|  |  |  |
| --- | --- | --- |
|  | UNIVERZITET U NOVOM SADU  **FAKULTET TEHNIČKIH NAUKA U NOVOM SADU** |  |

Nikola Berdić (E2-66/2015)

Autorigovanje 3D

modela pomoću „Pinocchio“

programske biblioteke

Seminarski rad

- iz predmeta Multimedijalni sistemi -

Mentor:

Prof. dr Ivetić Dragan

**1. Uvod**

U kompjuterskoj animaciji, skeletalna animacija je često korišćena kao intuitivna metoda za modelovanje. U slučaju animacije skeleta, meš modela je predstavljen sa odgovarajućim skeletom, koji određuje na koji način će se meš model promeniti u slučaju kretanja. Proces kreiranja i uklapanja skeleta sa modelom naziva se rigovanje i uglavnom se radi ručno, pomoću softvera za 3D modelovanje, kao što su Maya ili Blender.

Ručno rigovanje može biti obeshrabrujuće za početnike, ali i vrlo vremenski zahtevno za eksperte zato što svaki model mora biti rigovan odvojeno. U radu biće proučena programska biblioteku za automatsko rigovanje meš modela „Pinocchio“. Glavna namena programske biblioteke je da omogući automatsko rigovanje bez ikakve pomoći korisnika.

Zadatak seminarskog rada je da prouči „Pinocchio“ programsku biblioteku, da da ocenu njene upotrebljivosti i da je proširi tako da se obezbedi jednostavnije okruženje (sa naprednim elementima interakcije) za njeno korišćenje.

Upotrebljivost „Pinocchio“ programske biblioteke biće odrađena tako što će se proučiti sledeća tri aspekta programske biblioteke:

* Preformanse – izvršavanje procesa rigovanja pomoću programske biblioteke bi trebao da traje manje od ručnog rigovanja,
* Samostalnost – izvršavanje procesa rigovanje pomoću biblioteke bi trebao što više da smanji umešanost korisnika,
* Uopštenost – upotrebljovost biblioteke bi trebala da bude na različitim tipovima modela.

U prvom poglavlju seminarskog rada je opisan način na koji funkcioniše „Pinocchio“ programska biblioteka. U drugom poglavlju su proučena tri aspekta „Pinocchio“ programske biblioteke. U trećem poglavlju seminarskog rada opisan je programski kod koji omogućuje korišćenje „Pinocchio“ programske biblioteke i njegova funkcionalnost aplikacije. Na kraju seminarskog rada doneti su zaključci o „Pinocchio“ programskoj biblioteci uz konstatacije kako se sve ova biblioteka može primeniti.

**2. Opis „Pinocchio“ programske biblioteke**

Programska biblioteka „Pinocchio“ je kreirana 2007 godine i njeno pokretanje se vrši iz komandne linije. Ovaj pristup znatno otežava korišćenje programske biblioteke korisnicima sa prosečnim znanjem o računaru. To je razlog zbog kojeg programska biblioteka mora da se prilagodi radu na Windows 10 operativnom sistemu i da se proširi grafičkim korisničkim interfejsom koji bi omogućio lakšu konfiguraciju parametara za rigovanje.

Trenutno pokretanje programske biblioteke iz komandne linije očekuje jedan uslovni argument, a to je putanja do meš modela koji je potrebno izrigovati. Pored uslovnog argumenta, putanje do modela, moguće je dodati i opcione argumente. Opcioni argumenti mogu biti rotacija meš modela za navedene stepene, skaliranje njihovog skeleta, kao i pokrete koje bi avatar primenio nakon završetka procesa rigovanja. Primer pokretanja programske biblioteke sa uslovnim i opcionim argumentima prikazan je u komandnoj liniji 1. Putanja do meš modela „**data\cheb.obj“**  je prvi i uslovni argument do meš modela. Rotacija meš modela se menja dodavanjem argumenta **-rot 0 1 0 90** , u ovom slučaju meš model će biti zarotiran oko *y* ose za 90 stepeni. Ukoliko je potrebno skaliranje konačnok skeleta dodaje se argument **-scale 2** , koji u ovom slučaju duplira veličinu konačnog skeleta. Poslednji opcioni argument naveden u komandnoj liniji 1 je dodavanje pokreta krajnje izrigovanom avataru, **-motion motion\jumpAround.txt**. U ovom slučaju, avatar bi skakutao.

**DemoUI.exe data\cheb.obj -rot 0 1 0 90 -scale 2** **-motion motion\jumpAround.txt**

Komandna linija 1. Primer pokretanja programske biblioteke sa uslovnim i opcionim argumentima

Rad sa programskom bibliotekom „Pinocchio“ je na prethodno opisan način vrlo zahtevan i težak za korisnika sa prosečnim znanjem računara. Takođe, na ovaj način nije moguće izvršiti rigovanje više meš modela od jednom. Razvijanje grafičkog korisničkog interfejsa bi olakšao podešavanje parametara, kao i dalo mogućnost rigovanja većeg broja meš modela.

Metoda rigovanja koja je iskorišćena može da se podeli u dva uzastopna koraka. Generisanje i postavljanje skeleta unutar modela i „skinning“ modela, što predstavlja kako će se model menjati skeletnom strukturom.

**2.1. Generisanje skeleta i njegovo postavljanje**

Prvi korak prilikom rigovanja je generisanje i postavljanje skeleta. Ovaj proces se vrši na jedan od dva načina, ugrađivanjem ili izvlačenjem skeleta iz 3D modela [1]. Prilikom ugrađivanja skeleta, postoji skelet šablon koji je potrebno ugraditi u model na najoptimalniji način. Kod izvlačenja skeleta, unutrašnjost modela se proučava i na osnovu proučene unutrašnjosti skelet za taj model se izvlači. Glavna razlika između ova dva procesa je ta da za proces ugrađivanje skeleta potreban je ulazni (šablon) skelet. Samim tim ovaj način zahteva inicijalnu interferenciju korisnika.

„Pinocchio“ biblioteka koristi prvi pristup za generisanje i postavljane skeleta, a to je ugrađivanje skeleta u 3D model [1]. Ugrađivanje skeleta zahteva već kreirane skelete koji će biti ugrađeni u meš, kao i funkciju koja bi odredila optimalni način da se to uradi. Funkcija stavlja volumen rigovanja meša u drugi plan da bi računica bila efikasnija, dok je finalno postavljanje skeleta određeno metodom optimizacije optimalnih margina.

Da bi se ugradio skelet na optimalan način, prvo se izoluje postojeći volumen, tako da se ne uzima ceo prostor unutar meša kao mogućnost postavljanja skeleta. Na kraju ovog koraka pronađena je medijalna, srednja površina meša. Srednja ili medijalna površina je isto što i medijalna osa, samo u 3D prostoru. Tačka pripada medijalnoj osi ukoliko ima više od jedne najbliže tačke na površini modela. Logično, medijalna površina se sastoji od tačaka koje centrirane u modelu, gde bi se očekivala skeletalna struktura. Slika 1. skroz levo prikazuje medialnu površinu. Idealno bi bilo da se nađe prostor koji bi bio odvojeniji od medijalne površine. Zbog toga je iskorišćena metoda za kreiranje konačnog grafa vrlo slična pakovanju sfera. Svaka tačka medijalne površine, počevši od najdalje od meša, dodaje sferu gde je centar sfere ta tačka, a radijus najbliža udaljenost do meša. Sfera se ne dodaje ukoliko sadrži centar neke druge sfere. Upakovane sfere unutar meša su prikazane na slici 1., u sredini. Finalni graf je onda konstruisan, centri sfera postaju čvorovi, a grane su postavljene između preklapajućih sfera, ili ukoliko je ta grana esencijalna za dalji nastavak. Svaki čvor je potencijalna pozicija zgloba skeleta, a svaka grana je potencijalna konekcija, kost izmedju dva zgloba.



Slika 1. Izolovanje postojećeg volumena unutrašnjosti meša. Levo je prikazana medialna površina, sredina prikazuje pakovanje sfera, desno je prikazan graf. [1]

Nakon ovog procesa, potrebno je nekako „ugurati“ skelet u dobijeni graf. Da bi se našao najbolji način potrebno je na neki način negativno bodovati potencijalne zglobove i kosti. Mogući opisi nepoželjnih delova skeleta bi bile kratke kosti, neprimerena orientacija među zglobovima, dužina kosti koje su označene kao simetrične, kosti koje dele više od jednog zgloba (u skeletu ne bi trebalo da bude petlji), kost označena kao „noge“ a udaljena od poda, neprimerena orientacija kosti, kao i zglobovi preblizu jedan drugome, a udaljeni od skeleta[1]. Da bi se našao optimalni način negativnog bodovanja, uči se maximalni marginalni linearni klasifikator, koji pokušava da maksimizuje margine između najboljeg dobrog i najboljeg lošeg dela skeleta. Formula 1 predstavlja maksimalni marginalni linearni klasifikator. [1]

(1)

Gde je **p** i **q** vektor sa bodovima (sastoji se od negativnih bodova koji su malo pre spomenuti) dobrih i loših delova respektivno. Г je vektor koji sadrži težinu, a **n** i **m** su brojevi dobrih i loših delova skeleta, respektivno. Idealno bi bilo da izbor Г bude takakv da je jednačina 1 maksimizovana, tako da se napravi očigledna razdvojenost dobrih i loših potencijalnih delova skeleta.

Sama procedura učenja se zasniva na Nedler-Mid metodi. Započinje se sa nasumičnom vrednošću za težine, koje su ocenjivane jednačinom 1, što bi trebalo da se maksimizuje na setu podataka za treniranje. Ovaj deo se radi nekoliko puta sa različitim početnim težinama, jer metoda Nedler-Mid može da se zaustavi (zaglavi) u lokalnom minimumu.

Sada je moguće izračunati optimalni način pripajanja skeleta u meš. Da bi se prostor potrage smanjio još više, zglobovi koji povezuju dve kosti (kao što su kolena i laktovi) se uklanjaju iz šablonskog skeleta. Rezultat svakog parcijalnog potencijalnog pripajanja skeleta mešu je suma negativnih bodova između dva međusobno spojena zgloba dodata na negativne bodove prva dva zgloba dodata na negativne bodove na prva tri zgloba i tako nadalje sve do poslednjeg pripojenog zgloba. Algoritam staje u slučaju da je pripajanje završeno, i na taj način je osigurano da je izabrano najoptimalnije rešenje. Kada je nađeno konačno rešenje, spojeni zglobovi se ponovo dodaju tako što se razdvoje na nađene ivice u proporcije odgovarajuće originalnom skeletu.

**2.2. „Skining“ modela**

„Skining“ modela je proces prijanjanja teksture na 3D model i definiše na koji način bi se geometrijska struktura menjala prilikom promene poze skeleta. Ovaj proces može biti jedan od zahtevnijih procesa prilikom kreiranja 3D avatara, zato što je potrebno da se kompleksna 3D površina pretoči u 2D sliku (teksturu) tako da se svaka tačka mapira.

„Skining“ modela u „Pinocchio“ biblioteci je odrađen korišćenjem tehnike zvanom LBS (Linear Blend Skinning [1]), jer je jedna od najrasprostranjenijih metoda u ovoj oblasti. Ukoliko je **vj** pozicija tačke ***j****,*  **T i** je transformacija ***i-te*** koske, a ***w ij*** je težina ***i-te*** koske za tačku ***j*,** LBS nam daje poziciju transformisane tačke ***j***  kao ***∑i w ij T i (vj).*** Problem je pronalazak koske koja teži ***wi*** za tačke, koliko svaka koska utiče na transformaciju svake tačke.

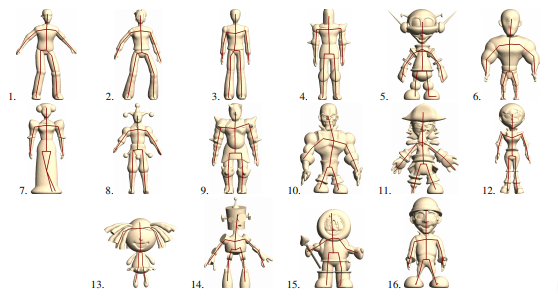
Postoje nekoliko prednosti koje se zahtevaju od težine. Prvo, ne bi trebalo da zavise na razlaganje meša. Drugo, da bi rezultat izgledao dobro, težina treba da varira blago po površini. Konačno, da bi se izbeglo savijanje artifakata, debljina između dve koske gde se dodiruju trebala bi da bude proporcionalna distanci od zgloba do površine druge koske. Iako se šema za dodeljivanje težine koski bazira na blizini, takva šema bi često bila neuspešna jer ignoriše geometriju karaktera.

**3. Upotrebljivost „Pinocchio“ programske biblioteke**

Upotrebljivost „Pinocchio“ programske biblioteke je ocenjena na proceni preformansi, samostalnosti i uspešnosti. Pod **preformansom** biblioteke smatra se brzina koja je potrebna da bi se prošao ceo proces autorigovanja. Idealno, proces bi trebao da bude brži od ručnog autorigovanja. Algoritam koji je korišćen u biblioteci, testiran je na 16 različitih modela. Najkraće potrebno vreme bilo je 12.6 sekundi za model koji sadrži 19,001 tačku. Najduže vreme je 77.1 sekundu za model od 56,865 tačaka[1]. Najviše vremena oduzima pretprocesiranje, zbog kalkulacije polja distanci. Samo „ugrađivanje“ traje od prilike jednu petinu celoukupnog vremena.

**Samostalnost** bi idealno trebala da bude što veća, jer na taj način konačni korisnik bi trebao uz minimum dodatnog truda da popravi greške algoritma. Restrikcije ili dodatna uputstva kroz proces rigovanja trebao bi biti minimalan, jer na taj način korisnik bi mogao da izvrši širok spektar mogućnosti uz minimalan broj koraka. Restrikcije kod biblioteke su vrlo velike, jer zahteva početni skelet, koji mora da se slaže sa modelom. Potrebno je da se model dosta poklapa sa nekim od šablon skeleta. Ukoliko bi ovo bilo drugačije broj negativnih bodova koji se dodeljuju kostima i zglobovima bi bio ogroman. Takođe, biblioteka očekuje da obe noge budu na dnu modela.

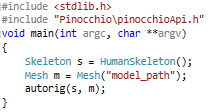
**Uopštenost** metode je vrlo visoka. Idealno bi bilo da je moguće primeniti način rigovanja na bilo koji model. „Pinocchio“ je testiran nad 16 modela koji su prikazani na slici 2. Većina modela su tačno rigovana, sem modela 7, 10 i 13. Ali njihova ispravka bi bila vrlo jednostavna, korisnik bi trebao samo da pokaže još jedan dodatni zglob. Biblioteka „Pinochio“ funkcioniše vrlo dobro u ovom pogledu, ali je vrlo ograničena na čovekolike modele. Ovaj problem bi mogao da se reši na način da se unos šablon skeleta poveća sa jednog na više, samim tim, skelet koji dobije najmanje negativnih ocena bio bi vraćen kao optimalni rezultat. Ovaj način nadograđivanja biblioteke bi, naravno, imao svoje mane. Prvo, problem bi predstavljalo vreme izvršavanja. Svako dodatni šablon skelet bi značajno uvećao vreme izvršenje algoritma, iako je rečeno da sam proces prilagođavanja skeleta oduzima samo petinu vremena celog procesa, ali u najgorem slučaju i taj proces bi mogao da se oduži i značajno da utiče na preformanse algoritma. Drugi problem je trenutna funkcija koja boduje delove skeleta. Iako je trenutna funkcija jednostavna i upotrebljiva nad različitim šablonima skeleta, bodovanje malih kosti negativno utiče na loše prilagođavanje skeleta kada su u pitanju modeli kao što su životinje sa repovima, zmije, oktopodi, ili bilo koji drugi karakter koji se sastoji od kratkih kosti. Naravno, ovo je rešivo ignorisanjem dužine prilikom traženja kosti koje su označene kao *rep*. Poslednji i najveći problem koji je nerešiv zbog same metode i načina na koji funkcioniše, je taj da šta ukoliko je najbolji mogući šablon skelet ipak nedovoljno dobar za model? Kao što je i rečeno, sama metoda „ugrađivanja“ skeleta je ograničena na šablon skelet koji mu se daje kao ulaz, i samim tim prilikom korišćenja ove biblioteke potreban je dodatni napor krajnjeg korisnika koji bi ispravio nedostatke.

Slika 2. Rezultati rigovanja 16 modela korišćenjem biblioteke.

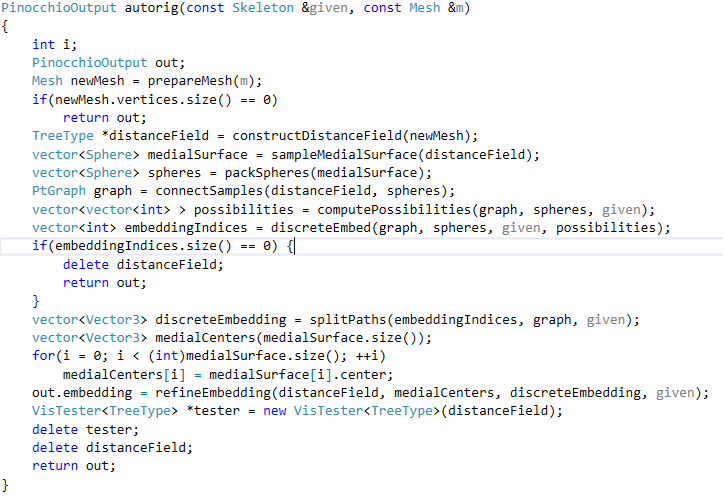
**4. Korišćenje i proširivanje „Pinocchio“ programske biblioteke**

Programska biblioteka „Pinocchio“ je slobodno za korišćenje (open source).

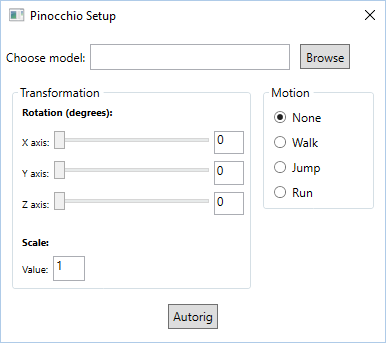
Prethodno opisana programska biblioteka „Pinocchio“ sadrži dosta metoda koje je moguće vrlo dobro iskoristiti u oblasti autorigovanja. Da bi se biblioteka upotrebila na najprostili način, moguće je izvršiti u četiri koraka koraka. Na slici 3 prikazan je najprostiji mogući način da se izvrši rigovanje nekog modela. Glavni deo koda na slici 3 je funkcija PinocchioOutput autorig(const Skeleton &s, const Mesh &m). Ova funkcija sadrži sve korake koje je potrebno izvršiti da bi se model izrigovao. Funkcija kao ulazne parametere uzima Mesh &m , što je model koji je potrebno izrigovati. Drugi parametar je Skeleton &s ovaj parametar predstavlja tip skeleta koji će model poprimiti na kraju. Klasa Skeleton predstavlja apstraktnu klasu i treba da bude ulazni skelet koji će se pripojiti modelu. Tipovi skeleta koji su definisani u biblioteci i mogu se iskoristiti kao ulazni skelet su: „CentaurSkeleton“, „FileSkeleton“, „HorseSkeleton“, „HumanSkeleton“ i „QuadSkeleton“.

Slika 3. Najjednostavniji način koišćenja biblioteke „Pinocchio“

Prethodno opisana funkcija „autorig“ predstavlja kombinaciju svih koraka koji su opisani u poglavlju 2.1. Sve funkcije koje su iskorišćene unutar „autorig“ funkcije moguće je izvršiti pojedinačno , sve to dolazi zajedno sa bibliotekom „Pinocchio“. Na slici 4 prikazana je cela funkcija „autorig“ i koraci kojima se ona izvršava.

Slika 4. Autoriging funkcija

Aplikacja koja je kreirana radi testiranja biblioteke predstavlja jednostavan i user-friendly interfejs kako bi prosečan korisnik mogao da upotrebi Pinocchio biblioteku. Na slici 4 prikazana je aplikacija. Sadrži osnovne parametre koje su potrebne korisniku da bi započeo rigovanje nekog modela. Koisnik je u mogućnosti da odabere model koji bi želeo da izriguje na dva načina, jedan je ukucavanjem pune adrese do modela ili klikom browse dugmeta koje bi mu otvorilo prozor za odabir modela. Zatim, korisnik može da izabere kako bi želeo da transformiše svoj model pomoću jedne od dve transformacije, a to su rotacija i skaliranje. Rotaciju je moguće izvršiti nad sve tri ose modela. Dok se skaliranje vrši nad autorigovanim skeletom, a ne nad modelom.

Slika 4. Interfejs aplikacija za autorigovanje modela pomoću Pinocchio biblioteke

Takođe korisnik ima opciju da odabere i pokret svog lika nakon rigovanja. Naravno, ovo nije potrebno, nego bi samo demonstriralo pokretanje modela sa prethodno definisanim pokretima. Trenutno su navedena tri: skakanje, hodanje i trčanje. Nakon pritiska dugmeta „Autorig“ počinje proces autorigovanja koji će kao rezultat prikazati željeni model. Takođe, aplikacija generiše dva fajla, „skeleton.out“ i „attachment.out“ od kojih „skeleton.out“ sadrži skelet modela, dok „attachment.out“ sadrži informacije za težinu koski. Proces autorigovanja nije dug, ne traje duže od minut, a i modeli su dosta dobro rigovani.

**5. Zaključak**

Testirana biblioteka je trenutno limitirana na karaktere sa dve noge, ali kao što je viđeno iz priloženog crtani karakteri su dobro izrigovani. Takođe, krajnji rezultat biblioteke korisnik može da popravi, za takav poduhvat korisnik ne mora biti obučeni 3D dizajner ili da zna u detalje proces 3D modeliranja. Zanimljivo bi bilo da se ova biblioteka uključi u programe koji generišu od 2D crteža 3D model sve dok nactani model je dvonog i zadrži karakteristiku i pozu sličnu početnom skeletu. Još jedna od zanimljivijih mogućih primena biblioteke bilo bi direktno rigovanje karaktera u igrama. Kao i ceo automatski proces od skeniranja nekim od uređaja za 3D skeniranje i na kraju rigovanje dobijenog 3D modela, takav avatar bi onda mogao da se implementira unutar igre, filma ili nekog video spota.

U ovom seminarskom opisana je jedna od starijih metoda za automatsko rigovanje 3D modela koja se zasniva na ugrađivanju već predefinisanog skeleta unutar 3D modela. Ručno rigovanje 3D modela može biti mukotrpan proces za početnike u ovoj oblasti. Dosta radova i istraživanja je urađeno na ovu temu i sam proces. Prvo je prikazan način na koji funkcioniše biblioteka i kakve algoritme koristi. Zatim smo je ocenili na tri prethodno definisana aspekta, preformanse, uopštenost i samostalnost. Potom je opisan i dat primer njene primene i opisani zaključci do kojih se došlo.

**6. Literatura**

[1] Ilya Baran and Jovan Popovi´c. Automatic rigging and animation of 3d characters. ACM Trans. Graph., 26(3), July 2007.