje izraza lica operatera pred kamerom na virtualni lik u računalnoj igri.

Ključne riječi: 3D grafika, Visage SDK, Torque 3D, praćenje lica, animacija, kosturna animacija

Summary

Performance-driven animation

Summary: In this thesis, tools Visage SDK and Torque 3D have been briefly introduced and used for creating an application that demonstrates mapping of facial expressions from face of the operator in front of a camera onto a virtual character in a computer game.

Keywords: 3D graphics, Visage SDK, Torque 3D, facial tracking, animation, bone rigging

Skraćenice

API	<i>Aplication Programmer's Interface</i>	sučelje programerima aplikacija
SDK	<i>Software Developerment Kit</i>	paket za razvoj softvera
IEEE	<i>Institute of Electrical and Electronics Engineers</i>	
MPEG	<i>Motion Picture Experts Group</i>	radna grupa IEEE-a za video
FBA	<i>Face and Body Animation</i>	animacija lica i tijela
FAP	<i>Facial Animation Point</i>	animacijska točka lica
MIT	<i>Massachusetts Institute of Technology</i>	
FPS	<i>First Person Shooter</i>	"pucačina" iz prvog lica
PHP	<i>PHP: Hypertext Preprocessor</i>	

Privitak

Instalacija programske podrške

Potrebni alati:

- Microsoft Visual Studio 2008 ili 2010
- Microsoft DirectX SDK April 2010 ili noviji
- Web kamera sa instaliranim pogonskim softverom

Instalacija sa priloženog DVD-a:

- Ekstrahirati samoekstrahirajuću arhivu Torque3D.exe na proizvoljno mjesto na disku, npr: D:\Torque3D
- Instalirati Visage SDK na proizvoljno mjesto na disku
- Registrirati licencu Visage SDK pokretanjem **registrationmanager.exe** u Visage SDK korijenskom imeniku
- Prekopirati registriranu Visage SDK licencijeku datoteku u D:\Torque3D\MyProjects\FPS Tutorial\game

Prevođenje projekta (po potrebi):

- U datoteci D:\Torque3D\MyProjects\FPS Tutorial\buildFiles\Config**project.conf** postaviti varijablu \$VISAGE_SDK_PATH na stazu gdje se nalazi Visage SDK.
- Pokrenuti D:\Torque3D\MyProjects\FPS Tutorial**generateProjects.bat**
- Otvoriti D:\Torque3D\MyProjects\FPS Tutorial\buildFiles\VisualStudio 20xx\FPS Tutorial.sln (xx je ili 08 ili 10, ovisno o korištenoj inačici Visual Studija)
- Pokrenuti prevođenje rješenja *Build – Build Solution*

Upute za korištenje programske podrške

- Provjeriti da li je web kamera uključena i da li je okrenuta prema licu
- Pokrenuti D:\Torque3D\MyProjects\FPS Tutorial\game\FPS Tutorial.exe
- Kad se pojavi naslovni ekran, odabrati opciju World Editor i odabrati mapu China Town, te pokrenuti igru
- Kad se igra učita, u donjem desnom uglu prozora prebaciti kameru sa "1st Person camera" na "Standard Camera"
- Koristeći miša (držati desnu tipku za rotaciju pogleda ili srednju za translaciju pogleda) i tipkovnicu (WASD) pozicionirati kameru oko modela po želji
- Pokrenuti praćenje lica tipkom F12